

Операционная система CP/M

**М. Уэйт
Дж. Ангермейер**

Операционная система

CP/M

**Операционная
система
CP/M**

CP / M® BIBLE

**The Authoritative Reference Guide
To CP / M®**

BY

Mitchell Waite and John Angermeyer

Howard W. Sams & Co., Inc.
4300 WEST 62ND ST. INDIANAPOLIS, INDIANA 46268 USA

М. Уэйт, Дж. Ангермейер

Операционная система

CP/M

Перевод с английского А. И. Стеркина
Под редакцией Б. А. Ашкинази



МОСКВА «РАДИО И СВЯЗЬ» 1986



ББК 32.973

у97

УДК 519.68

Редакция переводной литературы

Уэйт М., Ангермейер Дж.

у97 Операционная система *CP/M*: Пер. с англ. — М.: Радио и связь,
1986. — 312 с., ил.

Книга американских авторов является руководством по операционной системе *CP/M* и содержит материал по командному языку, утилитам, языкам программирования и трансляторам. Отмечены особенности различных версий и вариантов системы для различных конфигураций микро-ЭВМ.

Для инженерно-технических работников, специализирующихся в области вычислительной техники и программирования.

у 2405000000-123
046 (01)-86 148-86

ББК 32.973

© 1983 by Mitchell Waite

© Перевод на русский язык, примечания переводчика и редактора.
Издательство "Радио и связь", 1986

ПРЕДИСЛОВИЕ

"Управляющая программа для микропроцессоров" *CP/M** относится к классу программ, широко используемых в персональных и микро-ЭВМ и называющихся "Дисковая операционная система". Вследствие того, что операционная система (ОС) *CP/M* сравнительно давно появилась на рынке, сегодня почти каждый производитель микро-ЭВМ, которые оснащены накопителями на гибких дисках, предлагает модели, в которых используется эта операционная система. Хотя ОС *CP/M* чрезвычайно популярна (во всем мире насчитывается около 300 000 установок, на которых она используется, и свыше 500 организаций поставляют совместимые с ней программы), она непривычна для неспециалистов, что усложняет процесс ее изучения. Кроме того, существуют сотни прикладных, сервисных программ и трансляторов с языков высокого уровня, ориентированных на работу в среде ОС *CP/M*, но имеющих свои собственные правила управления. Это приводит к очевидной необходимости иметь полное, понятно написанное и удобное в использовании руководство по ОС *CP/M* и наиболее распространенным программам.

Таким руководством является данная книга, в которой представлена информация как о текущей версии 2.2 *CP/M-80*, так и о более ранних версиях 1.3 и 1.4. Кроме того, здесь представлен краткий обзор версии 3.0 *CP/M-80*. Книга содержит подробные сведения о различных командах и транзитных программах ОС *CP/M*. Здесь детально описаны следующие команды: DIR, TYPE, ERA, REN, SAVE и USER. В описание стандартных утилит *CP/M* включены программы FORMAT, SYSGEN, MOVCPM, STAT, PIP, ED, SUBMIT, XSUB, ASM, DDT, DUMP и LOAD.

Кроме того, в приложении А описаны наиболее распространенные дополнительные программы ОС *CP/M*: DESPOOL, SID, ZSID, LINK-80**, LIB-80**, XREF, CREF** и XLT86.

Приложение Б содержит обзор совместимых с ОС *CP/M* компиляторов: MAC, RMAC, MACRO-80**, PL/I-80, BASIC-80**, BASCOM**, CBASIC, CB-80, PASCAL/MT+, FORTRAN-80**, COBOL-80**.

* Наименования *CP/M*, DESPOOL, SID, ZSID, XREF, XLT86, MAC, RMAC, PL/I-80, CBASIC, CB-80, PASCAL/MT+, MP/M-II, CP/M-86, MP/M-86 и CP/NET являются зарегистрированными торговыми марками фирмы Digital Research.

** Наименования LINK-80, LIB-80, CREF-80, MACRO-80, BASIC-80, BASCOM, FORTRAN-80 и COBOL-80 являются зарегистрированными торговыми марками фирмы Microsoft.

Приложение В посвящено некоторым другим операционным системам, построенным на базе ОС *CP/M*, а именно: *MP/M-II*, *CP/M-86*, *MP/M-86* и *CP/NET*.

Наконец в четырех дополнительных приложениях представлены краткие перечни команд, программ и промптов ОС *CP/M*, а также таблица символов в коде ASCII и список литературы.

В целом книга составлена таким образом, чтобы в ней можно было легко и быстро найти необходимые сведения по ОС *CP/M*.

Различные варианты использования команд и программ, а также соответствующий синтаксис проиллюстрированы многочисленными примерами, которые представлены в формате экрана терминала и снабжены необходимыми пояснениями.

Эта книга была специально написана как руководство по ОС *CP/M*. Хотя она содержит все необходимые для повседневной работы сведения, мы рекомендуем начинающим пользователям предварительно ознакомиться с книгой [5].

Тем, кто хочет глубже изучить работу ОС *CP/M*, особенностям тем программистам, которые хотят модифицировать секцию драйверов ввода-вывода (BIOS), подключать новые печатающие устройства или использовать в своих программах системные вызовы ОС *CP/M*, мы рекомендуем книгу [6].

Авторы постарались сделать книгу максимально полезной в повседневной работе с ОС *CP/M*, построили ее таким образом, чтобы упростить и ускорить отыскание и использование нужного материала. Мы искренне надеемся, что эта книга поможет вам научиться шутя пользоваться таким мощным инструментальным средством, каким является ОС *CP/M*.

Митчелл Уэйт и Джон Ангермайер

ГЛАВА 1

ВВЕДЕНИЕ

НАЗНАЧЕНИЕ КНИГИ

Эта книга написана в качестве подробного и удобного руководства по операционной системе *CP/M*. Если вы не знакомы с ОС *CP/M* или вообще ни с какой операционной системой для микро-ЭВМ, то в данной книге вы сможете найти подробную информацию об использовании команд и средств типовой операционной системы для микро-ЭВМ. В частности, сможете научиться пользоваться командами ОС *CP/M*. Эта книга, однако, не является учебным пособием по ОС *CP/M* или операционным системам в целом. Ее основное назначение состоит в том, чтобы дать пользователям *CP/M* удобный справочный материал. В качестве учебного пособия по ОС *CP/M* можно использовать книгу [5].

Вся книга выдержана в едином стиле: описание каждой команды дано в сжатой и точной форме и там, где это целесообразно, сопровождается простыми примерами.

КАК ПОЛЬЗОВАТЬСЯ КНИГОЙ

Примеры использования разновидностей команд представлены в левом верхнем углу условного изображения экрана дисплея:



При этом текст командной строки представляется в формате, в котором он должен быть введен с консоли, будь то видеотерминал, телепринтер или телетайп. Любой текст, выдаваемый ОС *CP/M* в ответ на введенную пользователем команду, представляется так, как он будет выглядеть на консоли.

Повсюду при описании команд делаются соответствующие ссылки на другие главы, разделы или приложения этой книги.

Построение глав в книге отражает наиболее удобный для пользователей порядок, при котором команды ОС *CP/M* описываются в логической последовательности, т. е. не делалось попытки объединить встроенные или транзитные команды. Команды наиболее общего назначения представлены в первых главах. Ниже представлен краткий обзор содержания глав.

Глава 2. Использование СС *CP/M*. В этой главе рассматриваются сведения об операционной системе *CP/M* общего характера: как происходит загрузка системы, как происходит обращение к командам. В эту главу включены, кроме того, различные соглашения относительно форматов команд, которые

необходимо иметь в виду при ссылке на остальные команды книги. Несмотря на то что эта книга задумана как руководство, имеет смысл сначала пропустить эту главу.

Главы 3 . . . 20. В этих главах описываются все встроенные и транзитные команды ОС *CP/M*. Главы представлены в логическом порядке так, что их легко использовать и читать в удобной последовательности. В конце этой главы представлено краткое описание каждой команды.

Глава 21. Обзор ОС *CP/M* версии 3.0. В этой главе дан обзор версии 3.0 системы *CP/M-PLUS*, приведены ее основные характеристики, а кроме того, краткие описания каждой команды.

Помимо основных глав эта книга содержит несколько приложений со вспомогательной справочной информацией, которая не имеет прямого отношения к командам ОС *CP/M*, но которую часто при работе полезно иметь под рукой. В этой книге имеются следующие приложения.

Приложение А. Дополнительные транзитные команды. Это приложение содержит краткие сведения о командах транзитных программ, которые могут быть дополнительно включены в состав ОС *CP/M*. Большинство из этих программ являются утилитами, которые расширяют возможности ОС *CP/M* как инструментального средства разработки программ.

Приложение Б. Трансляторы ОС *CP/M*. Это приложение содержит краткие сведения о нескольких трансляторах с языков программирования высокого уровня, совместимых с ОС *CP/M*.

Приложение В. Операционные системы, построенные на базе ОС *CP/M*. В этом приложении кратко описываются характеристики операционных систем *MP/M-II*, *CP/M-86*, *MP/M-86* и *CP/NET*, а также указывается, как они связаны с ОС *CP/M*.

Приложение Г. Список литературы. В этом подробном приложении приводится список книг и журнальных статей, имеющих непосредственное отношение к ОС *CP/M* и ее различным применением.

Приложение Д. Таблица символов кода ASCII и их значений. В этом приложении описывается Американский стандартный код для обмена информацией (ASCII) и даны определения его кодов в шестнадцатеричном, десятеричном и восьмеричном форматах.

Приложение Е. Промпты ОС *CP/M*. В этом приложении приводится список различных промптов, которые ОС *CP/M* выдает на консоль. Промптами называются сопровождающие сообщения, которые ОС *CP/M* выдает в зависимости от выполняемой программы. В этот набор промптов входят те, которые генерируются непосредственно ОС *CP/M*, и те, которые генерируются различными программами.

Приложение Ж. Сообщения об ошибках. В этом приложении содержится список выдаваемых ОС *CP/M* сообщений об ошибках. Для каждой ошибки указаны возможные причины возникновения и возможные меры устранения.

Приложение З. Краткое описание команд. В этом приложении дано краткое описание всех команд и их разновидностей, описанных в этой книге.

ОПЕРАЦИОННЫЕ СИСТЕМЫ ДЛЯ МИКРО-ЭВМ

С первых дней развития вычислительной техники по настоящее время всегда существовала необходимость в управляющих программах, которые бы

удовлетворяли многочисленным нуждам по управлению вычислительными системами. Для этого разработанные управляющие программы были наделены функциями "надсмотрщиков" за оборудованием ЭВМ. Эти управляющие программы обеспечили пользователей руководствами и соглашениями по пользованию различными средствами вычислительной системы. Они определили последовательность выполнения тех действий в вычислительной системе, в которых наиболее часто возникала необходимость, как, например, точное определение состава информации, хранящейся на дисках (т. е. какой диск содержит конкретную программу), создание и удаление файлов, запуск программ, измерение размера доступного пространства на дисковых носителях и т. п. На протяжении уже ряда лет по отношению к этим программам употребляется термин "операционные системы". Вообще можно рассматривать операционную систему как программу, которая необходима для выполнения всех других программ. Операционная система (ОС или ДОС, если для ее хранения используется диск) всегда находится в памяти ЭВМ, пока та находится в рабочем состоянии. Другие программы перед исполнением загружаются в память и удаляются из нее после завершения.

Почти всегда операционная система является первой и наиболее важной программой в любой ЭВМ, будь то большая универсальная система, мини- или микро-ЭВМ. Начиная с середины 70-х годов операционные системы стали играть самую существенную роль в работе с большинством микро-ЭВМ. Хотя некоторые ЭВМ не используют операционную систему как таковую, отсутствие подобной управляющей программы зачастую существенно ограничивает использование этих ЭВМ во многих важных приложениях. Что касается ОС *CP/M*, то она была разработана для специфического носителя, а именно для гибкого диска. Поэтому структура операционной системы отчасти отражает способ организации гибкого диска (например, максимальный объем памяти и т. д.). Операционная система *CP/M* не будет работать с кассетной магнитной лентой, но работает с жестким диском.

Большинство операционных систем выполняет следующие основные функции.

1. Представляют упорядоченный и согласующийся набор подпрограмм ввода-вывода для различных периферийных устройств ЭВМ (терминалы, устройства печати, накопители на обычных или гибких дисках, накопители на магнитных лентах и т. д.). Термин "ввод-вывод" по существу обозначает способ, посредством которого ЭВМ осуществляет обмен данными с различными подключенными к ней периферийными устройствами.

2. Обеспечивают "управление файлами" и получение отчетов о состоянии хранимых данных. Система управления файлами дает пользователю возможность получить информацию о файлах, хранящихся на дисковых носителях: имена файлов, оглавления дисков, размеры файлов, объем неиспользованной памяти и т. д. Именно эти сведения необходимы для манипулирования дисковыми файлами. Эта система управления файлами взаимодействует также с подпрограммой ввода-вывода для дисковых устройств, которая используется для записи (чтения) данных на жесткий или гибкий диск (с диска) и т. д.

3. Обеспечивают загрузку и выполнение стандартных программ системы и программ, указанных пользователем.

Многие операционные системы обладают утонченными возможностями.

Сюда входит обеспечение одновременного выполнения нескольких программ (задач), подсчет суммарного времени работы пользователя с системой, отчет о степени использования программой центрального процессора, представление системы парольной защиты для обеспечения секретности и т. д. Однако, несмотря на кажущееся разнообразие, все операционные системы (в том числе ОС *CP/M*, которая первоначально была разработана для использования на микро-ЭВМ) в той или иной форме выполняют только что отмеченные три основные функции.

ОПЕРАЦИОННАЯ СИСТЕМА *CP/M*

В 1973 г. микропроцессор Intel 8080 стал первым достаточно мощным устройством, которое можно было применить для создания микро-ЭВМ. Несколько компаний начали производство малых вычислительных систем, базирующихся на этой интегральной микросхеме. В начале эти ЭВМ были достаточно примитивны, поскольку не было совместимых с ними операционных систем. Однако вскоре было разработано несколько операционных систем, ориентированных на работу с микро-ЭВМ, выполненных на микропроцессоре Intel 8080, а также с ЭВМ, построенными на базе других микропроцессоров, созданных позже. В число этих систем входит операционная система *CP/M*, созданная в 1975 г. Г. Килдэлом, который в 1976 г. организовал фирму Digital Research. Операционная система *CP/M* первоначально была разработана для микро-ЭВМ, применяющих Intel 8080, хотя созданные позже совместимые с ним микропроцессоры Intel 8085 и Zilog Z80 также поддерживают ОС *CP/M*. С того времени ОС *CP/M* стала одной из наиболее популярных и широко используемых операционных систем и считается многими "как промышленный стандарт", поскольку она была применена многими производителями микро-ЭВМ и обеспечивает работу буквально сотен различных прикладных программ, созданных разработчиками программного обеспечения микро-ЭВМ.

Беспрецедентная популярность ОС *CP/M* объясняется простотой принципов ее работы, наличием простого и компактного набора команд, а также исключительной гибкостью в адаптации к различному периферийному оборудованию. Операционная система *CP/M* снабжена несколькими стандартными командами и программами-утилитами. Некоторые из них – это команды просмотра каталогов файлов и команды манипулирования файлами, команды получения характеристик файлов, устройств и данных об объеме свободного пространства, программа-редактор текста, программа для копирования файлов и т. д. Большинство из них служат тем же целям, что и аналогичные команды в других операционных системах. Дополнительные программы, такие как программа динамической отладки и программа-оборщик, которые в основном предназначены для программистов, делают ОС *CP/M* полезным и мощным средством разработки программ.

Первоначально операционная система *CP/M* была разработана для микро-ЭВМ, оснащенных накопителями на гибких дисках, и является, следовательно, дисковой операционной системой (ДОС). Стандартным форматом гибкого диска, принятым для использования в ОС *CP/M*, является стандарт IBM 3740. Этот стандарт был установлен фирмой International Business Machines (IBM) и описывает формат для односторонних гибких дисков диаметром в

восемь дюймов с одинарной плотностью записи, программным секторированием и емкостью 128 байт на сектор. В промышленности этот стандарт обычно называют "стандарт для 8-дюймовых дисков с одинарной плотностью". Поскольку этот стандарт был принят многими производителями микро-, мини- и универсальных ЭВМ в качестве основного стандарта для 8-дюймовых гибких дисков, то большинство производителей ЭВМ сегодня обеспечивают совместимость своих моделей ЭВМ с этим стандартом. Использование данного стандарта позволяет осуществлять обмен гибкими дисками между всеми микро-ЭВМ, работающими под управлением ОС *CP/M*. Поскольку очень многие пользуются гибкими дисками, имеет прямой смысл разрабатывать программное обеспечение для работы под управлением этой дисковой операционной системы. Это существенно облегчает распространение программного обеспечения, хранящегося на гибких дисках при производстве микро-ЭВМ в сети микро-ЭВМ, созданной на базе ОС *CP/M*.

С середины 70-х годов были разработаны другие типы накопителей на гибких дисках, использующие различные форматы. Например, были созданы накопители на 5½-дюймовых (мини) одно- и двусторонних гибких дисках с программным секторированием и одинарной или двойной плотностью записи. Совершенствование 8-дюймовых гибких дисков с программным секторированием также продвинулось в сторону осуществления двойной плотности записи для одно-, так же как и для двусторонних дисков. Некоторые изготавливатели приняли формат с аппаратным секторированием и в 5¼, и 8-дюймовых дисках. Хотя стандарт IBM 3740 все еще остается основным стандартом для поддержания совместимости с созданными на основе *CP/M* операционными системами, саму же ОС *CP/M* можно модифицировать для работы с накопителями на гибких дисках других типов. Кроме того, ОС *CP/M* можно видоизменить при использовании ее совместно с оборудованием накопителей на жестких дисках различных типов, включая накопители типа "Winchester". Системы, имеющие по меньшей мере одно устройство с 8-дюймовым гибким диском с одинарной плотностью записи, имеют доступ к наибольшему числу программ ОС *CP/M*. Системы, использующие специальные форматы, вынуждают пользователя либо приобретать программное обеспечение у разработчика ЭВМ, либо ждать, пока разработчик программного обеспечения не начнет поставлять соответствующую версию для его модели ЭВМ. Различные вычислительные системы, работающие под управлением ОС *CP/M*, могут быть связаны друг с другом через последовательные порты ввода-вывода, что позволит осуществлять передачу наборов данных (подробности см. в книге [5]).

Приспособляемость ОС *CP/M* к различным типам оборудования и форматам накопителей на гибких и жестких дисках безусловно делает ее универсальной системой. К тому же ОС *CP/M* поддерживает различные периферийные устройства (телефайпы, видеотерминалы телепринтеры, цветные дисплеи, устройства отображения памяти, устройства ввода-вывода перфоленты, модемы). Помимо дисковых устройств, стандартная конфигурация ОС *CP/M* поддерживает внешние устройства следующих четырех основных типов:

Консоль. Консоль представляет собой стандартное устройство для взаимодействия пользователя с ОС *CP/M* и микро-ЭВМ. Консоль может быть как телефоном (устройство типа TTY), так и видеотерминалом (устройство типа CRT).

Устройства, работающие в режиме только ввод. Эти устройства первоначально предназначались для высокоскоростного считывания с перфоленты. Однако ОС *CP/M* можно модифицировать так, что будут доступны другие, определенные пользователем, устройства (такие как модем или прочие устройства сопряжения с другими ЭВМ).

Устройства, работающие в режиме только вывод. Устройства данного типа также первоначально предназначались для высокоскоростного вывода информации на перфоленту. С другой стороны, ОС *CP/M* можно модифицировать для подключения других, определенных пользователем, устройств вывода информации.

Печатающее устройство. Термин "печатывающее устройство" – это обобщенное название для всех устройств получения "твердой копии". Оно может относиться к телетайпу или печатающему устройству (матричному или барабанному).

ОПЕРАЦИОННЫЕ СИСТЕМЫ, ПОСТРОЕННЫЕ НА БАЗЕ ОС *CP/M*

Рост в течение последних нескольких лет популярности микро-ЭВМ и операционных систем ускорил появление нескольких значительных достижений в технологии проектирования микро-ЭВМ и их применения. Эти достижения привели к тому, что одновременно микро-ЭВМ стало применять большое количество людей, а это, в свою очередь, привело к необходимости в дальнейшем развитии возможностей операционных систем. Поскольку ОС *CP/M* является операционной системой исключительно "для одного пользователя" и может одновременно управлять выполнением только одной программы или задачи, она стала бесполезной во многих приложениях, где требуется дешевая (на базе микропроцессоров), для многих пользователей (одновременно работает более чем один пользователь) и пригодная для выполнения нескольких задач (пользователь может одновременно запускать более одной программы), система.

Начиная с 1977 г. фирма Digital Research откликнулась на эти нужды операционной системой *MP/M*, которая создана на базе ОС *CP/M* и обеспечивает режимы работы для многих пользователей и с несколькими задачами. Кроме того, в последние годы был достигнут прогресс в понимании распределенной обработки данных и сетей микро-ЭВМ. С этой целью фирмой Digital Research была разработана операционная система *CP/NET* – управляющая программа для объединенных в сеть микро-ЭВМ, использующих ОС *CP/M* или *MP/M*.

Хотя микропроцессор Intel 8080 и его "ближайшие родственники" Intel 8085 и Zilog Z80 в настоящее время все еще популярны, в технологии производства микропроцессоров на базе интегральных микросхем для повышения их мощности сделаны многие усовершенствования – с 1977 г. было разработано несколько 16-битовых и 32-битовых микропроцессоров, включая Intel 8088, Intel 8086, Motorola 68000, Zilog Z8000 и National Semiconductor 16000 (фирма Intel в настоящее время разрабатывает 32-битовый микропроцессор iAPX-432).

Одним из первых удачных 16-битовых микропроцессоров был микропроцессор Intel 8086. Поскольку этот микропроцессор некоторым образом подобен микропроцессору Intel 8080, фирма Digital Research разработала для него операционные системы *CP/M-86* и *MP/M-86*, которые являются 16-бито-

выми аналогами систем *CP/M* и *MP/M*. К тому же недавно разработана мультизадачная версия ОС *CP/M* – Concurrent *CP/M-86*, обеспечивающая мультизадачный режим работы микро-ЭВМ в реальном масштабе времени. Указанные системы были специально созданы для работы с микропроцессорами Intel 8086 и Intel 8088.

Новое поколение 16-битовых и 32-битовых микропроцессоров включает в себя микропроцессоры со столь различной архитектурой, что для всех них фактически невозможно создать универсальную операционную систему. Недавно фирма Digital Research заявила о создании операционной системы *CP/M-68K* для микропроцессора Motorola 68000 и планирует создание специальной версии ОС *CP/M* для микропроцессора National Semiconductor 16000 (подобная версия ОС *CP/M* может быть настолько мощной, что составит конкуренцию большим универсальным ЭВМ). Несмотря на эти заявления перечень микропроцессоров, микро-ЭВМ и операционных систем еще пополнится.

Эта книга была написана специально для пользователей микро-ЭВМ, сознанных на базе микропроцессоров Intel 8080, Intel 8085 и Zilog Z80. Краткая сравнительная характеристика операционных систем *MP/M*, *CP/M-86*, *MP/M-86* и *CP/NET* приведена в конце книги в приложении В.

КРАТКОЕ ОПИСАНИЕ КОМАНД ОС *CP/M*

В операционной системе *CP/M* существует два типа команд: встроенные команды и транзитные команды. Встроенные команды являются частью операционной системы ОС *CP/M* и постоянно хранятся вместе с ней в одной и той же области памяти. Транзитные программы команд – это те, которые всегда хранятся на диске в виде командных (COM) файлов, т. е. для использования какой-либо транзитной команды необходимо наличие соответствующего файла с транзитной командой.

Каждая из встроенных и стандартных транзитных команд ОС *CP/M* описана в соответствующей главе (главы с 3 по 20).

Встроенные команды. Операционная система *CP/M* содержит секцию, которая называется "процессор команд консоли" (CCP). Секция CCP является подпрограммой ОС *CP/M*, которая постоянно проверяет клавиатуру консоли на соответствие введенных символов командам ОС *CP/M*. Секция CCP включает в себя набор встроенных или "резидентных" команд, для обращения к которым не требуется использование диска. Вызов этих встроенных команд осуществляется с помощью ввода в качестве ответа на промпт ОС *CP/M* (*A >*, *B >* и т. д.) имени команды. Поскольку эти команды не содержатся в отдельных файлах, они доступны для пользования независимо от того, какое дисковое устройство является текущим, то есть, в отличие от транзитных программ, для выполнения встроенных команд обращения к диску не требуется. Каждая из встроенных команд описана в отдельной главе (см. гл. 3...20). Ниже представлено краткое описание каждой встроенной команды ОС *CP/M*.

Команда DIR (гл. 3). Команда DIR является, по-видимому, наиболее часто используемой командой ОС *CP/M*. Название DIR является сокращением английского слова directory (каталог). Команда DIR используется для отображения или печати справочных данных всех файлов, содержащихся на указанном диске (см. стр. 45).

Команда REN (гл. 4). Название REN является сокращением английского сло-

ва rename (переименовать). Команда REN используется для изменения имени специфицированных файлов, хранящихся на обозначенном диске (см. стр. 48).

Команда ERA (гл. 5). Название ERA является сокращением английского слова erase (стереть). Команда ERA используется для удаления специфицированных файлов, хранящихся на обозначенном диске (см. стр. 48).

Команда TYPE (гл. 6). Команда TYPE используется для просмотра на консоли содержимого файлов (см. стр. 48).

Команда USER (гл. 7). Команда USER предоставляет различным пользователям ОС CP/M средства обращения к своим файлам, хранящимся на одном и том же диске. В ОС CP/M с одним и тем же диском могут работать до 15 пользователей, причем рабочие области диска каждого пользователя (совокупность его личных файлов) не пересекаются. Команда USER имеется в распоряжении только в версии 2.0 и более поздних версиях ОС CP/M (см. стр. 48).

Команда SAVE (гл. 16). Команда SAVE используется для того, чтобы взять содержимое указанного числа страниц из системной области оперативной памяти и сохранить его в специфицированном файле на обозначенном диске. Эта команда обычно используется только системными программистами (см. стр. 48).

Транзитные команды. Транзитные команды (утилиты) отличаются от встроенных команд тем, что они временно, лишь во время их выполнения, хранятся в оперативной памяти. Поэтому необходимо, чтобы транзитные программы сохранялись в файлах, расположенных на дисковых носителях. Стандартные транзитные программы ОС CP/M написаны на ассемблере микропроцессора Intel 8080, поэтому они хранятся в машинном коде Intel 8080. Следовательно, все команды или файлы транзитных программ имеют расширение ".COM", что обозначает файл "командного" типа. Расширение ".COM" указывает на то, что это — файл для непосредственного исполнения, т. е. достаточно ввести имя файла и сразу начнется выполнение программы. Другими словами, при использовании транзитной команды необходимо ввести только имя соответствующего файла без расширения, поскольку ОС CP/M автоматически будет искать файл с указанным именем и расширением ".COM". Для того чтобы пользоваться транзитной командой, ее следует хранить на диске, установленном в одном из устройств системы. Например, для того чтобы выполнить команду STAT файл "STAT.COM" следует хранить на диске, вставленном в одно из устройств системы.

В книге охвачены только те команды, которые всегда находятся на системном диске ОС CP/M. Отметим, что на системном диске могут находиться и другие, нестандартные программы-утилиты (фирмой Digital Research не поставляются). Большинство из этих программ в первую очередь добавлены той компанией (не фирмой Digital Research), у которой приобретена ОС CP/M. Обычно типы и способы использования этих дополнительных нестандартных программ сильно варьируются в зависимости от того, у кого первоначально получено ОС CP/M — у разработчика или у распространителя. Команды некоторых наиболее распространенных нестандартных программ описаны в приложении А.

В каждой главе, где даются команды транзитной программы, подробно рассмотрены все возможные способы применения программы-утилиты. По-

рядок представления команд в главах отражает, вероятно, наиболее распространенную последовательность, в которой начинающие пользователи ОС *CP/M* стали бы их применять. Например, команды STAT и PIP описаны первыми среди транзитных, поскольку эти команды дают возможность манипулирования файлами. Команды FORMAT и SYSGEN представлены среди последних, так как они используются не так часто, как остальные команды, хотя могут быть применены даже первыми.

Ниже представлено краткое описание стандартных транзитных команд. Эти команды имеются во всех версиях (1.3, 1.4, 2.0 и 2.2) ОС *CP/M*, если не оговорено особо.

Команда STAT (гл. 8). Название STAT означает Status или Statistics – состояние или характеристики. Команда STAT предоставляет пользователю ОС *CP/M* различные средства для назначения статуса и характерных свойств файлам, дисковому носителю и другим устройствам и вырабатывает основную методику ведения "файлового хозяйства" (см. стр. 60).

Команда PIP (гл. 9). Название PIP является сокращением от английского Peripheral Interchange Program (программа обмена данными между внешними устройствами). Команда PIP используется для копирования файлов с какого-либо устройства на устройство того же или другого типа, например с диска на диск, с диска на печатающее устройство, с консоли на диск, с диска на модем и т. д. (см. стр. 82).

Команда ED (гл. 10). Название ED является сокращением английского слова EDitor (редактор). Команда ED позволяет пользователю ОС *CP/M* запустить текстовый редактор, который может создавать и редактировать текстовые файлы (см. стр. 112).

Команда SUBMIT (гл. 11). Команда SUBMIT представляет возможность автоматического запуска указанной последовательности команд, предварительно подготовленных пользователем в отдельном файле (см. стр. 147).

Команда XSUB (гл. 12) (имеется только в версии 2.0 и более поздних). Название XSUB является сокращением от английского EXtended SUBMIT (расширенная программа SUBMIT). Программа XSUB входит в состав программ команды SUBMIT. Она позволяет вводить ответы на промпты программы, уже запущенной на исполнение программой SUBMIT.

Команда ASM (гл. 13). Название ASM является сокращением английского слова Assembler. Программа ASM "ассемблирует" файлы с программой на ассемблере, созданные по команде ED или по команде аналогичного текстового редактора, в так называемые ".HEX" файлы (шестнадцатеричные), которые, в свою очередь, можно по команде LOAD преобразовать в файлы с чисто машинным или объектным кодом. Программа ASM используется программистами на ассемблере при создании собственных транзитных программ для микропроцессора Intel 8080 (см. стр. 157).

Команда LOAD (гл. 14). Команда LOAD используется для преобразования шестнадцатеричного ("HEX") файла в файл, содержащий машинный или объектный код программы (см. стр. 171).

Команда DDT (гл. 15). Название DDT означает Dynamic Debugging Tool (инструментальное средство для динамической отладки программ). Это – утилита, которая предоставляет пользователю ОС *CP/M* метод определения и устранения ошибок в программах, представленных в машинном объект-

ном коде или шестнадцатеричном формате, в целях проверки секции базовой системы ввода-вывода (BIOS) и базовой дисковой операционной системы (BDOS) и т. д. (см. стр. 173).

Команда DUMP (гл. 17). Команда DUMP используется для вывода в шестнадцатеричном формате содержимого файла с объектным/машическим кодом на экран дисплея или печатающее устройство (см. стр. 196).

Команда FORMAT (гл. 18). Команда FORMAT служит для подготовки диска к приему информации. Эта форматирующая программа подготавливает гибкий диск к записи стандартных совместимых с ОС *CP/M* файлов. Кроме того, форматирующую команду можно включить в систему, содержащую жесткий диск типа "Winchester". Программа FORMAT является не стандартной и может иметь несущественные различия в работе в зависимости от поставщика версии операционной системы *CP/M*. На различных ЭВМ назначение и принципы действия программы FORMAT остаются примерно одинаковыми, в то время как детали физической организации дорожек и секторов меняются от системы к системе. Таким образом, большинство форматов являются зависимыми от технического обеспечения ЭВМ (см. стр. 198).

Команда SYSGEN (гл. 19). Название SYSGEN является сокращением от английского System Generation (генерация системы). Команда SYSGEN предоставляет пользователю средства для генерирования операционной системы *CP/M* или копирования ее с одного диска на другой (см. стр. 206).

Команда MOVCMP (гл. 20). Название MOVCMP является сокращением от английского "Move CP/M" (переместить *CP/M*). Команду MOVCMP применяют для модификации операционной системы *CP/M* с тем, чтобы она либо использовала всю доступную память ЭВМ, либо оставляла часть памяти под специальные системно-независимые программы пользователей (см. стр. 213).

Команды управляющих символов. ОС *CP/M* предоставляет пользователю несколько специальных команд, так называемые "команды управляющих символов". Эти команды дают пользователю ОС *CP/M* возможность управлять и изменять способ ввода командных строк и возможность реагировать на них. Эти команды называются "команды управляющих символов", поскольку для их выполнения необходимо одновременно нажать на терминале клавишу "CONTROL" и клавишу, соответствующую указанному символу. Полностью эти команды описаны в гл. 2.

В следующей главе представлен обзор операционной системы *CP/M*. В ней содержатся сведения, которые необходимы для понимания основных характеристик ОС *CP/M* и которые помогут в дальнейшем пользоваться командами, описанными в последующих главах.

ГЛАВА 2

ИСПОЛЬЗОВАНИЕ ОС *CP/M*

ОБЩИЕ СВЕДЕНИЯ

Каждому, кто впервые использует ОС *CP/M*, следует ознакомиться с запуском системы и с вводом данных с терминала. Кроме того, следует освежиться о том, как вставлять и вынимать диски из дисковых устройств. В этой главе принципы работы ОС *CP/M* излагаются с точки зрения того пользователе-

ля, который сначала включает систему, затем вставляет системный диск ОС *CP/M* в дисковое устройство, загружает ОС *CP/M* и вводит команды с консоли.

При более сложном использовании ОС *CP/M* вносятся изменения в саму программу *CP/M*. Это необходимо, когда микро-ЭВМ куплены и устанавливаются. При установке на них ОС *CP/M* обычно модифицируется для работы с конкретными, подключенными к данной ЭВМ, внешними устройствами. Обычно это делается до ввода системы в эксплуатацию поставщиком ОС *CP/M* или микро-ЭВМ, либо разработчиком микро-ЭВМ. Однако, в тех случаях, когда такая настройка ОС *CP/M* не произведена, следует самостоятельно внести изменения в секцию драйверов ввода-вывода ОС *CP/M*, называемую BIOS. Описание метода внесения подобных изменений выходит за рамки этой книги, поскольку для этого требуется подробное техническое описание структуры таких секций ОС *CP/M*, как BIOS, BDOS и пр. Однако подобное описание можно найти в книге [6].

При установке ОС *CP/M* микро-ЭВМ пользователю необходимо изучить, как инициализировать систему и как проверить, что ее работа осуществляется должным образом. В этой главе предполагается, что в распоряжении пользователя имеется системный диск с версией ОС *CP/M*, уже настроенный на работу с конкретным оборудованием, подключенным к микро-ЭВМ. Если это так, то последующее описание поможет адаптировать операционную систему к вашей ЭВМ с тем, чтобы можно было немедленно начать работу.

Как отмечалось ранее, если системный диск с ОС *CP/M* не настроен на работу с внешними устройствами вычислительной системы, то следует проконсультироваться с разработчиками или поставщиками ОС *CP/M* о возможности ее модификации. При намерении провести модификацию ОС *CP/M* самостоятельно некоторые, но не все, сведения о том, как можно сделать, даны в руководстве пользователя, поставляемому с системой ОС *CP/M*. Кроме того, существуют другие книги и журнальные статьи по ОС *CP/M*, в которых приводится информация о некоторых способах вынесения изменений в систему (см. приложение Б, в котором представлен подробный список литературы).

ЗАГРУЗКА ОС *CP/M*

Для того чтобы привести ОС *CP/M* в рабочее состояние, сначала следует "загрузить" ее с системного диска, т. е. разместить копию ОС *CP/M* в оперативной памяти ЭВМ. Оперативная память ЭВМ используется для временного хранения программ и данных, так что в процессе работы ее содержимое может многократно меняться. Пока микро-ЭВМ подключена к сети питания, информация, содержащаяся в ее оперативной памяти, остается целой. При отключении питания или нажатии клавиши "сброс системы" информация в оперативной памяти будет сразу уничтожена. По этой причине каждый раз при отключении и повторном включении питания или сбросе системы копия операционной системы *CP/M* в оперативной памяти будет утеряна. Поэтому прежде чем начинать работу, необходимо снова разместить копию ОС *CP/M* в оперативной памяти. Операция возвращения ОС *CP/M* к жизни называется "загрузкой" ОС *CP/M*. В процессе загрузки программа *CP/M*, хранящаяся на дисковом носителе, копируется в системную область оперативной памяти микро-ЭВМ. Загрузка операционной системы является необходимой процедурой

рой, поскольку если в оперативной памяти микроЭВМ нет программы операционной системы, то центральному процессору ЭВМ не с чем будет работать.

Программа начальной загрузки. На всех микро-ЭВМ, совместимых с операционной системой *CP/M*, имеется программа начальной загрузки. Это — программа, которая хранится в программируемом постоянном запоминающем устройстве (ППЗУ) микро-ЭВМ, размещенном в составе оборудования центрального процессора и содержимое которого не уничтожается при отключении питания. Эта специальная программа обеспечивает выполнение действий, необходимых для загрузки ОС *CP/M* (или другой операционной системы) с системного диска и размещения ее копии в оперативной памяти микро-ЭВМ. Поскольку программа *CP/M*, которую иногда называют также "образ системы", полностью записана на первых двух (нулевой и первой) дорожках системного диска, программа начальной загрузки содержит команды центрального процессора для считывания данных только с этих двух дорожек диска. При включении микро-ЭВМ в сеть питания или сбросе системы можно услышать звуковой сигнал, означающий, что скопированная в память системы ОС *CP/M* загружена по программе начальной загрузки.

Программа "Монитор". Метод, при помощи которого осуществляется загрузка ОС *CP/M*, может быть различным в зависимости от типа микро-ЭВМ. В большинстве систем загрузка ОС *CP/M* происходит просто при сбросе или включении в сеть. В отдельных системах имеется программа "Монитор" или "Мониторная управляющая программа", которая также хранится в ППЗУ и предназначена для обработки специальных команд, вводимых с консоли. Одна из этих команд может быть определена как команда загрузки операционной системы с диска. Другие команды могут использоваться для обращения к ячейкам оперативной памяти, изменения условий по умолчанию в ЭВМ и т. д. В одних системах программа "Монитор" выдает промпт или какое-либо сообщение сразу после включения, в других — только после нажатия определенной клавиши (как правило, клавиши "пробел"). Пример типового общения программы "Монитор" приведен ниже:

```
CPU MONITOR PROGRAM  
4800 BAUD SERIAL  
?
```

Начальная загрузка ОС *CP/M* ("Холодный старт"). В предыдущем примере символ "?" является промптом программы "Монитор". Этот символ означает, что программа "Монитор" готова к обработке очередной команды. В зависимости от типа программы "Монитор" специальный символ, используемый в качестве команды загрузки, в разных системах может быть различным. В приведенном примере командной загрузки операционной системы является символ "B" (по начальной букве английского слова Boot — загрузка). Поскольку символ "B" определен как команда загрузки не во всех программах "Монитор", следует обратиться к Руководству по эксплуатации системы, если в ней принят данный способ загрузки. Для загрузки ОС *CP/M* сначала необходимо вставить системный диск в дисковое устройство, которое определено как "Устройство загрузки" (обычно это устройство с именем "0" или "A"). Как только системный диск должным образом установлен, вводится команда загрузки. В некоторых системах вводить команду не обязательно, поскольку для осуществления загрузки ОС *CP/M* достаточно на-

жать клавишу "сброс системы". Начальная загрузка ОС *CP/M* называется "холодным стартом". Это название относится к процессу загрузки операционной системы в первый раз после передачи на ЭВМ питания (и таким образом являющаяся еще "холодной"). Если загрузка ОС *CP/M* прошла успешно, то появится сообщение, аналогичное представленному ниже:

XXK CP/M VERS Y.Y

A>

Сообщение, выдаваемое при загрузке ОС *CP/M*. Сообщение, представленное выше, является типичным для большинства версий ОС *CP/M*. Здесь "XXk" представляет собой числовое значение, равное объему оперативной памяти (в килобайтах), на использование которого настроена данная версия ОС *CP/M*. Например, в версии 1.4 ОС *CP/M* значение "XX" может меняться от 16К до 64К. В версии 2.0 и выше это значение может меняться от 20К до 64К или более. Точное числовое значение зависит от того, какой объем оперативной памяти имеется в микро-ЭВМ и какой объем специфицирован разработчиком или поставщиком при инициализации ОС *CP/M* как необходимый для нормальной работы (см. стр. 16).

Вторая часть сообщения "vers Y.Y" указывает версию ОС *CP/M*: 1.3, 1.4, 2.0, 2.2 и т. д. В зависимости от версии ОС *CP/M* и от ее поставщика в сообщении может содержаться дополнительная информация, а именно имя разработчика, дата проведения генерации, список подключенных внешних устройств и т. п. Эта дополнительная информация может относиться к изменениям в секции BIOS и всем изменениям, сделанным для обеспечения корректной работы с имеющейся в данной системе специфической аппаратурой, такой как контроллеры дисков.

Последней частью сообщения в данном примере является промпт ОС *CP/M*. Этот промпт всегда расположен там же, где и курсор экрана или головка печатающего устройства, и указывает на начальную позицию, с которой можно вводить команды ОС *CP/M*. Сначала промпт имеет вид "A >", и это означает, что в данный момент осуществляется работа с диском "A" (вставленным в устройство загрузки), а ОС *CP/M* ожидает ввода команды. Для предоставления другого диска этот промпт можно изменить. Это будет обсуждено позже.

Важно помнить, что первые две дорожки диска "A" всегда должны содержать "образ" операционной системы *CP/M* (обычно все диски, используемые при работе с данной системой, содержат этот "образ"). Если после запуска и начала работы в устройство загрузки будет вставлен диск, не содержащий на первых двух дорожках ОС *CP/M*, то при первой же попытке выполнить инициализацию системы (поскольку ЭВМ уже не является "холодной") произойдет сбой. Поэтому на каждом новом диске, предназначенном для работы с ОС *CP/M*, рекомендуется создавать с помощью команды SYSGEN (см. гл. 19) резервную копию ОС *CP/M*.

Создание резервных копий дисков. В начале работы с ОС *CP/M* сразу после загрузки системы необходимо создать резервную копию системного диска. Данные действия являются стандартными и в целях предосторожности их необходимо выполнять как можно скорее. Это существенно, поскольку системный диск ОС *CP/M* является местом постоянного хранения самой операционной системы, а также ее транзитных и прикладных программ. При вы-

полнении указанных рекомендаций на случай выхода из строя системного диска, всегда остается по меньшей мере одна (лучше если несколько) работоспособная копия ОС *CP/M* и ее программ (фактически лучше всего создавать резервные копии всех дисков, а не только тех, с которых выполняется загрузка).

Создание резервной копии можно выполнить при нормальной работе системы сразу после загрузки ОС *CP/M*. Для создания резервной копии системного или любого другого диска ОС *CP/M* всякий раз следует выполнить следующие операции:

1. Разметить новый диск при помощи имеющейся в наборе программ ОС *CP/M* программы форматирования диска. Обычно эта программа называется "FORMAT", хотя ее имя может быть и другим. Описание использования команды FORMAT см. в гл. 18 (стр. 198).
2. После разметки диска на него можно копировать необходимые данные. Если требуется осуществить загрузку с нового диска, то необходимо использовать команду SYSGEN, которая выполняет копирование системной программы *CP/M* в первые две дорожки нового диска. Описание использования команды SYSGEN см. в гл. 19. Если с нового диска не предполагается проводить загрузку, то эту операцию можно пропустить.
3. После того как диск размечен и к тому же предназначен для загрузки, на него можно копировать все данные и программы, которые требуется сохранить. Обычно это осуществляется с помощью команды PIP. Команда PIP позволяет копировать отдельные файлы, группы файлов или все файлы с исходного диска на резервный. Описание команды PIP см. в гл. 9. Некоторые пакеты программ ОС *CP/M* содержат специальную программу копирования, которая иногда называется "COPY". Эта программа осуществляет копирование не только всех файлов, но и "образа" системы, что исключает необходимость в использовании команды SYSGEN (операция 2).

Полезно создавать резервную копию диска всякий раз после изменения его содержимого. В этом случае при выходе из строя диска с чрезвычайно важной информацией остается его копия. Однако нельзя целиком полагаться на это средство, поскольку иногда происходят непредсказуемые явления, приводящие к потере информации. Диски с резервными копиями позволяют экономить время и сохранять ценное программное обеспечение.

Управление памятью. После завершения загрузки ОС *CP/M* лишь небольшая часть оперативной памяти занята операционной системой. Оставшаяся часть памяти может быть использована для прикладных программ. Управление использованием свободной памяти осуществляется ОС *CP/M*, так что вся свободная память или отдельные ее участки могут использоваться повторно всякий раз, когда требуется выполнение очередной прикладной программы.

Реинициализация системы ("теплый старт"). После рассмотрения первичной загрузки ("холодного старта") ОС *CP/M* перейдем к описанию другого типа загрузки, который называется "реинициализация" или "теплый старт". "Теплый старт" используется при выходе из исполнявшейся в системе программы. Он вынуждает ОС *CP/M* реинициализировать саму себя путем считывания с системного диска только части операционной системы, оставляя все остальное не-

изменным. Эта процедура выполняется ОС *CP/M* автоматически и вводить команду реинициализации не требуется. Существуют, однако, другие обстоятельства (см. далее), при которых обязательно следует вводить команду реинициализации. Когда это необходимо, можно ввести команду реинициализации, одновременно нажав на клавиатуре консоли клавиши "CTRL" (Control) и "C" (CTRL/C).

Команду CTRL/C необходимо вводить в тех случаях, когда ОС *CP/M* не может должным образом инициализировать саму себя из-за смены диска на устройстве загрузки или при возникновении определенного типа ошибок, или при сбоях. Поскольку подобные ошибки сопровождаются выдачей на консоль соответствующих сообщений, то пользователь будет знать о том, что необходимо выполнить реинициализацию системы. Другие ошибки, требующие выполнить реинициализацию системы, не сопровождаются выдачей сообщений, и тогда единственной причиной возникновения ошибки является потеря работоспособности системы.

Если реинициализация ОС *CP/M* не удалась, то необходимо выполнить полную перезагрузку ОС *CP/M*. При этом следует попытаться определить причину возникновения ошибки или сбоя. Ошибка может быть вызвана нарушением работоспособности оборудования или дефектом, возникшим при чтении с диска ОС *CP/M* или какой-либо другой программы. Если причина появления ошибки заключается в неисправности дискового носителя (например, повреждена рабочая поверхность, или произошел "мягкий" сбой), то вместо него следует использовать соответствующую резервную копию. Подобные ошибки могут также возникать при попытке выполнить программу, которая либо содержит ошибки, либо является несовместимой с ОС *CP/M*.



ИСПОЛЬЗОВАНИЕ ДИСКОВ

После успешной загрузки ОС *CP/M* диск с именем "A" автоматически становится "текущим" диском. Текущим или "диском по умолчанию" называется диск, обращение к которому подразумевается во всех последующих командах, если в них имя диска не указывается явно*.

Какой диск является в настоящий момент текущим, можно определить по промпту, выдаваемому ОС *CP/M* на консоль. Например, промпт "A>" означает, что текущим является диск с именем "A". В ОС *CP/M* возможно, кроме того, обращение и к другим дискам, для чего необходимо точно указать их имена. Например, в ОС *CP/M* версии 1.3 разрешается работа максимум с двумя дисками (A и B). С другой стороны, в ОС *CP/M* версии 1.4 — с четырьмя дисками (A, B, C и D). Наконец, ОС *CP/M* версии 2.0 и выше поддерживает максимум шестнадцать дисков (от A до P).

Для переназначения текущего диска необходимо ввести в качестве ответа на промпт ОС *CP/M* имя нового диска, двоеточие (:) и нажать клавишу "RETURN". Ниже представлен пример, иллюстрирующий переназначение текущего диска с именем "A" на диск с именем "B":

```
A>B:  
B>
```

Как показано в предыдущем примере, за именем нового текущего диска должен следовать символ ":" , с тем чтобы ОС *CP/M* могла распознать это имя как имя логического диска. Если двоеточие опустить, то символ "B" будет рассмотрен как имя команды. Например, если бы командный файл с именем "B.COM" хранился на диске с именем "A", то данная команда была бы выполнена. Если бы такого файла не оказалось, появилось бы следующее короткое сообщение об ошибке:

```
A>B  
B?  
A>
```

В любом случае это не дало бы желаемых результатов. Отметим, что в случае ошибки система сохранит предыдущее назначение текущего диска. Поэтому важно помнить, что при переназначении текущего диска или при указании имени диска в командной строке всегда необходимо вводить символ ":" . Другой пример: для того чтобы отменить назначение диску с именем "A" статуса текущего и назначить его диску с именем "M", следует в качестве ответа на промпт "A>" ввести "M:".

Прежде чем назначать новый текущий диск, необходимо сначала вставить его в соответствующее дисковое устройство**. Если в качестве нового текущего диска был назначен диск с именем "B", а соответствующее устройство не содержит диск, появится сообщение об ошибке, аналогичное представле-

* Иными словами, поиск или создание всех файлов (включая командные), необходимых для исполнения каждой последующей команды, по умолчанию будет производиться на текущем диске. — Прим. перев.

** С тем же именем. — Прим. перев.

ному ниже (точный текст сообщения об ошибке будет зависеть от версии ОС *CP/M*; например в некоторых модификациях ОС *CP/M* никакие сообщения об ошибках не выводятся):

A>B:

DRIVE NOT READY

При возникновении ошибок такого типа клавиатура консоли будет заблокирована и для того, чтобы реинициализировать систему, будет недостаточно ввести команду *CTRL/C*. В большинстве систем для устранения ошибки достаточно просто вставить диск в нужное устройство (в данном случае в устройство с именем "В") и закрыть дверцы устройства (если таковые имеются), после чего ОС *CP/M* автоматически реинициализирует саму себя, и на консоли появится промпт "В >". Если это не даст результата, то необходимо выполнить полную перезагрузку (сброс) системы, затем вставить диск в нужное устройство (например, в устройство с именем "В"), причем сделать это надо до переназначения текущего диска.

Если при назначении текущего диска (не диска "А") в устройство будет вставлен диск с несоответствующей плотностью записи или несовместимым с ОС *CP/M* форматом (это более вероятно), то на консоли появится сообщение об ошибке, разъясняющее суть возникшей проблемы (точный текст сообщения будет различным в зависимости от типа ошибки и версии ОС *CP/M*). В этом случае можно выполнить реинициализацию системы (ввести команду *CTRL/C*), а если это не поможет, то необходимо извлечь диск из устройства и провести полную перезагрузку (сброс) системы.

Если установлено, что данный диск совместим с ОС *CP/M*, но имеет некорректный формат, то прежде чем воспользоваться им снова, следует инициализировать с помощью имеющейся на системном диске программы формирования. Отметим, что после выполнения большинства используемых для этих целей форматирующих программ содержимое диска будет полностью уничтожено.

Если нужно обратиться к файлу, расположенному на произвольном отличном от текущего диске, не всегда необходимо назначать новый текущий диск. Например, к файлам, расположенным на диске с именем "В", можно обращаться и тогда, когда текущим является диск с именем "А". Это можно сделать, поставив перед именем файла в командных строках имя текущего диска. (Подробнее эта процедура будет описана ниже.)

ОБРАЩЕНИЕ К КОМАНДАМ И ФАЙЛАМ

Существует два способа хранения программ ОС *CP/M* на дисковых носителях, соответственно которым команды ОС *CP/M* разбиваются на две категории:

1. Встроенные команды (которые также называются "резидентными" командами).

2. Транзитные команды.

Встроенные команды. Встроенными командами являются те, которые постоянно находятся в оперативной памяти и по сути являются частью операционной системы *CP/M*. Обращение к этим программам и их использование осу-

ществляется простым вводом имени программы независимо от того, какой диск является в данный момент текущим. В операционной системе *CP/M* имеются следующие встроенные команды:

- DIP (Просмотр оглавления диска)
- REN (Переименование файлов)
- ERA (Удаление файлов)
- TYPE (Просмотр содержимого файлов)
- USER (Изменение "области пользователя") (имеется только в ОС *CP/M* версии 2.0 и выше)
- SAVE (Сохранение на диске содержимого системной области оперативной памяти)

В следующем примере показано, как для получения на консоли оглавления дискового устройства с именем А происходит обращение к команде DIP:

A>DIR

```
A: STAT COM : PIP COM : SYSGEN COM : MOVCPM COM  
A: FORMAT COM : LOAD COM : TEXT1 COM : DOT COM
```

A>

Отметим, что для этого понадобилось всего лишь набрать на клавиатуре символы "DIP" и нажать клавишу "RETURN". Кроме того, после строки "DIP" можно было через пробел ввести имя диска и символ ":" (например, "B:"). Это означает, что на консоль необходимо выдать оглавление диска с именем "B", при этом в качестве текущего остается диск с именем "A". Поскольку встроенные команды входят в состав ядра ОС *CP/M* и, следовательно, хранятся в оперативной памяти ЭВМ, то при их использовании не имеет значения, какой диск является в данный момент текущим. Можно было бы также ввести командную строку "DIR C:", но если устройство с именем "C" не подключено к системе, то появится сообщение об ошибке.

Транзитные команды. Транзитные команды — это те, которые хранятся на диске в отдельных файлах с именами, которые оканчиваются на ".COM". На самом деле транзитные команды являются программами ОС *CP/M*. Термин "транзитные" подчеркивает, что данные программы в действительности находятся в файлах, содержимое которых можно загрузить в оперативную память и затем удалить, тем самым отражает их временную природу. Приставка к имени файла (.COM) определяет тип файла и означает, что данный файл является "командным". Для вызова транзитных команд на исполнение достаточно просто ввести имя соответствующего файла без расширения ".COM". Обработка транзитных команд осуществляется аналогично обработке описанных выше встроенных команд. ОС *CP/M* определит наличие файла с указанным именем и расширением ".COM" в каталоге текущего диска и при его обнаружении вызовет данную программу на исполнение. В ОС *CP/M* имеются следующие стандартные командные файлы (программы):

- STAT.COM (Статус или характеристики)
- PIP.COM (Программа обмена данными между внешними устройствами)
- ED.COM (Текстовый редактор)
- SUBMIT.COM (Автоматический запуск последовательности команд ОС *CP/M*)

XSUB.COM	(Расширенная программа SUBMIT) (Только в ОС <i>CP/M</i> версии 2.0 и выше)
ASM.COM	(Ассемблер)
LOAD.COM	(Программа-сборщик)
DDT.COM	(Инструментальное средство динамической отладки программ)
DUMP.COM	(Просмотр в шестнадцатеричном формате содержимого файлов с объектным/машинным кодом)
SYSGEN.COM	(Генерация системы)
MOVCPM.COM	(Настройка ОС <i>CP/M</i> на новое адресное пространство)

В ОС *CP/M* может быть много других транзитных командных файлов, которые не приведены в этом перечне.

Описание всех стандартных транзитных команд дано в последующих главах. Ниже представлен пример, показывающий, как происходит обращение к команде STAT:

```
A>STAT
A: R/W, SPACE: 140K
A>
```

Как видно, здесь введено только имя командного файла без расширения ".COM". Если имя файла будет введено с расширением ".COM", то это расширение будет рассматриваться как часть командной строки, а на консоли появится сообщение об ошибке:

```
A>STAT.COM
STAT.COM?
A>
```

Когда бы не появилось это сообщение, оно означает, что ОС *CP/M* не может опознать указанную команду. Такая команда не существует или неправильно идентифицирована.

Обращение к файлам. Для того чтобы определить, находится ли на диске файл с транзитными командами или какой-либо другой файл, можно с помощью команды DIR просмотреть полное оглавление этого диска. Кроме того, командой DIR можно воспользоваться для того, чтобы определить, находится ли на данном диске конкретный файл. Для этого после команды DIR необходимо ввести спецификацию файла (filespec). Впредь обозначение filespec в этой книге будет использовано для представления последовательности символов, содержащей имя требуемого файла и дополнительно имя диска, содержащего этот файл. Спецификация файла имеет следующую структуру:

$$\text{filespec} = x : \text{filename} . \text{typ}.$$

Спецификация файла (filespec) всегда содержит имя файла (filename) и расширение, обозначающее тип файла (typ). Имя файла может быть длиной от одного до восьми символов. Расширение может быть длиной не более трех символов. Имена почти всех файлов имеют расширения. Обычно расширение используется для определения типа файла, хотя для некоторых файлов оно

не обязательно. Если в спецификации файла необходимо указать его тип (расширение), то расширение всегда должно быть отделено от имени файла при помощи точки ("."). Если расширение не используется, то указывать точку не нужно. Список наиболее употребительных типов файлов будет дан в этой главе. Для уточнения местонахождения файла с именем "filename.typ" дополнительно в спецификации файла может быть указано имя диска ("x:"). Если спецификация файла не содержит имени диска, то ОС CP/M предполагает, что указанный файл находится на текущем диске. Если имя диска содержится в спецификации файла, то его следует отделить от имени файла при помощи двоеточия (":"). Имена транзитных командных файлов и имена файлов, к которым обращаются встроенные и транзитные команды, всегда указываются в спецификации. Ниже приведены примеры правильных спецификаций файлов:

x : filename . typ

FILE-1
STAT.COM
A : LETTER1
B : PROGRAM.BAS

В следующем примере показано, как определить, находится ли файл с именем "STAT.COM" на текущем диске:

A>DIR STAT.COM

В следующем примере показано, как можно определить, находится ли файл с именем "STAT.COM" на диске с именем "B", в то время как текущим является диск с именем "A". Отметим, что в данном случае спецификация файла включает в себя имя диска.

A>DIR B:STAT.COM

Если данный файл находится на указанном диске, то в зависимости от того, какой именно диск был указан, на консоли появится сообщение

A: STAT COM

либо сообщение

B: STAT COM

Если данный файл не находится на указанном диске, то на консоли появится сообщение

NO SUCH FILE

После того как определено, что файл с именем "STAT.COM" находится на требуемом диске, к соответствующей транзитной программе можно обращаться, просто указывая имя "STAT". Она будет исполняться, поскольку ОС CP/M будет автоматически искать файл с данным именем и расширением ".COM", если тип файла в команде не указан. Несомненно, вводить команду STAT очень полезно, и если файла с именем "STAT.COM" нет на диске, то появится сообщение "NO SUCH FILE".

Если текущим является диск с именем "В", а нужно использовать команду "STAT", соответствующий файл которой находится на диске с именем "А", то нет необходимости переназначать текущий диск. Вместо этого следует ввести команду STAT в формате, представленном ниже:

```
B>A:STAT  
A: R/W, SPACE 140K  
B: R/W, SPACE 358K
```

в)

Для достижения требуемых результатов спецификация файла может включать в себя различные комбинации из имен командных файлов и дисков. В следующем примере показано, что текущим является диск с именем "В", транзитная команда ED, которую необходимо использовать, находится в файле (ED.COM) на диске с именем "А", а редактируемый файл находится на диске с именем "В":

```
B>A:ED B:TEXT1.TXT  
*  
*Q
```

в)

То, что в ОС *CP/M* можно пользоваться программами и обращаться к файлам, расположенным на дисках, которые не являются текущими, придает ей значительную гибкость. Например, если текущим является диск с именем "А", и он "переполнился", т. е. на него больше нельзя записывать данные, то можно поместить оставшиеся файлы на другой диск. Или, если надо выполнить программу, файл которой находится на диске, который не является текущим, с помощью команды DIR можно просмотреть оглавления всех дисков и выполнить программу, не переназначая текущий диск.

ИСПОЛЬЗОВАНИЕ МЕТАСИМВОЛОВ

Если спецификация файла указывается следом за командой, то в некоторых случаях имя файла в ней можно указывать не полностью. Это бывает удобно, когда имя файла полностью неизвестно или когда необходим доступ к группе файлов, имеющих сходные имена. Часть символов в имени файла или в расширении могут быть заменены метасимволами, что позволит ОС *CP/M* найти все файлы, имена которых приблизительно совпадают с тем именем, которое необходимо. В операционной системе *CP/M* определены метасимволы следующих двух типов:

- * (используется для обозначения группы символов);
- ? (используется для обозначения одного символа).

Метасимволы могут быть использованы при спецификации файлов со следующими стандартными командами ОС *CP/M*:

Встроенные команды: DIR и ERA.

Транзитные команды: STAT и PIP.

Для всех других команд требуется, чтобы связанные с ними спецификации файлов содержали конкретные имена. Использование метасимволов в данном случае приведет к возникновению ошибки.

ИСПОЛЬЗОВАНИЕ МЕТАСИМВОЛА “*”:

DIR TEST.*

В приведенном ниже примере демонстрируется использование метасимвола “*”. Здесь в спецификации файла указано конкретное имя, а вместо конкретного расширения указан метасимвол “*”. Следовательно, на консоль будет выдан список всех файлов, независимо от расширения:

A>DIR TEST.*

A: TEST DOC : TEST BAS : TEST INT : TEST TXT

В данном случае выдан список всех файлов с именем TEST.

DIR *.COM

В этом примере, наоборот, иллюстрируется замена имени файла метасимволом “*”, несмотря на то, что указывается расширение, т.е. тип файла. В данном случае на консоль выдается список файлов с тем же расширением независимо от имени файла:

A>DIR *.COM

A: SYSGEN COM : MOVCPRM COM : STAT COM : PIP COM
A: SUBMIT COM : XSUB COM : ED COM : ASM COM
A: DDT COM : DUMP COM : LOAD COM : FORMAT COM

DIR TEXT*.TXT

Кроме того, метасимвол “*” можно использовать в качестве символа “произвольного замещения” в имени файла для замены одиночного символа или группы символов в спецификации файла. В данном случае на консоль выдается список всех файлов, имена которых начинаются “TEXT” и которые имеют расширение “.TXT”:

A>DIR TEXT*.TXT

A: TEXT10 TXT : TEXT50 TXT : TEXT100 TXT : TEXT200 TXT

A>

ИСПОЛЬЗОВАНИЕ МЕТАСИМВОЛА "?"

DIR *.*?

Использование метасимвола "?" подобно использованию метасимвола "*", но с одним значительным отличием. Метасимвол "?" можно использовать для замены одиночных и символов в конкретных позициях спецификации файла. Другими словами, если есть сомнения относительно первого и последнего символов в расширении имени файла, но есть уверенность, что средний символ "O", то для обнаружения нужного файла чрезвычайно полезно воспользоваться метасимволом "?".

В следующем примере демонстрируется использование символа "?" в качестве символа замены при работе с командой DIR. Здесь командная строка имеет вид "DIR *?O?", что означает: "Выдать на консоль имена всех файлов с расширениями, у которых средний символ "O". Результат выполнения этой команды представлен ниже:

```
A>DIR *?O?  
A: SYSGEN COM : MOVCPM COM : DISKINFO DOC : STAT COM  
A: COS FOR : FLOW DOC : PIP COM : SAMPLE DOC  
A: SUBMIT COM : DISPLAY FOR : XSUB COM : ED COM  
A: ASM COM : ACCOUNT FOR : IDT COM : LOAD COM
```

A>

Подобным образом можно задавать любую комбинацию конкретных символов и метасимволов "?". Более того, в командной строке можно использовать любую комбинацию конкретных символов и метасимволов, если первым символом имени файла не является метасимвол "?". В имени файла может быть указано не более восьми метасимволов "?", а в расширении имени файла – не более трех.

КОМАНДЫ УПРАВЛЯЮЩИХ СИМВОЛОВ

В операционной системе CP/M имеется набор специальных команд "управляющих символов", которые используются для различных целей, в основном для редактирования командных строк. Данные команды отличаются от всех остальных команд ОС CP/M тем, что для их ввода на клавиатуре консоли используется специальная клавиша, которая в зависимости от типа терминала может называться "CTRL", "CNTL", "CNTRL" или "CONTROL". В данной книге для обозначения этой клавиши используется наиболее распространенное название "CTRL". Эта клавиша нажимается одновременно с одной или несколькими клавишами, обозначающими конкретную функцию. Поскольку обе клавиши нажимаются одновременно, для обозначения команды управляющих символов используется, например, запись вида CTRL/x, где x – некоторый символ. Управляющие символы при выводе на экран дисплея или на печатающее устройство будут отображаться как "^x", где "^" – символ, обозначающий CTRL.

КАК ПОЛЬЗОВАТЬСЯ КЛАВИШЕЙ CTRL

На клавиатуре консоли клавиша CTRL обычно расположена в крайнем левом ряду. Один хороший способ ввода команд управляющих символов: мизинец левой руки поместить на клавишу CTRL, затем указательным пальцем левой руки нажать нужные клавиши с левой стороны клавиатуры, а правым указательным пальцем нужные клавиши с правой стороны.

В следующих параграфах описывается назначение каждой команды управляющих символов ОС *CP/M*.

Команда реинициализации системы. Как было указано ранее, командой реинициализации ОС *CP/M* является команда CTRL/C.

Команды завершения ввода строки. Вместо обычной клавиши "RETURN" для завершения ввода командной или текстовой строки можно использовать одну из следующих специальных команд ОС *CP/M*:

1. CTRL/M: эта команда вызовет возврат каретки, что эквивалентно использованию клавиши "RETURN".
2. CTRL/J: эта команда вызовет перевод строки, что эквивалентно использованию клавиши "LINE FEED". Данная команда доступна только в ОС *CP/M* версии 2.0 и выше.

Команды удаления символов. Для удаления отдельных символов в командной строке можно использовать одну из следующих специальных команд ОС *CP/M*:

1. CTRL/H и BACKSPACE: Команда CTRL/H вызовет возврат курсора на одну позицию влево и таким образом стирание находящегося там символа. Это эквивалентно нажатию клавиши "BACKSPACE". Данная команда доступна только в ОС *CP/M* версии 2.0 и выше.
2. DELETE или RUBOUT: При нажатии этой клавиши будет удален символ, расположенный слева от курсора; удаленный символ будет отображен на консоли. В зависимости от типа терминала эта клавиша может называться "DEL", "DELETE" или "RUBOUT". Если в качестве консоли используется телетайп и по мере удаления символов их желательно распечатать, то вместо команды CTRL/H или BACKSPACE иногда используется рассматриваемая команда. В некоторых конфигурациях ОС *CP/M*, если в качестве консоли используется видеотерминал, эта команда модифицируется под аналогичные команды CTRL/H и BACKSPACE.

Команды удаления строки. Если ввод командной строки еще не завершен при нажатии клавиши "RETURN", то эту строку можно удалить, применив одну из следующих двух команд:

1. CTRL/U: приведет к удалению уже набранной командной строки. После этого курсор переместится в начало следующей строки. Эта команда доступна во всех версиях ОС *CP/M*. В ОС *CP/M* версии 2.0 и выше выполнение данной команды приведет к выводу на следующей строке символа "#". Часто этот символ используется в качестве символа конца строки при передаче сообщений между двумя пунктами, связанными посредством модема. Набрав команду CTRL/U, можно отменить обработку введенной информации операционной системой

CP/M, однако изображение введенной строки останется на экране консоли.

2. **CTRL/X:** В ОС *CP/M* версии 1.4 и выше эта команда выполняет те же действия, что и команда **CTRL/U**. В ОС *CP/M* версии 2.0 и выше в результате выполнения данной команды курсор возвратится в начало командной строки, а ее текст будет удален.

Команды управления выводом. Для управления выдачей текста на консоль в ОС *CP/M* можно использовать одну из следующих трех команд:

1. **CTRL/E.** Используется для переноса командной строки на следующую строку консоли, пока пользователь не нажал клавишу "RETURN". Когда клавиша "RETURN", наконец, нажата, все предыдущие строки, выведенные с помощью команды **CTRL/E**, будут восприняты ОС *CP/M* как единая командная строка. Эта команда доступна в ОС *CP/M* версии 1.4 и выше.
2. **CTRL/R.** Используется для повторного вывода командной строки, содержащей символы, удаленные при нажатии одной из клавиш "DEL", "DELETE" или "RUBOUT". В результате выполнения данной команды из командной строки будут исключены все удаленные символы, и она будет выдана в более разборчивом виде на следующей строке консоли. Эта команда доступна только в ОС *CP/M* версии 1.4 и выше. В ОС *CP/M* версии 2.0 в следующей строке, предшествующей повторно выведенной командной строке, помещается символ "#".
3. **CTRL/S.** Используется для временной задержки вывода текста на консоль. При первом использовании данной команды вывод текста прекращается, а при повторном — возобновляется. Подобную процедуру можно повторять столько раз, сколько потребуется. Эта команда доступна во всех версиях ОС *CP/M*.

Команда управления устройством печати. Команда управления устройством печати используется для вывода на устройство печати любого текста, который имеется на консоли. Она позволяет одновременно выводить текст на консоль и устройство печати. Команда называется **CTRL/P**. При первом использовании этой команды начинается вывод текста на устройство печати, а при повторном использовании — вывод текста прекращается. Подобные действия могут повторяться столько раз, сколько требуется. Эта команда доступна во всех версиях ОС *CP/M*.

СТАНДАРТНЫЕ ТИПЫ ФАЙЛОВ

Ниже (см. таблицу) перечислены расширения имен файлов, которые используются для определения типов файлов ОС *CP/M*. Некоторые из этих расширений необходимы для выполнения различных команд или программ ОС *CP/M*. Во многих случаях независимо от типа файла, введенного с консоли, перед выполнением команды производится автоматический поиск файла требуемого типа. Другие команды или программы *CP/M* будут просто производить проверку соответствия надлежащего типа файла специфицированному. Кроме того, существуют другие типы файлов, которые не требуются для работы самой ОС *CP/M* или для выполнения команд ОС *CP/M*, но которые рекомендуется использовать только для сохранения единства, с тем чтобы все файлы можно было опознавать по их типам.

Расширение имени файла	Тип файла	Пример
.ASM	Необходимо для файлов, обрабатываемых при помощи команды ASM, и обозначает, что файл содержит исходный код (текст) программы на ассемблере	DUMP.ASM PRINTER.ASM
.BAK	Обозначает, что данный файл является резервной копией другого файла. Подобные файлы автоматически порождаются текстовыми редакторами (например, ED) и другими текстовыми редакторами и процессорами непосредственно после открытия текстового файла	TEXT.BAK LETTER1.BAK
.BAS	Указывает, что файл содержит исходный текст программы на языке Бейсик. Наличие данного расширения у имени файла для некоторых трансляторов с языка Бейсик является необязательным, а для некоторых оно необходимо	ACCOUNT.BAS GAME1.BAS
.CRF	Указывает, что файл содержит таблицу перекрестных ссылок программы на ассемблере	COMP.CRF MAP.CRF
.COB	Указывает, что файл содержит исходный текст программы на языке Кобол	ACCOUNT.COB SEC.COB
.COM	Указывает, что файл содержит транзитную программу ОС CP/M. Это расширение должны иметь имена в всех файлах с транзитными программами ОС CP/M с тем, чтобы их можно было распознать как командные. Это дает право пользователю просто ввести имя соответствующего файла без расширения	STAT.COM PIP.COM
.DAT	Указывает, что файл содержит данные определенного типа (обычно символы в коде ASCII). В некоторых трансляторах с языка Фортран это расширение используется по умолчанию	PROG1.DAT TEXT.DAT
.DOC	Необязательное расширение для обозначения текстовых файлов (или документов)	INFO.DOC HELP.DOC
.FOR	Указывает, что файл содержит исходный текст программы на языке Фортран. Требуется для некоторых трансляторов с языка Фортран, а для других оно необязательно	DISPLAY.FOR MATH1.FOR
.HEX	Указывает, что файл создан в результате выполнения команды ASM и содержит шестнадцатеричный машинный код программы. Необходимо при обращении к команде LOAD с целью создания командного (*.COM) файла	DUMP.HEX PRINTER.HEX

Расширение имени файла	Тип файла	Пример
.INT	Указывает, что файл содержит промежуточный объектный код программы на языке Бейсик. Необходимо для большинства трансляторов с языка Бейсик, работающих в режиме компиляции	ACCOUNT.INT GAME1.INT
.LIB	Указывает, что файл является библиотечным и содержит текст. Может быть использовано для различных целей, но при работе с программой ED – для определения местонахождения замещения или добавления текста в другом текстовом файле, имя файла должно иметь расширение ".LIB"	BUF1.LIB TEXT1.LIB
.LST	Указывает, что файл содержит текст программы на языке Бейсик и сообщения об ошибках, что облегчает отладку	SPACE.LST ERRS.LST
.MAC	Указывает, что файл содержит исходный текст программы на ассемблере	BIOS.MAC PRINTER.MAC
.MSG	Обычно используется для тех же целей, что и расширение "DOC"	ERR.MSG READ-ME.MSG
.OVR	Указывает, что файл содержит программу оверлейной структуры. Используется в других программах, где требуется запоминание информации для оверлейных процедур	MSG1.OVR OVLY1.OVR
.PAS	Указывает, что файл содержит исходный текст программы на языке Паскаль. Необходимо для некоторых компиляторов с языка Паскаль	LEDGERS.PAS SCREEN.PAS
.PRN	Указывает, что файл содержит текст, подготовленный к выводу на печать, и создан текстовыми форматерами, трансляторами с языков программирования высокого уровня, ассемблерами и макроассемблерами	LOAD.PRN PRINTER.PRN
.REL	Указывает, что файл содержит перемещаемый объектный код и создан ассемблерами и другими трансляторами с языков программирования высокого уровня	CIRCLE.REL SCREEN.REL
.SUB	Указывает, что файл содержит команды ОС CP/M, подготовленные для автоматического запуска с помощью команды SUBMIT. Имена всех файлов, используемых командой SUBMIT, должны иметь расширение ".SUB"	INIT.SUB LOAD.SUB
.SYM	Указывает, что текстовый файл, содержащий таблицу символов, создан некоторыми типами макроассемблеров	BOOT.SYM BIOS.SYM

Расширение имени файла	Тип файла	Пример
.TEX	При работе с некоторыми текстовыми форматерами указывает файл с исходным текстом	MANUAL.TEX CHAPT1.TEX
.TXT	Используется для тех же целей, что и расширение ".DOC"	LETTER.TXT INFO.TXT
.QQQ*	Указывает временный файл, созданный программами PIP, ED и др.	MANUAL.QQQ INFO.QQQ

* Вместо символа Q может использоваться символ \$. – Прим. перев.

Как можно было убедиться, в ОС *CP/M* существует много возможных типов файлов. Расширения имен, соответствующие этим типам, не всегда являются понятными пользователю настолько, чтобы определить, что они обозначают. Более того, разработчики часто дают собственные обозначения типов файлов, создавая тем самым еще большую путаницу. Например, один разработчик называет оверлейным ("OVR") файл "WSMSGS.OVR", имя которого является сокращением от английского "wordstar messages"*. Если возникают сомнения относительно типа файла, можно обратиться к данной таблице, а если здесь он отсутствует, обратитесь к документации, поставляемой вместе с программным продуктом.

СРАВНЕНИЕ ВЕРСИЙ ОС *CP/M*

С 1975 г., когда была разработана первоначальная версия ОС *CP/M*, для ее усовершенствования было сделано много изменений. Спустя годы эти изменения были внесены в различные версии ОС *CP/M*. В представленных ниже таблицах отражены изменения, внесенные в ОС *CP/M*, начиная с версии 1.3, вплоть до наиболее поздней версии 3.0.

ВСТРОЕННЫЕ КОМАНДЫ

Команда	Версии ОС <i>CP/M</i>					
	1.3	1.4	2.0	2.2	3.0	3.0 BANKED
DIR	Есть	Отличие: Выдает на консоль по 4 спецификации файлов на одну строку; выдаются спецификации только обычных файлов "DIR" и спецификации файлов, принадлежащих исключительно текущему пользователю		Отличие: Содержит дополнительные режимы работы, которые заменяют выполнение функций команды STAT. Выдает на консоль спецификацию файла, размер, количество записей, атрибуты, статус защиты, дату и время начала и окончания. При необходимости обеспечивает обращение к файлу "DIR.COM"		

* Буквально означает "сообщения текстового процессора Wordstar". – Прим. перев.

Команда	Версии ОС CP/M					
	1.3	1.4	2.0	2.2	3.0	3.0 BANKED
DIRSYS	Нет				Отличие: Выполняет те же действия, что и команда "DIP", но выдает на консоль характеристики системных файлов	
ERA или (ERA или ERA-SE только в версии 3.0)	Есть	Отличие: После ввода команды "ERA *.*" выдает запрос "ALL?"	Отличие: Удаляет файлы, принадлежащие исключительно текущему пользователю	Отличие: Добавлен режим "CONFIRM" ("Подтверждение")		
REN	Есть				Отличие: Использует метасимволы; работает в режиме диалога при использовании команды без параметров; имеет возможность удаления существующего файла "NEWNAME"; при необходимости производит обращение к файлу "RENAME.COM"	
SAVE	Есть	Отличие: Не меняет содержимое области памяти пользователя	Отличие: Вызывается до того, как данные будут загружены в память с помощью программы SID; при завершении работы программы запрашивает спецификацию файла, при необходимости удаляется прежний файл с данной спецификацией, и запрашивает начальный и конечный адреса сохраняемой области памяти			
TYPE	Есть				Отличие: Добавлен режим "PAGE", обеспечивающий приостановку вывода на каждой 14-й строке	
USER	Нет	Есть	Отличие: Работает в режиме диалога при вводе без параметров, используя промпт "ENTER USER #:"; выдает в составе промпта вместе с именем диска идентификатор текущего пользователя, если он больше нуля (1A>, 5C> и т. д.)			

ТРАНЗИТИНЫЕ КОМАНДЫ

Команда	Версии ОС CP/M							
	1.3	1.4	2.0	2.2	3.0	3.0 BANKED		
ASM	Есть				Нет (заменена командами MAC и RMAC)			
COPSYS	Нет (аналогична команде SYSGEN)				Есть			
DATE	Нет							
DDT	Есть				Нет (заменена командой SID)			
DEVICE	Нет				Есть			
DUMP	Есть				Отличие: Данные выдаются в шестнадцатеричном и ASCII форматах			
ED	Есть	Отличие: +/-V, 0V, nX, 0X, nnnn:, :mmmm, E, H, Q, O. Символы табуляции всегда отображаются в виде пробелов	Отличие: +V установлено по умолчанию. Не может изменить содержимое файла, для которого установлен режим доступа "только чтение". Запрещено обращение к системным файлам. Добавлен параметр n2	Отличие: Специфицирует входной и выходной файлы; при вводе команды "ED" без параметров имена файлов задаются в режиме диалога. Команда "X" выполняется для n строк. Примеры: n X filespec ^ Z, 0X ^ Z, 0X filespec ^ Z				
FORMAT	Зависит от конкретного пакета программ и поставщика ОС CP/M. Форматирующая программа может иметь другое имя, такое как INIT, FMT, INITDSK, IN и т. д. Стандартной форматирующей программы не существует							
GENCPM	Нет (аналогична команде MOVCPM)				Есть			
GENCOM	Нет							
GET								
HELP	Нет (специальная версия данной программы может быть предоставлена поставщиками ОС CP/M, приобретена дополнительно или создана пользователем)							
HEXCOM	Нет (аналогична команде LOAD)				Есть			

Команда	Версии ОС CP/M					
	1.3	1.4	2.0	2.2	3.0	3.0 BANKED
INITDIR	Нет					
LIB	Дополнительная					
LINK						
LOAD	Есть					Нет (заменена командой HEXCOM)
MAC	Дополнительная					Есть
MOVCPM	Есть					Нет (заменена командой GENCPM)
PATCH	Нет					Есть
PIP	Есть	Отличие: Дополнительные параметры. Имена физических устройств. Не полностью специфицированные имена файлов	Отличие: Параметры: [Gn], [R], [W]	Отличие: Добавлены физические устройства AXI и AХО. При работе с текущим устройством его имя можно не указывать. Дополнительная возможность для подтверждения копирования. Возможно копирование между областями разных пользователей		
PUT	Нет					Есть
RMAC	Дополнительная					
SET	Нет					Есть
SETDEF	Нет					Есть
SHOW	Нет					Есть
SID	Дополнительная					Есть

Окончание табл.

Команда	Версии ОС CP/M					
	1.3	1.4	2.0	2.2	3.0	3.0 BANKED
STAT	Есть Выдает на консоль объем свободной памяти на диске	Отличие: Выдает на консоль размер каждого файла; назначение физических устройств логическим устанавливает вход в режим "только чтение"	Отличие: Режим "меню" (параметр "VAL"); параметры "DSK:", "X:DSK", "USR:", "S". Атрибуты файлов "R/W", "R/O", "DIR", "SYS"			
		Обнаруживает смену дисков. Выдает объем свободного пространства на всех активных дисках				
SUBMIT	Есть		Отличие: Работает вместе с командой XSUB	Отличие: команда XSUB, встроена в команду SUBMIT. Файлы SUBMIT и GET могут быть вложены в любой комбинации. Файлы SUBMIT ищутся на связных дисках		
SYSGEN	Есть				Нет (заменена командой COPSYS)	
XREF	Дополнительная				Есть	
XSUB	Нет	Есть	Нет (функция команды включена в программу SUBMIT, см. команду GET)			

КОМАНДЫ УПРАВЛЯЮЩИХ СИМВОЛОВ

Команда	Версии ОС CP/M					
	1.3	1.4	2.0	2.2	3.0	3.0BANKED
CTRL/A	Нет	-	-	-	-	Есть
CTRL/B						
CTRL/C	Есть					

Окончание табл.

Команда	Версии ОС CP/M					
	1.3	1.4	2.0	2.2	3.0	3.0 BANKED
CTRL/E	Нет	Есть				
CTRL/F	Нет					Есть
CTRL/G						
CTRL/H или BACKSPACE	Нет		Есть			
CTRL/I или TAB						
CTRL/J или LINEFEED						
CTRL/K	Нет					Есть
CTRL/M или RETURN CTRL/P	Есть					
CTRL/R	Нет	Есть	Помещает символ "#" в текущей позиции курсора перед повторным выводом только что набранной командной строки			
CTRL/S	Есть					
CTRL/U	Есть		Помещает символ "#" в текущей позиции курсора и переводит курсор на следующую командную строку			
CTRL/W	Нет					Есть
CTRL/X	Нет	Аналогична команде CTRL/U	Аналогична команде CTRL/U, однако она возвращает курсор в начало строки и стирает все ее содержимое			
DEL, DELETE, RUBOUT	Есть					

СПЕЦИФИКАЦИИ ДИСКА

Спецификация	Версии ОС CP/M				
	1.3	1.4	2.0	2.2	3.0
Максимальное количество дисковых устройств	2	4	16		
Максимальный объем памяти на один диск		1 мегабайт	16 мегабайт (допускается увеличение объема)		512 мегабайт
Максимальный размер файла		128 килобайт	8 мегабайт		32 мегабайт
Максимальное количество файлов на диске		64		Допускается увеличение размера в зависимости от плотности записи	
Метод доступа		Последовательный		Последовательный или прямой	
Место хранения характеристик диска	BDOS	Параметр диска		Блок параметров диска (DPB) расположен в резидентной секции BIOS	

НОВЫЕ ИЛИ ИЗМЕНЕННЫЕ ФУНКЦИИ СЕКЦИИ BDOS

Функция	Версии ОС CP/M				
	1.3	1.4	2.0	2.2	3.0
0 – Сброс системы	Есть				
1 – Ввод с консоли					
2 – Вывод на консоль					
3 – Ввод с перфоленты	Есть				Нет
Вспомогательный ввод	Нет				Есть
4 – Вывод на перфоленту	Есть				Нет
Вспомогательный вывод	Нет				Есть
5 – Вывод на устройство печати:	Есть				
6 – Прямой ввод-вывод с консоли	Нет		Есть		
7 – Получить байт ввода-вывода	Есть				Нет

Продолжение табл.

Функция	Версии ОС CP/M										
	1.3	1.4	2.0	2.2	3.0						
Статус вспомогательного устройства ввода	Нет			Есть							
8 – Установить байт ввода-вывода	Есть			Нет							
Статус вспомогательного устройства вывода	Нет			Есть							
9 – Печатать строку	Есть										
10 – Читать буфер консоли	Есть	Улучшенные средства редактирования строк									
11 – Получить статус консоли	Есть										
12 – Подвести головку диска											
Получить номер версии	Нет	Есть									
13 – Сброс дисковой системы	Есть										
14 – Выбрать диск											
15 – Открыть файл	Есть	Улучшенная версия									
16 – Закрыть файл	Есть										
17 – Найти первый блок файла	Есть	Улучшенная версия									
18 – Найти следующий блок файла	Есть	Улучшенная версия									
19 – Удалить файл											
20 – Последовательное чтение	Есть										
21 – Последовательная запись	Есть										
22 – Создать файл	Есть	Улучшенная версия									
23 – Переименовать файл											
24 – Получить вектор регистрации											
25 – Получить имя текущего диска	Есть										

Функция	Версии ОС CP/M								
	1.3	1.4	2.0	2.2	3.0				
26 – Установить адрес области пользователя на диске	Есть								
27 – Получить адрес размещения									
28 – Защитить диск от записи	Нет	Есть							
29 – Получить адрес вектора ввода-вывода									
30 – Установить атрибуты файла									
31 – Получить адрес области параметров диска									
32 – Получить/установить код пользователя	Нет	Есть							
33 – Прямое чтение									
34 – Прямая запись									
35 – Вычислить размер файла									
36 – Установить запись прямого доступа									
37 – Сброс дискового устройства	Нет	Есть							
38 и 39	Нет	Не определено							
40 – Прямая запись с заполнением нулями	Нет	Есть							
41 – Проверить и записать	Нет	Есть							
42 – Захват записи файла									
43 – Освободить запись файла									
44 – Установить значение счетчика секторов:									
45 – Секции BDOS перейти в режим обработки ошибок									

Функция	Версии ОС CP/M				
	1.3	1.4	2.0	2.2	3.0
46 – Получить информацию об объеме свободного дискового пространства					
47 – Установить связь с программой					
48 – Переписать буфера на диск	Нет				Есть
49 – Считать/записать блок управления системой					
50 – Непосредственные вызовы секции BIOS					
с 51 по 58	Нет				Не определены
59 – Оверлейная загрузка или расширение резидентной части системы:	Нет				Есть
60 – Вызвать расширение резидентной части системы					
с 61 по 97	Нет				Не определено
98 – Освободить блоки занятых файлов	Нет				Есть
99 – Уменьшить размеры файла					
100 – Записать метку каталога					
101 – Получить данные о метке каталога					
102 – Считать даты обращения к файлу и пароль					
103 – Записать блок управления файлом XFCB					
104 – Установить дату и время	Нет				Есть
105 – Получить дату и время					

Функция	Версии ОС <i>CP/M</i>				
	1.3	1.4	2.0	2.2	3.0
106 – Записать значения пароля по умолчанию					
107 – Получить серийный номер ЭВМ					
108 – Получить/установить код возврата программы					
109 – Получить/установить режим работы консоли					
110 – Получить/установить разделитель вывода	Есть				Есть
111 – Отпечатать содержимое блока					
112 – Вывести содержимое блока на консоль					
Со 113 по 151	Нет				Не определено
152 – Разбор имени файла	Нет				Есть

Прямой вызов "ввод-вывод с консоли" был добавлен в версию 2.0 операционной системы *CP/M*, а в версиях 1.3 и 1.4 он отсутствует. Этот вызов позволяет пользователю исключить используемые для редактирования и фильтрации входных данных стандартные символы ОС *CP/M* (CTRL/S, CTRL/Q, CTRL/X и т. д.) [6]. Системный вызов с кодом 10 ("Читать буфер консоли") расширяет возможности ОС *CP/M* по редактированию строк, введенных пользователем с консоли. В версии 2.0 были усовершенствованы системные вызовы, обеспечивающие открытие, поиск следующего, удаление, создание и переименование файлов. Самое важное – были добавлены системные вызовы, обеспечивающие прямой метод доступа к файлам (вызовы с 33 по 36: "Прямое чтение", "Прямая запись", "Вычислить размер файла", "Установить запись прямого доступа"). Кроме того, в версию 2.0 были добавлены вызовы для защиты дисков от записи, установления атрибутов файла и получения значения кода пользователя. Системные вызовы с кодами с 37 по 152 были добавлены в версию 3.0 (некоторые из этих вызовов не определены, а зарезервированы для использования в будущем). Большинство из этих системных вызовов, такие как "Оверлейная загрузка или расширениерезидентной части системы", "Считать дату обращения, дату записи файла и па-

роль", "Получить данные о метке каталога" и т. д., применяются наряду с дополнительными особенностями ОС *CP/M* версии 3.0. По сравнению с другими версиями некоторые из этих вызовов обеспечивают расширение основных возможностей ОС *CP/M*. Обзор ОС *CP/M* версии 3.0 представлен в гл. 21.

Эта справочная информация завершает гл. 2 и ту часть книги, в которой в общих чертах описывается операционная система *CP/M*. В последующих главах рассматриваются отдельные команды ОС *CP/M*, поддерживаемые версиями с 1.3 по 2.2.

ГЛАВА 3



Версии ОС *CP/M*:

1.3 1.4 2.0 2.2

Тип команды:

встроенная

ОБЩИЕ СВЕДЕНИЯ

Команда DIR обеспечивает обращение к специальной встроенной функции, которая позволяет пользователю ОС *CP/M* выводить на консоль список имен файлов, расположенных на некотором диске. Этот список представляет собой "каталог" файлов с программами, данными, текстами, а также другой хранимой на диске информацией, дает сведения об именах и типах файлов, которые хранятся на диске, вставленном в конкретное устройство (описание допустимых имен и типов файлов см. в гл. 2). При использовании в спецификации имени файла метасимвола ("*" и "?") по команде DIR можно получить полный список имен всех файлов, содержащихся на каком-либо конкретном диске, список имен файлов, принадлежащих к определенной группе, либо имя одного специфицированного файла.

Для того чтобы использовать эту команду, достаточно в качестве ответа на промпт ОС *CP/M* (A >, B > и т. д.) ввести строку "DIR", за которой, кроме того, может следовать спецификация имени файла.

ОБРАЩЕНИЕ К КОМАНДЕ DIR

ВЫВОД НА КОНСОЛЬ ПОЛНОГО
ОГЛАВЛЕНИЯ ДИСКА



DIR

Использование команды "DIR" в ее простейшей форме позволяет получить полное оглавление текущего диска. В следующем примере показано, как осуществляется вывод на консоль оглавления диска с именем "A":

A>DIR

```
A: STAT COM : PIP COM : SYSGEN COM : MOVCPRM COM
A: FORMAT COM : LOAD COM : ASM      COM : DDT      COM
A: SUBMIT COM : XSUB COM : ED       COM
```

A>

Видно, что, когда команда DIR вводится без спецификации имени файла, на консоль выводится полное оглавление текущего диска. Отметим, что при завершении вывода оглавления на консоли снова появится основной промпт ОС *CP/M* (например, "A >"), что указывает на готовность ОС *CP/M* к обработке другой команды.

Команду DIR можно, кроме того, использовать для получения оглавления любого, а не текущего диска. Для этого после команды DIR необходимо ввести имя диска (например, "B:" или "C:"). Имя диска должно быть отделено от команды DIR при помощи пробела. В следующем примере показано, как получено оглавление диска с именем "B", когда текущим является диск с именем "A":

A>DIR B:

```
B: STAT      COM : PIP      COM : TEXT01 DOC : YBASIC COM
B: ACCOUNTS BAS : ACCOUNTS TXT : LIBRARY LIB : TEXT01 BAK
B: SUBMIT    COM : XSUB    COM : ED      COM
```

A>

Даже если в данной командной строке был указан диск с именем "B", после завершения вывода оглавления на консоли появится промпт "A >", поскольку текущим пока является диск с именем "A".

Если бы в устройство был вставлен правильно отформатированный чистый диск, совместимый с ОС *CP/M*, то в ответ на команду DIR на консоль было бы выдано следующее сообщение:

A>DIR B:

NO FILE

A>

Это сообщение было бы выдано, кроме того, при использовании команды DIR для получения списка имен файлов, которые отсутствуют на диске. То же самое происходит, если специфицировано имя несуществующего файла.

Использование команды DIR для вывода на консоль имен конкретных файлов рассматривается ниже.

ВЫВОД НА КОНСОЛЬ ОГЛАВЛЕНИЯ ДЛЯ
ОДНОГО ФАЙЛА ИЛИ ГРУППЫ ФАЙЛОВ

DIR TEXT01.DOC
DIR *.DOC

В следующем примере продемонстрировано использование команды DIR вместе со спецификацией имени файла (в данном случае "TEXT01"). Если

файл существует на диске, то на консоль будут выданы только его имя и расширение:

```
A>DIR TEXT01.DOC  
A: TEXT01 DOC  
A>
```

Команда DIR часто используется для получения списка группы имен файлов. Это осуществляется при помощи определенных в ОС CP/M метасимволов (см. гл. 2). Метасимволы также полезны, когда полное имя файла не известно, но когда предполагается, что известны некоторые символы из этого имени. В следующем примере (см. рис. 3.5) иллюстрируется использование в спецификации метасимвола "*" для замены имени файла с расширением "DOC". Таким образом, будет дан список только имен файлов, которые в качестве последних трех символов (расширение) содержат символы "DOC":

```
A>DIR *.DOC  
A: TEXT01 DOC : TEXT05 DOC : MANUAL1 DOC : INFO DOC  
A: ACCOUNTS DOC : HELP DOC  
A>
```

Задержка вывода оглавления. Во многих случаях оглавления дисков весьма длинные. В результате этого выведенная информация будет вытеснена за верхнюю часть экрана (если в качестве консоли используется видеотерминал) до полного окончания вывода оглавления. В подобных случаях можно одновременно нажать клавиши "CTRL" и "S", что "заморозит" перемещение текста на экране дисплея. Это позволит не спеша читать оглавление по частям. При повторном вводе команды CTRL/S перемещение текста на экране дисплея возобновится. Подобная процедура может повторяться столько раз, сколько необходимо. При использовании высокоскоростных терминалов (4800 Бод и выше) после ввода команды DIR ответ на нее может появиться очень быстро, и при этом информация может быть утеряна до того, как она будет прочитана. Рекомендуется предварительно поместить пальцы на клавиатуре консоли так, чтобы ввести команду "CTRL/S" до того, как оглавление исчезнет с экрана.

Завершение выполнения команды DIR. При нажатии любой клавиши, отличной от "CTRL/S", перемещение оглавления на экране приостановится, и выполнение команды DIR завершится, после чего на консоли снова появится основной промпт ОС CP/M (A>, B> и т. д.).

ГЛАВА 4

REN

Версии ОС *CP/M*:

1.3 1.4 2.0 2.2

Тип команды:

встроенная

ОБЩИЕ СВЕДЕНИЯ

Команда REN обеспечивает обращение к специальной встроенной функции ОС *CP/M* для переименования файлов, содержащихся в каталоге диска; может быть использована в любое время независимо от того, какой диск используется. После того, как файл переименован, его новое имя хранится на диске до тех пор, пока оно не будет изменено либо пока файл не будет удален. За одно обращение к команде REN может быть переименован только один файл, поэтому в спецификации имен файлов для этой команды всегда требуется указывать явные имена. Метасимволы совместно с командой REN использоваться не могут.

ОБРАЩЕНИЕ К КОМАНДЕ REN

ПЕРЕИМЕНОВАНИЕ ФАЙЛА,
РАСПОЛОЖЕННОГО НА ТЕКУЩЕМ ДИСКЕ

REN NEW.TXT = OLD.TXT

Для изменения имени файла необходимо использовать специальный формат команды. Может показаться, что для того, чтобы переименовать файл, нужно произнести: "Заменить старое имя на новое". Операционная система *CP/M* подобно большинству операционных систем выполняет эти действия по-другому. Команда REN осуществляет переименование файла "логическим" путем, что характерно для большинства ЭВМ, то есть формат командной строки для переименования файла является обратным обычному. Таким образом, процедуру переименования, выполняемую по команде REN, можно перефразировать так: "Переименовать: новое имя эквивалентно старому имени". Это важное отличие необходимо запомнить, поскольку все другие команды ОС *CP/M*, для которых требуется специфицировать два или более файла, работают аналогичным образом. Это можно легко запомнить при помощи следующего правила: "Новое – слева, старое – справа" (в строю на левом фланге обычно стоят более молодые люди, чем на правом). В следующем примере иллюстрируется изменение имени файла с "OLD.TXT" на "NEW.TXT":

```
A>DIR *.TXT  
A:OLD  TXT  
A>REN NEW.TXT=OLD.TXT  
A>DIR *.TXT  
A:NEW  TXT
```

Как показано, для проверки наличия на диске файла, который должен быть переименован, можно воспользоваться командой DIR. После переименования файла можно снова воспользоваться командой DIR, с тем чтобы убедиться в том, что файл переименован правильно. При переименовании файлов необходимо указывать полное имя, включая расширение (если оно есть) того файла, который должен быть переименован. Для нового имени файла указывать расширение не обязательно, однако рекомендуется использовать одно из допустимых расширений, поскольку тип файла при этом будет определен. При попытке переименовать несуществующий файл (файл неправильно специфицирован либо отсутствует на диске) на консоль будет выдано следующее сообщение об ошибке:

```
A>REN NEW.TXT=OLDFILE.TXT  
NO FILE  
A>DIR OLDFILE.TXT  
NO FILE  
A>
```

В ОС CP/M версии 2.0 или выше для файлов обычно устанавливается конкретный статус защиты. Это означает, что имя файла и его содержимое могут быть изменены только в определенном случае. Файлу можно установить статус защиты "R/O" ("Только чтение") или "R/W" ("Чтение/запись"). Файлы со статусом защиты "R/O" являются полностью защищенными и не могут быть изменены (в частности, переименованы). При попытке переименовать защищенный файл на консоль будет выдано сообщение об ошибке:

```
A>REN NEW.TXT=OLD.TXT  
FILE IS SET R/O          <---- Для того чтобы возвратить промпт,  
                           нажимается клавиша RETURN  
A>
```

В предыдущем примере файлу "OLD.TXT" назначен статус защиты "R/O" ("Только чтение") и переименование файла впоследствие запрещено. Для получения информации о том, как проверить состояние защиты файла перед изменением его имени, можно обратиться к гл. 8, где дано описание команды STAT.

ПЕРЕИМЕНОВАНИЕ ФАЙЛА,
РАСПОЛОЖЕННОГО НА ПРОИЗВОЛЬНОМ ДИСКЕ

REN B:NEW.TXT=OLD.TXT

Команду REN можно использовать для переименования файла, расположенного на произвольном (не обязательно на текущем) диске. Для этого в первую спецификацию, введенную за командой REN (новое имя файла), необходимо включить имя диска. Во вторую спецификацию (старое имя файла) включать имя диска не надо. В следующем примере продемонстрировано переименование файла, расположенного на диске с именем "B", в то время как текущим остается диск с именем "A":

A>REN B:NEW.TXT=OLD.TXT

A>

В командной строке команды REN имя диска должно присутствовать только один раз и предшествовать как новому, так и старому именам файла. Командная строка, представленная ниже, является также допустимой, но не обязательной:

A>REN B:NEW.TXT=B:OLD.TXT

A>

Если имя диска указано в несоответствующем месте или если командная строка содержит два различных имени диска, то возникнет ошибка. В следующем примере приведены неправильно составленные командные строки:

A>REN NEW.TXT=B:OLD.TXT
B:OLD.TXT?

A>
A>REN B:NEW.TXT=A:OLD.TXT
A:OLD.TXT?

A>

Сообщения об ошибках, возникающих при переименовании файлов, расположенных на текущем диске, совпадают с сообщениями об ошибках, возникающих при переименовании файлов, расположенных на произвольном диске.

ГЛАВА 5

The logo consists of the letters "ERA" in a bold, stylized font, enclosed within a thick black rectangular border.

Версии ОС *CP/M*

1.3 1.4 2.0 2.2

Тип команды:

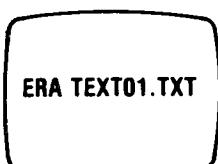
встроенная

ОБЩИЕ СВЕДЕНИЯ

Команда ERA обеспечивает обращение к специальной встроенной функции ОС *CP/M*, которая используется для удаления хранящихся на диске файлов. С помощью команды ERA можно удалять как файлы, расположенные на текущем диске, так и файлы, расположенные на произвольном диске. В командной строке можно специфицировать один файл, группу файлов или все файлы, расположенные на одном диске.

ОБРАЩЕНИЕ К КОМАНДЕ ERA

УДАЛЕНИЕ ОДНОГО ФАЙЛА



ERA TEXT01.TXT

Для того чтобы удалить файл, необходимо просто в качестве ответа на промпт ОС *CP/M* ввести команду ERA и полное имя того файла, который должен быть удален. Если такой файл существует на диске и не защищен каким-либо способом, то его имя из каталога соответствующего диска будет удалено. Удаленный файл будет недоступен для стандартных средств обращения к файлам ОС *CP/M**. В представленном ниже примере показано, как осуществляется удаление файла "TEXT01.TXT", находящегося в каталоге текущего диска и затем проверка того, что он более не является доступным:

```
A>DIR TEXT01.TXT
A:TEXT01  TXT
A>ERA TEXT01.TXT
A>DIR TEXT01.TXT
NO FILE
A>
```

* По команде ERA реально ничего не уничтожается. Удаление файла состоит всего лишь в том, что в каталоге диска перед его именем ставится код, означающий, что данный элемент каталога свободен. Для восстановления удаленных файлов можно использовать специальную программу, которая распространяется коммерческим образом. В книге [6] описано, как восстанавливать файлы с помощью программы "DDT".

Командой ERA следует пользоваться с величайшей осторожностью: удаленный файл в дальнейшем становится недоступным. Поэтому перед тем, как использовать команду ERA, следует дважды удостовериться в том, что действительно нужно удалить этот файл. Хорошей является практика создания резервных копий всех важных файлов, так что если некоторый файл случайно удален, то можно его восстановить с резервного диска. (Как создавать резервную копию диска см. в гл. 2.)

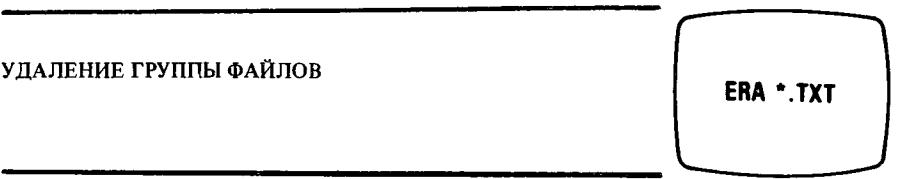
С помощью команды ERA можно, кроме того, удалять файлы, расположенные на произвольном (не обязательно текущем) диске. Для этого в спецификацию файла необходимо включить имя нужного диска.

Например:

```
A>ERA B:TEXT05.TXT
```

```
A>
```

УДАЛЕНИЕ ГРУППЫ ФАЙЛОВ



```
ERA *.TXT
```

Группа файлов, предназначенных для удаления при помощи команды ERA, может быть специфицирована в одной командной строке с использованием метасимволов так, как это описано в гл. 2. Использование любой комбинации метасимволов и обычных символов в спецификации файла приведет к тому, что с соответствующего диска будут удалены все файлы, попадающие под эту категорию. Для того чтобы не удалить файл, содержащийся в группе файлов, но который нежелательно удалять, следует соблюдать осторожность.

КАК БЕЗОПАСНО ИСПОЛЬЗОВАТЬ МЕТАСИМВОЛЫ В СПЕЦИФИКАЦИИ ФАЙЛОВ ДЛЯ КОМАНДЫ ERA

Во избежание неожиданностей, прежде всего, используйте команду DIR, за ней точно такую же последовательность метасимволов и обычных символов для проверки того, что все файлы, содержащиеся в этой группе, должны быть удалены. Если в группе, которую необходимо удалять, содержится один или более файлов, то их необходимо переименовать так, чтобы они не попадали в эту группу. Другая мера предосторожности предполагает удаление каждый раз по одному файлу.

Пример: необходимо удалить все файлы, входящие в группу “*.TXT” (т. е. все файлы с расширением “.TXT”). Здесь (для получения списка имен этих файлов) используется команда DIR, после чего устанавливается, что файл с именем “SAVE-ME.TXT” удалять не надо. Затем этот файл переименовывается так, чтобы можно было использовать предполагаемую командную строку команды ERA:

```
A>DIR *.TXT  
A:TEXT01  TXT : TEXT02  TXT : SAVE-ME TXT  
A>REN SAVE-ME.DOC=SAVE-ME.TXT  
A>ERA *.TXT  
A>DIR *.TXT  
NO FILE  
A>
```

УДАЛЕНИЕ ВСЕХ ФАЙЛОВ НА ДИСКЕ

ERA *.*

Команда ERA может быть использована для удаления всех файлов, расположенных на одном диске. В этом случае спецификация файла должна содержать последовательность символов “*.*”, которая указывает, что в процессе удаления необходимо задействовать все возможные имена файлов и их расширения. В ОС CP/M версии 1.4 и выше в ответ на такую командную строку появится запрос, действительно ли необходимо удалить все файлы. Это средство обеспечивает двойной контроль. В более ранних версиях ОС CP/M такой двойной контроль не производится. Если в ответ на этот запрос будет введен символ “Y”, что означает “Да”, то все файлы будут удалены. Если будет введен символ “N”, что означает “Нет”, то выполнение команды ERA будет прекращено до того, как будет удален хотя бы один файл. Отметим, что если символ “у” будет введен в нижнем регистре, то удаление файлов выполнено не будет, поскольку предполагается, что пользователь имеет в виду ответ “N”. Оба случая проиллюстрированы в представленном примере:

```
A>ERA *.*  
ALL (Y/N)?N      <---- Не было удалено ни одного файла  
A>  
A>ERA *.*  
ALL (Y/N)?Y      <---- Все файлы были удалены  
A>
```

Если при использовании команды ERA заданный файл или группа файлов не будут обнаружены на диске, то на консоль будет выдано сообщение об ошибке. Этот случай проиллюстрирован в следующем примере:

```
A>ERA PROG1.BAS
```

```
NO FILE
```

```
A>
```

```
A>DIR PROG1.BAS
```

```
NO FILE
```

```
A>
```

Как указывалось ранее, в ОС *CP/M* версии 2.0 и выше обычно всем файлам назначается определенный статус защиты: "R/W" ("Чтение/запись") или "R/O" (Только чтение"). Файлы, имеющие статус защиты "R/O", являются полностью защищенными, т. е. ни имя файла, ни его содержимое не могут быть изменены никаким образом. Информацию о статусах защиты файла см. в описании команды STAT в гл. 8. При попытке удалить файл, имеющий статус защиты "R/O", на консоль будет выдано следующее сообщение об ошибке:

```
A>ERA PROG15 BAS
```

```
FILE IS SET R/O
```

----- Для того чтобы возвратить промпт,
нажимается + павиша RETURN

```
A>
```

В этом примере файлу "PROG15.BAS" был назначен статус защиты "R/O" и удаление этого файла, таким образом, стало невозможным. О том, как проверить статус защиты файла до использования команды ERA и как его изменить, см. в описании команды STAT, в гл. 8.

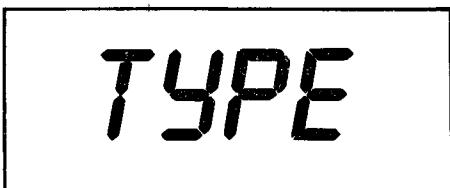
ГЛАВА 6

Версии ОС *CP/M*:

1.3 1.4 2.0 2.2

Тип команды

встроенная



ОБЩИЕ СВЕДЕНИЯ

Команда TYPE обеспечивает обращение к встроенной функции, которая позволяет пользователю ОС *CP/M* выводить на консоль содержимое текстового файла. Поскольку команда TYPE является встроенной, к ней можно обращаться в любое время независимо от того, какой диск является текущим. Команда TYPE может использоваться для вывода на консоль содержимого любых файлов, в которых имеются тексты, пригодные для вывода на экран или печатающее устройство (например, файлы с расширениями ".TXT", ".DOC", ".HEX", ".PRN" и т. д.). Обработка с помощью команды TYPE файлов, содержащих тексты с непечатаемыми символами, приводит к непредсказуемым последствиям.

ОБРАЩЕНИЕ К КОМАНДЕ TYPE

ВЫВОД СОДЕРЖИМОГО ФАЙЛА
ПРИ ПОМОЩИ КОМАНДЫ TYPE

TYPE TEXT01.TXT

Для вывода на консоль содержимого некоторого текстового файла необходимо в качестве ответа на промпт ОС *CP/M* (A>, B> и т. д.) ввести команду TYPE и спецификацию этого файла. Спецификация файла для команды TYPE не должна содержать метасимволы:

A>TYPE TEXT01.TXT

QUOTES:

A SHIP ON THE BEACH IS A LIGHTHOUSE TO THE SEA.
- DUTCH PROVERB

NEVER GO TO SEA WITH TWO CHRONOMETRS; TAKE ONE OR THREE.
- ANONYMOUS

THE IMPORTANT THING IS NOT TO STOP QUESTIONING. CURIOSITY
HAS ITS OWN REASON FOR EXISTING. - ALBERT EINSTEIN

A>

При обнаружении конца файла вывод его содержимого на консоль прекращается, и на экране снова появляется основной промпт ОС *CP/M*. Команду TYPE можно использовать для просмотра содержимого файла, расположенного на произвольном (не обязательно текущем) диске, для чего спецификация файла должна содержать имя этого диска. При использовании команды TYPE можно указывать только конкретные файлы, т. е. использование метасимволов в спецификации файлов для команды TYPE запрещено. Если специфицированный файл не содержится на указанном диске, то на консоль будет выдано следующее сообщение об ошибке:

A>TYPE TEST20.ASM

TEST20.ASM?

A>DIR TEST20.ASM

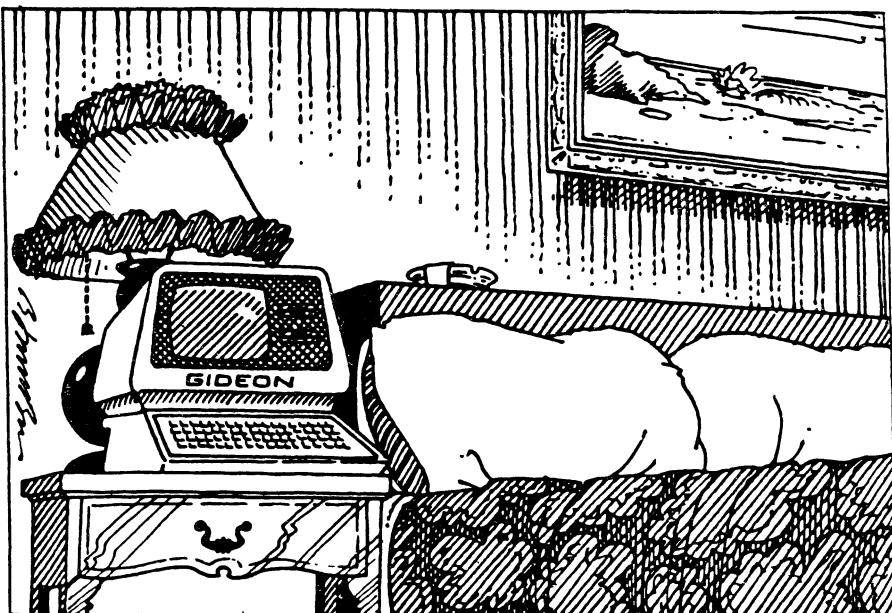
NO FILE

A>

То же самое сообщение будет выдано на консоль в том случае, если в имени файла содержатся некорректные символы (например, метасимвол "*" или "?").

Задержка просмотра содержимого файла. Во многих случаях длина специфицированного для команды TYPE файла будет превышать размеры экрана (если в качестве консоли используется видеотерминал). В результате этого часть выведенных данных будет вытеснена за верхнюю часть экрана до того, как будет обнаружен конец файла. В этом случае задержки перемещения

текста на экране необходимо одновременно нажать клавиши "CTRL" и "S" (CTRL/S), что позволяет просматривать содержимое файла по частям. При повторном вводе команды CTRL/S перемещение на экране возобновится. Данную процедуру можно повторить столько раз, сколько потребуется.



Прекращение выполнения команды TYPE. При нажатии любой, но не CTRL/S клавиши вывод содержимого файла прекращается и выполнение команды TYPE завершается, после чего на консоль будет выдан основной промпт ОС CP/M (A >, B > и т. д.).

Если специфицированный с командой TYPE файл не пригоден к выводу на консоль, т. е. не является текстовым файлом, то выводимая на экране информация может состоять из последовательности непонятных символов. Иногда, когда это происходит, промпт ОС CP/M на консоли не появляется, а система становится временно заблокированной. Обычно для того, чтобы возвратить промпт ОС CP/M, достаточно выполнить реинициализацию системы (команда CTRL/C). Если это не помогает, то необходимо выполнить сброс и полную перезагрузку системы. Следует избегать вывода на консоль содержимого такого типа файлов, поскольку выводимая информация в большинстве случаев будет бессмысленной и бесполезной.

USER

Версии ОС *CP/M*:

2.0 2.2

Тип команды:

встроенная

ОБЩИЕ СВЕДЕНИЯ

Команда USER доступна только в ОС *CP/M* версии 2.0 и выше. Эта команда позволяет нескольким различным пользователям ОС *CP/M* обращаться к их собственным областям на диске. На одном диске может быть определено до 16 различных областей пользователя с номерами от 0 до 15. Это, однако, не означает, что несколько пользователей могут одновременно работать с одним и тем же диском. ОС *CP/M* была разработана как однопользовательская система, тогда как ОС *MP/M* предназначается для работы многих пользователей. Команда USER позволяет записывать программы в различные области на одном и том же диске, а затем каждая область может быть предназначена конкретному пользователю.

ОБРАЩЕНИЕ К КОМАНДЕ USER

После загрузки ОС *CP/M* по умолчанию областью пользователя на системном диске является область с номером 0. Это означает, что до тех пор, пока системе не указана иная область, областью пред назначенной пользователю, будет область с номером 0. Каталог файлов ограничен, так как в него входят только те файлы, которые принадлежат области пользователя с кодом 0 (при этом, однако, можно использовать все свободное пространство на диске). При первоначальном определении новая область пользователя (например, область с номером 1) не содержит никаких файлов и ее каталог будет пустым. Те файлы, которые находятся в области пользователя с кодом 0 (или в какой-либо другой области), недоступны пользователю, работающему с областью под номером 1 и наоборот. После выбора конкретного кода пользователя соответствующая область пользователя становится активной на всех дисках независимо от того, какой из них является текущим*.

* Можно пересыпать программы между различными областями пользователя при помощи транзитной команды PIP или, что менее удобно, при помощи встроенной команды SAVE.

СОЗДАНИЕ ИЛИ ИЗМЕНЕНИЕ ОБЛАСТЕЙ ПОЛЬЗОВАТЕЛЯ

USER 1

Для создания на всех активных дисках новой области пользователя или для замены одной конкретной области пользователя на другую необходимо в качестве ответа на промпт ОС *CP/M* (A>, B> и т. д.) ввести команду USER, а за ней номер требуемой области. Номер области пользователя должен быть в интервале от 0 до 15.

A>USER 1

A>

После ввода этой команды на консоль не выдается никаких сообщений, кроме основного промтга ОС *CP/M* (A>, B> и т. д.). Однако в результате введения этой команды в системе произошли некоторые изменения. В новой области пользователя будет недоступен каталог файлов области с номером 0. Если ввести команду DIR, а новая область пользователя еще не содержит ни одного файла, то в ответ будет получено сообщение "NO FILES". При первоначальном создании области пользователя в ней доступными остаются только встроенные команды ОС *CP/M* (DIR, REN, ERA, TYPE и т. д.). Все файлы с транзитными командами ОС *CP/M* недоступны, поскольку они находятся в другой области пользователя.

При вводе команды USER с некорректным кодом пользователя на консоль будет выдано следующее сообщение об ошибке:

A>USER 16

16?

A>

Приведенное выше сообщение об ошибке было выдано на консоль потому, что, хотя всего имеется 16 областей пользователя, области с номером 16 не существует. Команда USER отсылает к первой области пользователя с номером 0 и к последней — с номером 15.

Первоначальное создание области пользователя. Если область пользователя создается впервые, то обратите внимание на то, что в ней не содержится никаких файлов. Это продемонстрировано в следующем примере:

A>USER 1

A>DIR

NO FILE

A>

Для того чтобы создать каталог файлов в новой области пользователя, в ней необходимо иметь файл с программой копирования (аналогичный фай-

лу "PIP.COM"). Наличие файла "PIP.COM" (файл с транзитной программой, соответствующей команде PIP) делает возможным копирование в новую область пользователя файлов из других областей пользователя (см. описание команды PIP). Ниже описывается последовательность шагов, которые необходимо выполнить для копирования файла "PIP.COM" в новую область и предоставления его таким образом пользователю.

1. При нахождении в области пользователя с номером 0 проверьте с помощью команды STAT, что файл "PIP.COM" действительно хранится на диске в этой области пользователя. Обратите внимание на размер файла. Размер файладается в килобайтах (обычно равен "8K"). Более важно, однако, запомнить значение, расположенное в колонке под именем "RECS". Это значение обычно равно 58 и представляет собой количество 128-байтных записей данного файла. Запишите на бумаге точное количество записей файла – оно потребуется в дальнейшем.

2. При помощи транзитной команды DDT загрузите файл "PIP.COM" в оперативную память (введите командную строку "DDT PIP.COM"). Как только на консоли появится промпт транзитной программы, соответствующий команде DDT (обычно это символ "-"), выйдите из программы при помощи команды CTRL/C (реинициализация системы). После этого на консоли снова появится промпт с именем текущего диска. Содержимое файла "PIP.COM" по-прежнему сохраняется в оперативной памяти.

3. Введите команду USER и номер требуемой области пользователя, например "USER3".

4. Как только новая область пользователя выбрана, при помощи встроенной команды SAVE перепишите в новую область содержимое файла "PIP.COM", по-прежнему хранящееся после загрузки с помощью команды DDT в оперативной памяти. Однако для использования команды SAVE сначала следует подсчитать количество страниц оперативной памяти (длина одной страницы равна 256 байтам), которое требуется под файл "PIP.COM". Для того чтобы это сделать, восстановите количество записей этого файла, полученное на первом шаге при помощи команды STAT, умножьте это количество на 128 (длина одной записи равна 128 байтам), и получите общее число байтов, занимаемое исходным файлом. Затем для получения общего количества страниц оперативной памяти разделите результат на 256*. Окончательный результат используется вместе с командой SAVE с тем, чтобы сохранить и заполнить содержимым файла "PIP.COM" новую область пользователя. Например, если количество записей, занимаемых файлом "PIP.COM", равно 58, то для того, чтобы получить соответствующее число страниц оперативной памяти, необходимо произвести следующие вычисления*:

$$58 \text{ записей} \times 128 \text{ байт} = 7424 \text{ байт (почти 8K)};$$

$$7424 \text{ байт} : 256 \text{ байт} = 29 \text{ страниц оперативной памяти},$$

* Авторы явно усложнили процедуру подсчета. Поскольку одна запись занимает ровно 0,5 страницы оперативной памяти, то для получения требуемого результата достаточно просто разделить общее количество записей файла на 2 и округлить результат в сторону большего целого (например, 58 : 2 = 29). – Прим. перев.

5. После того как с помощью команды STAT файл "PIP.COM" сохранен на диске, проверьте с помощью команды DIR, что этот файл действительно записан на диск. После этой проверки для копирования файлов из других областей пользователя можно использовать команду PIP. Чтобы специфицировать номер области пользователя, из которой необходимо выполнять копирование, в командную строку команды PIP добавляется специальный параметр (см. описание команды PIP в гл. 9).

Отметим, что при помощи команды PIP операционная система *CP/M* всегда обеспечивает только копирование из других областей пользователя в необходимую область. Копировать файлы в другие области пользователя нельзя. Прежде чем приступить к выполнению этой процедуры, ознакомьтесь с описаниями упомянутых выше команд и примерами их использования, представленными в других разделах данной книги. Поняв, как пользоваться этиими командами, возвратитесь к данному разделу и следуйте инструкциям. Изменение областей пользователя. Максимальное число областей пользователя на одном диске равняется 16 (номера от 0 до 15). Для перехода в нужную область пользователя необходимо просто ввести команду USER с номером области.

Вывод номера активного пользователя и номеров непустых областей пользователя. Это осуществляется при использовании транзитной команды STAT. Для того чтобы воспользоваться этой командой, следует поместить файл "STAT.COM" в каталоге в рабочей области пользователя. Команда используется следующим образом:

```
A>STAT USR:  
ACTIVE USER: 0  
ACTIVE FILES: 0 1 2  
A>
```

В приведенном примере текущей областью пользователя является область с номером 0, областями пользователя, содержащими файлы, являются области с номерами 0, 1, 2. Остальные области пользователя являются пустыми.

ГЛАВА 8

Версии ОС *CP/M*:

1.3 1.4 2.0 2.2

Тип команды:

транзитная



ОБЩИЕ СВЕДЕНИЯ

Наряду с командой DIR команда STAT является одной из наиболее часто используемых команд ОС *CP/M*. Команда STAT (сокращение от английского Status или Statistics – состояние или характеристики) обеспечивает вызов

на исполнение транзитной программы, которая хранится на диске в файле под именем "STAT.COM". Команда широко применяется пользователями ОС *CP/M* для получения списка имен файлов, определения размеров файла, определения объема неиспользованного дискового пространства (в килобайтах) и для определения значений различных атрибутов файла (ключ защиты и пр.). Программа, соответствующая команде STAT, выполняет множество других функций, которые используются не так часто, но все обеспечивают дополнительную гибкость в управлении файлами и дисками. В данном разделе рассматриваются все варианты использования команды STAT. Для каждой указывается соответствующая командная строка. Ниже представлено краткое описание функций команды STAT.

Вывод на консоль объема свободного пространства на диске (ОС *CP/M* версии 1.3 или выше) и режима доступа для всех дисков (только ОС *CP/M* версии 2.0 или выше). Используя команду STAT в ее простейшей форме, можно определить объем свободного пространства на диске. В ОС *CP/M* версии 2.0 или выше проверка наличия свободного пространства на диске включает в себя проверку статуса защиты, назначенного всему диску*.

Печать характеристик файлов. Подобно команде DIR команду STAT можно использовать для вывода на консоль списка файлов, хранящихся на диске. Однако при помощи команды STAT можно получить информацию, о размере каждого файла и оставшемся объеме неиспользованного пространства на диске. В ОС *CP/M* версии 2.0 или выше по команде STAT, кроме того, выдается на консоль статус защиты каждого файла ("R/O" или "R/W"). С помощью команды STAT можно получать оглавление любого диска, установленного в какое-либо подключенное к ЭВМ устройство. Спецификация файла для команды STAT, как и в случае команды DIR, может содержать любые метасимволы ("*" и "?"), что позволяет получать информацию как об одном файле, так и о группе файлов или обо всех файлах на диске.

Защита файлов и дисков (только ОС *CP/M* версии 2.0 и выше). Отдельные файлы или целые каталоги могут быть защищены так, что их содержимое нельзя будет удалить до тех пор, пока данная защита не будет отменена. Это осуществляется путем присваивания файлу (файлам) статуса защиты "R/O" ("Только чтение"), которое означает, что можно выводить содержимое данного файла (файлов) на консоль или устройство печати, но нельзя удалить или изменить никакую часть его содержимого. Наоборот, статус защиты "R/W" ("Чтение/запись") наряду с операцией чтения допускает изменение и удаление содержимого всего файла или отдельных его частей. Несмотря на кажу-

* *Статус защиты.* Любой диск считается защищенным от записи, если его статус защиты "R/O" ("Только чтение"), и считается незащищенным, если "R/W" ("Чтение/запись"). С защищенного диска можно только считывать информацию, т. е. никакие операции, связанные с добавлением, удалением или изменением данных, выполнять нельзя. Защита диска от записи необходима, если содержимое его файлов не будет никогда переписано или изменено. С другой стороны, с незащищенного диска можно считывать информацию или производить на него запись, т. е. выполнять любые операции, связанные с добавлением, удалением или изменением данных. Команду STAT можно адресовать любому диску, установленному в устройство, подключенное к ЭВМ при условии, что диски надлежащим образом опознаны операционной системой *CP/M*.

щуюся ограниченность, режим "только чтение" является весьма полезным, поскольку часто желательно или даже необходимо защитить определенные файлы (а не весь диск), с тем чтобы их содержимое не могло быть случайно удалено. Таким образом, можно устанавливать режим "только чтение" как для всего диска, так и для отдельных файлов. Однако если необходимо удалить или изменить содержимое защищенного файла, то с помощью команды STAT можно легко отменить защиту файла или всего диска, назначив файлам или всему диску статус защиты "R/W" ("Чтение/запись").

"Скрытые" файлы (только ОС CP/M версии 2.0 и выше). Отдельным файлам или целым каталогам могут быть назначены специальные "атрибуты", при наличии которых имена файлов не будут выводиться на консоль по команде DIR. Атрибуты назначаются, когда желательно сделать какой-либо файл для команды DIR. Подобное "упрятывание" файла может выполняться для того, чтобы избавить пользователя от необходимости просматривать длинные оглавления файлов, особенно в тех случаях, когда программа содержит много маленьких модулей. Имена скрытых файлов могут быть выведены на консоль только с помощью транзитной команды STAT. Не считая отсутствия имени файла в оглавлении, полученном по команде DIR, все остальные характеристики файла при назначении ему атрибута "скрытый" остаются те же самые. Этот атрибут в любой момент можно отменить, назначив файлу с помощью команды STAT атрибут с противоположным значением.

Определение номеров активных пользователей (только ОС CP/M версии 2.0 или выше). Бывают ситуации при которых в результате выполнения команды USER текущим пользователем становится пользователь с кодом, отличным от принятого по умолчанию кода 0. В случае, если пользователь не помнит, какой код пользователя он указал вместе с командой USER, можно воспользоваться транзитной командой STAT для определения текущего пользователя (например, пользователь 1, пользователь 2, ..., пользователь 15) и количества областей других пользователей, разметивших файлы на данном диске. Подключение к диску пользователя с кодом, отличным от принятого по умолчанию, позволяет создавать и обрабатывать вместе с другими пользователями файлы на одном и том же диске. При этом невозможно "читать" файлы других пользователей, а они не смогут "читать" ваши файлы (исключение составляют те пользователи, которые подключаются к системе с кодом пользователя, принадлежащим кому-нибудь другому). Это означает, что управление файлами (чтение, запись, удаление, переименование) осуществляется только в соответствии с номером пользователя. См. описание команды USER.

Определение характеристик диска. Для отображения информации о хранении данных на конкретных дисках, которые в ОС CP/M считаются активными, можно использовать команду STAT. Эта информация включает в себя имя физического устройства (A, B, C и т. д.) и определенные сведения о том, как сформирован диск. Эти сведения зависят от типа диска (программно-секторированный, аппаратно-секторированный или жесткий диск) и от типа примененного формата (единичная или двойная плотность записи). В результате обработки команды STAT может быть получена следующая информация: максимальная емкость диска (в килобайтах), количество входов в каталог плюс сведения об использовании дорожек и секторов данного диска.

Назначение логических имен устройствам и просмотр логических и физичес-

ких имен устройств. Для назначения таким периферийным устройствам, как видеотerminalы, телетайпы, модемы, логических имен (например "CON:", "PTR:", "LPT:") используется команда STAT. Назначение имен необходимо для того, чтобы ОС CP/M могла соответствующим образом опознавать данные устройства. При помощи команды STAT можно одновременно назначать и переназначать столько имен, сколько допускается в ОС CP/M. Команда STAT обеспечивает вывод на консоль списка существующих (назначенных) логических имен, а также получение списка всех возможных имен устройств. Можно, например, назначить логическое имя устройства печати ОС CP/M "физическому" устройству печати или "физическому" видеотerminalу.

ОБРАЩЕНИЕ К КОМАНДЕ STAT

Команда STAT является транзитной командой, соответствующая программа которой хранится в своем файле. Обычно этот файл называется "STAT.COM", хотя его можно было бы назвать по-разному, например "STATUS.COM", "DISK.COM", "FILES.COM", "S.COM" и т. д. Прежде чем использовать команду, необходимо, чтобы этот файл находился на диске, вставленном в подключенное к ЭВМ устройство. Для того чтобы удостовериться, что этот файл действительно находится на одном из доступных дисков, следует использовать команду DIR.

Как только установлено, что файл "STAT.COM" находится на заданном диске, достаточно ввести команду "STAT", поскольку ОС CP/M автоматически осуществляет поиск файла типа ".COM", если в командной строке отсутствует расширение имени файла. Вызов команды STAT можно осуществить с любого диска путем указания перед командой STAT имени диска или требуемого устройства.

СПОСОБЫ ПРИМЕНЕНИЯ КОМАНДЫ STAT

ВЫВОД ОБЪЕМА СВОБОДНОГО ПРОСТРАНСТВА
И СТАТУСА ЗАЩИТЫ ДЛЯ ВСЕХ ДИСКОВ

STAT

При последовательном вводе команд STAT и RETURN на консоль выдается информация о состоянии текущего диска. Например:

```
A>STAT
A: R/W, SPACE: 140K
```

В ОС CP/M версии 2.0 или выше сначала выводится статус защиты диска, который указывает, установлен ли для этого диска режим "Чтение/запись" (значение "R/W") или же установлен режим "Только чтение" (значение "R/O"). В приведенном выше примере ключ защиты диска равен значению "R/W", которое означает, что содержимое данного диска можно читать, изменять или удалять. При выводе статуса защиты "R/O", содержимое диска можно толь-

ко читать, изменение или удаление любых данных запрещено. Последняя по счету характеристика (здесь 140К) обозначает объем оставшегося на диске свободного для доступа пространства (выраженный в килобайтах). В ОС *CP/M* версий 1.3 и 1.4 значение статуса защиты не приводится. В этих версиях выдается только объем свободного пространства на диске. В нашем примере объем свободного пространства равен 140К, и это означает, что на диске имеется примерно 143 360 байтов (140×1024), или 140 килобайтов свободного пространства, которое можно использовать для хранения данных.

В случае если в устройства, отличные от устройства с именем "A", вставлены диски, которые считаются в ОС *CP/M* активными (при вставке диска в устройство введена команда *CTRL/C* или же эти диски явно указаны в *каких-либо командных строках*), в результате обработки команды *STAT* на консоль выводятся объемы свободного пространства и статусы защиты для всех таких дисков. Это иллюстрируется следующим примером:

```
A>STAT
A: R/W, SPACE: 140K
B: R/W, SPACE: 358K
C: R/W, SPACE: 402K
D: R/W, SPACE: 210K
A>
```

Эта информация выводится независимо от того, в ответ на какой промпт ("A >", "B >", "C >" или "D >") введена команда *STAT*. В данном примере полученный отчет содержит те же сведения, что и в предыдущем. Для того чтобы при использовании команды *STAT* диск, вставленный в устройство с именем "B", считался активным, необходимо, чтобы после того, как этот диск был вставлен в устройство, произошла реинициализация системы (выполнена команда *CTRL/C*) или чтобы этот диск был явно указан в какой-либо предыдущей командной строке. Иногда при невыполнении указанных условий после обработки команды *STAT* выводится только информация о свободном пространстве и статусе защиты диска, вставленного в устройство с именем "A". Однако, для того чтобы нужный диск стал активным, достаточно просто ввести команду *STAT*.

ВЫВОД ОБЪЕМА СВОБОДНОГО ПРОСТРАНСТВА
И СТАТУСА ЗАЩИТЫ ДЛЯ КОНКРЕТНОГО ДИСКА

STAT A:
STAT B:
STAT C:
STAT D:

В случае, если после команды *STAT* указано имя какого-либо устройства ("A", "B", "C" и т. д.), то выводится обычный отчет, содержащий значение объема оставшейся на этом диске свободной памяти. Например:

```
A>STAT B:
BYTES REMAINING ON B: 358K
A>
```

В данном примере показано, что такой отчет содержит только значение объема памяти, оставшейся неиспользованной на диске, который вставлен в устройство с именем "B". Значение ключа защиты диска здесь не представлено.

ВЫВОД НА КОНСОЛЬ СТАТУСА
ЗАДАННЫХ ФАЙЛОВ

STAT TEST.DOC

Если после команды STAT специализировано имя какого-либо файла, то выдается таблица характеристик этого файла. Эти характеристики содержат размер файла, статус защиты ("R/W" или "R/O") и количество записей в данном файле. Верхняя часть каждого столбца такой таблицы представляет собой заголовок, который определяет, значение какой характеристики файла расположено под ним:

```
A>STAT TEST.DOC
RECS    BYTES   EXT ACC
 110     14K     1 R/W A:TEST.DOC
BYTES REMAINING ON A: 476K
```

A>

STAT TEST.DOC \$\$

Дополнительно в командную строку "STAT TEST.DOC" в этом столбце включен атрибут "размер" ("size"). Однако этот атрибут никогда не используется, поскольку это приводит лишь к дублированию информации, напечатанной в столбце под именем "RECS". Ниже показано, как выглядит отчет в этом случае.

```
A>STAT TEST.DOC *S
SIZE RECS    BYTES   EXT ACC
 110    110    14K     1 R/W A:TEST.DOC
BYTES REMAINING ON A: 476K
```

A>

Представленные характеристики имеют следующий смысл: SIZE. В этом столбце печатается количество записей в файле. Это значение совпадает с тем, которое печатается в столбце под именем "RECS". Данная характеристика является необязательной и печатается только тогда, когда в качестве последнего параметра команды STAT указан атрибут "Q S".

RECS. Запись является единицей измерения объема хранимых на диске данных и используется в ОС CP/M для повышения эффективности обработки

файла путем группирования байтов. В ОС *CP/M* одна запись содержит 128 байт. Отметим, что на каждые 1024 байт, или 1 килобайт, приходится по 8 записей. В представленном выше примере число в столбце "RECS" показывает, что в данном файле содержится сто десять 128-байтовых записей.

BYTES. Число, напечатанное в этом столбце, представляет собой округленное значение фактического размера файла в килобайтах. В приведенном выше примере это число равно 14K, что означает 14 килобайт, или 14 336 байт (14×14 , где $1K = 1024$ байт). Однако, на самом деле значение 14K отражает объем, меньший, чем 14 336, потому что число записей файла, умноженное на 128, равно значению, меньшему, чем 14K. Для более точного определения размера, чтобы более точно определить размер файла в байтах, можно провести следующие расчеты:

$$\text{Количество записей} \times 128 \text{ байтов} = \text{Размер файла в байтах}$$
$$110 \quad \times 128 \quad = 14\,080$$

Таким образом, размер данного файла равен 14 080 байт, а не 14 336. В результате этого может создаться впечатление, что в файле имеется 256 байтов, или 2 неиспользованные дорожки. На самом деле это не лишено оснований. Определение точной цифры размера файла дополнительno усложняется тем, что в ОС *CP/M* файлы хранятся на диске большими порциями, которые называются "единицы распределения". В ОС *CP/M* версии 2.2 при двойной плотности записи на диск единица распределения равна 2K, или 2048 байтам. Таким образом, число в столбце "BYTES" представляет собой размер файловой системы с двойной плотностью записи, выраженный в единицах распределения, и не отражает реального размера файла в записях. В результате минимальный размер всех файлов равен 2K. Если для хранения файла необходим объем памяти, больший, чем 2K, то в столбце "BYTES" печатается значение 4K, поскольку память под файл распределяется с приращением 2K. В случае каких-либо сомнений точный размер файла необходимо определять путем умножения на 128 числа, которое напечатано в столбце "RECS" при использовании команды STAT. Отметим, что для дисков с единичной плотностью записи единица распределения равна 1K и при этом значение в столбце "BYTES" является более точным.

EXT. Экстент является еще одной единицей измерения размеров файла в ОС *CP/M*. Каждый экстент равен 16 килобайтам дисковой памяти. Таким образом, если файл занимает не более 16K, то печатается только единица. Если размер файла превышает 16K, то количество экстентов увеличивается на 1 для каждого дополнительных 16K, занимаемых файлом. В приведенном выше примере число в колонке "EXT" означает, что файл состоит из одного "логического экстента". Как правило, цифру в столбце "EXT" можно не принимать во внимание.

ACC. Наименование "ACC" является сокращением от английского Access (доступ). Символы, напечатанные в этом же столбце, представляют собой статус защиты файла ("R/W" – "Чтение/запись", "R/O" – "Только чтение"). Данная характеристика включается в отчет только в ОС *CP/M* версии 2.0 или выше.

В последней позиции в приведенном примере печатается полная спецификация файла. В обычных условиях наиболее важной характеристикой в этом отчете является число, напечатанное под заголовком "BYTES", поскольку

здесь печатается приближенное значение размера файла. Размер файла в байтах можно сравнить с объемом свободного пространства на диске. Например, эту информацию можно было бы использовать для того, чтобы определить, что для размещения файла размером в 82К можно использовать диск, объем свободного пространства которого равен 83К.

Ниже представлено несколько примеров, которые иллюстрируют различные варианты использования команды STAT.

ВЫВОД НА КОНСОЛЬ ХАРАКТЕРИСТИК ВСЕХ ФАЙЛОВ

STAT *.*

В результате обработки этой команды выдаются характеристики всех файлов, которые хранятся на текущем диске. Как следует из приведенного ниже примера, по команде STAT выдается список имен файлов (в алфавитном порядке) в соответствии с именем файла) и их расширений.

A>STAT *.*

RECS	BYTES	EXT	ACC
35	6K	1	R/W A:COS.FOR
258	34K	3	R/W A:DISKINFO.DOC
151	20K	2	R/W A:FLOW.DOC
1	2K	1	R/W A:INIT.SUB
58	8K	1	R/W A:SAMPLE.DOC
10	2K	1	R/W A:SINE.FOR
41	6K	1	R/W A:STAT.COM
71	10K	1	R/W A:TAN.FOR
110	14K	1	R/W A:TEST.DOC
149	20K	2	R/W A:TEST.BAS
149	20K	2	R/W A:TEST.INT
BYTES REMAINING ON A: 476K			

Помимо вывода полного оглавления диска команду STAT можно использовать для вывода списка имен заданной группы файлов. Это можно сделать тем же способом, что и в случае использования команды DIR. Можно применить те же комбинации метасимволов ("*" и "?"), которые рассмотрены в гл. 2. Ниже приведено несколько примеров таких командных строк:

STAT FILE.*
STAT *.TYP
STAT FOR *.*
STAT FOR ??..T?T
STAT FOR ??.*
... и т. д.

УПРАВЛЕНИЕ ВЫВОДОМ КОМАНДЫ STAT

Во многих случаях каталоги дисков содержат так много файлов, что начало отчета, полученного по команде STAT *.*., вытесняется за верхнюю часть экрана дисплея до того, как его можно будет прочитать. Для того чтобы это предотвратить, в момент ввода команды необходимо поставить свои пальцы рядом с клавишами

CTRL и S. Как только на экране начнет перемещаться текст оглавления, необходимо одновременно нажать клавиши CTRL и S. В результате печать оглавления останавливается до повторного ввода команды CTRL/S. Такую операцию можно повторять многократно.

ЗАЩИТА ФАЙЛОВ

STAT TEST.DOC \$R/O
STAT TEST.DOC \$R/W

В ОС CP/M версии 2.0 или выше с помощью команды STAT можно устанавливать и отменять защиту отдельных файлов и целых каталогов. Часто бывает желательно защитить файл так, чтобы его нельзя было случайно удалить. При этом нужно намеренно отменить защиту файла прежде, чем этот файл можно будет удалить.

Ниже приведен пример использования параметров команды STAT, обеспечивающих защиту файла от записи:

```
A>STAT TEST.DOC
RECS  BYTES  EXT  ACC
 110    14K    1 R/W A:TEST.DOC
BYTES REMAINING ON A: 476K
```

В этом примере файлу TEST.DOC назначен статус защиты "R/W". Предположим, что надо защитить файл так, чтобы его нельзя было уничтожить. Для этого можно ввести следующую командную строку:

```
A>STAT TEST.DOC >R/O
TEST.DOC SET TO R/O
A>STAT TEST.DOC
RECS  BYTES  EXT  ACC
 110    14K    1 R/O A:TEST.DOC
BYTES REMAINING ON A: 476K
```

После назначения статуса защиты файлу "R/O", можно еще раз использовать команду STAT с тем, чтобы распечатать характеристики этого файла и проверить значение статуса защиты. Отметим, что параметр "ACC" теперь имеет значение "R/O". При попытке удалить файл, со статусом "R/O", выдается следующее сообщение об ошибке*:

```
A>ERA TEST.DOC
BDOS ERR ON A: FILE R/O
```

* При описании команды ERA (см. гл. 5) для аналогичной ситуации (попытка удалить защищенный файл) авторы привели другой вариант сообщения об ошибке. По-видимому, это объясняется тем, что иллюстративные примеры были подготовлены с использованием различных версий ОС CP/M. – Прим. перев.

При выводе такого сообщения об ошибке курсор остается на той же строке, что и само сообщение. При нажатии клавиши RETURN или при вводе команды CTRL/C (реинициализация системы) курсор перемещается на следующую строку, и на экране снова появляется промпт ОС CP/M.

При помощи команды STAT можно в любой момент изменить статус защиты всего диска (с "R/W" на "R/O" или наоборот). Как отмечено выше, значение "R/O" используется для предотвращения случайного уничтожения файлов. Помимо этого этот статус можно назначать файлам, которые расположены на произвольном (не обязательно текущем) диске. Для этого в спецификацию файла включают имя данного диска. В пределах спецификации файла можно применять те же комбинации метасимволов "*" и "?", которые используются для получения списка имен файлов (см. выше). Это бывает полезно в случаях, когда для всех файлов или для определенной группы файлов необходимо установить одно и то же значение ключа защиты.

ОБМЕН ФАЙЛАМИ МЕЖДУ РАЗЛИЧНЫМИ ВЕРСИЯМИ ОС CP/M

При передаче файлов из ОС CP/M версии 2.0 или выше в более ранние версии статусы защиты всех файлов должны соответствовать "R/W". В противном случае при обработке файлов в более ранних версиях ОС CP/M могут возникнуть ошибки. Отметим, что в более ранних версиях ОС CP/M не существует способа, установки и изменения статуса защиты файла. При передаче файлов из ранних версий ОС CP/M в более поздние статусы защиты этих файлов по умолчанию присваивается атрибут "R/W". Поэтому при необходимости защиты файлов следует изменить этот атрибут.

ВРЕМЕННАЯ ЗАЩИТА ВСЕГО ДИСКА

STAT A:=R/O

ОС CP/M предоставляет возможность временно назначить статус защиты диску "R/O" или "R/W". Если назначается защита диска "R/O", то все содержимое этого диска будет защищено от записи независимо от того, что статусы защиты некоторых файлов "R/W". Это средство позволяет защищать диск так, чтобы нельзя было осуществить на него случайную запись. Этот статус защиты диска имеет силу только до тех пор, пока диск вставлен в устройство. Как только диск извлекается из устройства, ему назначается первоначальный статус "R/W". Как правило, для того чтобы отменить защиту диска, необходимо использовать команду CTRL/C (реинициализация системы), при обработке которой статусы защиты дисков соответствуют "R/W" ("Чтение/запись"). Кроме того, это показывает, что установленная таким образом временная защита диска является неэффективной, при вводе команды CTRL/C.

В очередном примере показано, как использовать команду STAT для временной установки для диска "A" режима "только чтение":

A>STAT A:=R/O

A>

Для того чтобы проверить, что диску назначен статус защиты "R/O", используется команда STAT в основном формате:

```
A>STAT
```

```
A:R/O, SPACE: 149K
```

При попытке записи на защищенный диск, выдается сообщение об ошибке (см. ниже). Нажатие любой клавиши в ответ на это сообщение вызывает реинициализацию системы и диск будет установлен в первоначальный режим "R/W" ("Чтение/запись"). На этом примере показано, что произойдет, если при помощи команды PIP попытаться скопировать все файлы с диска "B" на диск "A", тогда как статус защиты последнего "R/O" (подробное описание команды PIP представлено в следующей главе).

```
A>STAT
```

```
A: R/O, SPACE: 149K
```

```
A>PIP A:=B:*.*
```

```
BDOS ERROR ON A: READ ONLY_
```

<---- Была нажата клавиша
RETURN, что привело к
реинициализации системы

```
A>
```

```
A>STAT
```

```
A: R/W, SPACE: 149K
```

Для того чтобы восстановить для диска режим "Чтение/запись", можно провести реинициализацию системы (команда CTRL/C) или полную перезагрузку системы.

```
A>C
```

```
A>STAT
```

```
A: R/W, SPACE:149K
```

"СКРЫТЫЕ" ФАЙЛЫ

STAT TEST.DOC \$SYS
STAT TEST.DOC \$DIR

В ОС CP/M версии 2.0 или выше команда STAT позволяет так "упрятывать" файлы, чтобы их имена не выводились при вводе команды DIR. Для того чтобы избавиться от монотонной печати имен "системных" файлов при каждом обращении к команде DIR, используется атрибут "\$ SYS". Этими "системными" файлами могли бы быть файлы STAT.COM, PIP.COM или SYSGEN.COM, которые обычно имеются на всех дисках. Функция команды STAT может быть использована для обеспечения определенного уровня секретности (например, можно предотвратить обращение неопытного пользова-

теля к этим файлам). Для того чтобы сделать файл "скрытым", ему или группе файлов, в которую он входит, необходимо назначить атрибут " SYS".

Ниже представлен пример, в котором иллюстрируется назначение атрибута " SYS" файлу STAT.COM.

```
A>STAT STAT.COM *SYS
```

```
STAT.COM SET TO SYS
```

```
A>
```

Как показано ниже, после назначения файлу атрибута " SYS" его имя выводится по команде DIR. Однако если для получения списка файлов используется команда STAT (см. выше), то имя файла выводится на консоль. Разница заключается в том, что теперь имя файла заключено в круглые скобки, чтобы отметить тот факт, что файл является системным "скрытым" файлом.

```
A>DIR STAT.COM
```

```
FILE NOT FOUND
```

```
A>STAT STAT.COM
```

```
RECS BYTES EXT ACC  
41 6K 1 R/W A:(STAT.COM)  
BYTES REMAINING ON A: 476K
```

```
A>
```

При помощи круглых скобок скрытый файл выделяется среди остальных не скрытых файлов на этом диске. Таким способом атрибут " SYS" может быть назначен произвольному количеству файлов или всем файлам. Для спецификации файлов, которым необходимо назначить атрибут " SYS", можно использовать метасимволы "*" и "?".

Для отмены статуса "скрытый" так же, как атрибут " SYS", используется другой атрибут. Это – атрибут " DIR", и, как следует из его названия, при введении команды DIR выводятся характеристики указанных файлов. Например, полагая, что файл STAT.COM является в данное время не скрытым, покажем, что происходит при выполнении команды "STAT STAT.COM DIR".

```
A>STAT STAT.COM *DIR
```

```
STAT.COM SET TO DIR
```

```
A>
```

```
A>DIR STAT.COM
```

```
A: STAT.COM
```

```
A>
```

```
A>STAT STAT.COM
```

```
RECS BYTES EXT ACC  
41 6K 1 R/W A:STAT.COM  
BYTES REMAINING ON A: 476K
```

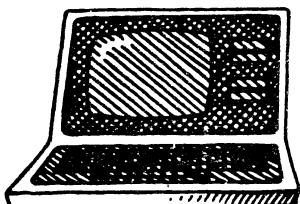
```
A>
```

ПОВТОРНАЯ ПРОВЕРКА ИМЕН ФАЙЛОВ

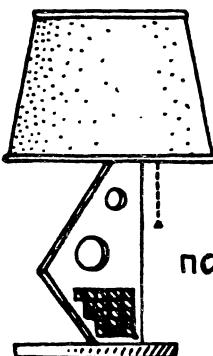
Если не удается получить список необходимых файлов с помощью команды DIR, то можно проверить эту же спецификацию при помощи команды STAT. Проверка осуществляется в случае, если файлу (файлам) назначен атрибут QSYS. Этот метод повторной проверки позволяет гарантировать существование или отсутствие файла (файлов). Помните, что при помощи атрибута QDIR можно отменить действие атрибута QSYS.

БУДУЩЕЕ ОС/М

СЕЙЧАС



ПОТОМ



МАСКИРОВКА:

ДЛЯ ТОГО ЧТОБЫ ВАШ ЛЮБИМЫЙ ПЕРСОНАЛЬНЫЙ КОМПЬЮТЕР НЕ БЫЛ УКРАДЕН В ВАШЕ ОТСУТСТВИЕ, ЕГО ПОМЕЩАЮТ В НЕПРИМЕТНОЙ С ВИДУ НАСТОЛЬНОЙ ЛАМПЕ

ОПРЕДЕЛЕНИЕ КОДОВ ПОЛЬЗОВАТЕЛЕЙ ДИСКА

STAT USR:

В ОС CP/M версии 2.0 или выше команда STAT предоставляет возможность выводить код текущего пользователя диска. В этом случае выдается тот код, под которым активный пользователь подключен к системе, и коды пользователей, имеющих файлы на этом диске. Для получения информации о том, как назначаются коды пользователей, см. описание команды USER:

A>STAT USR:

ACTIVE USER: 0
ACTIVE FILES: 0 1 2

A>

В приведенном выше примере активным пользователем (т. е. тем, чья дисковая область используется в настоящий момент) является пользователь с

кодом 0. В системе может быть максимум 16 пользователей. В строке, озаглавленной "ACTIVE FILES", перечислены коды тех пользователей, файлы которых действительно хранятся на диске. В примере файлы есть у пользователей с кодами 0, 1 и 2. (Если бы кроме области пользователя с кодом 0 на диске не было других областей пользователя, содержащих файлы, то в обеих строках был бы напечатан только код 0.) С помощью команды STAT можно получить имена только тех файлов, которые принадлежат *активному пользователю*; файлы, принадлежащие неактивным пользователям, при обработке этой команды не задействуются. В примере задействованы только файлы, принадлежащие пользователю с кодом 0. Все остальные файлы во внимание не принимаются.

ВЫВОД НА КОНСОЛЬ ХАРАКТЕРИСТИК ДИСКА

STAT DSK:

Команда STAT предоставляет возможность получить подробные сведения об активных дисках, включая емкость, количество входов и каталог и т. д. Эта информация может представлять ценность только для искушенных пользователей ОС *CP/M*, которых интересуют внутренние детали формата дискеты. Если за командой STAT указан аргумент DSK:, то выдаются сведения обо всех дисках, которые опознаны ОС *CP/M*. Помимо этого вместе с параметром может быть специфицировано имя конкретного диска, и в этом случае выдаются характеристики только этого диска. В последнем случае команда имеет вид "STAT X: DSK:", где "X:" – имя заданного диска.

В последующих пяти примерах представлена информация, которая получена для стандартных 8-дюймовых гибких дисков с программным секторированием и различной плотностью записи. В примерах 1 . . . 4 представлены характеристики односторонних дисков с одинарной (128 байт на сектор) плотностью записи, двойной плотностью при емкости в 256 байт на сектор, двойной плотностью при емкости в 1024 байт на сектор и еще одного диска с двойной плотностью записи при емкости в 256 байт на сектор соответственно. В примере 5 представлены характеристики двустороннего диска с двойной плотностью записи при емкости в 1024 байт на сектор. Примеры сопровождаются объяснениями каждой характеристики.

В примерах представлены характеристики диска с именем "A". При этом предполагается, что в данный момент остальные диски считаются в ОС *CP/M* отсутствующими.

ПРИМЕР 1

(ОДНОСТОРОННИЙ ГИБКИЙ ДИСК С ОДИНАРНОЙ ПЛОТНОСТЬЮ ЗАПИСИ ПРИ ЕМКОСТИ В 128 БАЙТ НА СЕКТОР)

A>STAT DSK:

A: DRIVE CHARACTERISTICS
1944: 128 BYTE RECORD CAPACITY
243: KILOBYTE DRIVE CAPACITY

```
64: 32 BYTE DIRECTORY ENTRIES
64: CHECKED DIRECTORY ENTRIES
128: RECORDS/ EXTENT
8: RECORDS/ BLOCK
26: SECTORS/ TRACK
2: RESERVED TRACKS
```

ПРИМЕР 2

(ОДНОСТОРОННИЙ ГИБКИЙ ДИСК С ДВОЙНОЙ ПЛОТНОСТЬЮ ЗАПИСИ ПРИ
ЕМКОСТИ В 256 БАЙТ НА СЕКТОР)

```
A>STAT DSK:
```

```
A: DRIVE CHARACTERISTICS
3856: 128 BYTE RECORD CAPACITY
482: KILOBYTE RECORD CAPACITY
128: 32 BYTE DIRECTORY ENTRIES
128: CHECKED DIRECTORY ENTRIES
128: RECORDS/ EXTENT
16: RECORDS/ BLOCK
52: SECTORS/ TRACK
2: RESERVED TRACKS
```

ПРИМЕР 3

(ОДНОСТОРОННИЙ ГИБКИЙ ДИСК С ДВОЙНОЙ ПЛОТНОСТЬЮ ЗАПИСИ ПРИ
ЕМКОСТИ В 1024 БАЙТ НА СЕКТОР)

```
A>STAT DSK:
```

```
A: DRIVE CHARACTERISTICS
4784: 128 BYTE RECORD CAPACITY
598: KILOBYTE RECORD CAPACITY
128: 32 BYTE DIRECTORY ENTRIES
128: CHECKED DIRECTORY ENTRIES
128: RECORDS/ EXTENT
16: RECORDS/ BLOCK
64: SECTORS/ TRACK
2: RESERVED TRACKS
```

ПРИМЕР 4

(ОДНОСТОРОННИЙ ГИБКИЙ ДИСК С ДВОЙНОЙ ПЛОТНОСТЬЮ ЗАПИСИ ПРИ
ЕМКОСТИ В 256 БАЙТ НА СЕКТОР)

```
A>STAT DSK:
```

```
A: DRIVE CHARACTERISTICS
3888: 128 BYTE RECORD CAPACITY
486: KILOBYTE RECORD CAPACITY
128: 32 BYTE DIRECTORY ENTRIES
128: CHECKED DIRECTORY ENTRIES
128: RECORDS/ EXTENT
16: RECORDS/ BLOCK
52: SECTORS/ TRACK
2: RESERVED TRACKS
```

ПРИМЕР 5

(ДВУСТОРОННИЙ ГИБКИЙ ДИСК С ДВОЙНОЙ ПЛОТНОСТЬЮ ЗАПИСИ ПРИ
ЕМКОСТИ В 1024 БАЙТ НА СЕКТОР)

A>STAT DISK:

A:	DRIVE CHARACTERISTICS
9600:	128 BYTE RECORD CAPACITY
1200:	KILOBYTE RECORD CAPACITY
256:	32 BYTE DIRECTORY ENTRIES
256:	CHECKED DIRECTORY ENTRIES
128:	RECORDS/ EXTENT
16:	RECORDS/ BLOCK
64:	SECTORS/ TRACK
4:	RESERVED TRACKS

Диски с одинарной плотностью записи и емкостью в 128 байт на сектор (см. пример 1) применяются во всех версиях ОС *CP/M* (с 1.3 по 2.2). Диски с двойной плотностью записи и емкостью в 256 или 1024 байт на сектор (см. примеры 2 и 3 соответственно) применяются только в ОС *CP/M* версии 2.2 или выше. Характеристики дисков другого типа, например двусторонних или жестких дисков, отличаются от характеристик стандартных односторонних гибких дисков. В примерах 4 и 5 представлены характеристики дисков для другой, отличной от примеров с 1 по 3, ЭВМ, отличающейся от той, характеристики дисков которой были представлены в примерах с 1 по 3. В примере 4 представлены характеристики одностороннего диска с двойной плотностью записи и емкостью в 256 байт на сектор, а в примере 5 представлены характеристики двустороннего диска с двойной плотностью записи и емкостью в 1024 байт на сектор. Ниже объясняется значение каждого параметра.

128 BYTE RECORD CAPACITY. Данный параметр обозначает максимальное число 128-байтных записей, которое может храниться на диске. Запись является единицей измерения объема данных. Она используется для группировки данных файла, для повышения эффективности обработки файлов. Одна запись является целой частью килобайта или одного килобайта. Поскольку 1 килобайт равен 1024 байт (а не 1000, как можно было бы предположить), то его можно разделить на 128 без остатка, т. е.

$$1024 : 128 = 8 \text{ записей.}$$

Из приведенных примеров следует, что значение данного параметра увеличивается при переходе от одинарной плотности записи к двойной, а при двойной плотности при переходе от емкости в 256 байт на сектор к емкости в 1024 байт на сектор.

KILOBYTE DRIVE CAPACITY. Данный параметр обозначает максимальную емкость диска в килобайтах. Как правило, этот параметр является наиболее важным. Он обозначает общий объем свободного пространства на пустом диске. Из предыдущих примеров следует, что емкость диска в килобайтах значительно увеличивается при переходе от одинарной плотности записи к двойной и при переходе от емкости в 256 байт на сектор к емкости в 1024 байт на сектор.

32 BYTE DIRECTORY ENTRIES. Данный параметр обозначает максимальное количество файлов, которые можно хранить на диске. Вход в каталог имеет фиксированную длину, равную 32 байт. На дисках с одинарной плотностью

записи (см. пример 1) обычно имеется 64 одинаковых входа в каталог, соответствующих 64 файлам. Для дисков с двойной плотностью записи, обладающих большей емкостью, данное значение увеличивается до 128.

Если какие-либо файлы занимают более одного "экстента" (т. е. занимают более 16К), то максимальное число файлов на диске уменьшится. Таким образом, с увеличением размеров файлов (более 16К) уменьшается максимальное число различных файлов, которое можно хранить на диске. Например, если на диск записать десять различных файлов объемом по 48К каждый, то произойдет следующее. Каждый файл займет три экстента (48 деленное на 16 равняется 3), следовательно, на диске будет занято $10 \times 3 = 30$ входов в каталог, а свободных останется только 34 ($64 - 30 = 34$).

CHECKED DIRECTORY ENTRIES. Если в качестве носителя используется гибкий диск, то значение данного параметра, как правило, совпадает со значением параметра, обозначающего количество входов в каталог. Для накопителей на жестких дисках с несъемными носителями данный параметр обычно равен нулю. Его значение печатается в отчете потому, что ОС CP/M обязана отслеживать смену носителя (т. е. смену гибких дисков без перезагрузки системы). Для несъемных жестких дисков данный параметр по умолчанию равен 0, поскольку такие диски никогда не снимаются.

RECORDS/EXTENT. Данный параметр определяет максимальное количество записей на один экстент, которому соответствует один вход в каталог. (Экстент занимает 16К (128 записей \times 128 байт/запись = 16 382)). Отметим, что один файл может занимать несколько экстентов. В приведенных примерах на один экстент приходится максимум 128 записей.

RECORDS/BLOCK. Данный параметр обозначает *минимальный* объем дисковой памяти, который можно предоставить для одного файла. Файлы размещаются в так называемых блоках. Каждый такой "блок" может быть длиной 1К или 2К. Для определения реального размера файла необходимо использовать количество записей файла (число в столбце "RECS" при выполнении команды STAT). Отметим, что в примере 1 один блок содержит 8 записей (8 записей \times 128 байт = 1024 байт, или 1К). В примерах со 2 по 5 один блок содержит 16 записей (16 записей \times 128 байт = 2048 байт, или 2К). Размер блока является стандартным для всего диска. Если содержимое файла занимает менее 1К при одинарной плотности записи или менее 2К при двойной плотности, то оставшееся пространство в блоке остается неиспользованным, хотя и закрепляется за данным файлом. Этим объясняется, например, почему файлы, хранящиеся на дисках с одинарной плотностью записи и занимающие менее 1К, при обработке команды STAT рассматриваются как файлы, занимающие 1К дисковой памяти.

SECTORS/TRACK. Данный параметр обозначает количество секторов на каждой дорожке диска, а это, в свою очередь, зависит от того, как форматирован диск. На дисках с одинарной плотностью записи одна дорожка содержит 26 секторов. На дисках с двойной плотностью записи и емкостью в 256 байт на сектор одна дорожка содержит 52 сектора. На дисках с двойной плотностью записи и емкостью в 1024 байт на сектор одна дорожка содержит 64 сектора. Можно увидеть, что формат с емкостью в 1024 байт/сектор увеличивает емкость диска.

RESERVED TRACKS. Этот параметр указывает количество дорожек диска

(обычно дорожки с номерами 0 и 1), которые не желательно использовать для хранения данных. Эта область резервируется и используется для хранения операционной системы *CP/M* (секций BIOS, BDOS и CCP).

ВЫВОД НА КОНСОЛЬ СПИСКА НАЗНАЧЕНИЙ УСТРОЙСТВ

STAT DEV:

Операционная система *CP/M* предоставляет пользователю возможность назначать имена физических (реальных) устройств именам логических (условных) устройств. Имена логических устройств являются условными в том смысле, что каждому из них может быть назначено имя реального физического устройства. Это означает, что каждое логическое устройство может представлять собой любое из четырех *реальных* устройств. В ОС *CP/M* существует четыре основных логических устройства: консоль, устройство ввода, устройство вывода и устройство печати. Каждому из них может быть назначено одно из четырех физических устройств. Физическими устройствами могут быть устройства печати, видеотerminalы, клавиатуры, модемы, графопостроители, самописцы, другие порты ввода или вывода ЭВМ и т. д.

При помощи команды STAT можно вывести на консоль список текущих назначений. Для этого следует ввести с консоли командную строку "STAT DEV:"

STAT DEV:

CON: IS CRT:
RDR: IS PTR:
PUN: IS PTP:
LST: IS LPT:

А)

Имена логических (символических) устройств напечатаны в левом столбце. Имена реальных физических устройств, которые назначены этим логическим устройствам, напечатаны в правом столбце. Обозначение логических устройств следующее:

CON: – Консоль. Это имя обычно обозначает любое физическое устройство, использованное как для ввода, так и для вывода данных. Консоль является логическим устройством двойного назначения, т. е. может осуществлять ввод и вывод данных. Это устройство обычно используется для установления связи между секцией CCP (процессором команд консоли) и терминалом оператора (дисплеем или телетайпом). Обычно логическое устройство CON: – это видеотерминал или телетайп, однако это также может быть встроенный дисплей или изолированная клавиатура. Имя этого устройства может быть использовано в качестве параметра команды PIP. Например, по команде "A > PIP CON: = B: file.typ" на консоль будет выдано содержимое указанного файла.

RDR: – Устройство ввода данных /Устройство чтения с перфоленты. Это имя используется для обозначения любого физического устройства, используемого только для ввода данных. Типичным физическим устройством явилось бы

устройство чтения с перфоленты. Им может быть также порт ввода с другой клавиатуры, оцифровывающее устройство, устройство RS-232-C, используемое для связи с другой ЭВМ в режиме "только чтение" (для передачи файлов вперед и назад) и т. д.

PUN: – Устройство вывода данных / Устройство вывода на перфоленту. Имя этого логического устройства применяется для обозначения любого физического устройства, используемого только для вывода данных. Типично таким явилось бы устройство вывода на перфоленту. Им также мог бы быть графопостроитель, цифровой самописец, печатающее устройство, вывод на другую ЭВМ и т. д.

LST: – Устройство печати. Имя этого логического устройства используется для обозначения любых печатающих устройств.

ВЫВОД ВСЕХ ВОЗМОЖНЫХ ИМЕН УСТРОЙСТВ И ДРУГИХ
ВОЗМОЖНЫХ ПАРАМЕТРОВ КОМАНДЫ STAT

STAT VAL:

В ОС *CP/M* версии 1.3 и выше команду STAT можно применять для получения списка допустимых имен всех физических устройств, которые могут быть обозначены четырьмя логическими именами. В ОС *CP/M* версии 2.0 и выше данный список включает в себя информацию обо всех допустимых параметрах команды STAT, описанных ранее в данной главе. В приводимом ниже примере представлен отчет команды STAT в ОС *CP/M* версии 2.0 или выше.

A>STAT VAL:

```
TEMP R/O DISK: D:=R/O
SET INDICATOR: D:FILENAME.TYP >R/O >R/W >SYS >DIR
DISK STATUS : DSK: D:DSK:
USER STATUS : USR:
IOBYTE ASSIGN:
CON: = TTY: CRT: BAT: UC1:
RDR: = TTY: PTR: UR1: UR2:
PUN: = TTY: PTP: UP1: UP2:
LST: = TTY: CRT: LPT: UL1:
```

A>

Первая часть отчета не нуждается в подробных пояснениях, поскольку здесь перечисляются те функции команды STAT, которые уже были описаны в этой главе. В первой строке дается запись, которая назначает диску статус "R/O" ("Только чтение"). Символ "D" указывает имя диска. Во второй строке перечислены параметры назначения файлам различных атрибутов. Эти четыре напечатанных параметра включают ">R/O" ("Только чтение") ">R/W" ("Чтение/запись"), ">SYS" (системный (скрытый) файл) и ">DIR" (каталог (отменяет действие параметра ">SYS")). В третьей строке дается запись, необходимая для вывода характеристик диска ("DSK:" для всех дисков, а "D:DSK:" для диска с именем "D"). В четвертой строке представлена запись для вывода на консоль кода текущего пользователя и номера тех областей пользователя, которые содержат какие-либо файлы.

Во второй части отчета команды "STAT VAL:", которая озаглавлена "IOBYTE ASSIGN", перечисляются имена всех физических устройств, которые при вводе-выводе могут быть назначены каждому логическому устройству. Байт ввода-вывода (IOBYTE) является параметром, который хранится в операционной системе и используется для управления назначением имен физических устройств логическим. При первичной загрузке ОС *CP/M* или при реинициализации системы этому параметру по умолчанию присваиваются определенные значения, которые по команде STAT отменяются.

Понятно, что физических устройств существует великое множество, но обычно очень немногие обеспечены современной аппаратурой и программами. Часто доступными являются только устройства TTY:, CRT: и LPT:. Изменить этот набор имен этих физических устройств может только поставщик или системный программист конкретной версии ОС *CP/M*. Имена этих физических устройств поясняются ниже.

TTY: – Низкоскоростная консоль (телетайп). Это физическое устройство обычно представляет собой терминал с низкой скоростью передачи данных, такой как телетайп или другой терминал, работающий со скоростью передачи 300 бит/с (bps) или меньше. Имя физического устройства TTY: может быть назначено любому логическому устройству, хотя обычно оно назначается логическому устройству CON: (консоль), если оно используется в качестве операторской консоли.

CRT: – Высокоскоростная консоль (устройство с катодной лучевой трубкой). Это устройство представляет собой высокоскоростное консольное устройство, которое может быть назначено только логическому устройству CON: (консоль). Видеотерминалы с катодной лучевой трубкой обычно работают со скоростью более высокой, чем телетайпы (часто до 9600 бит/с (bps) или выше).

BAT: – Устройство пакетной обработки. Устройства пакетной обработки можно назначать только логическому устройству CON: (консоль). Ввод данных в ОС *CP/M* осуществляется через логическое устройство RDR: (устройство ввода данных). Этому логическому устройству может быть, например, назначено физическое устройство TTY:..

Вывод данных в ОС *CP/M* осуществляется на логическое устройство LST (устройство печати), которому может быть назначено физическое устройство TTY:, CRT: или LPT: (печатывающее устройство).

UC1: – Консоль, определенная пользователем. Когда в качестве консольного устройства используется специальное оборудование, имя этого физического устройства назначается логическому устройству CON. Обычно для подключения таких специальных устройств в ОС *CP/M* необходимо модифицировать секцию BIOS. В результате данное назначение информирует ОС *CP/M* о том, что в качестве консоли используется нестандартное устройство и что следует использовать модифицированную секцию BIOS.

PTR: – Устройство чтения с перфоленты. Имя этого физического устройства назначается логическому устройству RDR: (высокоскоростное устройство ввода) в тех случаях, когда к системе подключено оборудование, обеспечивающее ввод данных с перфоленты.

UR1: – Определенное пользователем устройство ввода номер 1. Имя этого устройства используется в тех случаях, когда для высокоскоростного ввода

данных системой CP/M подключается специальное оборудование. Физическое устройство с именем UR1 назначается логическому устройству RDR: (высокоскоростное устройство ввода). Как и для всех физических устройств, для подключения такого устройства необходимо модифицировать секцию BIOS.

UR2: – Определенное пользователем устройство ввода номер 2. Имя этого устройства используется таким же образом, как UR1, за исключением того, что оно обычно обозначает различное нестандартное оборудование. Часто этим именем определяют оборудование, требующее различных модификаций секции BIOS операционной системы CP/M.

PTP: – Устройство вывода на перфоленту. Имя этого физического устройства назначается логическому устройству PUN: (высокоскоростное устройство вывода) в тех случаях, когда к системе подключено оборудование, обеспечивающее вывод данных на перфоленту.

UP1: – Определенное пользователем устройство вывода номер 1. Имя этого устройства используется в тех случаях, когда для высокоскоростного вывода данных системой CP/M подключается специальное оборудование. Это устройство назначается логическому устройству PUN: (высокоскоростное устройство вывода).

UP2: – Определенное пользователем устройство вывода номер 2. Имя этого устройства используется так же, как имя UP1, за исключением того, что оно обычно обозначает различное специальное оборудование, для подключения которого часто требуется внесение различных изменений в секцию BIOS.

LPT: – Построчное печатающее устройство. Имя этого физического устройства назначается логическому устройству LST: (устройство печати), когда к системе подключено печатающее устройство.

UL1: – Определенное пользователем устройство печати номер 1. Имя этого устройства используется для обозначения нестандартного оборудования вывода данных. Оно включает в себя устройства печати, отличающиеся от устройства LPT: такие, например, как графопостроители. Устройство UL1: назначается логическому устройству LST: (устройство печати). Эти виды специального оборудования обычно требуют внесения изменений в секцию BIOS. При назначении физическому устройству имени UL1, а не LPT, ОС CP/M будет указано на необходимость использовать для обмена с этим устройством специальную программу в секции BIOS.

НАЗНАЧЕНИЕ ЛОГИЧЕСКИХ ИМЕН ФИЗИЧЕСКИМ УСТРОЙСТВАМ

STAT log:=phy:

Помимо печати списка назначений физических устройств логическим устройствам команда STAT может быть использована для того, чтобы выполнить эти назначения. Для этого в качестве параметра необходимо указать имя логического устройства (CON:, RDR:, PUN: или LST:), потом знак равенства (=) и имя физического устройства (TTY:, CRT:, LPT: и т. д.). В представленном ниже примере иллюстрируется выполнение действий: вывод на консоль списка текущих назначений физических устройств логическим, переназначение одного из устройств и затем повторный вывод списка назначений:

```
A>STAT DEV:  
CON: IS TTY:  
RDR: IS PTR:  
PUN: IS PTP:  
LST: IS LPT:  
  
A>STAT CON:=CRT:  
A>STAT DEV:  
  
CON: IS CRT:  
RDR: IS PTR:  
PUN: IS PTP:  
LST: IS LPT:  
  
A>
```

Если имя устройства задано с грамматическими ошибками или если заданное назначение недопустимо, на консоли появится следующее сообщение:

```
A>STAT CON:=LPT:  
INVALID ASSIGNMENT  
A>
```

В тех случаях, когда необходимо выполнить несколько назначений, их можно сгруппировать в одной командной строке. Каждое имя физического устройства, назначенное логическому, отделяется одно от другого запятой, а при окончании ввода командной строки необходимо нажать клавишу "RETURN". Ниже представлен пример, иллюстрирующий одновременное назначение физических устройств всем логическим устройствам:

```
A>STAT CON:=CRT:,RDR:=UR1:,PUN:=UP1:,LST:=UL1:  
A>
```

ПРЕДОСТЕРЕЖЕНИЕ

При назначении устройств необходимо соблюдать осторожность. Не назначайте физические устройства, для которых нет программной поддержки. Например, особые затруднения можно испытать с устройствами, определенными пользователем. Если данные устройства не описаны в секции BIOS операционной системы *CP/M*, то использование имен этих устройств в командных строках приведет к "зависанию" системы. В этом случае придется полностью перезагрузить систему.

Версии ОС *CP/M*:

1.3 1.4 2.0 2.2

Тип команды:

транзитная



ОБЩИЕ СВЕДЕНИЯ

Название PIP является сокращением от английского Peripheral Interchange Program (программа обмена с периферийными устройствами). Команда PIP обеспечивает обращение к транзитной программе ОС *CP/M*, которая также называется "PIP" и предоставляет пользователю возможность осуществлять обмен данными между различными периферийными устройствами, например копировать файлы с диска на печатающее устройство и т. д. Эта программа позволяет, кроме того, копировать файл на тот же диск под другим именем. Слова "обмен с периферийными устройствами" в названии этой программы означают, что она обеспечивает обмен данными между всеми типами устройств, а не только между дисками. При помощи команды PIP можно копировать файлы *всех* типов, включая исполняемые ".COM" файлы, текстовые файлы и т. д. Кроме того, команда PIP может быть использована для копирования групп файлов или отдельных файлов между различными комбинациями устройств. (Безусловно, данные устройства должны быть описаны как в утилите "PIP", так и в секции BIOS операционной системы *CP/M*.)

Ниже кратко описываются основные возможности команды "PIP".

Создание копии файла на том же диске. Команда PIP может быть использована для создания копии файла, расположенного на диске. При этом копия данного файла может быть размещена на том же диске в другом файле. Имя файла, содержащего копию, может быть произвольным, по желанию пользователя.

Копирование одного файла или группы файлов на другие диски. Данная команда может быть использована для копирования на другие диски любого файла, или почти любой заданной группы файлов. Файлы, содержащие данные копии, могут иметь те же самые имена, что и исходные файлы или другие. **Объединение нескольких файлов в один файл.** При помощи команды PIP можно скопировать группу файлов и объединить их копии в одном файле. Результирующий файл может быть расположен как и на том же самом диске, так и на других.

Обмен данными между дисками и специальными устройствами. При помощи специальных параметров команды PIP пользователь может осуществлять обмен между диском и специальным периферийным устройством, таким как логическое устройство построчной печати, модем и т. д. Команда PIP позволяет осуществлять копирование данных в обоих направлениях, т. е. с диска на специальное устройство или со специального устройства в файл на диске. **Передача данных между стандартными устройствами ОС *CP/M* и специальными устройствами утилиты "PIP".** С помощью команды PIP можно осущес-

ствлять передачу данных между большинством стандартных (логических и физических) устройств ОС CP/M, а также некоторыми специальными устройствами, определенными в утилите "PIP". Команду PIP можно, например, использовать для копирования файлов между двумя ЭВМ с различными дисковых форматами.

Дополнительные параметры команды PIP. Почти в любой форме записи команды PIP пользователь может указать один или несколько дополнительных параметров. Эти параметры указывают на необходимость выполнить в процессе копирования определенные вспомогательные действия и добавляются к обычным параметрам команды PIP в качестве атрибутов или "ключей". Ниже представлен список возможных вспомогательных действий утилиты "PIP":

- Отображение текстов на консоли по мере их копирования.
- Проверка каждой операции копирования путем сравнения исходного файла с файлом, содержащим копию.
- Начало копирования после того, как обнаружена заданная строка символов, и/или завершение копирования после того, как обнаружена заданная строка символов.
- Преобразование в процессе копирования символов верхнего регистра в символы нижнего регистра (или наоборот).
- Удаление в каждой строке текста всех символов, расположенных после указанного столбца.
- Вставка символа табуляции в каждой указанной позиции.
- Удаление в процессе копирования всех символов перевода формата (form-feed) или вставка символов перевода формата каждый раз после указанного числа строк.
- Добавление в процессе копирования к каждой строке ее порядкового номера.
- Установление блочного режима передачи данных.
- Копирование "системных" файлов.
- Удаление файла со статусом защиты "R/O" ("Только чтение") в том случае, если он имеет то же имя, что и выходной файл (только в ОС CP/M версии 2.0 и выше).
- Копирование файлов из областей других пользователей.
- Копирование файлов с машинным/объектным кодом; маркеры конца файла при этом игнорируются.
- Проверка формата шестнадцатеричного файла. В результате могут быть удалены все нулевые ("00") записи копируемого файла.
- Установление бита четности каждого копируемого байта в ноль.

ОБРАЩЕНИЕ И ИСПОЛЬЗОВАНИЕ УТИЛИТЫ "PIP"

Для обращения к утилите "PIP" необходимо, чтобы файл PIP СОМ хранился на одном из вставленных в устройства диске. Существует два основных способа обращения и использования утилиты "PIP". А именно:

1. В тех случаях, когда желательно с помощью утилиты "PIP" выполнить ряд последовательных операций, обратиться к ней можно, набрав команду PIP в ответ на промпт с именем текущего диска (A >, B > и т. д.). При этом утилита "PIP" будет загружена в оперативную память, а

на консоли появится новый промпт, который называется "PIP-промпт" и представляет собой символ "*".

2. Кроме того, команду PIP можно использовать в тех случаях, когда необходимо выполнить только одну операцию по обмену между устройствами и желательно, чтобы сразу после ее завершения на консоли появился основной промпт ОС CP/M (A>, B> и т. д.). Для этого после команды PIP необходимо ввести командную строку, описывающую операцию, которую требуется выполнить. В конце ввода командной строки необходимо нажать клавишу "RETURN".

Различные способы использования команды PIP описываются в последующих параграфах. Для всех комбинаций параметров команды PIP дано подробное описание допустимого синтаксиса и формата командной строки.

КОПИРОВАНИЕ ФАЙЛОВ С ДИСКА НА ДИСК

ЗАГРУЗКА УТИЛИТЫ "PIP" И ВЫПОЛНЕНИЕ
ОПЕРАЦИЙ КОПИРОВАНИЯ

PIP

При первом использовании утилиты "PIP" вам целесообразно для обращения к ней использовать простейший формат. Это осуществляется путем загрузки данной программы в оперативную память без спецификации каких бы то ни было параметров. Если в качестве ответа на промпт с именем текущего диска будет введена команда PIP, то утилита будет загружена в оперативную память, и на консоли появится промпт "*". В качестве ответа на этот промпт можно вводить спецификацию любых операций утилиты "PIP". Это действие проиллюстрировано в представленном ниже примере:

A>PIP
*

Как только на консоли появится промпт "*", это означает, что утилита "PIP" готова к обработке очередной командной строки.

Основным форматом командной строки является следующий:

* d : filespec = s : filespec1, filespec2, . . . , и т. д. [option flags],

где * — PIP-промпт, d — выходное устройство (куда), s — входное устройство (откуда), filespec — необязательные имя файла и расширение (по крайней мере требуется одно имя файла), option flags — один или несколько однобуквенных параметров.

Поскольку утилита "PIP" уже загружена в оперативную память, при вводе данных командных строк нет необходимости задавать саму команду PIP. В следующем ниже примере представлена командная строка для копирования одного файла с диска "A" на диск "B". После того, как командная строка полностью введена, нажмите клавишу "RETURN". После выполнения указанной операции на следующей строке снова появится промпт "*":

```
A>PIP  
*B:=A:LETTER1.TXT  
*
```

Отметим, что в этом примере имена дисков введены в особом порядке. Командная строка гласит: Я хочу, чтобы файл LETTER1.TXT, который в настоящее время хранится на диске "А", хранился бы также и на диске "В", а не Скопировать файл LETTER1.TXT с диска "А" на диск "В". При обращении к команде PIP имена дисков всегда перечисляются именно в такой последовательности. Кроме того, подобная последовательность используется в большинстве аналогичных случаев при обращении к остальным командам ОС CP/M (например, к встроенной команде REN). В предыдущем примере устройство (диск) "А" является входным, а устройство (диск) "В" является "выходным". Таким образом, говоря об указанной выше командной строке, можно перефразировать ее следующим образом: "Выходное устройство (диск) "В" установить равным входному файлу на устройстве (диске) "А". В командных строках утилиты "PIP" выходное устройство всегда указывается слева, а входное устройство – справа.

Существует два других способа скопировать файл с диска "А" на диск "В". Однако запомните, что синтаксис или последовательность записи командной строки "имя выходного устройства, знак равенства, имя входного устройства" ("B: = A:" в предыдущем примере) остаются неизменными. Они не меняются во всех остальных случаях использования команды PIP. В следующем примере показано два способа копирования файла "LETTER1.TXT" с диска "А" на диск "В":

```
A>PIP  
*B:LETTER1.TXT=A:LETTER1.TXT  
*<(CR)  
A>
```

или

```
A>PIP  
*B:LETTER1.TXT=A:  
*<(CR)  
A>
```

Из трех способов спецификации командной строки для копирования файла наиболее предпочтительным и используемым является способ, представленный в первом примере (т. е. имя выходного устройства (диска), знак равенства, имя входного устройства (диска), имя файла на входном устройстве (диске)).

Как только операция копирования заданного файла закончится, на консоли снова появится промпт утилиты "PIP" – "*". С этого момента утилита "PIP" готова к обработке другой командной строки. В этом режиме можно специфицировать столько операций, сколько потребуется, для чего необходимо в качестве ответа на промпт утилиты "PIP" повторно вводить соответствующие командные строки. Можно любой момент вернуть промпт ОС CP/M, для чего в ответ на промпт утилиты "PIP" необходимо нажать клавишу RETURN.

В следующем примере иллюстрируется копирование нескольких различных файлов с диска "А" на диск "В", наконец возврат промпта ОС CP/M:

```
A>PIP  
*B:=A:LETTER1.TXT  
*B:=A:LETTER2.TXT  
*B:=A:STAT.COM  
*B:=A:ACCOUNTS.BAS  
*B:=A:MAPS.FOR  
*B:=A:NUMBERS.PAS  
*(CR)
```

А>



При копировании группы файлов для ее указания могут быть использованы метасимволы "*" и "?". В следующем примере показано, как утилита "PIP" будет реагировать в этом случае:

```
A>PIP  
*B:=A:LETTER?.*  
  
COPYING -  
LETTER1.TXT  
LETTER2.TXT  
  
*B:=A:*.BAS  
  
COPYING -  
ACCOUNTS.BAS  
PLOTS.BAS  
  
*(CR)
```

А>

При спецификации файлов для команды PIP можно использовать любые комбинации метасимволов и обычных символов. Более подробную информацию об использовании метасимволов см. в гл. 2.

Кроме того, утилита "PIP" допускает, чтобы имя файла-копии отличалось от имени файла-оригинала. Это позволит переименовывать не файлы-оригиналы, а их копии. В следующем примере показано, как файл-оригинал с именем "ACCOUNTS.TXT", расположенный на диске "A", можно скопировать на диск "B" в файл с именем "ACCOUNTS.DOC":

```
A>PIP  
*B:ACCOUNTS.DOC=A:ACCOUNTS.TXT  
*(CR)  
  
A>
```

При копировании файла на тот же самый диск, на котором хранится оригинал, нужно назначить файлу-копии другое имя. Это необходимо, поскольку в ОС *CP/M* наличие на одном диске двух файлов с одинаковым именем не допускается. В следующем примере показано, как копируется файл с именем "LETTER1.TXT", хранящийся на диске "A", и таким образом создается файл с именем "LETTER1B.TXT" на том же самом диске:

```
A>PIP  
*A:LETTER1B.TXT=A:LETTER1.TXT  
*(CR)  
  
A>DIR LETTER*.  
  
A : LETTER1 TXT : LETTER1B TXT  
  
A>
```

В процессе копирования при спецификации нового имени для выходного файла нельзя использовать метасимволы, поскольку в ОС *CP/M* имя файла при его создании должно быть известно точно. В команде PIP метасимволы можно использовать только в именах входных файлов и только тогда, когда имя выходного файла (файлов) не указывается. Если при спецификации имен выходных файлов для команды PIP будут использованы метасимволы, то возникнет ошибка. Это продемонстрировано в следующем примере:

```
A>PIP  
*B:SECOND.*=A:FIRST.*  
  
IMPROPER FORMAT: B:SECOND.*  
  
*(CR)  
  
A>
```

Сообщение об ошибке, представленное в данном примере, является типичным для большинства ошибок при спецификации параметров команды PIP. Обычно в сообщении указывается та часть командной строки, в которой была обнаружена ошибка (в данном случае — "B: SECOND.*").

Сообщение "IMPROPER FORMAT" выдается на консоль всегда при обнаружении метасимволов в спецификации имени выходного файла. Метасимволы могут использоваться только при спецификации имен входных файлов.

Копирование несуществующих файлов. Если в качестве входного файла для команды PIP специфицирован несуществующий файл, то на консоль будет выдано сообщение об ошибке. Это продемонстрировано в следующем примере:

```
A>PIP  
*B:=A:ZEROTEXT.DOC  
  
NO FILE: =A:ZEROTEXT.DOC  
  
*<CR>  
  
A>DIR ZEROTEXT.DOC  
  
NO FILE  
  
A>
```

Как видно из предыдущего примера, отсутствие данного файла на указанном диске подтверждено при помощи команды DIR. Несмотря на то что входной файл не был скопирован, утилита "PIP" все же создает на выходном диске рабочий файл. Если вывести оглавление данного диска, то на консоль будет выдано имя и этого "незавершенного" файла. Имя этого файла будет иметь расширение "XXX", которое означает, что данный файл был создан, но его обработка не завершилась должным образом*.

Для вывода оглавления диска можно воспользоваться транзитной командой STAT. В отчете команды STAT будет указано, что файл "ZEROTEXT.XXX" занимает ноль килобайт (OK). Для того чтобы удалить этот файл, используйте команду ERA.

Копирование данных в существующие файлы. В ОС CP/M версии 2.0 и более поздних при использовании команды PIP часто возникают конфликтные ситуации, связанные со статусом защиты файла "R/O" ("Только чтение"). (Статус защиты "R/O" рассматривался при описании транзитной команды STAT). Если файл, специфицированный в командной строке команды PIP, в качестве выходного уже существует на диске, то утилита "PIP" перепишет в него новую копию входного файла только в том случае, если файл не является защищенным, т. е. его статус защиты — "R/W" ("Чтение/запись"). (В ОС CP/M версий 1.3 и 1.4 файлам всегда назначается статус защиты "R/W".) В ОС CP/M версии 2.0 и более поздних утилита "PIP" сообщает о том, что выходной файл уже существует и имеет статус защиты "R/O" и запросит, переписывать ли в данный файл копию входного файла или же завершить процесс копирования. При ответе "Y" процесс копирования будет возобновлен и первоначальное содержимое выходного файла будет удалено. При ответе "N" на консоль будет выдано сообщение "NOT DELETED" и процесс копирования завершится. Для того чтобы определить статус защиты файла ("R/W" или "R/O") используйте команду STAT. В следующем примере демонстрируется, как будет реагировать утилита "PIP" в данной ситуации:

* Причина создания файла с расширением "XXX" заключается в том, что утилита "PIP" пытается полностью завершить копирование файлов до того, как будет удалено прежнее содержимое выходного файла. Только *после того*, как копирование закончится успешно, утилита "PIP" удалит старую версию выходного файла и переименует вновь созданный файл.

```
A>PIP  
*B:=A:SAMPLE.TXT
```

DESTINATION IS R/O, DELETE (Y/N)?N

NOT DELETED

```
*B:=A:SAMPLE.TXT
```

DESTINATION IS R/O, DELETE (Y/N)?Y

*<(CR)>

A)

В первой части данного примера в ответ на вопрос утилиты "PIP" был введен символ "N". На консоль было выдано сообщение "NOT DELETED", и операция копирования была завершена. Во второй части данного примера в качестве ответа введен символ "Y", который указал утилите "PIP" на необходимость продолжить процесс копирования и удалить содержимое выходного файла.

Копирование защищенных файлов (только в ОС CP/M версии 2.0 и выше). Если в командной строке в качестве входного специфицирован файл, статус защиты которого "R/O", то это состояние копируется только при определенных условиях. Например, при копировании отдельного файла выходному файлу будет в любом случае присвоен статус защиты "R/W", даже если статус защиты входного файла был "R/O". Однако при копировании группы файлов (с использованием метасимволов) статус защиты файлов сохраняется.

Копирование системных файлов (только в ОС CP/M версии 2.0 и выше). Если в командной строке в качестве входного специфицирован файл, которому назначен атрибут "SYS" ("системный"), то на консоль выдается сообщение "NO FILE". Это как раз тот самый случай, когда утилита "PIP" считает файл несуществующим. Для того чтобы обойти эту ситуацию, следует с помощью команды STAT присвоить данному файлу атрибут "DIR" ("обычный файл"). После этого файл можно копировать обычным образом. С другой стороны, при помощи специального параметра можно дать утилите "PIP" указание игнорировать атрибут "SYS" и выполнить копирование файла. Для этого можно воспользоваться специальным параметром команды PIP. Этот специальный параметр описывается в этой главе позже.

СПЕЦИФИКАЦИЯ ОПЕРАЦИЙ УТИЛИТЫ "PIP" В ОДНОЙ КОМАНДНОЙ СТРОКЕ

PIP B:=A:SAMPLE.TXT

Существует второй способ спецификации операций утилиты "PIP". Он отличается от описанного ранее тем, что утилита "PIP" временно загружается в оперативную память, и тем, что за одно обращение может быть выполнена только одна операция. Второй способ позволяет ввести командную строку в качестве ответа на промпт с именем текущего диска (A>, B> и т. д.) и затем сразу после завершения обработки команды вернуть на консоль тот же

самый промпт. (Это напоминает способ функционирования других стандартных программ ОС CP/M.) Так можно специфицировать все операции утилиты "PIP" в командном режиме, т. е. когда текущим является промпт "*". Для этого необходимо ввести в качестве ответа на промпт с именем текущего диска команду PIP и затем через пробел обычную командную строку утилиты "PIP". Здесь при записи командной строки действуют те же синтаксические правила, что и в остальных случаях.

В следующем примере показано, как копируется файл с диска "A" на диск "B" и как промпт с именем текущего диска после завершения операции копирования возвращается на консоль:

```
A>PIP B:=A:sample.txt
```

```
A>
```

В следующем примере представлен отчет утилиты "PIP", когда в одной командной строке специфицировано копирование группы файлов с диска "A" на диск "B". После того как все файлы скопированы, на консоли снова появляется промпт с именем текущего диска:

```
A>PIP B:=A:*.txt
```

```
COPYING -  
SAMPLE.TXT  
ACCOUNTS.TXT  
LETTER1.TXT  
LETTER2.TXT  
MESSAGE.TXT
```

```
A>
```

КОПИРОВАНИЕ НЕСКОЛЬКИХ ФАЙЛОВ В ОДИН

PIP B ALL TXT =
A 1.TXT,2 TXT

Утилита "PIP" предоставляет средства копирования нескольких файлов в один. Данный процесс называется объединением, поскольку копируемые файлы объединяются или группируются в том порядке, в котором их имена были перечислены в командной строке. После завершения ввода имени последнего файла на консоль будет выведен текст первого исходного файла, затем текст второго и т. д.

Синтаксис для этой командной строки таков: следует указать имя файла, в который производится копирование, затем спецификации файлов, которые требуется скопировать в порядке их объединения. Имя каждого исходного файла отделяется от других запятой. В представленном ниже примере иллюстрируется следующая последовательность действий: вывод на консоль и проверка содержимого файлов "1.TXT" и "2.TXT", хранящихся на диске "A", копирование и объединение их копий в файле "ALL.TXT", расположенным на диске "B":

A>TYPE 1.TXT

DOCUMENT #1: THIS IS THE FIRST IN A SERIES
OF TEXT FILES TO BE CREATED ON THIS SYSTEM.

A>TYPE 2.TXT

DOCUMENT #2: THIS IS THE SECOND IN A SERIES
OF TEXT FILES TO BE CREATED ON THIS SYSTEM.

A>PIP B:ALL.TXT=A:1.TXT,2.TXT

A>TYPE B:ALL.TXT

DOCUMENT #1: THIS IS THE FIRST IN A SERIES
OF TEXT FILES TO BE CREATED ON THIS SYSTEM.

DOCUMENT #2: THIS IS THE SECOND IN A SERIES
OF TEXT FILES TO BE CREATED ON THIS SYSTEM.

A>

При помощи команды PIP можно объединять большое число файлов, хотя общая длина командной строки не должна превышать длину строки на экране видеотерминала или пишущей машинке (телетайпе). Объединять можно файлы различных типов, однако чаще всего объединяются файлы, которые содержат тексты, пригодные к выводу на консоль или печатающее устройство, например, файлы с объектным/машинным кодом, такие как командные (*.COM") файлы, хотя при этом можно сэкономить разве что дисковую память. В этом случае файл, содержащий объединенные файлы, нельзя использовать в качестве командного, если он впоследствии не будет должным образом изменен или же если данное объединение не было предусмотрено заранее.

Объединяемые файлы не обязаны находиться все на одном диске. Если перед именем каждого исходного файла указано имя соответствующего диска, то утилита "PIP" сможет обрабатывать файлы, расположенные на различных дисках. В представленном ниже примере исходные файлы расположены на дисках "B", "B", "C" и "D", тогда как результирующий файл расположен на диске "B":

A>PIP B:ALL.TXT=A:1.TXT,B:2.TXT,C:3.TXT,D:4.TXT

A>

Командная строка, описывающая объединение файлов, может содержать любые сочетания имен исходных файлов и дисков. Последовательность имен файлов отражает ту последовательность, в которой требуется их объединить. Данная команда будет выполнена корректно только в том случае, если имена всех файлов и дисков, перечисленные в командной строке, являются допустимыми.

ИСПОЛЬЗОВАНИЕ СТАНДАРТНЫХ И СПЕЦИАЛЬНЫХ ВЫХОДНЫХ УСТРОЙСТВ

Утилита "PIP" дает пользователю ОС CP/M средства копирования не только между дисковыми устройствами, но и между другими устройствами. В качестве параметров команды PIP могут быть специфицированы как имена

почти любого логического и физического устройства ОС *CP/M*, так и имена некоторых специальных устройств утилиты "PIP" (существует два специальных устройства, используемые для вывода данных). Стандартные логические и физические устройства ОС *CP/M* даны при описании транзитной команды STAT.

В качестве выходного устройства для утилиты "PIP" может быть специфицировано практически любое стандартное логическое или физическое устройство ОС *CP/M*. Исключение составляет устройство с именем "BAT:" (устройство пакетной обработки). Ниже представлен список допустимых выходных устройств утилиты "PIP".

Логические:	CON: (консоль)	PUN: (устройство вывода)	LST: (устройство печати)
Физические:	TTY:	PTP:	LPT:
	CRT:	UP1:	UL1:
	UC1:	UP2:	
Специальные:		OUT	PRN:

За более подробной информацией о характеристиках стандартных логических и физических устройств можно обратиться к описанию команды STAT. Ниже описывается использование различных выходных устройств утилиты "PIP".

ВЫВОД ДАННЫХ НА КОНСОЛЬ (ВИДЕОТЕРМИНАЛ ИЛИ ТЕЛЕТАЙП)

PIP CON = A:SAMPLE.DOC
PIP TTY = A:SAMPLE.DOC
PIP CRT = A:SAMPLE.DOC
PIP UC1 = A:SAMPLE.DOC

Консоль может быть специфицирована в качестве выходного устройства при копировании файла. При этом содержимое файла выводится на экран терминала или распечатывается на бумаге. (Команда PIP в данном формате производит то же самое действие, что и встроенная команда TYPE.) В данном случае можно указать или имя логического устройства CON:, или имена физических устройств TTY:, CRT: или UC1:. Помните, что имя физического устройства может быть указано только в том случае, когда по команде STAT это устройство было назначено логическому устройству CON:. Исключение составляют устройства TTY: и CRT:, поскольку для них утилита "PIP" автоматически предполагает, что вывод данных на эти устройства должен осуществляться через логическое устройство CON:. Что касается устройства UC1:, поскольку оно определяется пользователем, необходимо внести соответствующие изменения в секцию подпрограмм ввода-вывода операционной системы *CP/M* (BIOS). Если устройство UC1: специфицировано в качестве выходного устройства, а в ОС *CP/M* необходимые изменения не внесены, то система скорее всего станет неработоспособной, что потребует провести полную перезагрузку. Таким образом, самое надежное – указать в качестве выходного устройства логическое устройство CON: .

ВЫВОД ДАННЫХ НА УСТРОЙСТВА,
РАБОТАЮЩИЕ В РЕЖИМЕ "ТОЛЬКО ВЫВОД"

PIP PUN = A PROG-1 HEX
PIP PTP = A PROG-1.HEX
PIP UP1 = A PROG-1 HEX
PIP UP2 = A PROG-1.HEX
PIP OUT = A PROG-1 HEX

В качестве выходного устройства утилиты "PIP" может быть специфицировано одно из устройств, работающих в режиме "только вывод". Для этой цели обычно используется логическое устройство PUN: (Выход на перфоленту). Иногда это устройство используется не только как ленточный перфоратор. Для удовлетворения этого требования также допускается спецификация любого физического устройства при условии, что в большинстве случаев это устройство будет назначено логическому устройству PUN. По умолчанию в качестве ленточного перфоратора используется физическое устройство PTP:. Устройства UP1 и UP2 определяются пользователем. Имена этих устройств никогда нельзя использовать в качестве параметров команды PIP, если в ОС CP/M не внесены изменения, необходимые для их использования.

Кроме того, в утилите "PIP" можно указать специальное устройство, которое называется "OUT:". Это еще одно специальное устройство вывода, определяемое пользователем. Обычно это устройство назначается логическому устройству PUN:. Для того чтобы использовать устройство OUT:, необходимо с помощью транзитной команды DDT внести изменения в утилиту "PIP". Если вы не знакомы с данным процессом или если вы не хотите вносить в утилиту "PIP" никаких изменений, пропустите оставшуюся часть данного. Если вы знакомы с командой DDT и уже знаете, какие изменения необходимо внести, то будет полезна следующая информация:

ПРИМЕЧАНИЕ ДЛЯ ПРОГРАММИСТОВ

В утилите "PIP" в области памяти, расположенной по шестнадцатеричным адресам с 103H по 105H, находится команда перехода в специальную подпрограмму пользователя, обеспечивающую ввод данных. Для ввода очередного символа утилита "PIP" вызывает подпрограмму пользователя по адресу 103H, а после возврата из данной подпрограммы считывает символ по адресу 109H. Область памяти, расположенная по адресам с 106H по 108H, занята под команды перехода в специальную подпрограмму пользователя, обеспечивающую вывод данных. Утилита "PIP" загружает очередной символ, предназначенный для вывода в регистр С, а затем вызывает подпрограмму пользователя по адресу 106H. Область памяти с адресами с 10AH по 1FFH обычно не используется и может быть занята специальными подпрограммами пользователя.

Для получения более подробной информации необходимо обратиться к технической документации по ОС CP/M.

ВЫВОД ДАННЫХ НА ПЕЧАТАЮЩЕЕ УСТРОЙСТВО

```
PIP LST = A SAMPLE DOC  
PIP LPT = A SAMPLE DOC  
PIP UL1 = A SAMPLE DOC  
PIP PRN = A SAMPLE DOC
```

В качестве выходного устройства утилите "PIP" может быть назначено устройство печати. Наиболее используемым является логическое устройство LST:, хотя можно использовать и физическое устройство LPT:. Кроме того, можно специфицировать определяемое пользователем физическое устройство UL1:. Во всех случаях эти устройства можно использовать только при соответствующем изменении ОС CP/M.

Кроме того, существует специальное устройство утилиты "PIP", которое называется "PRN:". Использование устройства PRN: эквивалентно использованию устройства LST: или LPT: за исключением того, что в процессе вывода на это устройство утилита "PIP" выполняет определенные действия по форматированию данных. Если специфицировано устройство PRN:, то утилита "PIP" выполняет следующие действия:

1. Каждый раз при обнаружении символа табуляции на выходном устройстве (videoterminal, печать и т. п.) происходит смещение курсора до ближайшей позиции с номером, кратным восьми (8, 16, 24 и т. д.).
2. Все строки текста при выводе последовательно нумеруются.
3. При обнаружении команды перевода формата (form-feed) происходит переход на следующую страницу, причем переход на следующую страницу происходит через каждые 60 строк.

ИСПОЛЬЗОВАНИЕ СТАНДАРТНЫХ И СПЕЦИАЛЬНЫХ ВХОДНЫХ УСТРОЙСТВ

При использовании команды PIP помимо выходных устройств могут быть специфицированы также различные *входные* устройства. Исключение опять же составляет физическое устройство BAT: (устройство пакетной обработки), которое не может быть указано в качестве параметра команды PIP. Кроме того, существуют три специальных устройства утилиты "PIP", которые могут быть специфицированы в качестве входных устройств. Допустимые входные устройства перечислены ниже.

Логические:	CON: (консоль)	RDR: (устройство ввода)
Физические:	TTY:	PTR:
	CRT:	UR1:
	UC1:	UR2:
Специальные:		INP: NUL: EOF:

Для получения более подробной информации о характеристиках стандартных логических и физических устройств обратитесь к описанию транзитной команды STAT. Описание специальных входных устройств утилиты "PIP" представлено ниже.

ВВОД ДАННЫХ С КОНСОЛИ В ФАЙЛ НА ДИСКЕ

```
PIP TEST TXT=CON  
PIP TEST.TXT=TTY:  
PIP TEST.TXT=CRT:  
PIP TEST TXT=UC1:
```

Утилиту "PIP" можно использовать для непосредственного ввода данных с консоли в файл на диске. Для этого необходимо в качестве входного устройства указать консоль, а в качестве выходного – указать имя файла на диске. Для этих целей обычно используется логическое устройство CON:, хотя, кроме того, могут использоваться устройства TTY: и CRT:. Определяемое пользователем физическое устройство UC1: можно использовать только в том случае, когда в ОС CP/M внесены необходимые изменения.

Для того чтобы инициировать эту операцию, необходимо после команды PIP ввести спецификацию выходного файла и имя входного устройства консоли. После нажатия клавиши "RETURN" утилита "PIP" перейдет в режим передачи "консоль – диск". В этом режиме терминал консоли функционирует необычным образом. Поэтому с этого устройства можно вводить отдельные строки текста. После завершения данного режима вводится команда CTRL/Z (одновременно нажимаются клавиши CTRL и Z), в результате которой утилита "PIP" вставит в файл маркер конца файла, закроет файл и возвратит управление в ОС CP/M. Позже для контроля содержимое файла может быть распечатано на консоли (например, при помощи команды TYPE). Как это все происходит, продемонстрировано в следующем примере:

```
A>PIP A TEST.TXT=CON  
NOW IS THE TIME FOR ALL GOOD PROGRAMMERS TO  
TO SAVE THEIR WORK IN FILES  
^Z                                              <---- Команда CTRL/Z завершает ввод  
                                                данных  
  
A>TYPE TEST.TXT  
NOW IS THE TIME FOR ALL GOOD PROGRAMMERS TO  
SAVE THEIR WORK IN FILES  
  
A .
```

В данном случае ввод данных завершается только после введения команды CTRL/Z. Все остальные командные строки попадут в выходной файл. Эти командные строки не окажут никакого влияния на режим работы утилиты "PIP".

ВВОД ДАННЫХ С УСТРОЙСТВ, РАБОТАЮЩИХ В РЕЖИМЕ "ТОЛЬКО ВВОД", В ФАЙЛ НА ДИСКЕ

```
PIP PGM1.HEX=RDR:  
PIP PGM1.HEX=PTR:  
PIP PGM1.HEX=UR1:  
PIP PGM1.HEX=UR2:  
PIP PGM1.HEX=INP:
```

Утилиту "PIP" можно использовать для копирования данных с устройств, работающих в режиме "только чтение", в некоторый файл на диске. В число

устройств, работающих в режиме "только чтение" входит логическое устройство RDR:. Обычно данному логическому устройству назначается физическое устройство ввода с перфоленты PTR:. В качестве параметров команды PIP может быть указано или логическое устройство RDR:, или физические устройства PTR:, UR1: и UR2:. Определяемые пользователем устройства UR1: и UR2: могут использоваться только в том случае, когда ОС CP/M соответствующим образом модифицирована для их поддержки. В процессе передачи данных с этих устройств консоль функционирует обычным образом. При вводе с консоли команды CTRL/Z передача данных заканчивается и выходной файл сохраняется на диске.

Существует также специальное устройство утилиты "PIP", называемое "INP:", которое можно указать в качестве входного устройства, работающего в режиме "только чтение". Это специальное устройство можно использовать для выборки данных из определяемой пользователем секции утилиты "PIP". Для того чтобы использовать устройство INP:, необходимо с помощью транзитной команды DDT внести соответствующие изменения в утилиту "PIP". Если вносить какие-либо изменения в утилиту "PIP" не требуется или если не известно, как это сделать, можно пропустить приведенное ниже замечание. Если же есть навык работы с командой DDT и если известно, какие изменения надо внести, будут полезны следующие сведения.

ПРИМЕЧАНИЕ ДЛЯ ПРОГРАММИСТОВ ОС СР/М

Утилита "PIP" воспринимает входные данные символ за символом при обращении к подпрограмме по шестнадцатеричному адресу 103H. Возвращаемая подпрограммой информация расположена по шестнадцатеричному адресу 109H. Бит четности каждого символа должен быть установлен в ноль.

ПРОПУСК УЧАСТКОВ ПЕРФОЛЕНТЫ В НАЧАЛЕ И КОНЦЕ ВЫВОДА ДАННЫХ НА УСТРОЙСТВА, РАБОТАЮЩИЕ В РЕЖИМЕ "ТОЛЬКО ВЫВОД"

PIP PUN = NUL
PIP PUN = NUL,A POM1 HEX
PIP PUN = NUL,A POM1 HEX,NUL
PIP PUN = A POM1 HEX,NUL

Устройство NUL: является специальным устройством утилиты "PIP". Оно используется только в тех случаях, когда выходным является устройство вывода на перфоленту, назначенное стандартному выходному устройству PUN:. Если в командной строке устройство NUL: указывается наряду с именем какого-либо файла в качестве источника информации, то их следует отделять друг от друга запятой. При обнаружении имени NUL: в списке имен входных файлов и устройств утилиты "PIP" выдает на выходное устройство 40 "нулевых" (NUL) символов в коде ASCII. При выводе "нулевых" символов на устройстве перфорации будет пропущено 4 дюйма ленты. Это делается для создания "головных" и "хвостовых" участков на бумажной ленте, которые отделяют реальные данные и которые полезны при разрезании ленты. Как правило, это специальное устройство для других целей не используется. Некоторые разработчики могут модифицировать ОС CP/M под другие устройства, использующие устройство NUL: и его нулевые символы.

ВЫВОД МАРКЕРА КОНЦА ФАЙЛА ПРИ ВЫВОДЕ ДАННЫХ НА УСТРОЙСТВА, РАБОТАЮЩИЕ В РЕЖИМЕ "ТОЛЬКО ВЫВОД"

PIP PUN = A:PGM1 HEX,EOF

Специальное входное устройство EOF: (End-of-file – конец файла) используется для передачи на выходное устройство символа, принятого в ОС *CP/M* в качестве маркера конца файла (символ SUB в коде ASCII (шестнадцатеричное 1A) или символ CTRL/Z). Выходным устройством в подобных случаях обычно является логическое устройство PUN:, которому назначено физическое устройство – ленточный перфоратор. Маркер конца файла используется главным образом для завершения передачи данных, поэтому его можно использовать в качестве признака конца дискового файла. Этот символ используется в тех случаях, когда копируемые файлы не содержат собственных маркеров конца файла или же когда выходное устройство не опознает данный маркер. В обычных условиях в данном специальном устройстве нет необходимости, поскольку все правильно сформатированные файлы ОС *CP/M* содержат в качестве маркера конца файла символ CTRL/Z. Это устройство можно использовать совместно со специальным устройством NUL, пока имена устройств и имя файла отделены запятой. Устройство EOF: может быть полезным при передаче данных файла с нестандартных устройств ОС *CP/M* или при копировании файлов, созданных в других вычислительных системах.

ДОПОЛНИТЕЛЬНЫЕ ПАРАМЕТРЫ КОМАНДЫ PIP

В командные строки утилиты "PIP" можно добавлять специальные параметры для обозначения выполнения определенных действий в процессе копирования данных. Эти параметры обычно вводятся в конец командной строки и заключаются в квадратные ("[" и "]") скобки. В той же самой командной строке можно ввести один или несколько параметров. Все они должны быть напечатаны в одну строку и заключены в одну пару квадратных скобок. Параметры являются необязательными, хотя может оказаться, что один или два используются почти все время. Все эти параметры описываются ниже в следующем порядке: сначала описываются те параметры, которые используются чаще всего, затем те, которые реже.

СРАВНЕНИЕ КОПИИ И ОРИГИНАЛА ФАЙЛА ДЛЯ ПРОВЕРКИ ПРАВИЛЬНОСТИ КОПИРОВАНИЯ

[V]

Параметр [V] (сокращение от Verify – проверить) используется, пожалуй, наиболее часто. Если он указан, то утилита "PIP" сравнивает файл-копию с файлом-оригиналом с целью проверки правильности копирования (для этого использует имя выходного файла и расширение ".QQQ"). Если ошибка

не обнаружена, то утилита "PIP" завершит свое выполнение или перейдет к обработке других команд. При обнаружении ошибки на консоль выдается соответствующее сообщение. Эта проверка будет произведена только в том случае, когда входной и выходной файлы расположены на диске. При копировании "файл-в-файл" параметр [V] вводится без пробела сразу после имен входного и выходного файлов. При копировании нескольких файлов и их объединении параметр [V] должен быть указан после имени первого исходного файла. В противном случае данный параметр игнорируется. Этот параметр полезно использовать всегда, поскольку он позволяет обнаруживать ошибки в копировании из-за сбоя программы, порчи дисков, сбоев в оборудовании дисковых устройств или в оперативной памяти ЭВМ.

```
A>PIP A:=B:SAMPLE.TXT[V]  
A>PIP A:ALL.TXT=B:1.TXT[V],2.TXT  
A>
```

[E]

ВЫВОД ВСЕГО ТЕКСТА НА КОНСОЛИ
В ПРОЦЕССЕ КОПИРОВАНИЯ

Параметр [E] (сокращение от Echo – эхо) может использоваться с любой командой утилиты "PIP" при копировании текстовых файлов. Параметр [E] можно использовать вместе с любыми другими параметрами, управляющими передачей данных, представленных в коде ASCII. Использование параметра [E] при копировании файлов с объектным/машинным кодом, таких как файл ".COM", приводит к непредвиденным результатам, поскольку такие файлы воздействуют на консоль (вызывают, например, инверсию, перемещение, сдвиг текста, мерцание, табуляцию и т. п.). Этот параметр может быть специфицирован при копировании одного файла, хотя чаще он используется при копировании и объединении нескольких файлов в один файл. Если при этом ввести параметр [E], то текст каждого файла по мере копирования будет отображен на консоли, что позволит пользователю судить о том, в какой стадии находится процесс копирования (середина файла, конец первого файла, начало второго файла, третий файл и т. д.).

Совместно с параметром [E] целесообразно использовать параметр [V] (Проверить). В следующем примере демонстрируется вывод текста одного копируемого файла на консоль:

```
A>ФАЙЛ TEST.TXT СОДЕРЖИТ СЛЕДУЮЩИЙ ТЕКСТ  
A>TYPE TEST.TXT  
THIS IS A TEST ASCII TEXT FILE.  
A>PIP B:=A:TEST.TXT[V]  
THIS IS A TEST ASCII TEXT FILE.
```

A>В ПРОЦЕССЕ КОПИРОВАНИЯ СОДЕРЖИМОЕ ФАЙЛА ВЫВОДИТСЯ НА
A>КОНСОЛЬ.
A>

В следующем примере демонстрируются достоинство параметра [E], которое проявляется при копировании и объединении нескольких файлов в один:

```
A>:ПОСЛЕДОВАТЕЛЬНЫЙ ВЫВОД НА КОНСОЛЬ СОДЕРЖИМОГО  
A>:НЕСКОЛЬКИХ ФАЙЛОВ  
A>PIP B:FINAL.TXT=A:TEST1.TXT,TEST2.TXT[E]  
THIS IS THE TEXT OF THE FIRST FILE.  
THIS IS THE TEXT OF THE SECOND FILE.  
A>
```

Этот параметр может оказаться не столь уж полезным в тех системах, в которых в качестве консоли используется видеотерминал, из-за того что часто копируемые данные отображаются на экране слишком быстро, чтобы их можно было бы прочитать, особенно если копируются файлы небольших размеров. Эта проблема не возникает в тех случаях, если в качестве консоли используется телетайп или другое печатающее устройство.

Конечно, параметр [E] потребуется только в тех случаях, когда уже нет уверенности, что тексты объединяемых файлов готовы.

КОПИРОВАНИЕ ФАЙЛА, НАЧИНАЯ С УКАЗАННОЙ СТРОКИ ТЕКСТА

[Sstring ^ Z]

Параметр [S] (сокращение от Start – начало) указывает на то, что копирование входного файла необходимо начать только *после* обнаружения указанной строки текста. Текст, следующий за этой строкой, будет скопирован, а текст, предшествующий ей, будет пропущен. Этот параметр полезен тогда, когда требуется скопировать не весь файл, а некоторую его часть.

Его можно использовать, например, для скопирования только последнего раздела документа, последнего участка программного кода и т. п.

Параметр [S] вводится в конце командной строки следующим образом: печатаются символы "[S", а затем строку текста, которая указывает на начало секции входного файла, которую надо скопировать. В конце данной строки вводят команду CTRL/Z (при одновременном нажатии клавиш CTRL и Z) и обязательно символ "]". Например, если требуется начать копирование файла, начиная со строки "CHAPTER2", то соответствующий параметр выглядит так: "SCHAPTER2 ^Z". Параметр [S] можно использовать и совместно с другими параметрами, пока все заданные параметры относятся к текстовым файлам.

В следующем примере показаны результаты использования параметра [S]:

```
A>TYPE MANUAL1.DOC  
CHAPTER 1 (INSERT TEXT AT A LATER DATE)  
CHAPTER 2 (INSERT TEXT AT A LATER DATE)  
CHAPTER 3 (INSERT TEXT AT A LATER DATE)  
CHAPTER 4 (INSERT TEXT AT A LATER DATE)  
  
A>PIP B:CHAP-4.TXT=A:MANUAL1.DOC[SCHAPTER 4^Z]  
  
A>TYPE B:CHAP-4.TXT  
CHAPTER 4 (INSERT TEXT AT A LATER DATE)  
  
A>  
4*
```

Параметр [S] может быть также использован при объединении нескольких файлов в один файл. Для этого данный параметр вместе со "стартовой" строкой необходимо указывать сразу после имени входного файла, но до запятой, отделяющей имена входных файлов друг от друга. В следующем примере показано, как текст из файла "1.DOC", начинающийся со строки "CHAPTER 3", и текст из файла "2.DOC", начинающийся со строки "CHAPTER 4", копируются и объединяются в файле "3.DOC":

```
A>PIP B:3.DOC=A:1.DOC[SCHAPTER 3^Z],A:2.DOC[CHAPTER 4^Z]
```

А)

КОПИРОВАНИЕ ФАЙЛА ДО УКАЗАННОЙ
СТРОКИ ТЕКСТА

[Qstring^Z]

Параметр [Q] (сокращение от Quit – прекратить) указывает утилите "PIP" на то, что *после* обнаружения указанной строки текста копирование файла должно быть прекращено. Весь текст, *предшествующий* данной строке, будет скопирован. Весь оставшийся текст будет пропущен. Параметр [Q] полезен в том случае, когда необходимо скопировать не весь файл, а его часть.

Этот параметр вводится в конце командной строки следующим образом: печатают символы "[Q", а затем та строка текста, которая указывает на конец копируемой секции входного файла. В конце данной строки вводят команду CTRL/Z (при одновременном нажатии клавиш CTRL и Z), а затем символ "]"). Например, если требуется завершить копирование файла после строки "CHAPTER 2", то соответствующий параметр выглядит "[QCHAPTER 2 ^Z]". Параметр [Q] можно использовать совместно со всеми остальными параметрами, пока все указанные параметры имеют отношение к текстовым файлам:

```
A>TYPE MANUAL1.DOC  
CHAPTER 1 (INSERT TEXT AT A LATER DATE)  
CHAPTER 2 (INSERT TEXT AT A LATER DATE)  
CHAPTER 3 (INSERT TEXT AT A LATER DATE)  
CHAPTER 4 (INSERT TEXT AT A LATER DATE)
```

```
A>PIP B:TEMP.TXT=A:MANUAL1.DOC[QCHAPTER 2^Z]
```

```
A>TYPE B:TEMP.TXT  
CHAPTER 1 (INSERT TEXT AT A LATER DATE)  
CHAPTER 2
```

А)

Кроме того, параметр [Q] можно использовать при копировании нескольких, объединяемых в один, файлов. Для этого параметр вместе с "ограничительной" строкой необходимо указывать сразу после имени каждого файла до запятой, отделяющей имена входных файлов друг от друга. В следующем примере показано, как текст из файла "1.DOC", заканчивающийся строкой "CHAPTER 1", и текст из файла "2.DOC", заканчивающийся строкой "CHAPTER 2", копируются и объединяются в файле "3.DOC":

A>

СОВМЕСТНОЕ ИСПОЛЬЗОВАНИЕ ПАРАМЕТРОВ "НАЧАЛО КОПИРОВАНИЯ" И "КОНЕЦ КОПИРОВАНИЯ"

[S]string1 ^ ZQstring2 ^ Z]

Параметры [S] (начало копирования) и [Q] (конец копирования) могут быть использованы вместе, при этом утилита "PIP" начнет копирование файла с определенной точки и завершит копирование в другой заданной точке. Для этого оба параметра вводятся внутри одной пары квадратных скобок ("[]"). Первым вводят параметр "S", затем параметр "Q".

В следующем примере показано, как можно скопировать текст из файла "MANUAL1.DOC", начинающийся со строки "CHAPTER 2" и заканчивающийся строкой "CHAPTER 4":

```
A>TYPE MANUAL1.DOC
CHAPTER 1 (INSERT TEXT AT A LATER DATE)
CHAPTER 2 (INSERT TEXT AT A LATER DATE)
CHAPTER 3 (INSERT TEXT AT A LATER DATE)
CHAPTER 4 (INSERT TEXT AT A LATER DATE)
CHAPTER 5 (INSERT TEXT AT A LATER DATE)

A>PIP B:=A:MANUAL1.DOC[CHAPTER 2^ZQCHAPTER 4^Z]

A>TYPE B:MANUAL1.DOC
CHAPTER 2 (INSERT TEXT AT A LATER DATE)
CHAPTER 3 (INSERT TEXT AT A LATER DATE)
CHAPTER 4 (INSERT TEXT AT A LATER DATE)
```

A>

Комбинации параметров "начало копирования" и "конец копирования" часто бывают полезны в тех случаях, когда требуется скопировать несколько файлов для объединения их в одном файле. Например, может понадобиться скопировать последнюю часть из первого файла, первую часть из второго файла и объединить все это в одном файле. Параметры "начало копирования" и "конец копирования" можно использовать для копирования отдельных частей из различных файлов в произвольных комбинациях.

ПРЕОБРАЗОВАНИЕ СИМВОЛОВ ВЕРХНЕГО РЕГИСТРА В СИМВОЛЫ НИЖНЕГО РЕГИСТРА ПРИ КОПИРОВАНИИ

[L]

Параметр [L] (сокращение от Lower – нижний) используется в тех случаях, когда в процессе копирования необходимо преобразовать символы верхнего регистра в эквивалентные символы нижнего регистра. Этот параметр можно использовать только при обработке текстовых файлов совмест-

но со всеми другими параметрами утилиты "PIP", применяемыми при обработке текстов. Параметр [L] полезен часто в тех случаях, когда все символы их исходного файла представлены в кодах верхнего регистра (т. е. все буквы являются заглавными), а требуется создать копию данного файла, со стандартным представлением в кодах верхнего и нижнего регистров. Утилита "PIP" не создает символы верхнего и нижнего регистра, однако впоследствии для выделения прописных букв в начале предложений и т. п. можно использовать текстовый редактор. Таким образом, параметр [L] имеет смысл использовать потому, что при помощи тестового редактора гораздо легче вставить в текст необходимые заголовки, чем вручную переводить все символы в коды нижнего регистра. В командной строке параметр [L] вводится после имени входного файла. Например:

```
A>TYPE MANUAL1.DOC
CHAPTER 1 (INSERT TEXT AT A LATER DATE)
CHAPTER 2 (INSERT TEXT AT A LATER DATE)
CHAPTER 3 (INSERT TEXT AT A LATER DATE)
CHAPTER 4 (INSERT TEXT AT A LATER DATE)
CHAPTER 5 (INSERT TEXT AT A LATER DATE)

A>PIP B:LOWER.TXT=A:MANUAL1.DOC[L]

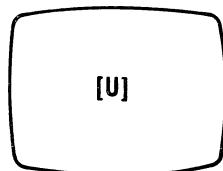
A>TYPE B:LOWER.TXT
chapter 1 (insert text at a later date)
chapter 2 (insert text at a later date)
chapter 3 (insert text at a later date)
chapter 4 (insert text at a later date)
chapter 5 (insert text at a later date)

A>
```

Заметим, что использование данного параметра не ограничивается процедурой копирования. Его можно также вводить при обмене данными между стандартными устройствами ОС *CP/M* или специальными устройствами утилиты "PIP". Например, можно обработать текст, полученный от входного устройства RDR: (устройство чтения с перфоленты), в процессе его передачи на выходное устройство LST:. Параметр [L] может быть использован почти при любых комбинациях входных и выходных устройств или файлов, которые допускают передачу текста:

```
A>PIP LST:RDR:FL3
A>PIP LST:=A:UPPER.TXT[L]
```

ПРЕОБРАЗОВАНИЕ СИМВОЛОВ НИЖНЕГО РЕГИСТРА В СИМВОЛЫ ВЕРХНЕГО РЕГИСТРА ПРИ КОПИРОВАНИИ



[U]

Параметр [U] (сокращение от Upper – верхний) используется при обработке только текстовых файлов совместно со всеми другими параметрами, применяемыми при обработке текстов. Этот параметр используется в тех случаях, когда в процессе копирования необходимо преобразовать все сим-

волы, представленные в кодах нижнего регистра, в эквивалентные символы, представленные в кодах верхнего регистра. Этот параметр часто полезен тогда, когда исходный файл содержит символы в кодах нижнего регистра или смешанные (в верхнем и нижнем регистре), а требуется создать копию файла с символами в кодах только верхнего регистра такого, который содержит исходный код для какого-нибудь транслятора (например, для ассемблера). Файлы с исходными кодами могут содержать символы или в кодах верхнего, или в кодах нижнего регистров, и для создания копий файлов, содержащих только символы в "официальном" коде верхнего регистра, можно использовать команду PIP с параметром [U].

Должны ли все символы файла быть представлены в кодах нижнего регистра, в кодах верхнего регистра или же в смешанном виде, обычно зависит от того, что предпочитает пользователь.

В командной строке параметр [U] вводится сразу после имени исходного файла. Например:

```
A>TYPE LOWER.TXT
chapter 1 (insert text at a later date)
chapter 2 (insert text at a later date)
chapter 3 (insert text at a later date)
chapter 4 (insert text at a later date)
chapter 5 (insert text at a later date)

A>PIP B:UPPER.TXT=A:LOWER.TXT[U]

A>TYPE B:UPPER.TXT
CHAPTER 1 (INSERT TEXT AT A LATER DATE)
CHAPTER 2 (INSERT TEXT AT A LATER DATE)
CHAPTER 3 (INSERT TEXT AT A LATER DATE)
CHAPTER 4 (INSERT TEXT AT A LATER DATE)
CHAPTER 5 (INSERT TEXT AT A LATER DATE)
```

A>

Отметим, что использование параметра [U] ограничивается процедурой копирования. Его можно также использовать при обмене данными между стандартными устройствами ОС CP/M или специальными устройствами утилиты "PIP". Например, можно переделать текст, полученный от входного устройства RDR: (устройство чтения с перфоленты) в процессе его передачи данных на выходное устройство LST:. Этот параметр может быть использован почти в любых комбинациях входных и выходных устройств или файлов, допускающих передачу текста.

```
A>PIP LST:=RDR:[U]
A>PIP LST:=A:LOWER.TXT[U]
```

УДАЛЕНИЕ В КАЖДОЙ СТРОКЕ ВСЕХ СИМВОЛОВ,
РАСПОЛОЖЕННЫХ ПОСЛЕ УКАЗАННОЙ ПОЗИЦИИ

[Dn]

Параметр [D] (сокращение от Delete – удалить) используется для замещения или удаления в каждой строке всех символов, которые расположены

после указанной позиции. Данное средство иногда полезно при копировании текстовых данных в предназначенный для этого файл, формат которого уже, чем средняя ширина строк текста (например, при использовании видеотерминала или принтера со строками шириной в 40 колонок или при выводе текста со строками шириной в 132 колонки на устройство со строками шириной в 80 колонок). Если, например, параметр [D] специфицирует удаление всех символов после 80-й колонки в тексте с шириной строк 132 колонки, то все символы с первого по восемьдесятый столбец включительно будут выданы на выходное устройство, а символы с восемьдесят первого по сто тридцать второй будут удалены. Параметр [D] редко используется при копировании данных в файл, а не на устройство.

Параметр [D] вводится следующим образом: печатаются символы "[D]" и число ("n"). Число n может принимать любые значения в интервале от 1 до 255. Если строки во входном файле окажутся длиннее, чем 255 символов, то удаление символов будет выполнено должным образом. После параметра необходимо ввести символ "]".

В следующем примере показано, как текст, содержащий строки длиной в 60 колонок, подготавливается к выводу на терминал со строками в 40 колонок.

```
A>TYPE 60-COL.TXT
      1   2   3   4   5   6
123456789012345678901234567890123456789012345678901234567890

A>PIP CON:=A:60-COL.TXT[40]
      1   2   3   4
1234567890123456789012345678901234567890
```

A>

[Tn]

ПРЕОБРАЗОВАНИЕ СИМВОЛОВ ТАБУЛЯЦИИ В УКАЗАННОЕ ЧИСЛО ПРОБЕЛОВ

Параметр [T] (сокращение от Tab – табуляция) используется в тех случаях, когда символ НТ (горизонтальная табуляция) или ^I (CTRL/I) необходимо заменить заданным числом пробелов. В тех случаях, когда тексты создаются при помощи некоторых текстовых редакторов или текстовых процессоров, символы горизонтальной табуляции кода ASCII не всегда представляются в виде пробелов. Поскольку некоторые терминалы и устройства печати не могут должным образом распознать и обработать символы НТ или ^I, то для преобразования всех символов ^I в заданное число пробелов можно использовать команду PIP с параметром [T]. (При этом печать текста продолжается с назначенной позиции.) Этот процесс преобразования затрагивает только символы ^I. Если пробелы в тексте порождены командой Tab текстового редактора или процессора, то пробелы будут проигнорированы параметром [T].

Параметр [T] вводится следующим образом: печатаются символы "[T]" и требуемое число ("n"). Это число может быть практически любым, хотя редко превышает 20.

Ввод этого параметра завершается символом "]".

```
A>TYPE PROG-1.BAS
0010^IREM HOME-BREW BASIC PROGRAM 1
0020^IREM THIS IS A TEST PROGRAMM LISTING
0030^IREM TO SEE HOW TABS DISPLAY

A>PIP PROG-1B.BAS=PROG-1.BAS[TB]

A>TYPE PROG-1.BAS
0010 REM HOME-BREW BASIC PROGRAMM 1
0020 REM THIS IS A TEST PROGRAMM LISTING
0030 REM TO SEE HOW TABS DISPLAY
```

A)

Отметим, что в предыдущем примере при выводе файла на терминал с помощью команды TYPE его текст не дополнен пробелами, хотя внутри его и содержатся символы табуляции. Предположим, что в данном случае символу табуляции по умолчанию соответствует 8 пробелов, тогда во всех строках текст, начинающийся с "REM", должен печататься с восьмой колонки. Однако если терминал, на который выводится этот файл, не обладает средствами для корректной обработки символов табуляции, то для решения данной проблемы можно воспользоваться командой PIP с параметром [T]. При этом можно создать новую копию исходного файла и затем вывести новый преобразованный таким образом файл на устройство печати или дисплей.

УДАЛЕНИЕ В ПРОЦЕССЕ КОПИРОВАНИЯ ВСЕХ СИМВОЛОВ ПЕРЕВОДА ФОРМАТА

[F]

Параметр [F] (сокращение от Form-Feed – перевод формата) используется в тех случаях, когда в процессе копирования необходимо удалить все символы перевода формата в коде ASCII (FF или ^L). Часто нежелательно хранить эти символы в файле, который печатается редко и используется только для хранения данных. Символы перевода формата используются в текстовых файлах в качестве команд перехода на новую страницу при форматированном выводе текста на печатающее устройство. Кроме того, параметр [F] можно использовать в тех случаях, когда файл должен быть обработан текстовым редактором, или текстовым форматером, или текстовым процессором (например, текстовым процессором WordStar), которые используют специальные команды перевода страниц. В этом случае наличие в тексте символов перевода формата привело бы к возникновению некоторых проблем. Использование параметра [F] приведет к тому, что утилита "PIP" удалит все допустимые в коде ASCII символы перевода формата, обнаруженные в процессе копирования.

В командной строке параметр [F] вводится после имени входного устройства или файла. Параметр может использоваться совместно со всеми другими параметрами команды PIP, применяемыми при копировании текстов.

Отметим, что данный параметр может использоваться в сочетании с параметром ”[Pn]” (вставка символов перевода формата). В этом случае можно удалить в тексте существующие символы перевода формата и вставить в указанные места новые. Когда используются оба параметра, параметр [F] вводится до параметра [P], и оба заключаются в одну пару квадратных скобок, т. е. ”[FPn]”.

ВСТАВКА СИМВОЛА ПЕРЕВОДА ФОРМАТА ПОСЛЕ КАЖДОГО УКАЗАННОГО ЧИСЛА СТРОК ПРИ КОПИРОВАНИИ

[Pn]

Параметр [P] (сокращение от Pagination – разбивка по страницам) используется для того, чтобы в процессе копирования вставлять в текст символы перевода формата (команды перехода на новую страницу) после каждого указанного числа строк текста. Этот параметр чаще всего используется тогда, когда выходным является печатающее устройство. Данный параметр вводится сразу после имени входного файла (устройства) и начинается с символов ”[P”, за которыми следует число (”п”). Число ”п” может иметь любое значение в интервале от 1 до 255, хотя это значение должно быть в пределах общей длины каждого листа бумаги печатающего устройства. Специфицированное значение ”п” имеет силу при выводе всего текста на выходное устройство. Если указать значение, превышающее общее число строк, которое можно напечатать на странице, то это приведет к нежелательным последствиям. Если число ”п” не указано или если равно 1, то утилита ”PIP” использует принятое по умолчанию значение 60. Например, если печатающее устройство допускает печать 66 строк на странице, то значение 60 может быть по умолчанию при помощи параметра ”[P]”. В этом случае до перехода на новую страницу каждый раз будет напечатано только 60 строк данного параметра. Ввод параметра [Pn] завершается символом ”]”.

Параметр [P] не следует использовать, если выходным устройством является печатающее устройство, которое в начале и конце каждой страницы автоматически оставляет пустые поля. Обычно эти два поля в сумме составляют 6 строк. Устройства этого типа печатают данные только в промежутке между этими двумя полями. Поэтому при $n = 60$ на одной странице будет напечатано только по 54 строки.

Отметим, что этот параметр можно использовать совместно с параметром ”[F]” (удаление существующих символов перевода формата), как было описано выше. При этом можно удалить старые символы перевода формата и вставить в определенных позициях текста новые. При указании обоих параметров параметр [F] вводится перед параметром [P], и оба заключаются в одну пару квадратных скобок, т. е. ”[FPn]”.

ВНЕСЕНИЕ В ТЕКСТ ПОРЯДКОВЫХ НОМЕРОВ СТРОК
ПРИ КОПИРОВАНИИ

[N]
[N2]

Параметры [N] и [N2] (сокращение от Number – номер) используются для внесения в текст порядковых номеров строк текста в процессе его копирования. Параметр [N] вызывает смещение каждой строки текста на восемь позиций вправо. Эти свободные места используются для размещения номеров строк. Самая правая цифра номера помещается в седьмой колонке. В восьмой колонке каждой строки помещается двоеточие (":"), а сам текст начинается с девятой колонки. Отметим, что смещение каждой строки на восемь позиций вправо увеличит общую ширину каждой строки на восемь символов. Это может оказаться нежелательным, особенно если ширина строки на выходном устройстве меньше ширины строки в выводимом тексте. Результаты использования параметра [N] представлены в следующем примере:

```
A>TYPE MANUAL1.TXT
FIRST LINE
*
TENTH LINE
*
HUNDREDTH LINE
*
THOUSANDTH LINE

A>PIP MANUAL2.TXT=MANUAL1.TXT[N]

A>TYPE MANUAL2.TXT
1:FIRST LINE
*
10:TENTH LINE
*
100:HUNDREDTH LINE
*
1000:THOUSANDTH LINE
```

A>

Порядковые номера строк могут оказаться полезными при редактировании текстов документов и когда желательно обращаться к строкам по их номерам. Например, можно обратиться к строкам 100 . . . 105, 2078 и т. д.

Параметр [N2] может использоваться вместо параметра [N] в тех случаях, когда номера всех строк должны содержать лидирующие нули, т. е. нули в позициях, предшествующих реальным порядковым номерам. При использовании параметра [N2] двоеточие (":") заменяется пробелом. Результаты использования параметра [N2] представлены в следующем примере:

```
A>TYPE MANUAL1.TXT
FIRST LINE
*
TENTH LINE
*
HUNDREDTH LINE
*
THOUSANDTH LINE
```

A>PIP MANUAL2.TXT=MANUAL1.TXT[N2]

```
A>TYPE MANUAL2.TXT
0000001 FIRST LINE
*
0000010 TENTH LINE
*
0000100 HUNDREDTH LINE
*
0001000 THOUSANDTH LINE
```

A>

[R]

КОПИРОВАНИЕ "СИСТЕМНЫХ" ФАЙЛОВ
(ТОЛЬКО В ОС CP/M ВЕРСИИ 2.0 И ВЫШЕ)

Все файлы в ОС *CP/M* имеют или атрибут "SYS" ("системный") или атрибут "DIR" ("несистемный"), которые назначаются им при помощи команды STAT. Обычно утилита "PIP" не "видит" и не копирует системный файл, указанный в качестве входного файла. Параметр [R] (сокращение от Read – читать), который доступен только в ОС *CP/M* версии 2.0 и выше, позволяет создать копию файла, которому назначен атрибут "SYS". Параметр [R] используется, таким образом, для того, чтобы при поиске входного файла в каталоге диска этот атрибут не учитывался.

Этот параметр используется в тех случаях, когда необходимо скопировать группу защищенных системных файлов, например все утилиты ОС *CP/M*, оверлейные файлы, специальные модули и т. д.

[W]

УДАЛЕНИЕ ЗАЩИЩЕННОГО ФАЙЛА, ИМЕЮЩЕГО ТО ЖЕ ИМЯ,
ЧТО И ВЫХОДНОЙ ФАЙЛ (ТОЛЬКО В ОС CP/M ВЕРСИИ 2.0
И ВЫШЕ)

Параметр [W] является сокращением от Write – писать. Этот параметр в тех случаях можно использовать, когда известно, что на выходном диске существует защищенный файл, статус защиты которого "R/O" и имя которого совпадают с именем выходного файла, указанным в командной строке утилиты "PIP". Защищенные файлы "R/O" нельзя ни удалить, ни изменить. Их статус защиты "R/O" назначается с помощью команды STAT. Обычно утилита "PIP" в подобных обстоятельствах запрашивает, требуется ли удалить прежнее содержимое защищенного файла и заменить его копией файла, указанного в командной строке для выходного устройства. Чтобы избежать такие запросы, можно указать параметр [W], при этом содержимое выходного файла будет заменяться автоматически.

Однако параметр [W] следует применять только тогда, когда есть полная убежденность в том, что эти действия необходимы. Средство защиты файлов

от записи и стандартный запрос утилиты "PIP", требующий подтверждения намерения удалить прежнее содержимое защищенного файла, являются ценными качествами ОС *CP/M*. Они могут помочь предотвратить случайную потерю важных файлов, и от них не следует отказываться.

КОПИРОВАНИЕ ФАЙЛОВ ИЗ ОБЛАСТЕЙ ДРУГИХ ПОЛЬЗОВАТЕЛЕЙ (ТОЛЬКО В ОС *CP/M* ВЕРСИИ 2.0 И ВЫШЕ)

[Gn]

Параметр [G] используется в тех ситуациях, когда необходимо скопировать файлы из области одного пользователя в другую. Если используется параметр [G], то он относится только к областям пользователя входных файлов. По существу, это означает, что утилиту "PIP" можно использовать только для копирования файлов из других областей пользователя в текущую, а не наоборот. Иными словами, ее нельзя использовать для копирования файлов в другие области пользователя.

Этот параметр вводится после имени входного файла следующим образом: сначала идут символы "[G]", затем номер области пользователя ("п") и символ "]". Можно указать любой номер от 0 до 15. В следующем примере продемонстрировано несколько типичных командных строк утилиты PIP для копирования файлов из других областей пользователя.

```
A>:КОПИРОВАТЬ ИЗ ОБЛАСТИ 0 НА ДИСКЕ "A" B  
A>:ТЕКУЩУЮ ОБЛАСТЬ НА ДИСКЕ "A"  
A>PIP A:=A:STAT.COM[G0]  
  
A>:КОПИРОВАТЬ ИЗ ОБЛАСТИ 7 НА ДИСКЕ "B"  
A>:В ТЕКУЩУЮ ОБЛАСТЬ НА ДИСКЕ "A"  
A>PIP B:=A:SYSGEN.COM[G7]  
  
A)
```

Все это дает возможность на обмен файлами с другими пользователями диска, если имена этих файлов известны.

КОПИРОВАНИЕ ФАЙЛОВ, СОДЕРЖАЩИХ ОБЪЕКТНЫЙ/МАШИННЫЙ КОД

[O]

Параметр [O] (сокращение от *Object* – объектный) используется только для входных файлов, которые содержат объектный или машинный код программы (т. е. "двоичную" информацию, а не код ASCII). Параметр [O] указывает на необходимость игнорировать в процессе копирования все действия управляющих символов CTRL/Z (SUB в коде ASCII), которые могли встретиться в "двоичном" файле. Файлы такого типа иногда содержат в качестве обычновенных символов один или несколько управляющих символов CTRL/Z. Поскольку в ОС *CP/M* символ CTRL/Z используется как маркер

конца файла, то утилита "PIP" разработана так, что при обнаружении этого символа операция копирования завершается. Поэтому утилите "PIP" следует предписать игнорировать в процессе копирования все символы CTRL/Z. Иначе утилита "PIP" может обнаружить символ CTRL/Z в середине файла и преждевременно завершить процедуру копирования.

Примечание. Этот параметр не нужно вводить с ".COM" файлами (файлы с транзитными программами), поскольку с этими файлами утилита "PIP" выполняет рассмотренную функцию автоматически. Файлы с объектными и машинными кодами необходимо копировать в режиме [O].

В следующем примере показано, как этот параметр добавляется в типичную командную строку утилиты "PIP". В первой команде иллюстрируется использование самого параметра [O]. В следующей та же самая командная строка вводится с параметрами [O] и [V] (проверка). Оба параметра вводятся в произвольном порядке:

```
A>PIP A:=B:PGM-01.BIN[O]
A>PIP A:=B:PGM-01.BIN[V,O]
A>PIP A:ALL.BIN=B:1.BIN[V,O],2.BIN
```

A)

УСТАНОВКА БЛОЧНОГО РЕЖИМА КОПИРОВАНИЯ

[B]

Параметр [B] (сокращение от Block Mode – блочный режим) указывает, что входные данные должны копироваться в буфер (временную область памяти) до тех пор, пока во входном файле не будет обнаружен символ DC3 кода ASCII (^S, иногда называемый "X-OFF"). Как только данный символ обнаружен, передача данных из входного файла прекращается, а данные из буфера передаются на выходное устройство. После копирования содержимого буфера на выходное устройство, утилита "PIP" возобновит ввод следующего "блока данных" из входного файла. Затем процесс повторяется. Параметр [B] используется при копировании с устройств с последовательной передачей данных, таких как устройство чтения с перфоленты. Параметр [B] необходимо вводить потому, что утилита "PIP" не в состоянии одновременно осуществлять наполнение и освобождение буфера при поступлении данных от входного устройства сплошным потоком. Однако исходные данные должны содержать правильно размещенные символы DC3 кода ASCII ("X-OFF") в соответствии с размером буфера, принятым в утилите "PIP". Этот параметр никогда не используют при копировании "файл-в-файл".

УКАЗАНИЕ ШЕСТНАДЦАТИЧНОГО ФОРМАТА
ПЕРЕДАЧИ ДАННЫХ

[H]

Параметр [H] (сокращение от Hexadecimal – шестнадцатеричный) указывает, что данные с входного устройства (или файла) поступают в специальном шестнадцатеричном формате микропроцессора фирмы Intel. Этот формат обычно используется только в устройствах ввода-вывода на перфоленту. Если параметр [H] указан, то в процессе копирования будет производиться специальная проверка правильности шестнадцатеричного формата данных. При обнаружении ошибок утилита "PIP" выдает на консоль сообщение о требующихся исправлениях.

ИГНОРИРОВАНИЕ ЛЮБОЙ НУЛЕВОЙ ЗАПИСИ ПРИ
ПЕРЕДАЧЕ ДАННЫХ В ШЕСТНАДЦАТИЧНОМ
ФОРМАТЕ ФИРМЫ Intel

[I]

Параметр [I] (сокращение от Ignore – игнорировать) указывает утилите "PIP" на необходимость пропускать при копировании данных в шестнадцатеричном формате микропроцессора фирмы Intel любой нулевой (:00) записи. Этот параметр используется только совместно с параметром [H] (см. выше).

УСТАНОВЛЕНИЕ В НОЛЬ БИТА ЧЕТНОСТИ КАЖДОГО
КОПИРУЕМОГО БАЙТА

[Z]

Параметр [Z] (сокращение от Zero – ноль) используется при выполнении операции копирования. Он указывает, что бит четности (8-й бит) во всех символах кода ASCII должен быть установлен в ноль ("0"). В сущности, использование параметра [Z] приводит к преобразованию символов в 8-битовом коде ASCII (с контролем четности) в символы 7-битового кода ASCII (без контроля четности). Поскольку обработка символов 8-битового кода ASCII приводит в ОС CP/M к некоторым недоразумениям, то часто параметр [Z] оказывается весьма полезным.

Версии ОС *CP/M*:

1.3 1.4 2.0 2.2

Тип команды:

транзитная

ОБЩИЕ СВЕДЕНИЯ

Команда ED (сокращение от английского EDitor – редактор) является транзитной командой ОС *CP/M* и обеспечивает обращение к транзитной программе, которая также называется "ED". Эта программа предоставляет пользователю ОС *CP/M* наиболее простые средства, при помощи которых он может создавать текстовые файлы, просматривать их содержимое и вносить в них любые изменения. В отличие от некоторых программ редактирования текстов, которые называются "экранными редакторами" (т. е. полностью используют возможности дисплея), программа "ED" является строкоориентированным текстовым редактором. Это означает, что она обрабатывает (читывает, изменяет) содержимое текстовых файлов построчно. Однако программа "ED" отличается от многих других строкоориентированных текстовых редакторов. Для внесения в строку каких-либо изменений программа "ED" удаляет, вставляет и изменяет отдельные символы или слова, а не требует перепечатывать строку полностью.

Программа "ED" предоставляет пользователю следующие возможности:

- Создать новый файл.
- Открыть существующий файл и добавить в буфер редактирования взятую из него одну, все или несколько строк.
- Временно открыть для редактирования вспомогательный файл, а после завершения вернуться к основному файлу.
- Переместить указатель текущего символа (то место в тексте, куда может быть вставлен новый текст и начиная с которого можно просматривать, копировать или удалять существующий текст).
- Указатель текущего символа может быть установлен в начало (первая строка) или в конец (последняя строка) редактируемого файла. Кроме того, он может быть установлен на любую заданную строку, или на любое слово, или на любой символ в заданной строке.
- Просмотреть отдельные (или все) строки редактируемого файла, которые находятся в буфере редактирования.
- Вставить текст или в начало указанной строки, или перед определенным словом, или символом в указанной строке.
- Указать, что все вновь вводимые символы должны быть представлены в кодах верхнего регистра, или же допустить представление символов в кодах верхнего и нижнего регистров зависимости от того, как они были введены с клавиатуры консоли.
- Удалить указанный символ или последовательность символов, слово

или последовательность слов в заданной строке. Допускается также удаление целой строки или группы строк.

- Найти заданную строку символов в редактируемом файле и установить курсор на начало найденной строки.
Найти следующее вхождение ранее определенной строки символов.
- Заменить одно или несколько вхождений указанной строки символов на заданную строку.
- Включить в редактируемый файл одну или несколько строк из другого файла.
- Скопировать одну или несколько строк внутри того же файла.
- Скопировать указанное число строк из основного файла во временный вспомогательный файл. Содержимое этого временного файла можно добавить в другой вспомогательный файл.
- Переместить одну или несколько строк внутри того же файла.
- Определить простой "макрос", который обеспечит выполнение заданной команды указанное число раз.
- Задержать выполнение команды на заданное время при помощи функции "Sleep".
- Определить общий размер и количество свободных байт в буфере редактирования.
- Завершить выполнение программы и сохранить отредактированный текст в выходном файле. Или завершить выполнение программы без сохранения результатов редактирования (при этом файл-оригинал не уничтожается).

Программа "ED" предоставляет различные способы использования перечисленных возможностей.

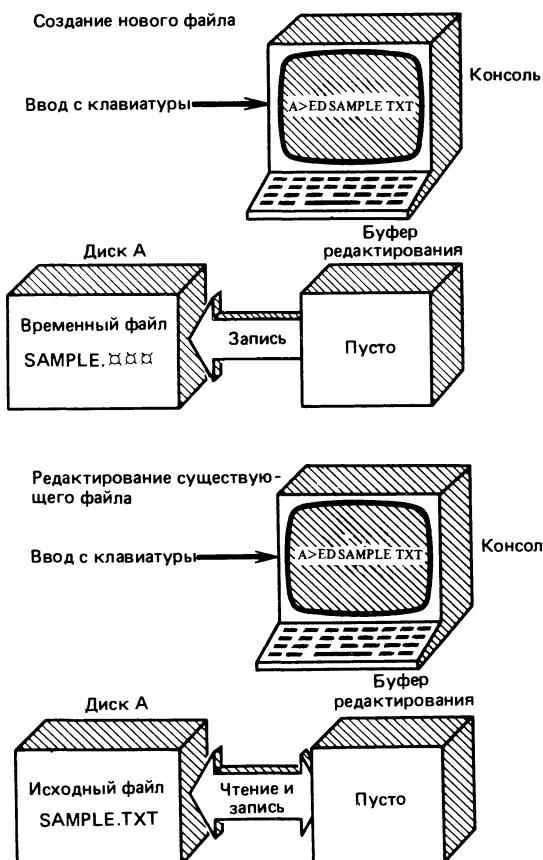
В следующем разделе даны основные принципы функционирования программы "ED", а потом детально описана каждая функция программы "ED".

АЛГОРИТМ РАБОТЫ

При обработке текстовых файлов и с помощью программы "ED" важно хорошо понимать основные принципы ее работы в среде операционной системы *CP/M*. Для начала в одно из дисковых устройств системы следует вставить диск, на котором хранится файл "ED.COM". Лучше, чтобы этот файл был расположен на системном диске (диске А). Текстовый файл, который требуется создать или отредактировать, можно расположить или на том же самом диске, или на другом. Для вызова программы "ED" на исполнение необходимо в качестве ответа на промпт с именем текущего диска (*A >*, *B >* и т. д.) ввести команду "ED" и через пробел имя редактируемого файла. Если указанный файл новый (т. е. еще не хранится на диске), то программа "ED" создаст его. Если файл уже существует, то он будет открыт, т. е. переведен в такое состояние, при котором возможен доступ к его содержимому. С помощью программы "ED" можно обрабатывать только те файлы, которые содержат текст, пригодный к выводу на консоль. В большинстве случаев имена текстовых файлов имеют расширение ".TXT", ".DOC" или какое-либо другое, принятое для обозначения текстовых файлов. Попытка открыть на редактирование файл с машинным или объектным кодом (например, ".COM"-файл) приведет к ошибке.

После того как программа "ED" вызвана на исполнение и загружена в оперативную память, для нужд редактирования в памяти ЭВМ выделяется специальная область, которая называется "буфер редактирования". Буфер является тем местом, где программа "ED" будет выполнять все требуемые функции редактирования. Если, например, создается новый файл, то весь его текст будет внесен в буфер. Если редактируется существующий файл, то программе "ED" можно дать команду загрузить в буфер все или часть строк текста этого файла. После того как все действия с текстом завершены, может потребоваться создать новую версию редактируемого файла. В этот момент программе "ED" можно дать команду переписать содержимое буфера в указанный файл и передать управление ОС *CP/M*.

Следующие две иллюстрации помогут лучше понять среду, в которой работает программа "ED".



Инициализация программы "ED". Для того чтобы отредактировать существующий файл или создать новый, введите имя этого файла вслед за командой "ED" в ответ на промпт с именем текущего диска (A >, B > и т. д.). Ес-

ли желательно создать файл и хранить его не на том же диске, где хранится файл "ED.COM", а на каком-либо другом, то перед именем файла необходимо указать имя этого диска ("B", "C" и т. д.). Это необходимо сделать также в том случае, когда редактируемый файл уже существует и расположен на другом диске.

Два примера правильно построенных командных строк приведены ниже. Здесь также представлены сообщения о состоянии редактируемого файла и буфера, которые выдаются на консоль.

Редактирование файла. После того как командная строка с именем редактируемого файла введена и нажата клавиша "RETURN", происходит загрузка программы "ED" и под буфер редактирования выделяется требуемая область оперативной памяти. Программа "ED" просматривает диск (текущий или указанный) с тем, чтобы определить, имеется ли на нем файл с указанным именем. Если нет, то на консоль выдается сообщение "NEW FILE" и промпт ".*" программы "ED". Если файл существует, то на консоль выдается только промпт программы "ED". Ниже проиллюстрированы оба случая:

A>:СОЗДАНИЕ НОВОГО ФАЙЛА

```
A>ED SAMPLE.TXT  
NEW FILE  
.*
```

A>:РЕДАКТИРОВАНИЕ СУЩЕСТВУЮЩЕГО ФАЙЛА

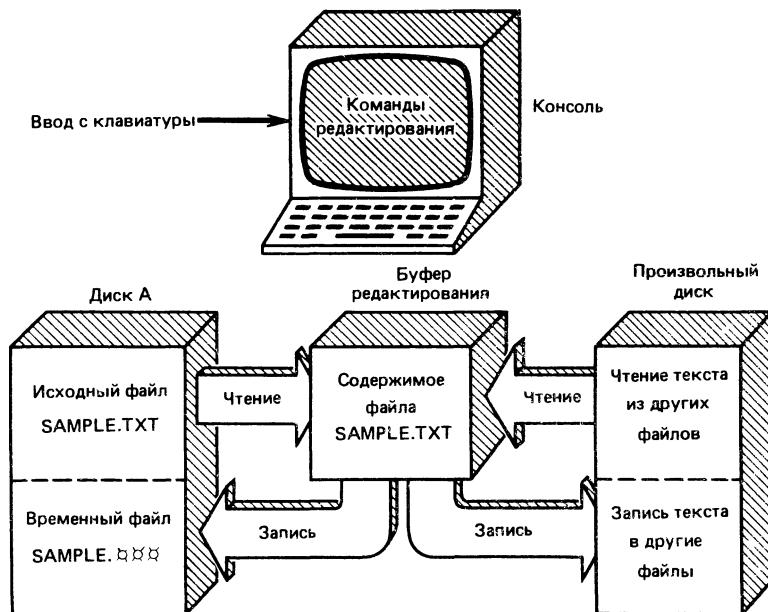
```
A>ED SAMPLE.TXT  
.*
```

Появление промпта программы "ED" означает, что программа загружена в оперативную память и что буфер редактирования готов для обработки текста. Если создается новый файл, то в ответ на промпт программы "ED" необходимо ввести команду "I" (Вставить) (команда "I" детально описывается ниже), что позволит ввести текст в буфер редактирования, содержимое которого позже можно записать в указанный файл. При редактировании существующего файла в ответ на промпт программы "ED" необходимо вводить команды "A" (Добавить). По этой команде указанное число строк редактируемого файла будет скопировано в буфер. До тех пор, пока текст находится в буфере, его можно обрабатывать при помощи различных команд программы "ED". В этом "командном" режиме в буфер может быть добавлен текст из любого файла по выбору и присоединен к тексту входного файла. Можно, кроме того, указать, что ряд строк текста нужно скопировать из буфера в отдельный временный файл. В процессе редактирований текста содержимое исходного файла не изменяется. Программа "ED" создаст временный результирующий файл с расширением ".QQQ", который будет использован для хранения окончательно отредактированного текста.

Основные принципы работы программы "ED" проиллюстрированы следующим рисунком.

Завершение работы программы "ED" и сохранение результатов редактирования. Как только все необходимые операции над буфером выполнены, работу программы "ED" обычно завершают по команде E (Выход). По этой команде программа "ED" перепишет содержимое буфера в указанный файл. При

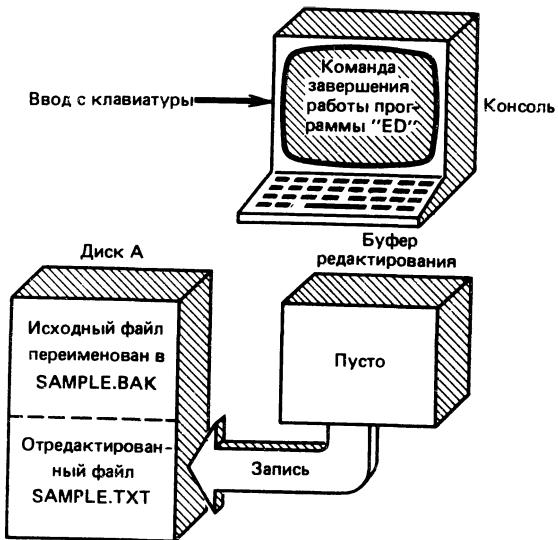
этом исходный файл (если таковой существовал ранее) переименовывается в файл с тем же именем и расширением ".BAK", которое означает, что файл содержит *резервную копию* основного файла:



Если решено не сохранять содержимое буфера, то введите команду выхода Q без сохранения результатов. По этой команде временный результирующий файл будет уничтожен, а исходный файл будет оставлен нетронутым.

После завершения работы программы "ED" по команде Е или Q на консоли снова появится основной промпт ОС CP/M с именем текущего диска. С этого момента можно использовать остальные команды ОС CP/M, такие как DIR или TYPE, с тем, чтобы проверить, например, наличие отредактированного файла на диске.

После более близкого знакомства с командами программы "ED", надписи на рисунках станут более понятными. Если программа "ED" используется впервые, то лучше всего создать небольшой файл и проэкспериментировать на нем со всеми описанными здесь командами. Сначала описываются наиболее часто используемые команды, а затем те, которые используются реже:



УКАЗАТЕЛЬ ТЕКУЩЕГО СИМВОЛА

В программе "ED" имеется уникальное средство, называемое "указатель текущего символа (CP)"*. Оно позволяет пользователю перемещаться в конкретные позиции в любом столбце на любой строке текста. В большинстве книг по ОС *CP/M* (включая эту), а также почти во всех руководствах указатель текущего символа обозначается как "CP".

Для пользователя указатель CP не виден. Это — "точка ссылки", которую программа "ED" использует для определения строки текста, откуда вступают в силу последующие команды. При помощи команд программы "ED" позицию указателя CP можно изменять. При обработке последующих команд программа "ED" использует текущее положение указателя CP до тех пор, пока оно не будет изменено при помощи другой введенной команды. Использование указателя CP позволяет вставлять, изменять или удалять определенные символы строки. Другие строчноориентированные текстовые редакторы для внесения каких-либо изменений требуют перепечатывания всей строки текста полностью.

После запуска программы "ED" указатель CP устанавливается на первую позицию первой строки в буфере редактирования. При добавлении строк текста в буфер редактирования или вывода на консоль указатель CP всегда перемещается на позицию первого символа текущей строки. Под первым символом строки подразумевается самый левый символ в этой строке. Указатель CP всегда будет перемещаться подобным образом независимо от того, предписывает ли заданная команда переместить его в какое-либо другое место. В следующем ниже примере иллюстрируется последовательность следующих действий: печать очередной строки и перемещение указателя CP впе-

* От английского Character Pointer. — Прим. перев.

ред на 16 символов. Здесь символ "^^" обозначает позицию указателя СР. На экране дисплея его видно не будет:

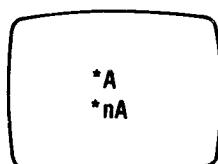
```
1. * FIRST LINE OF TEXT
1. * ^
2. * SECOND LINE OF TEXT.
2. * ^
2. * SECOND LINE OF TEXT.
2. * +16C ^ <---- Переместить указатель СР
вперед на 16 символов
```

```
2. * SECOND LINE OF TEXT.
2. * INEW ^Z ^ <---- Вставить "NEW " перед символом,
на который указывает СР
2. * SECOND LINE OF NEW TEXT
2. * -3C ^ <---- Переместить указатель СР
назад на 3 символа
```

Существует несколько команд программы "ED", которые используются для перемещения указателя СР. Эти команды обеспечивают перемещение указателя СР назад или вперед вдоль строки, переход на другую строку текста, а также в начало или в конец буфера. В этой главе представлено детальное описание каждой отдельной команды программы "ED". Команды, которые изменяют позицию указателя СР или же зависят от нее, описываются более детально.

ОПИСАНИЕ КОМАНД ПРОГРАММЫ "ED"

ДОБАВЛЕНИЕ В БУФЕР УКАЗАННОГО ЧИСЛА
СТРОК ИЗ ИСХОДНОГО ФАЙЛА



*A
*nA

Если в качестве параметра команды "ED" указано имя существующего файла, то содержимое этого файла нельзя вывести на консоль до тех пор, пока оно не будет скопировано в буфер редактирования. Как только текст окажется в буфере, его можно выводить на консоль и изменять. Для загрузки содержимого существующего файла в буфер используется команда "A" (от английского Append – добавить). Данная команда вводится в ответ на промпт программы "ED" и состоит из числа и символа "A", после которого необходимо нажать клавишу "RETURN". Число, предшествующее символу "A", определяет количество строк, которое необходимо переписать в буфер. Здесь может быть указано любое число от 1 до 65,535.

В следующем примере показано, как первая строка исходного файла может быть добавлена в буфер. Команда может иметь вид "1A" или просто "A", поскольку число "1" всегда принимается по умолчанию, когда число или символ отсутствует:

```
A>ED SAMPLE.TXT
: *1A
1: * THIS IS THE FIRST LINE OF TEXT
1: *EJ
```

```
A>ED SAMPLE.TXT
: *A
1: * THIS IS THE FIRST LINE OF TEXT
1: *[J]
```

В предыдущих примерах иллюстрируется загрузка в буфер только одной строки исходного файла. Отметим, что здесь символы "[]" обозначают позицию курсора на экране дисплея.

Когда команда загрузки ("A") является первой при редактировании, процесс загрузки в буфер редактирования всегда начинается с первой строки текста в исходном файле. После загрузки строки в буфер редактирования, ей присваивается порядковый номер, который затем добавляется к промпту программы "ED". При нажатии клавиши RETURN на консоль после промпта программы "ED" выдается содержимое строки с этим номером. Кроме того, курсор будет установлен на следующей строке, которая начинается с такого же промпта и номера строки. С этого момента можно вводить любые команды, связанные с обработкой строки под номером 1. Точно так же из исходного файла можно загрузить в буфер редактирования текст, следующий за первой строкой. После того как команды загрузки выполнены, можно нажать клавишу "RETURN" с тем, чтобы вывести на консоль содержимое строк, загруженных в буфер. В случае последовательного ввода команд загрузки программа "ED" запоминает, какие строки уже были загружены в буфер. Например, если команда 2A введена после команды 1A, то программа "ED" загрузит в буфер вслед за первой следующие две строки, т. е. вторую и третью строки:

```
: * 1A
1: * THIS IS THE FIRST LINE OF TEXT.
1: * <CR>
: * 2A
2: * THIS IS THE SECOND LINE OF TEXT.
2: * <CR>
3: * THIS IS THE THIRD LINE OF TEXT.
3: * <CR>
: *
```

В процессе редактирования можно использовать любые комбинации команд загрузки. Например, можно загрузить в буфер несколько строк текста из исходного файла, а затем вставить после последней строки текст, введенный с консоли. Более того, команды загрузки размещают оставшийся текст *после* вставленного текста. Программа "ED" автоматически нумерует строки текста в порядке введения в буфер редактирования независимо от того, загружены они или вставлены вручную.

ДОБАВЛЕНИЕ ВСЕГО ТЕКСТА ИСХОДНОГО ФАЙЛА ИЛИ
ЕГО ЧАСТЕЙ ДО ТОГО, КАК БУФЕР БУДЕТ ЗАПОЛНЕН
НАПОЛОВИНУ ИЛИ ДО КОНЦА

*0A
*#A

Обычно при загрузке строк текста из исходного файла бывает желательно загрузить либо большую их часть, либо все строки, либо столько, сколько позволяют размеры буфера. Программа "ED" может загрузить в буфер максимум 65,535 строк. Часто такая потребность возникает в тех случаях, когда желательно внести изменения в различных местах по всему тексту. Разместив текст в буфере редактирования полностью, можно получить доступ ко всем строкам.

В качестве параметра команды загрузки вместо конкретного числа строк может быть использован символ "0" (ноль), т. е. команда загрузки строк в буфер редактирования имеет вид "0A". Если задана команда 0A, то строки исходного файла будут последовательно загружены в буфер до тех пор, пока либо все строки не будут загружены в буфер, либо пока буфер не заполнится наполовину. В противоположность этому использование символа "#" (т. е. команды "#A") вызовет загрузку в буфер редактирования *всех* строк исходного файла. В данном случае процесс загрузки продолжается до тех пор, пока не будут скопированы все строки, либо пока не будет достигнута строка с номером 65,535, либо пока буфер не переполнится. Размер буфера зависит от объема оперативной памяти ЭВМ. Общий объем определяется количеством символов (байтов), которые могут быть загружены в буфер из исходного файла или введены с консоли вручную.

В следующем ниже примере показаны результаты выполнения команды "#A"; при многократном нажатии клавиши RETURN произведена проверка того, что текст из исходного файла был загружен в буфер полностью:

```
: * #A
1: * THIS IS THE FIRST LINE OF TEXT.
1: * <CR>
!
V
500: * THIS IS THE LAST LINE OF TEXT.
500: * <CR>
: *
```

В этом примере для просмотра каждой строки текста пользователь последовательно нажимает клавишу RETURN (<cr>). После того как весь необходимый текст загружен в буфер, можно использовать любые другие команды программы "ED" с тем, чтобы удалить ненужный текст, внести изменения и т. п. Команду загрузки с параметром, указывающим количество загружаемых строк, можно использовать в сочетании почти со всеми другими командами программы "ED". Программа "ED" всегда сохраняет информацию о том, какие строки уже загружены в буфер редактирования. Таким образом, при обработке последующих команд загрузки программа "ED" перепишет в буфер только те строки, которые не были загружены ранее.

ВСТАВКА ТЕКСТА В БУФЕР

*I
*Text
*Text ^ Z

При создании нового файла с помощью программы "ED" желательно использовать команду, которая обеспечила бы вставку текста в буфер редактирования между строками и между столбцами. Такой командой является команда I (вставить), которая в любой момент времени может быть введена в ответ на промпт программы "ED". Обычно команда вставки вводится в виде одного символа "I", после которого сразу нажимается клавиша "RETURN" (в общем случае команда вставки может быть задана как в виде одного символа, так и в сочетании с короткой строкой текста, которую необходимо вставить). Когда команда вставки задана без того текста, который необходимо ввести в буфер, на консоли появится видоизмененный промпт программы "ED", в котором обычная пара символов ":" заменена одиночным символом ".:", что означает переход в режим вставки текста. С этого момента их до ввода команды CTRL/Z (что достигается одновременным нажатием клавиш CTRL и Z) любой текст, набираемый на клавиатуре консоли, вводится в буфер редактирования. По команде CTRL/Z программа "ED" весь текст, который был введен после команды I, рассматривает как текст, который необходимо "вставить" в буфер редактирования; после выполнения команды CTRL/Z снова появится обычный промпт программы "ED".

В следующем ниже примере (рис. 10.13) иллюстрируется использование команды I для вставки текста в буфер при создании нового файла. Здесь символы "^Z" обозначают команду CTRL/Z:

```
A>ED SAMPLE.TXT
NEW FILE
: *I
1: THIS IS THE POINT FROM WHICH NEW TEXT IS ENTERED,
2: WHEN CREATING A NEW FILE. AS NEW LINES OF TEXT ARE
3: INSERTED, ED WILL AUTOMATICALLY ASSIGN LINE NUMBERS
4: FOR YOU. A "LINE" WITH A "LINE NUMBER" IS A LINE OF
5: TEXT WHICH IS TERMINATED BY PRESSING THE RETURN KEY.
6: THE RETURN KEY WILL CAUSE THE ASCII CARRIAGE RETURN
7: (CR) AND LINE FEED (LF) CHARACTERS TO BE INSERTED IN
8: THE EDITING BUFFER AS THE FINAL CHARACTERS OF THE LINE.
9: THE CR/LF CHARACTERS WILL BE INVISIBLE WHEN A LINE OF
10: TEXT IS DISPLAYED ON THE SCREEN.
11: "."
: *
```

Как видно, программа "ED" автоматически присваивает порядковые номера строкам вставляемого текста.

Каждое нажатие клавиши RETURN программа "ED" воспринимает как символы CR (возврат каретки) и LF (перевод строки), представленных в коде ASCII. Эти два символа (и другие команды программы "ED") можно использовать для перемещения курсора и указателя CP (указатель текущего символа) на первую позицию очередной строки в буфере редактирования; строке при этом присваивается новый номер. Таким способом текст можно вставлять до ввода команды CTRL/Z. По команде CTRL/Z работа в режиме вставки завершается и снова выдается обычный промпт программы "ED".

(":*"). С этого момента указатель СР устанавливается на первую позицию строки *непосредственно за последней строкой введенного текста*.

В режиме вставки большинство управляющих символов команд редактирования ОС *CP/M* (включая [^]X, [^]U, [^]Y и т. д., описанные в гл. 2) выполняются обычным образом. Эти команды позволяют исправлять введенный текст (кроме того, команду вставки и ее различные виды можно использовать в сочетании практически со всеми другими командами программы "ED"). Ниже представлено описание команд, которым можно пользоваться в режиме вставки.

CTRL/H или BACKSPACE. По этой команде производится удаление символа, расположенного слева от курсора.

RUBOUT или DELETE. По этой команде производится удаление символа, расположенного слева от курсора. Удаленный символ выводится на консоль (вывод производится только в стандартной ОС *CP/M*; некоторые поставщики ОС *CP/M* придают командам RUBOUT и DECETE функции команд CTRL/H и BACKSPACE).

CTRL/M или RETURN. По этой команде в текст вставляются символы CR (возврат каретки) и LF (перевод строки). Эти символы обозначают конец текущей строки и начало следующей.

CTRL/R. По этой команде производится повторный вывод текущей строки тем же способом, что и вывод промпта ОС *CP/M*. Эта команда доступна только в ОС *CP/M* версии 1.4 и более поздних.

CTRL/U и CTRL/X. Каждую из этих команд можно использовать для удаления последней введенной строки. Обе эти команды выполняются так же, как в основном командном режиме ОС *CP/M*.

CTRL/L. Эту команду можно использовать для вставки в текст символа перевод формата (символ "FF" в коде ASCII). Символ перевода формата при печати файла вызывает переход на новую страницу.

В режиме вставки можно использовать практически все другие управляющие последовательности кода ASCII. Другие, отличные от описанных выше, управляющие последовательности используются редко. Однако полезно знать, какие управляющие последовательности при необходимости можно вводить.

Кроме того, команду I можно использовать для вставки в текст одной конкретной строки. Для этого вставляемую строку вводят после команды I. При вводе команды программа "ED" рассматривает весь текст, заключенный между командами I и RETURN, как отдельную новую строку текста. Цель этой команды быстро вставить текст в буфер редактирования. Весь текст, введенный после команды I, вставляется в текущую строку сразу после символа, на который установлен указатель СР. Поэтому полезно убедиться, что печатаемый текст по длине не превышает размер строки консоли.

В следующем примере этот формат команды вставки (т. е. текст расположен сразу после команды I) сравнивается со стандартным (вставка текста завершается вводом команды CTRL/Z, как описано ранее). Здесь символы "[^]Z" обозначают команду CTRL/Z:

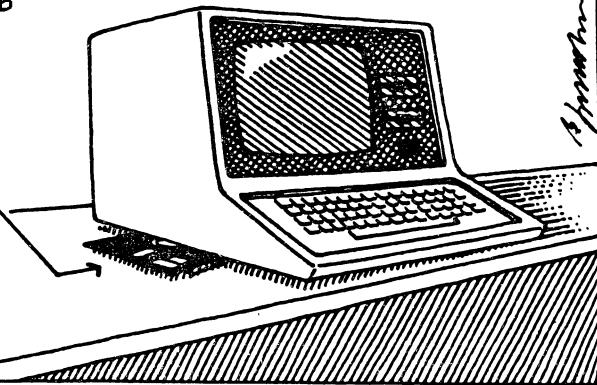
```
: *I
1: THIS IS THE FIRST LINE OF TEXT.
2: ^Z
2: *IT THIS IS THE SECOND LINE OF TEXT.
2: *
```

Как только вставка текста закончена, можно вернуться в начало буфера редактирования и, многократно нажимая клавишу "RETURN", вывести для проверки вновь введенные строки текста:

```
1: *<CR>
1: * THIS IS THE FIRST LINE OF TEXT.
1: *<CR>
2: * THIS IS THE SECOND LINE OF TEXT.
2: *
```

СОВРЕМЕННЫЙ ТЕРМИНАЛ

ИМЕЕТСЯ
ВОЗМОЖНОСТЬ
РАБОТАТЬ С
ГИБКИМИ
ДИСКАМИ
РАЗЛИЧНЫХ
ТИПОВ



Место в буфере редактирования, куда вставляется текст, зависит от позиции указателя СР. В начале сеанса редактирования указатель СР автоматически устанавливается на первый символ первой строки в буфере редактирования. При нажатии клавиши "RETURN", вызывающем переход на следующую строку, указатель СР перемещается и устанавливается на первую позицию текущей строки. Поскольку команда I связана с позицией указателя СР, текст можно вставлять, начиная почти с любой позиции строки. Используя определенные команды программы "ED", смещающие указатель СР, можно вставлять один или несколько символов или слов в любую указанную позицию.

Существует еще один формат команды I, при использовании которого текст вводится сразу после команды I и завершается командой CTRL/Z. Этот формат полезно использовать при вставке текста, когда указатель СР перемещается вдоль строки. Например, пропущено слово в данной строке текста. Для исправления ошибки указатель СР устанавливают на позицию, где требуется вставить новое слово. Затем вводят команду "Itext ^Z"*. Текст, рас-

* Здесь "text" обозначает текст, который необходимо вставить. – Прим. перев.

положенный после символа, на который установлен указатель СР, смещается вправо, освобождая место для новых символов. Сразу после выполнения команды ED восстановится обычный режим работы. Эта процедура иллюстрируется примером

```
1: * THIS IS THE FIRST LINE OF TEXT.  
1: *
```

По команде "С" указатель СР перемещается на 26 символов вправо:

```
1: * THIS IS THE FIRST LINE OF TEXT.  
1: *
```

Теперь указатель СР установлен на позицию символа "t":

```
1: * THIS IS THE FIRST LINE OF TEXT.  
1: * INEW ^Z
```

Вставьте "new" на позицию СР и вернитесь к обычному режиму:

```
1: * THIS IS THE FIRST LINE OF NEW TEXT.  
1: *
```

Указатель СР по-прежнему установлен на 26-й позиции символа.

Как только новый текст записан с помощью команды I, в буфер редактирования всегда полезно вернуться к вставленной или измененной строке и вывести ее с целью проверки правильности произведенного действия. Такую проверку рекомендуется делать потому, что указатель СР не виден на консоли и очень легко ошибиться при определении его позиции.

УСТАНОВЛЕНИЕ ИЛИ ОТМЕНА РЕЖИМА ПЕРЕКОДИРОВКИ
СИМВОЛОВ В КОДЫ ВЕРХНЕГО РЕГИСТРА

*U
*-U

В некоторых случаях необходимо выполнение с помощью программы "ED" преобразования символов, введенных с консоли в нижнем регистре, в соответствующие символы верхнего регистра. Это иногда может потребоваться при редактировании файлов, содержащих исходные тексты программ на ассемблере, транслятор с которого требует представления всех символов только в кодах верхнего регистра. Обычно программа "ED" преобразования не выполняет. Все символы записываются в буфер редактирования в кодах верхнего и нижнего регистров, так как они введены. Подобным образом программа "ED" функционирует до получения команды U (от английского Upper case – верхний регистр). Для завершения преобразования символов в коды верхнего регистра можно использовать команду -U. После перевода символов в коды верхнего регистра обратное преобразование в коды нижнего регистра можно выполнить, перепечатав символы.

Следующий пример показывает результаты выполнения этой команды:

```
: *I
1: This is the first line of text.
2: ^Z
: #U
: *I
2: This is the second line of text.
3: ^Z
: #-U
: *I
3: This is the third line of text.
3: ^Z
: #B
: *<CR>
1: * This is the first line of text.
1: *<CR>
2: * THIS IS THE SECOND LINE OF TEXT.
2: *<CR>
3: * This is the third line of text.
3: *
```

В этом примере символы первой строки не перекодированы, поскольку команда U еще не была задана. Вторая строка введена после команды U, и поэтому все ее символы преобразованы в символы верхнего регистра, хотя символы были введены в верхнем и в нижнем регистрах. Перед вводом третьей строки была задана команда -U, и потому эта строка записана в буфер редактирования без каких-либо преобразований. Команда U используется относительно редко, поскольку большинство пользователей не желает иметь весь текст в верхнем регистре. При использовании команды U следует отчего-либо представлять себе ее назначение, поскольку иногда перевод символов в коды верхнего регистра включается просто по недоразумению.

ПЕРЕМЕЩЕНИЕ УКАЗАТЕЛЯ СР НА ОДИН ИЛИ НЕСКОЛЬКО СИМВОЛОВ ВПЕРЕД ИЛИ НАЗАД

*пС
*-пС

Полезно делать вставки в любом указанном месте строки, а не перепечатывать ее. Для этих целей служит указатель текущего символа (СР), определяющий место, куда вставляется текст. Для того чтобы вставить текст в определенную позицию строки, следует переместить указатель СР в эту позицию. Для перемещения указателя СР используется команда пС. Обычно указатель СР устанавливают на позицию первого символа текущей строки перед пС или перед командой вставки. Чтобы переместить указатель СР на определенное количество символов вправо, вводится команда пС, где п — количество позиций символов. Чтобы переместить указатель СР влево, вводится команда -пС.

Следующий пример показывает результаты выполнения данной команды:

```
1. * FIRST LINE OF TEXT
1. * ^
2. * SECOND LINE OF TEXT
2. * ^
```

```

2: * SECOND LINE OF TEXT.
2: * 16C          ^ <---- Переместить указатель СР
                  вперед на 16 символов

2: * SECOND LINE OF TEXT.
2: * -3C          ^ <---- Переместить указатель СР
                  назад на 3 символа

2: * SECOND LINE OF TEXT.
2: * <CR>          ^
3: * THIRD LINE OF TEXT.
3: * ^             <---- Теперь указатель СР установлен
                  на первый символ текущей строки

```

Команды nC или -nC эффективны только для текущей строки текста. Если для перехода к следующей строке нажата клавиша RETURN, указатель СР перемещается вместе с курсором и устанавливается на первую позицию новой строки. При последующем использовании команды nC указатель СР перемещается вдоль новой строки.

ПЕРЕМЕЩЕНИЕ УКАЗАТЕЛЯ СР НА ОДНУ ИЛИ НЕСКОЛЬКО СТРОК ВПЕРЕД ИЛИ НАЗАД

```

*NL
-*NL

```

Программа "ED" допускает перемещение указателя СР на заданное число строк вперед или назад. Для этого используется команда nL, где n – число строк, на которое с текущей строки перемещается указатель СР. Для перемещения указателя СР вперед по направлению к концу буфера редактирования используется команда nL. Для перемещения указателя СР назад по направлению к началу буфера редактирования используется команда -nL. После выполнения команды nL или -nL указатель СР устанавливается на первую позицию символа новой строки. С этого момента новая строка выдается на консоль, а курсор или головка печатающего механизма располагаются под этой строкой.

Если значение n в команде nL (или -nL) больше нуля, то указатель СР перемещается на n строк вперед или назад. Если n равно нулю (задана команда 0L), то указатель СР перемещается в начало текущей строки. Командой 0L удобно пользоваться, когда указатель СР находится по команде nC где-либо внутри текущей строки, а его желательно установить в начало строки.

В следующем ниже примере показаны различные варианты использования данной команды:

```

1: * FIRST LINE OF TEXT.
1: * 10L
10: * TENTH LINE OF TEXT.
10: * -5L
5: * FIFTH LINE OF TEXT.
5: * ^

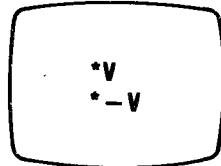
5: * FIFTH LINE OF TEXT.
5: * 10C

```

```
5: * FIFTH LINE OF TEXT.  
5: * 0L  
5: * FIFTH LINE OF TEXT.  
5: *
```

Эта команда используется часто, поскольку нередко необходимо перемещать указатель СР на несколько строк вперед или назад для просмотра текста и редактирования. В качестве параметра команды L (число n) можно указать любое число, однако номер той строки, на которую необходимо переместить указатель СР, не должен превышать общего количества строк, находящихся в данный момент в буфере редактирования. Если это число превышает номер последней строки в буфере, то указатель СР устанавливается в конце буфера редактирования, что отмечается промптом ":"* (промптом без номера строки). Достижение конца буфера редактирования, однако, не обязательно означает, что достигнут конец исходного файла.

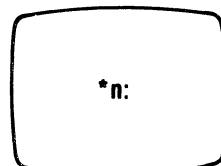
УСТАНОВЛЕНИЕ ИЛИ ОТМЕНА РЕЖИМА ВЫВОДА НОМЕРОВ СТРОК



```
*V  
*-V
```

Обычно программа "ED" назначает порядковые номера всем строкам, расположенным в буфере редактирования. Для запрещения вывода номеров строк можно использовать команду -V. Эта команда используется довольно редко, но в ряде случаев она может оказаться полезной. Пока печать номеров строк запрещена, все промпты программы "ED" выдают в обычном виде, но без номеров строк. Программа "ED" всегда контролирует номера строк независимо от того, печатаются они или нет. Предполагается, что номера строк распечатываются всегда. Программа "ED" является строчноориентированным текстовым редактором, и номера строк облегчают пользователю работу с текстом, расположенным в буфере редактирования. Если для запрещения вывода номеров строк использована команда -V, то для снятия запрета можно использовать команду V.

ПЕРЕМЕЩЕНИЕ УКАЗАТЕЛЯ СР В СТРОКУ С УКАЗАННЫМ НОМЕРОМ



```
*n:
```

Для перемещения указателя СР в строку буфера редактирования с заданным номером можно использовать команду n:. Эта команда подобна описанной команде nL за тем исключением, что указывается реальный номер строки. Для того чтобы использовать эту команду, необходимо ввести просто номер нужной строки и за ним двоеточие (":"). Как это делается, показано в следующем примере:

```
1: * THIS IS THE FIRST LINE.  
1: * 10:  
10: * THIS IS THE TENTH LINE.
```

```
10: * 100:  
100: * THIS IS THE HUNDREDTH LINE.  
100: * 50:  
50: * THIS IS THE FIFTIETH LINE.  
50: * 20:  
20: * THIS IS THE TWENTIETH LINE.  
20: * 1:  
1: * THIS IS THE FIRST LINE.  
1: *
```

По команде **n**: можно переместить указатель СР на любую строку в буфере редактирования, если известно, что строка с заданным номером существует.

ПЕРЕМЕЩЕНИЕ УКАЗАТЕЛЯ СР В НАЧАЛО ИЛИ
КОНЕЦ БУФЕРА

*B
*-B

Программа "ED" предоставляет в распоряжение пользователя команду, по которой указатель СР можно переместить (независимо от его текущей позиции) в начало или в конец буфера редактирования. Для перемещения указателя СР в начало буфера (т. е. на первую строку) используется команда **B**. Для перемещения указателя СР в конец буфера (т. е. на последнюю строку) используется команда **-B**.

В следующем примере представлены результаты выполнения обеих этих команд:

```
1: * THIS IS THE FIRST LINE.  
1: * 50:  
50: * THIS IS THE FIFTIETH LINE.  
50: * B  
1: * THIS IS THE FIRST LINE.  
1: * -B  
100: * THIS IS THE HUNDREDTH (LAST) LINE.  
100: * <CR>  
: * B  
1: * THIS IS THE FIRST LINE.  
1: *
```

ВЫВОД ТЕКСТА, РАСПОЛОЖЕННОГО ДО ИЛИ ПОСЛЕ
УКАЗАТЕЛЯ СР

*T
*OT
*OTT
*nT
*-nT
*#T
*-#T

Для просмотра текста буфера редактирования, можно использовать команду **T**. В зависимости от позиции указателя СР (указателя текущего символа) и объема выводимого текста данная команда используется в различных форматах. Каждый из этих форматов описывается в последующих разделах.

T (Вывод текста строки, расположенного в текущей строке после указателя СР).

Формат **T** является основным и используется для вывода *всех символов текущей строки, расположенных после позиции указателя СР*:

```
1: * THIS IS THE FIRST LINE.  
1: * 12C  
1: * T  
*FIRST LINE.  
1: *
```

Этот пример показывает, что указатель СР смещен на 12 символов вправо от начала текущей строки.

По команде Т в следующей строке выдаются все символы текущей строки, расположенной после новой позиции указателя СР. Текущая позиция указателя СР отмечается символом “*”.

0T (Вывод текста, расположенного в текущей строке до указателя СР).

По команде 0T выдаются все символы текущей строки, расположенные до позиции указателя СР. Это демонстрируется в следующем примере:

```
1: * THIS IS THE FIRST LINE.  
1: * 12C  
1: * 0T  
1: * THIS IS THE *(CR)  
1: *
```

Если текущей позицией указателя СР является начало строки, то текст не выводится. Если указатель СР смещен вправо от начала строки, то по команде 0T выдаются все символы до указателя СР, то есть слева от указателя СР. При этом курсор или головка печатающего механизма остается на той же строке до ввода следующей команды. Текущая позиция указателя СР отмечается символом “*”. В предыдущем примере после ввода символа “*” нажата клавиша “RETURN”. Однако с этого момента можно вводить любые команды программы “ED”. Например, можно задать команду вставки текста I) или описанную выше команду Т.

0TT (Вывод всех символов текущей строки). По команде “0TT” выдаются все символы текущей строки независимо от текущей позиции указателя СР. Команду 0TT полезно использовать в тех случаях, когда текущей строкой является первая строка буфера редактирования. Это та строка текста, которую программа “ED” обычно не выводит. Использование этой команды иллюстрируется следующим примером:

```
1: * 0TT  
1: * THIS IS THE FIRST LINE.  
1: *
```

nT (Вывод заданного числа строк, расположенных после указателя СР). По команде nT выводится заданное число строк, расположенных в буфере редактирования *после* текущей строки и позиции указателя СР. Указанное количество строк (n) вводят перед командой T. Можно вывести любое число строк, не превышающее число строк, оставшихся в буфере редактирования. После того как выведены все строки, курсор или головка печатающего механизма возвращается на исходную позицию указателя СР (исходную текущую строку). В следующем ниже примере демонстрируется использование этой команды:

```
1: * 5T  
1: * THIS IS THE FIRST LINE.  
2: * THIS IS THE SECOND LINE.  
3: * THIS IS THE THIRD LINE.
```

```
4: * THIS IS THE FOURTH LINE.  
5: * THIS IS THE FIFTH LINE.  
1: *
```

-пT (Вывод заданного числа строк, расположенных до указателя СР). Эта команда напоминает описанную выше команду за тем исключением, что указанные для вывода строки – это те, которые расположены до текущей строки и позиции указателя СР. При таком выводе строк перед командой пT указывают символ “–” (обозначающий обратное направление). После того как все строки выведены, курсор или головка печатающего механизма возвращаются на исходную строку и позицию указателя СР. Использование этой команды иллюстрируется в следующем примере:

```
6: * -5T  
1: * THIS IS THE FIRST LINE.  
2: * THIS IS THE SECOND LINE.  
3: * THIS IS THE THIRD LINE.  
4: * THIS IS THE FOURTH LINE.  
5: * THIS IS THE FIFTH LINE.  
6: *
```

#T (Вывод всех строк, расположенных после указателя СР). По этой команде выдаются все строки, расположенные в буфере после текущей позиции указателя СР. Команда состоит из символов “#” и символа “T”. Если указатель СР стоит на первой строке, то по этой команде выдается все содержимое буфера редактирования. Как только все строки выданы на консоль, курсор или головка печатающего механизма возвращается на исходную строку и позицию указателя СР.

-#T (Выдача всех строк текста, расположенных до указателя СР). Эта команда аналогична предыдущей за исключением того, что в данном случае выдаются все строки, предшествующие текущей позиции указателя СР. После того как все строки выданы, курсор или головка печатающего механизма возвращается на исходную строку и позицию указателя СР.

ПЕРЕМЕЩЕНИЕ УКАЗАТЕЛЯ СР НА ОДНУ ИЛИ
НЕСКОЛЬКО СТРАНИЦ И ВЫДАЧА ТЕКСТА

*пP
* – пP

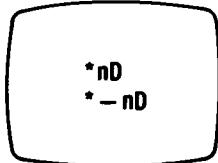
Другим способом просмотра расположенного в буфере текста является использование команды пP. По этой команде выдается так называемая “страница” текста, состоящая из фиксированного числа строк. По умолчанию это число равно 23, хотя в некоторых программах ОС CP/M, предназначенных для работы с видеотерминалами, у которых число строк на экране меньше, чем 24, это число равняется 15. Эту команду полезно использовать для вывода текста полными экранами (по 16 или 24 строки в зависимости от типа терминала). Команду вводят либо в формат Р для выдачи одной страницы после указателя СР, либо в формате пP, где п указывает число выводимых страниц после указателя СР. Кроме того, команду можно задать так, что на консоль будут выданы страницы, предшествующие текущей позиции указателя СР,

для чего перед символом "P" или "nP" вводят символ "-". После выполнения команды указатель СР перемещается в конец последней выданной страницы. Однако по команде OP (выдача страницы, расположенной после указателя СР) или -OP (выдача страницы, расположенной до указателя СР) можно выдать одну страницу текста, не изменяя при этом позиции указателя СР. В следующем примере показаны результаты выполнения команды P стандартной программой "ED" (поддерживающей 23 строки на странице) на видеотерминале с 24 горизонтальными строками (здесь символ "^" обозначает текущую позицию указателя СР):

```
1: * P
2: * THIS IS THE SECOND LINE.
3: * THIS IS THE THIRD LINE.
4: * THIS IS THE FOURTH LINE.
5: * THIS IS THE FIFTH LINE.
|
|
|
^
22: * THIS IS THE TWENTY SECOND LINE.
23: * THIS IS THE TWENTY THIRD LINE.
23: *
```

Может показаться, что эта команда не очень полезна, если используемый видеотерминал имеет, по меньшей мере, 24 строки. Однако эта команда может оказаться весьма полезной, если в качестве консоли вы используете термайп.

УДАЛЕНИЕ ОДНОГО ИЛИ НЕСКОЛЬКИХ СИМВОЛОВ,
РАСПОЛОЖЕННЫХ ДО ИЛИ ПОСЛЕ УКАЗАТЕЛЯ СР



*nD
* - nD

Команду удаления символов полезно использовать для внесения изменений в отдельные поля какой-либо строки текста. Для удаления одного или нескольких символов, расположенных до или после текущей позиции указателя СР (указатель текущего символа), используется команда nD. Для удаления символов, расположенных после указателя СР, используется команда D для одного символа или команда nD для нескольких символов, где n – количество символов. По этой команде программа "ED" отсчитает заданное количество символов (n), расположенных после текущей позиции указателя СР, и удалит их. Для удаления символов, указанных до текущей позиции указателя СР, перед командой D или nD вводят символ "-". В следующем примере показано, как сначала устанавливается указатель СР, а затем удаляется определенное число символов:

```
5: * THIS IS THE FFIFTH LINE.
5: * 16C
5: * T
*FFIFTH LINE.
5: * D
5: * OTT
```

```
5: * THIS IS THE FIFTH LINE.  
5: * OT  
5: * THIS IS THE * -4D  
5: * OTT  
5: * THIS IS THE FIFTH LINE.  
5: *
```

При пользовании командой nD обратите внимание на то, что признаком окончания строки и возврата каретки являются символы в коде ASCII : CR (возврат каретки) и LF (перевод строки), их следует рассматривать как два отдельных символа, хотя ни один из них не выводится на консоль.

УДАЛЕНИЕ ОДНОЙ ИЛИ НЕСКОЛЬКИХ СТРОК,
РАСПОЛОЖЕННЫХ ДО ИЛИ ПОСЛЕ УКАЗАТЕЛЯ СР

*nK
*-nK

Эта команда используется для удаления строк текста из буфера редактирования. По команде "K" будут удалены все символы текущей строки, расположенные после указателя СР (указатель текущего символа). Если указатель СР установлен на позиции первого символа строки, то будут удалены все символы данной строки. Если указатель СР установлен за первым символом строки, то все символы, расположенные после указателя СР, будут удалены, а символы, расположенные до указателя СР, сохраняются. Для удаления всех символов, расположенных до текущей позиции указателя СР, можно использовать команду -K. Для удаления нескольких строк из буфера редактирования перед командой K необходимо указать количество удаляемых строк (nK). Отсчет удаляемых строк производится, начиная с текущей строки и позиции указателя СР. По команде nK будут удалены n строк, расположенных после текущей позиции указателя СР, а по команде -nK – n строк, расположенные до текущей позиции указателя СР. Ниже приведены примеры использования этой команды:

```
50: * ORIGINAL LINE 50.  
50: * K  
50: * <CR>  
50: * ORIGINAL LINE 51.  
50: *
```

```
1: * ORIGINAL LINE 1.  
1: * <CR>  
2: * ORIGINAL LINE 2.  
2: * -K  
1: * OTT  
1: * ORIGINAL LINE 2.  
1: *
```

Во втором примере показаны результаты использования команды -K для удаления строки, предшествующей той, на которую установлены курсор и указатель СР.

ПОИСК СТРОКИ ТЕКСТА В БУФЕРЕ РЕДАКТИРОВАНИЯ (ОДНО- И МНОГОКРАТНЫЙ)

*Ftext^Z
*nFtext^Z

Для поиска в буфере редактирования определенной строки текста используется команда F (от английского Find – искать). Команда вводится в формате "Ftext^Z", где "text" представляет собой ту строку текста, которую необходимо найти, а команда ^Z (CTRL/Z) является признаком конца строки. Как только после этой команды нажата клавиша RETURN, программа "ED" начнет просматривать буфер от текущей позиции указателя CP (указатель текущего символа) до тех пор, пока заданная строка текста не будет найдена. После того как заданная строка будет найдена, указатель CP перемещается за нее*.

Поиск строк можно осуществлять как до, так и после позиции указателя CP. Если рассмотренной командной строке предшествует символ "—" ("—Ftext^Z"), то программа "ED" просмотрит текст от текущей позиции указателя CP по направлению к началу буфера. Если известно, что искомая строка встречается в тексте несколько раз, можно указать, какое по счету вхождение этой строки необходимо найти, для чего перед командой поиска указывают соответствующее число. Это число (n) указывает программе "ED" на необходимость найти n-тое вхождение заданной строки и переместить в соответствующую позицию указатель CP. Если в процессе поиска требуется включить символы конца строки CR LF (символы возврата каретки и перевода строки в коде ASCII), введите команду CTRL/L и в командной строке появится символ "^L". Использование этой команды иллюстрируется в следующих двух примерах (здесь символ "[^]" обозначает текущую позицию указателя CP):

```
1: * PREFERENCE DOCUMENT NUMBER 30XX40.Z
263: * FOR INFORMATION, REFERENCE DOCUMENT NUMBER 30XX40. THE
263: *
```

```
1: * 4F, Z
5: * SET N, X, Y, 0 AND THEN Z. USE THIS METHOD FOR FUTURE.
5: *
```

Отметим, что эта команда обеспечивает просмотр только текущего содержимого буфера редактирования. Если существует вероятность того, что искомая строка находится в той части исходного файла, которая еще не загружена в буфер редактирования, используйте команду N, которая описана в следующем разделе.

* Точнее, на позицию первого символа за найденной строкой. – Прим. перев.

ПОИСК СТРОКИ ТЕКСТА В БУФЕРЕ РЕДАКТИРОВАНИЯ (ОДНО- И МНОГОКРАТНЫЙ) И ВЫПОЛНЕНИЕ ПРИ НЕОБХОДИМОСТИ ОПЕРАЦИЙ ДОБАВЛЕНИЯ И ЗАПИСИ

*Ntext[~]Z
*nNtext[^]Z

Эта команда используется аналогично команде F. Однако при выполнении команды N происходит автоматическая загрузка очередной порции текста из исходного файла в буфер редактирования, если искомая строка не найдена в тексте, который уже загружен. По мере загрузки в буфер новых порций текста программа "ED", кроме того, переписывает просмотренный текст во временный выходной файл, что гарантирует наличие в буфере свободного пространства. Формат командной строки для этой команды такой же, как для команды F.

ПОИСК СТРОКИ ТЕКСТА В БУФЕРЕ РЕДАКТИРОВАНИЯ И ЗАМЕНА ЕЕ НА УКАЗАННЫЙ ТЕКСТ (ОДНО- И МНОГОКРАТНО)

*Sfind[~]Zreplace[^]Z
*nSfind[^]Zreplace[~]Z

Существует команда, которая напоминает команду F. Это – команда подстановки, по которой происходит поиск в тексте заданной строки и автоматическое замещение ее другой указанной строкой. Для того чтобы ввести эту команду, указывают символ "S" (от английского Substitute – подставить), строку, которую необходимо найти. В этой командной строке команда CTRL/Z (^Z) используется дважды. Первый раз – для того чтобы отделить "искомую" строку от строки "замещения" и второй раз в качестве признака конца всей команды. Символу "S" может предшествовать число (n). Оно указывает программе "ED" выполнить операцию подстановки заданное число раз.

Операции подстановки в тексте буфера редактирования можно выполнять до и после текущей позиции указателя CP (указатель текущего символа). После завершения операции подстановки указатель CP сразу перемещается на позицию, расположенную сразу за вновь вставленной строкой, в качестве символов конца каждой строки CR LF (символы возврата каретки и перевода строки в коде ASCII) необходимо использовать команду CTRL/L (^L).

В следующем примере представлены результаты выполнения команды подстановки (здесь символ "^" обозначает текущую позицию указателя CP):

```
30: * THIS IS THE THIRTY FIRST LINE.  
30: * STHIRTY FIRST^ZTHIRTIETH^Z  
30: * OTT  
30: * THIS IS THE THIRTIETH LINE.  
30: *  
  
50: *THE LOCATIONS ARE 300, 400, 500, AND 600.  
50: *450^Z^Z  
50: * OTT  
50: *THE LOCATIONS ARE 30H, 40H, 50H, AND 60H.  
50: *
```

При просмотре буфера редактирования в прямом направлении всегда убедитесь в том, что указатель СР установлен на позицию символа, расположенного *до* текста, который необходимо найти и заменить. При просмотре буфера в обратном направлении указатель СР должен быть установлен на позицию символа, стоящего *за* текстом, который необходимо найти.

ПОИСК СТРОКИ ТЕКСТА В БУФЕРЕ РЕДАКТИРОВАНИЯ И СЦЕПЛЕНИЕ ЕЕ С УКАЗАННЫМ ТЕКСТОМ (ОДНО- И МНОГОКРАТНО)

*Jfind ^Zinsert ^Zend ^Z
*nJfind ^Zinsert ^Zend ^Z

Команда J (от английского Juxtaposition – сцепление) является разновидностью команды подстановки (см. предыдущий раздел). Она позволяет осуществлять поиск в буфере редактирования (начиная с текущей строки и позиции указателя СР) заданной строки текста и добавлять к найденной строке указанный текст. Кроме того, можно удалять текст между двумя указанными строками.

Для ввода команды печатают символ "J", строку, которую необходимо найти ("find"), команду CTRL/Z ("^Z"), строку, которую необходимо вставить (наложить) в текст за найденной, еще одну команду CTRL/Z, строку, вплоть до которой удалить текст ("end"), завершающую команду CTRL/Z. Иными словами, командная строка имеет вид "Jfind^Zinsert^Zend^Z".

Если после вставки никакой текст не удаляется, то после "insert" в конце командной строки просто вводят две команды CTRL/Z (без текста между ними). Если вставку или сцепление текста необходимо выполнить несколько раз, то перед командой J должно быть указано число повторений (n). Отметим, однако, что этот счетчик повторений относится только к первому вхождению заданной строки. Для всех остальных команду J повторяют вручную.

Ниже представлены примеры использования этой команды:

```
40: * WHEN UNPLUGGING THE CABLE, BE EXTREMELY CAREFULL.  
40: * JCABLE, ^ZDO SO WITH CAUTION^Z.^Z  
40: * OTT  
40: * WHEN UNPLUGGING THE CABLE, DO SO WITH CAUTION.  
40: *
```

```
50: * 2T  
50: *  
51: * ! STEP ! ACTION ! VERIFICATION !  
52: * =====!=====+====+====!=!=====!  
52: * 8J!=====^Z=^Z=====!^Z  
52: * -3T  
50: *  
51: * ! STEP ! ACTION ! VERIFICATION !  
52: * =====!=====+====+====!=!=====!  
52: *
```

КОМБИНИРОВАНИЕ КОМАНД ПЕРЕМЕЩЕНИЯ УКАЗАТЕЛЯ СР

*#AB#T

Программа "ED" допускает комбинирование различных команд редактирования в одной командной строке. Например, если требуется загрузить содержимое исходного файла в буфер редактирования, вернуться в начало и распечатать содержимое, то следует ввести все необходимые для этого команды в одной командной строке #A (загрузка), В (перемещение указателя СР в начало буфера) и #T (печатать содержимое буфера).

В следующих примерах иллюстрируются два способа ввода этой последовательности команд:

```
: * #A
1: * B
1: * #T
1: * THIS IS THE FIRST LINE.
2: * THIS IS THE SECOND LINE.
!
!
v
72: * THIS IS THE SEVENTY SECOND (LAST) LINE.
72: *
```

```
: * #AB#T
1: * THIS IS THE FIRST LINE.
2: * THIS IS THE SECOND LINE.
!
!
v
72: * THIS IS THE SEVENTY SECOND (LAST) LINE.
72: *
```

Как видно из второго примера, команды следует вводить в том порядке, в котором требуется их выполнить. Не все команды программы "ED" можно использовать в сочетании. Применение некоторых команд в сочетании с другими может привести к нежелательным последствиям. Другое предостережение включает в себя определенные синтаксические правила, которым необходимо следовать. Обратите особое внимание на приведенные ниже правила, поскольку комбинации команд нередко становятся весьма длинными и сложными.

1. В ОС CP/M версии 1.4 и более поздних вместе можно группировать все команды программы "ED", кроме следующих:
 - Е (Завершение работы программы "ED" и сохранение содержимого буфера в результирующем файле).
 - Н (Сохранение содержимого буфера в результирующий файл без завершения работы программы "ED").
 - Q (Завершение работы программы "ED" без сохранения содержимого буфера).
 - О (Удаление содержимого буфера и результирующего файла без завершения работы программы "ED").

Любую из перечисленных команд можно вводить только в качестве коман-

ды командной строки. Это ограничение предотвращает случайное использование любой из этих команд в комбинации с другими командами. Эти команды оказывают существенное воздействие на работу программы "ED". Следовательно, их выполнение в первую очередь отразится на состоянии любого обрабатываемого файла.

2. Если в группу команд включаются команды, использующие в качестве параметров строки текста, то для обозначения конца строки следует вводить команду CTRL/Z ("^Z"). Следующие команды используют строки текста:

- I (Вставить текст в буфер редактирования).
- F (Найти текст в буфере редактирования).
- N (Найти текст в буфере редактирования и загрузить его, если необходимо).
- S (Заменить найденный в буфере текст указанным текстом).
- J (Сцепление текста с найденным и при необходимости удаление текста).

Как выглядит группа команд, содержащих строки текста, отражено в следующем примере:

```
40: * THE DICTIONARY SHOULD BE REFERENCED IF IN DOUT.  
40: * FDIS^Z-1DFDOUT^Z-1DBT^ZOTT  
40: * THE DICTIONARY SHOULD BE REFERENCED IF IN DOUBT.  
40: *
```

В предыдущем примере показано, как исправлены в целой строке текста все орфографические ошибки. Сначала был осуществлен поиск символов "dis". В результате этой операции указатель СР был установлен на позицию, расположенную справа от символа "s". Затем по команде -1D был удален символ "s" слева от позиции указателя СР. Потом была найдена последовательность символов "dout", а указатель СР установлен на позицию после символа "t". По команде -1D символ "t" был удален, и на его место вставлены символы "bt". Наконец, по команде 0TT исправленная строка полностью была выведена на контроль. Помните, что в конце каждой строки символов всегда следует вводить команду CTRL/Z. Клавиша "RETURN" используется только для завершения ввода группы команд.

МНОГОКРАТНОЕ ИСПОЛНЕНИЕ ГРУПП КОМАНД

*nM

При использовании групп команд (как описано выше) может потребоваться, чтобы определенная последовательность команд была повторена несколько раз. Для того чтобы задать это повторение, используется команда nM, где n есть число повторений последовательности команд. При использовании команды nM программа "ED" выполняет заданную команду над каждым вхождением (после текущей позиции указателя СР) специфицированной строки в буфере редактирования. Команда выполняется указанное число

раз. Параметр *n* должен иметь значение, большее 1, иначе команда будет последовательно повторяться до тех пор, пока не возникнет состояние ошибки. То же самое произойдет, если заданный параметр равен нулю.

Ниже приводится пример правильно построенной повторяющейся группы команд:

```
40: * THE DICTIONARY SHOULD BE REFERENCED IF IN DOUT. THE
40: * 5MFDIS^Z-1DFDOUT^Z-1DIBT^ZOTT
40: * END. THE DICTIONARY SHOULD BE REFERENCED IF IN DOUBT.
67: * THE DICTIONARY SHOULD BE REFERENCED IF IN DOUBT. IF THE
95: * CHANCE. THE DICTIONARY SHOULD BE REFERENCED IF IN DOUBT.
178: * THE DICTIONARY SHOULD BE REFERENCED IF IN DOUBT.
332: * LAST. THE DICTIONARY SHOULD BE REFERENCED IF IN DOUBT.
332: *
```

В этом примере группа команд повторялась 5 раз. Каждый раз обрабатывалось новое вхождение заданной строки текста.

ОПРЕДЕЛЕНИЕ РАЗМЕРОВ СВОБОДНОГО И ЗАНЯТОГО ОБЪЕМА В БУФЕРЕ РЕДАКТИРОВАНИЯ

*OV

В ОС *CP/M* версии 1.4 и более поздних команда *V* (см. выше) выполняет еще и специальную функцию. По команде *OV* на консоль выдается информация об использовании объема в буфере редактирования. Эта информация включает в себя общий объем памяти, отведенной под буфер редактирования, и объем памяти, находящейся в использовании. Оба эти значения представлены в байтах. Ниже приводится пример такого отчета:

```
: * OV
: * 21506/32183
: *
```

В примере количество уже использованных байтов напечатано в начале. После этого числа напечатан символ "/", а за ним — общее количество байтов, имеющихся в буфере. Вычитая объем использованной памяти (21 506 в данном примере) из общего объема буфера (32 183), можно получить объем свободного пространства в буфере (10 677). Количество байтов, которое можно отвести под буфер редактирования, определяется общим объемом оперативной памяти ЭВМ. Значение, представленное в данном примере, относится к системе с общим объемом памяти, равным 64К (64 килобайт или 64×1024 байт). Для систем с меньшим объемом памяти (например, 48К, 32К, 16К и т. д.) это значение будет меньше, хотя в системах с большей памятью (например, 128К) оно возможно больше в зависимости от того, как сгенерирована ОС *CP/M*.

*W
*nW
*0W
*#W

ЗАПИСЬ НЕСКОЛЬКИХ СТРОК ИЗ БУФЕРА РЕДАКТИРОВАНИЯ ВО ВРЕМЕННЫЙ РЕЗУЛЬТИРУЮЩИЙ ФАЙЛ

Команда W (от английского Write – записать) позволяет пересыпать строки текста из буфера во временный результирующий файл. По этой команде указанные строки текста "записываются" во временный результирующий файл вслед за строками, которые "записаны" самыми последними. После завершения этой операции записанные строки удаляются из буфера редактирования.

По этой команде можно записать несколько строк или все. Если команда W задана без параметров, то записывается только первая строка из буфера редактирования. Команде W может предшествовать число (nW). Если заданная команда 0W, то во временный файл записываются все строки из первой половины буфера редактирования. Если параметр n больше нуля, то записываются первые n строк из буфера редактирования. Если введена команда #W, то записываются все редактируемые строки. Запись всегда начинается с первой строки в буфере редактирования.

Использовать команду W полезно в тех случаях, когда текст из исходного файла загружается в буфер по команде A небольшими порциями (Загрузка). Если программа "ED" выполняется подобным образом, то для записи каждой такой порции текста во временный выходной файл может быть использована команда #W. После того, как текст отредактирован (т. е. строки удалены из буфера), в освободившуюся область можно загрузить очередную порцию. Команду W рекомендуется использовать только в подобных ситуациях. В противном случае использование команды W может привести к неправильным результатам.

Различные варианты использования этой команды продемонстрированы в следующем примере:

```
1: * 2T
1: * THIS IS THE FIRST LINE.
2: * THIS IS THE SECOND LINE.
1: * W
1: * 2T
1: * THIS IS THE SECOND LINE.
2: * THIS IS THE THIRD LINE.
1: *
```

В этом примере во временный выходной файл записана первая строка буфера. После того как строка записана, она удаляется из буфера редактирования, а все остальные строки, следующие за ней, сдвигаются к началу буфера. Номер каждой строки уменьшается на единицу (строка под номером 2 стала под номером 1 и т. д.).

В следующем примере демонстрируется запись во временный файл всех строк из буфера:

```
1: * 2T
1: * THIS IS THE FIRST LINE.
2: * THIS IS THE SECOND LINE.
```

1: * #W
: * \$T
: *

В данном примере демонстрируется запись всех строк из буфера по команде #W. После того как все строки записаны во временный файл, они удаляются из буфера. С этого момента буфер редактирования пуст, что отмечается промптом ":" без номера строки. Теперь по команде A можно загружать в буфер очередную порцию текста из исходного файла или вставлять текст из других временных файлов*. Здесь следует быть внимательным и не повторять без необходимости одни и те же операции загрузки и записи текста. Если это произойдет, то в результирующий файл попадут две одинаковые копии одного и того же текста**.

ЗАПИСЬ ГРУППЫ СТРОК ИЗ БУФЕРА РЕДАКТИРОВАНИЯ ВО ВРЕМЕННЫЙ БИБЛИОТЕЧНЫЙ ФАЙЛ

* nX

Помимо результирующего файла текст из буфера (начиная с текущей строки и позиции CR) можно записать в специальный временный файл, который называется "библиотечным" файлом. Это происходит подобно выполнению описанной выше команды W (Запись). Разница между этими двумя командами заключается в том, что для команды X следует указывать количество записываемых строк. Для этого введите строку nX, где n – количество строк.

По команде nX указанное число строк записывается в специальный временный файл с именем "ХОХОХОХОХ.LIB". Отметим, что строки, записанные во временный библиотечный файл, из буфера не удаляются. После того как этот файл создан, программа "ED" по команде R (от английского Read – читать) будет "читать" содержимое файла назад в буфер и размещать его сразу же после позиции, на которую установлен указатель CR. Временный файл "ХОХОХОХОХ.LIB" существует до тех пор, пока программа "ED" не завершит свое выполнение, тогда он будет удален автоматически. Этот файл, однако, не удаляется, если в процессе выполнения программы "ED" в системе происходит сбой и требуется полная перезагрузка. В этом случае можно повторить ввод команды ED, найти место, где работа была прервана, и вновь использовать файл "ХОХОХОХОХ.LIB".

При последующем использовании команды nX в течение того же сеанса редактирования предыдущее содержимое файла "ХОХОХОХОХ.LIB" удаляется. Любой новый текст заменит старый текст в файле "ХОХОХОХОХ.LIB".

В следующем ниже примере показано, как используется эта команда:

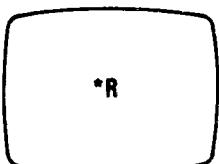
* Здесь авторы допускают неточность: загрузка текста из любого файла, кроме исходного, осуществляется при помощи команды R, описанной ниже. – Прим. перев.

** Неточность, дублирование текста возможно только при чтении его из вспомогательных файлов (см. описание команды R), поскольку при каждом выполнении команды A в буфер загружается очередная порция текста из исходного файла. – Прим. перев.

```
70: * 6T  
70: * SHOULDN'T BE TAKEN FOR GRANTED.  
71: *  
72: * THIS PORTION OF THE DOCUMENT WILL BE UPDATED WHEN THE  
73: * DOCUMENT IS REISSUED TO A LATER REVISION.  
74: *  
75: * SEVERAL ITEMS SHOULD BE TAKEN INTO CONSIDERATION WHEN  
70: * 71:  
71: * 4X  
71: *
```

Отметим, что в этом примере строки с 71 по 74 записаны в файл "ХØØØØØØ.LIB" по команде 4X. Эти четыре строки не удаляются из буфера, и, следовательно, нумерация всех строк остается неизменной. Теперь текст из файла "ХØØØØØØ.LIB" можно по команде R (читать) копировать в любое место в буфере редактирования (см. ниже).

ЧТЕНИЕ В БУФЕР РЕДАКТИРОВАНИЯ СОДЕРЖИМОГО ВРЕМЕННОГО ФАЙЛА, СОЗДАННОГО КОМАНДОЙ nX



Команда R (Читать) используется для копирования в буфер редактирования текста из файла "ХØØØØØØ.LIB", который создан по команде nX (см. описание предыдущей команды). Изменяя нужным образом позицию указателя CP в буфере редактирования, можно "переместить" любой текст из одного места в другое. Это осуществляется за счет использования комбинации команд nX и R. При применении команды R содержимое файла "ХØØØØØØ.LIB" не уничтожается. В силу этого за один и тот же сеанс редактирования команду R можно использовать необходимое число раз. Файл "ХØØØØØØ.LIB" не удаляется до обычного завершения работы программы "ED".

В следующем примере демонстрируется использование команды R; этот пример является продолжением предыдущего примера, иллюстрирующего использование команды nX:

```
71: * 85:  
85: * -21  
84: * TAKE CAREFUL NOTE OF HOW THIS PROCEDURE IS FOLLOWED.  
85: * IN FUTURE REVISIONS, THIS PROCEDURE WILL BE MODIFIED.  
85: * R  
85: * 84:  
84: * 6T  
84: * TAKE CAREFUL NOTE OF HOW THIS PROCEDURE IS FOLLOWED.  
85: *  
86: * THIS PORTION OF THE DOCUMENT WILL BE UPDATED WHEN THE  
87: * DOCUMENT IS REISSUED TO A LATER REVISION.  
88: *  
89: * IN FUTURE REVISIONS, THIS PROCEDURE WILL BE MODIFIED.  
84: *
```

В этом примере содержимое файла "ХØØØØØØ.LIB" прочитано в буфер и размещено *после* текущей позиции указателя CP в строке с номером 85. Поскольку указатель CP установлен на начало строки с номером 85, весь текст из файла "ХØØØØØØ.LIB" записан *перед* текстом этой строки.

В результате предыдущий текст строки с номером 85 смешен на 4 строки вниз, на строку с номером 89. Номера всех последующих строк в буфере увеличены на 4. Команду R можно использовать в различных местах в буфере редактирования необходимое число раз — содержимое файла "ХОХОХОХОХ.LIB" при этом не удаляется.

ЧТЕНИЕ В БУФЕР СОДЕРЖИМОГО БИБЛИОТЕЧНОГО ФАЙЛА

*Rfilename

По команде R в буфер редактирования можно читать содержимое указанного текстового файла. Имя этого файла при этом должно иметь расширение ".LIB". Если это условие выполнено, то для чтения содержимого файла в буфер редактирования достаточно просто ввести команду "R", за ней имя файла без расширения. Программа "ED" будет автоматически искать файл с указанным именем и расширением ".LIB" и, если такой файл будет найден, поместит копию его содержимого в буфер после текущей позиции указателя СР. Прежде чем применить эту команду, убедитесь, что указатель СР установлен в ту позицию, где требуется разместить копируемый файл.

В следующем примере показано, как используется эта команда:

```
1: * -B
: * -1L
67: * OTT
67: * THIS CONCLUDES CHAPTER2.
67: * (CR)
67: * RCHAPTER3
67: * 67:
67: * #T
67: * THIS CONCLUDES CHAPTER2.
68: *          ** CHAPTER 3: APPLICATIONS **
69: *
70: * THIS CHAPTER DESCRIBES THE VARIOUS APPLICATIONS OF THE
!
!
V
231: * THIS CONCLUDES CHAPTER 3.
67: *
```

В этом примере последней строкой текста в буфере редактирования является "THIS CONCLUDES CHAPTER 2". По команде -B указатель СР был установлен на последнюю строку. Затем с командой R был указан файл "CHAPTER 3.LIB". Это привело к тому, что содержимое данного файла было скопировано в буфер редактирования и размещено *после* позиции указателя СР. Наконец, указатель СР был перемещен в конец буфера редактирования. Выполнение этой команды не привело к изменению содержимого файла "CHAPTER 3.LIB".

СОХРАНЕНИЕ ОТРЕДАКТИРОВАННОГО ТЕКСТА В
РЕЗУЛЬТИРУЮЩЕМ ФАЙЛЕ БЕЗ ЗАВЕРШЕНИЯ
ПРОГРАММЫ "ED"

*Н

В процессе редактирования часто полезно иметь возможность сохранить в результирующем файле весь отредактированный текст. Как описано выше, программа "ED" переименовывает исходный файл с тем же именем и расширением ".BAK". При этом создается временный результирующий файл, который содержит текст для окончательного результирующего файла. Если для завершения работы программы "ED" используется команда E (от английского Exit – выход), то этот временный файл сохраняется на диске (описание команды E представлено ниже). При этом временному файлу присваивается имя исходного файла, которое указано при первом обращении к программе "ED". Команда H позволяет выполнить ту же функцию без завершения работы программы "ED". При использовании этой команды происходит следующее.

1. Весь текст, расположенный в буфере редактирования, пересыпается во временный результирующий файл ("filename.ØØØ"). Затем текст удаляется из буфера редактирования.
2. Любой оставшийся в исходном файле ("filename.ext") текст, который не был загружен в буфер, копируется в соответствующее место во временном результирующем файле ("filename.ØØØ").
3. Затем исходный, неизменный файл ("filename.ext") переименовывается в файл с тем же именем и расширением ".BAK" ("filename.BAK") и становится резервной копией основного файла.
4. Затем временный результирующий файл ("filename.ØØØ") переименовывается в исходный ("filename.ext") и становится новым исходным файлом.
5. Затем создается новый временный результирующий файл ("filename.ØØØ").

После выполнения указанных действий появится промпт программы "ED" без номера строки ("_:*"). Это означает, что буфер редактирования пуст и в него можно загружать текст. Теперь по команде "A" загрузка текста осуществляется из нового исходного файла ("filename.ext"). Любой текст, выводимый из буфера редактирования по команде W, поступит в новый временный результирующий файл ("filename.ØØØ").

При редактировании больших документов или при внесении в текст большого количества изменений хорошо выполнять эту команду через каждые 15 – 20 минут. В этом случае, если в системе произойдет сбой, большая часть выполненной работы сохранится на диске. Помните, что если при сбое системы требуется полная перезагрузка, то содержимое буфера будет стерто. Поэтому в системах, склонных к подобного рода сбоям, эту команду имеет смысл использовать чаще (систем, полностью защищенных от сбоев, не существует).

ЗАВЕРШЕНИЕ РАБОТЫ ПРОГРАММЫ "ED" С СОХРАНЕНИЕМ ОТРЕДАКТИРОВАННОГО ТЕКСТА В РЕЗУЛЬТИРУЮЩЕМ ФАЙЛЕ

*E

Для завершения работы программы "ED" и сохранения отредактированного текста в результирующем файле применяйте команду E (от английского Exit – выход). Эта команда предписывает программе "ED" выполнить все операции над дисковыми файлами так, как описано в предыдущем разделе для команды H, и восстановить режим команд ОС CP/M. Использование команды E для завершения работы программы "ED" является обычным делом, поскольку при этом все результаты редактирования сохраняются на диске. После ввода этой команды (ее можно ввести в виде строки в буфере редактирования) курсор или головку печатающего механизма смещают на одну строку вниз. Затем следует короткая пауза, в течение которой программа "ED" выполняет все операции чтения/записи с диском и буфером редактирования. После завершения этих операций появится основной промпт ОС CP/M, содержащий имя текущего диска, и восстановится обычный режим команд ОС CP/M.

Следующий пример демонстрирует, что происходит при выполнении этой команды:

45: * E

a)

После завершения работы программы "ED" полезно при помощи команды DIR или STAT проверить, что файлы, которые подверглись обработке, представлены в должном виде. Новый результирующий файл ("filename.ext") должен иметь то же самое имя, что и файл, первоначально указанный при обращении к программе "ED". Кроме того, на диске должен быть файл с резервной копией, имеющий имя, что и исходный, и расширение ".BAK".

ОЧИСТКА БУФЕРА РЕДАКТИРОВАНИЯ И РЕЗУЛЬТИРУЮЩЕГО ФАЙЛА БЕЗ ЗАВЕРШЕНИЯ ПРОГРАММЫ "ED"

*0

Команда O напоминает команду H, описанную выше, за исключением того, что ни содержимое буфера, ни содержимое временного результирующего файла на диске не сохраняются. Эта команда используется для ликвидации результатов проделанной до сих пор работы и создания "безупречной репутации". Имеется в виду то, что буфер редактирования и временный результирующий файл пусты. Другими словами, по этой команде можно вернуться к начальному моменту редактирования (т. е. к первому вводу команды ED). Поскольку данная команда может быть весьма "разрушительной", используйте ее осторожно и только в том случае, если абсолютно уверены в выполнении описанных действий.

После выполнения команды О появится промпт ".*" (без номера строки). Это означает, что произошла очистка буфера. Выполнение этой команды приводит не к завершению программы "ED", а лишь к возврату в исходное положение — к первому вводу команды ED. При этом исходный файл остается неизменным и теперь любая операция загрузки текста из исходного файла (инициируемая при помощи команды А) выполняется так, точно введен на команду ED.

ПРЕКРАЩЕНИЕ РАБОТЫ ПРОГРАММЫ "ED" С ОЧИСТКОЙ
БУФЕРА РЕДАКТИРОВАНИЯ И УНИЧТОЖЕНИЕМ
РЕЗУЛЬТИРУЮЩЕГО ФАЙЛА

*Q

Команда Q (от английского Quit — прекратить) используется для завершения работы программы "ED" и возврата в основной режим команд ОС CP/M. Однако программа "ED" выполнит все действия, что и по команде О (см. описание предыдущей команды). Эта команда применяется в тех случаях, когда работу программы "ED" необходимо завершить *без сохранения* любой проделанной работы (т. е. необходимо очистить буфер и временный результирующий файл, а исходный файл оставить без изменений). Командой "Q" следует пользоваться с осторожностью. Не вызывайте эту команду, если нет абсолютной уверенности в необходимости получить описанные результаты.

В качестве меры предосторожности против случайного применения этой команды программа "ED" выдает запрос (один и только один раз), на самом деле требуется выполнить команду или прервать. Если в качестве ответа введен символ "N", то выполнение этой команды *прерывается* и программа "ED" продолжит свою работу. Если же в качестве ответа введен символ "Y", то команда "Q" будет *выполнена* и появится основной промпт ОС CP/M, содержащий имя текущего дискового устройства.

Обратите внимание на последний вопрос программы "ED", поскольку легко случайно ввести команду Q. Эта ситуация продемонстрирована в следующем примере.

Отрицательный ответ:

20: * Q

Q-(Y/N)?N

20: *

Положительный ответ:

20: * Q

Q-(Y/N)?Y

A)

После ввода символа "N" или "Y" в ответ на запрос Q = (Y/N)? не нажмите после ответа клавишу RETURN. Программа "ED" отреагирует сразу после ввода символа "N" или "Y".

СООБЩЕНИЯ ОБ ОШИБКАХ ПРОГРАММЫ "ED"

Существует ряд ситуаций, при которых программа "ED" выдает сообщение об ошибке. Обычно сообщение об ошибке относится к неправильному вводу команды или некорректному использованию какой-либо команды. Однако каждое сообщение содержит символ, который описывает природу ошибки. Сообщение об ошибке печатается в формате "BREAK-X-C". Слово "BREAK" означает, что программа "ED" вынуждена прервать выполнение последней команды и выдать сообщение об ошибке; символ "X" — специальный символ, описывающий тип ошибки; символ "C" — последний символ, считанный программой "ED" в команде. В поле "X" ошибки сообщения могут быть напечатаны следующие символы.

? **Неопознанная команда.** Символ "?" означает, что введенная команда не является командой программы "ED".

Это означает, что данная команда либо сама по себе не является командой программы "ED", либо в командной строке допущена синтаксическая ошибка. Сразу же после символа "?" в сообщении об ошибке приводится последний символ неточной команды или командной строки.

> **Переполнение буфера.** Присутствие символа ">" означает, что весь свободный объем в буфере редактирования заполнен и он не может принимать больше входные данные. Для получения свободного пространства в буфере редактирования следует выполнить одну из следующих команд: nD, nK, nW, H, E, O, Q. При получении такого сообщения об ошибке перед тем, как предпринимать какие-либо действия, изучите описание этих команд. Тщательное изучение команд поможет принять решение, какая из них подходит.

Число повторений слишком большое. Если в сообщении об ошибке напечатан символ "#", то это означает, что в последней команде было задано слишком большое число повторений. Например, если задана команда "# Ftextstring ^Z", то программа "ED" определит, что значение "#" (все повторения заданной строки в буфере редактирования) для команды F (искать) не разрешено. В ответ на это сообщение укажите меньшее значение числа и повторений.

О **Нельзя открыть библиотечный файл, имя которого указано после команды R.** Присутствие символа "O" означает, что программа "ED" не может открыть библиотечный файл, имя которого задано в качестве параметра команды R (читать файл). Это сообщение означает, что либо указанного файла просто не существует, либо файл с заданным именем существует, но имеет расширение, отличное от ".LIB". Для разрешения данной проблемы следует по команде "E" закончить выполнение программы "ED" и затем найти файл, первоначально указанный с командой R. Если файл существует и имеет расширение ".LIB", то проверьте, что он не является "системным" файлом (см. описание команды STAT) и занимает более ОК (0 килобайт) памяти. Если нужный файл не имеет расширения "LIB", тогда временно переименуйте его или же сделайте с него копию, имеющую корректное расширение.

ГЛАВА 11



SUBMIT

Версии ОС *CP/M*:

1.3 1.4 2.0 2.2

Тип команды:

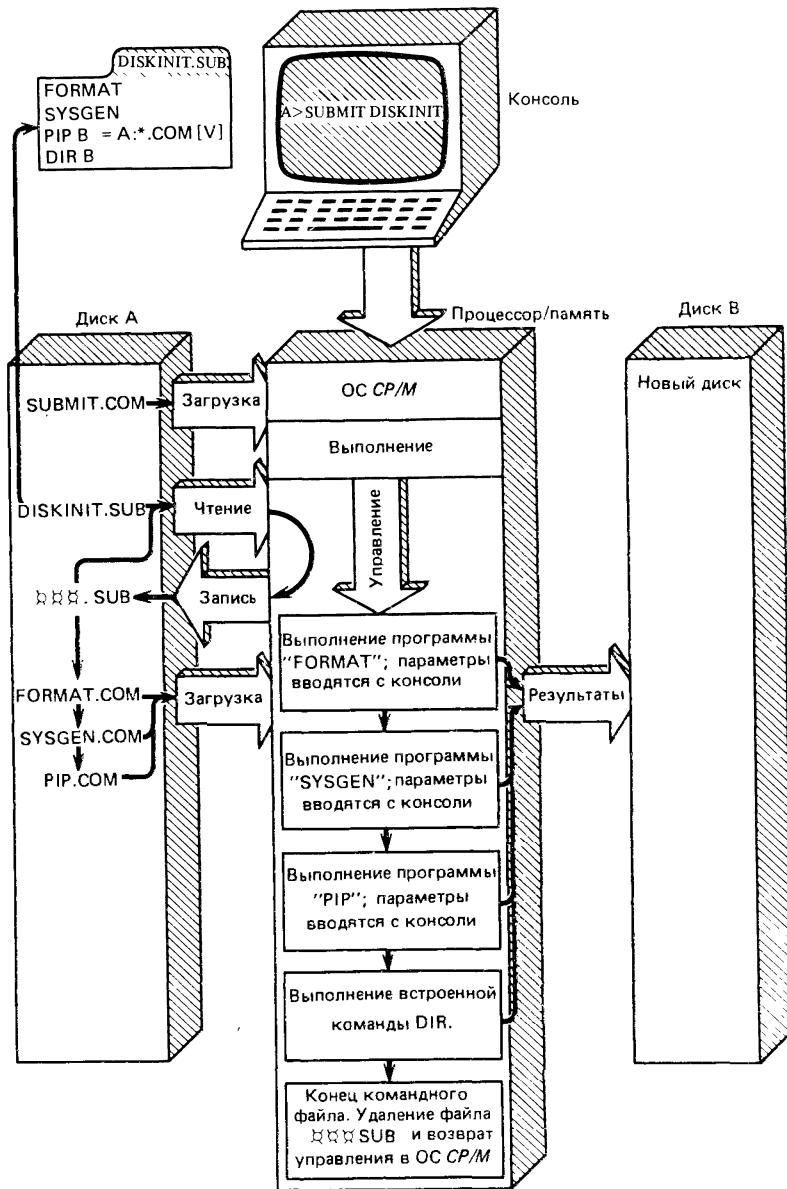
транзитная

ОБЩИЕ СВЕДЕНИЯ

Команда SUBMIT является транзитной командой, которая позволяет автоматически запускать на исполнение последовательности команд ОС *CP/M*. Для этих целей создается специальный файл, содержащий в текстовой форме требуемую последовательность команд ОС *CP/M*. Команда SUBMIT используется для того, чтобы "передать" команды из этого специального файла в ОС *CP/M* так, как будто они введены пользователем с пульта каждая отдельно. Поскольку команды вводятся косвенным образом, данный процесс "передачи" команд напоминает "косвенную обработку команд", применяемую во многих операционных системах.

Команда SUBMIT часто используется программистами, разрабатывающими программы, для исполнения под управлением ОС *CP/M*. Представляемый в распоряжение командный файл может содержать набор часто необходимых команд, например команды, обеспечивающие ассемблирование и их загрузку. Для автоматизации процедуры создания резервных копий командный файл может также содержать набор простых команд копирования с диска или команд запуска определенной последовательности набора прикладных программ.

Процесс автоматического запуска команд ОС *CP/M* проиллюстрирован следующей схемой:



ЗАПУСК И ИСПОЛЬЗОВАНИЕ КОМАНДЫ SUBMIT

Для того чтобы использовать команду **SUBMIT**, в первую очередь следует при помощи текстового редактора "ED" или какого-либо другого текстового редактора, совместимого с **ОС CP/M**, создать "командный" ("SUBMIT") файл. Имена всех командных файлов должны иметь расширение ".SUB".

Командный файл может содержать последовательность команд ОС CP/M (встроенных или транзитных). Вводите команды в том порядке: в котором их необходимо выполнять. При завершении ввода каждой команды или командной строки нажмите клавишу "RETURN". Для этих команд используйте тот же синтаксис, что и при вводе их с пульта вручную.

Ниже представлено содержание командного файла с последовательностью команд, которая используется для инициализации нового гибкого диска:

```
A>TYPE DISKINIT.SUB
```

```
FORMAT  
SYSGEN  
PIP B:=A::*COMEV3  
DIR B:
```

A>

Команды, представленные в этом примере, можно ввести отдельно (в той же последовательности), однако при этом нельзя ни на минуту отойти от машины. Файл "DISKINIT.SUB" содержит команды, необходимые при разметке нового гибкого диска, вставленного в устройство с именем "B".

Для запуска на выполнение команд из файла "DISKINIT.SUB" необходимо в основном командном режиме ОС CP/M ввести команду SUBMIT, затем имя файла (для использования команды SUBMIT файл "SUBMIT.COM" должен находиться на текущем диске). При этом расширение ".SUB" не вводится, поскольку существующий с указанным именем и расширением ".SUB" файл будет найден автоматически. По команде SUBMIT (транзитная программа "SUBMIT") полностью берет на себя управление ЭВМ. В результате этого каждая команда из командного файла выполняется в том порядке, в котором они изначально в него записаны.

Для тех, кто интересуется внутренним механизмом выполнения команды SUBMIT, ниже описаны реальные действия ЭВМ. Сначала содержимое командного файла копируется во временный файл с именем "XXX.SUB". Именно из этого временного файла выбираются на выполнение команды, указанные в исходном файле. Как только все команды выполнены, файл "XXX.SUB" удаляется, а исходный командный файл остается без изменений.

Пример, являющийся продолжением предыдущего примера, демонстрирует, что происходит при запуске на выполнение команд из файла "DISKINIT.SUB" (текст, заключенный в круглые скобки, представляет собой авторские комментарии):

```
A>SUBMIT DISKINIT (эта строка вводится с консоли оператором)
```

```
A>FORMAT (эта строка печатается в процессе выполнения команды SUBMIT)  
(команда FORMAT выполняется, как обычно  
Данные, необходимые для форматирования  
диска B, вводятся с консоли)
```

```
A>SYSGEN (печатается в процессе выполнения команды SUBMIT)  
(команда SYSGEN выполняется, как обычно.  
Данные, необходимые для копирования образа  
системы с диска A на диск B, вводятся с консоли)
```

```
A>PIP B =A.* COMEV3 (печатается в процессе выполнения команды SUBMIT)  
(команда PIP выполняется, как обычно. Все  
"COM"-файлы, расположенные на диске A,  
копируются на диск B)
```

A>DIR B: (печатается в процессе выполнения команды SUBMIT)
(на консоль выводится список всех файлов,
расположенных на диске B)

A>
(После выполнения команды SUBMIT на консоль выдается
основной промпт с именем текущего диска Теперь ОС CP/M
готова к выполнению новой команды)

Как видно из примера, запуск команд из файла "DISKINIT.SUB" осуществляется в том порядке, в котором они в этом файле расположены. После выполнения очередной команды запускается команда из следующей строки командного файла. Программа "SUBMIT" продолжает работать подобным образом до тех пор, пока не будет выполнена последняя команда. После этого выдается основной промпт ОС CP/M, содержащий имя текущего диска (A>, B> и т. д.).

По команде SUBMIT может быть запущена любая команда ОС CP/M и фактически любая транзитная программа. Программы, требующие ввода пользователя каких-либо данных (например, "FORMAT" или "SYSGEN"), выполняются как обычно, т. е. как при индивидуальном вводе.

ИСПОЛЬЗОВАНИЕ ПЕРЕМЕННЫХ В КОМАНДНЫХ ФАЙЛАХ

Помимо команд ОС CP/M в командных файлах можно использовать специальные "символы переменных". Эти переменные позволяют создавать командные файлы без указания в командных строках имен файлов, дискового устройства или каких-либо других символьных строк. Вместо этого при вводе команды SUBMIT совместно с именем одного из этих файлов можно ввести конкретные значения этих переменных сразу после имени командного файла. Затем указанные в командной строке параметры вставляются на место переменных в командном файле, придавая им тем самым конкретные значения. Имена переменных в командном файле должны иметь формат "Q1", "Q2", "Q3" и т. д., где переменная "Q1" соответствует первому параметру, переменная "Q2" – второму и т. д. При вводе команды SUBMIT для замены переменных конкретными значениями используется следующий синтаксис:

A > SUBMIT FILENAME X1 X2 X3 . . .

. . . , где значение "X1" предназначено для замены всех вхождений переменной "Q1", значение "X2" предназначено для замены всех вхождений переменной "Q2", значение "X3" – для замены всех вхождений переменной "Q3" и т. д.

В следующем примере показано, как файл "DISKINIT.SUB" (см. первый пример из этой главы) может быть преобразован в файле, содержащем переменные. После внесения переменных использование командного файла не ограничивается только дисками "A" и "B":

A>TYPE DISKINIT.SUB

FORMAT
SYSGEN
PIP <2:=<1:*.COM[V]
DIR <2

A>

За счет того, что вместо имени "В" используется переменная "Q 2", вместе с "А" — переменная "Q 1", этот командный файл можно использовать при любой комбинации дисков. Если, например, вместо дисков "А" и "В" требуется использовать диски "В" и "С", то следует команду SUBMIT ввести следующим образом:

```
B>SUBMIT DISKINIT В С  
  
B>FORMAT  
(Выполнение команды FORMAT)  
B>SYSCEN  
(Выполнение команды SYSCEN)  
B>PIP C:=B.*.COM[V]  
(Выполнение команды PIP)  
B>DIR C:  
(Выполнение команды DIR)  
B>
```

Этот пример показывает, что в результате выполнения операции подстановки файл "DISKINIT.SUB" стал таким же, как и до этого, за исключением того, что теперь имена дисков заданы конкретно. Все вхождения переменной "Q 1" были заменены именем "В", а все вхождения переменной "Q 2" — именем "С"). Замена переменных может быть выполнена корректно (т. е. так, как требуется) только при условии, что параметры в командной строке

SUBMIT FILENAME VARIABLE1 VARIABLE2"

введены в требуемой последовательности. Если, например, задана командная строка "SUBMIT DISKINIT С В", то вместо команды "PIP C: = B:*.COM" [V]" в командном файле будет команда "PIP B: = C: *.COM [V]", а вместо команды "DIR C:" — команда "DIR B:". Полученные при этом результаты могут быть весьма нежелательными. Поэтому перед запуском на исполнение командного файла всегда полезно предварительно просмотреть его (при помощи команды TYPE). В этом случае можно проверить, сколько и какие переменные используются, а кроме того определить порядок, в котором они вводятся в командной строке.

В командных файлах переменные можно использовать для замены фактически любой строки текста в команде. Например, переменную можно использовать для замены имени файла в командной строке. Переменную можно даже применить для замены команды ОС CP/M. Одну переменную можно использовать в командном файле необходимое число раз. По команде SUBMIT все необходимые подстановки выполняются до конца файла. Пользуйтесь символами этих переменных и благородно и осторожно. Перед запуском командного файла всегда проверяйте назначение символов переменной.

Одиночные символы "Q" в командных файлах всегда распознаются как символы переменной. Если символ "Q" необходимо применить в командной строке в качестве обычного символа, то ему должен предшествовать еще один символ "Q". Этот хитроумный трюк сработает, потому что при обнаружении в командном файле двух символов "QQ" — первый из них отбрасывается, а второй трактуется как допустимый, фиксированный символ командной строки. Если, например, необходимо использовать команду "STAT *.* Q R/O" для присвоения статуса защиты всем файлам на указанном диске "R/O" ("Только чтение"), командный файл может содержать строку

"STAT*.***Q**R/O". При этом символ "**Q**" трактуется как "реально существующий", а не как символ переменной, что продемонстрировано в примере:

```
A>TYPE ALL-R/O SUB
STAT *.* >>R/O
STAT *.*
A>
A>SUBMIT ALL-R/O

A>STAT *.* >>R/O
(после того, как будет напечатана данная
командная строка, команда STAT начнет
выполняться, как обычно)
A>STAT *.*
(команда STAT выполняется, как обычно)

A>
```

В примере показано, что если командный файл содержит строку "STAT *.***Q**R/O", то это приведет к ошибке.

ИСПОЛЬЗОВАНИЕ КОММЕНТАРИЕВ В КОМАНДНЫХ ФАЙЛАХ

Для тех, кто собирается использовать большие или сложные командные файлы, полезно вставлять в эти файлы комментарии. В командный файл можно вставить неограниченное количество комментариев, пока каждая строка с комментариями начинается с точки с запятой. Каждую строку с комментариями также следует отделять от командных строк, которые должны быть запущены на выполнение. После ввода команды SUBMIT комментарии распознаются ОС CP/M как комментарии, пока первым символом в строках комментарий является точка с запятой.

В следующем примере показано, как файл "DISKINIT.SUB" (см. предыдущий пример) можно снабдить комментариями:

```
A>TYPE ALL-R/O SUB
;СЛЕДУЮЩАЯ НИЖЕ КОМАНДНАЯ СТРОКА СОДЕРЖИТ ДВОЙНОЙ
;СИМВОЛ «», СЛЕДОВАТЕЛЬНО ЭТОТ СИМВОЛ НЕ БУДЕТ
;ИНТЕРПРЕТИРОВАТЬСЯ КАК ПЕРЕМЕННАЯ КОМАНДЫ SUBMIT
STAT *.* >>R/O
;ПОСЛЕ ТОГО, КАК КЛЮЧИ ЗАЩИТЫ ВСЕХ ФАЙЛОВ ПОЛУЧАТ
;ЗНАЧЕНИЕ "R/O", СПИСОК ИМЕН ЭТИХ ФАЙЛОВ ВЫВОДИТСЯ
;НА КОНСОЛЬ
STAT *.*

A>
```

После того как командный файл будет запущен на выполнение, комментарии также будут выведены на консоль. В комментариях каждый символ "**Q**", который должен быть частью комментария, безусловно необходимо заменять на пару символов "**QQ**". Кроме того, если необходимо использовать в комментариях переменные. Тогда при каждом выполнении командного файла на консоль в виде комментария будет выдаваться значение соответствующего параметра. При просмотре командного файла с помощью команды TYPE или при внесении в него изменений с помощью какого-либо текстового редактора или текстового процессора, эти комментарии облегчают понимание его содержимого.

СПЕЦИАЛЬНЫЕ ВОЗМОЖНОСТИ КОМАНДЫ SUBMIT

В ОС *CP/M* версии 2.0 и более поздних имеется специальная программа, которая называется "XSUB.COM". Строго говоря, программа "XSUB.COM" является не транзитной программой ОС *CP/M*, а скорее подпрограммой программы "SUBMIT.COM"^{*}. Эта специальная программа обеспечивает передачу текста из командного файла в качестве входных данных программ, вызываемых SUBMIT. В этом случае информация передается так же, как при вводе с пульта вручную. С помощью этой утилиты можно создать командный файл, содержащий командную строку для запуска какой-либо программы, затем вставить в него текст, представляющий собой ответы на промпты этой программы, обычно вводимые с консоли. Эта утилита существенно расширяет возможности команды SUBMIT за счет повышения уровня автоматизации. Команда XSUB описывается в следующей главе.

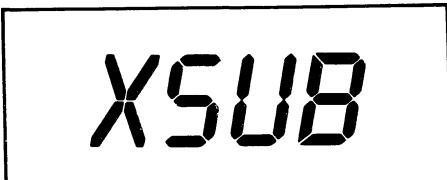
ГЛАВА 12

Версии ОС *CP/M*:

2.0 2.2

Тип команды:

транзитная



ОБЩИЕ СВЕДЕНИЯ

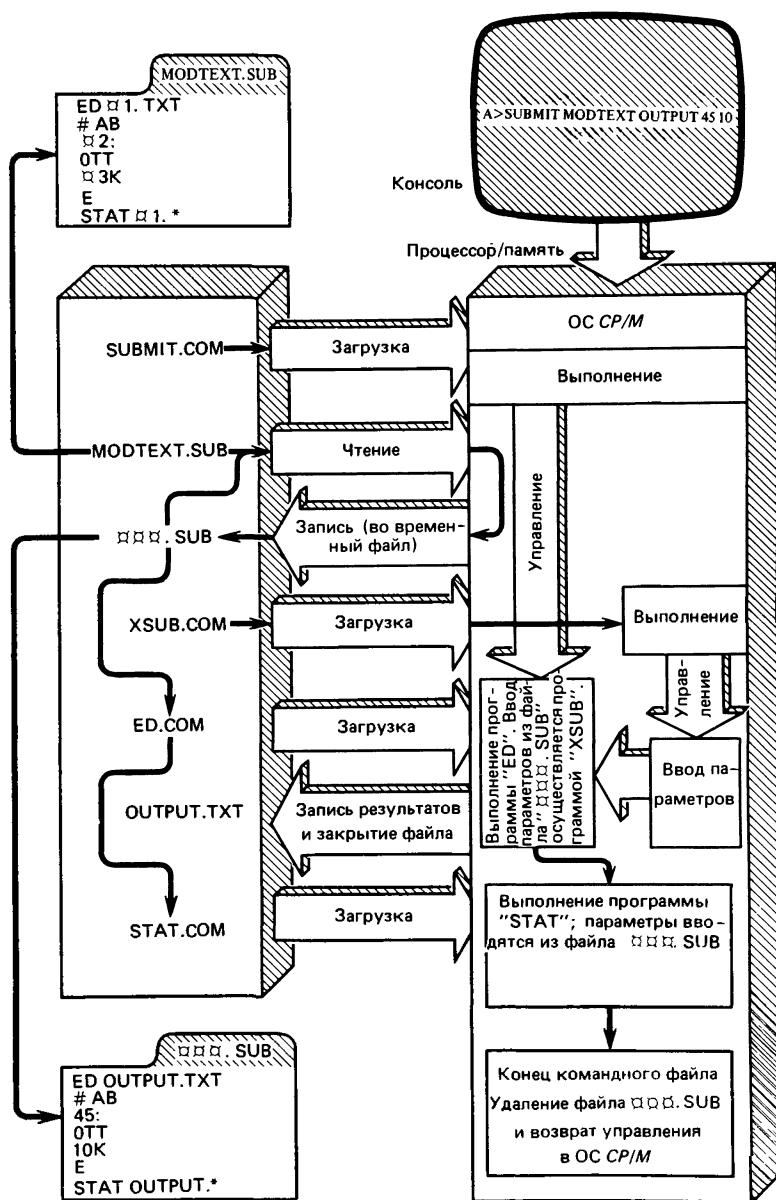
Команда XSUB (от английского Extended SUBMIT – расширенная программа "SUBMIT") является специальной командой, которая используется только в сочетании с командой SUBMIT. Команда XSUB позволяет пользователю существенно повысить уровень автоматизации процесса запуска команд на выполнение по команде SUBMIT. Прежде чем пытаться использовать команду XSUB, желательно хорошо изучить команду SUBMIT.

Команда XSUB позволяет создавать командные файлы (файлы с расширением ".SUB") со специальными командными строками, передаваемыми некоторым командам и программам ОС *CP/M*. Командные файлы, созданные по команде XSUB, кроме того, содержат текстовую информацию, которая обычно вводится с консоли в ответ на промпты программы. Команду XSUB можно использовать совместно с командой SUBMIT в различных случаях, например, при модификации существующей программы на языке Бейсик, запуске ее, автоматическом вводе данных в процессе выполнения, повторной, отличной от первой, модификации программы с целью последующего исполнения и т. д.

Ниже представлена схема выполнения команды SUBMIT совместно с

* В этом смысле команда XSUB является командой не ОС *CP/M*, а командой программы "SUBMIT.COM". – *Прим. перев.*

командой XSUB. Здесь показано, как использовать команду XSUB для передачи команд или данных программам, запущенным на выполнение по команде SUBMIT.



ОБРАЩЕНИЕ К КОМАНДЕ И АЛГОРИТМ ЕЕ РАБОТЫ

Для того чтобы использовать команду XSUB, необходимо, чтобы оба файла "SUBMIT.COM" и "XSUB.COM" хранились на одном из дисков, вставленных в настоящий момент в дисковое устройство. Обе программы используются совместно, но команда XSUB сама по себе не вводится. Вместо этого команду XSUB помещают в первую строку командного файла. При обработке командного файла программа "XSUB" будет запущена на выполнение в специальной области оперативной (RAM) памяти ЭВМ. После этого программе "SUBMIT" будут добавлены специальные средства, а в оперативной памяти будет учреждена специальная буферная область, с помощью которой обеспечивается соответствующая передача текста из командного файла.

Достаточно однажды загрузить программу "XSUB" в оперативную память и она останется там, если только не потребуется полная перезагрузка системы. При вводе команды CTRL/C (реинициализация системы) программа "XSUB" из оперативной памяти удалена не будет.

ИСПОЛЬЗОВАНИЕ КОМАНДЫ XSUB

При использовании команды XSUB можно передавать входные данные программам из командного файла только, если существует стандартный способ завершения этих программ. Иными словами, программы (например, PIP), для завершения которых необходимо нажать клавишу RETURN или ввести команду CTRL/C (реинициализация системы), нельзя использовать совместно с командой XSUB, обладающей особыми свойствами. Это объясняется тем, что программа "SUBMIT" не опознает пустые строки (т. е. те, которые содержат только символы возврата каретки и перевода строки в коде ASCII) в командном файле. Эти строки будут рассматриваться как ошибочные. Строки, содержащие символ "^C" (команда ETX в коде ASCII или значение 03 в шестнадцатеричном формате) для обозначения команды CTRL/C (реинициализация системы), будут также рассматриваться как ошибочные. Поэтому если необходимо автоматически завершить программы, запущенные командой SUBMIT, то следует использовать только те программы, которые для завершения требуют ввода специальных команд (таких как E при использовании программы "ED"), или те программы, которые завершаются сразу после получения ответа на последний промпт. Пользователь, безусловно, всегда имеет право ввести вручную с консоли специальную команду выхода из программы.

В следующем примере представлено содержимое командного файла, в котором используется команда XSUB:

```
A>TYPE MDTXT.SUB
```

```
XSUB
ED *1.TXT
$AB
^2:
OTT
^3K
E
STAT *1.*
```

```
A>
```

В этом примере показано, что команда XSUB используется совместно с командой ED. При обработке файла "MODTEXT.SUB" в первую очередь происходит загрузка программы "XSUB". Затем будет запущена команда "ED Q1. TXT" с переменной "Q1", замененной значением первого параметра. Программа "ED" открывает специфицированный файл, и, как только на консоли появляется промпт ":"*, средства программы "XSUB" будут приведены в действие.

Следующая строка командного файла содержит группу команд программы "ED", и они передаются программе "ED", как и введенные с консоли. Кроме того, последующие строки содержат команды программы "ED" за исключением последней строки ("STAT Q1.*"), содержащей команду, которая должна выполняться в основном командном режиме ОС CP/M. Это не составляет проблем, поскольку в предыдущей строке содержится команда E, которая предписывает программе "ED" завершить сеанс редактирования и восстановить режим команд ОС CP/M. Таким образом, команда "STAT Q1.*" будет обработана правильно.

В следующем примере показано, что происходит в процессе обработки этого командного файла:

```
A>SUBMIT MODTEXT OUTPUT 45 10
A>XSUB
A>ED OUTPUT.TXT

    * #AB
 1. * 45.
45. * FROM THIS, KILL THE NEXT 10
45. * 10K
45. * E

(команда XSUB по-прежнему активна)

A>STAT OUTPUT *
RECS   BYTES   EXT ACC
 110     14K      1 R/W A:OUTPUT.BAK
 109     13K      1 R/W A:OUTPUT.TXT
BYTES REMAINING ON A:  476K

A>
```

Из примера видно, что в процессе обработки командного файла обращение к программе "ED" происходит обычным образом. Кроме того, все команды программы "ED" из этого файла были переданы программе "ED" и затем успешно выполнены. После обработки команды E (выход) выполнение программы "ED" было завершено. Сообщение "(XSUB ACTIVE)" означает, что программа "SUBMIT" перешла в режим команд ОС CP/M и готова к обработке очередной команды. Поскольку программа "XSUB" в этот момент остается активной, ее средства можно в дальнейшем использовать для передачи программам исходных данных.

Существует множество различных приложений, в которых совместно с командой SUBMIT можно использовать команду XSUB. Однако, не ожидайте, что командные файлы с первого раза начнут работать должным образом. Чтобы файлы этого типа работали должным образом, их часто приходит-

ся многократно исправлять. Например, вопреки утверждению о том, какие программы следует использовать совместно с командой XSUB для автоматического завершения программы и какие нет, можно столкнуться с программой, которая должна работать, но по какой-либо причине не работает. Хорошо экспериментировать с командами SUBMIT и XSUB на пустом диске (который не содержит ни одного представляющего интерес файла).

ГЛАВА 13



Версии ОС *CP/M*:

1.3 1.4 2.0 2.2

Тип команды:

транзитная

ОБЩИЕ СВЕДЕНИЯ

Данная команда обеспечивает обращение к транзитной программе ОС *CP/M*, которая также называется "ASM". Программа "ASM" является ассемблером ОС *CP/M*, т. е. преобразует файлы с исходными текстами программ на языке ассемблера в файлы с шестнадцатеричным кодом для микропроцессора Intel 8080. Файлы с шестнадцатеричным кодом не пригодны к непосредственному исполнению и сначала должны быть обработаны командой LOAD. СОМ*. Дальше речь пойдет о том, как работает программа "ASM". Файлы с исходными текстами программ создаются при помощи стандартного текстового редактора ОС *CP/M* "ED" или любого другого текстового редактора или текстового процессора, совместимого с ОС *CP/M*. Текст в исходном файле должен содержать операторы языка ассемблера для микропроцессора Intel 8080. Эти операторы должны быть расположены так, чтобы весь текст в целом представлял собой требуемую программу для Intel 8080. В исходном тексте указывается начальный адрес программы, объем памяти, отведенной под переменные, имена специальных ячеек нулевой страницы и многое другое. Программа "ASM" и используется для сканирования текста программы и преобразования всех мнемонических обозначений в соответствующие шестнадцатеричные коды, а затем сбора результатов в результирующем файле в шестнадцатеричном формате. Этот файл позже можно преобразовать в файл с исполняемым кодом при помощи команды LOAD.

Программа "ASM" выполняет следующие три основные функции:

* Причина, по которой ассемблер фирмы Digital Research ("ASM") не выдает чистого объектного кода, пригодного для немедленного исполнения, состоит в том, что в ранних моделях микро-ЭВМ для загрузки программ использовалась перфолента. Перфолента была разбита на специальные блоки так, чтобы можно было обнаруживать ошибки чтения.

Это и есть тот шестнадцатеричный формат, который до сих пор порождает программа "ASM". Отметим также, что файлы с шестнадцатеричными кодами программ можно загружать в оперативную память по команде DDT.

1. Считывает содержимое исходного файла (FILENAME.ASM), асSEMBЛИрует программу и помещает результаты в выходной файл (FILENAME.HEX) в шестнадцатеричном формате.

Файл с шестнадцатеричным кодом можно преобразовать в файл с исполняемым машинным кодом при помощи команды LOAD. Исходный файл остается при этом без изменений.

2. Помимо файла с шестнадцатеричным кодом исходной программы программа "ASM" создает также еще один выходной файл, который представляет собой специальную версию исходного файла. Этот специальный выходной файл называется файлом "листинга" или файлом "распечатки" (FILENAME.PRN). Он содержит тот же текст, что и исходный файл, помимо указанного для каждой строки результата асSEMBЛИРОВания, представленного в шестнадцатеричном коде. Этот шестнадцатеричный код помещается в левой части каждой строки.

3. В процессе асSEMBЛИРОВания программа "ASM" составляет список всех обнаруженных ошибок. Этот список выводится на консоль (по мере выполнения программы "ASM") и в результирующий файл распечатки (FILENAME.PRN). Сообщения об ошибках представляются в виде специальных символов и размещаются в соответствующих строках исходного текста в файле распечатки.

Программа "ASM" асSEMBЛИРУЕТ только те программы, которые написаны на языке асSEMBлера для микропроцессора Intel 8080. Хотя многие микроЭВМ, созданные на базе микропроцессора Intel 8085 или Zilog Z80, используют ОС CP/M, не следует пытаться асSEMBЛИРОВАТЬ с помощью команды ASM программы, которые специально написаны для микропроцессора Zilog Z80. Микропроцессор Intel 8085 имеет всего одну команду, отличающуюся от команд микропроцессора Intel 8080, а именно команду, ИНИЦИИРУЮЩУЮ последовательный ввод-вывод, поэтому обычно все программы для Intel 8080 и Intel 8085 являются совместимыми. В любом случае программа в машинном коде для Intel 8080 также выполняется на ЭВМ с микропроцессором Intel 8085 или Zilog Z80, однако программы, написанные *специально* для Intel 8085 или Zilog Z80, не будут выполняться на ЭВМ с микропроцессором Intel 8080. Программа "ASM" позволяет программисту перемещать программы с ЭВМ, построенной на базе Intel 8080, на ЭВМ, построенные на базе Intel 8085 или Zilog Z80, не заботясь при этом о совместимости машинных кодов.

Описание программы "ASM", представленное в этой книге, содержит только информацию, связанную с программой "ASM" как таковой. Оно не содержит информации о том, как писать программы на языке асSEMBлера для микропроцессора Intel 8080. Здесь представлены сведения обо всех директивах, командах и функциях асSEMBлера, связанные со спецификой использования программы "ASM". Следовательно, программисты могут воспользоваться этой информацией для того, чтобы убедиться, что их программы будут правильно обрабатываться программой "ASM".

ОБРАЩЕНИЕ К ПРОГРАММЕ "ASM"

Для использования программы "ASM" необходимо, чтобы файл "ASM.COM" хранился на диске, вставленном в одно из устройств системы. Синтаксис

сис команды ASM отличается от синтаксиса остальных команд ОС *CP/M*. Исходный файл может храниться на любом диске, но его имя должно иметь расширение ".ASM". Если исходный файл находится на том же диске, что и файл "ASM.COM", просто введите следующую командную строку:

A > ASM FILENAME,

где "FILENAME" – имя исходного файла (FILENAME.ASM). Расширение ".ASM" не надо, поскольку программа "ASM" автоматически ищет файл с указанным именем, предполагая, что оно имеет расширение ".ASM". Если такой файл найден и имеет корректное расширение, то программа "ASM" начнет читать его содержимое и преобразовывать исходный текст в шестнадцатеричный код. Если команда ASM вызвана таким способом, то результирующие файлы, содержащие шестнадцатеричный код программы (FILENAME.HEX) и распечатку (FILENAME.PRN), создаются автоматически и размещаются на текущем диске. Оба файла ("HEX" и ".PRN") имеют то же самое имя, что и исходный файл.

ИСПОЛЬЗОВАНИЕ КОМАНДЫ ASM

Вместе с командой ASM можно ввести специальные дополнительные параметры, определяющие диски, на которых должны быть размещены исходный и результирующие файлы. Эти параметры представляют собой три отдельных символа, которые вводятся в командной строке после точки ("."). Вследствие своего местоположения они выглядят как расширение имени исходного файла, хотя на самом деле это не так.

Командная строка при этом вводится в следующем формате:

A > ASM FILENAME.123,

где символ "1" обозначает позицию, на которой вводится имя устройства ("A", "B", "C" и т. д.), содержащего файл с исходным текстом (FILENAME.ASM). Символ "2" обозначает позицию, на которой вводится имя диска, где необходимо разместить файл с шестнадцатеричным кодом (FILENAME.HEX). Символ "3" обозначает позицию для имени диска, на котором требуется разместить файл с распечаткой (FILENAME.PRN). Вводите только одиночные символы без двоеточия (":"). В поле дополнительных параметров можно задать любую комбинацию имен дисков. Эти указанные диски должны быть вставлены в соответствующие устройства, подключенные к ЭВМ. При указании имен дисков необходимо соблюдать ограничения, которые накладывает используемая версия ОС *CP/M* (в версии 1.3 – имена "A" или "B", в версии 1.4 – имена от "A" до "D", в версии 2.0 и более поздних имена от "A" до "P"). Например, командная строка

A > ASM PROG-1.ABB

указывает программе "ASM" найти исходный файл "PROG-1.ASM" на диске "A", а оба результирующих файла ("HEX" и "PRN") разместить на диске "B".

Существуют два специальных символа "Z" и "X", которые можно вводить в поле параметров. Если необходимо запретить вывод либо ".HEX", либо "PRN" файла, либо обоих, то символ "Z" должен быть указан соответственно во второй или в третьей позиции поля параметров. Если символ "Z" задан в обеих позициях, то эффект будет тот же. Если необходимо вывести рас-

печатку на консоль и не создавать при этом "PRN" файл, то в третьей позиции поля параметром должен быть задан символ "X".

АССЕМБЛИРОВАНИЕ ФАЙЛА, РАСПОЛОЖЕННОГО НА ТЕКУЩЕМ ДИСКЕ

ASM TEST01

По сравнению с написанием программы на языке Ассемблер обработка ее с помощью программы "ASM" сама по себе является относительно простой операцией. Если введена командная строка "ASM FILENAME" (где "FILENAME" – имя файла, который необходимо обработать), то программа "ASM" предполагает, что указанный файл находится на текущем диске. После того как команда, указывающая конкретный исходный файл, введена программа "ASM" выдает на контроль свое имя и номер версии. Затем вывод сообщений на консоль прекращается на короткое время, в течение которого происходит ассемблирование кода. Если в процессе ассемблирования обнаружены какие-либо ошибки, то по мере их обнаружения выдаются соответствующие сообщения. После завершения процесса ассемблирования результирующие файлы с шестнадцатеричным кодом (FILENAME.HEX) и распечаткой (FILENAME.PRN) сохраняются на текущем диске.

Ниже представлен пример отчета о выполнении программы "ASM". Он включает в себя адрес первого байта свободной (неиспользованной) области, следующей за отассемблированной программой, а также общий процент использования таблицы символов в процессе ассемблирования:

```
A>ASM TEST01
CP/M ASSEMBLER - VER X.X
020F
006H USE FACTOR
END OF ASSEMBLY
```

A>

Из этого примера видно, что процесс ассемблирования исходного файла "TEST01.ASM" прошел успешно. Это можно заключить из того факта, что на консоль не было выдано ни одного сообщения об ошибке. Символы "X.X" в первой строке отмечают то место, где печатается номер версии программы "ASM". Обычно этот номер совпадает с номером версии ОС CP/M (1.4, 2.0 и т. д.). На следующей строке напечатан адрес первого байта свободной (неиспользованной) области, который равен "020F". Этот адрес напечатан в шестнадцатеричном формате и представляет собой адрес байта, следующего непосредственно за последним байтом, отведенным под ассемблированную программу. Вычитая из этого числа 100 в шестнадцатеричном формате, можно определить общий объем памяти, занимаемой данной программой. В следующей строке напечатано сообщение "006H USE FACTOR", которое говорит о том, что в процессе ассемблирования таблица символов программы "ASM" использовалась на 6 (шестнадцатеричное) процентов. Таблица символов про-

граммы "ASM" представляет собой область, которая используется для хранения значений символьических имен из исходной программы (этим символьическим именам могут соответствовать конкретные адреса в программе или же данные). Эта область занимается под программу "ASM" на время процесса ассемблирования. Процент использования здесь таблицы символов зависит от размеров исходной программы и от количества временно хранящихся символьических имен. Эта таблица символов имеет ограниченные размеры и не должна переполняться, чтобы процесс ассемблирования протекал нормально.

После того как процесс ассемблирования закончен и программа "ASM" завершила свое выполнение, убедитесь, что все относящиеся к программе файлы хранятся на диске. С каждой программой должны быть связаны три основных файла. Это показано в следующем примере:

```
A>DIR TEST01.*  
A: TEST01 BAK : TEST01 ASM : TEST01 HEX  
A: TEST01 PRN  
A>
```

Файл с резервной копией ("BAK") присутствует, если он не был удален после последнего сеанса редактирования исходного ("ASM") файла. В процессе ассемблирования исходный ("ASM") файл не изменяется, в чем можно убедиться, просмотрев его содержимое.

Файл с шестнадцатеричным кодом ("HEX") впоследствии используется для создания файла с исполняемым машинным кодом ("COM") и, если необходимо, для нужд отладки.

Файл с распечаткой ("PRN"), по существу, представляет собой копию исходного ("ASM") файла. В левой части его текста размещаются результаты ассемблирования исходной программы. Результат ассемблирования каждого оператора или команды размещается в одной с ним строке.

Теперь можно просмотреть эти основные файлы с тем, чтобы проверить их содержимое.

В следующих трех примерах приводится содержимое соответственно исходного ("ASM"), объектного ("HEX") файлов и файла с распечаткой ("PRN") демонстрационной программы "TEST01".

Исходный ("ASM") файл:

```
A>TYPE TEST01.ASM  
; ПРИМЕР ПРОГРАММЫ НА АССЕМБЛЕРЕ ОС СР/М(8080).  
; ВЫПОЛНЕНИЕ НАЧИНАЕТСЯ С ПЕРВОГО БАЙТА ОБЛАСТИ  
; ТРАНЗИТНЫХ ПРОГРАММ.  
ORG 100H  
SORT:  
    LXI H,SW      ;ЗАГРУЗИТЬ В РЕГИСТР АДРЕС КЛЮЧА  
    MVI M,1        ;ДЛЯ ПЕРВОЙ ИТЕРАЦИИ УСТАНОВИТЬ В 1  
    LXI H,I        ;ИНДЕКС АДРЕСА  
    MVI H,0        ;I = 0  
  
END  
A>
```

Объектный (*.HEX) файл:

```
A>TYPE TEST01.HEX
: 10010000214601360121470136007EFE09D2190140
: 100110002146017EB7C20001FF5F16002148011983
: 10012000194E79234623965778239ED43F01B2CAA7
: 100130003F0156702B5E712B722B732146013421C7
: 070140004770134C30A01006E
: 04015800640001B0BE
: 0000000000
A)
```

Файл с распечаткой (*.PRN):

```
A>TYPE TEST01.PRN
; ПРИМЕР ПРОГРАММЫ НА
; АССЕМБЛЕРЕ ОС СР/М(8080).
; ВЫПОЛНЕНИЕ НАЧИНАЕТСЯ С ПЕРВОГО
; САЙТА ОБЛАСТИ ТРАНЗИТНЫХ ПРОГРАММ.
0100          ORG 100H
0100 214601   SORT: LXI H,SW  ;ЗАГРУЗИТЬ АДРЕС
                           ;КЛЮЧА В РЕГИСТР H
0103 3601     MVI M,1   ;ДЛЯ ПЕРВОЙ ИТЕРАЦИИ
                           ;УСТАНОВИТЬ В 1
0105 214701   LXI H,I   ;ЗАГРУЗИТЬ АДРЕС ИНДЕКСА
0100 3600     MVI H,0   ;I = 0
                           ;END
020F          D
A>
```

На первый взгляд содержимое объектного (*.HEX) файла во втором примере (последовательность шестнадцатеричных чисел) выглядит очень неразборчиво. Его, безусловно, можно расшифровать, однако понять, как объектный код соотносится с исходным текстом, проще по распечатке (содержимому файла *.PRN), представленной в третьем примере. Обычно содержимое файла *.PRN используется для проверки программы, а также для согласования процедур отладки с исходным текстом. Отметим, что любые строки исходного текста, начинающиеся с точки с запятой (;;) воспринимаются программой "ASM" как комментарии (файл, см. *.PRN), а не как реальные операторы и команды.

Отметим, что исходный текст из "ASM" файла сдвигается в файле *.PRN на шестнадцать пробелов вправо с тем, чтобы освободить место под объектный код. Эти пробелы истолковываются как два символа горизонтальной табуляции (HT) в коде ASCII (обычно символ табуляции в коде ASCII по умолчанию равен восьми пробелам). При выводе текста из файла "PRN" на печатающее устройство может оказаться, что его правый край выходит за границы бумаги (это обязательно произойдет, если печатающее устройство обеспечивает печать только восьмидесяти символов в строке). Для того чтобы устраниТЬ это несоответствие, если имеется возможность, переведите работающее устройство в режим печати 132 символов. Или же отредактируйте исходный (*.ASM) файл и уменьшите ширину текста на шестнадцать символов, чтобы файл "PRN" при последующем использовании команды ASM был сформирован должным образом. Или согласитесь на то, что часть отчета будет напечатана на полосе бумаги, которая используется для перфорации, если только печатающее устройство не делает автоматический перенос текста на следующую строку.

СООБЩЕНИЯ ОБ ОШИБКАХ, ВЫДАВАЕМЫХ НА КОНСОЛЬ

Существуют различные сообщения об ошибках, которые выдаются в процессе ассемблирования программой "ASM" и операционной системой *CP/M*. Это могут быть следующие сообщения:

NO SOURCE FILE PRESENT. Выдается на консоль в том случае, если исходный (".".ASM") не будет найден на указанном диске. Ошибка может быть вызвана тем, что файл отсутствует на текущем (если имя диска не задано) или на указанном (если используются дополнительные параметры) диске. Ошибки могут также состоять в том, что указанный файл не имеет расширения ".ASM". Кроме того, это сообщение будет получено в том случае, если после команды ASM отсутствует имя исходного файла.

SOURCE FILE NAME ERROR. Это сообщение выдается на консоль в том случае, если имя исходного файла задано некорректно. Это происходит, например, если в имени файла указаны метасимволы "?" или "*", поскольку использовать эти символы в командной строке команды ASM не разрешается.

SOURCE FILE READ ERROR. Выдается на консоль в том случае, если в процессе чтения и ассемблирования исходного файла программа "ASM" обнаружит некорректные символы. Программа "ASM" считает символ некорректным и не понимает его в том случае, если он является непечатаемым. К таким символам относятся специальные команды, которые вводятся при использовании клавиши CTRL или ESC. Эти символы могут быть вставлены в текст или умышленно, или случайно при копировании, объединении или при нарушении защиты файлов. Не всегда можно воспользоваться командой TYPE, чтобы обнаружить такие символы в тексте. При выводе текста, содержащего эти символы, на консоль с экраном или головкой печатающего механизма скорее всего будут происходить необычные вещи. При этом можно приблизительно определить, где расположены некоторые из этих символов. Для того чтобы удалить их, воспользуйтесь текстовым редактором. В других случаях при использовании команды TYPE эти символы не окажут влияния на работу устройства CRT. Их можно обнаружить только с помощью текстового редактора или команды DDT.

OUTPUT FILE WRITE ERROR. Выдается на консоль в том случае, если программа "ASM" пытается записать результирующие ".HEX" и ".PRN" файлы на диск (диски), статус защиты которых "R/O" ("Только чтение"). Кроме того, оно выдается, когда на диске нет свободного места. Прежде чем ввести команду ASM повторно, следует присвоить статус защиты диску (дискам) "R/W" ("Чтение/запись") или же обеспечить на диске необходимое свободное пространство (за счет копирования или удаления некоторых существующих файлов).

CANNOT CLOSE FILE. Выдается на консоль в том случае, если ОС *CP/M* обнаружит на диске указанный файл, но не сможет записать в него никаких данных, поскольку статус защиты этого файла "R/O" ("Только чтение"). Перед повторным использованием команды ASM присвойте файлу статус защиты "R/W" ("Чтение/запись").

NO DIRECTORY SPACE. Выдается на консоль в том случае, если программа "ASM" обнаружит, что число свободных входов в каталог указанного диска недостаточно для того, чтобы записать на этот диск окончевые ".PRN" и ".HEX" файлы. Это сообщение вовсе не означает, что объем свободного про-

странства на диске недостаточен, а говорит лишь о реальном числе входов в каталог (обычно число входов в каталог любого диска в ОС *CP/M* не должно превышать 64). Для того чтобы разрешить эту проблему, замените указанный диск (или диски) другим диском, каталог которого содержит достаточное число свободных входов. Помимо этого можно скопировать и удалить существующие файлы на имеющихся дисках с тем, чтобы освободить в каталоге нужное число элементов.

СООБЩЕНИЯ ОБ ОШИБКАХ В ИСХОДНОЙ ПРОГРАММЕ

Если в процессе ассемблирования программа "ASM" обнаружит в исходном тексте ошибки, то сообщения об этих ошибках будут выдаваться на консоль по мере их обнаружения до завершения процесса ассемблирования. Сообщения об ошибках содержат специальные коды, по которым можно определить тип ошибки. Эти коды также выдаются в соответствующих строках распечатки в файле ".PRN".

Программа "ASM" выдает сообщения об ошибках в следующем формате:

C	ШЕСТНАДЦАТИЧНЫЙ КОД	МЕТКА	ОПЕРАЦИЯ	ОПЕРАНД ;КОММЕНТАРИЙ
L	0100 214601	SORT	LXI	H, SW; ;ЗАГРУЗИТЬ АДРЕС КЛЮЧА В РЕГИСТР H,

где символ "C" обозначает код ошибки. Код ошибки всегда распечатывается в первой позиции строки, содержащей ошибку. За ним через пробел следует четырехзначный шестнадцатеричный адрес команды. Следом за адресом печатается реальный шестнадцатеричный код команды, который получен в результате ассемблирования и который обычно состоит из всех нулей в зависимости от серьезности ошибки. Если значение этого кода больше, чем "0000", то не стоит заблуждаться. Программа "ASM" всегда будет пытаться ассемблировать команду до тех пор, пока не обнаружит ошибку. Этот код обычно обозначает, что команда была отассемблирована лишь частично. Программа "ASM" выдает следующие сообщения об ошибках:

D – Ошибка в данных. Эта ошибка возникает в тех случаях, когда выражение в исходной строке не соответствует назначенному ему области данных. Обычно это означает, что выражение слишком длинное и его необходимо по возможности сократить.

E – Ошибка в выражении. Это сообщение означает, что выражение в исходной строке сформировано неправильно или что значение этого выражения слишком длинное и не может быть записано в указанную область данных.

L – Ошибка в символьском имени. Эта ошибка возникает при неправильном использовании символьского имени. Наиболее распространенной причиной такой ошибки является спецификация метки более чем в одной исходной строке.

N – Средство не реализовано. Эта ошибка возникает в тех случаях, когда программа "ASM" обнаруживает в исходном тексте директиву, которую она опознает, но не может обработать. Наиболее вероятной причиной такой ошибки является то, что в исходной программе используются директивы макроассемблера "MAC" фирмы Digital Research. Смысл данного сообщения, следовательно, заключается в том, что указанная исходная строка содержит ди-

рективу, которую может обработать только программа "MAC", так как программа "ASM" не располагает соответствующими для этого средствами.

О – Ошибка переполнения. Эта ошибка возникает в тех случаях, когда программа "ASM" обнаруживает в исходном тексте строку, содержащую слишком сложное выражение, которое нельзя обработать. Для того чтобы разрешить эту проблему, разбейте выражение на более мелкие части, или же уменьшите число операторов в данном выражении.

Р – Ошибка фазы. Эта ошибка возникает в тех случаях, когда значение символического имени в исходной программе изменяется некорректным образом. Для изменения значения символического имени следует применять директиву SET. Эта ошибка может также возникнуть, если в процессе асSEMBлирования программы "ASM" обнаружит повторное определение символического имени.

R – Ошибка в имени регистра. Это сообщение означает, что в исходной строке указано имя регистра, которое не соответствует мнемоническому. Если, например, задано выражение "POP A", то в результате появится сообщение об ошибке в имени регистра, поскольку в языке ассемблера для микропроцессора Intel 8080 это выражение является недопустимым.

S – Ошибочный символ. Это сообщение означает, что в поле комментариев исходной строки использован недопустимый символ (например, символ "?").

U – Неопределенное символическое имя. Эта ошибка возникает в тех случаях, когда в выражении обнаружено символическое имя с неопределенным значением. Например, если задано выражение "LXI CONT", а символическое имя "CONT" не определено, то соответствующая исходная строка будет содержать код этой ошибки.

V – Ошибка в значении. Эта ошибка имеет место в тех случаях, когда программа "ASM" обнаруживает выражение или operand, которые указаны неправильно. Обычно это происходит в результате типографских ошибок, таких как отсутствие запятой или других необходимых специальных символов.

ФОРМАТ ИСХОДНОГО ТЕКСТА ПРОГРАММЫ

При написании программ для микропроцессора Intel 8080, которые предполагается асSEMBлировать при помощи команды ASM, необходимо соблюдать определенные требования относительно формата исходного текста программы. Эти требования не обязательно совпадают с требованиями, предъявляемыми другими версиями языка ассемблера для микропроцессора Intel 8080. **Формат исходной строки.** В языке ассемблера каждая исходная строка (предложение) может содержать от одного до пяти полей. Эти поля (группы символов) отделяются друг от друга при помощи пробелов или символов табуляции. Формат этих полей следующий:

НОМЕР СТРОКИ	МЕТКА	ОПЕРАЦИЯ	ОПЕРАНД	;	КОММЕНТАРИЙ
0010	SORT:	LXI	H, SW	;	ЗАГРУЗИТЬ АДРЕС КЛЮЧА В РЕГИСТР H

Программа "ASM" расставляет ограничители табуляции в каждой восьмой позиции вдоль командной строки. Таким образом, ограничители табуляции расставляются в позициях 1, 9, 17, 25, 33 и т. д.

Каждое из указанных выше пяти полей описывается в следующих разделах:

1. Номера строк. Номера строк представляют собой целые десятичные числа и являются необязательным полем исходной строки. Если номера присутствуют, то они должны быть расположены в начале строки. Номера строк являются только справками для программиста. В процессе ассемблирования они игнорируются.

2. Метки. Метка является идентификатором (символическим именем), который используется для представления некоторого адреса или конкретного значения. Метка может быть длиной от 1 до 16 символов и должна оканчиваться двоеточием (":"). Для облегчения чтения исходного текста в метке можно использовать знак денежной единицы ("Q"). Все символы "Q", обнаруженные в поле метки, программа "ASM" игнорирует. Этот символ не рассматривается как составной элемент символического имени. Метки являются необязательными элементами для всех операторов за исключением директив EQU (приравнять) и SET. Этим двум директивам всегда должна предшествовать метка. Существует несколько зарезервированных слов, которые нельзя использовать в поле метки, поскольку они имеют другое значение. Этими словами являются:

- a) Все мнемонические обозначения команд микропроцессора Intel 8080.
- b) Все имена регистров: A, B, C, D, E, H, L, M, SP и PSW.
- v) Все директивы программы "ASM": ORG, EQU, SET, DB, DS, DW, IF, ENDIF и END.

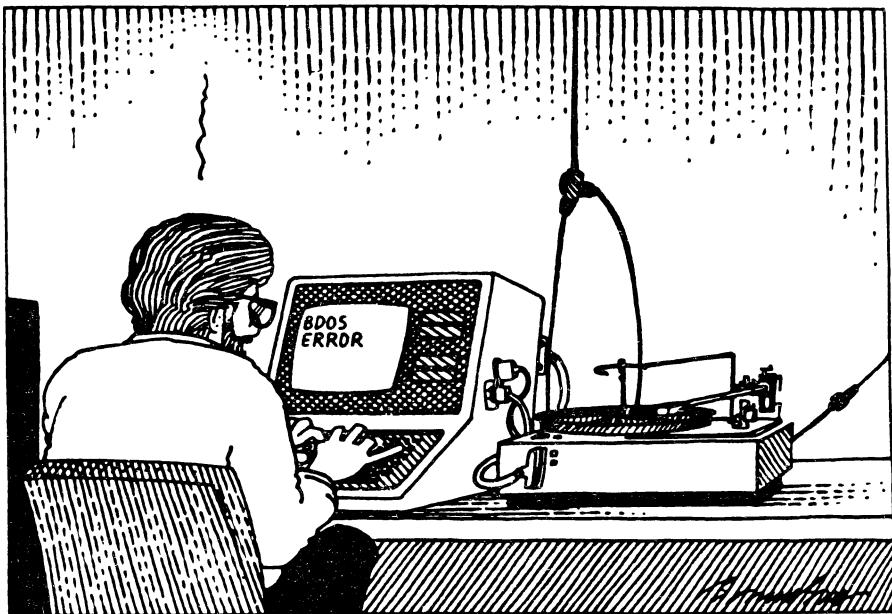
3. Операция. Поле операции должно присутствовать в каждой исходной строке. Оно является обязательным. Это поле может содержать или мнемоническое обозначение команды микропроцессора Intel 8080 или директиву программы "ASM".

4. Операнды. В зависимости от заданной операции поле операнда может быть либо пустым, либо содержать один или два операнда. В качестве операнда можно использовать константу, метку или выражение. Директивы программы "ASM" в основном требуют наличия одного или двух операндов в этом поле.

5. Комментарии. Поле комментария может быть использовано программистом для описания функции соответствующего оператора. Это поле не является обязательным, но если оно используется, то должно начинаться с точки с запятой (";"). Когда программа "ASM" обнаруживает в исходной строке точку с запятой, она игнорирует весь расположенный после нее текст и рассматривает его как комментарий.

КОНСТАНТЫ И ВЫРАЖЕНИЯ

Ниже коротко описываются правила использования констант и выражений в программе, предназначеннй для ассемблирования с помощью команды ASM.



Константы. Существует два основных типа констант, которые можно размещать в поле операнда. Это числовые и строковые константы, которые описаны ниже.

1. Числовые константы. Числовая константа представляет собой фиксированное число по одному из следующих четырех оснований:

а) **Двоичному (основание 2).** Двоичная числовая константа представляет собой последовательность цифр "0" и "1", за которой должен следовать символ "B".

б) **Восьмеричному (основание 8).** Восьмеричная числовая константа представляет собой последовательности цифр, каждая из которых может быть от "0" до "7". Последовательность также завершается символом "O" или "Q".

в) **Десятичному (основание 10).** Десятичная числовая константа представляет собой последовательность цифр, каждая из которых может быть в интервале от "0" до "9". Для ясности в конце последовательности можно дополнительно поставить символ "D". Это делать не обязательно, поскольку программа "ASM" любую числовую константу без символа основания счисления рассматривает как десятичное число.

г) **Шестнадцатеричному (основание 16).** Шестнадцатеричная числовая константа представляет собой последовательность цифр и символов (цифры – в интервале от "0" до "9" и символы от "A" до "F"). Константа этого типа должна начинаться с цифры (от "0" до "9"). Эта последовательность должна заканчиваться символом "H", который можно не указывать, если первой цифрой в последовательности является "0", поскольку в каждом случае программа "ASM" автоматически рассматривает константу как шестнадцатеричное число.

2. Строковые константы. Строковая константа представляет собой последовательность символов, заключенную в апострофы ('). Константы этого ти-

па должны быть длиной не более 64 символов и должны содержать только печатаемые символы (использование непечатаемых управляющих символов не допускается). Если данная строка должна включать в себя апостроф, то следует ввести последовательность из двух апострофов. Это необходимо сделать потому, что программа "ASM" проигнорирует первый апостроф, а второй рассмотрит как литерал. Отметим, что программа "ASM" не осуществляет преобразований символов нижнего регистра в символы верхнего регистра. Значение строковой константы программа "ASM" вычисляет путем добавления старшего нулевого бита семибитовому (ASCII) коду каждого символа.

Выражения. Программа "ASM" допускает использование выражений в поле операнда. Выражение может быть составлено из констант, символьических имен, арифметических операторов, логических операторов и скобок. В процессе ассемблирования программа "ASM" преобразует выражение и вычисляет его значение. Программа "ASM" выполняет эти операции над 16-битовыми числами без знака и выдает результат также в виде 16-битового числа без знака. Программа "ASM" поддерживает следующие арифметические и логические операторы:

1. Арифметические операторы. В процессе преобразования выражений программа "ASM" может выполнять простейшие арифметические действия. В сочетании с идентификаторами и константами в выражении могут быть использованы следующие операторы:

- Сложение (выражение " $A + B$ " эквивалентно арифметической сумме значений A и B).
- Вычитание (выражение " $A - B$ " обозначает вычитание значения B из значения A).
- Умножение (выражение " $A * B$ " обозначает умножение значения A на значение B).
- Деление (выражение " A/B " обозначает целочисленное деление значения A на значение B, а выражение " $A \text{ MOD } B$ " обозначает остаток от деления значения A на B).
- Унарный плюс (выражение "+ A" эквивалентно выражению "A").
- Унарный минус (выражение "-A" эквивалентно выражению "0 - A").

2. Логические операторы. Кроме того, программа "ASM" может выполнять логические (булевы) операции. В ней поддерживаются следующие логические операторы:

- NOT (выражение "NOT B" равно обратному значению или дополнению выражения "B").
- AND (значением выражения " $A \text{ AND } B$ " является результат логической операции "И" над значениями A и B).
- OR (значением выражения " $A \text{ OR } B$ " является результат логической операции "ИЛИ" над значениями A и B).
- XOR (значением выражения " $A \text{ XOR } B$ " является результат логической операции "Исключающее ИЛИ" над значениями A и B).
- SHL и SHR (результатом выражения " $A \text{ SHL } B$ " является сдвиг значения A влево на число битов, установленное значением B. Сдвинутые старшие разряды теряются, а освободившиеся младшие разряды заполняются нулями. Выражение " $A \text{ SHR } B$ " таким же образом обеспечивает сдвиг вправо).

ДИРЕКТИВЫ ПРОГРАММЫ "ASM"

Существует несколько специальных инструкций, которые не входят в стандартный набор команд микропроцессора Intel 8080, но которые можно использовать в программах, асSEMBлируемых с помощью команды ASM . Этими инструкциями являются так называемые "директивы асSEMBлера". Эти инструкции используются для управления процессом асSEMBлирования и, следовательно, оказывают влияние на формирование результирующего машиного кода.

Строки исходного текста, содержащие директивы асSEMBлера, имеют в основном тот же формат, что и строки с обычными операторами. Имя директивы размещается в поле операции соответствующей исходной строки. Может сложиться впечатление, что эти директивы используются и другими асSEMBлерами. Важно, однако, знать, что асSEMBлер "ASM" обрабатывает только те директивы, которые описаны в этом разделе.

ORG. Директива ORG (от английского Origin – начальный адрес) указывает программе "ASM" адрес в памяти, начиная с которого необходимо размещать результаты асSEMBлирования последующих операторов. Исходная строка, содержащая директиву ORG, должна иметь следующий формат:

LABEL ORG EXPRESSION ; COMMENTS.

Здесь поле "EXPRESSION" содержит адрес памяти следующей инструкции или директивы. Необязательное поле "LABEL" может содержать символическое имя, которому будет присвоено значение выражения, заданного в поле "EXPRESSION". В программе директиву ORG можно использовать произвольное число раз.

EQU. Директива EQU (от английского Equate – приравнять) используется для присваивания символическому имени определенного значения. Исходная строка, содержащая эту директиву, имеет следующий формат:

LABEL EQU EXPRESSION ; COMMENTS.

Здесь поле "EXPRESSION" может содержать любое допустимое число, адрес, константу или выражение. В исходных строках, содержащих директиву ORG, поля "LABEL" и "EXPRESSION" являются обязательными. Как указывалось выше, поле "EXPRESSION" наряду с операторами и числовыми константами может содержать символические имена.

SET. Директива SET (от английского Set – назначить) используется для присваивания данному символическому имени определенного значения. Эту директиву можно, кроме того, использовать для переопределения символического имени (с помощью директивы EQU значение символическому имени может быть присвоено только один раз). Одно и то же символическое имя может использоваться совместно с директивой SET несколько раз. Исходная строка с директивой SET должна иметь следующий формат:

LABEL SET EXPRESSION ; COMMENTS.

В исходной строке, содержащей эту директиву, оба поля "LABEL" и "EXPRESSION" являются обязательными.

IF и **ENDIF.** Директивы IF (от английского If – если) и ENDIF (от английского "End IF" – "Конец IF") описывают соответственно начало и конец секции исходной программы, которую необходимо асSEMBлировать, если

только оператор IF правильно задан в исходной строке. Исходная строка с директивой IF является первой строкой данной сёкции, а строка с директивой ENDIF — последней. Эти директивы вводятся в следующем формате:

IF EXPRESSION

...

(исходные строки, асSEMBЛИруемые по условию)

...

ENDIF

В поле "EXPRESSION" вводятся условия, при которых оператор IF должен быть выполнен. Если поле "EXPRESSION" содержит все нули, то все строки, расположенные между операторами IF и ENDIF, игнорируются. Если в этом поле содержится величина больше нуля, то обусловленные строки будут асSEMBЛИроваться.

DB, DW и DS. Существует три директивы, которые используются для организации в оперативной памяти областей под данные. Этими директивами являются:

DB. Директива DB (от английского Define Byte – определить байт) используется для "побайтного" описания областей данных. Исходные строки, содержащие эту директиву, имеют следующий формат:

LABEL DB EXP1, EXP2, EXP3 ; COMMENTS.

Поле "EXP1, EXP2, EXP3" может содержать произвольное количество выражений, разделенных запятой. При обработке такой исходной строки программа "ASM" последовательно вычисляет значение каждого выражения и запоминает их в 8-битовом (1 байт) формате в очередной ячейке памяти. В исходной строке, содержащей директиву DB, оба поля "LABEL" и "COMMENTS" являются необязательными.

DW. Директива DW (от английского Define Word – определить слово) также используется для описания областей в памяти по одному машинному слову (2 байта) за раз. Исходные строки с этой директивой имеют следующий формат:

LABEL DW EXP1, EXP2, EXP3 ; COMMENTS.

Поле "EXP1, EXP2, EXP3" может содержать произвольное количество выражений, разделенных запятой. При обработке этой исходной строки программа "ASM" последовательно вычисляет значение каждого выражения и запоминает его в 16-битовом (одно слово или 2 байта) формате в очередных двух ячейках памяти. При этом первым запоминается младший байт, а следом за ним – старший байт. В исходной строке, содержащей директиву DW, оба поля "LABEL" и "COMMENTS" являются необязательными.

DS. Директива DS (от английского Define Storage – определить область памяти) используется для резервирования в памяти области указанного размера. Исходные строки, содержащие эту директиву, имеют следующий формат:

LABEL DS EXP1 ; COMMENTS.

Поле "EXP1" содержит единственное выражение, значение которого определяет объем резервируемой памяти. В исходной строке с директивой DS оба поля "LABEL" и "COMMENTS" являются необязательными.

В исходных строках, содержащих директивы DB, DW и DS, поле выражений может содержать любые допустимые выражения, символьические имена, числовые или линейные константы и т. д. Совместно с директивами DB и DW можно задать произвольное количество выражений при условии, что все они разделяются запятыми. Совместно с директивой DS можно задавать только одно выражение.

На этом описание программы "ASM" заканчивается. Более детальную информацию о программировании на языке Ассемблер для микропроцессора Intel 8080 можно получить из соответствующих книг по данному вопросу. Список некоторых имеющихся публикаций приведен в приложении Г этой книги.

ГЛАВА 14

Версии ОС *CP/M*:

1.3 1.4 2.0 2.2

Тип команды:

транзитная



LOAD

ОБЩИЕ СВЕДЕНИЯ

Эта команда обеспечивает обращение к транзитной программе ОС *CP/M*, которая также называется "LOAD". Программа-утилита "LOAD" позволяет пользователям программ "ASM" и "DDT" преобразовывать файлы с шестнадцатеричным кодом программ (с расширением ".HEX"), созданные ассемблером "ASM", в файлы с исполняемым машинным кодом (с расширением ".COM").

ОБРАЩЕНИЕ К ПРОГРАММЕ "LOAD"

Для обращения к программе "LOAD" необходимо, чтобы файл "LOAD.COM" хранился на диске, вставленном в одно из устройств ЭВМ. Для того чтобы обработать конкретный файл, необходимо в качестве ответа на промпт с именем текущего диска (A >, B > и т. д.) ввести команду LOAD и через пробел спецификацию файла. Спецификация файла не должна включать расширение ".HEX", поскольку программа "LOAD" будет автоматически искать файл с указанным именем, и если он имеет корректное расширение, считывать его содержимое.

После ввода командной строки программа "LOAD" выдает информацию, относящуюся к указанному файлу. В следующем примере показано, как использовать команду DIR для проверки наличия файла на диске. Помимо этого здесь продемонстрировано использование команды LOAD для того, чтобы преобразовать шестнадцатеричный код из файла ".HEX" в исполняемый машинный код и создать таким образом исполняемый файл ".COM":

```
A>DIR TEST01.*  
A : TEST01 ASM : TEST01 HEX  
A : TEST01 FRN  
  
A>  
A>LOAD TEST01  
  
FIRST ADDRESS 0100  
LAST ADDRESS 01F8  
BYTES READ 00F9  
RECORDS WRITTEN 02  
  
A>  
A>DIR TEST01.*  
  
A : TEST01 ASM : TEST01 HEX  
A : TEST01 FRN : TEST01 COM  
  
A>
```

В этом примере показано создание исполняемого кода программы "TEST01". До команды LOAD для того чтобы проверить, что все необходимые файлы программы "TEST01" находятся на диске, используется команда DIR. После завершения выполнения программы "LOAD" команда DIR применяется повторно для проверки того, что файл "TEST01.COM" корректно записан на диск. В процессе выполнения программы "LOAD" выдается следующая информация:

FIRST ADDRESS: Адрес первой ячейки программы.
LAST ADDRESS: Адрес последней ячейки программы.
BYTES READ: Общая длина программы в байтах.
RECORDS WRITTEN: Общее количество 128-байтовых записей, записанных в результирующий файл.

Отметим, что файл с шестнадцатеричным кодом ("HEX") должен содержать записи с адресами в *возрастающем порядке*. Если это не так, то исполняемый ("COM") файл будет создан программой "LOAD" некорректно.

ПРИМЕЧАНИЕ ДЛЯ ПРОГРАММИСТОВ

При создании с помощью команды LOAD исполняемого кода программы на языке Ассемблер убедитесь, что начальный адрес программы больше или равен шестнадцатеричному адресу 0100. В противном случае при попытке создать по команде LOAD исполняемый код такой программы возникает ошибка. Начальный адрес программы устанавливается с помощью директивы ORG и используется для того, чтобы информировать ОС CP/M о расположении программы в памяти. При загрузке в память и выполнении программы ее необходимо будет разместить по указанному начальному адресу и передать ей управление. С адреса 0100 обычно начинается область транзитных программ ОС CP/M.

Версии ОС *CP/M*:

1.3 1.4 2.0 2.2

Тип команды:

транзитная

ОБЩИЕ СВЕДЕНИЯ

Эта команда обеспечивает обращение к транзитной программе ОС *CP/M*, которая также называется "DDT". Программа "DDT" позволяет пользователю загружать в оперативную память, просматривать, тестировать и изменять и отлаживать любую программу на языке ассемблера, представленную в машинном коде. Помимо этого программа "DDT" допускает просмотр содержимого текстовых файлов любого типа. Программа "DDT" работает в динамическом режиме, что позволяет пользователю запускать находящуюся в памяти отлаживаемую программу и отслеживать каждый шаг ее выполнения. С ее помощью можно также вносить небольшие изменения в существующие программы (такие, как текстовый процессор "WordStar") или реассемблировать их с тем, чтобы понять, как они работают. Эта программа является весьма сложной. Программа "DDT" представляет собой важнейшее инструментальное средство для программистов, использующих язык ассемблера. Для эффективного использования этой программы необходимо понимание принципов программирования на языке ассемблера. Программа "DDT" содержит также встроенный мини-ассемблер, который представляет собой "ослабленную" версию ассемблера "ASM". В [6] программа "DDT" используется для обучения технике программирования на языке ассемблера и модификации секций драйверов ввода-вывода (BIOS) операционной системы *CP/M*.

Ниже коротко перечислены основные возможности программы "DDT":

- Заполнение указанного участка памяти константами, т. е. ввод определенных символов в последовательно расположенные ячейки памяти.
- Создание блока управления файлом по умолчанию (FCB) путем указания его имени и считывание названного файла в оперативную память блока FCB.
- Вывод на консоль содержимого областей памяти как в шестнадцатеричном, так и в ASCII форматах.
- Реассемблирование части объектного кода и вывод результатов на консоль в коде ASCII.
- Вывод в шестнадцатеричном формате и изменение содержимого области памяти, расположенной по заданному адресу.
- Вставка операторов языка ассемблера в любой точке программы.
- Копирование содержимого заданной области памяти по указанному адресу.
- Исполнение части программы, установление точек прерывания (вставляемых в программу для остановки и возврата в "DDT") и назначение дополнительных точек останова для просмотра содержимого регистров центрального процессора.

- Пошаговая трассировка выполнения заданного числа команд или всей программы.
- Вывод и, возможно, изменение текущего содержимого регистров центрального процессора.
- Использование встроенного калькулятора для сложения и вычитания шестнадцатеричных чисел.

Указанные выше функции выполняются различными командами, которые вводятся с консоли, пока программа "DDT" находится в оперативной памяти.

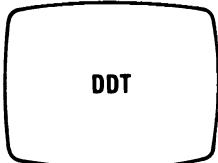
ОБРАЩЕНИЕ К ПРОГРАММЕ "DDT"

Для того чтобы обратиться к программе "DDT", необходимо, чтобы файл "DDT.COM" хранился на диске, вставленном в одно из устройств ЭВМ. По команде DDT эта программа будет загружена в оперативную память в секцию "Процессора команд пульта", тем самым заменяется содержимое, обычно хранящееся в этой секции. По сравнению с остальными командами ОС CP/M команда DDT обладает уникальными свойствами. Хотя программа "DDT" является "транзитной программой", она не загружается в область транзитных программ по шестнадцатеричному адресу 100. Более того, она загружается перед областью транзитных программ так, что она перекрывает область секции CCP операционной системы CP/M*. Область транзитных программ можно затем использовать для загрузки и выполнения программ пользователя, отладка которых производится с помощью программы "DDT". Файл с машинным кодом, загружаемый в оперативную память, размещается в области транзитных программ, где под управлением программы "DDT" может просматриваться, изменять его содержимое.

В процессе загрузки файла в оперативную память программа "DDT" "перекрывает" прежнее содержимое области транзитных программ содержимым указанного файла. Вследствие этих необычных свойств для завершения программы "DDT" следует использовать команду CTRL/C (реинициализация системы) или команду G (описание команды G представлено ниже). В результате выполнения одной из этих команд стандартное содержимое секции CCP будет вновь загружено с диска в оперативную память.

Ниже описываются два способа обращения к программе "DDT" и загрузки ее в оперативную память.

**ЗАГРУЗКА ПРОГРАММЫ "DDT" В ПАМЯТЬ
БЕЗ УКАЗАНИЯ ОТЛАЖИВАЕМОЙ ПРОГРАММЫ**



DDT

Программу "DDT" можно загрузить в оперативную память без указания имени файла, предназначенного для проверки и модификации. Для этого в качестве ответа на промпт с именем текущего диска (A >, B > и т. д.) необходимо

* Фактически временно выполняет некоторые функции ОС CP/M. – Прим. перев.

димо просто ввести команду DDT и нажать клавишу "RETURN". При этом программа "DDT" будет загружена в оперативную память в секцию CCP, а содержимое области транзитных программ сохраняется прежним. Этот случай проиллюстрирован в следующем примере:

```
A>DDT  
DDT VERS X.X
```

Как только программа "DDT" загружена в память, выводится сообщение с именем и номером версии (в предыдущем примере номер версии был заменен символами "X.X"). Номер версии соответствует версии программы "DDT", хотя сам по себе он обычно отражает номер версии ОС *CP/M* (1.4, 2.2 и т. д.).

ПРИМЕЧАНИЕ ДЛЯ ПРОГРАММИСТОВ

Другие сообщения о загрузке программы "DDT"

В некоторых специальных версиях ОС *CP/M* программа "DDT" модифицирована соответственно тому режиму, в котором она функционирует. Эта модификация иногда отмечается специальным символом после слова "DDT". При этом сообщение о загрузке программы "DDT" выдается в формате "DDT - X", где символ "X" имеет одно из следующих значений:

- D – стандартная версия фирмы Digital Research.
- M – версия MDS фирмы Intel.
- I – стандартная версия фирмы IMSAI.
- O – версия фирмы Omcron Systems.
- S – версия фирмы Digital Systems.

После того как программа "DDT" загружена в память, выводится символ "–", который является промптом программы "DDT" (см. предыдущий пример). В качестве ответа на этот промпт можно ввести команды программы "DDT".

Для загрузки с диска файла необходимо использовать команды I (вставить имя) и R (читать содержимое файла). Команды I и R удобны тем, что они позволяют создать блок FCB, поэтому файлы можно легко загрузить в память в любой момент.

ЗАГРУЗКА ПРОГРАММЫ "DDT" С УКАЗАНИЕМ
ОТЛАЖИВАЕМОЙ ПРОГРАММЫ

DDT FILENAME.COM
DDT FILENAME.HEX
DDT FILENAME.EXT

Вместо того чтобы после загрузки программы "DDT" выполнять последовательность команд для загрузки в оперативную память файла, программу "DDT" и требуемый файл можно загрузить в память одновременно. Для этого после команды DDT необходимо через пробел ввести имя файла и его расширение. Обычно совместно с программой "DDT" в память загружаются

файлы, содержащие транзитные программы (файлы ".COM"), или файлы с шестнадцатеричным кодом (файлы ".HEX"), однако программу можно использовать и для просмотра текстовых (*.TXT) файлов. Файлы с расширениями ".HEX" и ".COM" всегда содержат машинные коды программ, и программа "DDT", таким образом, предоставляет пользователю средства для тестирования и исправления программ. Совместно с программой "DDT" в оперативную память можно также загрузить любой другой файл, содержащий машинный код или текст. Сюда входят файлы с данными (расширение ".DAT" и проч.), оверлейные файлы (расширения ".OVR", ".OVL" и проч.), файлы с документацией (расширение ".TXT") и так далее.

После того как программа "DDT" и указанный файл загружены в оперативную память, выводится информация о состоянии памяти указанного файла, загруженного в оперативную память. Ниже приводится пример такого сообщения:

```
A>DDT TEST01.COM  
DDT VERS X.X  
NEXT PC  
1E00 0100
```

После начального сообщения о загрузке программы "DDT" выводится некоторая информация, связанная с состоянием той области памяти, куда загружено содержимое файла "TEST01.COM". В этом примере число, расположеннное под заголовкой "NEXT", представляет собой адрес следующей свободной ячейки памяти (после загрузки программы) в области транзитных программ. Иными словами, это адрес первой свободной (неиспользованной) ячейки памяти, расположенной за последней ячейкой, занятой под загруженную программу. Число, расположенное под заголовком "PC" (сокращение от английского Program Counter – счетчик команд), представляет собой текущее, установленное в программе "DDT", значение счетчика команд. Это значение информирует пользователя о месте в памяти программы "DDT", с которого можно выполнить проверку и изменение данных. Это "место" представляет собой адрес в памяти и во многом напоминает указатель текущего символа программы "ED". Программа "DDT" использует указатель PC как точку ссылки, после которой пользователь получает доступ к ячейкам памяти.

В оставшейся части этого раздела описаны различные команды, которые можно использовать совместно с программой "DDT".

ОПИСАНИЕ КОМАНД

ВЫВОД НА КОНСОЛЬ СОДЕРЖИМОГО
ОБЛАСТЕЙ ПАМЯТИ

- D
- Dstart
- Dstart,end

Команда D (от английского Display – вывести) используется для выдачи содержимого некоторой области памяти в шестнадцатеричном и ASCII фор-

матах. Обычно эта команда используется первой, когда программа "DDT" загружается совместно с файлом в оперативную память, и пользователю нужно просмотреть определенные ячейки памяти. Если команда D задана без параметров, то программа "DDT" выдает следующие за счетчиком команд PC 192 байт. Этот случай проиллюстрирован в следующем примере. Здесь счетчик команд PC содержит шестнадцатеричный адрес 100, который программа "DDT" по умолчанию присваивает ему сразу после ее загрузки в оперативную память:

```
A>DDT TEST01.COM
DDT VERS X.X
NEXT PC
0200 0100
-D
0100 21 00 00 39 22 FA 01 31 3C 02 AF 32 F9 01 0E 0F !..9" ..1<..2..
0110 11 5C 00 CD 05 00 11 D5 01 3C CA 77 01 01 00 00 \.....<.W..
0120 C5 0E 14 11 5C 00 CD 05 00 C1 B7 C2 67 01 21 80 .....\\.....G..
0130 00 7E E6 7F FE 1A CA 67 01 FE 20 CA 59 01 FE 0D ^....G....Y.
0140 CA 59 01 FE 0A CA 59 01 3A F9 01 FE 00 CA 5E 01 .Y....Y:.....
0150 03 3E 00 32 F9 01 C3 5E 01 3E 01 32 F9 01 23 7C >..2...>..2..
0160 B7 CA 31 01 C3 20 01 11 5C 00 C5 0E 10 CD 05 00 ..1.....\.....
0170 C1 CD B1 01 11 E5 01 0E 09 CD 05 00 2A FA 01 F9 .....*.
0180 C9 C5 E1 06 00 11 F0 D8 CD A5 01 11 18 FC CD A5 .....>.
0190 01 11 9C FF CD A5 01 11 F6 FF CD A5 01 70 C6 30 .....>.
01A0 4F C0 C8 01 C9 0E 2F OC 19 DA A7 01 7A 2F 57 7B 0.....\....Z/
01B0 2F 5F 13 19 79 FE 31 D2 C2 01 78 B7 79 C8 CD C8 `...Y.1...X.Y.
```

В этом примере после ввода команды D программа "DDT" распечатала в шестнадцатеричном формате содержимое первых 192 ячеек памяти. Числа, расположенные в самой левой колонке, представляют собой начальные адреса каждой строки. После начального адреса в каждой строке следует 16 шестнадцатеричных чисел, которые представляют собой значения ячеек, расположенных по соответствующему адресу. Например, в первой строке сразу после адреса "0100" стоит число "21", которое представляет собой значение, хранящееся в ячейке памяти по адресу 100. Значения ячеек памяти расположены в порядке слева направо. Например, первая строка текста начинается с адреса 0100. По мере перемещения вправо адреса ячеек увеличиваются от 0100 до 010F в шестнадцатеричном формате (всего 16 ячеек).

Самая правая часть отчета представляет собой последовательность алфавитно-цифровых символов, следующих вперемежку с точками. В этом поле распечатывается (в коде ASCII) представление соответствующих данных символами. Символ на первой позиции этого поля является представленным в коде ASCII содержимым ячейки памяти и распечатанным в соответствующей строке на первой позиции. Второй символ является представленным в коде ASCII содержимым ячейки памяти и распечатан в той же строке вторым по счету и т. д. Это поле содержит только печатаемые символы кода ASCII. Эти символы в шестнадцатеричном коде имеют значение от 32 до 7E. Любые символы с меньшими (от 00 до 31) или большими (от 7F до FF) значениями распечатываются в виде точек. Описание символов в коде ASCII см. в приложении Е. Обычно значения, не соответствующие коду ASCII, относятся к программным кодам, которые полностью используют все 8 бит. Например, если совместно с программой "DDT" в память загружен фрагмент некоторого текста, то правая часть отчета, полученного по команде D, будет содержать

только печатаемые символы кода ASCII и, наоборот, преобразование точек в правой части отчета команды D говорит о том, что на консоль выведен фрагмент программы.

Существуют и другие варианты использования команды D. Вместо печати содержимого 192 ячеек памяти, начиная с той, на которую установлен счетчик команд PC, по команде D можно распечатать содержимое 192 ячеек, начиная с любого адреса по выбору. Если после команды D указан начальный адрес, то программа "DDT" распечатает содержимое 192 ячеек памяти, начиная с этого адреса. Начальный адрес может иметь любое значение в интервале от 0000 до FFFF, т. е. указывать на любую ячейку памяти системы. Использование команды D в этом формате проиллюстрировано в следующем примере:

```
A>DDT TEST01.COM
DDT VERS X.X
NEXT PC
0200 0100
-I0130
0130 00 7E E6 7F FE 1A CA 67 01 FE 20 CA 59 01 FE 0D .^....6....Y...
0140 CA 59 01 FE 0A CA 59 01 3A F9 01 FE 00 CA 5E 01 .Y....Y:....".
!
V
01E0 2F 5F 13 19 79 FE 31 D2 C2 01 78 B7 79 C8 C0 C8 /...Y.1...X.Y...
-
```

Помимо этого по команде D можно вывести на консоль содержимое ячеек памяти, расположенных между указанными начальным и конечным адресами. Этот формат команды D напоминает предыдущий. Разница заключается в том, что после адреса первой ячейки необходимо ввести запятую и адрес последней ячейки. Использование этого формата команды D проиллюстрировано в следующем примере:

```
A>DDT TEST01.COM
DDT VERS X.X
NEXT PC
0200 0100
-D0100,012FF
0100 21 00 00 39 22 FA 01 31 3C 02 AF 32 F9 01 0E 0F !..9"..1(..2....
0110 11 5C 00 CD 05 00 11 B5 01 3C CA 77 01 01 00 00 .\.....<W...
0120 C5 0E 14 11 5C 00 CD 05 00 C1 B7 C2 67 01 21 80 ...\\.....G.!
-D0120,012F
0120 C5 0E 14 11 5C 00 CD 05 00 C1 B7 C2 67 01 21 80 ...\\.....G.!
-D0120,012A
0120 C5 0E 14 11 5C 00 CD 05 00 C1 B7 ...\\.....
-D0120,0122
0120 C5 0E 14 ...
```

Отметим, что в этом примере первая выданная на консоль порция данных представляет собой содержимое ячеек памяти, расположенных по адресу с 0100 по 12FF. На консоль выдано содержимое только указанных ячеек. Счетчик команд PC установлен на последнюю указанную ячейку. Следующая порция данных представляет собой содержимое ячеек памяти, расположенных по адресу с 0120 по 012F. Третий отчет содержит данные ячеек с адресами 0120 по 012A, тем самым выделена только необходимая часть из предыдущей строки. Если выдаваемая порция данных содержит менее 16 значений,

то весь отчет, включая представление данных символами, смещается влево.

По этой команде можно вывести содержимое произвольного участка памяти любого размера. При выводе больших порций данных можно использовать команду CTRL/S для управления перемещением данных на экране.

ЗАПОЛНЕНИЕ ПАМЯТИ КОНСТАНТАМИ

– Fstart,end,data

Команда F (от английского Fill – заполнить) используется для записи указанного значения в каждую ячейку памяти, расположенную между заданными начальным и конечным адресами. Эту команду удобно использовать в тех случаях, когда программа "DDT" загружается в оперативную память без файла. Указывая каждый раз при использовании этой команды различные данные, можно тестировать участки памяти путем заполнения их конкретными константами с последующим наглядным контролем за правильностью записи. Для проверки содержимого указанных участков памяти после команды F можно использовать описанную выше команду D.

Команда записи данных в память вводится следующим образом. Сначала в качестве ответа на промпт программы "DDT" вводится символ "F". Сразу после него должен следовать шестнадцатеричный начальный адрес, запятая, шестнадцатеричный конечный адрес, еще одна запятая и шестнадцатеричный код символа, который должен быть записан. После того как команда введена, предыдущее содержимое указанных ячеек памяти заменяется указанным значением.

Правильный синтаксис этой команды продемонстрирован в следующем примере:

Совместно с командой F можно указать любой двузначный шестнадцатеричный код в интервале от 00 до FF. В этом примере по команде F во все ячейки памяти с адресами 0100 по 01FF записано шестнадцатеричное число 26. Отметим, что указанный участок памяти целиком заполнен этим кодом. В коде ASCII шестнадцатеричное значение 26 используется для представления символа (знака) "&". Когда была использована команда D, поле данных в коде ASCII было сплошь заполнено этим символом.

При использовании команды F укажите начальный адрес, соблюдая осто-

рожность, с тем чтобы не затронуть области памяти, занятые под ОС *CP/M* и программу "DDT". Размер тех областей памяти, которые никогда нельзя заполнять, в основном определяется общим объемом оперативной памяти ЭВМ и размерами конкретной версии ОС *CP/M*. Согласно основному правилу, в команде F никогда нельзя указывать адреса первых 256 байт и последних 11К байт оперативной памяти. При записи данных в эти области памяти целостность ОС *CP/M* нарушается, а выполняемая программа (в данном случае программа "DDT") утрачивается. Помимо этого система скорее всего полностью выйдет из строя, что потребует ее полной перезагрузки.

Отметим, что в примере в качестве начального адреса указан адрес 0100. Обычно это адрес первой ячейки, доступной в области транзитных программ (ТРА).

ВСТАВКА В УПРАВЛЯЮЩИЙ БЛОК FCB ИМЕНИ
ЗАГРУЖАЕМОГО ФАЙЛА

– Ifilename.ext

Команда I (от английского Insert – вставить или Identify – отождествить) используется для того, чтобы вставить имя файла в блоке FCB (Блок управления файлом) для последующей загрузки этого файла в память. Эта команда предшествует команде R (читать), описанной ниже, поскольку при выполнении команды I содержимое файла фактически в память не записывается, а записывается в блок FCB только имя файла. Команда I всего лишь подготавливает файл к последующей загрузке по команде R.

Команда I используется в тех случаях, когда команда "DDT" задана без спецификации файла или же когда необходимо загрузить в память и просмотреть одно за другим содержимое различных файлов. Помимо этого, команды I и R можно использовать в тех случаях, когда файл уже загружен в оперативную память и необходимо, начиная с некоторого адреса, загрузить в память содержимое второго файла (например, изменения в программе). Это как раз та ситуация, которая возникает при модификации секции драйверов ввода-вывода (BIOS) ОС *CP/M* (см. [6]). Если команды I и R используются для загрузки в память файла с шестнадцатеричным или объектным кодом программы на языке ассемблера, для того чтобы внести изменения в программу, которая уже находится в памяти, то программа "DDT" автоматически разместит содержимое этого файла по нужному адресу. Причем эта операция выполняется независимо от того, где установлен счетчик команд (PC).

В следующем примере показан корректный синтаксис для команды I:

```
A>DDT
DDT VERS X.X
-I TEST01.COM
```

В этом примере после команды I указано имя файла "TEST01.COM". Имя файла всегда следует непосредственно за символом "I". Пробелов между символом "I" и именем файла быть не должно. После того как эта команда введена, снова появится промпт программы "DDT". Теперь можно ввести команду R (читать) так, как это описано ниже.

Файл, имя которого указывается после команды I, должен храниться на текущем диске. В этой командной строке перед именем файла нельзя указывать имя диска (A:, B: и т. д.), в противном случае возникает ошибка. Существует, однако, другой способ загрузки файла с диска, не являющегося текущим. Это достигается путем изменения содержимого ячейки памяти, расположенной в области, занятой программой "DDT", по шестнадцатеричному адресу 005C. Для того чтобы изменить содержимое этой ячейки памяти, используйте команду S, описанную ниже. Для определения кода, необходимо для представления имени требуемого диска, можно воспользоваться следующей таблицей.

КОДЫ ИМЕН ДИСКОВ, ИСПОЛЬЗУЕМЫЕ В БЛОКЕ УПРАВЛЕНИЯ ФАЙЛОМ (FCB)

Имя диска	Соответствующее значение ячейки памяти, расположенной по шестнадцатеричному адресу 005C	Имя диска	Соответствующее значение ячейки памяти, расположенной по шестнадцатеричному адресу 005C
Текущий диск		I	09
A	00	J	0A
B	01	K	0B
C	02	L	0C
D	03	M	0D
E	04	N	0E
F	05	O	0F
G	06	P	10
H	07		
	08		

Отметим, что набор допустимых имен дисков зависит от версии ОС CP/M. В ОС CP/M версии 1.3 можно использовать только диски с именами "A" и "B". В ОС CP/M версии 1.4 можно использовать любое имя от "A" до "D". В ОС CP/M версии 2.0 и более поздних – любое имя от "A" до "P".

При изменении значения ячейки памяти с адресом 005C убедитесь, что требуемый диск установлен в соответствующее устройство, которое подключено к системе. После того как значение ячейки изменено и содержимое файла считано в оперативную память, не забудьте восстановить стандартное значение содержимого ячейки "00" до завершения работы программы "DDT".

ЧТЕНИЕ ФАЙЛА С ДИСКА И ЗАГРУЗКА ФАЙЛА В ПАМЯТЬ

– R
– Restart

Команда R (от английского Read – читать) используется для загрузки и размещения в памяти содержимого файла, указанного в команде I (вставить) так, как описано выше. Команду R никогда не следует использовать, если перед этим не была выполнена команда I.

Эту команду можно вводить в одном из двух форматов. В первом случае

вводится символ "R" и нажимается клавиша "RETURN". Это приводит к тому, что файл загружается в оперативную память, начиная с адреса 0100 (адрес первой ячейки области транзитных программ), если только он не содержит шестнадцатеричный или машинный код программы с другим начальным адресом. Если это так, то файл будет содержать "смещение", которое укажет программе "DDT", по какому адресу (обычно большему, чем шестнадцатеричное 100) загрузить его содержимое. Второй формат этой команды предполагает спецификацию ячейки загрузки посредством идущего за символом "R" шестнадцатеричного числа, являющегося начальным адресом для загружаемого файла. Этот параметр является необязательным и при его использовании следует соблюдать особую осторожность, с тем чтобы не указать адрес вне области транзитных программ.

В последующих двух примерах демонстрируется синтаксис для первого и второго формата команды R. Команда I также включена в эти примеры. Здесь показаны обе команды, потому что ни одну из этих команд нельзя использовать отдельно от другой:

```
A>DDT  
DDT VERS X.Y  
-TEST01.COM  
-R  
-
```

```
A>DDT  
DDT VERS X.X  
-TEST01.COM  
-R0200  
-
```

Если используется второй формат команды R, то указываемый адрес не должен превышать шестнадцатеричного числа FFFF. Иначе начальный адрес уменьшается до 0100 (начальный адрес, обычно принимаемый по умолчанию). Никогда не следует также указывать адреса в интервале от 0000 до 0100, поскольку эта область памяти используется ОС CP/M и программой "DDT" для хранения их собственных переменных. Не следует также использовать смещение, которое в сочетании со стартовым адресом приводит к тому, что файл записывается в область, где хранится программа "DDT" и ОС CP/M. Последствия изменения содержимого этой области памяти очевидны.

РЕАССЕМБЛИРОВАНИЕ И ВЫВОД ЧАСТИ КОДА ПРОГРАММЫ

- L
- Lstart
- Lstart,end

Команда L (от английского List – печатать) используется для реассемблирования хранящихся в памяти шестнадцатеричных кодов и вывода операторов языка ассемблера. Если команда L задана без параметров, то программа "DDT" выводит реассемблированный код очередных одиннадцати команд языка ассемблера. При этом выводятся также начальные адреса для каждой строки. Эта ситуация проиллюстрирована в следующем примере:

```
ADDI TEST01.COM
DDT VERS X.X
NEXT PC
0200 0100
-L
0100 LXI H,0000
0103 DAD SP
0104 SHLD 01FA
0107 LXI SP,023C
010A XRA A
010B STA 01F9
010E MVI C,0F
0110 LXI D,0005C
0113 CALL 0005
0116 LXI D,01B5
0119 INR A
```

После того как одиннадцать операторов языка ассемблера распечатаны, появится промпт программы "DDT", а счетчик команд PC будет увеличен до адреса последней распечатанной команды.

В команде L можно также указать адрес первой команды. Для этого после символа "L" вводят шестнадцатеричное число, представляющее собой требуемый начальный адрес. Результаты использования этого формата команды L представлены в следующем примере:

```
-L0119
0119 INR A
011A JZ 0177
!
!
V
012E LXI H,00080
```

Кроме того, в команде L можно указать конечный адрес, который вводится после начального адреса через запятую. При этом выдаются мнемонические обозначения всех машинных команд, хранящихся в памяти между указанными адресами. В следующем примере показаны некоторые варианты использования команды L в этом формате:

```
-L0119,011D
0119 INR A
011A JZ 0177
011D LXI B,
-
-L0100,0101
0100 LXI H,
```

Изредка при использовании команды L можно обнаружить следующее сообщение об ошибке: "?=?". Обычно это сообщение обозначает, что программа "DDT" не может истолковать шестнадцатеричный код (напечатанный после символов "?=?"), поскольку он не является кодом операции ни для одной машинной команды микропроцессора Intel 8080. Кроме того, сообщение может означать, что указанная ячейка содержит данные или поле операндов какой-либо команды.

Команда S (от английского Set – установить) выполняет две функции. По этой команде программа "DDT" выдает на консоль последовательность адресов ячеек памяти, начиная с указанного адреса, а также изменяет их содержимое. Одновременно с адресом каждой ячейки печатается также и ее текущее значение. После того как команда S задана и выданы адрес и содержимое первой ячейки, курсор или головка печатающего механизма устанавливается справа от текущей ячейки. С этого момента можно вводить новые данные, заменяющие прежние. Чтобы оставить прежние данные без изменений, можно просто нажать клавишу "RETURN". В любом случае после того, как нажата клавиша "RETURN", на следующей строке распечатываются адрес следующей ячейки памяти и ее содержимое. Программа "DDT" продолжает работу в таком режиме до ввода символа ". " (точка). При этом выполнение команды S завершается и снова появляется основной промпт программы "DDT". В целом этот процесс продемонстрирован в следующем примере:

```
A>DDT TEST01.COM
DDT VERS X.X
NEXT FC
0200 0100
-S0105
0105 FA <CR>
0106 01 <CR>
0107 31 <CR>
0108 3C 00
0109 02 01
010A AF <CR>
010B 32 .
```

В предыдущем примере команда S была задана, начиная с адреса 0105. Первая строка, таким образом, отражает содержимое ячейки памяти, расположенной по этому адресу, которое равно FA. Поскольку это значение не изменялось, была нажата клавиша "RETURN", после чего были выданы адрес следующей ячейки и ее содержимое. В этом примере строки, отмеченные символами "< cr >", являются строками, в которых данные не изменились при нажатии клавиши "RETURN". Ячейкам с адресами 0108 и 0109 были присвоены новые значения "00" и "01" соответственно. Если новые данные вводятся таким образом, то они замещают прежнее содержимое ячейки, расположенной по распечатанному адресу. В последней строке был введен символ ". " (точка). Это привело к завершению выполнения команды S и, следовательно, к появлению основного промпта программы "DDT". Команду S можно применять для изменения значений переменных в некоторых областях памяти для модификации порядка выполнения программы. Таким способом можно, например, модифицировать текстовый процессор WordStar, если известно назначение переменных в его внутренних таблицах.

ВСТАВКА КОМАНД АССЕМБЛЕРА ПО УКАЗАННОМУ АДРЕСУ

- Start

Команда A (от английского Assemble – асSEMBЛИровать) используется для вставки инструкций языка ассемблера в программу, начиная с некоторого заданного адреса. С помощью этой команды можно вводить мнемонические обозначения машинных инструкций микропроцессора Intel 8080 и операнды. Однако все числовые операнды должны быть представлены в шестнадцатеричном формате. С помощью этой команды нельзя вводить символьические имена и нельзя также назначать символьическим именам значения. С этой точки зрения к команде A можно обращаться как к мини-ассемблеру.

Для ввода этой команды необходимо, во-первых, указать символ "A", а затем (без пробела) начальный адрес, представленный в шестнадцатеричном формате. После того как команда введена, на консоли выводится начальный адрес, а курсор или головка печатающего механизма устанавливается справа от этого адреса. Теперь вводится очередной оператор языка ассемблера и нажимается клавиша "RETURN". После того как введен текст и нажата клавиша "RETURN", на следующей строке печатается адрес следующей ячейки памяти (не занятой предыдущей машинной инструкцией). Программа "DDT" ожидает ввода данных. Если клавиша "RETURN" нажата, а текст не введен, то выполнение команды A заканчивается и снова появляется основной промпт ОС CP/M. Использование команды A продемонстрировано в следующем примере:

```
A>DDT DDT.COM
DDT VERS X:X
NEXT PC
1400 0100
-L01BB,0194
 018B MVI C,10
 018D CALL 0005
 0190 STA 1E5F
 0193 RET
 0194
-
-A01BB
 018B LHLD 1EC
 018E SHLD 8E7
 0191 INX H
 0192 SHLD 8ED
 0195 LDH 1EB
 0198 .(CR)
```

В этом примере в реассемблированной форме представлен фрагмент программы, расположенный в памяти с адреса 018B по 0194. Когда программа "DDT" используется в качестве мини-ассемблера, обычно перед вводом операторов языка ассемблера задают команду L с тем, чтобы сначала просмотреть содержимое ячеек памяти, которые требуется изменить. В этом примере корректный синтаксис команды A показан после распечатки фрагмента програм-

мы. Указанный в команде начальный адрес такой же, как у кода программы, и после ввода команды адрес первой ячейки, предназначенный для записи операторов языка ассемблера, выведен на консоль. В процессе выполнения команды А каждый раз после нажатия клавиши "RETURN" на консоль выдавался адрес следующей ячейки для записи машинного кода очередного оператора. Отметим, что в последней строке для завершения выполнения команды А использован символ ":" (точка).

Отметим, что все значения, расположенные справа (в поле операндов), являются шестнадцатеричными числами. Если после команды А введено любое значение, которое не является шестнадцатеричным адресом в интервале от 0 до FFFF, то выполнение этой команды завершается и на консоль выдается сообщение об ошибке ("?"). Помимо этого подобное сообщение об ошибке появляется в тех случаях, когда программа "DDT" не может истолковать введенное мнемоническое обозначение символа или значение операнда. При этом необходимо просто дать команду А с тем начальным адресом, по которому обнаружена ошибка, и правильно ввести оператор.

Из примера видно, что при выполнении команды А программа "DDT" выдает адрес следующей ячейки, предназначенной для изменения под строкой, которую введен текст предыдущего оператора. Отметим, однако, что между адресами, напечатанными в двух идущих подряд строках, могут быть пропуски. Это объясняется тем, что операторы языка ассемблера могут занимать различное число ячеек памяти. В результате адрес в следующей строке будет отличаться от предыдущего адреса на +1, +2 или +3 в зависимости от напечатанных мнемонических обозначений машинной инструкции микропроцессора Intel 8080. Для завершения выполнения команды А нажмите клавишу "RETURN", после чего снова появится промпт программы "DDT". Для завершения выполнения этой команды можно также использовать символ ":" (точка), однако достаточно нажать клавишу "RETURN", поскольку это укажет программе "DDT" на то, что дальнейший ввод не требуется.

КОПИРОВАНИЕ СОДЕРЖИМОГО ОБЛАСТИ ПАМЯТИ
ПО УКАЗАННОМУ АДРЕСУ

– Mstart1,end1,start2

Команда M (от английского Move – переслать) используется для копирования указанной порции данных, кодов программы для Intel 8080 и пр. в новую область памяти. Данные, которые копируются, указываются при помощи начального (start1) и конечного (end1) адресов. Эта порция данных называется блоком. Размер блока равен разности этих двух параметров. Адрес, по которому необходимо переслать этот блок, указывается в параметре "start2". Следует иметь в виду, что по этой команде блок данных копируется по указанному адресу точно в таком виде, в каком он хранился в исходной области. Эта область редко используется для "перемещения" машинного кода, поскольку этот процесс не упорядочивает адреса, содержащиеся в перемещаемом блоке. Следовательно, программа, которая обращается сама к себе, не будет выполняться должным образом при перемещении ее кода по

команде M в область с другим начальным адресом. Командой M следует пользоваться с осторожностью, поскольку само по себе копирование данных из одной области в другую, как правило, приводит к нарушению целостности любой программы, на место которой копируются данные. Любые данные в области, куда производится копирование, замещаются копируемыми данными и, следовательно, претерпевают изменения.

Для ввода этой команды необходимо сначала напечатать символ "M", затем (без пробела) начальный адрес, запятую, конечный адрес копируемого блока, еще одну запятую и начальный адрес области, в которую блок копируется. Следующий пример демонстрирует эту процедуру:

```
A>DDT TEST01.COM
DDT VERS X.X
NEXT PC
0200 0100
-D0110,012F
0110 11 5C 00 CD 05 00 11 D5 01 3C CA 77 01 01 00 00 .\.....<.W....
0120 C5 0E 14 11 5C 00 CD 05 00 C1 B7 C2 67 01 21 80 ....\.....6.!
-M0110,011F,0120
-
-D0110,012F
0110 11 5C 00 CD 05 00 11 D5 01 3C CA 77 01 01 00 00 .\.....<.W....
0120 11 5C 00 CD 05 00 11 D5 01 3C CA 77 01 01 00 00 .\.....<.W....
```

Этот пример иллюстрирует использование команды M. Сначала для ясности распечатываются данные, расположенные с адреса 0110 по 012F. Затем вводят команду M, которая указывает программе "DDT" скопировать данные из ячеек с 0110 по 011 (блок длиной в 16 байтов) в эквивалентное количество ячеек, начиная с адреса 0120. Затем распечатывается содержимое той же области памяти. Отметим, что теперь вторая строка содержит те же данные, что и первая. Пользуйтесь данной командой всегда с осторожностью, чтобы не нарушить целостность программы, находящейся в памяти. Будьте особенно аккуратны при указании адреса области размещения копии, чтобы не затронуть содержимого зарезервированных областей памяти.

С помощью команды можно ускорить заполнение произвольного количества ячеек памяти константами.

ИСПОЛНЕНИЕ ЧАСТИ ПРОГРАММЫ И УСТАНОВЛЕНИЕ ТОЧЕК ПРЕРЫВАНИЯ

- G
- Gstart
- Gstart,end
- G,end
- G,end1,end2

Команда G (от английского Go – запустить) используется для инициирования выполнения программы, находящейся под контролем программы "DDT", т. е. для запуска находящейся в памяти программы. Существуют различные форматы данной команды, некоторые из которых редко используются непрограммистами.

Различные форматы команды G позволяют начинать программу с текущего адреса, хранящегося в счетчике команд РС, выполнять программу, начиная с указанного адреса, задавать для программы точку останова и т. д.

Использование команды G при запуске программы для микропроцессора Intel 8080 следует рассмотреть особо. Программа, выполняемая под управлением программы "DDT", должна завершаться специальной инструкцией RST (Рестарт), а не инструкцией RET (Возврат), которая обычно используется для завершения программ. Если инструкция RST не используется, то программа "DDT" вероятно выполнит незаконный возврат (RET) и потеряет контроль над данной программой. В результате программа "зависнет" и только сброс системы вернет ее в исходное состояние. Если команда G используется в обычном формате, то инструкцию RST можно вставлять в конце программы при помощи команды A, в случае использования других форматов команды G инструкция RST автоматически вставляется по указанному адресу или адресам. На профессиональном языке программистов инструкция RST называется "точкой прерывания", поскольку она определяет то место, где программа будет "прервана".

Когда программа "DDT" обнаруживает инструкцию RST (точку прерывания), она печатает адрес ячейки, где она была обнаружена, а перед адресом — "*" (звездочка). Например, сообщение "*7FFF" означает, что инструкция RST обнаружена по адресу 7FFF. Эта информация представляет интерес для программистов, поскольку с ее помощью можно определить, в каком месте выполнение программы завершено.

Ниже перечислены допустимые форматы команды G:

G. Используется для того, чтобы начать выполнение программы с ячейки, на которую указывает счетчик команд РС. Этот формат используется редко, если только пользователь не устанавливает счетчик команд на нужный стартовый адрес. В противном случае программа "DDT" теряет контроль над программой и вернется в программу "DDT" или ОС CP/M без рестарта (полной перезагрузки) системы невозможно.

Gstart. Используется для того, чтобы начать выполнение программы с заданного шестнадцатеричного адреса, который в командной строке обозначен как "start". Выполнение этой командной строки приведет к тем же непредсказуемым результатам, что и при вводе команды G без параметров, если в программу предварительно не вставлена инструкция RST.

Gstart, end. При использовании команды G в этом формате выполнение программы начинается с адреса "start" и завершается в ячейке с адресом "end". Программа выполняется так, как будто точка прерывания установлена в ячейке с адресом "end" и выдает адрес, а перед ним символ "*". Однако реальная инструкция, расположенная по адресу "end", выполниться не будет. Если до ячейки с адресом "end" обнаружена другая инструкция RST, то это приведет к завершению программы. Отметим, что использование этой командной строки может привести к тем же неприятным последствиям, что и при использовании командных строк "G" и "Gstart".

Gstart, end1, end2. Формат команды G такой же, как и предыдущий, за тем исключением, что здесь указываются два конечных адреса "end1" и "end2". При достижении любой точки прерывания выполнение программы останавливается и управление передается программе "DDT". Этот формат используется в тех случаях, когда программа может завершиться в одной из двух точек и программист хочет отличить их.

G, end. Формат команды G такой же, как и "Gstart, end", за тем исключени-

ем, что выполнение программы начинается с ячейки, на которую установлен счетчик команд РС. При достижении точки прерывания с адресом "end" выполнение программы прекращается.

Gend1, end2. Формат команды G такой же, как "G start, end1, end2", за тем исключением, что выполнение программы начинается с ячейки, на которую установлен счетчик команд РС.

Отметим, что выполнение программы "DDT" завершается по команде GO или CTR/C. В обоих случаях они вынуждают начать выполнение программы с адреса 0. Ячейка с адресом 0 содержит код, который вызывает инициализацию системы и возврат в режим команд ОС CP/M.

ИСПОЛНЕНИЕ ОПРЕДЕЛЕННОГО ЧИСЛА КОМАНД И ВЫВОД СОДЕРЖИМОГО РЕГИСТРОВ

– Unumber

Команда U (от английского Untrace – отключить трассировку) используется для выполнения определенного числа машинных инструкций программы, размещенной в памяти, без отображения реального процесса их выполнения (подобное происходит по команде T). По команде U содержимое регистров микропроцессора Intel 8080 выдается на консоль непосредственно перед началом выполнения программы. По команде T содержимое регистров выводится после выполнения каждой машинной инструкции. По команде U выводятся только результаты выполнения последней машинной инструкции. Обе команды U и T имитируют выполнение программы. Хотя команда U выполняется быстрее, чем команда T, обе они выполняются в 500 раз медленнее команды G.

Для запуска команды U необходимо сначала ввести символ "U", а затем (без пробела) шестнадцатеричное число, представляющее собой количество машинных инструкций, которое необходимо выполнить. После выполнения программы (выполнено заданное количество машинных инструкций или достигнута точка прерывания) обработка команды U заканчивается. Кроме того, программа "DDT" присваивает счетчику команд РС значение, содержащееся в паре регистров H и L. Она также устанавливает начальный адрес команды L, соответствующий адресу следующей выполняемой инструкции.

Если в процессе обработки команды U требуется прервать выполнение программы и вернуть управление в программу "DDT", можно нажать любую клавишу на клавиатуре консоли. Этот случай показан в следующем примере:

```
A>DDT TEST01.COM
DDT VERS X.X
NEXT PC
0200 0100
-U5
COZ0M0E010 A=00 B=0000 D=0000 H=0000 S=0100 P=0100 LXI H,0000*01
-U10
COZ1M0E110 A=00 B=0000 D=0000 H=0100 S=023C P=010B STA 01F9*0123
-U20
COZ0M0E010 A=04 B=0014 D=01D5 H=0003 S=023A P=0123 LXI D,005C*01
```

В этом примере команда U применена для выполнения сначала 5 машинных инструкций, затем 10 и, наконец, 20 инструкций. Каждый раз перед выполнением первой машинной инструкции выдается состояние регистров центрального процессора. В приведенном примере каждое поле первой строки имеет следующий смысл:

COZOCOEIO	разбивается на поля следующим образом: "CO" означает, что флагок переноса равен 0. "Z0" означает, что флагок нуля равен 0. "M0" означает, что флагок минуса равен 0. "E0" означает, что флагок четности равен 0. "IO" означает, что флагок внутреннего переноса равен нулю.
A = 00	Сумматор содержит 00.
B = 0000	Регистр B содержит 00, а регистр C содержит 00.
D = 0000	Регистр D содержит 00, а регистр E содержит 00.
H = 0000	Регистр H содержит 00 и регистр L содержит 00.
S = 0100	Указатель стека содержит 0100.
P = 0100	Счетчик команд содержит 0100.
LXI, H, 0000 * 010B	Оператор языка Ассемблер расположен по адресу 0100 (указывается посредством "P = 0100").

Эта информация полезна при анализе отчета, получаемого по командам T и X, которые описываются ниже.

ТРАССИРОВКА ИСПОЛНЕНИЯ ОПРЕДЕЛЕННОГО ЧИСЛА КОМАНД

– Tnumber

Команда T (от английского Trace – трассировать) имеет то же назначение, что и описанная выше команда U. Отличие заключается в том, что состояние регистров центрального процессора и мнемонические обозначения машинных инструкций выдаются на консоль после выполнения каждой инструкции. Эта команда выполняется существенно медленнее команд U и G, поскольку выдается информация о выполнении каждой машинной инструкции.

Для запуска команды необходимо сначала ввести символ "T", а затем (без пробела) шестнадцатеричное число, представляющее собой количество машинных инструкций, которое необходимо выполнить. После того как выполнение команды T закончено, программа "DDT" присваивает счетчику команд PC значение, хранящееся в паре регистров H и L. Помимо этого программа устанавливает для команды L (реассемблировать код) начальный адрес, соответствующий адресу следующей выполняемой инструкции.

Если необходимо прекратить выполнение команды T и вернуть управление в программу "DDT", то для этого можно нажать любую клавишу на клавиатуре консоли. В следующем примере показано, как используется эта команда:

```

A>DDT
DDT VERS X.X
NEXT PC
0200 0100
-TS
COZOMOE0IO A=00 B=0000 D=0000 H=0000 S=0100 P=0100 LXI H,0000
COZOMOE0IO A=00 B=0000 D=0000 H=0000 S=0100 P=0103 DAD SP
COZOMOE0IO A=00 B=0000 D=0000 H=0100 S=0100 P=0104 SHLD 01FA
COZOMOE0IO A=00 B=0000 D=0000 H=0100 S=0100 P=0107 LXI SP,023C
COZOMOE0IO A=00 B=0000 D=0000 H=0100 S=023C P=010A XRA A*010B
-T10
COZIMOE110 A=00 B=0000 D=0000 H=0100 S=023C P=010B STA 01F9
COZIMOE110 A=00 B=0000 D=0000 H=0100 S=023C P=010E MWI C,0F
COZIMOE110 A=00 B=000F D=0000 H=0100 S=023C P=0110 LXI D,005C
COZIMOE110 A=00 B=000F D=005C H=0100 S=023C P=0113 CALL 0005
COZIMOE110 A=00 B=000F D=005C H=0100 S=023A P=0005 JMP AB00
COZIMOE110 A=00 B=000F D=005C H=0100 S=023A P=A800 JMP AEA2
COZIMOE110 A=00 B=000F D=005C H=0100 S=023A P=AEA2 XTHL
COZIMOE110 A=00 B=000F D=005C H=0116 S=023A P=AEA3 SHLD B74A
COZIMOE110 A=00 B=000F D=005C H=0116 S=023A P=AEA6 XTHL
COZIMOE110 A=00 B=000F D= <ПРОБЕЛ>

```

В второй части этого примера показано, что выполнение команды T прервано после того, как на клавиатуре нажата клавиша "пробел". Можно прекратить вывод данных, нажав любую клавишу (за исключением CTRL/C, так как это приводит к завершению выполнения программы "DDT" и возврату в ОС CP/M).

ВЫВОД И ИЗМЕНЕНИЕ СОДЕРЖИМОГО ОДНОГО ИЛИ НЕСКОЛЬКИХ РЕГИСТРОВ

- X
- Xregister

Команда X используется для вывода текущего значения всех регистров микропроцессора Intel 8080 или же содержимого указанного регистра. Она также используется для внесения требуемых изменений в указанный регистр. Информация, которая выдается при использовании команды X, аналогична информации, выдаваемой при использовании команд U и T. Разница здесь состоит в том, что информация отражает текущее состояние регистров центрального процессора независимо от состояния процесса выполнения программы.

Если команда X введена без параметров, то имена всех регистров и их текущее содержимое распечатываются в одной строке. После этого снова появится промпт программы "DDT". С другой стороны, после команды X можно ввести имя специфицированного регистра с целью вывода его содержимого на консоль. Если используется этот формат команды X, то курсор или головка печатающего механизма устанавливается справа от имени регистра и его содержимого, куда при желании можно ввести новые дополнительные данные. Для того чтобы оставить содержимое регистра без изменений, необходимо нажать клавишу "RETURN", после чего снова появится промпт программы "DDT".

Команду X можно использовать для вывода на консоль и изменения содержимого любого из регистров центрального процессора, имена которых были распечатаны в одной строке (см. следующий пример):

```
A>DDT TEST01.COM
DDT VERS X.X
NEXT PC
0200 0100
-X
COZOMOE010 A=00 B=0000 D=0000 H=0000 S=0100 P=0100 LXI H,0000
-
XC
CO <CR>
-XZ
ZO <CR>
-XM
MO <CR>
-XE
EO <CR>
-XI
IO <CR>
-XA
A=00 <CR>
-XB
B=0000 <CR>
-XD
D=0000 <CR>
-
```

В примере показаны различные варианты использования команды X. В первом случае команда X введена без параметров. При этом были распечатаны имена и текущее содержимое всех регистров центрального процессора. Далее в примере показано использование команды X с именами отдельных флагков и регистров. Символы "<ст>" применяются как указатели того, что клавиша "RETURN" нажималась после каждого вывода данных. Однако в то место, где стоит этот символ, можно при желании ввести новые значения для регистров или флагков. Это показано в следующем примере:

```
A>DDT TEST01.COM
DDT VERS X.X
NEXT PC
0200 0100
-XZ
ZO 1
-XZ
Z1 0
-XZ
ZO
-
-XB
B=0000 FFFF
-XB
B=FFFF 0000
-XB
B=0000
-
-^C
```

В этом примере сначала показано, как состояние флагка Z (флагок нуля) изменено с 0 на 1, а затем опять на 0. Затем, как содержимое регистров, В и С изменено с 0000 на FFFF, а потом, обратно, на 0000.

ИСПОЛЬЗОВАНИЕ ВСТРОЕННОГО КАЛЬКУЛЯТОРА ДЛЯ СЛОЖЕНИЯ И ВЫЧИТАНИЯ ШЕСТНАДЦАТИЧНЫХ ЧИСЕЛ

– Hxxxx, xxxx

Малоизвестную команду Н (от английского Нех – шестнадцатеричный) используют для выполнения операций сложения и вычитания шестнадцатеричных чисел. Эта команда выполняет функцию калькулятора. Она не оказывает воздействие на содержимое оперативной памяти. Команда Н полезна в тех случаях, когда в процессе выполнения программы "DDT" необходимо сложить или вычесть какие-либо два шестнадцатеричных числа. Часто команда Н используется для вычисления адресных смещений при замещении кода в секции драйверов ввода-вывода (BIOS) ОС CP/M (см. [6]).

Для запуска команды необходимо ввести сначала символ "Н", а затем (без пробела) первое шестнадцатеричное число, запятую и второе шестнадцатеричное число. После нажатия клавиши "RETURN" выводятся два числа. Первое число представляет собой сумму двух чисел, введенных вместе с командой Х (xxxx + xxxx). Второе число представляет собой разность между этими двумя числами (xxxx – xxxx). Вместе с этой командой можно вводить только шестнадцатеричные числа (цифры 0 . . . 9 и символы A . . . F, что эквивалентно десятичным числам от 0 до 15). Числа могут быть длиной от одной до максимум четырех цифр. Команда Н производит операцию сложения правильно независимо от порядка, в котором введены числа. Однако для того, чтобы получить в результате вычитания положительное число, необходимо первым вводить число, большее или равное второму числу. Если первое число меньше второго, то вычисление результата вычитания начнется с числа FFFF:

```
A>DDT  
DDT VERS X.X  
  
-H1,1          (1 + 1 = 2)(1 - 1 = 0)  
0002 0000  
-H2,1  
0003 0001  
-H5,2  
0007 0003  
-HA,5  
000F 0005  
-H100,50  
0150 00B0  
-HABCD,9ABC  
4689 1111  
  
-H1,2          (1 + 2 = 3)(1 - 2 = FFFF)  
0003 FFFF  
-H200,300  
0500 FF00  
-H5000,AAAA  
FAAA A556  
-  
-`C
```

В этом примере сначала показан порядок вычислений, когда числа с командой введены в правильной последовательности. Затем даны результаты операции вычитания, полученные в случае, когда второе число, введенное с командой, больше первого.

На этом описание команды DDT закончено. В следующей главе описывается встроенная команда SAVE.

ГЛАВА 16



Версии ОС *CP/M*:

1.3 1.4 2.0 2.2

Тип команды:

встроенная

ОБЩИЕ СВЕДЕНИЯ

Команда SAVE используется в тех особых случаях, когда требуется сохранить содержимое оперативной памяти в указанном файле на диске. Обычно потребность в команде SAVE возникает только у программистов, наиболее часто она используется при программировании на языке ассемблера для хранения на дисковом носителе копии содержимого оперативной памяти. В команде SAVE также возникает необходимость при использовании команды MOVCPCM для внесения изменений в секцию драйверов ввода-вывода (BIOS) ОС *CP/M*. Команда SAVE используется к тому же для создания файлов, размещаемых во вновь определенных областях пользователя (см. выше описание команды USER). У тех пользователей, которые не программируют на языке ассемблера и не используют на одном диске различные области пользователя, потребность в этой команде, вероятно, не возникнет.

ОБРАЩЕНИЕ К КОМАНДЕ

Формат командной строки для команды SAVE очень простой, хотя для формирования командной строки необходимо располагать некоторой дополнительной информацией. Чтобы сохранить содержимое оперативной памяти в файле, следует указать количество сохраняемых "страниц оперативной памяти" и имя создаваемого файла. Чтобы сохранить файл, команду SAVE следует вводить, пока данные, которые нужно сохранить, еще находятся в оперативной памяти. До тех пор, пока команда SAVE не выполнена, нельзя запускать на выполнение никакую другую программу. Если перед командой SAVE запустить на выполнение какую-либо другую программу, то файл, который предполагалось сохранить на диске, будет потерян при загрузке в оперативную память новой программы.

Команда SAVE обычно используется в сочетании с трактирной программой "DDT" (см. описание команды DDT). Программу "DDT" можно использовать для определения количества "страниц оперативной памяти", которое должно быть указано вместе с командой SAVE. Кроме того, программу

"DDT" можно использовать для загрузки в оперативную память всех необходимых данных. Программа "DDT" предоставляет пользователю возможность определить шестнадцатеричные (по основанию 16) номера первой и последней страниц оперативной памяти после того, как ей задано имя файла, размеры которого необходимо вычислить (одна страница оперативной памяти представляет собой блок данных длиной 256 или 100 шестнадцатеричных байтов). Затем шестнадцатеричный номер последней страницы оперативной памяти необходимо преобразовать в десятеричное число, которое и должно быть указано с командой SAVE.

После того как количество страниц оперативной памяти определено, можно использовать команду SAVE с целью сохранения указанного файла. В следующем примере иллюстрируется использование этой команды для записи файла на диск А. Количество страниц оперативной памяти равно 12, а файл, в котором необходимо сохранить данные, должен иметь имя "SPECIAL.COM". Поскольку текущим является диск А и в командной строке не указано имя никакого другого диска, то данный файл будет записан на диск А:

```
A>SAVE 12 SPECIAL.COM
A>
A>STAT SPECIAL.COM
RECS    BYTES EXT ACC
 32        4K   1 R/W A:SPECIAL.COM
BYTES REMAINING ON A:  476K
```

A>

Отметим, что в примере после того, как файл сохранен, использована команда STAT для проверки его размеров. Хотя в командной строке указано 12 страниц оперативной памяти, что равняется 3,072 байтам (или 3К), в отчете команды STAT указано, что файл занимает 4К. Это объясняется тем, что в ОС CP/M дисковая память под файлы выделяется блоками по 2К. Поэтому хотя данные в файле занимают только 3К, реальный объем использованного на диске пространства составляет 4К.

Файл можно записать на любой диск, а не только на текущий. Для этого в командной строке необходимо после числа страниц оперативной памяти и перед именем файла указать имя диска. Прежде чем использовать команду SAVE, убедитесь в том, что на диске имеется достаточный объем свободного пространства. Если на диске не окажется достаточного пространства для того, чтобы разместить указанный файл, выдается следующее сообщение об ошибке:

```
A>SAVE 12 SPECIAL.COM
NO SPACE
A>
```

При возникновении указанной ситуации следует вычислить, сколько примерно килобайт дисковой памяти потребуется для записи файла. Эта операция является относительно простой, поскольку одна страница оперативной памяти занимает 256 байт, а в одном килобайте содержится 1,024 байт:

1,024 деленное на 256 равно 4.

Таким образом, можно сказать, что страница оперативной памяти занимает $\frac{1}{4}$ килобайта или что один килобайт содержит четыре страницы оперативной памяти. В этом примере, разделив 12 (количество страниц оперативной памяти) на 4, можно убедиться, что файл занимает 3К или 3 килобайта. Зная это, прежде чем использовать команду SAVE, можно выяснить, достаточно ли свободного пространства на диске для записи данного файла.

ПРИМЕЧАНИЕ

Если в результате указанных выше вычислений получено нечетное число (например, 3), то его следует округлить до ближайшего четного числа (в данном случае до 4). Это объясняется тем, что в ОС *CP/M* дисковая память под файлы выделяется блоками по 2К.

При спецификации результирующего файла следует указать конкретное имя, поскольку в команде SAVE использование метасимволов не допускается. Если в имени файла обнаружен любой метасимвол или же если не указано количество страниц оперативной памяти, то выдается следующее сообщение об ошибке:

```
A>SAVE 12 SPECIAL.*  
FILENAME?  
A>  
A>SAVE SPECIAL.COM  
FILENAME?  
A>
```

При получении этого сообщения необходимо заново правильно ввести команду SAVE. Данные, предназначенные для записи в файл, по-прежнему хранятся в оперативной памяти.

ГЛАВА 17

Версии ОС *CP/M*:

1.3 1.4 2.0 2.2

Тип команды:

транзитная

DUMP

ОБЩИЕ СВЕДЕНИЯ

Команда DUMP является транзитной командой ОС *CP/M* и обеспечивает обращение к транзитной программе, которая также называется "DUMP".

Программа "DUMP" предоставляет пользователю ОС *CP/M* возможность вывести в шестнадцатеричном формате содержимое файла. Это действие известно под названием "дампирование" файла; отметим, что обычно содержимое такого файла можно просмотреть только с помощью программы "DDT" (см. описание команды DDT). С помощью программы "DUMP" можно просмотреть файлы, содержащие как машинные коды, так и транзитные программы, т. е. файлы с расширением ".COM". Фактически выполнение команды DUMP эквивалентно выполнению команды D программы "DDT".

ОБРАЩЕНИЕ К ПРОГРАММЕ "DUMP"

Для обращения к программе "DUMP" необходимо, чтобы файл "DUMP.COM" хранился на диске, вставленном в одно из устройств системы. Существует только один формат команды DUMP, при котором в качестве ответа на промпт с именем текущего диска вводится строка символов "DUMP" и спецификация файла. Как только команда введена, программа "DUMP" выводит в шестнадцатеричном формате полное содержимое указанного файла. Формат распечатки такой же, как у программы "DDT", но без представления данных в коде ASCII.

Программа "DUMP" допускает использование метасимволов (символы "*", "?"") для примерного определения имени файла. Если в спецификации файла применены метасимволы, то программа "DUMP" выдает содержимое только первого файла согласно указанной спецификации.

Команда DUMP выполняется так же, как истроенная команда TYPE, в том смысле, что содержимое файла выдается непрерывно до тех пор, пока не обнаружен признак конца файла. Для прекращения и возобновления перемещения распечатки на экране можно использовать команду CTRL/S, при нажатии любой другой клавиши выполнение программы "DUMP" заканчивается.

В целом программа "DUMP" выдает шестнадцатеричный код в том же формате, что и программа "DDT". Разница заключается в том, что в данном случае представление данных в коде ASCII не печатается.

В следующем примере продемонстрировано использование команды DUMP и показан формат, в котором выводится и шестнадцатеричный код:

A>DUMP TEST01.COM

```
0000 21 00 00 39 22 FA 01 31 3C 02 AF 32 F9 01 0E 0F
0010 11 5C 00 CD 05 00 11 B5 01 3C CA 77 01 01 00 00
0020 C5 0E 14 11 5C 00 CD 05 00 C1 E7 C2 67 01 21 80
0030 00 7E E6 7F FE 1A CA 67 01 FE 20 CA 59 01 FE 0D
0040 CA 59 01 FE 0A CA 59 01 3A F9 01 FE 00 CA 5E 01
0050 03 3E 00 32 F9 01 C3 5E 01 3E 01 32 F9 01 23 7C
0060 B7 CA 31 01 C3 20 01 11 5C 00 C5 0E 10 CD 05 00
0070 C1 CD B1 01 11 E5 01 0E 09 CD 05 00 2A FA 01 F9
0080 C9 C5 E1 06 00 11 F0 DB CD A5 01 11 18 FC CD A5
0090 01 11 9C FF CD A5 01 11 F6 FF CD A5 01 7D C6 30
00A0 4F CD C8 01 C9 0E 2F 0C 19 DA A7 01 7A 2F 57 7B
00B0 2F 5F 13 19 79 FE 31 B2 C2 01 78 B7 79 C8 CD C8
00C0 01 C9 06 FF CD C8 01 C9 D5 C5 E5 59 0E 02 CD 05
00D0 00 E1 C1 B1 C9 6E 6F 20 73 6F 75 72 63 65 20 66
00E0 69 6C 65 2E 24 20 77 6F 72 64 73 20 69 6E 20 74
00F0 68 65 20 66 69 6C 65 2E 24 00 00 00 00 00 00 00
```

В этом примере показано шестнадцатеричное представление содержимого указанного файла. Четырехзначные числа в крайнем левом столбце представляют собой начальные адреса для каждой строки с кодом. В каждой строке распечатывается по шестнадцать двузначных чисел. Каждое двузначное число соответствует ячейке с определенным адресом и представляет собой то значение, которое хранится в этой ячейке памяти. Например, число, расположеннное сразу после начального адреса, соответствует ячейке с первым (начальным) адресом. Второе число соответствует следующей ячейке и т. д. В этом примере первая строка имеет начальный адрес 0000. Расположенные в этой строке шестнадцать чисел соответствуют ячейкам памяти с шестнадцатеричными адресами от 0000 до 000F. Во второй строке представлено содержимое ячеек памяти с адресами от 0010 до 001F. Все остальные строки организованы точно так же.

ГЛАВА 18

Версии ОС *CP/M*:

1.3 1.4 2.0 2.2

По указанию поставщика

Тип команды:

дополнительная транзитная

ОБЩИЕ СВЕДЕНИЯ

Практически все версии ОС *CP/M* поставляются с программами форматирования дисков. Эта программа необходима потому, что она позволяет пользователю ОС *CP/M* инициализировать новые гибкие или жесткие диски (сменные или постоянные) для записи на него информации. Она необходима и потому, что нет никакой гарантии в том, что новый гибкий диск будет работать правильно с вашей конфигурацией ОС *CP/M* или дисковых устройств, или же жесткий диск будет правильно сформатирован при первом включении системы (возможно даже, что каждый гибкий диск содержит информацию о том, как он был предварительно отформатирован на предприятии-изготовителе).

Форматирующая программа организует на диске дорожки и сектора таким образом, чтобы ОС *CP/M* могла правильно к ним обращаться. Обычно программа "FORMAT", включенная в конкретную версию ОС *CP/M*, настроена на определенное оборудование. Разработчики дисковых устройств и контроллеров при проектировании своего оборудования не учитывают его совместимость с другими разработчиками. Это означает, что форматирующая программа, имеющаяся в одной версии ОС *CP/M*, возможно не будет работать на ЭВМ, оснащенной другим оборудованием. В этой главе описано, как пользоваться типичной программой форматирования дисков. Она возможно незначительно или даже сильно отличается от программы, с которой придется иметь дело, однако имеющиеся приведенные примеры дают представление о том, как должна работать любая форматирующая программа.

Довольно быстро обнаруживается, что программа форматирования дис-

ков является первой программой, используемой при получении и установке ОС *CP/M*. Используйте ее для форматирования нового гибкого диска с тем, чтобы затем на этом диске создать резервную копию дистрибутивного* диска ОС *CP/M*. Дистрибутивный диск практически всегда является 8-дюймовым гибким диском с единичной плотностью записи. Как правило, такой диск является программно-секторированным. Это означает, что на диске имеется только одно индексное окошко и что размер секторов определяется ЭВМ программным способом. Дисковый формат для дистрибутивных дисков ОС *CP/M* базируется на стандарте IBM 3740. Помимо этого ОС *CP/M* может поставляться на 5 ¼-дюймовых дисках. Поскольку не существует установленного стандарта для дисковых носителей диаметром 5 ¼ дюйма, формат такого дистрибутивного диска выбирается по усмотрению разработчика ЭВМ для данной версии ОС *CP/M*.

Некоторые версии ОС *CP/M* поставляются на аппаратно-секторированных дисках. В аппаратно-секторированном диске имеется 32 отдельных "индексных" окошко, равномерно расставленных по самой внутренней дорожке. Электронное устройство в контроллере диска обнаруживает эти окошки и использует их для определения начальных позиций секторов самого диска (а не магнитные указатели секторов, которые используются в системах с программно-секторированными дисками). Однако для разметки аппаратно-секторированного диска используются только окошки индикации. Для того чтобы использовать диск для хранения файлов ОС *CP/M*, необходимо еще поместить на него дополнительную информацию о секторах и дорожках. Если файл с данными или программой, который должен "храниться" на дисковом носителе, представить в виде поезда, то можно сказать, что при форматировании диска для него прокладываются рельсы, по которым он следует в депо.

В системах, где используются аппаратно-секторированные гибкие диски, программа форматирования не нужна, поскольку формат такого типа дисков нельзя изменять. Тем, кто использует аппаратно-секторированные диски, нужно просто точно скопировать дистрибутивный диск и пропустить оставшуюся часть главы. Тем, кто использует программно-секторированные диски, потребуется еще один такой же диск, чтобы создать резервную копию дистрибутивного диска. Этот диск необходимо форматировать с единичной плотностью записи (в отличие от двойной плотности). Впоследствии, если система это поддерживает, можно форматировать другие диски с двойной плотностью записи.

ОБРАЩЕНИЕ К ПРОГРАММЕ ФОРМАТИРОВАНИЯ ДИСКА

Для форматирования нового диска необходимо, чтобы файл с программой форматирования (называемую возможно "FORMAT.COM") хранился на диске, вставленном в одно из устройств системы, например на диске "A", если это гибкий диск. Если программа форматирования хранится на жестком диске, то необходимо использовать то, что называется логическим име-

* Дистрибутивный диск содержит полную версию ОС *CP/M*, он является источником информации при генерации заданной конфигурации ОС *CP/M*. – Прим. перев.

нем дискового устройства. Для некоторых версий этих программ необходимо наличие как минимум двух дисковых устройств: одно содержит диск с программой форматирования, а другое — тот диск, который необходимо форматировать. Однако большинство форматирующих программ позволяют форматировать диски при использовании только одного дискового устройства (форматирование диска с использованием одного устройства является, безусловно, весьма утомительной процедурой).

В конкретной системе файл с программой форматирования может и не называться "FORMAT.COM", а может иметь одно из следующих имен или им подобных:

DSKFMT.COM
DSKINIT.COM
FORMAT2.COM
IFORMAT.COM
IN.COM
INIT.COM
INITDSK.COM
MFORMAT.COM

. . . и т. д.

Видно, что потенциального пользователя ОС *CP/M* подобная ситуация может привести в замешательство. При первоначальной установке ОС *CP/M* можно попытаться найти одно из этих имен. Если найти имя все же не удалось, то можно просто попытаться запускать различные программы, которые должны выполнять операцию форматирования. Точно идентифицировать программу форматирования можно, выполнив последовательность операций, аналогичную описанной ниже.

В этом разделе предполагается, что программа форматирования хранится в файле с именем "FORMAT.COM". Этот файл можно назвать и по-другому, пока он имеет расширение ".COM". При этом к нему можно обращаться как к файлу, содержащему исполняемую транзитную программу.

Плотность записи. Гибкие диски можно форматировать с различной плотностью записи. Плотность записи определяет количество дорожек и секторов, которые программа форматирования размечает на диске. Например, при единичной плотности записи на диске размечается 40 дорожек, 32 сектора емкостью в 128 байт на сектор. При двойной плотности записи за счет использования 77 дорожек объем диска увеличивается более чем вдвое. И, если при единичной плотности записи количество байтов в каждом секторе фиксировано и определяется в соответствии со стандартом фирмы IBM, то при двойной плотности записи количество секторов и количество байтов на сектор изменяются в зависимости от того, кем разработана программа форматирования. Стандарта на двойную плотность записи не существует, и поэтому все разработчики программ делают это по-разному. Однако, как правило, формат с двойной плотностью записи имеет емкость либо 256, либо 512, либо 1024 байт на сектор. При большем количестве байт на сектор память используется эффективнее, поскольку при этом требуются меньшие накладные расходы в меньшем числе секторов на диск. Хотя общая емкость диска колеблется около $\frac{1}{2}$ мегабайта, при уменьшении числа секторов она несколько возрастает.

С точки зрения совместимости неважно, какой из этих форматов выбран,

все равно этот диск нельзя прочитать при каком-либо другом стандарте на двойную плотность, не имея в своем распоряжении точно такую же ЭВМ с такой же версией ОС *CP/M*.

Примеры, представленные в этом описании, относятся к типичной форматирующей программе, для которой используются дисковые устройства с одно- или двусторонними программно-секторированными гибкими дисками с различной плотностью записи. Под "различной плотностью записи" подразумевается, что диски этого типа обычно форматируются как с единичной, так и с двойной плотностью, а в некоторых редких случаях и с четырехкратной плотностью. В этом примере использованы форматы с единичной и двойной плотностью записи.

ФОРМАТИРОВАНИЕ ГИБКОГО ПРОГРАММНО-СЕКТОРИРОВАННОГО ДИСКА

FORMAT

Для форматирования гибкого диска необходимо диск с программой форматирования вставить в одно из устройств системы, например в устройство "A". Если на вычислительной установке имеется только одно дисковое устройство, то следует сначала запустить программу форматирования, а затем, если указано снять диск, сделать это и заменить его гибким диском, который надо форматировать. Если на вычислительной установке имеется более чем одно дисковое устройство, то гибкий диск, который требуется сформатировать, можно вставить в другое устройство, например в устройство "B". Затем необходимо выполнить реинициализацию системы (команда *CTRL/C*). Помимо этого убедитесь, что текущим является диск "A" (диск, вставленный в устройство "A"), на что указывает промпт с именем текущего диска (*A >*). Введите команду "форматировать диск" (в представленном ниже примере этой командой является команда "FORMAT") и затем нажмите клавишу *RETURN*:

```
A>FORMAT

XYZ COMPANY FORMAT VERSION X.X
THIS COMMAND FORMATS A DISK, ERASING ALL DATA ON IT.
IS THIS WHAT YOU WANT? [Y/N]: Y
REMOVE SYSTEM DISK AND PRESS THE RETURN KEY: <CR>
ENTER DRIVE NAME CONTAINING THE DISK TO BE FORMATTED: B
DISK TO BE FORMATTED IN DRIVE "B". IS THIS OK? [Y/N]: Y

! 1 = 128 BYTES PER SECTOR, SINGLE DENSITY FORMAT !
! 2 = 256 BYTES PER SECTOR, DOUBLE DENSITY FORMAT !
! 3 = 1024 BYTES PER SECTOR, DOUBLE DENSITY FORMAT !

SELECT ONE OF THE ABOVE FORMAT [1,2,3]: 1
"128 BYTES PER SECTOR, SINGLE DENSITY". IS IT OK? [Y/N]: Y

FORMATTING IN PROCESS. PLEASE WAIT UNTIL FINISHED.
```

FORMATING COMPLETE. TO FORMAT ANOTHER DISK IN THE SAME
DRIVE, TYPE IN "Y", OTHERWISE PLACE SYSTEM DISK BACK IN
DRIVE A AND TYPE RETURN: <CR>

A>

В этом примере* показана программа, сообщения (промпты) которой понять очень легко. Ответы пользователя на эти промпты видны здесь также четко. Промпт "REMOVE SYSTEM DISK, AND TYPE RETURN" (Снимите системный диск и нажмите клавишу RETURN) обозначает, что эту программу можно использовать в системе с одним накопителем на гибком диске. Если в системе имеется только одно дисковое устройство, то после того, как выведен этот промпт, необходимо вынуть из устройства системный диск и вставить в него диск, который должен быть сформатирован. Это следует также сделать и в том случае, если впоследствии в качестве устройства, содержащего форматируемый диск, указывается устройство "A" (даже если к системе подключено несколько дисковых устройств). Если в вычислительной системе имеется более чем одно дисковое устройство и его имя отлично от имени диска с программой "FORMAT", то на этот промпт можно не

* Чтобы облегчить читателю понимание этого примера, приводим построчный перевод сообщений, выдаваемых программой "FORMAT":

XYZ COMPANY FORMAT VERSION X.X.

Программа "FORMAT" фирмы XYZ, версия X.X.

THIS COMMAND FORMATS A DISK, ERASING ALL DATA ON IT.

Эта программа форматирует диск, удаляя все хранящиеся на нем данные.

IS THIS WHAT YOU WANT?

Это то, что Вам надо?

REMOVE SYSTEM DISK AND PRESS THE RETURN KEY:

Снимите системный диск и нажмите клавишу RETURN:

ENTER DRIVE NAME, CONTAINING THE DISK TO BE FORMATTED:

Введите имя устройства, содержащего диск, который должен быть сформатирован:

DISK TO BE FORMATTED IN DRIVE "B". IS THIS OK?

Диск, который должен быть сформатирован, вставлен в устройство "B". Это правильно?

1 = 128 BYTES PER SECTOR, SINGLE DENSITY FORMAT

1 = 128 байт на сектор, формат с единичной плотностью записи

2 = 256 BYTES PER SECTOR, DOUBLE DENSITY FORMAT

2 = 256 байт на сектор, формат с двойной плотностью записи

3 = 1024 BYTES PER SECTOR, DOUBLE DENSITY FORMAT

3 = 1024 байт на сектор, формат с двойной плотностью записи

SELECT ONE OF THE ABOVE FORMATS [1, 2, 3] : 1

Выберите один из перечисленных выше форматов 1, 2, 3, :

"128 BYTES PER SECTOR, SINGLE DENSITY". IS THIS OK?

"128 байт на сектор, единичная плотность записи". Это правильно?

FORMATING IN PROCESS. PLEASE WAIT UNTIL FINISHED.

Идет процесс форматирования. Ждите завершения.

FORMATING COMPLETE. TO FORMAT ANOTHER DISK IN THE SAME DRIVE,
TYPE IN "Y", OTHERWISE PLACE SYSTEM DISK BACK IN DRIVE A AND TYPE
RETURN:

Форматирование закончено. Для того чтобы начать форматирование другого диска на этом же дисковом устройстве, введите символ "Y" или вставьте системный диск в устройство "A" и нажмите клавишу RETURN.: – Прим. перев.

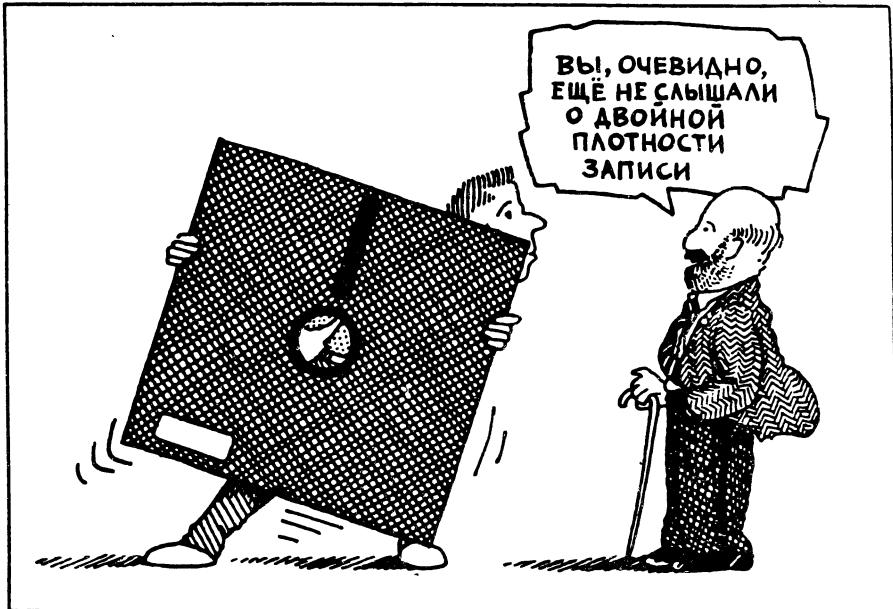
обращать внимания, поскольку основной диск снимать не надо. Некоторые пользователи предпочитают всегда вынимать диск из устройства "A" независимо от того, в какое устройство предполагается вставить форматируемый диск, поскольку если программе "FORMAT" по ошибке указано форматировать дистрибутивный диск, то она сотрет всю имеющуюся на нем информацию.

После загрузки программы форматирования в оперативную память ЭВМ и вплоть до ее завершения ОС *CP/M* не обращается к системному диску. Однако на основании ответов на предыдущие промпты программа "FORMAT" сразу начнет обращаться к тому диску, который должен быть сформирован. По мере того, как программа последовательно проставляет на поверхности носителя метки начала секторов, раздаются специфические щелкающие звуки.

Появление сообщения "FORMATING COMPLETE" (Форматирование закончено) означает, что поверхность диска полностью сформирована. Теперь можно либо вставить в устройство другой чистый диск, который необходимо отформатировать и затем ввести символ "Y" (чтобы повторить процесс форматирования), либо снова вставить системный диск и нажать клавишу "RETURN" для завершения выполнения программы и возврата в основной режим команд ОС *CP/M*. Эти заключительные действия являются важными, поскольку программу форматирования можно корректно завершить только при условии, что в системное устройство вставлен диск, содержащий файл с этой программой (в данном случае файл "FORMAT.COM").

В процессе форматирования может оказаться, что программа не может должным образом сформировать диск. Это может произойти из-за типа диска (например, аппаратно-секторированный вместо программно-секторированного, не та плотность записи и т. д.) или вследствие наличия на диске физического дефекта и т. п. При этом программа форматирования просто "зависнет", когда получит подтверждение начать форматирование диска, и продолжит работу только после того, как будут открыты и закрыты дверцы устройства. В другом случае программа "FORMAT" выдает на консоль свое собственное сообщение об ошибке, которое означает, что дверцы дискового устройства открыты. Однако существуют некоторые сообщения ОС *CP/M*, которые выдаются в случае подобного сбоя.

Если, например, пользователь инициирует команду форматирования и укажет для форматирования диск, вставленный в устройство "B", но не поместит диск в это устройство, то, вероятно, будет выдано следующее сообщение ОС *CP/M*: "DRIVE NOT READY" (Устройство не готово), хотя некоторые программы форматирования вообще не выдают никаких сообщений. Какое сообщение реально будет выдано, зависит от программы форматирования. В случае, аналогичном этому, программа форматирования, скорее всего, завершит свое выполнение. Если это произошло, то следует перезагрузить систему и начать все снова. При получении других сообщений об ошибках следует обратиться к приложению H, в котором детально описываются все сообщения об ошибках ОС *CP/M*.



Для того чтобы проверить, что диск сформатирован правильно, можно воспользоваться командой DIR и вывести на консоль оглавление этого диска, поскольку на вновь сформатированном диске еще нет никаких данных, то, если он сформатирован правильно, появится сообщение "NO FILE". Если получено сообщение типа

"BDOS ERR ON B: BAD SECTOR"

(Ошибка BDOS: плохой сектор на диске B)

или что-нибудь в этом роде, то оно скорее всего указывает на то, что процесс форматирования не удался. Сообщение означает, что в процессе чтения диска обнаружена ошибка. Она может возникнуть или из-за неправильно выполненной процедуры форматирования или вследствие того, что на диске имеется физический дефект. При этом следует повторить процедуру форматирования, с тем чтобы посмотреть, не разрешится ли проблема сама по себе. В некоторых версиях ОС CP/M пользователю предоставляется возможность нажимать различные клавиши для определения различных реакций на ошибки базовой дисковой операционной системы (BDOS). В некоторых случаях можно повторить попытку прочитать данные, вернуть управление ОС CP/M или проигнорировать ошибку. Если ошибка повторилась, то диск, вероятно, испорчен, и им не следует пользоваться (если же такая ошибка возникает при работе со всеми дисками, то возможно, но не обязательно, что дело вовсе не в дисках). Если форматирование диска прошло успешно, то для определения объема имеющегося на диске свободного пространства можно воспользоваться транзитной командой STAT (описанной выше).

Для того чтобы скопировать системный образ ОС CP/M в первые две дорожки диска и тем самым обеспечить возможность загружать ОС CP/M с

этого диска, следует воспользоваться командой SYSGEN (описанной в следующей главе). Для копирования обычных файлов можно применить транзитную команду PIP. Однако если в вычислительной системе имеется только одно дисковое устройство, то использовать команду PIP для копирования файлов с одного гибкого диска на другой нельзя. Обычно в системах с одним дисковым устройством для этих целей имеется специальная программа копирования. Эта программа обычно хранится в файле с именем "COPY.COM", но поскольку она не является стандартной, принцип ее действия для различных моделей ЭВМ и версий ОС CP/M может быть различным.

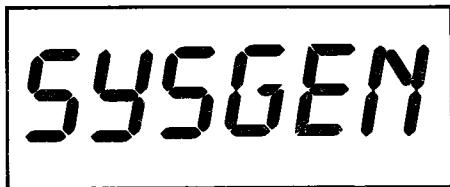
Другие программы форматирования. Как отмечалось ранее, не все программы форматирования дисков работают одинаково. Ниже приводится пример другой программы. Хотя эта программа вызывается точно так же, как в предыдущем примере, ее действие – иное:

```
A>FORMAT  
COMPUPRO DISK SUB-SYSTEM FORMAT VERSION 2.2B  
  
SPECIFY DRIVE (A: - D:) :B DISK IS FORMATED AS 1024 BYTE  
SECTORS.  
SELECT DISK FORMAT MODE (0,1,2,3):  
0 = 128      2 = 512  
1 = 256      3 = 1024    >3  
CONFIRM READY FOR FORMAT ON DISK DRIVE B (Y).Y  
1   2   3   4   5   6   7  
0123456789012345678901234567890123456789012345678901  
F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F  
U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U  
FORMAT COMPLETE.  
DO YOU WANT TO FORMAT ANOTHER DISK ?  
R - REQUEST NEW PARAMETERS.  
CR - DUPLICATE LAST FORMAT.  
N - EXIT BACK TO SYSTEM.          >N  
  
A>STAT B:  
BYTES REMAINING ON B: 1192K
```

В этом примере программа запрашивает имя устройства, содержащего диск, который требуется сформатировать. Здесь введено имя "B". Программа информирует о том, что текущий диск (диск, вставленный в устройство "A") имеет плотность записи в 1024 байт на сектор, а затем запрашивает, с какой плотностью записи желательно сформатировать диск, вставленный в устройство "B". В данном случае выбрана плотность в 1024 байт на сектор (двойная плотность записи). Затем программа запрашивает подтверждения запроса (это является мерой предосторожности, чтобы не допустить ошибки в имени диска, плотности записи и т. д.). Затем начинается форматирование диска. При этом распечатывается строка, состоящая из ряда чисел, которые представляют собой номера дорожек диска (в этом примере – для двустороннего диска, следовательно, процесс форматирования затрагивает обе стороны диска, и поэтому указаны номера 77 дорожек). По мере завершения форматирования каждой дорожки под соответствующим номером печатается символ "F" (сокращение от английского FORMAT – формат). После того как форматирование на всех дорожках завершено, программа последовательно проверяет каждую дорожку, с тем чтобы проконтролировать пра-

вильность формата. Для дорожек с правильным форматом под символом "F" печатается символ "V" (от английского Verify – проверить). Для дорожек с неправильным форматом под символом "F" печатается символ "E" (от английского Error – ошибка), если при этом возникла неисправимая ошибка, и символ "S" (от английского Soft – мягкий), если ошибка была исправлена (т. е. она возникла, но при повторной попытке была выявлена). Большое количество символов "S" говорит о том, что диск старый и сильно изношен. Большое количество символов "E" говорит о том, что этим диском дальше нельзя пользоваться.

ГЛАВА 19



Версии ОС *CP/M*:

1.3 1.4 2.0 2.2

Тип команды:

транзитная

ОБЩИЕ СВЕДЕНИЯ

Транзитная команда SYSGEN (сокращение от английского System Generation – генерация системы) является основной и необходимой частью операционной системы *CP/M*. Она используется в первую очередь для копирования действующей версии ОС *CP/M* с одного диска на другой (т. е. для "генерации" ОС *CP/M*). Действующая версия ОС *CP/M* представляет собой ту реальную операционную систему, которая загружается в оперативную память ЭВМ каждый раз при включении вычислительной системы и проведении начальной загрузки. Пользователю, который впервые имеет дело с ОС *CP/M*, может показаться, что эта версия хранится в виде файла где-то на диске. В основном это так и есть, но поскольку к файлу ОС *CP/M* можно обращаться только *после* загрузки ОС *CP/M* в оперативную память, должен существовать другой способ "оживления" ОС *CP/M* и размещения ее в оперативной памяти с последующим запуском.

Основная программа операционной системы *CP/M* хранится на первых двух дорожках диска – нулевой и первой. Эти две дорожки являются специальными и их нельзя использовать под другую программу ОС *CP/M*. Их называют "системными дорожками". Во всех гибких дисках, совместимых с ОС *CP/M*, эти две дорожки используются исключительно для хранения ОС *CP/M*. К ОС *CP/M* нельзя получить доступ при помощи стандартных команд манипулирования файлами, таких как PIP, STAT, DIR и т. д. Более того, для копирования ОС *CP/M* с одного диска на другой используется специальная программа. Эта программа называется "SYSGEN". Название "SYSGEN" является сокращением от английского System Generation и обозначает, что эта программа "генерирует" операционную систему с одного диска на другой. Помимо этого программа "SYSGEN" используется для записи на другой диск новой версии модифицированной операционной системы (обычно содержа-

щей изменения в секции драйверов ввода-вывода BIOS). В этом качестве программа обычно используется только программистами для внесения в систему тех или иных изменений, например в секцию BIOS (см. книгу [6]). Напомним, что при включении вычислительной системы программа загрузки или программа монитор считывают программный код ОС *CP/M* с первых двух дорожек диска и размещают его в соответствующей области оперативной памяти.

Некоторые ЭВМ, на которых используется ОС *CP/M*, оснащены жесткими дисками взамен, или в дополнение, к обычным гибким дискам. В некоторых из этих ЭВМ ОС *CP/M* хранится не на первых двух дорожках. Способ обращения к ней меняется в зависимости от ее конфигурации. В результате этого многие разработчики модифицировали программу "SYSGEN" для удовлетворения нужд конкретной модели ЭВМ.

В этой главе не описывается, как программа "SYSGEN" работает на этих ЭВМ, поскольку существует слишком много различных вариантов конфигурации вычислительной системы. При работе на таких системах для получения информации о работе программы "SYSGEN" обратитесь к техническому руководству, поставляемому разработчиками. В этой главе объясняется, как работает только стандартная, немодифицированная программа, хранящаяся в файле "SYSGEN.COM" и предназначенная для обычной микро-ЭВМ, в которой используются в качестве основного средства хранения информации накопители на гибких дисках.

ОБРАЩЕНИЕ К ПРОГРАММЕ "SYSGEN"

Если программа "SYSGEN" используется впервые, то необходим гибкий диск, в первых двух дорожках которого уже записана действующая версия *CP/M*. Если ЭВМ или новое программное обеспечение устанавливаются впервые, то в первую очередь создается резервная копия всего содержимого системного диска. После того как новый резервный гибкий диск должным образом сформирован (при помощи программы форматирования дисков "FORMAT" или ее эквивалентной) для копирования всех файлов, в том числе с ОС *CP/M*, используется программа копирования, например "PIP" (см. описание команды PIP). Программа "SYSGEN" используется для того, чтобы выполнить последний шаг: создать копию операционной системы *CP/M*.

Транзитная программа "SYSGEN" обычно хранится в файле, который называется "SYSGEN.COM". Некоторые разработчики называют этот файл несколько по-другому, например "SYSGEN2.COM", чтобы отметить, что эта программа соответствует текущей версии ОС *CP/M* (версии 1.4, 2.0, 2.2 и т. д.). Как бы этот файл не назывался, он должен храниться на диске, вставленном в одно из устройств системы, обычно в устройстве "A". Кроме того, диск, на который предполагается копировать ОС *CP/M*, следует должным образом сформатировать (см. описание команды FORMAT в гл. 18).

СПОСОБЫ ИСПОЛЬЗОВАНИЯ КОМАНДЫ SYSGEN

SYSGEN

КОПИРОВАНИЕ ОС CP/M

В следующем примере приведены вопросы, которые программа "SYSGEN" выдает пользователю в процессе копирования ОС *CP/M* с диска "A" на диск "B". Поскольку все эти вопросы сформулированы очень точно, отвечать на них достаточно легко.

```
A>SYSGEN
SYSGEN VER X.X
SOURCE DRIVE NAME (OR RETURN TO SKIP)A<(CR>
SOURCE ON A, THEN TYPE RETURN<(CR>
FUNCTION COMPLETE
DESTINATION DRIVE (OR RETURN TO REBOOT)B<(CR>
DESTINATION ON B, THEN TYPE RETURN<(CR>
FUNCTION COMPLETE
DESTINATION DRIVE (OR RETURN TO REBOOT)<(CR>
```

A>

Отметим, что первым промптом (запросом), напечатанным после ввода команды SYSGEN, является "SOURCE DRIVE NAME (OR RETURN TO SKIP)" (введите имя исходного диска или нажмите клавишу "RETURN", если пропуск). Здесь запрашивается имя устройства, содержащего диск, на котором хранится действующая версия ОС *CP/M*. В качестве ответа на этот промпт введите имя устройства, содержащего диск, с которого предполагается выполнять копирование. Здесь в отличие от команд ОС *CP/M* после имени диска вводить символ ":" (двоеточие) не надо. Программа "SYSGEN" автоматически опознает это имя как имя имеющегося диска, при условии, что такое имя вообще может быть опознано ОС *CP/M**. В примере указано имя "A", поскольку, как обычно, устройство "A" является основным системным диском.

После того как следом за именем диска, введенным в качестве ответа на первый промпт, нажата клавиша RETURN, на консоль выдан второй промпт "SOURCE ON A, THEN TYPE RETURN" (если исходным является диск "A", то нажмите клавишу "RETURN"). В этом случае программа "SYSGEN" запрашивает пользователя, уверен ли он в том, что исходным диском должен быть диск "A". Если допущена ошибка, то можно завершить выполнение программы вводом команды CTRL/C (ренициализация системы). Если имя исходного диска задано правильно, то необходимо нажать клавишу "RETURN". После этого программа "SYSGEN" загрузит содержимое первых двух дорожек диска в оперативную память. После того как ОС *CP/M* загружена в память, появилось подтверждающее сообщение "FUNCTION COMPLETE"**.

* Т. е. устройство с таким же именем подключено к ЭВМ. – Прим. перев.

** Выполнение функции закончено. Здесь имеется в виду выполнение функции загрузки ОС *CP/M* в оперативную память. – Прим. перев.

После загрузки ОС *CP/M* в оперативную память ЭВМ выдан промпт "DESTINATION DRIVE (OR RETURN TO REBOOT)" (введите имя выходного диска или нажмите клавишу "RETURN" для перезагрузки). В качестве ответа на этот промпт введите имя диска, на который требуется скопировать ОС *CP/M*. Имя введите также, как ответ на первый промпт. Затем повторяется описанный выше диалог, в результате которого подтверждается имя выходного диска. И снова после успешной записи ОС *CP/M* на первых двух дорожках диска появилось сообщение "FUNCTION COMPLETE". Затем снова повторяется промпт "DESTINATION DRIVE". Теперь можно скопировать ОС *CP/M* на другой диск, или на новый диск, вставленный в то же самое устройство. Если в этом есть необходимость, то просто введите имя выходного устройства. Повторение процесса возможно потому, что программный код ОС *CP/M* по-прежнему хранится в оперативной памяти. Для завершения работы программы "SYSGEN" просто нажмите клавишу "RETURN", после чего снова появится основной промпт ОС *CP/M* (A>, B> и т. д.).

При попытке создать копию ОС *CP/M* с диска, первые две дорожки которого содержат неправильную информацию, выдается следующее сообщение об ошибке:

```
A>SYSGEN  
SYSGEN VER X.X  
SOURCE DRIVE NAME (OR RETURN TO SKIP)A<CR>  
SOURCE ON A, THEN TYPE RETURN<CR>  
PERMANENT ERROR, TYPE RETURN TO IGNORE<CR>  
PERMANENT ERROR, TYPE RETURN TO IGNORE'C
```

A>

Это сообщение выдается, если диск, указанный как "исходный", имеет некорректный формат (например, неправильную плотность записи или формат, несовместимый с ОС *CP/M*). Помимо этого оно появится в том случае, если первые две дорожки диска имеют физические повреждения, и означает, что программа "SYSGEN" не может прочитать данные правильно. То же самое произойдет при попытке скопировать ОС *CP/M* на несформатированный или неправильно сформированный "выходной" диск (безусловно, если в устройство вообще не вставлен никакой диск, то возникнет та же самая ошибка). Отметим, что если в качестве ответа на сообщение об ошибке нажата клавиша "RETURN", то программа "SYSGEN" предпримет еще одну попытку выполнить операции чтения и записи. Если эти операции не удались, то выдается то же самое сообщение об ошибке. При получении такого сообщения для завершения выполнения программы "SYSGEN" следует ввести команду CTRL/C (реинициализация системы). Как только это сделано, появится основной промпт ОС *CP/M* (A>, B> и т. д.), и можно все повторить сначала.

НЕУДАЧНЫЙ ВЫБОР НА ДИСКЕ МЕСТА ДЛЯ ХРАНЕНИЯ ОПЕРАЦИОННОЙ СИСТЕМЫ

Одним из недостатков ОС *CP/M* является выбор места для хранения программного кода операционной системы. В результате этого оказывается, что дорожки 0 и 1 являются самыми внешними дорожками диска. Следовательно, они расположе-

ны наиболее близко к краю диска. Любой электрический разряд, произошедший вне дискового устройства, может задеть край диска, и, следовательно, в операционной системе, хранящейся на дорожках 0 и 1, в первую очередь возникнут искажения, и она станет недостоверной. Если уж это произошло, то загрузить ОС *CP/M* с такого диска невозможно. В подобном случае с помощью программы "SYSGEN" можно восстановить утерянную операционную систему и тем самым опять загружать ее с данного диска. Отметим, что при правильной постановке дела системный диск ОС *CP/M* должен всегда содержать файл с программой "SYSGEN".

ЗАПИСЬ ОС *CP/M* В ФАЙЛ

SYSGEN
SAVE 41 CP/M-56K.SYS

Кроме того, по команде SYSGEN можно создать файл, содержащий программный код ОС *CP/M*. Впоследствии этот файл может быть изменен программистом. Для записи ОС *CP/M* в файл программу "SYSGEN" используют совместно со встроенной командой SAVE. Команда SAVE была описана ранее.

Если программа "SYSGEN" используется для записи ОС *CP/M* в файл, то в качестве ответа на промпт с именем текущего диска также, как обычно, необходимо ввести команду SYSGEN.

В качестве ответа на промпт, запрашивающий имя исходного диска, введите имя диска, первые две дорожки которого содержат требуемую версию ОС *CP/M*. После этого ОС *CP/M* будет размещена в оперативной памяти ЭВМ, что отмечается сообщением FUNCTION COMPLETE. В качестве ответа на промпт, запрашивающий имя выходного устройства, просто нажмите клавишу "RETURN", поскольку не предполагается записывать систему на первые две дорожки другого диска. После этого выполнение программы "SYSGEN" будет завершено, и снова появится основной промпт ОС *CP/M*.

```
A>SYSGEN  
  
SYSGEN VER X.X  
SOURCE DRIVE NAME (OR RETURN TO SKIP) A<(CR)>  
SOURCE ON A, THEN TYPE RETURN<(CR>  
  
FUNCTION COMPLETE  
  
DESTINATION DRIVE (OR RETURN TO REBOOT)<(CR)>  
  
A>
```

Теперь программный код ОС *CP/M* временно хранится в оперативной памяти ЭВМ. Затем для сохранения содержимого оперативной памяти в файле на диске применяется команда SAVE. Прежде чем использовать команду SAVE, необходимо тщательно изучить ее описание (см. гл. 17), поскольку чрезвычайно важно хорошо понять, как она работает. В процессе чтения описания команды SAVE станет очевидно, что для записи всех необходимых данных в файл следует вычислить требующееся под эти данные примерное количество страниц оперативной памяти.

Следующий пример является продолжением предыдущего примера. Предположим, что ОС *CP/M* занимает 56К. Если выполнить все необходимые расчеты, то окажется, что требуемое число страниц оперативной памяти равно примерно 41. Это число может быть больше необходимого, но гораздо лучше работать с большим нежели с числом, меньшим точного значения. В этом примере ОС *CP/M* записана в файл с именем "CP/M-56K.SYS". Имя файла должно отражать размеры версии ОС *CP/M*, которая в нем хранится. Расширение должно указывать, что файл содержит машинный код, а не печатаемый текст:



```
A>SAVE 41 CP/M-56K.SYS
```

```
A>
```

```
A>DIR CP/M-56K.SYS
```

```
A: CP/M-56K SYS
```

```
A>
```

Отметим, что расширение ".SYS" указывает на то, что файл "CP/M-56K.SYS" содержит машинный код. Вместо расширения ".SYS" можно использовать также расширения ".COM", поскольку оно также указывает на то, что файл содержит машинный код. Однако следует избегать использования расширения ".COM" для файлов этого типа, поскольку обычно оно указывает на то, что файл содержит машинный код исполняемой транзитной программы. Поскольку файл "CP/M-56K.SYS" содержит ОС *CP/M*, которая не является транзитной программой, расширение ".SYS", или ему подобные, является предпочтительным.

КОПИРОВАНИЕ ОС CP/M ИЗ ФАЙЛА

SYSGEN CP/M-56K.SYS

Как описано в предыдущих разделах, программа "SYSGEN" может считывать программный код ОС *CP/M* из файла, созданного при помощи программы "MOVCPM", и записывать его на системные дорожки любого диска. Файл, указанный для этих целей*, должен быть первоначально создан при помощи программы "MOVCPM" или другим способом. Файл должен содержать программный код ОС *CP/M* в специальном формате, который называется перемещаемый машинный код. Для того чтобы применить программу "SYSGEN" подобным образом, введите команду SYSGEN, затем через пробел имя файла, содержащего ОС *CP/M*, и нажмите клавишу "RETURN".

В следующем примере показано перемещение файла с именем "CP/M-56K.SYS" на диск В:

```
A>SYSGEN CP/M-56K.SYS
DESTINATION DRIVE (OR RETURN TO REBOOT) B<(CR)>
FUNCTION COMPLETE
DESTINATION DRIVE (OR RETURN TO REBOOT)<(CR)>
A>
```

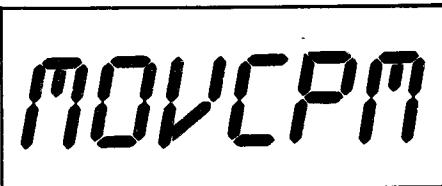
В этом примере, заметьте, пропущен промпт, запрашивающий имя исходного диска, поскольку в ОС *CP/M* можно передавать информацию об имени файла программе "SYSGEN". Следовательно, программа непосредственно рассматривает файл с указанным именем как исходный. В примере также предполагается, что файл "CP/M-56K.SYS" находится на диске А (поскольку диск А является текущим диском), и поэтому перед именем файла имя диска не указывается. Однако перед именем файла можно указать любое имя устройства, которое ОС *CP/M* опознает. Помните, что между командой SYSGEN и именем файла (возможно с именем диска) всегда должен быть пробел.

Если по ошибке после команды SYSGEN указано имя несуществующего файла, то появится следующее сообщение об ошибке:

```
A>SYSGEN CP/M-01K.SYS
SYSGEN VER X,X
NO SOURCE FILE ON DISK
A>
```

Заметьте, что вслед за сообщением об ошибке сразу появится основной промпт ОС *CP/M* (A >, B > и т. д.).

* Имеется в виду файл, имя которого указывается после команды SYSGEN. –
Прим. перев.



Версии ОС CP/M:

1.3 1.4 2.0 2.2

Тип команды:

транзитная

ОБЩИЕ СВЕДЕНИЯ

Транзитная команда MOVCPM (сокращение от английского Move CP/M – Перемещение ОС CP/M) используется для настройки ОС CP/M на использование памяти другого объема*. На каждой ЭВМ, работающей с ОС CP/M, обычно используется вся доступная оперативная память. Поскольку этот объем на различных ЭВМ может меняться от 20К до 64К, то необходима настройка ОС CP/M. В результате изменения размеров с помощью команды MOVCPM ОС CP/M она в сущности перенастраивается на работу в другом адресном пространстве. Помимо этого команду MOVCPM применяют для перемещения ОС CP/M немного вниз и освобождения тем самым пространства для программ, которые необходимо хранить в оперативной памяти вне (и до) операционной системы CP/M.

Когда ОС CP/M устанавливается на ЭВМ впервые, системный диск содержит саму операционную систему и все команды. В версии 2.0 и выше ОС CP/M, хранящаяся на этом диске, обычно настроена на работу на ЭВМ с объемом оперативной памяти 20К. Для версий ОС CP/M, предшествующих версии 2.0, этот объем равен как минимум 16К. Это означает, что в этих версиях ОС CP/M использование всей оперативной памяти не предусматривается. Однако если вычислительная система обладает оперативной памятью большего объема, например 32К, 48К или 64К, то у пользователя может возникнуть желание настроить ОС CP/M на использование этого максимального объема. Это важно, поскольку если операционная система объемом 20К установлена на ЭВМ с объемом оперативной памяти 32К, то используются только 20К оперативной памяти, а остальные 12К остаются неиспользованными. Таким образом, команду MOVCPM применяют для настройки ОС CP/M с тем, чтобы использовать оперативную память и в полном объеме.

В версиях 1.3 и 1.4 ОС CP/M можно настроить на использование оперативной памяти объемом от 16К до 64К. В ОС CP/M версии 2.0 и более поздних это значение может изменяться в интервале от 20К до 64К. Выбор конкретного значения зависит от того, какой объем оперативной памяти реально доступен на ЭВМ. Для большинства моделей ЭВМ реальная ОС CP/M требует приблизительно 11К оперативной памяти, предоставляемая оставшийся объем памяти, указанный в команде MOVCPM, в область транзитных программ. Например, если система настроена на использование 64К оперативной памяти, то приблизительно 11К используются под собственно операционную систему,

* То есть перевод ОС CP/M на работу в другом адресном пространстве. – Прим. перев.

а остальные 53К доступны для временных потребностей (загрузка транзитных программ и т. п.). Информацию о том, на какие адреса настроена конкретная версия ОС *CP/M* и какой объем оперативной памяти остается свободным под транзитные программы, можно найти в книге [6].

В некоторых микро-ЭВМ имеется область памяти, которая зарезервирована для нужд, не связанных с выполнением ОС *CP/M*. Например, если в ЭВМ имеется 64К оперативной памяти и зарезервированные 4К используются под аппаратуру контроллера накопителей на гибких дисках, то настроить ОС *CP/M* на использование в точности 64К оперативной памяти невозможно. Операционной системе *CP/M* доступно использование только 60К. В некоторых ЭВМ под специальные нужды резервируется до 8К, таким образом, для ОС *CP/M* остается только 56К оперативной памяти.

Существует два основных способа использования команды MOVCPM, а именно:

1. Настроить ОС *CP/M* на использование всей оперативной памяти или указанного объема и передать ей управление.

2. Настроить ОС *CP/M* на использование всей или указанного объема оперативной памяти и записать ее в системный файл, а не передавать сразу управление ОС *CP/M* и затем переместить ее по команде SYSGEN на системные дорожки диска.

ОБРАЩЕНИЕ К КОМАНДЕ MOVCPM

Для обращения к команде MOVCPM необходимо, чтобы файл "MOVCPM. COM" хранился на диске, вставленном в дисковое устройство системы. Для нормального выполнения команды необходимо загрузить ОС *CP/M* в оперативную память. При инициировании команды MOVCPM настраивается версия ОС *CP/M*, находящаяся в оперативной памяти ЭВМ. В отличие от команды SYSGEN при выполнении команды MOVCPM обращения к первым двум дорожкам системного диска не происходит.

СПОСОБЫ ИСПОЛЬЗОВАНИЯ КОМАНДЫ MOVCPM

НАСТРОЙКА ОС *CP/M* НА ИСПОЛЬЗОВАНИЕ ВСЕЙ
ДОСТУПНОЙ ОПЕРАТИВНОЙ ПАМЯТИ И
ПЕРЕДАЧА ЕЙ УПРАВЛЕНИЯ

MOVCPM

Это простейший формат команды. Он предписывает ОС *CP/M* осуществить настройку самой себя на использование всей доступной в ЭВМ оперативной памяти (16К, 32К, 48К и т. д.). После этого управление передается вновь сформированной операционной системе. При таком формате команды ОС *CP/M* к записи на диск не подготавливается. Как это происходит, показано в следующем примере:

A>MOVCPM

CONSTRUCTING 48K CP/M VER. X.X

48K CP/M VERS. X.X

A>

В примере предполагается, что максимальный объем доступной в системе памяти составляет 48К. Поскольку из этой памяти для хранения собственно ОС *CP/M* требуется вполне определенная часть (обычно 11К), приблизительно 40К оперативной памяти доступны для удовлетворения временных нужд (загрузка транзитных программ и т. д.). В примере в первой строке наряду с номером версии ОС *CP/M* (1.4, 2.0 и т. д.) выдан объем памяти. Следующая строка является точной копией сообщения о загрузке ОС *CP/M* за исключением того, что объем памяти может отличаться, если в исходной ОС *CP/M* использован меньший или больший объем памяти.

НАСТРОЙКА ОС *CP/M* НА ИСПОЛЬЗОВАНИЕ
УКАЗАННОГО ОБЪЕМА ОПЕРАТИВНОЙ ПАМЯТИ
И ПЕРЕДАЧА ЕЙ УПРАВЛЕНИЯ

MOVCPM nn

Этот формат команды MOVCPM аналогичен предыдущему, за исключением того, что в данном случае указывается конкретный используемый объем памяти. Этот формат часто используется для оптимизирования объема памяти под ОС *CP/M* на конкретной вычислительной системе. Он также полезен в тех случаях, когда некоторая область оперативной памяти работает неправильно и ее необходимо изолировать, с тем чтобы ОС *CP/M* к ней не обращалась (при использовании этого формата команды MOVCPM ОС *CP/M* к записи на диск не подготавливается). Формат полезен для быстрого предоставления пространства программам, хранящимся в памяти вне ОС *CP/M*.

Для того чтобы ввести команду MOVCPM в этом формате, необходимо в качестве ответа на промпт с именем текущего диска ввести сначала "MOVCPM", пробел, а затем число (nn), представляющее собой объем оперативной памяти под ОС *CP/M*. Если, например, ОС *CP/M* должна использовать 56К, то после команды MOVCPM необходимо ввести число "56". После введения команды управление ЭВМ передается вновь сформированной ОС *CP/M*. Как это делается, показано в следующем примере:

A>MOVCPM 56

CONSTRUCTING 56K CP/M VER. X.X

56K CP/M VERS. X.X

A>

В этом примере предполагается, что максимальный объем памяти, используемый ОС *CP/M*, составляет 56К. Поскольку для хранения собственно ОС *CP/M* требуется определенный объем оперативной памяти, то для удовлетворения временных потребностей доступно примерно 42К.

Если после команды MOVCPM указано значение, превышающее общий объем доступной оперативной памяти, то ОС *CP/M* будет настроена на использование максимально допустимого объема. Если, например, будет предпринята попытка настроить ОС *CP/M* на использование 50К оперативной памяти, в то время как доступно только 48К, то в процессе выполнения команды MOVCPM все равно будет предпринята попытка настроить ОС *CP/M* на использование этого объема оперативной памяти. В результате возникнет ошибка, поскольку вся указанная память не может быть выделена ОС *CP/M*. Для ЭВМ с 48К оперативной памяти та же ошибка возникнет при любом значении объема памяти от 49К до 64К, поскольку для всей ОС *CP/M* в оперативной памяти еще недостаточно пространства. Полученное сообщение об ошибке в различных системах может быть различным, но наиболее часто выдается сообщение "DRIVE NOT READY" (устройство не готово), поскольку вновь сформированная ОС *CP/M*, или то, что от нее осталось, не сможет обратиться к системному диску. Для устранения этой ошибки необходимо перезагрузить систему для того, чтобы скопировать в оперативную память исходную версию ОС *CP/M*, хранящуюся на первых двух дорожках системного диска, и повторить все сначала. Это продемонстрировано в следующем примере:

```
A>MOVCPM 50  
CONSTRUCTING 50K CP/M VER. X.X  
DRIVE NOT READY
```

(В этом месте необходимо выполнить перезагрузку системы, чтобы загрузить в память исходную версию ОС *CP/M*.)

МИНИМАЛЬНЫЙ И МАКСИМАЛЬНЫЙ ОБЪЕМ ОПЕРАТИВНОЙ ПАМЯТИ, НЕОБХОДИМЫЙ ДЛЯ ВЫПОЛНЕНИЯ КОМАНДЫ MOVCPM

Для того чтобы команда MOVCPM выполнялась правильно, параметр nn должен иметь значение, лежащее в определенном интервале. В ОС *CP/M* версий 1.3 и 1.4 это значение должно лежать в интервале от 16К до 64К. В ОС *CP/M* версии 2.0 и более поздних это значение должно лежать в интервале от 20К до 64К. Если для ЭВМ с 64К оперативной памяти после команды MOVCPM указано значение "65" и более, то это приведет к ошибке. В этом случае будет предпринята попытка перевести ОС *CP/M* на использование "01К" оперативной памяти, поскольку максимально доступный объем оперативной памяти составляет 64К, и превышение его приведет к тому, что оставшаяся от 64К часть будет размещена, опять начиная с адреса 01К. Это в высшей степени нежелательно, поскольку вновь сформированная операционная система становится при этом неработоспособной. В подобных случаях необходима полная перезагрузка системы, так как будет предпринята попытка передать управление неработоспособной операционной системе.

НАСТРОЙКА ОС CP/M НА ИСПОЛЬЗОВАНИЕ ВСЕЙ ДОСТУПНОЙ ОПЕРАТИВНОЙ ПАМЯТИ И ПОДГОТОВКА ЕЕ К ЗАПИСИ В ФАЙЛ

MOVCPM * *

Формат команды MOVCPM такой же, как первый формат, описанный в этом разделе, за исключением того, что управление вновь сформированной операционной системе не передается. Вместо этого новая версия ОС CP/M размещается в оперативной памяти так, чтобы ее можно было записать на диск. Команда MOVCPM* предписывает ОС CP/M осуществить перевод на использование всей доступной оперативной памяти (16К, 32К, 48К и т. д.). После этого с помощью команды SAVE ОС CP/M (переведенная теперь на использование всей памяти) переписывается в файл, который обычно называется "CP/M-XX.COM" или "CP/M-XX.SYS", где "XX" – объем памяти для записываемой ОС CP/M. Затем по транзитной команде SYSGEN можно взять версию ОС CP/M и разместить на первых двух дорожках диска в указанном устройстве (до перезагрузки управление по-прежнему у исходной версии системы). Кроме того, с помощью команды SYSGEN ОС CP/M может быть записана на первые две дорожки диска прямо из оперативной памяти (после размещения там с помощью команды MOVCPM). До перезагрузки диска управление вновь сформированной системе не передается.

Для ввода команды MOVCPM необходимо в качестве ответа на промпт с именем текущего диска ввести сначала "MOVCPM", пробел, затем символ "*" (звездочка), еще один пробел и еще один символ "*". Если после команды MOVCPM или между двумя символами "*" не стоит пробел, то данная команда не будет выполнена по назначению. В следующем примере показано, как используется команда MOVCPM и как используются команды SYSGEN и SAVE для сохранения ОС CP/M на диске:

```
A>MOVCPM * *
CONSTRUCTING 64K CP/M VER. X.X
READY FOR "SYSGEN" OR
"SAVE 45 CPM64.COM"
```

A>

В этом примере предполагается, что максимальный объем оперативной памяти составляет 64К. После того как ОС CP/M переслана и размещена в памяти, выдано сообщение, информирующее пользователя о том, что для сохранения вновь сформированной ОС CP/M на диске можно использовать команду SYSGEN или команду SAVE. Отметим, что программа "MOVCPM" автоматически подсчитывает количество страниц оперативной памяти, которое должно быть указано при записи ОС CP/M на диск по команде SAVE. Сообщение содержит также предполагаемое имя для файла: "CPM64.COM". Можно назвать файл как угодно, но важно, чтобы имя файла идентифицировало файл как машинный или объектный код, а не текст. Для этих целей можно использовать расширение ".COM", но предполагается, что вместо него используется расширение ".SYS", поскольку расширение ".COM" обычно

указывает, что файл содержит исполняемый код транзитной программы. Файл, содержащий ОС *CP/M*, не является таковым.

РАЗМЕР СИСТЕМНОГО ОБРАЗА ОС *CP/M*

Отметим, что число, стоящее в сообщении после "SAVE . . .", отражает реальный объем памяти ОС *CP/M*. Как указано при описании команды SAVE, это число представляет собой количество 256-байтовых страниц, которые необходимо хранить в файле. В примере это количество равно 45, и поскольку $45 \times 256 = 11,250$, то можно сделать вывод, что ОС *CP/M* занимает около 11К байт оперативной памяти.

В следующем примере показано, как вновь сформированную ОС *CP/M* можно сохранить в файле "CPM-64K.SYS" с помощью встроенной команды SAVE:

```
A>SAVE 45 CPM-64K.SYS  
A>  
A>DIR CPM-64K.SYS  
  
A: CPM-64K  SYS  
  
A>
```

В следующем примере показано, как с помощью транзитной команды SYSGEN вновь сформированная ОС *CP/M* размещается на первых двух дорожках диска, вставленного в устройство "A":

```
A>SYSGEN  
SYSGEN VER. X.X  
SOURCE DRIVE NAME (OR RETURN TO SKIP)<CR>  
DESTINATION DRIVE (OR RETURN TO REBOOT)A<CR>  
DESTINATION ON A, THEN TYPE RETURN<CR>  
FUNCTION COMPLETE  
DESTINATION DRIVE (OR RETURN TO REBOOT)<CR>  
  
A>
```

Заметьте, что в этом примере в ответ на промпт "SOURCE DRIVE" (Исходное устройство) была нажата клавиша "RETURN". Это объясняется тем, что ОС *CP/M* уже находится в оперативной памяти. В ответ на промпт "DESTINATION DRIVE" (окончное устройство) вводится имя устройства, в данном случае "A". Это вынуждает программу "SYSGEN" переписать (т. е. записать снова) вновь сформированную ОС *CP/M* на первые две дорожки диска в устройстве "A". Выполнение команды SYSGEN можно продолжить, последовательно вводя имена устройств в ответ на выводимые промпты "DESTINATION DRIVE". Это возможно потому, что модифицированная версия ОС *CP/M* по-прежнему находится в оперативной памяти.

НАСТРОЙКА ОС *CP/M* НА ИСПОЛЬЗОВАНИЕ
УКАЗАННОГО ОБЪЕМА ОПЕРАТИВНОЙ ПАМЯТИ
И ПОДГОТОВКА ЕЕ К ЗАПИСИ В ФАЙЛ

MOVCPM nn *

Этот формат команды MOVCPM напоминает формат "MOVCPM nn" тем, что позволяет пользователю указать объем оперативной памяти, предоставляемый для записи в файл.

ляемый в распоряжение ОС *CP/M*. Он напоминает также формат "MOVCPM" тем, что не позволяет передавать управление вновь сформированной ОС *CP/M*. Команда "MOVCPMnn*" предписывает ОС *CP/M* осуществить перевод на использование указанного объема оперативной памяти (16К, 32К, 48К и т. д.). Команда MOVCPM позволяет пользователю создать версию ОС *CP/M*, которую можно сохранять в файле, при этом перед ОС *CP/M* остается пространство для системно-независимых программ. Для запуска команды MOVCPM в этом формате необходимо в качестве ответа на промпт с именем текущего диска ввести "MOVCPM" пробел, число, представляющее собой объем доступной оперативной памяти, еще один пробел и символ "*" (звездочка). После ввода командной строки ОС *CP/M* размещается в оперативной памяти с тем, чтобы ее можно было сохранить на диске с помощью транзитной команды SYSGEN или же с помощью встроенной команды SAVE. При использовании этой команды MOVCPM управление остается у исходной версии операционной системы. Вновь сформированная ОС *CP/M* управления не получает. Как это происходит, показано в следующем примере:

```
A>MOVCPM 48 *
CONSTRUCTING 48K CP/M VER. X.X
READY FOR "SYSGEN" OR
"SAVE 45 CPM48.COM"
```

A>

Отметим, что в данном примере ОС *CP/M* настроена на использование 48К оперативной памяти. Для того чтобы записать модифицированную версию ОС *CP/M* на первые две дорожки диска, можно воспользоваться командой SYSGEN. Кроме этого, версию ОС *CP/M* можно записать в файл с помощью встроенной команды SAVE. Если данные действия выполнялись на ЭВМ с объемом оперативной памяти 64К, то перед ОС *CP/M* останется свободная область объемом 64К – 48К = 16К, которую можно использовать для других программ, буферов ввода-вывода и т. д.

ГЛАВА 21

ОБЗОР ОС *CP/M* ВЕРСИИ 3.0 (*CP/M Plus*)

ВВЕДЕНИЕ

Последняя версия операционной системы *CP/M*, версия 3.0, которая называется "*CP/M Plus*", является дальнейшим развитием версии 2.2, описанной в книге, и по сравнению с предыдущими версиями содержит много изменений, которые как показывает практика, представляет собой значительные достижения в технологии 8-битовых операционных систем. Хотя в ОС *CP/M Plus* сохранены основные характеристики предыдущих версий ОС *CP/M*, внесенные изменения являются существенными как для опытных пользователей ОС *CP/M*, так и для программистов. Эта операционная система дополнена некоторыми новыми командами, некоторые из которых повышают удобство использования ОС *CP/M* в целом, а другие являются уникальными и могут выполняться только под управлением ОС *CP/M Plus*. В ОС *CP/M Plus*

перенесено большинство команд версии 2.2, и почти каждая из них усовершенствована. В этой главе представлены новые команды, а также изменения в командах, перенесенных из предыдущих версий.

ОПЕРАЦИОННАЯ СРЕДА

Различия в командах ОС *CP/M Plus* и предыдущих версий ОС *CP/M* являются существенными как для оператора микро-ЭВМ, так и для программиста. Частично это объясняется тем, что в саму операционную систему внесены изменения, в результате которых возрастает число различных типов операционной среды, в которой она может работать. Эти изменения непосредственно затрагивают программиста потому, что вследствие повышения гибкости операционной системы конфигурация системы, взаимодействие с системой и генерация программ стали более сложными. Для пользователя эти изменения отражаются в наличии новых команд, модификации команд предыдущих версий и, наконец, в изменении конфигурации системы, в которой работает ОС *CP/M Plus*.

Существуют две версии ОС *CP/M Plus*: с обычным (*CP/M Plus Non-Banked*) и расширенным (*CP/M Plus Banked*) адресным пространством. Благодаря постоянному развитию и снижению стоимости твердотельных устройств памяти необходимым и зачастую существенным моментом для пользователей микро-ЭВМ стало увеличение объема памяти их систем. Операционная система *CP/M Plus* с обычным адресным пространством, так же как и предыдущие версии ОС *CP/M*, предназначена для функционирования в системе с максимальным объемом оперативного запоминающего устройства 64 килобайт. Хотя ОС *CP/M* можно специально модифицировать для работы с большим объемом памяти в определенных системах с расширенным адресным пространством, в версии ОС *CP/M Plus* эта модификация стандартно встроена. Все команды и функции, имеющиеся в версии с обычным адресным пространством, имеются также и в версии с расширенным адресным пространством. Однако в версии с расширенным адресным пространством имеется несколько команд и системных функций, которых нет в версии с обычным адресным пространством. Эти различия на протяжении всей данной главы будут отмечаться.

В ОС *CP/M Plus* существенно изменены внутренние компоненты и модули ОС *CP/M*, такие как процедуры секции BIOS (Базовая система ввода-вывода), системные вызовы Базовой дисковой операционной системы (BDOS) и общий метод использования оперативной памяти. Эти изменения представляют интерес, в частности, для программистов, которые занимаются модификацией ОС *CP/M* или написанием программ общего назначения. Краткая справочная информация о системных вызовах секции BDOS ОС *CP/M Plus* приводится при сравнении различных версий ОС *CP/M* в конце гл. 2. Подробную информацию о внутренней структуре ОС *CP/M Plus* можно найти в книге [6].

ИСПОЛЬЗОВАНИЕ ОС *CP/M Plus*

Операционная система *CP/M Plus* используется в основном так же, как и предыдущие версии ОС *CP/M*, но с некоторыми изменениями. Эти изменения описываются в следующих разделах.

Загрузка системы CP/M Plus. Процесс начальной загрузки (холодный старт) ОС CP/M Plus протекает так же, как описано в гл. 2, за исключением одного момента. Во многих приложениях из-за специально видоизмененной архитектуры или внутренних компонентов, например таких, как секция BIOS, программный код системы является слишком большим для хранения на первых двух дорожках системного диска. В таком случае программный код операционной системы полностью или частично записывается в файл с именем "CPM.SYS", который должен храниться на диске, предназначенному для загрузки системы. Если программный код системы не является слишком большим и его можно хранить на первых двух дорожках системного диска, то данный файл является необязательным и может отсутствовать. По команде COPSYS, которая выполняет те же функции, что и команда SYSGEN в предыдущих версиях ОС CP/M, содержимое файла "CPM.SYS" копируется на первые две дорожки любого диска.

Стартовый командный файл. В процессе начальной загрузки ОС CP/M Plus каждый раз автоматически выполняется поиск файла с именем "PROFILE.SUB". Если этот файл найден, доступ к нему осуществляется так же, как и с помощью команды SUBMIT. Этот файл можно организовать так же, как любой файл, предназначенный для запуска с помощью команды SUBMIT. Если этот файл не найден, то ОС CP/M продолжает свою работу в обычном режиме. Этот файл может содержать любые команды, необходимые при каждой начальной загрузке или перезагрузке системы, а именно: установку даты и времени, назначение физических устройств логическим, печать текста на консоли и т. д.

Использование имен дисков и кодов пользователей. В ОС CP/M Plus коды пользователей и имена дисков используются так же, как в версиях 2.0 и 2.2. В ОС CP/M Plus системный промпт (промпт с именем текущего диска) расширен так, что выводится не только имя устройства (A >, B >, C > и т. д.), но и текущий код пользователя (установленный с помощью команды USER), если значение кода больше нуля. Поэтому если текущим является диск A, а текущей областью пользователя является область с кодом 0, то в качестве промпта выдается сообщение "A >" до тех пор, пока не специфицирован другой код пользователя, например "1" или "2", в результате чего промпт изменится на "1A >" или "2A >". В ОС CP/M Plus код пользователя в интервале от 1 до 15 включается в системный промпт независимо от того, какой диск является текущим. Кода пользователя нет в промпте только в том случае, когда текущим кодом пользователя является код "0".

Обращение к дискам и файлам и использование паролей (только в ОС CP/M Plus с расширенным адресным пространством) В ОС CP/M Plus обращение к дискам и файлам происходит в основном так же, как описано в гл. 2. Однако в ОС CP/M Plus с расширенным адресным пространством пользователю предоставляется возможность назначать пароли для дисков и файлов. Пароли можно назначить дискам и файлам с помощью специальных команд. Если диску назначен какой-либо пароль, то перед тем, как пользователь сможет к нему обратиться, система осуществляет запрос правильного пароля. Если файл имеет пароль, то этот пароль добавляется в конец спецификации файла в следующем формате:

x : filename.typ; password,

где "x:" – имя диска, "filename" – имя файла, "тип" – расширение (тип) файла и "; password" – соответствующий пароль. Существуют четыре режима защиты файла с помощью пароля, а именно:

READ (Чтение). Если для файла установлен режим защиты "READ", то любые операции чтения, записи, копирования, удаления или переименования могут быть выполнены только с использованием пароля.

WRITE (Запись). Если для файла установлен режим защиты "WRITE", то любые операции записи, удаления или переименования могут быть выполнены только с использованием пароля. При чтении или копировании файла указывать пароль не требуется.

DELETE (Удаление). Если для файла установлен режим защиты "DELETE", то удаление или переименование файла возможно только с использованием пароля. При выполнении операций чтения, копирования или записи в файл указывать пароль не требуется.

NONE (Ничего). Если для файла установлен режим защиты "NONE", то при обращении к нему указывать пароль не требуется.

Группирование дисков. В ОС *CP/M Plus* имеется новое средство, при помощи которого поиск файлов с транзитными командами и командных файлов может осуществляться автоматически в заданной группе дисков. Такой поиск выполняется только в том случае, если спецификация файла в командной строке не содержит имя диска. Искомый файл будет найден ОС *CP/M* только в том случае, если ему назначен атрибут "SYS" (системный). Атрибут "SYS" в основном такой же, как описанный в гл. 8 для команды STAT. Если этот атрибут не назначен искомому файлу, то ОС *CP/M* выдает сообщение "FILE NOT FOUND" (файл не найден). Группы файлов устанавливаются при помощи команды SETDEF.

Логические и физические устройства. Организация логических и физических устройств в ОС *CP/M Plus* существенно отличается от предыдущих версий ОС *CP/M*. ОС *CP/M Plus* поддерживает пять основных логических устройств, а именно:

CONIN: или KEYBOARD:	(ввод с консоли)	}	или CON: (консоль)
CONOUT:	(вывод на пульт)		
AUXIN:	(вспомогательный ввод)		
AUXOUT:	(вспомогательный вывод)		
LST: или PRINTER:	(вывод на печать)		или AUX: или AUXILIARY: (вспо- могательное устройство ввода-вывода)

В ОС *CP/M Plus* логическим устройствам можно назначать различные физические характеристики. Логическим устройствам можно назначать не только различные физические устройства, но и различные стандартные характеристики, такие как направление потока данных (только ввод, только вывод или двунаправленный обмен), скорость передачи данных в битах в секунду (bps), тип интерфейса (стандартный или нестандартный) и протокол обмена

(такой как X-ON/X-OFF). Эти характеристики, так же как имена физических устройств, назначаются логическим устройствам с помощью команды DEVICE. Кроме этого команду DEVICE можно использовать для вывода списка текущих назначений. Помимо стандартных физических устройств, поддерживаемых предыдущими версиями ОС CP/M (CRT:, TTY:, LPT: и т. д.), в ОС CP/M Plus в секции BIOS могут быть описаны специальные физические устройства, причем каждое из таких устройств должно иметь уникальное имя длиной не более шести символов. Эти специальные физические устройства могут обслуживаться специальными драйверами, встроенными в секцию BIOS, и могут в любое время назначаться и переназначаться логическим устройствам с помощью команды DEVICE. Примерами специальных устройств могут быть: MODEM 1 – некоторый modem или другое устройство передачи данных, DIABLO – печатающее устройство барабанного типа*, DOT1 – высокоскоростное устройство растровой мозаичной печати и т. д.

Установление даты и времени. В ОС CP/M Plus предусмотрена возможность модификации операционной системы с целью обращения к электронным часам реального времени, расположенным в аппаратуре вычислительной системы. Это позволяет с помощью команды DATE устанавливать текущее время и дату, а также при желании обращаться к ним. При наличии в системе такой возможности для записи в файлы отметки о дате и времени можно использовать команду INITDIR. Запись подобных отметок осуществляется только в следующих случаях: при создании файла, его корректировке или при проведении обеих операций. Отметки о дате и времени для индивидуальных файлов можно выдавать в оглавлениях файлов, если с командой DIR или DIR-SYS указан соответствующий параметр. Если при наличии возможности установки даты и времени обращение к часам системы происходит правильно, запись отметки для отдельных файлов осуществляется автоматически.

Использование метасимволов. Метасимволы используются в ОС CP/M Plus в основном так же, как в предыдущих версиях ОС CP/M (см. гл. 2), за исключением того, что здесь их можно использовать вместе с командой REN (RENAME). Для переименования группы файлов можно использовать метасимволы "*" и "?", размещая их в "старой" и "новой" спецификации файлов на тех же позициях. Например, для команды RENAME допустимыми являются следующие последовательности:

A > REN *.BK 1 = *.BAK

A > REN A*.BK? = B*.SK?

В то же время при использовании с командой RENAME последовательности символов выдается сообщение об ошибке

A > REN A*.TXT = B.T*

За исключением команды RENAME метасимволы в ОС CP/M Plus можно использовать так же, как в предыдущих версиях ОС CP/M.

* В устройствах барабанного типа печатающий механизм имеет набор дисков с литерами, размещенными по окружности этих дисков. Каждый такой диск соответствует одной позиции печати в строке. Скорость печати на устройствах барабанного типа, как правило, выше, чем на построчно печатающих устройствах. – *Прим. перев.*

ОБЗОР КОМАНД ОС CP/M Plus

В ОС CP/M Plus все команды подразделяются на три основных типа: встроенные, транзитные и специальные. Можно считать, что многие из команд аналогичны командам, имеющимся в предыдущих версиях ОС CP/M. Однако почти все эти команды функционально расширены. Для выполнения всех функций встроенных команд некоторые из них требуют наличия сопровождающего транзитного файла. Список команд ОС CP/M Plus приведен в последующих разделах.

Встроенные команды. Встроенные команды ОС CP/M Plus реализованы непосредственно в самой операционной системе, что позволяет выполнять их независимо от того, какой диск в настоящий момент является текущим. Однако три команды составляют исключение. Хотя файлы можно использовать как обычные встроенные файлы, для получения дополнительных специальных параметров каждой из них необходим объединенный транзитный файл, хранящийся на текущем диске. Этот дополнительный транзитный командный файл используется только для обращения с помощью встроенной команды. ОС CP/M Plus отличается от предыдущих версий ОС CP/M тем, что имена встроенных команд можно вводить с консоли как полностью, так и сокращенно длиной вплоть до трех символов. В представленной ниже таблице в левой колонке перечислены встроенные команды ОС CP/M Plus, в средней колонке – имена дополнительных транзитных файлов и в третьей колонке – соответствующая команда ОС CP/M версии 2.2 (если таковые имеются).

CP/M Plus	Дополнительный файл	CP/M версии 2.2
DIR	DIR.COM	DIR (Выполняет функции команды STAT) – (Выполняет функции команды STAT)
DIRS или DIRSYS	DIR.COM	REN ERA TYP или TYPE USE или USER SAV или SAVE
REN или RENAME ERA или ERASE TYP или TYPE USE или USER SAV или SAVE	RENAME.COM Отсутствует Отсутствует Отсутствует Отсутствует	REN ERA TYP USER SAVE

Транзитные команды. Транзитные команды в ОС CP/M Plus функционируют в основном так же, как и в предыдущих версиях ОС CP/M. Для выполнения транзитной команды необходимо, чтобы файл, содержащий соответствующую транзитную программу, имел расширение ".COM" и хранился на указанном диске. Кроме того, в ОС CP/M Plus файлы с расширением ".PRL" (страничные перемещаемые файлы) рассматриваются как файлы, которые могут содержать транзитные программы. Расширение ".PRL" является стандартным для файлов с транзитными программами в ОС MP/M-II и MP/M-86 (операционные системы для многих пользователей при одновременном решении нескольких задач). Кроме того, в ОС CP/M Plus используются некоторые транзитные команды предыдущих версий ОС CP/M, в то время как остальные заменены новыми командами. В представленной ниже таблице в левой колонке перечислены транзитные команды ОС CP/M Plus, а в правой – соответствующие команды ОС CP/M версии 2.2, если таковые имеются.

<i>CP/M</i> Plus	<i>CP/M</i> версии 2.2	<i>CP/M</i> Plus	<i>CP/M</i> версии 2.2
COPSYS	SYSGEN	LINK	дополнительная
DATE	—	MAC	дополнительная (выполняет функции команды ASM)
DEVICE	— (выполняет функции команды STAT)		
DUMP	DUMP	PATCH	—
ED	ED	PIP	PIP
FORMAT (специальная)	FORMAT (специальная)	PUT	—
GENCPM	MOVCPM	RMAC	дополнительная (выполняет функции команды ASM)
GENCOM	—	SET	— (выполняет функции команды STAT)
GET	—	SETDEF	—
HELP	—	SHOW	— (выполняет функции команды STAT)
HEXCOM	LOAD	SID	дополнительная (выполняет функции команды DDT)
INITDIR	—	SUBMIT	SUBMIT
LIB	дополнительная	XREF	дополнительная

Команды управляющих символов. В состав специальных команд ОС *CP/M* Plus входят такие команды, как CTRL/C, которая вызывает инициализацию системы, команда CTRL/S, которая задерживает вывод данных на консоль, команда CTRL/H, которая обеспечивает возврат курсора назад на одну позицию, и т. д. Все специальные команды в предыдущих версиях ОС *CP/M*, имеются также и в ОС *CP/M* Plus. В обеих версиях ОС *CP/M* Plus (как с обычным, так и с расширенным адресным пространством) команда CTRL/Q используется для возобновления перемещения данных на экране пульта после ввода команды CTPL/S (остановить перемещение). Операционная система *CP/M* Plus с расширенным адресным пространством дополнена несколькими новыми командами редактирования строк. Отметим, что эти дополнительные команды имеются только в версии ОС *CP/M* Plus с расширенным адресным пространством, в версии с обычным адресным пространством они отсутствуют. Дополнительными командами являются:

Новые команды редактирования строк (только в ОС *CP/M* Plus с расширенным адресным пространством)

CTRL/A	Перемещает курсор на одну позицию влево
CTRL/B	Перемещает курсор в начало строки, не изменяя ее содержимого. Если курсор находится в начале строки, то по команде CTRL/B он перемещается в конец строки
CTRL/F	Перемещает курсор на одну позицию вправо
CTRL/G	Удаляет символ, стоящий на той же позиции, что и курсор. Положение курсора при этом не меняется
CTRL/K	Удаляет все символы с текущей позиции курсора до конца строки

CTRL/W	Если команда CTRL/W введена в качестве первого после промпта символа, то она обеспечивает повторный вызов и вывод на пульт введенной перед этим командной строки, которая может относиться как к уровню операционной системы, так и к уровню исполняемой программы. Командная строка может быть запущена на повторное выполнение либо после введения последовательностей CTRL/U, CTRL/M, CTRL/J, либо после нажатия клавиши "RETURN". Если вызванная на повторное выполнение командная строка содержит какие-либо символы, то курсор перемещается в конец командной строки. Если после вызванной на повторное исполнение строки будет нажата клавиша "RETURN", то команда строка будет выполнена
CTRL/X	Эта команда усовершенствована по сравнению с предыдущими версиями ОС CP/M. При ее вводе удаляются символы, расположенные слева от курсора; при этом символы, расположенные справа от курсора, переносятся в начало командной строки

Ниже представлен краткий обзор всех команд ОС *CP/M Plus*.

DIR и **DIRSYS**. Команды DIR и DIRSYS (или DIRS) выполняют основные функции по распечатке оглавления файлов на диске. Команды DIR и DIRSYS выполняют одну и ту же функцию за исключением того, что команда DIRSYS используется для распечатки оглавления "системных" файлов, которым назначен атрибут SYS. Для выполнения различных функций вместе с командой можно задавать до восемнадцати различных параметров. Примененные параметры определяют тип информации, которую необходимо вывести, например выдать оглавление с отметками о дате и времени, выдать оглавление всех или указанных дисков, выдать список всех файлов и/или дисков за исключением указанных, выдать оглавление всех или определенных областей пользователя и т. д.

RENAME или **REN**. Эта команда имеет следующие отличия от аналогичной команды в предыдущих версиях ОС *CP/M*:

1. Можно использовать метасимволы, как описано выше в этой главе.
2. Для обращения к команде можно использовать как имя RENAME, так и имя REN.
3. В режиме диалога можно вводить существенную командную строку или имя команды без параметров. Если произведен ввод без параметров, то по команде осуществляется запрос промпта "NEW NAME" (Новое имя), а затем промпта "OLD NAME" (Старое имя).

4. Предоставляется возможность удалять существующий файл, имя которого совпадает с новым именем.

5. При использовании возможностей, указанных в пунктах 3 и 4, встроенная команда обращается к транзитной команде "RENAME.COM". Она должна храниться в текущем диске или на одном из связанных (сгруппированных) для этого дисков.

ERASE или **ERA**. Эта команда имеет следующие отличия от аналогичной команды в предыдущих версиях ОС *CP/M*:

1. Сообщение "ALL?" (подтвердить удаление всех файлов), выдаваемое после введения параметра "*.*" (все файлы), заменено сообщением "CON-

FIRM (Y/N)?"'. В ОС CP/M Plus это сообщение выдается при использовании в командной строке любых метасимволов (а не только ".*.*").

2. При использовании метасимволов в командную строку для подтверждения необходимости удаления каждого файла можно ввести дополнительный параметр "[CONFIRM]".

3. Только в ОС CP/M Plus с расширенным адресным пространством перед удалением файла в командную строку для запроса пароля можно ввести дополнительный параметр "[PASSWORD]", который указывает на необходимость запрашивать пароль прежде, чем удалять файл.

TYPE. Эта команда аналогична команде TYPE в предыдущих версиях ОС CP/M за исключением того, что если она вводится без параметров, то запрашивается необходимая для ввода спецификация файла. В эту команду введен дополнительный параметр. Для остановки печати после каждой 24-й строки до момента возобновления печати с помощью нажатия произвольной клавиши в командную строку можно дополнительно ввести параметр [PAGE].
USER. В ОС CP/M Plus в эту команду по сравнению с предыдущими версиями ОС CP/M внесены следующие изменения:

1. Команда USER вводится без параметров, при этом выдается промпт "ENTER USER #" (Ввести код пользователя).

2. Так же как в ОС MP/M, основной промпт ОС CP/M Plus помимо имени текущего диска осуществляет вывод кода пользователя (5A >, 10C > и т. д.) за исключением того случая, когда текущий код пользователя равен 0.

3. С помощью команды PIP файлы здесь можно копировать в любую область пользователя путем задания параметра [Gn]* в спецификации результирующего файла и (по усмотрению) еще одного параметра "[Gn]" в спецификации исходного файла.

4. Поиск файлов может осуществляться в других областях пользователя в порядке объединения дисковых устройств в группе при условии, что эти файлы имеют атрибут SYS (системный).

5. Команда STAT, имеющаяся в предыдущих версиях ОС CP/M, здесь заменена командой SHOW, которая выдает расширенную информацию о файлах; с помощью команды SHOW можно выводить оглавление файлов из других областей пользователя (всех или только некоторых).

SAVE. Команда SAVE в ОС CP/M Plus существенно отличается от аналогичной команды в предыдущих версиях ОС CP/M, а ее применение является чрезвычайно важным для программистов. Для сохранения данных, расположенных в памяти системы до загрузки данных в память, вводится без параметров команда SAVE. После ввода команды SAVE для загрузки конкретного файла используется основной способ загрузки данных в память с помощью программы SID (Символический отладчик команд). При завершении работы программы "SID"** автоматически исполняется команда SAVE и осуществляется запрос имени файла, который необходимо сохранить. Как только это имя введено, происходит проверка на наличие на диске како-

* Где "п" – код пользователя. – *Прим. перев.*

** Транзитная программа, которая вызывается на исполнение по команде SID. – *Прим. перев.*

го-либо файла с таким же именем*, в случае обнаружения выдается запрос о необходимости его удаления. Далее выдается промпт, запрашивающий начальный адрес сохраняемой области данных, а затем – промпт, запрашивающий конечный адрес этой области. После введения этой информации данные сохраняются в указанном файле.

COPSYS. Эта команда выполняет те же функции, что команда SYSGEN в предыдущих версиях ОС *CP/M* (см. гл. 19). Поскольку иногда программный код ОС *CP/M* Plus полностью или частично хранится в файле с именем "CPM.SYS", при выполнении команды COPSYS выдается запрос, должен ли данный файл участвовать в операции копирования.

DATE. В ОС *CP/M* Plus команда DATE используется для обращения к часам реального времени. Эту команду можно использовать следующими способами:

1. Ввести команду DATE для вывода даты и времени, а затем вернуть управление ОС *CP/M*.
2. Ввести команду DATE С для непрерывного счета текущего времени.
3. Ввести команду DATE (дата-время) для установки даты и текущего времени, а затем ввести ответ на промпт "PRESS ANY KEY TO SET TIME" ("Для установки текущего времени нажмите любую клавишу").
4. Ввести команду DATE SET для ввода даты и времени в режиме диалога.

Отметим, что эта команда выполняется только в том случае, если на микро-ЭВМ имеются системные часы, а ОС *CP/M* Plus модифицирована для доступа к часам. Приходится устанавливать текущие дату и время каждый раз при включении микро-ЭВМ в сеть или при перезагрузке операционной системы, если только в часовом механизме нет устройства восстановления с автономным питанием, а ОС *CP/M* Plus не модифицирована для использования этого механизма.

DEVICE. Эта команда выполняет функции команды STAT, имеющейся в предыдущих версиях ОС *CP/M*, для назначения логических и физических устройств. Команду DEVICE можно использовать следующими способами:

1. Ввести команду DEVICE без параметров для вывода заранее определенных пользователем физических устройств и их характеристик (Входное, Выходное, Последовательное, Обмен по протоколу XON/XOFF), а затем текущих назначений физических устройств логическим. После вывода выполнение команды приостановится в ожидании указаний о переназначениях устройств.
2. Ввести команду DEVICE NAMES для вывода только, как указано выше, списка заранее определенных пользователем физических устройств.
3. Ввести команду DEVICE VALUES для вывода только списка текущих назначений физических устройств логическим.
4. Ввести команду DEVICE (имя физического или логического устройства) для вывода информации только об этом устройстве.
5. Назначить физическое устройство логическому (так же, как в случае команды STAT) или не назначать логическое устройство никакому физическому устройству (т. е. имя логического устройства -NULL). В командную строку можно вводить дополнительные параметры, устанавливающие или от-

* Обращаем внимание читателя на то, что в данный момент ОС *CP/M* выполняет встроенную команду SAVE. – Прим. перев.

меняющие использование протокола XON/XOFF, а также определяющие скорость передачи данные в бит/с. Одному логическому устройству может быть назначено более одного физического устройства.

6. Назначить атрибуты физическому устройству (установление или отмена использования протокола XON/XOFF и/или определение скорости передачи данных в бит/с). Эти атрибуты переносятся вместе с последующим назначением устройства какому-либо логическому устройству.

7. Вывести или установить текущий размер экрана консоли (количество колонок и строк).

DUMP. Эта команда в основном такая же, как аналогичная команда в предыдущих версиях ОС *CP/M*, хотя в ОС *CP/M Plus* эта команда позволяет выводить данные объектного кода как в шестнадцатеричном формате, так и в формате кода ASCII. При этом шестнадцатеричное представление данных печатается слева, а соответствующее представление в коде ASCII – справа. Формат печати почти совпадает с форматом, выдаваемым по команде "D" (печатать) при использовании программы "SID" (см. приложение А) или программы "DDT" (см. гл. 15).

ED. В ОС *CP/M Plus* редактор строк "ED" имеет следующие отличия от аналогичной программы в предыдущих версиях ОС *CP/M*:

1. Наряду с именем исходного файла в командной строке может быть задано имя результирующего файла, отличное от имени исходного файла. Команду ED можно вводить без параметров, при этом программа "ED" сначала выдает запрос на имя выходного (исходного) файла, а затем – на имя выходного файла, указывать которое не обязательно.

2. В ОС *CP/M Plus* добавлены следующие новые команды программы "ED":

а) "n command" – Эта командная строка предписывает программе "ED" выполнять команду "command" над всеми строками вплоть до строки с номером "n".

б) "n X filespec ^Z" – Эта командная строка предписывает программе "ED" записать "n" строк в заданный файл "filespec". Если с помощью этой же команды ранее произведена запись в тот же самый файл, то строки, записанные при последующем использовании этой команды, добавляются в конец файла.

в) OX ^Z – Эта командная строка предписывает удалить библиотечный файл "X.OBJ.OBJ.OBJ.LIB".

г) OX filespec ^Z – Эта командная строка предписывает удалить файл, заданный спецификацией filespec.

FORMAT. В ОС *CP/M Plus* так же, как в предыдущих версиях ОС *CP/M*, команда FORMAT является нестандартной утилитой. Программа этого типа необходима для разметки гибких дисков в формате с различной плотностью записи для совместимости с ОС *CP/M*. Эта команда обычно предоставляется поставщиком ОС *CP/M Plus* и настроена на работу с тем конкретным типом дисковых устройств, которые имеются в данной вычислительной системе.

GENCPM. Эта команда выполняет в основном те же функции, что команда MOVCMP в предыдущих версиях ОС *CP/M* (см. гл. 20). Ее основная функция состоит в настройке ОС *CP/M* на более эффективное использование оперативной памяти вычислительной системы.

GENCOM. Эта команда используется в ОС *CP/M Plus* для подключения к су-

ществующим файлам транзитных команд специальной программы RSX (от английского Resident System Extensions – Расширения резидентной системы). Программа RSX является программой специального типа, которая может быть подключена к операционной системе, с тем чтобы модифицировать или расширить возможности секции BDOS ОС CP/M Plus. Файлы или модули программы RSX используются для того, чтобы перехватывать обращение к функциям секции BDOS и либо выполнить их и преобразовать в другие функции секции BDOS, либо заблокировать обращение к ним. Более подробную информацию о модулях программы RSX и о том, как их использовать, можно найти в книге [6].

GET. Эта команда используется для того, чтобы объявить некоторый файл источником команд и данных, которые в противном случае пришлось бы вводить с консоли вручную, и автоматически запускать его на выполнение. Подобный файл может содержать команды ОС CP/M, данные для уже выполняющихся программ и т. д. Дополнительными параметрами команды GET являются следующие параметры:

1. Параметр SYSTEM может быть использован для указания того, что ввод данных для любой исполняющейся программы должен осуществляться непосредственно из GET файла; если данный параметр опущен, то программа будет получать данные из GET файла только в случае необходимости.

2. Параметр ECHO указывает на то, что ввод из GET файла необходимо отображать на консоли; параметр NO ECHO указывает на то, что входные данные и команды должны быть не видны пользователю.

3. Параметр GET CONSOLE переключает ввод любых данных или команд обратно на ручной ввод с консоли.

HELP. Эта команда является стандартной командой ОС CP/M Plus, с помощью которой пользователь может получить вспомогательную информацию об указанных командах ОС CP/M Plus. Команда сопровождается исходным текстовым файлом "HELP.HLP", содержащим всю вспомогательную информацию, запрашиваемую с помощью команды HELP. Этот файл может быть преобразован в файл "HELP.DAT", с тем, чтобы в исходный текст можно было вносить изменения или дополнения с помощью текстового редактора или текстового процессора. Эта команда используется следующим образом:

1. При обращении к команде HELP в командной строке используется следующий синтаксис:

A > HELP {topic} {subtopic 1} ... {subtopic 8},

где "topic" – команда ОС CP/M Plus, "subtopic 1" – имя первого параметра данной команды, "subtopic 2" – имя второго параметра и т. д. В командной строке может быть перечислено до восьми параметров.

2. Параметр [P] используется для того, чтобы отменить паузы в печати через каждые 23 строки.

3. После каждого вывода вспомогательной информации до нажатия клавиши "RETURN" выдается промпт "HELP >".

4. Параметр [E] используется для того, чтобы преобразовать файл "HELP.HLP" в файл "HELP.DAT", с тем чтобы его можно было отредактировать с помощью текстового редактора или текстового процессора.

5. Параметр [C] используется для того, чтобы преобразовать файл "HELP.DAT" обратно в файл "HELP.HLP".

HEXCOM. Эта команда в точности соответствует команде LOAD (см. гл. 14) в предыдущих версиях ОС *CP/M*. Здесь изменено только название.

INITDIR. Эта команда используется для инициализации каталога диска для записи отметки даты и времени обращения к файлам. Эта команда может быть использована только в том случае, если в вычислительной системе имеются встроенные часы и ОС *CP/M Plus* настроена на работу с ними.

LIB. Команда обеспечивает обращение к транзитной программе ("LIB"), которая в предыдущих версиях ОС *CP/M* входит в набор дополнительных программ, а в ОС *CP/M Plus* включена в качестве стандартной. Описание программы "LIB" представлено в приложении А. Программа выполняет следующие основные функции:

1. Создание библиотечного файла, состоящего из нескольких файлов с перемещаемым объектным кодом (файлы ".REL").

2. Создание индексного библиотечного файла.

3. Построение списка имен модулей, содержащихся в библиотечном или файле с именем ".REL".

LINK. Команда является стандартной командой ОС *CP/M Plus* и дополнительной командой в предыдущих версиях ОС *CP/M Plus*. Эта программа осуществляет объединение различных файлов с перемещаемым объектным кодом и создание единого файла с выполняемой транзитной программой. Описание этой программы представлено в приложении А.

MAC. Эта команда обеспечивает обращение к транзитной программе ("MAC"), которая в предыдущих версиях ОС *CP/M* входит в набор дополнительных программ, а в ОС *CP/M Plus* включена в качестве стандартной. Программа "MAC" является макроассемблером и существенно расширяет возможности стандартного ассемблера (см. гл. 13) предыдущих версий ОС *CP/M*.

В ОС *CP/M Plus* команда ASM отсутствует. Описание программы "MAC" представлено в приложении Б.

PATCH. Эта команда используется для внесения изменений в операционную систему *CP/M Plus* или в файлы ОС *CP/M Plus*. Команда PATCH используется для автоматического введения изменений, предоставляемых фирмой Digital Research или поставщиком версии ОС *CP/M Plus*.

PIP. В ОС *CP/M Plus* команда PIP (см. гл. 9) имеет следующие отличия от аналогичной команды в предыдущих версиях ОС *CP/M*:

1. Для того чтобы скопировать файл в произвольную область пользователя, в спецификации результирующего файла может быть указан дополнительный параметр "[Gn]", где "n" – номер области пользователя (для того чтобы скопировать файл из произвольной области пользователя в спецификации исходного файла, может быть указан еще один параметр "[Gn]").

2. Параметр [A] можно использовать для того, чтобы автоматически присвоить файлу (файлам) атрибут ARCHIVE, так что копироваться будут только те файлы, которые модифицированы со времени создания последней резервной копии (при копировании каждому такому файлу будет проставлена отметка о времени создания последней копии).

3. Параметр [C] указывает на необходимость запрашивать у пользователя подтверждение на копирование каждого файла.

4. (Только для ОС *CP/M Plus* с расширенным адресным пространством). Если для исходного файла установлен режим защиты READ (Чтение, см. опи-

сание команды SET), то в спецификации исходного файла необходимо указать пароль; если спецификация результирующего файла не задана, то ему назначаются все атрибуты защиты, устанавливаемые командой SET.

5. Атрибуты DIR, SYS, R/O и R/W при копировании переносятся всегда.

6. Так же, как в предыдущих версиях ОС CP/M, в командных строках команды PIP могут быть заданы физические и логические устройства с теми исключениями, которые рассмотрены в этой главе ранее.

7. В командной строке команды PIP имена логических устройств могут быть наряду с именами файлов, что будет соответствовать первой, второй, . . . , последней секции данных.

PUT. Команда PUT аналогична команде GET за исключением того, что она переназначает вывод на консоль или печатающее устройство в файл с указанным именем. Эта команда имеет те же дополнительные параметры, что и команда GET, плюс параметры FILTER/NO FILTER: параметр FILTER используется для преобразования всех управляющих символов, включая символы управления экраном в пару печатаемых символов в коде ASCII*. Для того чтобы восстановить обычный режим вывода, необходимо ввести команду PUT CONSOLE или PUT PRINTER в зависимости от того, какая команда указана перед этим.

RMAC. Данная команда является стандартной в ОС CP/M Plus и дополнительной в предыдущих версиях ОС CP/M. Эта программа выполняет те же функции, что и программа "MAC", за исключением того, что она используется для создания файлов с перемещаемым объектным кодом вместо файлов с абсолютным объемным кодом. Для объединения объектных файлов и создания файлов с исполняемой транзитной программой используется команда LINK. Описание программы "RMAC" представлено в приложении Б.

SET. Эта команда предоставляет следующие возможности:

1. Устанавливать для файла защиту по паролю для чтения, записи, удаления (только в ОС CP/M Plus с расширенным адресным пространством).

2. Назначать диску условное имя (метку), пароль или назначить пароль файлу (только в ОС CP/M Plus с расширенным адресным пространством).

3. Назначать файлу общий пароль, который используется тогда, когда в командной строке после имени файла пароль не указан; в зависимости от установленной для файла защиты можно работать с файлом только в том случае, если его пароль совпадает с общим паролем (только в ОС CP/M Plus с расширенным адресным пространством).

4. Устанавливать или отменять для всех файлов на диске запись отметок о дате и времени создания (CREATE) последнего обращения на чтение (ACCESS) и/или запись (UPDATE).

5. Назначать или отменять файлам атрибут ARCHIVE (атрибут ARCHIVE автоматически устанавливается при использовании команды PIP с параметром [A]); наличие данного атрибута у файла можно определить как с помощью команды DIR, так и с помощью команды SHOW.

6. Назначать файлу атрибуты DIR, SYS, R/O, R/W так же, как в случае ис-

* Параметр NO FILTER используется для отмены этого преобразования. – *Прим. перев.*

пользования команды "STAT", но вместо использования символа "⟨" зна-
чения атрибутов заключают в скобки.

7. Назначать или отменять файлу атрибуты с F1 по F4, определяемые поль-
зователем.

SETDEF. Данная команда используется для объединения дисков в группы.
Группа дисков позволяет осуществлять автоматический поиск файлов с
транзитными программами и командных файлов более чем на одном диске.

Доступ к файлу, поиск которого производится таким способом, осущес-
твляется только тогда, когда ему назначен атрибут SYS (системный). Коман-
да SETDEF используется для обеспечения следующих возможностей:

1. При выводе команды SETDEF без параметров выдается список имен
текущей группы дисков в порядке поиска.

2. Назначение диска для хранения временных (с расширением ".QQQ")
файлов.

3. Назначение порядка поиска для не более чем четырех устройств.

4. Назначение поиска типа файла ("*.COM", "*.SUB" или оба).

SHOW. Эта команда в основном выполняет те же функции, что команда
STAT в предыдущих версиях ОС CP/M для вывода характеристики диска.
Команда используется следующими способами:

1. По команде SHOW x: выдается объем свободного пространства на дис-
ке с именем "x:".

2. По команде SHOW LABEL выдается расширенная информация о диске:
условное имя (метка) диска, требуется ли при обращении к диску указы-
вать пароли (только в ОС CP/M Plus с расширенным адресным про стран-
ством), инициирована ли запись отметок о дате и времени создания файлов
и последнего обращения к ним, дата и время создания и последнего обновле-
ния метки диска.

3. По команде SHOW USERS выдается та же информация, что и по коман-
де STAT USR:, включая общее количество файлов каждого пользователя.

4. По команде SHOW DIR выдается число свободных входов в каталог
диска.

5. По команде SHOW DRIVE выдаются заданные характеристики текуще-
го дискового устройства.

SID. Эта команда является стандартной в ОС CP/M Plus и дополнительной в
предыдущих версиях ОС CP/M. Команда обеспечивает символьическую отлад-
ку программ. Она выполняет функции программы "DDT" (см. гл. 15) пре-
дыдущих версий ОС CP/M. Краткое описание команды приведено в прило-
жении А.

SUBMIT. Эта команда имеет несколько основных отличий от аналогичной
команды в предыдущих версиях ОС CP/M. А именно:

1. В исходном командном файле с именем "SUBMIT" строки могут начи-
наться с символа "<", который обозначает, что соответствующая строка со-
держит данные для исполняющейся в данный момент программы. Выполня-
ет функции команды XSUB, имеющейся в ОС CP/M версии 2.0 и 2.2. В ОС
CP/M Plus команда XSUB отсутствует. Стока, содержащая только символ
"<", обозначает возврат каретки (эквивалентно нажатию клавиши "RE-
TURN"), что в ОС CP/M можно использовать для завершения работы про-
граммы.

2. В командных файлах можно имитировать управляющие символы, для чего перед соответствующим символом помещается символ "```"; в процессе выполнения команды SUBMIT каждая такая пара символов будет преобразована в правильный управляющий символ кода ASCII. Для представления собственно символа "```" необходимо использовать пару "```", что аналогично использованию символа "```".

3. В ОС CP/M Plus имеется специальный командный файл "PROFILE.SUB", который запускается на выполнение каждый раз при начальной загрузке или перезагрузке системы.

4. Командные файлы могут иметь вложенность произвольной глубины. По концу командного файла, находящегося на нижнем уровне вложенности, управление передается файлу, находящемуся на уровне выше, и так продолжается до тех пор, пока управление не будет возвращено командному файлу, запущенному на выполнение изначально. Командные файлы на любом уровне вложенности могут содержать команду GET.

XREF. Эта команда используется для получения отчета о перекрестных ссылках в программах, предназначенных для обработки макроассемблером "MAC" или "RMAC". Краткое описание данной программы представлено в приложении А.

ПРИЛОЖЕНИЕ А ДОПОЛНИТЕЛЬНЫЕ ТРАНЗИТНЫЕ КОМАНДЫ

ВВЕДЕНИЕ

В настоящем приложении рассматриваются транзитные команды, обычно не входящие в стандартный набор ОС CP/M, но которые можно приобрести отдельно. Здесь описываются только те команды, которые приобрели за последние несколько лет значительную популярность у пользователей.

Все команды, описанные в этом приложении, являются транзитными программами, т. е. разработанными с учетом того, что в момент вызова они будут находиться на диске в виде файла с расширением ".COM" (означает команда). Многие из описанных программ предназначены для совместного использования с некоторыми трансляторами с языков высокого уровня (в соответствующем месте это будет отмечено). Сами трансляторы с языками высокого уровня описываются в приложении Б.

Для каждой программы, описанной в этом приложении, представлены только ее основные характеристики: как обратиться к программе, как про текает ее выполнение в среде ОС CP/M, кроме того, какие файлы необходимы для ее выполнения и какие создаются в результате использования.

В данном приложении описаны следующие дополнительные транзитные программы:

- "DESPPOOL". Программа печати в фоновом режиме (Digital Research)*.
- "SID". Символический отладчик для микро-ЭВМ, созданных на базе микропроцессора Intel 8080 и Intel 8085 (Digital Research).

* В скобках указывается наименование фирмы-разработчика. – *Прим. перев.*

- "ZSID". Символический отладчик для микро-ЭВМ, созданных на базе микропроцессора Zilog Z80 (Digital Research).
- "LINK-80" (Digital Research). Программа-сборщик* для файлов с перемещаемым объектным кодом, созданных транслятором "PL/I-80", макроассемблером "RMAC" фирмы Digital Research, макроассемблером "MACRO-80" фирмы Microsoft или другими программами, порождающими файлы с перемещаемым объектным кодом.
- "LINK-80" (Microsoft). Программа-сборщик для файлов с перемещаемым объектным кодом, созданных следующими трансляторами фирмы Microsoft: "MACRO-80", FORTRAN-80 и COBOL-80.
- "LIB-80" (Digital Research). Программа-библиотекарь для файлов с перемещаемым объектным кодом, созданных трансляторами с языков высокого уровня или утилитами фирмы Digital Research.
- "LIB-80" (Microsoft). Программа обслуживания библиотек времени исполнения, которые используются для хранения объектных кодов подпрограмм, созданных трансляторами "MACRO-80", "FORTRAN-80" или "COBOL-80" фирмы Microsoft.
- "XREF" (Digital Research). Утилита, обеспечивающая построение отчета о перекрестных ссылках для программ, оттранслированных с помощью макроассемблера "MAC" или "RMAN" фирмы Digital Research.
- "CREF-80" (Microsoft). Утилита, обеспечивающая построение отчета о перекрестных ссылках, протокола межмодульных связей и точек вызова для программ, оттранслированных с помощью макроассемблера "MACRO-80" фирмы Microsoft.
- "XLT-86" (Digital Research). Утилита, обеспечивающая преобразование файлов с программами на языке ассемблера для микропроцессора Intel 8080 в файлы с эквивалентными программами на языке ассемблера для микропроцессоров Intel 8088, Intel 8086.

В оставшейся части данного приложения содержится краткое описание перечисленных выше программ.

ПРОГРАММА ФОНОВОЙ ПЕЧАТИ ("DESPool")

ОБЩИЕ СВЕДЕНИЯ

Программа "DESPool" является дополнительной транзитной программой ОС *CP/M*. С ее помощью можно выполнять распечатку файлов на системном печатающем устройстве. Печать этих файлов может проходить параллельно с исполнением других команд ОС *CP/M*, таких как PIP, STAT и т. д. Это осуществляется за счет резервирования в оперативной памяти специальной области фоновой печати, которая не занимается под другие программы, пока исполняется программа "DESPool".

Обычно в ОС *CP/M* одновременно может выполняться только одна команда. Следовательно, во время вывода содержимого файлов на печатающее устройство придется дожидаться завершения печати, прежде чем выполнить другую команду. Это может быть особенно неудобно в тех случаях, когда

* Программы, выполняющие сходные функции, в отечественной литературе иногда называются "Редактор связей". – *Прим. перев.*

устройство печати работает медленно, поскольку распечатка файла большого размера на длительное время заблокирует ЭВМ. В программе "DESPOOL" эта проблема решается за счет того, что задача печати выполняется в фоновом режиме с более низким приоритетом, чем остальные команды, которые вводятся с консоли. Другими словами, печать происходит только тогда, когда от ЭВМ не требуется выполнение других, более важных, функций.

ОБРАЩЕНИЕ К ПРОГРАММЕ

Для того чтобы обратиться к программе "DESPOOL" и исполнить ее, необходимо чтобы файл "DESPOOL.COM" находился на диске, вставленном в одно из устройств системы. После того как команда DESPOOL введена в качестве ответа на промпт с именем текущего диска, выдается сообщение о том, что программа "DESPOOL" загружена в память и ожидает ввода команд с консоли. Для того чтобы распечатать содержимое какого-либо файла, необходимо ввести команду CTRL/F (одновременно нажать клавиши CTRL и F), после чего программа "DESPOOL" выдаст запрос об имени файла, который должен быть распечатан.

В то время как программа "DESPOOL" находится в оперативной памяти, можно вводить с консоли любые другие команды ОС CP/M независимо от того, выполняет ли программа "DESPOOL" распечатку содержимого какого-либо файла. При печати содержимого файла процесс печати получает более низкий приоритет, чем любая другая команда, выполняемая одновременно. Если при выполнении программой "DESPOOL" печати файла обратиться к команде, требующей значительной обработки и операций ввода-вывода, то можно заметить, что печать временно от времени замедляется. Насколько печать фактически замедляется и когда, зависит от ввода и выполнения других операций обработки. Однако сам ввод с пульта замедляется несущественно.

После того как файл распечатан, программа "DESPOOL" остается в оперативной памяти и может быть снова активизирована в любое время. Команда CTRL/C (реинициализация системы) не окажет влияния на программу "DESPOOL", пока та находится в оперативной памяти. Для того чтобы удалить программу "DESPOOL" из оперативной памяти, вводится команда CTRL/B, заменяющая команду CTRL/C. Помимо этого программа "DESPOOL" будет удалена из оперативной памяти при сбросе системы и при полной перезагрузке ОС CP/M. Программу "DESPOOL" можно перезагрузить в оперативную память в любое время. При попытке загрузить программу "DESPOOL", уже находящуюся в оперативной памяти, выдается сообщение о том, что программа "DESPOOL" уже активизирована.

СИМВОЛИЧЕСКИЙ ОТЛАДЧИК ДЛЯ МИКРОПРОЦЕССОРОВ

Intel 8080, Intel 8085 ("SID")

ОБЩИЕ СВЕДЕНИЯ

Программа "SID" представляет собой инструментальное средство символьской отладки программ, написанных на языке ассемблера для микропроцессора Intel 8080 или Intel 8085. Программа "SID" поставляется фирмой Digital Research. Эта программа выполняет в основном те же функции, что и программа "DDT" (Инструментальное средство динамической отладки),

входящая в стандартный набор транзитных программ ОС *CP/M*. Отличие заключается в том, что программа "SID" предоставляет пользователю дополнительные возможности по проверке и отладке программ.

Программа "SID" предоставляет пользователю следующие возможности:

- Выполнение всех команд программы "DDT" с некоторыми изменениями и расширениями.
- Полный просмотр и изменение содержимого оперативной памяти.
- Команды шестнадцатеричной арифметики.
- Получение полной информации относительно исполнения программы.
- Установление точек прерывания в реальном времени.
- Реассемблирование части кода отлаживаемой программы.
- Просмотр текста отлаживаемой программы и таблицы символов.
- Динамическая загрузка двух раздельных вспомогательных программ, трассировки и построения гистограмм.

Команды программы "SID" в основном совпадают с теми командами, которые выполняет программа "DDT". Помимо этого существует несколько команд, свойственных только программе "SID".

ОБРАЩЕНИЕ К ПРОГРАММЕ

Обращение к программе "SID" осуществляется так же, как к программе "DDT", описанной в гл. 15. Для обращения к программе "SID" необходимо, чтобы файл "SID.COM" хранился на диске, вставленном в одно из устройств системы. Команду SID можно вводить без параметров в качестве ответа на промпт с именем текущего диска (A >, B > и т. д.). При этом программа "SID" загружается в оперативную память без указания файла, предназначенного для просмотра и отладки. В таком случае команду загрузки отлаживаемого файла в оперативную память можно ввести позже.

Команду SID, кроме того, можно ввести вместе с именем файла. При этом в оперативную память будут загружены обе программы*. Специфицированный вместе с командой файл должен содержать машинный код какой-либо программы. Он может быть или в формате исполняемого машинного кода (т. е. расширение ".COM" или другие), или в формате неперемещаемого объектного (шестнадцатеричного) кода ("HEX" файлы).

Совместно с программой "SID" используются два специальных вспомогательных файла: "TRACE.UTL" и "HIST.UTL", каждый из которых можно загрузить в оперативную память с программой "SID" для выполнения специальных функций. Файл "TRACE.UTL" содержит программу трассировки, а файл "HIST.UTL" — программу построения гистограмм. Любой из этих файлов или оба можно загружать в оперативную память в процессе загрузки программы "SID" или после ее завершения.

Совместно с программой "SID" можно загрузить в оперативную память содержимое нескольких файлов. Например, файл с шестнадцатеричным кодом (расширение ".HEX") и соответствующий ему файл таблицы символов (расширение ".SYM"), созданные программой "LINK-80" фирмы Digital Research. Программа "SID" будет автоматически управлять памятью и по-

* Программа "SID" и отлаживаемая программа. – *Прим. перев.*

строением перекрестных ссылок. Для того чтобы специфицировать больше чем один файл, необходимо ввести сначала команду SID, затем пробел, имя первого файла (с расширением ".COM", ".HEX" или др.), еще один пробел и имя второго файла (с расширением ".SYM" или др.).

После загрузки программы "SID" в память выдается специальный промпт "#". Появление этого промпта всегда означает, что программа "SID" ожидает ввода команд с консоли. Все команды управляющих символов ОС CP/M (CTRL/S, CTRL/C и т. д.) можно использовать как обычные. По команде CTRL/C (реинициализация системы) работа программы "SID" завершится, а управление будет возвращено ОС CP/M. После завершения выполнения программы "SID" модифицированный файл можно сохранить на диске при помощи встроенной команды SAVE (см. гл. 16), точно так же, как в случае команды DDT.

СИМВОЛИЧЕСКИЙ ОТЛАДЧИК ДЛЯ МИКРОПРОЦЕССОРА Zilog Z80 ("ZSID")

ОБЩИЕ СВЕДЕНИЯ

Программа "ZSID" фирмы Digital Research представляет собой инструментальное средство символической отладки программ, написанных на языке ассемблера для микропроцессора Zilog Z80. Эта программа выполняет в основном те же функции, что и стандартная программа "DDT" или описанная выше дополнительная программа "SID". Отличие заключается в том, что программа "ZSID" имеет дополнительные возможности для проверки и отладки программ.

Программа "ZSID" предоставляет пользователю следующие возможности:

- Выполнение всех команд программ "DDT" и "SID" с некоторыми изменениями и расширениями.
- Полный просмотр и изменение содержимого оперативной памяти.
- Команды шестнадцатеричной арифметики.
- Получение полной информации относительно исполнения программы.
- Установление точек прерывания в реальном времени.
- Реассемблирование части кода отлаживаемой программы.
- Просмотр текста отлаживаемой программы и таблицы символов.
- Динамическая загрузка двух раздельных вспомогательных программ трассировки и построения гистограмм.

Команды программы "ZSID" в основном совпадают с теми командами, которые предусмотрены в программе "DDT" или "SID". Помимо этого существует несколько команд, свойственных только программе "ZSID".

ОБРАЩЕНИЕ К ПРОГРАММЕ

Обращение к программе "ZSID" осуществляется в основном так же, как к программе "DDT", описанной в гл. 15, или к только что описанной программе "SID". Для обращения к программе "ZSID" необходимо, чтобы файл "ZSID.COM" хранился на диске, вставленном в одно из устройств системы. Для того чтобы загрузить программу "ZSID" в оперативную память саму по себе, не указывая имени файла с отлаживаемой программой, можно в качестве ответа на промпт с именем текущего диска (A >, B > и т. д.) вве-

сти команду ZSID без параметров. При этом программа "ZSID" загружается в память, а команду загрузки файла с отлаживаемой программой можно ввести позже. Кроме того, команду ZSID можно ввести вместе с именем файла, при этом в оперативную память загружаются обе программы. Файл, указанный вместе с командой ZSID, должен содержать машинный код какой-либо программы. Он может быть или в формате исполняемого машинного кода (расширение ".COM" или др.), или в формате неперемещаемого объектного (шестнадцатеричного) кода (расширение ".HEX").

Программой "ZSID" используются два специальных вспомогательных файла "TRACE.UTL" и "HIST.UTL", каждый из которых можно загрузить в оперативную память с программой "ZSID" для выполнения специальных функций. Файл "TRACE.UTL" содержит программу трассировки, а файл "HIST.UTL" — программу построения гистограмм. Любой из этих файлов или каждый в отдельности можно загрузить в оперативную память в процессе загрузки программы "ZSID" или после его завершения.

С программой "ZSID" в оперативную память можно загрузить содержимое нескольких файлов. Например, загрузка объектного ("".HEX") файла и соответствующего ему файлу таблицы символов ("".SYM") приведет к тому, что программа "ZSID" автоматически будет управлять памятью для загрузки файлов и построения перекрестных ссылок. Для того чтобы ввести командную строку такого типа, необходимо ввести сначала команду ZSID, затем пробел, имя первого файла (с расширением ".COM", ".HEX" или др.), еще один пробел и затем имя второго файла (с расширением ".SYM" или др.).

После загрузки программы "ZSID" в оперативную память выдается промпт "#". Появление этого промпта всегда означает, что программа "ZSID" ожидает ввода команд с консоли. Все команды управляющих символов ОС CP/M (CTRL/C, CTRL/S) и т. д. можно использовать как обычные. По команде CTRL/C (реинициализация системы) выполнение программы "ZSID" завершится, а управление будет возвращено ОС CP/M. Точно так же, как и в случае программы "DDT" и "SID", модифицированный файл после завершения работы программы "ZSID" можно сохранить на диске при помощи встроенной команды SAVE, описанной в гл. 16.

ПРОГРАММА-СБОРЩИК ("LINK-80")

ОБЩИЕ СВЕДЕНИЯ

Программа "LINK-80" является дополнительной транзитной программой, которую можно добавить в стандартный набор транзитных программ ОС CP/M. Существует две версии этой программы, называемые "LINK-80". Одна поставляется фирмой Digital Research, другая — фирмой Microsoft. Обе версии данной программы выполняют в основном одинаковые функции, хотя ход выполнения и области приложений у них во многом отличаются. В последующих параграфах описаны характеристики каждой версии этой программы.

ОПИСАНИЕ ПРОГРАММЫ "LINK-80" ФИРМЫ Digital Research

Программа "LINK-80" фирмы Digital Research является дополнительной программой, которую можно ввести в стандартный набор транзитных про-

грамм ОС *CP/M*. С ее помощью можно осуществлять преобразование файлов с перемещаемым объектным кодом, созданных транслятором "PL/I-80", макроассемблером "RMAC" или любым другим транслятором, который порождает перемещаемый объектный код, в файлы с исполняемым машинным кодом. Эта программа, кроме того, обеспечивает сборку нескольких файлов с перемещаемым объектным кодом и создание единого файла с программой, загружаемой на исполнение по фиксированному адресу.

Программа "LINK-80" выполняет в основном те же функции, что и стандартная транзитная программа "LOAD" в ОС *CP/M*. Отличие заключается в том, что программа "LINK-80" создает два файла: файл с исполняемым машинным кодом (с расширением ".COM" для ОС *CP/M* и с расширением ".PRL" для ОС *MP/M*) и файл таблицы символов (с расширением ".SYM"), который впоследствии может быть использован в процессе отладки (см. описание программ "SID" и "ZSID").

ОБРАЩЕНИЕ К ПРОГРАММЕ "LINK-80" ФИРМЫ Digital Research

Для обращения к программе "LINK-80" необходимо, чтобы файл "LINK.COM" хранился на диске, вставленном в одно из устройств системы. Для того чтобы запустить программу, необходимо в качестве ответа на промпт с именем текущего диска (A>, B> и т. д.) ввести сначала команду LINK, а затем одно или несколько имен файлов. Если необходимо обработать только один файл, то его имя вводится после команды LINK, в той же командной строке. Если необходимо осуществить сборку нескольких файлов, то их имена вводятся через запятую.

Программа "LINK-80" создаст результирующий файл с тем же именем, что и у файла, указанного в командной строке первым. Для того чтобы специфицировать другое имя результирующего файла, необходимо сразу после команды LINK ввести это имя, знак равенства "=" и имена всех перемещаемых файлов.

В процессе работы программа "LINK-80" может создавать до восьми временных файлов (определенных по расширению ".OBJ"), которые находятся на диске до полного завершения операции сборки. Поэтому перед тем, как запустить программу "LINK-80", необходимо предварительно определить суммарный объем свободного дискового пространства с тем, чтобы проверить, достаточно ли его для выполнения всех действий.

При использовании программы "LINK-80" для обработки файлов с объектным кодом, созданных транслятором "PL/I-80", поиск всех стандартных подпрограмм, к которым имеются обращения в соответствующих этим файлам программах*, написанных на языке ПЛ/1**, будет проводиться в специально

* Поскольку при описании процесса сборки файлов с объектным кодом в единый файл с исполняемым кодом часто фигурируют две программы, одна из которых программа "LINK-80" (программа-сборщик), а другая та, которая будет получена в результате сборки, то в дальнейшем, чтобы избежать путаницы, для обозначения результирующей программы будет использоваться термин "программа", а для обозначения программы-сборщика – программа "LINK-80". – Прим. перев.

** Точнее, написанных на принятом в ОС *CP/M* диалекте языка ПЛ/1, трансляцию с которого осуществляет транслятор "PL/I-80" (см. приложение Б). – Прим. перев.

предназначенной для этого библиотеке времени исполнения. Этот библиотечный файл называется "PLILIB.IRL" и при применении программы "LINK-80" для обработки файлов, созданных транслятором "PL/I-80", должен всегда находиться на диске.

В командной строке команды LINK могут быть указаны различные параметры, которые определяют режим работы программы "LINK-80". Эти параметры заключают в квадратные скобки ("[" и "]"), как и при использовании параметров команды PIP, за исключением того, что здесь составные параметры разделяются запятой. Некоторые параметры указываются после имени первого файла, в то время как другие — после имени последнего файла.

Параметры команды LINK определены следующим образом:

- [A] Используется для обеспечения программы "LINK-80" дополнительным внутренним буфером для запоминания таблицы символов. Этот параметр является необязательным и должен использоваться только тогда, когда требуемый под таблицу символов объем памяти превышает установленный по умолчанию.
- [D] Используется для спецификации начального адреса COMMON-области и сегмента данных. Если параметр не указан, то эти сегменты размещаются непосредственно после программного сегмента. Параметр D вводится в формате "D nnnn", где "nnnn" — шестнадцатеричное число, представляющее собой начальный адрес области данных.
- [G] Используется для спецификации адреса, с которого необходимо начать исполнение программы. Если этот параметр не указан, то исполнение начнется с первого байта программы. Параметр G вводится в формате "Glabel", где "label" — алфавитно-цифровое имя, представляющее собой метку адреса передачи управления.
- [L] Используется для спецификации базового адреса результирующей программы, создаваемой программой "LINK-80". Если этот параметр не указан, то программа "LINK-80" автоматически назначит в качестве адреса загрузки шестнадцатеричный адрес 100, который является базовым адресом области транзитных программ в стандартной версии ОС CP/M. Параметр L вводится в формате "L nnnn", где "nnnn" — шестнадцатеричное число, представляющее собой адрес загрузки.
- [M] Используется при создании файлов с транзитными программами для операционной системы MP/M (файлы с расширением ".PRL"). Этот параметр указывает, что для правильного исполнения сформированной программы требуется дополнительное пространство под область данных. Параметр M вводится в формате "M nnnn", где "nnnn" — шестнадцатеричное число, представляющее собой требуемый объем оперативной памяти. Этот параметр никогда не используется при создании файлов с расширением ".COM" для ОС CP/M.
- [NL] Используется для запрещения вывода на консоль содержимого файла таблицы символов при выполнении программы "LINK-80".
- [NR] Используется для запрещения записи файла таблицы символов на диск.
- [OC] Используется для указания программе "LINK-80" записать результирующую программу на диск в файл с расширением ".COM". Использовать данный параметр нет необходимости, поскольку выполнение

- [OP] этой функции предусмотрено в программе "LINK-80" по умолчанию. Используется только в системе *MP/M* для создания файла с расширением ".PRL" вместо обычного файла с расширением ".COM".
- [P] Используется для установления начального адреса программного сегмента. Если параметр не указан, то программный сегмент будет размещен по адресу загрузки, равному шестнадцатеричному числу 100 (если параметр L не указан особо). Этот параметр вводится в формате "Рппп", где "ппп" – шестнадцатеричное число, представляющее собой начальный адрес программного сегмента.
- [Q] Используется для указания печатать символ "?" всякий раз при обращении программы "LINK-80" к библиотеке времени исполнения транслятора "PL/I-80". Если этот параметр не используется, то символ "?" не будет напечатан ни разу.
- [S] Указывает на то, что заданное после него имя следует трактовать, как имя библиотечного файла. Параметр "S" инициирует просмотр указанного библиотечного файла. При этом программа "LINK-80" включит в результирующий файл только те модули, которые содержат внешние имена, на которые были ссылки, но которые еще не были определены в ранее подключенных модулях.

Как было отмечено ранее, программу "LINK-80" фирмы Digital Research можно, кроме того, использовать для сборки объектных файлов, созданных трансляторами фирмы Microsoft. Однако при сборке файлов с объектным кодом в формате фирмы Microsoft необходимо соблюдать осторожность при использовании описанных выше параметров, поскольку не все из них в этом случае являются допустимыми.

Кроме того, отметим, что если одна и та же командная строка с большим количеством параметров используется часто, то ее удобно поместить в командный файл и впоследствии для выполнения той же самой операции вводить команду SUBMIT и имя этого файла.

ОПИСАНИЕ ПРОГРАММЫ "LINK-80" ФИРМЫ Microsoft

Программа "LINK-80" фирмы Microsoft является дополнительной транзитной программой, которая может быть добавлена в стандартный набор транзитных программ ОС *CP/M*. С помощью этой программы можно осуществлять преобразование файлов с объектным кодом, созданных макроассемблером "MACRO-80" или другим транслятором фирмы Microsoft (FORTRAN-80, COBOL-80 и др.), в файлы с исполняемым машинным кодом. Она, кроме того, предоставляет средства для сборки различных файлов с перемещаемым объектным кодом и создает единый файл с программой, загружаемой на исполнение по фиксированному адресу.

Программа "LINK-80" выполняет в основном те же функции, что и стандартная программа "LOAD" в ОС *CP/M*. Программа "LINK-80" фирмы Microsoft отличается от аналогичной программы фирмы Digital Research одной существенной чертой: вместо ввода всей спецификации в одной командной строке команду, запускающую на исполнение программу "LINK-80" фирмы Microsoft, можно ввести без параметров. Таким способом программа "LINK-80" выдает специальный промпт для ввода команд пользователя до тех пор, пока не дана команда завершения работы. Подобным образом ис-

пользуется программа "PIP". Другое отличие заключается в том, что по специальной команде программы "LINK-80" фирмы Microsoft может создавать файл таблицы символов, который позже можно использовать в процессе отладки (см. описание программы "SID" и "ZSID").

ОБРАЩЕНИЕ К ПРОГРАММЕ "LINK-80" ФИРМЫ Microsoft

Для обращения к программе "LINK-80" необходимо, чтобы файл "L80.COM" хранился на диске, вставленном в одно из устройств системы. Команду вызова программы "LINK-80" на исполнение можно вводить двумя способами. При первом в качестве ответа на промпт с именем текущего диска (A >, B > и т. д.) вводится команда L80, за ней все необходимые спецификации. При втором способе вводится только команда L80 и нажимается клавиша "RETURN". При таком способе программа "LINK-80" будет загружена в оперативную память и на пульт выдается специальный промпт, состоящий из символа "*". В качестве ответа на этот промпт можно вводить другие команды программы "LINK-80" и все спецификации обрабатываемых файлов.

Если команда L80 и все необходимые параметры вводятся в одной командной строке, то необходимо перечислить имена всех объединяемых файлов, отделив их друг от друга символом "/". В конце командной строки *перед тем, как нажата клавиша "RETURN"*, необходимо ввести параметр E (от английского Exit – выход). Если параметр E в командной строке отсутствует, то программа "LINK-80" воспримет все указанные параметры, но перейдет в режим ввода команд, что отмечается выдачей промпта "*". Программа "LINK-80" будет ожидать ввода других команд (на этой стадии завершить работу программы "LINK-80" и передать управление ОС CP/M при помощи параметра E). После того как процесс сборки завершен, программа "LINK-80" поместит результаты в результирующий файл с расширением ".COM". Этот файл имеет то же имя, что и файл с перемещаемым объектным кодом, указанный в командной строке первым.

Если команда L80 введена без параметров, то программа "LINK-80" загружается в оперативную память и на консоль выдается промпт "*". В качестве ответа на этот промпт можно ввести имена файлов, которые должны быть обработаны, так же как и другие команды, управляющие работой программы "LINK-80". После ввода каждого имени файла или команды программа "LINK-80" начинает его обработку. Как только программа готова воспринимать другую команду, выдается промпт "*".

Существует несколько параметров (ключей), которые можно специфицировать в командной строке команды L80 или вводить в качестве ответа на промпт "*". Эти параметры, кроме того, иногда называют командами. Они используются для управления работой программы "LINK-80". При вводе параметров в одной командной строке вместе с именами файлов их необходимо отделять запятой от имен тех файлов, к которым они относятся. Однако параметры должны вводиться вместе с именем файла после символа "/". Если в качестве ответа на промпт "*" параметры вводятся отдельно (в виде группы), то их необходимо отделять друг от друга символом "/". Если в качестве ответа на промпт "*" вводятся одиночные ключи, то каждому должен предшествовать символ "/".

Параметры команды программы "LINK-80" фирмы Microsoft определяются следующим образом:

- /G Предписывает программе "LINK-80" передать управление сформированной программе после того, как закончена ее сборка, и затем вернуть управление ОС *CP/M* после завершения программы. С помощью этой команды можно, кроме того, указать стартовый адрес, с которого необходимо начинать выполнение. Для этого параметр G вводится в формате "/G: nnnn", где "nnnn" – шестнадцатеричное число, представляющее собой стартовый адрес.
- /E Используется для завершения работы программы "LINK-80" и возврата управления ОС *CP/M* после выполнения всех необходимых действий. С помощью этой команды можно, кроме того, указать стартовый адрес, для которого необходимо выполнить сформированную программу перед тем, как вернуть управление ОС *CP/M*. Для этого параметр E вводится в формате "/E: nnnn", где "nnnn" – шестнадцатеричное число, представляющее собой требуемый стартовый адрес. При использовании этого формата управление ОС *CP/M* не возвращается до тех пор, пока программа не завершит свое выполнение.
- /N Указывает на то, что все предварительно загруженные программы и подпрограммы должны быть сохранены в файле, имя которого непосредственно предшествует этому параметру. Другой формой данного параметра является "/N: P". При этом в файл записывается только содержимое области транзитных программ.
- /P/D/R "/P" используется для установки начального адреса сформированной программы и области данных, "/D" – для установки начального адреса только области данных. Если параметр "/P" используется совместно с параметром "/D", то он определяет начальный адрес только для программы. Параметр "/R" используется для возврата программы "LINK-80" в начальное состояние.
- /S Указывает на то, что файл, имя которого указано непосредственно после него, является библиотечным файлом. Программа "LINK-80" будет просматривать библиотечные файлы, созданные программой "LIB-80" фирмы Microsoft, и искать те модули, которые могли быть, но еще не использованы в процессе сборки.
- /U/M "/U" используется для печати неопределенных внешних имен. Параметр "/M" задает печать всех внешних ссылок.
- /O/H "/O" устанавливает восьмеричное основание счисления, а параметр "/H" – шестнадцатеричное.
- /X/Y "/X" указывает на то, что вместо исполняемого машинного кода результирующий файл должен содержать неперемещаемый объектный (шестнадцатеричный) код (т. е. вместо расширения ".COM" результирующий файл должен иметь расширение ".HEX"). При использовании этого параметра необходимо, кроме того, указать параметр "/N". Параметр "/Y" предписывает программе "LINK-80" создавать файл таблицы символов с расширением ".SYM", который впоследствии может быть использован в процессе отладки с программами "SID" и "ZSID". При использовании этого параметра необходимо также указать параметры "/N" и "/E".

Как отмечалось ранее, программу "LINK-80" фирмы Digital Research можно использовать для сборки файлов с перемещаемым объектным кодом, созданных трансляторами фирмы Microsoft. При работе с библиотечными файлами необходимо соблюдать осторожность, поскольку возможно нарушение совместимости. Кроме того, программу "LINK-80" фирмы Microsoft нельзя использовать для обработки файлов, созданных трансляторами "PL/I-80", если при этом необходимо обращение к библиотеке времени исполнения транслятора "PL/I-80", поскольку программа "LINK-80" такого обращения не обеспечивает.

ПРОГРАММА ОБСЛУЖИВАНИЯ БИБЛИОТЕК ("LIB-80")

ОБЩИЕ СВЕДЕНИЯ

Программа "LIB-80" является программой обслуживания библиотек. Она позволяет группировать и запоминать в одном библиотечном файле различные часто используемые подпрограммы, представленные в перемещаемом объектном коде или в виде исходного текста. Существует две версии этой программы, каждая из которых называется "LIB-80". Одна поставляется фирмой Digital Research, другая – фирмой Microsoft. Эти две версии программы "LIB-80" выполняют в основном одинаковые функции, хотя несколько отличаются друг от друга. Обе версии обеспечивают хранение совместно используемых подпрограмм, которые можно при необходимости включать в различные программы. Это избавляет пользователя каждый раз вводить вручную тексты этих подпрограмм. В следующих двух разделах описывается каждая версия программы "LIB-80".

ОПИСАНИЕ ПРОГРАММЫ "LIB-80" ФИРМЫ Digital Research

Программа "LIB-80" фирмы Digital Research является дополнительной транзитной программой, которую можно добавить в стандартный набор транзитных программ ОС CP/M. Эта программа выполняет основные функции по управлению библиотечными файлами, содержащими произвольное число модулей в перемещаемом объектном коде. Программа "LIB-80" используется для объединения (группирования) нескольких файлов с перемещаемым объектным кодом в один библиотечный*, а кроме того, для создания индексных библиотек, выборки модулей и печати оглавления библиотеки.

ОБРАЩЕНИЕ К ПРОГРАММЕ "LIB-80" ФИРМЫ Digital Research

Для обращения к программе "LIB-80" необходимо, чтобы файл "LIB.COM" хранился на диске, вставленном в одно из устройств системы. Для того чтобы вызвать программу "LIB-80" на исполнение, необходимо в качестве

* Модули, запоминаемые в библиотеке, должны быть представлены в специальном формате, который позволяет реассемблировать их так, чтобы они могли исполняться в любой области оперативной памяти. Этот формат необходим потому, что заранее адрес загрузки программисту может быть неизвестен. Этот формат называется "перемещаемым". Это означает, что адреса в библиотечных модулях являются переменными и фактически выбираются при окончательной сборке.

ответа на промпт с именем текущего диска (A >, B > и т. д.) ввести команду LIB, а затем имена файлов и параметры, определяющие те функции, которые должны быть выполнены. Программу "LIB-80" можно использовать для следующих целей:

Создание библиотечного файла. В этом случае после команды LIB необходимо ввести имя файла, который должен быть создан, затем через пробел знак равенства ("=") и список имен файлов, которые необходимо сгруппировать и поместить вместе в результирующий библиотечный файл. Каждое имя файла отделяется от другого запятой. Если для результирующего или исходного файла не указан его тип (расширение), то программа "LIB-80" по умолчанию предполагает расширение ".REL". Для результирующего файла можно указать другие расширения (например, ".LIB"). Исходные файлы могут также иметь другие расширения при условии, что эти файлы содержат перемещаемый код.

Создание индексированного библиотечного файла. Для создания индексированного библиотечного файла командная строка должна выглядеть примерно так же, как описано в предыдущем пункте. Отличие заключается в том, что в конце командной строки вводится параметр "[I]". Если для имени результирующего файла не указано расширение, то программа "LIB-80" автоматически присвоит ему расширение "IRL".

Печать списка имен модулей. Для того чтобы распечатать имена всех модулей, содержащихся в перемещаемом ("REL") или библиотечном ("LIB") файле, после команды LIB необходимо ввести имя этого файла, а затем (без пробела) параметр "[M]" или "[P]". Если указан параметр "[M]", то программа "LIB-80" распечатает на консоли имена всех модулей. Если указан параметр "[P]", то программа "LIB-80", кроме того, распечатает имена всех модулей, однако дополнительно при этом будут распечатаны все имена типа PUBLIC. Имя типа PUBLIC представляет собой внешнее имя, на котором могут быть ссылки из других модулей.

ОПИСАНИЕ ПРОГРАММЫ "LIB-80" ФИРМЫ Microsoft

Программа "LIB-80" фирмы Microsoft является дополнительной транзитной программой, которая может быть добавлена в стандартный набор транзитных программ ОС CP/M. Эта программа обеспечивает выполнение основных функций по управлению библиотечными файлами, содержащими произвольное число модулей в перемещаемом объектном коде, созданных трансляторами фирмы Microsoft: MACRO-80, FORTRAN-80 или COBOL-80. Программу "LIB-80" можно, кроме того, использовать для объединения (группирования) различных программ, написанных на языке ассемблера, которые являются подпрограммами для трансляторов MACRO-80, FORTRAN-80 или COBOL-80. Кроме того, эту программу можно применить для создания специальных библиотек времени исполнения, содержащих специфические модули, разработанные программистом и необходимые ему в процессе исполнения его программ.

Программа "LIB-80" фирмы Microsoft отличается от аналогичной программы фирмы Digital Research тем, что команду LIB можно вводить как со всеми необходимыми параметрами и именами файлов, так и отдельно. В последнем случае выдается промпт, состоящий из символа "*", что позволит

вводить все необходимые имена и параметры в отдельных строках до тех пор, пока не задана команда завершения работы программы "LIB-80".

ОБРАЩЕНИЕ К ПРОГРАММЕ "LIB-80" ФИРМЫ Microsoft

Для обращения к программе "LIB-80" необходимо, чтобы файл "LIB.COM" хранился на диске, вставленном в одно из устройств системы. Для того чтобы вызвать программу "LIB-80" на исполнение, необходимо в качестве ответа на промпт с именем текущего диска (A>, B> и т. д.) ввести команду LIB, за ней имена файлов и параметры, определяющие выполнение требуемых функций. Команду LIB можно, кроме того, ввести отдельно, если имена файлов и параметры введены в отдельных строках. В этом случае программа "LIB-80" загружается в оперативную память и появится промпт "*". Этот промпт означает, что программа "LIB-80" ожидает ввода команд с консоли.

Программа "LIB-80" обеспечивает выполнение следующих функций:
Создание библиотечного файла. В этом случае после команды LIB необходимо ввести имя создаваемого файла, затем пробел, символ "=" и список имен файлов, которые должны быть сгруппированы и помещены в результирующий библиотечный файл. Имя каждого файла отделяется от остальных запятой.

Все исходные файлы должны содержать перемещаемый объектный код. Если для результирующего или исходного файла не указан его тип (расширение), то программа "LIB-80" по умолчанию будет предполагать расширение ".REL". Для результирующего файла можно указать другое расширение (например, ".LIB"). Исходные файлы могут, кроме того, иметь другие расширения при условии, что они содержат перемещаемый код. В командной строке команды LIB можно также перечислить имена модулей, принадлежащих указанному исходному файлу. Эти имена вводят после соответствующего имени исходного файла, но до запятой, и заключаются в угловые скобки ("<" и ">"). Если для одного файла необходимо указать имена нескольких модулей, то эти имена можно перечислить внутри одних скобок через запятую.

В командную строку команды LIB можно включать различные параметры (ключи). Эти параметры иногда называют командами и могут вводиться в единой командной строке, а также в качестве ответов на промпт "*". Перед параметрами необходимо указать символ "/" в формате "/параметр". Обычно параметры размещаются рядом с именем того файла, к которому они относятся. Параметры команды LIB определены следующим образом:
/E Используется для завершения работы программы "LIB-80" (в режиме командных строк) и возвращения управления ОС CP/M. Использовать этот параметр можно только при создании нового библиотечного файла или при внесении изменений в существующий. Во всех остальных случаях для завершения работы программы "LIB-80" необходимо использовать команду CTRL/C (реинициализация системы). Это важно, поскольку параметр "/E" предписывает программе "LIB-80" переименовать создаваемый (обновляемый) файл "LIB" в файл "REL" и уничтожить его предыдущую версию. Если параметр "/E" использован с существующим файлом, а файл не обновлен, то файл будет уничтожен, что нежелательно.

Этого, однако, не произойдет, если наряду с именем исходного существующего файла указано имя результирующего файла, поскольку программа "LIB-80" зашифтует при этом результирующий файл на диске.

- /R Используется для изменения расширения имени обрабатываемого файла ".LIB" на ".REL". При использовании этого параметра необходимо соблюдать те же меры предосторожности, что и при использовании параметра "/E". Использовать этот параметр можно только при создании нового библиотечного файла. Параметр "/R" выполняет те же функции, что и параметр "/E", кроме завершения выполнения программы "LIB-80". Используйте этот параметр, когда необходимо завершить обработку текущего файла, но желательно продолжить работу с программой "LIB-80".
- /L Предписывает программе "LIB-80" выдать на консоль список всех модулей, содержащихся в указанном файле, и определения всех внешних имен, имеющихся в модулях.
- /U Предписывает программе "LIB-80" выдать на консоль список всех неопределенных внешних имен, найденных при однократном просмотре библиотечного файла. Если в библиотечном файле какой-либо модуль содержит внешнее имя, которое относится к предыдущему модулю, то при использовании параметра "/U" это имя выдается на консоль.
- /C Используется для отмены всех ранее введенных команд без завершения работы программы "LIB-80". Создаваемый библиотечный файл уничтожается, и программа "LIB-80" начинает работу заново. Этот параметр полезен в том случае, когда в процессе работы специфицирован некорректный модуль или когда модули перечислены в ошибочной последовательности. В любом случае необходимо использовать параметр "/C", поскольку всю работу придется начать заново.
- /O Используется для установки восьмеричного основания системы счисления. При использовании этого параметра следует, кроме того, указать параметр "/L".
- /H Используется для установки шестнадцатеричного основания системы счисления. Шестнадцатеричное основание системы счисления установлено по умолчанию, поэтому использовать параметр "/H" нет необходимости, если только перед этим не указан параметр "/O".

ПРОГРАММА ПОСТРОЕНИЯ ОТЧЕТА О ПЕРЕКРЕСТНЫХ ССЫЛКАХ ДЛЯ ПРОГРАММ, НАПИСАННЫХ НА ЯЗЫКЕ АССЕМБЛЕРА ("XREF")

ОБЩИЕ СВЕДЕНИЯ

Программа "XREF" фирмы Digital Research является дополнительной транзитной программой, которая может быть добавлена в стандартный набор транзитных программ ОС CP/M. Программа "XREF" обеспечивает построение отчета о перекрестных ссылках для программ, написанных на языке ассемблера, и для получения информации об использовании переменных в конкретной программе просматривает файл печати и файл таблицы символов, созданные макроассемблером "MAC" или "RMAN".

ОБРАЩЕНИЕ К ПРОГРАММЕ

Для обращения к программе "XREF" необходимо, чтобы файл "XREF.COM" хранился на диске, вставленном в одно из устройств системы. Для то-

го чтобы запустить программу "XREF", необходимо в качестве ответа на промпт с именем текущего диска (A>, B> и т. д.) ввести команду XREF, затем имя файла без расширения и нажать клавишу RETURN. Имя, указанное в командной строке, относится к двум расположенным на диске файлам: файлу печати (файл с расширением ".PRN") и файлу таблицы символов (файлу с расширением ".SYM"), которые имеют то же имя.

Для корректного выполнения программы "XREF" необходимо, чтобы оба эти файла хранились на диске, в противном случае произойдет ошибка. После ввода команды XREF программа "XREF" начнет считывать каждую строку из файла печати (файл с расширением ".PRN"). При этом программа "XREF" присваивает каждой строке номер в виде префикса и копирует про-нумерованную строку в результирующий файл с тем же именем и расшире-нием ".XRF". Попутно программа "XREF" просматривает каждую строку в поисках всех имен, встречающихся в файле таблицы символов (файл с рас-шире-нием ".SYM"). После завершения процедуры копирования и поиска программа "XREF" добавит в результирующий файл (файл с расширением ".XRF") отчет о перекрестных ссылках. Такой отчет содержит номера всех строк, в которых встречается имя, определенное в таблице символов в фай-ле с расширением ".SYM". Вместе с тем программа "XREF" в каждой строке отметит флагом "#" символическое имя, которое входит в нее в качестве первой лексемы. Для каждого имени печатается, кроме того, значение, кото-рое присвоено ему макроассемблером "MAC" или "RMAC".

В командной строке команды XREF перед именем файла можно указать имя диска, при этом оба исходных файла (с расширением ".PRN" и с расшире-нием ".SYM") должны быть расположены на диске с указанным именем; результирующий файл (файл с расширением ".XRF") будет также записан на указанный диск.

Для вывода отчета о перекрестных ссылках на устройство печати (назна-ченное логическому устройству "LST:") в командную строку команды XREF можно добавить специальный параметр. Этот параметр управляет записью отчета о перекрестных ссылках на диск. Для вывода отчета о пере-крестных ссылках на печать после имени файла необходимо через пробел указать параметр "QP".

ПРОГРАММА ПОСТРОЕНИЯ ОТЧЕТА О ПЕРЕКРЕСТНЫХ ССЫЛКАХ ("CREF-80")

ОБЩИЕ СВЕДЕНИЯ

Программа "CREF-80" фирмы Microsoft является дополнительной тран-зитной программой, которая может быть добавлена в стандартный набор программ ОС CP/M. Программа "CREF-80" обеспечивает построение отчета о перекрестных ссылках для программ, написанных на языке ассемблера ОС CP/M, и используется для обработки специального файла печати, создан-ного макроассемблером "MACRO-80" фирмы Microsoft. Это позволяет по-лучить список всех межмодульных ссылок и точек определения внешних имен. Программа "CREF-80" создает отчет о перекрестных ссылках, который в дальнейшем можно использовать в процессе отладки.

ОБРАЩЕНИЕ К ПРОГРАММЕ

Для обращения к программе "CREF-80" необходимо, чтобы содержащий ее файл хранился на диске, вставленном в одно из устройств системы. Этот файл может называться "CREF80.COM", "CREF.COM" или "C80.COM". Последнее имя является наиболее распространенным.

Для вызова программы "CREF-80" на исполнение необходимо в качестве ответа на промпт с именем текущего диска (A>, B> и т. д.) ввести команду C80*, потом через пробел символ "=" и имя файла без расширения. В конце ввода командной строки необходимо нажать клавишу "RETURN". Имя, указанное в командной строке, относится к специальному файлу, созданному программой "MACRO-80" и имеющему расширение ".CRF". Программа "CREF-80" может обрабатывать файлы только этого типа.

В результате выполнения программы "CREF-80" создается результирующий файл, по своему содержанию напоминающий результирующий файл печати (файл с тем же именем и расширением ".PRN"). Отличие заключается в расширении ".LST". При вводе командной строки команды C80 программа "CREF-80" начнет считывать каждую строку из файла с расширением ".CRF". При этом она назначит каждой строке номер в виде префикса и скопирует ее в результирующий файл с тем же именем и расширением ".LST". Полутно программа "CREF-80" просматривает каждую строку в поисках символьических имен, которые встречаются в других операторах. После завершения процедуры копирования и поиска программы "CREF-80" добавит в результирующий файл (файл с расширением ".LST") отчет о перекрестных ссылках с номерами строк, в которых обнаружен каждый символ. Вместе с этим программа "CREF-80" в каждой строке отмечает флагком "#" символическое имя, которое входит в нее в качестве первой лексемы. Для каждого символического имени в отчете также представлено его значение, назначенное ему макроассемблером "MACRO-80".

При вводе команды C80 перед именем файла можно задать имя файла. Это укажет программе "CREF-80" на то, что файл с расширением ".CRF" находится в устройстве с указанным именем. Если результирующий файл с расширением ".LST" необходимо вывести на другой диск, то перед символом "=" следует указать имя требуемого поиска. Если результирующий файл с отчетом о перекрестных ссылках требуется записать под именем, отличным от имени исходного ".CRF" файла, то новое имя должно быть указано перед знаком равенства (без пробела).

Вместо записи отчета о перекрестных ссылках на диск в файл с расширением ".LST" программа "CREF-80" допускает применение имени логического устройства "LST:" и имени физического устройства "TTY:". При использовании одного из них имя следует указать перед знаком равенства (без пробела). Имя "LST:" используется для вывода содержимого файла с именем ".LST" на устройство печати, а имя "TTY:" – для вывода на пульт.

* Или CREF-80, или CREF – в зависимости от имени файла, в котором хранится программа "CREF-80". – Прим. перев.

**КОНВЕРТОР ПРОГРАММ НА ЯЗЫКЕ АССЕМБЛЕРА ДЛЯ МИКРОПРОЦЕССОРА
Intel 8080 В ПРОГРАММЫ НА ЯЗЫКЕ АССЕМБЛЕРА ДЛЯ МИКРОПРОЦЕССОРА
Intel 8086 ("XLT86")**

ОБЩИЕ СВЕДЕНИЯ

Программа "XLT86" фирмы Digital Research является дополнительной транзитной программой, которая может быть добавлена в стандартный набор транзитных программ ОС *CP/M*. Эта программа используется в тех случаях, когда необходимо выполнить преобразование программ на языке ассемблера для микропроцессора Intel 8080 в эквивалентные программы на языке ассемблера для микропроцессора Intel 8088 или Intel 8086. В процессе преобразования программа "XLT86" учитывает тип операционной системы, в которой она выполняется (*CP/M* или *MP/M*). В результате преобразования программа может функционировать или под управлением ОС *CP/M-86* или под управлением ОС *MP/M-86*. В программе "XLT86" применяются методы просмотра всех потоков данных для определения использования регистров микропроцессора Intel 8080, что позволяет уменьшить число операторов языка ассемблера для микропроцессора Intel 8088 или Intel 8086.

ОБРАЩЕНИЕ К ПРОГРАММЕ

Для обращения к программе "XLT86" необходимо, чтобы файл "XLT86.COM" (командный файл) и файлы "XLT00.OVL" и "XLT01.OVL" (оверлейные файлы) хранились на диске, вставленном в одно из устройств системы. Для вызова программы необходимо в качестве ответа на промпт с именем текущего диска (A >, B > и т. д.) ввести команду XLT86, за ней через пробел имя файла. Если имя файла указано без расширения, то программа "XLT86" по умолчанию предполагает расширение ".ASM" (файл с исходным текстом программы на языке ассемблера ОС *CP/M*). Специфицированный файл должен содержать исходный текст на языке ассемблера в формате, пригодном для обработки транслятором "ASM", "MAC" или "RMAC".

После завершения процесса преобразования программа "XLT86" создаст результирующий файл с тем же именем, что и исходный, в который поместит результаты преобразования. Этот результирующий файл, однако, имеет расширение ".A86" (стандартное расширение для файлов с программами на языке ассемблера в ОС *CP/M-86*). Кроме того, программа "XLT86" создаст файл печати с расширением ".PRN". Этот файл содержит список ошибок, обнаруженных в процессе преобразования. Содержимое файла печатидается в виде текста со вставленными знаками табуляции и символами смещения, т. е. может быть непосредственно выведено на устройство печати.

В процессе преобразования исходного файла программа "XLT86" обрабатывает операторы условного ассемблирования и помещает результаты в результирующий файл. Однако макроопределения, макровызовы и повторяющиеся циклы обрабатываться не будут. Для обработки макросов исходный файл должен быть предварительно оттранслирован при помощи макроассемблера "MAC" или "RMAC". Полученный в результате файл печати (файл с расширением ".PRN") следует изменить при помощи текстового редактора с тем, чтобы удалить содержимое первых двух столбцов (содержащих отасSEMBЛИРОВАННЫЙ код). Затем модифицированный файл необходимо пере-

именовать так, чтобы он имел расширение ".ASM", после чего этот файл может быть обработан программой "XLT86".

Для управления работой программы "XLT86" в командную строку можно добавлять параметры. Эти параметры вводят после имени файла и заключают в квадратные скобки ("[" и "]"). В одной командной строке можно вводить несколько параметров. Все параметры должны следовать друг за другом без пробелов и должны быть заключены в одни квадратные скобки.

Эти параметры определены следующим способом:

- [Ax] Предписывает программе "XLT86" поместить результирующий файл (файл с расширением ".A86") в указанное дисковое устройство. Символ "x" представляет собой имя дискового устройства (вводится без двоеточия).
- [B] Используется для создания списка основных блоков в результирующем файле печати (файл с расширением ".PRN").
- [C] Указывает программе "XLT86", что выполнение программ на микропроцессоре Intel 8086 производится с использованием "модуля сжатия".
- [J] Используется для преобразования команд условного перехода в условные команды с коротким форматом.
- [L] Используется для вывода файла печати (файл с расширением ".PRN") непосредственно на печатающее устройство.
- [N] Используется для вывода на консоль каждой обрабатываемой команды и ее порядкового номера.
- [Px] Используется для спецификации имени дискового устройства, в котором размещается файл печати (файл с расширением ".PRN"). Символ "x" является именем дискового устройства (вводится без двоеточия).
- [R] Указывает программе "XLT86" при возврате из подпрограмм рассматривать все флагги активными.
- [S] Указывает программе "XLT86" на то, что в микропроцессоре Intel 8086 коды и сегменты не перекрываются.
- [Tx] Используется для спецификации имени диска, на котором размещается временный файл, созданный программой "XLT86". В этом параметре "x" является именем диска (вводится без двоеточия).
- [80] Используется для вывода исходного текста на языке ассемблера для микропроцессора Intel 8080 в файл печати (файл с расширением ".PRN").
- [86] Используется для вывода в файл печати (файл с расширением ".PRN") текста на языке ассемблера для микропроцессора Intel 8086, полученного в результате преобразования исходного файла.

ПРИЛОЖЕНИЕ Б

ТРАНСЛЯТОРЫ С ПРИНЯТЫХ В ОС СР/М ВЕРСИЙ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ ВЫСОКОГО УРОВНЯ

ВВЕДЕНИЕ

В настоящем приложении описываются трансляторы с языков высокого уровня, которые обычно не входят в стандартный набор транзитных про-

грамм ОС *CP/M*, но могут быть включены в него дополнительно. В приложении описываются только те трансляторы, которые за последние несколько лет приобрели достаточную популярность у пользователей. Многие из дополнительных транзитных программ общего назначения, описанных в приложении А, используются совместно с определенными трансляторами, описанными в данном приложении (в дальнейшем при необходимости это отмечается).

Для каждого транслятора, описанного в данном приложении, представлены только его основные характеристики, содержащие информацию о том, как обратиться к каждому из трансляторов и как происходит его выполнение в среде ОС *CP/M*. Помимо того, здесь указывается, какие файлы необходимы для работы и какие файлы создаются при использовании транслятора. Иными словами, в описании приведена информация о взаимодействии различных трансляторов с ОС *CP/M*.

В приложении описаны следующие трансляторы с языков высокого уровня:

- "MAC". Макроассемблер, поставляемый фирмой Digital Research.
- "RMAC". Перемещаемый макроассемблер, поставляемый фирмой Digital Research.
- "MACRO-80". Макроассемблер, поставляемый фирмой Microsoft.
- "PL/I-80". Компилятор с принятого в ОС *CP/M* подмножества языка ПЛ/1, поставляемый фирмой Digital Research.
- "BASIC-80". Интерпретатор языка Бейсик, поставляемый фирмой Microsoft.
- "BASCOM". Компилятор с языка Бейсик, поставляемый фирмой Microsoft.
- "CBASIC". Компилятор с языка Бейсик, поставляемый фирмой Digital Research.
- "CB-80". Компилятор чистого кода с языка Бейсик, поставляемый фирмой Digital Research.
- "PASCAL/MT+". Компилятор с языка Паскаль, поставляемый фирмой Digital Research.
- "FORTRAN-80". Компилятор с языка Фортран, поставляемый фирмой Microsoft.
- "COBOL-80". Компилятор с языка Кобол, поставляемый фирмой Microsoft.

Оставшаяся часть приложения содержит краткое описание перечисленных трансляторов.

МАКРОАССЕМБЛЕР "МАС"

ОБЩИЕ СВЕДЕНИЯ

Макроассемблер "МАС", поставляемый фирмой Digital Research, является дополнительной транзитной программой, которая может быть добавлена в стандартный набор транзитных программ ОС *CP/M*. Макроассемблер "МАС" используется для ассемблирования (трансляции) программ, написанных на языке ассемблера для микропроцессора Intel 8080 и содержащих макросы. Макросы — это некоторая последовательность операторов языка ассемблера, которой присвоено определенное имя; обращение к такой последовательности и ее ассемблирование производится тогда, когда в исходной программе пользователя обнаружено соответствующее имя. В процессе ассемблирова-

ния исходного файла макроассемблер "MAC" создает файл с шестнадцатеричным кодом исходной программы, который впоследствии можно обрабатывать с помощью команды LOAD, файл печати, аналогичный создаваемому стандартным ассемблером ОС CP/M "ASM", и файл таблицы символов. Для получения информации о том, как использовать макросы при написании программ на языке ассемблера, следует обратиться к руководству по макроассемблеру "MAC". С помощью макроассемблера "MAC" нельзя создавать файлы с перемещаемым объектным кодом, это можно сделать только с помощью макроассемблера "RMAC".

ОБРАЩЕНИЕ К МАКРОАССЕМБЛЕРУ "MAC"

Для того чтобы обратиться к макроассемблеру "MAC", необходимо, чтобы файл "MAC.COM" хранился на диске, вставленном в одно из устройств системы. Для запуска макроассемблера "MAC" на выполнение необходимо в качестве ответа на промпт с именем текущего диска (A>, B> и т. д.) ввести команду MAC, а затем имя исходного файла. Указанный файл должен иметь расширение ".ASM", однако это расширение в командной строке не указывается.

В процессе ассемблирования исходного файла макроассемблер "MAC" создает три результирующих файла. Первым является файл, содержащий объектный код в шестнадцатеричном формате (файл с расширением ".HEX"). Вторым является файл печати (файл с расширением ".PRN"). И, наконец, третьим является файл таблицы символов (файл с расширением ".SYM"). Последний файл не создается стандартным ассемблером "ASM". Все три файла имеют то же имя, что и исходный файл (файл с расширением ".ASM"). Содержимое файла с объектным кодом (файла с расширением ("HEX") представляется в стандартном шестнадцатеричном формате фирмы Intel. С помощью команды LOAD этот файл может быть далее преобразован в файл с исполняемым машинным кодом. Формат файла печати (файла с расширением ".PRN") такой же, как формат файла печати, создаваемого стандартным ассемблером "ASM". Разница заключается в том, что он содержит расширения всех обнаруженных макросов. Отметим, что в исходном файле присутствуют только символические обозначения макросов — их имена (т. е. в исходном тексте вместо макроса стоит что-либо подобное имени "EOUT"). Файл таблицы символов (файл с расширением ".SYM") содержит информацию о всех символических именах, обнаруженных в исходной программе, и может быть впоследствии использован в процессе отладки (см. описание дополнительных транзитных программ "SID" и "ZSID" в приложении А).

В командную строку команды MAC после имени исходного файла могут быть добавлены специальные параметры, которые управляют работой макроассемблера "MAC". В одной командной строке может быть указано несколько параметров, каждый параметр должен отделяться от других пробелом. Первому параметру должен предшествовать символ "O". Параметры команды MAC определены следующим образом:

Aх Определяет имя диска, на котором расположен исходный файл (файл с расширением ".ASM"). Здесь "x" представляет собой имя диска (вводится без двоеточия).

Hx	Определяет имя диска, на котором должен быть записан результатирующий файл с шестнадцатеричным (объектным) кодом (файл с расширением ".HEX"). Здесь "x" представляет собой имя диска (вводится без двоеточия). Вместо имени диска может быть указан один из трех символов: "X" (вывод содержимого файла на консоль), "P" (вывод на устройство печати) и "Z" (подавить вывод файла). Эти символы можно указывать в любом параметре, который обычно определяет имя диска, за исключением параметра "A", описанного выше.
+Lx или -Lx	Определяет имя диска, на котором расположены библиотечные файлы, содержащие определения используемых в исходном файле макросов. Здесь "x" представляет собой имя диска или специальный символ. Параметр "+L" указывает на то, что исходные строки, прочитанные из библиотечного файла, необходимо распечатывать на консоли. Параметр "-L" установлен по умолчанию и используется для подавления такого рода печати.
+M или -M	Управляет распечаткой макросов в выходном файле печати (файле с расширением ".PRN"). Если указан параметр "+M", то все макростроки по мере их обработки выводятся в файл печати. Если указан параметр "-M", то макросы в файл печати не выводятся.
Rx	Используется для определения имени диска, на который должен быть записан файл печати (файл с расширением ".PRN"). Здесь "x" представляет собой имя диска или специальный символ.
+Q или -Q	Управляет выводом имен типа LOCAL в файл таблицы имен (файл с расширением ".SYM"). Если указан параметр "+Q", то все имена типа LOCAL выдаются в файл таблицы символов. Параметр "-Q" используется для подавления вывода в файл таблицы символов имен типа LOCAL.
+S или -S	Используется для указания, необходимо ли добавить в файл печати таблицу символов (содержимое файла с расширением ".SYM"). Если указан параметр "+S", то таблица символов добавляется в файл печати. Если указан параметр "-S", то в файл печати ничего не добавляется.
+1 или -1	Используется в процессе ассемблирования для задания вывода в файл печати результатов первого прохода ассемблирования. Если указан параметр "+1", то в файл печати выдаются результаты первого прохода ассемблирования, которые впоследствии могут быть использованы для отладки макросов. Параметр "-1" установлен по умолчанию и используется для подавления вывода в файл печати результатов первого прохода ассемблирования.

МАКРОАССЕМБЛЕР "RMAC"

ОБЩИЕ СВЕДЕНИЯ

Макроассемблер "RMAC" фирмы Digital Research представляет собой дополнительную транзитную программу, которая может быть добавлена в стан-

дартный набор программ ОС *CP/M* и которая используется для создания файлов с перемещаемым объектным кодом. В основном макроассемблер "RMAC" идентичен макроассемблеру "MAC". Отличие состоит в том, что макроассемблер "RMAC" вместо файлов с загружаемым шестнадцатеричным (объектным) кодом создает файлы с перемещаемым объектным кодом. Файл с перемещаемым объектным кодом может быть объединен с другими файлами, содержащими перемещаемый объектный код, с помощью дополнительной программы "LINK-80" (см. приложение А). Результаты сборки файлов с перемещаемым объектным кодом представляются в формате исполняемого машинного кода.

ОБРАЩЕНИЕ К МАКРОАССЕМБЛЕРУ "RMAC"

Обращение к макроассемблеру "RMAC" и процесс трансляции исходного файла протекают так же, как для описанного выше макроассемблера "MAC", за исключением того, что формат параметров команды RMAC несколько изменен. В командной строке команды RMAC вместо параметра "H", который используется командой MAC, необходимо использовать параметр "R". Данный параметр используется для управления выводом результирующего файла с перемещаемым объектным кодом, имеющего расширение ".REL", вместо расширения ".HEX", которое присваивается результирующему файлу макроассемблером "MAC". Файл с перемещаемым объектным кодом может быть далее обработан программой "LINK-80" с тем, чтобы объединить его с остальными файлами, содержащими перемещаемые объектные коды других модулей. Результаты сборки модулей могут быть помещены в файл, содержащий исполняемый код окончательно сформированной программы.

Ниже перечисляются дополнительные параметры, предназначенные для использования в командной строке команды RMAC. В командной строке может быть указано произвольное число параметров, при этом первому параметру должен предшествовать символ "Q".

QA Определяет имя диска, на котором расположен исходный файл (файл с расширением ".ASM"). Параметр "A" вводится в формате "QA" <имя диска>, например "QAA" (исходный файл расположен на диске А), "QAB" (исходный файл расположен на диске В) и т. д.

QR Определяет имя диска, на котором должен быть расположен результирующий файл с перемещаемым объектным кодом (файл с расширением ".REL"). Параметр "R" вводится в том же формате, что и параметр "A", например "QRA", если результирующий файл должен быть расположен на диске А, "QRB", если результирующий файл должен быть расположен на диске В, и т. д.

QL Определяет имя диска, на котором расположены библиотечные файлы, содержащие определения используемых в исходном файле макросов. Параметр "L" вводится в том же формате, что и параметр "A", т. е. "QLA", если библиотечные файлы расположены на диске А, "QLB", если библиотечные файлы расположены на диске В, и т. д.

Q+L и Q-L Управляет распечаткой исходных строк, прочитанных из ма-

кробиблиотеки. Если в командной строке параметр "L" задан в формате " $\text{Q}+L$ ", то в процессе ассемблирования исходные строки, прочитанные из библиотечного файла, будут распечатываться на консоли, если параметр "L" задан в формате " $\text{Q}-L$ ", то исходные строки, прочитанные из библиотечного файла, на консоли печататься не будут.

$\text{Q}+\text{M}$ и $\text{Q}-\text{M}$ Управляет распечаткой макрострок по мере их обработки в процессе ассемблирования. Подобная печать будет выполняться, если параметр "M" задан в формате " $\text{Q}+\text{M}$ ". Если параметр "M" задан в формате " $\text{Q}-\text{M}$ ", то печать макрострок выполниться не будет.

$\text{Q}*\text{M}$ Указывает на то, что в процессе макрорасширения необходимо порождать только шестнадцатеричный код.

QR Определяет имя диска, на котором должен быть расположен файл печати (файл с расширением ".PRN"). Этот параметр вводится в том же формате, что и параметр "A", например " QPA ", если файл печати должен быть расположен на диске A, " QPB ", если файл печати должен быть расположен на диске B, и т. д.

$\text{Q}+\text{Q}$ и $\text{Q}-\text{Q}$ Управляет выводом имен типа LOCAL в файл таблицы символов. Если параметр "Q" задан в формате " $\text{Q}+\text{Q}$ ", то в файл таблицы символов будут выданы все обнаруженные имена типа LOCAL. Для подавления вывода имен типа LOCAL в файл таблицы символов параметр "Q" необходимо использовать в формате " $\text{Q}-\text{Q}$ ".

$\text{Q}+\text{S}$ и $\text{Q}-\text{S}$ Определяет, необходимо ли добавить в файл печати таблицу символов (содержимое файла с расширением ".SYM"). Если параметр "S" указан в формате " $\text{Q}-\text{S}$ ", то таблица символов будет добавлена в файл печати. Если параметр "S" задан в формате " $\text{Q}+\text{S}$ ", то в файл печати ничего не добавляется.

$\text{Q}+\text{1}$ и $\text{Q}-\text{1}$ Управляет выводом в файл печати результатов первого прохода ассемблирования. Результаты первого прохода ассемблирования могут быть использованы в процессе отладки макросов. Если в командной строке указан параметр " $\text{Q}+\text{1}$ ", то результаты первого прохода ассемблирования выдаются в файл печати. Если указан параметр " $\text{Q}-\text{1}$ ", который установлен по умолчанию, то результаты первого прохода в файл печати не выдаются.

МАКРОАССЕМБЛЕР "MACRO-80"

ОБЩИЕ СВЕДЕНИЯ

Макроассемблер "MACRO-80" фирмы Microsoft является дополнительной транзитной программой, которая может быть добавлена в стандартный набор транзитных программ ОС CP/M. Назначение и процесс выполнения данного макроассемблера в основном совпадают с макроассемблерами "MAC" и "RMAC". Макроассемблер "MACRO-80" используется для трансляции (ассемблирования) программ, написанных на языке ассемблера и содержащих макросы. Результатом ассемблирования является файл, содержащий перенемещаемый объектный код исходной программы. Сборка этих файлов с перенемещаемым объектным кодом может выполняться при помощи программы

"LINK-80" фирмы Microsoft (см. приложение А). В результате этого будет файл, содержащий исполняемый машинный код окончательно сформированной программы.

Помимо этого макроассемблер "MACRO-80" создает специальный файл, который позже можно обработать с помощью дополнительной программы "CREF-80" фирмы Microsoft (см. приложение А). Этот файл используется для создания отчета о перекрестных ссылках в исходной программе и по своему содержанию аналогичен файлу таблицы символов (файл с расширением ".SYM"), создаваемому макроассемблерами "MAC" и "RMAC".

ОБРАЩЕНИЕ К МАКРОАССЕМБЛЕРУ "MACRO-80"

Для того чтобы обратиться к макроассемблеру "MACRO-80", необходимо, чтобы файл "M80.COM" хранился на диске, вставленном в одно из устройств системы. Для вызова макроассемблера "MACRO-80" на исполнение необходимо в качестве ответа на промпт с именем текущего диска (A >, B > и т. д.) ввести команду M80. В командной строке после команды M80 могут быть указаны имя исходного файла и специальные параметры. С другой стороны, если сразу после команды M80 будет нажата клавиша "RETURN", имена файлов и параметры могут быть заданы в режиме диалога. При этом программа "MACRO-80" будет загружена в оперативную память и выдан промпт "*". В качестве ответа на этот промпт могут быть введены имена исходных файлов и специальные параметры по одной строке в один раз.

Если имена файлов и специальные параметры вводятся в той же командной строке, что и команда M80, то перед именем исходного файла должен быть указан знак равенства ("="). Если в командной строке необходимо указать имя файла с перемещаемым объектным кодом и имя файла печати, то эти имена должны быть указаны перед символом "=" в следующем порядке: имя файла с перемещаемым объектным кодом, запятая, имя файла печати. Кроме того данная командная строка или какие-либо другие ее варианты могут быть введены в качестве ответа на промпт "*".

В командной строке команды M80 могут быть указаны различные специальные параметры (ключи). Эти параметры, если они используются, необходимо вводить в конце командной строки. Каждому параметру должен предшествовать символ "/". Параметры команды M80 определяются следующим образом:

- /O Указывает на то, что все адреса в файле печати (файл с расширением ".LST") должны быть представлены в восьмеричном формате (в виде восьмеричных чисел).
- /H Указывает на то, что в файле печати (файл с расширением ".LST") все адреса должны быть представлены в шестнадцатеричном формате (в виде шестнадцатеричных чисел).
- /R Указывает на то, что результирующий файл с перемещаемым объектным кодом должен иметь то же имя, что и исходный файл. Этот параметр можно использовать вместо спецификации в командной строке имени файла с перемещаемым объектным кодом.
- /L Указывает на то, что файл печати (файл с расширением ".LST") должен иметь то же имя, что и исходный файл. Этот параметр можно использовать вместо спецификации в командной строке имени файла печати.

- /C Предписывает макроассемблеру "MACRO-80" создать специальный файл с информацией о перекрестных ссылках (который позже может быть обработан с помощью программы "CREF-80"). Этот файл имеет расширение ".CRF".
- /Z Указывает на то, что исходный файл содержит программу на языке ассемблера для микропроцессоров Zilog Z80. В этом случае асsembлирование исходного файла макроассемблером "MACRO-80" выполняется правильно. Этот параметр необходимо указывать только тогда, когда исходный файл не содержит псевдокоманду ".Z80".
- /I Указывает на то, что исходный файл содержит программу на языке ассемблера для микропроцессора Intel 8080. В этом случае асsembлирование исходного файла выполняется макроассемблером "MACRO-80" правильно. Этот параметр необходимо указывать только тогда, когда исходный файл не содержит псевдокоманду ".8080".
- /P Обеспечивает увеличение на 256 байт размера области стека, которая используется в процессе асsembлирования. Обычно необходимость в данном параметре не возникает, но его следует использовать тогда, когда в процессе асsembлирования исходного файла возникает переполнение стека.
- /M Обеспечивает инициализацию областей с данными. Параметр необходимо использовать только в том случае, если исходный файл содержит псевдокоманду DS (Резервирование области в оперативной памяти), инициированную нулями.
- /X Обеспечивает распечатку секций условного асsembлирования (т. е. секций, в которых выражение, заданное в операторе условного асsembлирования, имеет значение "ложь"). Если данный параметр не используется, то ни одна секция условного асsembлирования не будет выведена.

ТРАНСЛЯТОР "PL/I-80" (КОМПИЛЯТОР С ЯЗЫКА ПЛ/1 ДЛЯ МИКРОПРОЦЕССОРА Intel 8080)

ОБЩИЕ СВЕДЕНИЯ

Транслятор "PL/I-80" (Транслятор с языка ПЛ/1 для микропроцессора Intel 8080) фирмы Digital Research является дополнительной транзитной программой, которая может быть добавлена в стандартный набор транзитных программ ОС CP/M. Этот транслятор базируется на так называемом подмножестве G (Подмножество общего назначения) языка программирования ПЛ/1, разработанном Комитетом по стандартизации языка программирования ПЛ/1 (Х3J1) Американского национального института стандартов. Входной язык для транслятора "PL/I-80" почти совпадает с подмножеством G языка ПЛ/1 за некоторыми различиями, которые описаны в руководстве по использованию транслятора "PL/I-80". Кроме того, этот язык напоминает язык Паскаль и язык ПЛ/М-80 фирмы Intel.

Язык программирования ПЛ/1 часто называют "язык для разработки систем", поскольку многие системные программисты рассматривают его как удобное средство для разработки операционных систем, трансляторов с языков высокого уровня и системных утилит. Транслятор "PL/I-80" используется для обработки (компиляции) файлов, содержащих программы на языке "ПЛ/1" в совместимом с транслятором "PL/I-80" формате. Подобная обработка

ка выполняется с целью создания файлов с перемещаемым объектным кодом. Файлы с перемещаемым объектным кодом могут впоследствии объединяться с другими библиотечными файлами при помощи программы "LINK-80" фирмы Digital Research (см. приложение А). В результате этого создается файл, содержащий исполняемый машинный код окончательно сформированной программы.

ОБРАЩЕНИЕ К ТРАНСЛЯТОРУ "PL/I-80"

Для того чтобы обратиться к транслятору "PL/I-80" необходимо, чтобы файлы "PLI.COM" (основной файл) и "PLILIB.IRL" (библиотека времени исполнения транслятора "PL/I-80") хранились на диске, вставленном в одно из устройств системы. Для запуска транслятора "PL/I-80" на исполнение необходимо в качестве ответа на промпт с именем текущего диска (A >, B > и т. д.) ввести команду PLI и имя файла, содержащего исходный текст транслируемой программы. Если для исходного файла не указано расширение (типа), то транслятор "PL/I-80" по умолчанию предполагает расширение ".PLI".

В процессе компиляции исходной программы транслятор "PL/I-80" автоматически обращается к библиотечному файлу "PLILIB.IRL" для получения необходимой информации. Результирующим является файл, содержащий перемещаемый объектный код и имеющий расширение ".REL". Этот файл может быть далее объединен с помощью программы "LINK-80" (см. приложение А) с файлами, содержащими перемещаемый объектный код других модулей, в результате чего будет создан файл с исполняемым машинным кодом окончательно сформированной транзитной программы.

ИНТЕРПРЕТАТОР "BASIC-80" (ИНТЕРПРЕТАТОР С ЯЗЫКА БЕЙСИК ДЛЯ МИКРОПРОЦЕССОРА Intel 8080)

ОБЩИЕ СВЕДЕНИЯ

Интерпретатор "BASIC-80" фирмы Microsoft является интерпретатором с языка Бейсик и может быть добавлен в стандартный набор транзитных программ ОС CP/M. Интерпретатор "BASIC-80" версии 5 удовлетворяет требованиям стандартного подмножества языка Бейсик, рекомендованного Американским национальным институтом стандартов. Помимо этого интерпретатор "BASIC-80" предоставляет пользователю многие другие уникальные средства и может быть использован как для создания файлов с исходными текстами программ на языке Бейсик, так и для выполнения в режиме интерпретации существующих программ.

ОБРАЩЕНИЕ К ИНТЕРПРЕТАТОРУ "BASIC-80"

Для того чтобы обратиться к интерпретатору "BASIC-80", необходимо, чтобы файл "MBASIC.COM" хранился на диске, вставленном в одно из устройств системы. Хотя имя "MBASIC.COM" является наиболее распространенным для этого файла, он может храниться и под другим именем.

Команда, обеспечивающая запуск интерпретатора "BASIC-80" на исполнение, может быть введена двумя способами. В первом случае в качестве ответа на промпт с именем текущего диска (A >, B > и т. д.) вводится команда

MBASIC и после нее нажимается клавиша "RETURN". Это приводит к тому, что интерпретатор "BASIC-80" будет загружен в оперативную память, после чего выдается промпт "Ок". Этот промпт означает, что интерпретатор "BASIC-80" готов к обработке команд. С помощью этих команд пользователь может создать новый файл с программой на языке Бейсик, внести изменения в существующий файл или исполнить (в режиме интерпретации) какую-либо программу. Кроме этого поле команды MBASIC можно указать имя файла, содержащего программу на языке Бейсик, которую необходимо выполнить. Если имя исходного файла указано без расширения, то интерпретатор "BASIC-80" автоматически предполагает расширение ".BAS". Существуют два типа файлов, которые содержат исходные тексты программ на языке Бейсик и которые могут использоваться совместно с интерпретатором "BASIC-80". Файлы первого типа имеют расширение ".BAS" и обычно содержат программы на языке Бейсик, которые предназначены для выполнения в режиме интерпретации. Обычно этот файл имеет специфический формат: он содержит много управляемых символов, которые затрудняют просмотр файла с помощью команды TYPE или аналогичной ей. Файлы другого типа содержат исходные тексты программ на языке Бейсик в формате кода ASCII. Файлы этого типа обычно имеют расширение ".ASC". Именно эти файлы чаще всего используются в тех случаях, когда изменения в программу вносятся не с помощью интерпретатора "BASIC-80", а с помощью какого-либо текстового редактора.

В командной строке команды MBASIC помимо имени исходного файла можно указать несколько дополнительных параметров, которые определяют некоторые характеристики интерпретатора "BASIC-80". Эти параметры вводятся с предшествующим символом "/", и после каждого из них обычно указывается некоторое число.

Параметры команды MBASIC определены следующим образом:

/F: nnnn Определяет максимальное число файлов с данными, которые могут быть открыты одновременно в процессе выполнения исходной программы. Здесь "nnnn" представляет собой количество файлов в десятичном формате. Если данный параметр не указан, то по умолчанию принимается значение 3.

/M: nnnn Определяет максимальный объем оперативной памяти, который может быть использован интерпретатором "BASIC-80"; здесь число "nnnn" представляет собой объем оперативной памяти. Данный параметр позволяет зарезервировать область оперативной памяти перед интерпретатором "BASIC-80", которая может быть использована для хранения программ, написанных на языке ассемблера для микропроцессора Intel 8080.

/S: nnnn Указывает максимальный размер записей для файлов прямого доступа. Здесь "nnnn" – десятичное число, представляющее собой количество байтов в одной записи. Если данный параметр не указан, то по умолчанию максимальный размер записи устанавливается равным 128 байт.

ТРАНСЛЯТОР "BASCOM" (КОМПИЛЯТОР С ЯЗЫКА БЕЙСИК ФИРМЫ Microsoft)

ОБЩИЕ СВЕДЕНИЯ

Транслятор с языка Бейсик фирмы Microsoft, который обычно называется "BASCOM", представляет собой набор дополнительных транзитных программ, который может быть включен в стандартный набор транзитных программ ОС CP/M и может использоваться совместно с интерпретатором "BASIC-80" фирмы Microsoft. Набор программ "BASCOM" предоставляет средство, при помощи которого файлы с исходными текстами программ на языке Бейсик, представленные в формате интерпретатора "BASIC-80", могут быть преобразованы (оттранслированы) в файлы, содержащие перемещаемый объектный код. Эти файлы с перемещаемым объектным кодом при помощи программы "LINK-80" фирмы Microsoft (см. приложение А) могут быть далее объединены с другими файлами в процессе окончательной сборки и создания файла с исполняемым машинным кодом. В набор программ BASCOM входят также различные файлы, содержащие подпрограммы, которые используются в различных версиях компилятора с языка Бейсик и в исходных программах.

В целом набор программ "BASCOM" предоставляет средство, при помощи которого файлы с исходными текстами программ на языке Бейсик могут быть преобразованы в файлы с исполняемым машинным кодом. Скорость выполнения полностью откомпилированной и скомпонованной программы на языке Бейсик существенно выше скорости ее выполнения в режиме интерпретации с помощью интерпретатора "BASIC-80".

ОБРАЩЕНИЕ К ТРАНСЛЯТОРУ "BASCOM"

В набор программ "BASCOM" входит файл "BASCOM.COM", который содержит собственно компилятор с языка Бейсик (представленный в виде транзитной программы), а также различные библиотечные файлы, содержащие все необходимые в процессе компиляции подпрограммы. Для того чтобы обеспечить полную обработку файлов с перемещаемым объектным кодом, созданных компилятором, в данный набор программ входят также программы "LINK-80", "MACRO-80", "CREF-80" и "LIB-80".

Для выполнения компиляции файла, содержащего исходный текст в формате интерпретатора "BASIC-80", необходимо в качестве ответа на промпт с именем текущего диска (A>, B> и т. д.) ввести команду BASCOM и имя этого файла. Если имя исходного файла указано без расширения, то компилятор "BASCOM" по умолчанию предполагает расширение ".BAS". По завершении процесса компиляции транслятор "BASCOM" создает файл, содержащий перемещаемый объектный код исходной программы. Этот файл имеет расширение ".REL". Кроме того, транслятор "BASCOM" создаст файл печати (файл с расширением ".PRN"), который, когда это необходимо, содержит исходный текст программы и параметры компиляции.

Если результирующий файл с перемещаемым объектным кодом (файл с расширением ".REL") должен иметь имя, отличное от имени исходного файла, то это имя необходимо указать в командной строке первым. Имя результирующего файла должно отделяться от имени исходного файла при помощи знака равенства ("="). Если файл печати (файл с расширением ".PRN") дол-

жен иметь имя, отличное от имени исходного файла, то это имя должно быть указано перед именем исходного файла. Имя файла печати должно отделяться от имени файла с перемещаемым объектным кодом запятой. Наконец, для изменения только имени файла печати перед новым именем должна стоять запятая, а после него – знак равенства ("=") и имя исходного файла.

После создания результирующих файлов они могут быть обработаны при помощи других программ фирмы Microsoft (например, программы "LINK-80" для создания файла с исполняемым машинным кодом).

В командной строке команды BASCOM указываются следующие параметры (ключи):

- /C Указывает на то, что должны быть ослаблены ограничения на нумерацию строк. Отметим, что данный параметр не может использоваться совместно с параметром "/4".
- /D Предписывает компилятору сформировать средства отладки, которые будут использоваться для обнаружения ошибок в процессе исполнения программы.
- /E Указывает на то, что в исходной программе содержатся операторы "ON ERROR GOTO" с командой "RESUME <номер строки>".
- /N Запрещает вывод результатов реассемблирования объектного кода в файл печати.
- /O Обеспечивает замену библиотеки времени исполнения "OBSLIB.REL" на "BASLIB.REL", которая будет использоваться в качестве основной библиотеки времени исполнения программой "LINK-80" фирмы Microsoft в тех случаях, когда ей задан параметр "/E" или "/G".
- /S Указывает на необходимость записывать заключенные в кавычки строки в результирующий файл с объектным кодом (файл с расширением ".REL"), а не в область данных в оперативной памяти.
- /T Указывает на необходимость использовать соглашение об исполнении программ на языке Бейсик, поддерживаемое интерпретатором "BASIC-80" версии 4.51 фирмы Microsoft. Отметим, что данный параметр не может использоваться совместно с параметром "/C".
- /X Указывает на то, что в исходной программе содержатся операторы "ON ERROR GOTO" с командой "RESUME", "RESUME O" или "RESUME NEXT".
- /Z Указывает на то, что в процессе компиляции необходимо по возможности использовать коды операций микропроцессора Zilog Z80.
- /4 Указывает на необходимость использовать соглашения о синтаксисе программ на языке Бейсик, поддерживаемые интерпретатором "BASIC-80" версии 4.51 фирмы Microsoft. Отметим, что данный параметр не может использоваться совместно с параметром "/C".

ТРАНСЛЯТОР "CBASIC" (КОМПИЛЯТОР С ЯЗЫКА БЕЙСИК)

ОБЩИЕ СВЕДЕНИЯ

Транслятор "CBASIC" представляет собой набор дополнительных транзитных программ, который может быть включен в стандартный набор программ ОС CP/M, и содержит собственно компилятор с языка Бейсик, а также интер-

претатор промежуточного кода, который обеспечивает выполнение оттранслированных программ. Компилятор "CBASIC" предоставляет средства, при помощи которых файлы, содержащие исходные тексты программ на языке Бейсик, можно преобразовать (оттранслировать) в файлы, содержащие промежуточный код. Программы на языке Бейсик, представленные в промежуточном коде, можно далее запустить на выполнение при помощи специальной программы (интерпретатора), которая также входит в набор программ транслятора "CBASIC".

Существуют две основные версии данного транслятора: версия 1, известная под именем "CBASIC", и версия 2, известная под именем "CBASIC2". Обе эти версии в основном одинаковы, хотя несколько отличаются друг от друга способом использования команд и операторов в исходных программах.

ОБРАЩЕНИЕ К ТРАНСЛЯТОРУ "CBASIC"

Для того чтобы обратиться к транслятору "CBASIC" или "CBASIC 2", необходимо, чтобы на диске, вставленном в одно из устройств системы, хранились следующие два файла: "CBAS.COM" и "CRUN.COM" или "CBAS 2.COM" и "CRUN 2.COM" в случае использования компилятора "CBASIC"; файл "CRUN.COM" ("CRUN 2.COM") содержит интерпретатор промежуточного кода, который используется для исполнения оттранслированных программ.

Для транслирования программы на языке Бейсик необходимо в качестве ответа на промпт с именем текущего диска (A >, B > и т. д.) ввести команду CBAS или CBAS 2. После данной команды указывается имя исходного файла без расширения, поскольку компилятор "CBASIC" по умолчанию предполагает, что исходный файл имеет расширение ".BAS". По завершении процесса компиляции исходной программы компилятор "CBASIC" осуществляет вывод на консоль сообщения обо всех обнаруженных ошибках. Кроме того, создается результирующий файл с промежуточным кодом исходной программы (файл с расширением ".INT"), содержащий результаты компиляции. Для того чтобы запустить на выполнение программу, представленную в промежуточном коде, необходимо ввести команду CRUN или CRUN 2, за которой следует имя файла с промежуточным кодом этой программы; имя файла указывается без расширения, поскольку интерпретатор промежуточного кода по умолчанию предполагает расширение ".INT".

При использовании транслятора "CBASIC" в командной строке команды CBAS или CBAS 2 можно вводить несколько дополнительных параметров. Эти параметры определяют некоторые характеристики транслятора "CBASIC". Все эти параметры должны быть указаны после имени исходного файла, и им должен предшествовать пробел и символ "Q". После символа "Q" может быть указано произвольное количество параметров.

Эти параметры определены следующим образом:

- QB Используется для подавления вывода исходной программы на консоль. Если в процессе компиляции в исходной программе обнаружены ошибки, то сообщения об этих ошибках выдаются на консоль независимо от того, указан данный параметр или нет.
- QC Используется для подавления вывода результатов компиляции в файл с промежуточным кодом (файл с расширением ".INT"). Данный параметр используется тогда, когда компиляция программы выполняется только с целью устранения синтаксических ошибок.

- D Используется для отмены преобразований символов нижнего регистра в символы верхнего регистра в процессе вывода исходного файла на консоль. Если в командной строке указан данный параметр, то все ключевые слова в исходном файле должны быть представлены символами верхнего регистра.
- E Использование этого параметра приводит к тому, что отметки обо всех обнаруженных в процессе компиляции ошибках помещаются в результатирующий файл с промежуточным кодом (файл с расширением ".INT"). В результате в процессе исполнения программы в соответствующих местах выдаются сообщения об этих ошибках. Данный параметр используется для отладки.
- F Использование этого параметра приводит к тому, что отчет о компиляции исходного файла выдается не на консоль, а на устройство печати.
- G Использование этого параметра приводит к тому, что отчет о компиляции исходного файла будет записываться на диске в файле с расширением ".LST".

ТРАНСЛЯТОР "СВ-80"

**(КОМПИЛЯТОР ЧИСТОГО КОДА С ЯЗЫКА БЕЙСИК ДЛЯ МИКРОПРОЦЕССОРА
Intel 8080)**

ОБЩИЕ СВЕДЕНИЯ

Транслятор "СВ-80" фирмы Digital Research представляет собой набор различных транзитных программ. Эти программы используются для преобразования файлов с программами, написанными в формате компилятора "CBASIC" ("CBASIC 2"), в файлы с перемещаемым объектным кодом, которые впоследствии могут быть преобразованы в файлы с исполняемым машинным кодом. В набор программ входит компилятор с языка Бейсик, а также программа, которая обеспечивает обслуживание библиотек и сборку модулей в перемещаемом объектном коде и которая называется "LK-80".

ОБРАЩЕНИЕ К ТРАНСЛЯТОРУ "СВ-80"

Для того чтобы использовать транслятор "СВ-80", необходимо, чтобы на диске, вставленном в одно из устройств системы, хранились следующие файлы: "CB80.COM", который является собственно компилятором с языка Бейсик, и "LK80.COM", который содержит программу обслуживания библиотек и сборки модулей.

Для того чтобы оттранслировать файл с исходным текстом, представленным в формате компилятора "CBASIC" или "CBASIC 2", необходимо в качестве ответа на промпт с именем текущего диска (A >, B > и т. д.) ввести команду CB80, за которой следует имя исходного файла. Если имя исходного файла указано без расширения, то по умолчанию предполагается расширение ".BAS". По завершении процесса компиляции создается файл, содержащий перемещаемый объектный код исходной программы и имеющий расширение ".REL". Этот файл может быть далее преобразован в файл с исполняемым машинным кодом при помощи программы "LK-80".

В командную строку команды CB80 можно добавлять различные параметры, которые определяют некоторые характеристики транслятора "СВ-80".

Эти параметры вводятся после имени исходного файла и заключаются в квадратные скобки ("[" и "]"). В одной командной строке может быть указано произвольное количество параметров, причем все они могут быть введены последовательно и заключены в одну пару квадратных скобок.

Эти параметры команды СВ80 определяются следующим образом:

- [B] Используется для подавления вывода содержимого исходного файла на консоль.
- [C(x)] Указывает имя диска, который должен использоваться вместо текущего диска в процессе поиска файлов, указанных в операторе "%INCLUDE" исходной программы. Здесь (x) представляет собой имя диска (например, "(A)", если необходимо использовать диск А, "(B)", если необходимо использовать диск В, и т. д.).
- [D] Обеспечивает вывод сообщений об ошибках, связанных с неописанными в исходной программе переменными.
- [I] Используется для вывода на консоль формированного кода вместе с исходным кодом программы.
- [L(n)] Используется для изменения размера страницы при выводе отчета о компиляции на печатающее устройство при помощи параметра "[P]". Здесь "n" – десятичное число, представляющее собой размер страницы.
- [N] Использование параметра приводит к тому, что объектный код формируется для каждой физической строки в исходном тексте программы.
- [O] Используется для подавления вывода результатов компиляции в файл с перемещаемым объектным кодом (файл с расширением ".REL").
- [P] Указывает на то, что исходный файл необходимо вывести на печатающее устройство. Содержимое исходного файла разбивается на страницы, и каждая страница имеет заголовок, содержащий номер страницы и имя исходного файла.
- [S] Использование параметра приводит к тому, что в результирующий файл с перемещаемым объектным кодом (файл с расширением ".REL") включается информация о переменных, используемых в программе.
- [T] Использование параметра приводит к тому, что после содержимого исходного файла выдается таблица символов.
- [W(n)] Определяет ширину страницы при выводе исходного текста программы на печатающее устройство при помощи параметра "[P]". Здесь "n" – десятичное число, которое представляет собой ширину страницы (например, "(65)", если страница должна иметь 65 символов в ширину, "(75)" – если 75, и т. д.).
- [X(x)] Определяет имя диска, который должен использоваться вместо текущего диска, для хранения временных рабочих файлов, необходимых в процессе компиляции. Здесь "x" представляет собой имя диска (например, "(A)", если необходимо использовать диск "А", "(B)", если необходимо использовать диск "В", и т. д.).

ТРАНСЛЯТОР "PASCAL/MT+" (КОМПИЛЯТОР С ЯЗЫКА ПАСКАЛЬ)

ОБЩИЕ СВЕДЕНИЯ

Транслятор "PASCAL/MT+" представляет собой набор дополнительных транзитных программ, который может быть включен в стандартный набор транзитных программ ОС CP/M. Транслятор "PASCAL/MT+" полностью обеспечивает компиляцию стандартной версии языка Паскаль (DPS/7185), разработанной Международной организацией по стандартам (ISO), а также имеет много других уникальных возможностей. В набор программ входят компилятор с языка Паскаль, программа-сборщик, реассемблер, отладчик и программа обслуживания библиотек. С помощью этих программ может быть выполнено преобразование файлов, содержащих исходный текст на языке Паскаль, который представлен в формате транслятора "PASCAL/MT+", в файлы с перемещаемым объектным кодом, которые впоследствии могут быть преобразованы в файлы с исполняемым машинным кодом.

ОБРАЩЕНИЕ К ТРАНСЛЯТОРУ "PASCAL/MT+"

Для использования транслятора "PASCAL/MT+" необходимо, чтобы на диске, вставленном в одно из устройств системы, хранились следующие файлы: "MTPLUS.COM" (компилятор с языка Паскаль), "LINKMT.COM" (программа-сборщик), "DIS8080.COM" (реассемблер), "LIBMT.COM" (программа обслуживания библиотек). Кроме того, существует много других файлов, необходимых для работы перечисленных выше программ (для того чтобы определить, какие из этих файлов являются обязательными, а какие могут отсутствовать, см. руководство по транслятору "PASCAL/MT+").

Для того чтобы запустить на выполнение компилятор с языка Паскаль, необходимо в качестве ответа на промпт с именем текущего диска (A >, B > и т. д.) ввести команду MTPLUS, за которой следует имя исходного файла. Если имя исходного файла указано без расширения, то по умолчанию предполагается расширение ".SRC". Компилятор "MTPLUS" выводит отчет о компиляции, а также список ошибок, обнаруженных в исходной программе. По завершении процесса компиляции формируется результирующий файл, содержащий перемещаемый объектный код исходной программы и имеющий расширение ".REL". Этот файл может быть затем преобразован в файл с исполняемым машинным кодом при помощи команды LINKMT.

В командной строке команды MTPLUS после имени исходного файла могут быть указаны дополнительные параметры (ключи), которые определяют некоторые характеристики компилятора "MTPLUS". Каждому такому параметру должен предшествовать символ "<Q>" или "#". В одной командной строке может быть указано произвольное количество параметров.

Эти параметры определяются следующим образом:

Q Rd или #Rd Определяет имя диска, на котором должен быть расположен результирующий файл с перемещаемым объектным кодом (файл с расширением ".REL"). Здесь "d" представляет собой имя диска (вводится без двоеточия).

Q Od или #Od Определяет имя диска, на котором расположены основной и оверлейные файлы. Здесь "d" представляет собой имя диска (вводится без двоеточия).

Q Pd или #Pd	Определяет имя диска, на котором должен быть расположен файл печати (файл с расширением ".PRN"). Здесь "d" представляет собой имя диска (вводится без двоеточия).
Q X или #X	Используется для создания расширенного файла с перемещаемым объектным кодом (файла с расширением ".REL"), содержащего записи для реассемблера.
Q D или #D	Предписывает включить в объектный код информацию, необходимую для программы отладки. Данный параметр обеспечивает также запись выходного файла с расширением ".PSY" на диск, имя которого задано при помощи параметра "P".
Q Ed или #Ed	Определяет имя диска, на котором расположен файл "MTERRS.TXT", который содержит сообщения об ошибках. Здесь "d" представляет собой имя диска (вводится без двоеточия).
Q Td или #Td	Определяет имя диска, на который должен быть записан файл "PASTEMP.TOK". Здесь "d" представляет собой имя диска (вводится без двоеточия).
Q Q или #Q	Используется для подавления вывода на консоль всех сообщений, которые не являются обязательными.
Q C или #C	Использование параметра приводит к тому, что при обнаружении любых ошибок выполнение программы* будет продолжено.
Q A или #A	Использование параметра приводит к тому, что по завершении процесса компиляции управление будет автоматически передано программе сборки.
Q B или #B	Указывает на необходимость использовать двоично-кодированное представление десятичных чисел (BCD) вместо представления с плавающей точкой.
Q Z или #Z	Указывает на необходимость генерировать оптимизированный код для микропроцессора Zilog Z80. Если данный параметр не используется, то компилятор генерирует объектный код для микропроцессора Intel 8080.
Q V или #V	Используется для вывода на консоль имен всех процедур и функций по мере их обнаружения в исходном тексте. Он используется как вспомогательное средство для обнаружения ошибок в течении первой фазы компиляции.
Q @ или #@	Указывает на то, что символ "@" необходимо рассматривать в качестве эквивалента символу "^".

ТРАНСЛЯТОР "FORTRAN-80" (КОМПИЛЯТОР С ЯЗЫКА ФОРТРАН ДЛЯ МИКРОПРОЦЕССОРА Intel 8080)

ОБЩИЕ СВЕДЕНИЯ

Транслятор "FORTRAN-80" фирмы Microsoft является дополнительной транзитной программой, которая может быть добавлена в стандартный набор

* Здесь имеется в виду выполнение программы-компилятора. – Прим. перев.

транзитных программ ОС *CP/M*. Транслятор "FORTRAN-80" обеспечивает преобразование (компиляцию) файлов с исходными текстами на языке Фортран в файлы с перемещаемым объектным кодом для дальнейшего преобразования в файлы с исполняемым машинным кодом. Транслятор "FORTRAN-80" обеспечивает трансляцию стандартной версии языка Фортран, которая описана в документе X3.9-1966 (Американский национальный стандарт на язык программирования Фортран), выпущенном Американским национальным институтом стандартов. Кроме того, этот документ содержит ряд расширений и ограничений на входной язык.

ОБРАЩЕНИЕ К ТРАНСЛЯТОРУ "FORTRAN-80"

Для использования транслятора "FORTRAN-80" необходимо, чтобы файл "F80.COM" хранился на диске, вставленном в одно из устройств системы. Для того чтобы запустить на выполнение транслятор "FORTRAN-80", необходимо в качестве ответа на промпт с именем текущего диска (A >, B > и т. д.) ввести команду F80 и имя исходного файла. Если имя файла указано без расширения, то по умолчанию предполагается расширение ".FOR". Помимо этого команда F80 может быть введена без параметров, что приводит к тому, что транслятор "FORTRAN-80" загружается в оперативную память, а на консоли появляется промпт, состоящий из символа "*". Этот промпт означает, что транслятор "FORTRAN-80" ожидает ввода команды с консоли.

По завершении процесса компиляции создается результирующий файл, содержащий перемещаемый объектный код исходной программы и имеющий расширение ".REL". Этот файл может быть затем преобразован в файл с исполняемым машинным кодом при помощи программы "LINK-80" (см. приложение А). Транслятор "FORTRAN-80" создает также файл печати (файл с расширением ".PRN"), который содержит отчет о компиляции. Синтаксис командной строки команды F80 является таким же, как в случае использования программы "MACRO-80" фирмы Microsoft.

В командную строку команды F80 могут быть включены дополнительные параметры (ключи), которые определяют некоторые характеристики транслятора "FORTRAN-80". Эти параметры вводятся после имени исходного файла. Каждому параметру должен предшествовать символ "/".

Параметры команды F80 определены следующим образом:

- /O Указывает на то, что в листинге компиляции все адреса должны быть представлены в восьмеричном коде (по основанию 8).
- /H Указывает на то, что в листинге компиляции все адреса должны быть представлены в шестнадцатеричном коде (по основанию 16).
- /N Запрещает вывод сформированного кода на консоль.
- /R Обеспечивает создание файла с перемещаемым объектным кодом (файла с расширением ".REL").
- /L Обеспечивает создание файла печати (файла с расширением ".PRN").
- /P Обеспечивает выделение дополнительных 100 байт под стек, который используется в процессе компиляции. Данный параметр может быть указан в командной строке несколько раз, что приводит к многократному увеличению размера стека на 100 байт.
- /M Указывает на то, что сформированный код должен быть представлен в формате, пригодном для загрузки в ПЗУ (постоянное запоминающее устройство).

ТРАНСЛЯТОР "COBOL-80" (КОМПИЛЯТОР С ЯЗЫКА КОБОЛ ДЛЯ МИКРОПРОЦЕССОРА Intel 8080)

ОБЩИЕ СВЕДЕНИЯ

Транслятор "COBOL-80", поставляемый фирмой Microsoft, является набором транзитных программ, которые могут быть добавлены в стандартный набор транзитных программ ОС CP/M. Сюда входят компилятор с языка Кобол и интерпретатор промежуточного кода, который обеспечивает исполнение откомпилированных программ.

ОБРАЩЕНИЕ К ТРАНСЛЯТОРУ "COBOL-80"

Для использования компилятора с языка Кобол и интерпретатора промежуточного кода необходимо, чтобы на диске, вставленном в одно из устройств системы, хранились следующие файлы: "COBOL.COM" (компилятор), "COBOL1.OVR", ..., "COBOL4.OVR" (оверлейные файлы), "RUNCOB.COM" (интерпретатор промежуточного кода), "COBLBX.REL" (библиотека времени исполнения для интерпретатора), "CRTDRV.REL" (базовый драйвер видеотерминала), "DEBUG.REL" (интерактивный отладчик программ на языке Кобол).

Для выполнения компиляции файла с исходным текстом на языке Кобол необходимо в качестве ответа на промпт с именем текущего диска (A>, B> и т. д.) ввести команду COBOL, за которой следует имя этого файла. Если имя исходного файла указано без расширения, то по умолчанию предполагается расширение ".COB". По завершении процесса компиляции создается файл с перемещаемым объектным кодом (файл с расширением ".REL") и файл печати (файл с расширением ".PRN"), содержащий отчет о компиляции. Синтаксис командной строки в данном случае такой же, как при использовании программ "MACRO-80" и "FORTRAN-80" фирмы Microsoft.

В командную строку команды COBOL могут быть включены дополнительные параметры (ключи), которые определяют некоторые характеристики компилятора "COBOL-80". Эти параметры должны быть указаны после имени исходного файла. Каждому параметру должен предшествовать символ "/".

Параметры команды COBOL определены следующим образом:

- /R Обеспечивает генерацию файла с перемещаемым объектным кодом (файла с расширением ".REL").
- /L Обеспечивает создание файла печати (файла с расширением ".PRN").
- /P Обеспечивает выделение дополнительных 100 байт под стек, используемый в процессе компиляции. Данный параметр может быть указан в командной строке несколько раз, что приведет к многократному увеличению размера стека на 100 байт.
- /D Используется для отмены создания отладочного файла, который содержит в объектном коде информацию о нумерации строк в исходном файле.
- /X Обеспечивает преобразование файла с объектным кодом таким образом, что он может быть использован только совместно с интерпретатором, имеющимся в библиотеке транслятора "COBOL-80".
- /Fn При использовании этого параметра в файл печати (файл с расширением ".PRN") выдаются предупреждающие сообщения об использова-

ния возможности FIPS*. Здесь "n" может принимать следующие значения:

- 0 Печать сообщения об использовании всех возможностей FIPS и расширений фирмы Microsoft.
- 1 Исключить печать сообщений об использовании средств нижнего промежуточного уровня.
- 2 Исключить печать сообщений об использовании средств как нижнего, так и верхнего промежуточного уровней.
- 3 Исключить печать сообщений об использовании всех расширений.
- 4 Отменить печать предупреждающих сообщений об использовании возможностей FIPS.

ПРИЛОЖЕНИЕ В

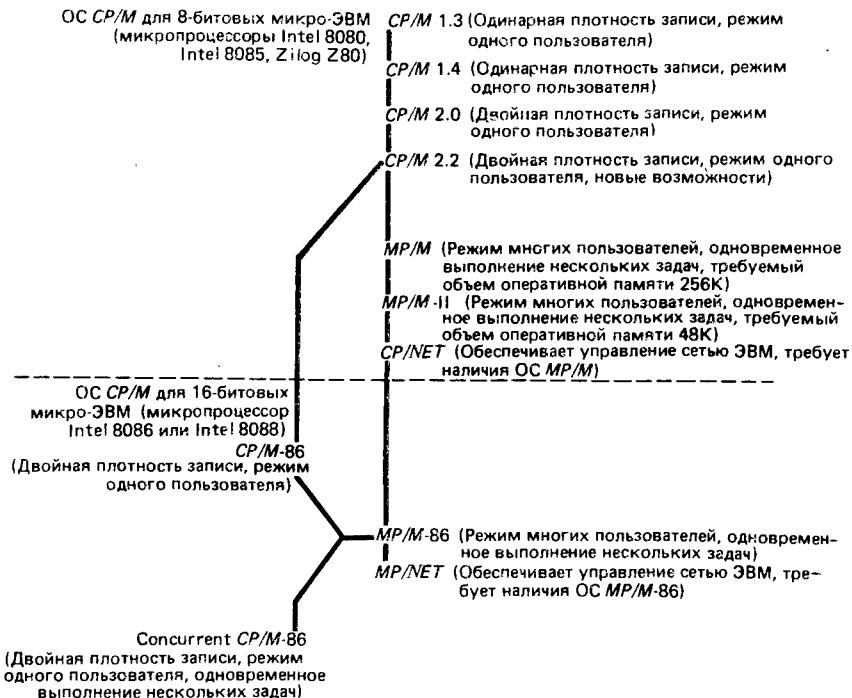
ОПЕРАЦИОННЫЕ СИСТЕМЫ, ПОСТРОЕННЫЕ НА БАЗЕ ОС CP/M

ВВЕДЕНИЕ

В настоящем приложении даны общие сведения о сходствах и различиях между операционной системой *CP/M* и операционными системами, созданными на ее основе. Конкретно в этом приложении содержится информация об операционных системах *MP/M-II*, *CP/M-86* и *MP/M-86*, а также об операционной системе *CP/NET*, которая обеспечивает управление сетью ЭВМ, оснащенных операционными системами *CP/M* и *MP/M-II*. С тех пор, как ОС *CP/M* приобрела большую популярность у разработчиков и пользователей микро-ЭВМ, создано много операционных систем, которые являются либо непосредственными модификациями ОС *CP/M*, либо очень похожи на нее. Фактически сейчас существует так много разновидностей ОС *CP/M* для различных ЭВМ, что включить в приложение описание всех новейших версий не представляется возможным. Именно по этой причине в приложении содержатся сведения только об операционных системах серии *CP/M* и *MP/M* фирмы Digital Research.

В данном приложении описаны основные различия между семействами операционных систем *CP/M* и *MP/M*, в частности, между имеющимися в них наборами команд, хотя специфика использования этих команд не рассматривается. Назначение этого приложения – дать новичку-пользователю этих операционных систем общее представление о каждой системе. Ниже представлена схема, иллюстрирующая основной путь развития операционной системы *CP/M*. В нее включены операционные системы только фирмы Digital Research, поскольку операционных систем других фирм слишком много. Для каждой операционной системы на данной диаграмме указаны ее основные характеристики, такие как поддержка режима одного или многих пользователей, работа с дисками с одинарной или двойной плотностью записи и т. д.

* Поскольку в данной книге сокращение "FIPS" нигде не расшифровано, за дополнительными разъяснениями читателю следует обратиться к руководству по транслятору "COBOL-80". – Прим. перев.



ОПЕРАЦИОННАЯ СИСТЕМА MP/M-II

Название *MP/M-II* является акронимом от английского Multi-Programming Monitor Control Program for Microcomputers*. Эта многопользовательская, многозадачная операционная система для микро-ЭВМ, созданных на базе микропроцессоров Intel 8080, Intel 8085 и Zilog Z80. Операционная система *MP/M-II* является второй версией ОС *MP/P* со значительно меньшей требуемой оперативной памятью и с новыми и расширенными возможностями. ОС *CP/M-II* является, однако, совместимой с ОС *MP/M* версии 1 по принципу "снизу вверх". Поскольку ОС *MP/M-II* весьма успешно заменила ОС *MP/M* версии 1, в этом разделе обсуждается только ОС *MP/M-II*. ОС *MP/M-II* имеет те же требования к оборудованию, что и ОС *CP/M* с некоторыми отличиями, а именно:

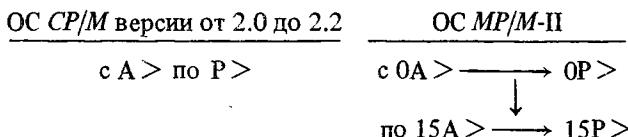
- Наличие как минимум 48К оперативной памяти.
- Наличие прерываний от таймера.

Помимо этих требований ОС *MP/M-II* может иметь такую же конфигурацию оборудования, что ОС *CP/M*, хотя в ОС *MP/M-II* имеются средства расширения возможностей оборудования до 16 печатающих устройств, 16 устройств ввода-вывода, из которых 8 могут быть консольными устройствами (видеотерминалами или телетайпами). В ОС *MP/M-II* используется тот же

* Буквально: Многопрограммная мониторная управляющая программа для микро-ЭВМ. – Прим. перев.

стандарт фирмы IBM на 8-дюймовые гибкие диски с одинарной плотностью записи, что в ОС *CP/M*, и таким образом можно легко осуществлять передачу файлов между системами. Кроме того, транзитные программы ОС *CP/M* могут использоваться и в ОС *MP/M-II*, хотя некоторые из них требуют модификации, поскольку использование оперативной памяти в ОС *MP/M-II* происходит иначе.

Работа ОС *MP/M-II* с точки зрения пользователя очень похожа на работу ОС *CP/M* за исключением формата некоторых промптов и многих команд, имеющих другие имена. Операционная система *MP/M-II* полностью использует возможности команды USER в том, что может поддерживать до 16 пользователей одновременно. Для предоставления информации о пользователе, который работает с системой, основной промпт ОС *MP/M-II* помимо имени текущего диска включает в себя текущий код пользователя:



Все встроенные и транзитные команды ОС *CP/M* используются в ОС *MP/M-II*, хотя во многих случаях функции этих команд расширены. Помимо этого существует много команд, свойственных только ОС *MP/M-II*. В основном синтаксис командной строки в ОС *MP/M-II* такой же, как в ОС *CP/M*, хотя имеется много отклонений, в том числе дополнительные функции, которые могут сделать ввод команд существенно более сложным, чем в ОС *CP/M*.

ОПЕРАЦИОННАЯ СИСТЕМА *CP/M-86*

Операционная система *CP/M-86* разработана на базе ОС *CP/M* и предназначена для функционирования на микро-ЭВМ, созданных на основе микропроцессоров Intel 8086, Intel 8088. Операционная система *CP/M-86* в основном аналогична ОС *CP/M* постольку, поскольку она является операционной системой одного пользователя, поддерживает в основном те же команды и фактически тем же способом организует взаимодействие с пользователем. Однако ОС *CP/M-86* предназначена для работы с микропроцессорами другого типа и поэтому обладает существенно большими по сравнению с ОС *CP/M* возможностями по взаимодействию с аппаратурой. В ОС *CP/M-86* сохранены все встроенные и транзитные команды ОС *CP/M*, хотя некоторые из этих команд имеют другое название, а возможности многих команд расширены. Например, команда ассемблирования в ОС *CP/M-86* называется не ASM, а ASM86, поскольку она используется только для ассемблирования программ, написанных на языке ассемблера для микропроцессора Intel 8086 или Intel 8088, и, следовательно, не является совместимой с ОС *CP/M*. Существует ряд других команд специального применения, которые называются иначе, чем соответствующие команды в ОС *CP/M*.

ОПЕРАЦИОННАЯ СИСТЕМА Concurrent *CP/M-86*

Операционная система Concurrent *CP/M-86* создана на базе ОС *CP/M-86* и включает многие характерные возможности ОС *CP/M* версии 3.0. Эта опе-

рационная система отличается от ОС *CP/M*-86 тем, что она поддерживает параллельный режим обработки, т. е. позволяет пользователю выполнять несколько программ одновременно. Это достигается за счет использования так называемой "виртуальной консоли". Виртуальная консоль создается самой операционной системой после того, как пользователь запустил программу (задачу) с пульта ЭВМ. С виртуальной консолью обращаются как с реальным физическим устройством, однако создание его и управление им полностью осуществляются операционной системой. Если запустить еще одну программу в то время, когда другая программа уже выполняется, операционная система создаст вторую виртуальную консоль по этому запросу. Одновременно может выполняться не более трех различных программ.

Кроме того, операционная система Concurrent *CP/M*-86 предоставляет средства обработки данных в реальном времени, что делает ее очень удобным инструментальным средством для разработки систем, эмуляции систем и для решения других задач в реальном времени.

Операционная система Concurrent *CP/M*-86 поддерживает использование до мегабайта оперативной памяти и максимум шестнадцати накопителей на магнитных дисках емкостью до 512 магабайт каждый.

ОПЕРАЦИОННАЯ СИСТЕМА *MP/M*-86

Операционная система *MP/M*-86 является многопользовательской, много задачной операционной системой, предназначенней для функционирования на микро-ЭВМ, созданных на основе микропроцессора Intel 8086 или Intel 8088. В целом ОС *MP/M*-86 разработана как дополнение к операционной системе одного пользователя *CP/M*-86 и обладает в основном теми же характеристиками, что ОС *MP/M*-II. По сравнению с ОС *MP/M*-II ОС *MP/M*-86 имеет значительно большие возможности по взаимодействию с аппаратурой за счет сохранения многих особенностей ОС *CP/M*-86 наряду с возможностью обслуживать одновременно большее число пользователей. Операционная система *MP/M*-86 может поддерживать до 254 устройств ввода-вывода, которые могут использоваться почти в любой комбинации. В ОС *MP/M*-86 сохранены все команды ОС *MP/M*-II и ОС *CP/M*-86 с некоторыми дополнительными возможностями, включая несколько новых команд. Системные промпты и формат командной строки в ОС *MP/M*-86 такие же, как в ОС *MP/M*-II.

ОПЕРАЦИОННАЯ СИСТЕМА ДЛЯ УПРАВЛЕНИЯ СЕТЬЮ ЭВМ *CP/NET*

Операционная система *CP/NET* предоставляет средства, с помощью которых несколько микро-ЭВМ, оснащенных ОС *CP/M*, ОС *MP/M* (версии 1) или ОС *MP/M*-II, можно объединять в сеть с тем, чтобы иметь возможность совместного использования ресурсов каждой системы. К таким совместно используемым ресурсам относятся дисковые файлы, устройства печати, консольные терминалы и различные программы и базы данных. Сеть обычно состоит из "главной" вычислительной системы, оснащенной ОС *MP/M* или ОС *MP/M*-II, и "вспомогательных" вычислительных систем, оснащенных ОС *CP/M*. Главная вычислительная система функционирует в качестве базовой ЭВМ, управляющей разделяемыми ресурсами, к которым могут обращаться вспомогательные системы, включенные в сеть. С помощью ОС *CP/NET* можно создавать вычислительные сети различного типа, а несколько вспомогатель-

ных систем можно соединить с отдельной главной системой. ОС *CP/NET* позволяет передавать сообщения между главной и вспомогательными системами, а при организации электронной почты обеспечивает пересылку корреспонденции между главной системой и вспомогательными.

ПРИЛОЖЕНИЕ Г

СПИСОК ЛИТЕРАТУРЫ

КНИГИ ПО ОС *CP/M*

1. Fernandez, Judi, and Ashley, Ruth. *Using CP/M*. New York: John Wiley Sons, 1980.
2. Brigham, Bruce, ed. *CP/M Summary Guide*. Glastonbury, CN. The Rosetta Stone, 1980.
3. Cortesi, David E. *Inside CP/M: A Guide For Users And Programmers*. New York: Holt, Rinehart and Winston, 1982.
4. Hogan, Thom. *CP/M User Guide*. Berkeley, CA. Osborne/McGraw-Hill, 1981.
5. Murtha, Stephen, and Waite, Mitchell. *CP/M Primer*. Indianapolis, IN, Howard W. Sams & Co., Ins., 1980.
6. Waite, Mitchell, and Lafore, Robert. *Soul Of CP/M*. Indianapolis, IN. Howard W. Sams & Co., Ins., 1983.
7. Zaks, Ronday. *The CP/M Handbook With MP/M*. Berkeley, CA, Sybex, 1980.

ЖУРНАЛЬНЫЕ СТАТЬИ ПО ОС *CP/M*

1. "A Seminar for Independet CP/M Software Vendors." *DataGast*, No. 001, pp. 24 – 28.
2. Anderton, Craig. "In Search of the Processed Word." *Kilobaud Microcomputing*, May, 1981, pp. 34 – 39.
3. Ballinger, Charles. "HDOS or CP/M?" *Interface Age*, September, 1980 pp. 88 – 91.
4. Barbier, Ken. "CP/M For Single-Drive Systems." *Kilobaud Microcomputing*, September, 1980, pp. 94 – 98.
5. Barbier, Ken. "CP/M Is for Me." *Kilobaud Microcomputing*, August, 1980, pp. 176 – 78.
6. Barbier, Ken. "Memory Map." *Kilobaud Microcomputing*, March, 1980, pp. 160 – 163.
7. Barbier, Ken. "Start/Exit Routine for CP/M." *Kilobaud Microcomputing*, October, 1980, pp. 82 – 83.
8. Barbier, Ken. "Relocating the Dymage Debugging Tool." *Kilobaud Microcomputing*, November, 1980, pp. 181 – 182.
9. Barbier, Ken. "A Print Utility for CP/M." *Kilobaud Microcomputing*, February, 1981, pp. 150 – 54.
10. Barbier, Ken. "The Secret Life Of the 8085." *Kilobaud Microcomputing*, September, 1980, pp. 114 – 19.
11. Barker, Ken. "Help With OSI's CP/M." *Dr. Dobbs Journal*, May, 1980, pp. 36 – 37.
12. Barkstorm, Bruse. "What Time Does the Sun Set?" *BYTE*, July, 1981, pp. 94 – 114.
13. Barley, Kathryn S., and Driscoll, James R. "A Survey of Data-Base Management Systems for

- Microcomputers." *BYTE*, November, 1981, pp. 208-234.
14. Biese, Leo, and Iannuccillo, Emilio. "MASTHEAD: Why Not Title Your Printouts?" *Interface Age*, August, 1980, pp. 122-27.
 15. Blank, Marc S., and Galley, S.W. "How to Fit a Large Program Into a Small Machine (or How to Fit the Great Underground Empire on Your Desk-Top)." *Creative Computing*, July, 1980, pp. 80-87.
 16. Bove, Tony. "Approaching CP/M." *DataCast*, No. 001, p. 29.
 17. Bove, Tony. "Writing With a CP/M System." *DataCast*, No. 002, pp. 6-11.
 18. Bozinovic, Dragan. "Software Addressing Modes for the 8080." *BYTE*, March, 1981, pp. 236-240.
 19. Brockman, D. M. "Binary-to-BCD Converter Program for the 8080." *BYTE*, August, 1981, pp. 418-419
 20. Calkins, Clark A. "Image Processing With a Printer." *BYTE*, February, 1981, pp. 220-248.
 21. Carbrey, Bruce D. "An Integer Math Package for the 8080." *BYTE*, May, 1981, pp. 204-226.
 22. Cecil, Alex. "ACT: An 8080 Macroprocessor." *Dr. Dobbs Journal*, March, 1978, pp. 20-45.
 23. Charnock, Glen. "Put a Dungeon In Your Computer." *Creative Computing*, July, 1980, pp. 66-78.
 24. Christensen, Ward. "An 8080 Disassembler." *Dr. Dobbs Journal*, February, 1977, pp. 30-43.
 25. Collins, Rosann; Hines, Theodore; and Rowan, George. "Manipulating Pencil Files; Convert Them to BASIC." *Creative Computing*, August, 1979, pp. 98-99.
 26. Conboy, Terry A. "Data Communication Error Measurement." *Kilobaud Microcomputing*, May, 1980, pp. 52-56.
 27. Corbin, Harold. "An Introduction to Data Compression." *BYTE*, April, 1981, pp. 218-250.
 28. Cotton, Gene. "How to Solve Your Damaged Disk Dilemma." *Interface Age*, September, 1980, pp. 80-86, 130-31.
 29. Craig, John. "A New Kind of Pencil!" *Creative Computing*, February, 1979, pp. 30-33.
 30. Cotter, Bill. "Microsoft Adventure." *Creative Computing*, May, 1980, pp. 20-21.
 31. Dahmke, Mark. "Wordsmith." *BYTE*, May, 1981, pp. 254-258.
 32. Dennon, Jack. "PUSH! POP! RAM! WOW!" *Kilobaud Microcomputing*, May, 1980, p. 200.
 33. Didday, Rich. "Universal Data Entry System; In a Car Pooling Application." *Creative Computing*, May, 1980, pp. 102-110.
 34. Doan, Daniel. "Computer Simulation of a Solar-Energy System." *BYTE*, July, 1981, pp. 158-172.
 35. Epstein, Jake, and Terry, Chris. "Introduction to CP/M; The CP/M Connection." *S100 Microsystems*, September/October, 1980, pp. 10-32.

36. Epstein, Jake. "An Introduction to CP/M." *S-100 Microsystems*, January/February, 1980, pp. 6-10; March/April, 1980, pp. 28-33; May/June, 1980, pp. 12-17.
37. Eubanks, Gordon. "Notes on CP/M's BASIC-E." *Dr. Dobbs Journal*, October, 1977, p. 35.
38. Firebaugh, M.; Fossum, T.; Sorensen, P.; and, Stone, W. "Comparative Evaluations of Basic Systems." *Creative Computing*, December, 1980, pp. 70-80.
39. Fitzgerald, Jim. "Off-the-Shelf Word Processing System." *Kilobaud Microcomputing*, September, 1979, pp. 92-94.
40. Foster, Charlie. "Pascal With a Z80." *Interface Age*, November, 1980, pp. 60-62.
41. Foster, Charlie, and Meador, Richard. "8080 Dynatrace." *S-100 Microsystems*, July/August, 1980, pp. 22-31.
42. Frantz, James. "Turn-Key CP/M Systems." *Creative Computing*, December, 1979, pp. 104-07
43. Freed, Edwin E. "Secret Codes Revealed." *Kilobaud Microcomputing*, April, 1981, pp. 58-60
44. Friedman, David. "Las Vegas Super Slot: A CP/M Game Machine Program Using Flashwriter I Graphics." *Dr. Dobbs Journal*, November/December, 1980, pp. 10-22.
45. Fritzson, Richard. "Data on Disk: Implementing File Systems." *Kilobaud Microcomputing*, January, 1981, pp. 111-115.
46. Fritzson, Richard. "The New CP/M: Is It Worth It?" *Kilobaud Microcomputing*, July, 1980, p. 66.
47. Fritzson, Richard. "Write Your Own FORTH Interpreter." *Kilobaud Microcomputing*, February, 1981, pp. 76-92.
48. Fritzson, Richard. "Write Your Own Pseudo-FORTH Compiler." *Kilobaud Microcomputing*, March, 1981, pp. 44-64.
49. Gagne, Jim. "Vice Versa - Pencil to CP/M and Reverse." *Dr. Dobbs Journal*, March, 1979, pp. 26-29.
50. Guralnick, Steven. "The Verdict Is In On Spellbinder." *Kilobaud Microcomputing*, May, 1981, pp. 72-78
51. Hayes Microcomputer Products, Inc. "Modem Control." *Kilobaud Microcomputing*, January, 1981, pp. 192-198
52. Haanstra, Bruce. "Optional Printing with CP/M and Microsoft BASIC." *Interface Age*, November, 1980, pp. 84-86.
53. Hallen, Rod. "Back-Space Mod for CP/M and Microsoft BASIC." *Kilobaud Microcomputing*, August, 1980, pp. 48-49.
54. Hallen, Rod. "Battle of the Assemblers." *Creative Computing*, December, 1979, pp. 42-45.
55. Hallen, Rod. "Double Your Memory, Double Your Fun." *Kilobaud Microcomputing*, May, 1981, pp. 190-192.
56. Hallen, Rod. "Super Word Processors." *Kilobaud Microcomputing*, June, 1980, pp. 214-217.
57. Hallen, Rod. "Tarbell Disk BASIC." *Kilobaud Microcomputing*, May, 1980, pp. 168-170.

58. Hallen, Rod. "The Battle of The Word Processors." *Creative Computing*, November, 1979, pp. 48-53.
59. Hamilton, R.W. "WPDaisy Word Processing System." *Creative Computing*, May, 1979, pp. 36-41.
60. Harrell, Martin D. "Operation Codes of the 8080, 8085, and Z80 Processors." *BYTE*, March, 1980, pp. 194-207.
61. Hart, Glenn. "A New BASIC From Tarbell." *Creative Computing*, January, 1980, pp. 20-23.
62. Hart, Glenn. "Magic Wand Word Processor." *Creative Computing*, August, 1980, pp. 38-45.
63. Hasset, Greg. "How to Write an Adventure." *Creative Computing*, July, 1980, pp. 88-90.
64. Heintz, Carl. "Analyst: Another Data-base Manager." *Interface Age*, December, 1980, pp. 42-44.
65. Heintz, Carl. "Maxiledger." *Interface Age*, September, 1980, pp. 42-44.
66. Heintz, Carl. "A Peach of a General Ledger Program." *Interface Age*, October, 1980, pp. 46-48.
67. Heintz, Carl. "Pearl - A Novel Programming Gem." *Interface Age*, November, 1980, p. 47.
68. Heyman, Victor. "IDSWORD - The Comprehensive Processing System for Home and Business." *Creative Computing*, May, 1979, pp. 43-44.
69. Hicks, Bruce. "Lost & Forgotten Island." *Creative Computing*, March, 1980, pp. 124-131.
70. Hoffer, W. C. "Data and Time for the CP/M Operating System." *Interface Age*, August, 1978, pp. 152-156.
71. Hogan, Thom. "CP/M and You." *Kilobaud Microcomputing*, February, 1980, pp. 183-184.
72. Hogg, Douglass. "How Good is Microsoft's FORTRAN-80?" *Creative Computing*, January, 1979, pp. 62-67.
73. Hunt, Daniel S. "DATALINE." *BYTE*, March, 1981, pp. 216-222.
74. Jackson, David R. and Callahan, John M. "Energy Conservation with a Microcomputer." *BYTE*, July, 1981, pp. 178-208.
75. Johnson, Bob. "Business Software Review." *Interface Age*, August, 1979, pp. 38-39.
76. Kaapke, Richard. "DUKEDOM." *Creative Computing*, February, 1980, pp. 100-110.
77. Kendall, Wallace. "Prettyprinting With Microsoft BASIC." *Kilobaud Microcomputing*, May, 1979, p. 80.
78. Kern, Christopher. "The BDS C Compiler." *BYTE*, June, 1981, pp. 356-362.
79. Kern, Christopher. "Printf for the C Function Library." *BYTE*, May, 1981, pp. 430-434.
80. Kildall, Gary. "CP/M: A Family of 8- and 16-Bit Operating Systems." *BYTE*, June, 1981, pp. 216-232.
81. Kildall, Gary. "Simple Technique for Static Relocation." *Dr. Dobbs Journal*, February, 1978, pp. 10-13.

82. Klem, David C. "Solar Specs." *Kilobaud Microcomputing*, March, 1980, pp. 68-70.
83. Knecht, Ken. "CBASIC Review." *80 Microcomputing*, April, 1980, pp. 130-32.
84. Larson, James. "The Ins and Outs of CP/M." *BYTE*, June, 1981, pp. 268-300.
85. Lehman, John A. "A Reformatter for CP/M and IBM Floppy Disks." *BYTE*, April, 1981, pp. 94, 96.
86. Lemmons, Phil. "Five Spelling-Correction Programs for CP/M-Based Systems." *BYTE*, November, 1981, pp. 434-448.
87. Levitan, Steven P., and Bonar, Jeffrey G. "Three Microcomputer LISP's." *BYTE*, September, 1981, pp. 388-412.
88. Lindsay, John. "New Version of BASIC." *Kilobaud Microcomputing*, May, 1980, pp. 72-74.
89. Lindsay, Jon. "For CP/M: Automatic Program Execution on Start-up." *Kilobaud Microcomputing*, January, 1981, pp. 184-186.
90. Lubar, David. "Pearl." *Creative Computing*, December, 1980, p. 68.
91. Lutz, Dick. "Sharpening Your Pencil." *Creative Computing*, March, 1980, pp. 30-35.
92. Lutz, Dick. "WordStar vs. Electric Pencil." *Creative Computing*, February, 1980, pp. 17-22.
93. Magruder, Bob. "The New WPDaisy: Word-Processing Software." *onComputing*, Fall 1980, pp. 68-74.
94. Marino, Albert J. "Sorta Super Fast." *Kilobaud Microcomputing*, April, 1981, pp. 164-166.
95. Maurer, W. D. "8080 Assembly Language for the Totally Lost." *onComputing*, Spring 1980, pp. 81-85.
96. Maxey, Dr. E. Stanton. "Tracking the Planets." *Kilobaud Microcomputing*, March, 1981, pp. 130-34.
97. McClure, James. "CBASIC - A Review." *Creative Computing*, September, 1979, pp. 48-51.
98. McClure, James. "Microsoft vs. MicroFocus COBOL." *Creative Computing*, March, 1980, pp. 20-29.
99. McClure, James. "A Personal Finance System." *Kilobaud Microcomputing*, June, 1979, pp. 74-78; July, 1979, pp. 50-56; August, 1979, pp. 66-75.
100. McClure, James. "Shoplist: The Latest Kitchen Utensil." *Creative Computing*, May, 1980, pp. 74-80.
101. Miller, Alan. "An Interrupt-Driven Keyboard Buffer." *Interface Age*, October, 1980, pp. 106-107, 137-141.
102. Miller, Alan. "BASCOM: Microsoft's BASIC Compiler for the 8080/Z80." *Interface Age*, July, 1980, pp. 124-26.
103. Miller, Alan. "CBASIC: A Business-Orientated Language for CP/M." *Interface Age*, August, 1979, pp. 116-119.
104. Miller, Alan. "CP/M for the TRS-80 Model II: Lifeboat and FMG Corp. Versions." *Interface*

Age, November, 1980, pp. 94-98.

105. Miller, Alan. "CP/M Part 2 - A Macro Assembler & Other Goodies." *Interface Age*, December, 1978, pp. 130-135.
106. Miller, Alan. "Diagnostics Package for CP/M." *Interface Age*, October, 1980, p. 104.
107. Miller, Alan. "Structured Assembly-Language Programming for the 8080." *Interface Age*, November, 1979, pp. 153-155.
108. Miller, Alan. "ZSID Z80 Debugger for CP/M." *Interface Age*, August, 1980, pp. 88-90.
109. Miller, Alan. "The Electric Pencil for CP/M." *Interface Age*, August, 1978, pp. 148-149.
110. Miller, Alan. "Pascal for CP/M: Digital Marketing's PascalM." *Interface Age*, September, 1980, pp. 96-103.
111. Monagan, James. "Efficient Data Storage For Microsoft BASIC." *Kilobaud Microcomputing*, November, 1980, pp. 68-69.
112. Morgan, Chris. "The New 16-Bit Operating Systems, or, The Search for Benutzerfreundlichkeit." *BYTE*, June, 1981, Editorial by the Editor in Chief, pp. 6, 10.
113. North, Steve. "The CP/M Disk Operating System." *Creative Computing*, November/December, 1978, pp. 52-53.
114. North, Steve. "Creative Computing Reviews Five Software Packages - Tiny C, Microsoft BASIC 5.0, Research Machines Z80 Algol, Structural Analysis SP80 Macros, Digital Research CP/M 2.0 and MP/M." *Creative Computing*, March, 1980, pp. 40-44.
115. Parsons, Ronald. "UCSD Pascal to CP/M File Transfer Program." *Dr. Dobbs Journal*, August, 1979, pp. 12-16.
116. Parsons, Thomas W. "All About ASCII." *Kilobaud Microcomputing*, November, 1980, pp. 129-132.
117. Pelczarski, Mark. "Microsoft Softcard." *BYTE*, November, 1981, pp. 152-162.
118. Platt, Charles; Dowden, Tony; and, Head, Gene. "Word Processing Roundup: WordStar & WP6502, Spellguard, VEDIT." *Kilobaud Microcomputing*, May, 1981, p. 40, p. 46 and pp. 50-51 respectively.
119. Pournelle, Jerry. "BASIC, Computer Languages, and Computer Adventures." *BYTE*, December, 1980, pp. 222-238.
120. Pournelle, Jerry. "New Software, New Hardware, Computer Languages, and Games." *BYTE*, November, 1981, pp. 449-457.
121. Pournelle, Jerry. "Omkron TRS-80 Boards, NEWDOS+, and Sundry Other Matters." *BYTE*, July, 1980, pp. 198-208.
122. Pournelle, Jerry. "Operating Systems, Languages, Statistics, Pirates, and the Lone Wolf." *BYTE*, January, 1982, pp. 132-158.
123. Press, Larry. "A Review of Four Text-Formatting Programs." *onComputing*, Fall 1980, pp. 48-54.
124. Press, Larry. "Word Processors: A Look at Four Popular Programs." *onComputing*, Summer

- 1980, pp. 38-52.
125. Pugh, Tim. "Intelligent Terminal Implementation on S100 Bus." *Dr. Dobbs Journal*, June/July, 1978, pp. 4-16.
126. Raymer, Paul. "Weather Station." *Creative Computing*, December, 1980, pp. 142-143.
127. Rhodes, Cheryl. "The DIALOG Information Retrieval System." *DataCast*, No. 002, pp. 51-62.
128. Rolander, Thomas A., "Microcomputer Software Meshes with Local Nets." *Electronics*, January, 1982, pp. 96-99.
129. Rose, Lance E. "Position-Independent Code for the 8080." *Kilobaud Microcomputing*, June, 1980, pp. 202-203.
130. Roxburgh, Alastair. "Fast Fourier Comes Back." *BYTE*, May, 1981, pp. 458-461.
131. Sanger M.D., Joseph Jay. "The Electronic Librarian." *Kilobaud Microcomputing*, November, 1979, pp. 44-62.
132. Sclawy, Alan. "CP/M Encryption Prescription." *Kilobaud Microcomputing*, November, 1980, pp. 42-46.
133. Sellier, Jane. "Auto Scribe." *Creative Computing*, January, 1980, pp. 24-25.
134. Sjowall, Tor. "CP/M to UCSD Pascal File Conversion." *Dr. Dobbs Journal*, October, 1980, pp. 16-19.
135. Stewart, John. "CP/M Primer: a Most Sophisticated Operating System." *Kilobaud Microcomputing*, April, 1978, pp. 30-34.
136. Stok, Glenn. "Video HARDCOPY For CP/M." *Kilobaud Microcomputing*, October, 1980, pp. 168-173.
137. Stoutemyer, David R. "Symbolic Math Using BASIC." *BYTE*, October, 1980, pp. 232-246.
138. Suits, David B. "Base Arithmetic." *Creative Computing*, November, 1980, pp. 146-149.
139. Swayne, Patrick. "Disassembler For the Heath H8." *Kilobaud Microcomputing*, August, 1980, pp. 88-94.
140. Terry, Chris. "The CP/M Connection." *S-100 Microsystems*, July/August, 1980, pp. 32-35.
141. "Upgraded CP/M Floppy Disc Operating System." *Dr. Dobbs Journal*, November, 1976, p. 51.
142. Van Buer, Darrel. "A Table-Driven Assembler on CP/M." *Dr. Dobbs Journal*, February, 1980, pp. 18-25.
143. VanHorn, Eric. "SuperSort by MicroPro International." *Creative Computing*, July, 1979, pp. 34-37.
144. Warren, Jim. "First Word on a Floppy-Disc Operating System." *Dr. Dobbs*, April, 1976, p. 5.
145. Warren, Jim. "The Time for Floppys is Just About Now!" *Dr. Dobbs Journal*, August, 1976, p. 5.
146. Williams, Gregg., Editor. "The muSIMP/muMATH-79 Symbolic Math System." *BYTE*, November, 1980, pp. 324-338.

147. Williams, Greg., sr. ed. "Three Versions of APL." *BYTE*, April, 1981, pp. 188 – 208.
148. Willoughby, Steve. "Hardcopy Device Driver Programs for CP/M." *Dr. Dobbs Journal*, September, 1980, pp. 34 – 37.
149. "Word Processing Directory." *Kilobaud Microcomputing*, May, 1981, pp. 68 – 70.
150. Work, Jim. "Hippity-Hop Memory Test." *Kilobaud Microcomputing*, May, 1981, pp. 166 – 75.
151. Zarucki, Paul. "Indirect I/O Addressing on the 8080." *BYTE*, August, 1981, pp. 402 – 403.

КНИГИ ПО ПРОГРАММИРОВАНИЮ НА ЯЗЫКЕ АССЕМБЛЕРА

1. *8080/8085 Assembly Language Programming Manual*. Santa Clara, CA, Intel Corp., 1980.
2. Leventhal, Lance, *8080A/8085 Assembly Language Programming*. Berkeley, CA. Osborne/McGraw-Hill, 1979.
3. Leventhal, Lance, *Z80 Assembly Language Programming*. Berkeley, CA. Osborne/McGraw-Hill, 1979.
4. Myers, Glenford, *Reliable Software Through Composite Design*. Van Nostrand Reinhold, 1975.
5. Zaks, Rodnay. *Programming the Z80*. Berkeley, CA. Sybex, 1979.

ПРИЛОЖЕНИЕ Д ТАБЛИЦА СИМВОЛОВ КОДА ASCII

Американский стандартный код для обмена информацией (ASCII) является 7-битовым кодом и известен также под названием Стандартный код США для обмена информацией (USASCII). Код ASCII используется в качестве средства взаимодействования с ЭВМ. Этот код позволяет преобразовывать машинные данные, представленные в двоичном коде, в обычные буквенные и числовые знаки, которые можно распечатать или вывести на терминал, и, кроме того, позволяет интерпретировать сигналы, введенные со специальной вынесенной клавиатуры, и преобразовывать их в соответствующие машинные данные в двоичном коде.

Поскольку 8-битовые коды используются в ЭВМ чаще, чем 7-битовые, то код ASCII обычно используется как 8-битовый код ASCII-8, который и представлен в следующих ниже таблицах. В таблицах перечислены 128 ($=2^7$) возможных комбинаций для 7-битового кода ASCII. Первые 128 значений в кодах ASCII и ASCII-8 не отличаются друг от друга. Восьмой (самый левый) бит в коде ASCII используется как бит четности. Бит четности равен единице, если правые семь бит содержат нечетное число единиц, и равен нулю, если четное. Бит четности часто используется в ЭВМ для проверки правильности переданных данных. Если полученный бит четности неправильно представляет семибитовый код, к которому он приписан, возникнет ошибка, и пользователю выдается сообщение Parity Errror (Ошибка четности), информирующая, что данные переданы неправильно.

Символы в коде ASCII разделяются на три основные категории, а именно:

1. Управляющие символы. К этой категории относятся специальные непечатаемые символы, которые используются для управления ЭВМ, терминалами и другими устройствами. Все управляющие символы имеют двоичное представление, но лишь несколько имеют символьное представление, которое пользователь может истолковать. Управляющие символы кода ASCII разбиваются кроме того на три категории:

А. Управляющие символы связи. Эти символы используются для управления передачей и приемом данных. Они используются в качестве маркеров начала и конца блока данных и для управления параметрами передачи.

Б. Символы спецификации формата. Эти символы используются для управления способом форматирования данных как в процессе передачи, печати или вывода на терминал, так и в процессе записи на носитель. В эту группу входят такие, как символ возврата каретки, перевода строки, табуляции и т. д.

В. Символы разделения данных. Эти символы используются для управления способом разделения блоков данных. Они представляют собой разделители файлов, групп, записей и элементов, удобные для пользователя. Существуют другие специальные управляющие символы, которые не попадают ни под одну из перечисленных выше трех категорий. Эти символы обычно используются для управления работой терминалов (телетайпов, видеотерминалов и т. п.) или других устройств и не оказывают непосредственного влияния на работу самой ЭВМ.

2. Алфавитно-цифровые символы. Эта группа символов включает в себя все буквы английского алфавита и десятичные цифры от 0 до 9.

3. Специальные графические символы. Эта группа символов включает в себя грамматические символы (точка, запятая, двоеточие и т. д.), экономические символы (знак денежной единицы, процент и т. д.), а также научные символы и символы общего назначения (знак "меньше", квадратные скобки, тильда и т. д.).

DEC X ₁₀	HEX X ₁₆	OCT X ₈	ASCII	Клавиш на клавиатуре терминала*
0	00	00	NUL	CTRL/1
1	01	01	SOH	CTRL/A
2	02	02	STX	CTRL/B
3	03	03	ETX	CTRL/C
4	04	04	EOT	CTRL/D
5	05	05	ENQ	CTRL/E
6	06	06	ACK	CTRL/F
7	07	07	BEL	CTRL/G
8	08	10	BS	CTRL/H, BACKSPACE, LEFT-ARROW
9	09	11	HT	CTRL/I, TAB
10	0A	12	LF	CTRL/J, LINE FEED, DOWN-ARROW
11	0B	13	VT	CTRL/K, UP-ARROW
12	0C	14	FF	CTRL/L, RIGHT-ARROW
13	0D	15	CR	CTRL/M, RETURN
14	0E	16	SO	CTRL/N
15	0F	17	SI	CTRL/O
16	10	20	DLE	CTRL/P
17	11	21	DC1	CTRL/Q, XON, NO SCROLL (OFF)
18	12	22	DC2	CTRL/R, TAPE
19	13	23	DC3	CTRL/S, XOFF, NO SCROLL (ON)
20	14	24	DC4	CTRL/T, TAPE

DEC X ₁₀	HEX X ₁₆	OCT X ₈	ASCII	Клавиш на клавиатуре терминала*
21	15	25	NAK	CTRL/U
22	16	26	SYN	CTRL/V
23	17	27	ETB	CTRL/W
24	18	30	CAN	CTRL/X
25	19	31	EM	CTRL/Y
26	1A	32	SUB	CTRL/Z, CLEAR SPACE
27	1B	33	ESC	ESC, ESCAPE
28	1C	34	FS	CTRL/\ (Control/Обратная косая)
29	1D	35	GS	CTRL/^ (Control/Слабое ударение)
30	1E	36	RS	CTRL/- (Control/Знак равенства)
31	1F	37	US	CTRL/- (Control/Минус) NEW LINE
32	20	40	SP	Пробел
33	21	41	!	! (Восклицательный знак)
34	22	42	"	" (Кавычки)
35	23	43	#	# (Знак номера или диез)
36	24	44	\$	\$ (Знак денежной единицы)
37	25	45	%	% (Процент)
38	26	46	&	& (Амперсанд)
39	27	47	,	' (Апостроф)
40	28	50	(((Левая круглая скобка)
41	29	51)) (Правая круглая скобка)
42	2A	52	*	* (Звездочка)
43	2B	53	+	+ (Плюс)
44	2C	54	,	, (Запятая)
45	2D	55	-	- (Минус)
46	2E	56	.	. (Точка)
47	2F	57	/	/ (Косая)
48	30	60	0	0
49	31	61	1	1
50	32	62	2	2
51	33	63	3	3
52	34	64	4	4
53	35	65	5	5
54	36	66	6	6
55	37	67	7	7
56	38	70	8	8
57	39	71	9	9
58	3A	72	:	: (Двоеточие)
59	3B	73	;	; (Точка с запятой)
60	3C	74	<	< (Знак "меньше")
61	3D	75	=	= (Знак равенства)
62	3E	76	>	> (Знак "больше")
63	3F	77	?	? (Вопросительный знак)
64	40	100	@	@ (Коммерческое "a")

Продолжение табл.

DEC X_{10}	HEX X_{16}	OCT X_8	ASCII	Клавиш на клавиатуре терминала*
65	41	101	A	A
66	42	102	B	B
67	43	103	C	C
68	44	104	D	D
69	45	105	E	E
70	46	106	F	F
71	47	107	G	G
72	48	110	H	H
73	49	111	I	I
74	4A	112	J	J
75	4B	113	K	K
76	4C	114	L	L
77	4D	115	M	M
78	4E	116	N	N
79	4F	117	O	O
80	50	120	P	P
81	51	121	Q	Q
82	52	122	R	R
83	53	123	S	S
84	54	124	T	T
85	55	125	U	U
86	56	126	V	V
87	57	127	W	W
88	58	130	X	X
89	59	131	Y	Y
90	5A	132	Z	Z
91	5B	133	[(Правая квадратная скобка)
92	5C	134	\	(Обратная косая)
93	5D	135]	(Левая квадратная скобка)
94	5E	136	^	(Знак карата)
95	5F	137	-	(Знак подчеркивания)
96	60	140	'	(Слабое ударение)
97	61	141	a	a
98	62	142	b	b
99	63	143	c	c
100	64	144	d	d
101	65	145	e	e
102	66	146	f	f
103	67	147	g	g
104	68	150	h	h
105	69	151	i	i
106	6A	152	j	j
107	6B	153	k	k
108	6C	154	l	l

DEC X_{10}	HEX X_{16}	OCT X_8	ASCII	Клавиш на клавиатуре терминала*
109	6D	155	m	m
110	6E	156	n	n
111	6F	157	o	o
112	70	160	p	p
113	71	161	q	q
114	72	162	r	r
115	73	163	s	s
116	74	164	t	t
117	75	165	u	u
118	76	166	v	v
119	77	167	w	w
120	78	170	x	x
121	79	171	y	y
122	7A	172	z	z
123	7B	173	{	{ (Правая фигурная скобка)
124	7C	174		(Вертикальная черта)
125	7D	175	}	} (Левая фигурная скобка)
126	7E	176	~	~ (Знак тильды)
127	7F	177	DEL	DEL, DELETE, RUBOUT

* Ввод последовательностей, содержащих "CTRL/", осуществляется путем одновременного нажатия клавиши CTRL и указанного в таблице. Эти последовательности указаны так, как они обозначены для стандартных терминалов, таких как клавиатура Diablo 1640 и терминалы серии Televideo. На других терминалах эти последовательности могут быть определены иначе.

Обозначения:

DEC – Десятичное представление (по основанию 10).

HEX – Шестнадцатеричное представление (по основанию 16).

OCT – Восьмеричное представление (по основанию 8).

ASCII – Американский стандартный код для обмена информацией.

CCITT – Международный консультативный комитет по телефонной и телеграфной связи.

ОБОЗНАЧЕНИЯ УПРАВЛЯЮЩИХ СИМВОЛОВ КОДА ASCII

ACK (Подтверждение) – управляющий символ связи, который используется в качестве обычного ответа "yes" (да) на различные вопросы, но иногда, кроме того означает: "Устройство получило предыдущее сообщение и готово к обработке следующего".

BEL (Звуковой сигнал) – управляющий символ общего назначения, используется в тех случаях, когда устройство, на которое он был послан, должно издать звуковой сигнал.

BS (Возврат) – символ используется для перемещения курсора или головки печатающего механизма назад на один символ или позицию.

CAN (Удаление) – управляющий символ общего назначения, используемый для указания, что часть переданных ранее данных подлежит уничтожению. Объем данных определяется пользователем.

CR	(Возврат каретки) – используется для перемещения курсора или головки печатающего механизма назад в начало строки. На большинстве терминалов нажатие клавиши RETURN равноценно введению символов CR и LF (перевод строки).
DC1-DC4	(Управление устройствами) – управляющие символы общего назначения, используемые для управления терминалом или подобными устройствами. Стандартные функции этим символам не предписываются, кроме символа DC4, обычно означающего "конец работы". Международный комитет CCITT предлагает ряд возможных назначений этим символам; обычно предпочтительно использовать первые два символа для "включения" и последние два символа для "выключения", а символы DC2 и DC4 для обращения к главному устройству. В некоторых системах эти символы обозначены XON, TAPE, XOFF и <u>TAPE</u> соответственно. Здесь "X" означает "передающее устройство", а "TAPE and <u>TAPE</u> " означают "tape on" и "tape off". В некоторых терминалах эти обозначения имеются на клавиатуре.
DEL	(Удаление) – управляющий символ общего назначения, используемый для удаления символа. На некоторых терминалах соответствующая клавиша называется RUBOUT. Этот символ не является строго говоря, управляющим символом. Управляющий символ DEL в двоичном представлении имеет одни единицы (1111 1111 ₂), что объясняется историческими причинами: единственным способом удаления отперфорированной на бумажной ленте последовательности бит был способ повторной перфорации всех пробивок с тем, чтобы результирующая комбинация пробивок была эквивалентна нулевой. В коде ASCII символ DEL до сих пор рассматривается в качестве эквивалента нулевого символа, хотя во многих операционных системах символ DEL используется для удаления предыдущего символа.
DLE	(Управление передачей данных) – символ используется для управления линиями связи и устройствами передачи данных.
EM	(Конец носителя – Eng of medium) – это управляющий символ общего назначения, используемый для индикации конца бумажной ленты (или другого носителя) или для обозначения конца материала носителя.
ENQ	(Запрос) – символ управления системами связи, обычно используемый в качестве запроса идентификатора абонента и его состояния. В некоторых системах этот код называется WRU (от английского "Who are you?" – "Кто Вы?").
EOT	(Конец передачи данных) – управляющий символ связи, используемый в качестве маркера конца после передачи одного или более сообщений.
ESC	(Управляющая последовательность) – символ общего назначения обозначает начало управляющей последовательности, которая состоит из ряда кодов, которые в качестве такой группы имеют специальное назначение (обычно функцию управления). В некоторых терминалах этот символ называется ALT MODE (Альтернативный режим).
ETB	(Конец пересылки блока данных) – управляющий символ систем связи, используемый при разбиении длинного сообщения на блоки. Символ ETB обозначает границы блока. Обычно блоки не оказывают влияния на формат передаваемого сообщения.
ETX	(Конец текста) – управляющий символ связи, используемый для обозначения конца текста. См. описание символа SOH. Этот код первоначально назывался EOM – "Конец сообщения" и может быть обозначен таким образом на некоторых терминалах.
FF	(Перевод формата) – управляющий символ вызывает переход каретки, головки печатающего механизма или курсора в начало следующей страницы.
ES, GS, RS, US	(Разделитель файлов, групп, записей и элементов данных) – набор управляющих символов используется для разграничения порций данных. Для этих

символов не предусмотрено стандартное использование за исключением символа FS, который предполагается применять для выделения наибольших порций данных, а символ US – для выделения наименьших порций.

- HT** (Горизонтальная табуляция) – управляющий символ, используемый для перемещения каретки, головки печатающего механизма или курсора в следующую, заранее определенную, позицию той же строки. Выбор позиций для горизонтальной табуляции обычно предоставляется пользователю.
- LF** (Перевод строки) – это управляющий символ, используемый для перемещения каретки, головки печатающего механизма или курсора на одну строку вниз. Во многих системах символ LF комбинируется с символом LF (возврат каретки), образуя комбинированный символ NL (новая строка).
- NAK** (Отрицательный ответ) – управляющий символ связи, используемый в качестве отрицательного ответа на различные вопросы. Иногда этот символ определяется как: "Устройство получило последнее сообщение, но оно содержит ошибки и необходимо повторить передачу".
- NUL** (Нуль-символ) – управляющий символ общего назначения, используемый главным образом в качестве заполнителя пробелов. См. также описание символа SYN.
- SI** (Восстановление) – управляющий символ общего назначения, используемый после кода SO для возврата к кодам в стандартном представлении символов в коде ASCII.
- SO** (Расширение кода) – управляющий символ общего назначения, указывающий, что последующие последовательности бит не соответствуют стандарту ASCII, и это справедливо до получения символа SI.
- SOH** (Начало заголовка) – управляющий символ систем связи для обозначения начала заголовка, используемого в сообщениях наряду с текстом. В заголовках обычно указывается имя и местоположение адресата. Этот код первоначально назывался SOM (Начало сообщения).
- STX** (Начало текста) – управляющий символ систем связи, используемый в качестве маркера начала текста и конца заголовка (если он используется). Первоначально этот код назывался EOА (конец адреса).
- SUB** (Подстановка) – управляющий символ общего назначения, используемый для обозначения символа, подставляемого вместо ошибочно введенного.
- SYN** (Синхронизация) – управляющий символ связи, используемый в некоторых высокоскоростных системах связи с синхронизацией в точках передачи и приема сообщений. В периоды простого при отсутствии битовых последовательностей, позволяющих приемному устройству поддерживать связь с передающим устройством, приемное устройство может выйти из синхронного режима. Поэтому каждому сообщению, передаваемому после периода простого, предшествует последовательность из трех или четырех символов SYN. Символ SYN имеет битовое представление, которое позволяет принимающему устройству синхронизироваться по передающему устройству, а кроме того определять начало и конец каждого символа. Помимо этого символы SYN можно использовать при небольших по продолжительности периодах простого с тем, чтобы поддерживать синхронизацию, отсюда и их название.
- VT** (Вертикальная табуляция) – это управляющий символ, используемый для перемещения каретки, головки печатающего механизма или курсора на следующую заранее определенную позицию (обычно на некоторую строку).

ПРИЛОЖЕНИЕ Е

ПРОМПТЫ ОС CP/M

X > ОС CP/M ожидает команду; диск с именем X является текущим.

NX > ОС MP/M ожидает команду; текущим является диск с именем X, а текущим пользователем – пользователь с кодом N.

- * Программа "PIP" или "ED" ожидает команду. Этот промпт используется, кроме того, программами "BASIC-80", "EDIT", "FORTRAN-80", "COBOL-80" и "PASCAL" фирмы Microsoft.
- nnnn: Программа "ED" ожидает команду; здесь "nnnn" представляет собой номер текущей строки.
- Программа "DDT" ожидает команду.

ПРИЛОЖЕНИЕ Ж

СООБЩЕНИЯ ОБ ОШИБКАХ ОС CP/M

В процессе использования ОС CP/M и ее различных команд редко случается, что пользователь не получает ни одного сообщения об ошибке. Время от времени возникают какие-нибудь ошибки. Однако в случае ОС CP/M сообщения об ошибках часто оказываются весьма загадочными и большей частью слабо или вовсе не снабжены пояснительными документами. Помимо этого обнаружилось, что многие сообщения являются неадекватными для многих видов ошибок (аппаратных или программных). По этим причинам принято решение привести список всех возможных сообщений об ошибках, возникающих в процессе использования ОС CP/M.

BDOS ERR ON X: BAD SECTOR

Это сообщение появляется в том случае, если контроллер диска обнаружит ошибку при запуске команд, например в команде "STAT*.*" или любой другой команде чтения и записи на диске. Эта ошибка обычно возникает вследствие неисправности оборудования или вследствие того, что диск чрезвычайно изношен. Эта ошибка может быть, кроме того, вызвана отличиями в формате диска при его форматировании на другой вычислительной установке, оснащенной оборудованием с другими контроллерами диска. Даже если большинство контроллеров дисков, совместимых с ОС CP/M, поддерживают стандарт фирмы IBM, можно обнаружить отличие в формате записи, даже если использованная для форматирования программа та же самая.

При получении этого сообщения курсор останется на той же строке. Предполагается, что для восстановления нормальной ситуации используется команда CTRL/C (реинициализация системы), после чего курсор переместится на следующую строку и появится основной промпт ОС CP/M. Если вместо этого нажата клавиша RETURN, то ОС CP/M проигнорирует ошибку и продолжит выполнение первоначально заданной команды. Предполагается, что клавишу не нажимают, поскольку целостность формата диска при этом может быть нарушена. После восстановления нормального режима работы необходимо либо переформатировать диск, либо скопировать его содержимое на другой диск, если в данной системе имеется программа копирования, которая игнорирует или исправляет ошибки. Не пытайтесь использовать программу "PIP", поскольку наиболее вероятно, что будет получено точно такое же сообщение об ошибке.

BDOS ERR ON X: FILE R/O

Это сообщение имеется только в ОС CP/M версии 2.0 или выше. Сообщение выдается в том случае, если пользователь пытается с помощью команды ERA удалить файл, статус защиты которого "R/O" ("Только чтение"). Кроме того, сообщение выдается при любых попытках осуществить запись в файл, например открыть и закрыть его с помощью текстового редактора или текстового процессора. Любая операция изменения содержимого файла со статусом защиты "R/O" вызовет появление этого сообщения. При возникновении ошибки курсор останется на той же строке. Нажмите клавишу RETURN или введите команду CTRL/C (реинициализация системы), после чего курсор переместится на следующую строку и появится основной промпт ОС CP/M (A >, B >, C > и т. д.). Если требуется удалить файл или изменить его содержимое, тогда с помощью команды STAT можно изменить статус защиты файла на "R/W" ("Только запись").

BDOS ERROR ON X: READ ONLY

Это сообщение об ошибке имеется только в ОС *CP/M* версий 1.4, 2.0 и выше. Сообщение выдается в том случае, если пользователь предпринял попытку осуществить запись на диск со статусом защиты R/O ("Только чтение"), присвоенным с помощью команды STAT или непосредственно в секции BDOS. Статус защиты "R/O" назначается диску по команде "STAT X: = R/O" (здесь "X" – имя диска) или в результате модификации секции BDOS.

После получения сообщения курсор останется на строке с сообщением. Нажмите клавишу RETURN или введите команду CTRL/C (реинициализация системы), после чего курсор переместится на следующую строку, а диску присваивается статус "R/W" ("Чтение/запись").

BDOS ERR ON X: SELECT

Это сообщение появляется, если в командной строке указано имя несуществующего дискового устройства (X:)*. При получении сообщения курсор останется на строке с сообщением. Нажатие любой клавиши (RETURN или CTRL/C) вызовет реинициализацию системы, после чего курсор переместится на следующую строку и появится основной промпт ОС *CP/M* (A >, B >, C > и т. д.).

Кроме этого сообщение об ошибке выдается в том случае, если в устройство, имя которого указано в командной строке, не вставлен диск. При этом система может "зависнуть". Для восстановления нормального режима работы вставьте диск в устройство и введите команду CTRL/C (реинициализация системы). При этом курсор переместится на следующую строку и появится основной промпт ОС *CP/M*.

FILE NOT FOUND

Это сообщение появляется в том случае, если указанный в командной строке файл (или файлы) отсутствуют на диске. Кроме того, это сообщение выдается при ошибке в спецификации имени диска (например, после имени диска не указано двоеточие). После того как сообщение выдано, курсор автоматически переместится на следующую строку и снова появится основной промпт ОС *CP/M*.

INVALID ASSIGNMENT

Это сообщение появляется в том случае, если при назначении с помощью команды STAT физического устройства логическому в имени физического или логического устройства допущена орфографическая ошибка; или такого устройства не существует. При получении сообщения курсор автоматически переместится на следующую строку, после чего появится основной промпт ОС *CP/M*. Для вывода списка допустимых назначений физических устройств логическим введите команду STAT VAL:. Для вывода списка текущих назначений введите команды "STAT DEV:".

INVALID DISK ASSIGNMENT

Это сообщение выдается только в ОС *CP/M* версии 2.0 или выше. Сообщение появляется в том случае, если при установлении статуса защиты диску допущена ошибка. Разрешенными командами являются "STAT X: = R/O" или "STAT X: = R/W". Если после "STAT X: =" не указано "R/O" или "R/W", то выдается это сообщение. После выдачи сообщения курсор автоматически переместится на следующую строку, на которой появится основной промпт ОС *CP/M*.

INVALID FILE INDICATOR

Это сообщение существует только в ОС *CP/M* версии 2.0 или выше. Сообщение выдается, если при назначении с помощью команды STAT атрибутов файла допущена ошибка. Разрешенными атрибутами файла являются: "R/O", "R/W", "DIR" и "SYS". Если

* Напоминаем читателю, что имя диска и имя дискового устройства в ОС *CP/M* совпадают. – Прим. перев.

указан другой атрибут, то появится это сообщение. После вывода сообщения курсор автоматически переместится на следующую строку, на которой появится основной промпт ОС CP/M.

ПРИЛОЖЕНИЕ 3

СВОДНЫЙ СПИСОК КОМАНД ОС CP/M

ВСТРОЕННЫЕ КОМАНДЫ

DIR (Печатать оглавление) :

- | | |
|------------------|--|
| DIR | Печатать оглавление текущего диска. |
| DIR *.* | Печатать список всех файлов, расположенных на текущем диске. |
| DIR *.ext | Печатать список всех файлов с расширением ".ext". |
| DIR filename.* | Печатать список всех файлов с именем "filename". |
| DIR filename.ext | Печатать имя только указанного файла. |

REN (Переименовать) :

- | | |
|--------------------------------------|--|
| REN new name.ext =
= oldname.ext" | Изменить имя файла с "oldname.ext" на "newname.ext". |
|--------------------------------------|--|

ERA (Удалить) :

- | | |
|------------------|---|
| ERA filename.ext | Удалить указанный файл. |
| ERA filename.* | Удалить все файлы с именем "filename". |
| ERA *.ext | Удалить все файлы с расширением ".ext". |
| ERA *.* | Удалить все файлы на диске. |

TYPE (Вывести содержимое текстового файла) :

- | | |
|-------------------|---|
| TYPE filename.ext | Вывести на консоль содержимое файла "filename.ext". |
|-------------------|---|

USER (Выбрать область пользователя на диске) :

- | | |
|--------|--|
| USER x | Установить текущей область пользователья с номером "x" (где "x" равно от 0 до 15). |
|--------|--|

SAVE (Сохранить содержимое оперативной памяти в файл на диске) :

- | | |
|---------------------|--|
| SAVE n filename.COM | Сохранить "n" страниц оперативной памяти (одна страница = 256 байт) на диск в файл "filename.COM". |
|---------------------|--|

Примечание. Для того чтобы специфицировать файл, который расположен не на текущем диске, а на каком-либо другом, в командной строке любой команды, за исключением команды USER, перед именем файла (заданным явно или с помощью метасимволов) может быть дополнительно указано имя диска (A:, B: и т. д.). В качестве метасимволов можно использовать символы "*" (замена группы символов) и "?" (замена одиночного символа). Имена файлов могут быть длиной не более восьми символов, а их расширение – длиной не более трех символов.

КОМАНДЫ УПРАВЛЯЮЩИХ СИМВОЛОВ

Команды управляющих символов используются в основном командном режиме ОС CP/M и иногда в процессе выполнения определенных программ;

ввод специальных команд осуществляется при одновременном нажатии клавиши CTRL и той, которая соответствует конкретной команде.

CTRL/C	Реинициализация системы.
CTRL/M	Возврат каретки.
CTRL/J	Перевод строки.
CTRL/H	Возврат курсора назад на одну позицию.
DELETE или	
RUBOUT	Удалить расположенный слева от курсора символ.
CTRL/U	Удалить текущую командную строку.
CTRL/X	Удалить и стереть текущую командную строку.
CTRL/E	Переход текущей командной строки на следующую строку.
CTRL/R	Перепечатать отредактированную командную строку на следующей строке и опустить все символы, удаленные с помощью команды DELETE или RUBOUT .
CTRL/S	Остановить/Продолжить перемещение текста на экране консоли.
CTRL/P	Вывести на устройство печати (LST) любой текст, распечатываемый на консоли.

ТРАНЗИТИВНЫЕ КОМАНДЫ

SYSGEN (Генерация или копирование ОС *CP/M*) :

SYSGEN	Программе "SYSGEN" перейти в режим диалога.
SYSGEN file-name.ext	Скопировать программный код ОС <i>CP/M</i> из файла "file-name.ext".

MOVCPM (Настройка ОС *CP/M* на использование указанного объема оперативной памяти) :

MOVCPM	Настроить ОС <i>CP/M</i> на использование всей доступной оперативной памяти и передать ей управление.
MOVCPM nn	Настроить ОС <i>CP/M</i> на использование "nn" килобайт оперативной памяти и передать ей управление.
MOVCPM **	Настроить ОС <i>CP/M</i> на использование всей доступной оперативной памяти и подготовить ее к записи в файл.
MOVCPM nn *	Настроить ОС <i>CP/M</i> на использование "nn" килобайт оперативной памяти и подготовить ее к записи в файл.

Примечание. При использовании команды **MOVCPM** в одном из последних двух форматов вновь сформированная операционная системы остается в оперативной памяти, и для того, чтобы сохранить ее на диске, необходимо воспользоваться командой **SAVE**. Затем для создания загружаемой версии операционной системы можно применить команду **SYSGEN**.

STAT (Статус для характеристики дисков и файлов) :

STAT Вывести объем свободного пространства на всех дисках.

Примечание. Обычно по этой команде выводится объем неиспользованного пространства на текущем диске. Однако если перед этим осуществляется обращение и к другим дискам (чтение, запись и пр.), то выводится объем неиспользованного пространства для всех этих дисков. Для вывода объема неиспользованного пространства на конкретном диске, после команды **STAT** можно указать его имя (A:, B: и т. д.).

STAT filename.ext	Вывести объем файла "filename.ext" в килобайтах.
STAT filename.*	Вывести размеры всех файлов с именем "filename" в килобайтах.
STAT*.ext	Вывести размеры всех файлов с расширением ".ext" в килобайтах.
STAT*.*	Вывести размеры всех файлов на диске.
Примечание. Для получения более детальной информации о размерах файлов в перечисленные выше командные строки можно дополнительно включить параметр "QS".	
STAT filename.ext Q R/O	Установить статус защиты файла (файлов) "R/O" ("Только чтение").
STAT filename.* Q R/O	
STAT*.ext Q R/O	
STAT*.* Q R/O	
STAT filename.ext Q R/W	Установить статус защиты файла (файлов) "R/W" ("Чтение/запись").
STAT filename.* Q R/W	
STAT*.ext Q R/W	
STAT*.* Q R/W	
STAT x: = R/O	
STAT filename.ext Q Q Q S	Временно установить статус защиты диска с именем "x" "R/O" ("Только чтение").
STAT filename.* Q SYS	Установить для файла (файлов) атрибут "системный" с тем, чтобы его (их) имя нельзя было вывести на консоль с помощью команды DIR.
STAT*.ext Q SYS	
STAT*.* Q SYS	
STAT filename.ext Q DIR	Установить файлу (файлам) атрибут "DIR" с тем, чтобы его (их) можно было вывести на консоль с помощью команды DIR.
STAT filename.* Q DIR	
STAT*.ext DIR	
STAT*.* Q DIR	
STAT USR:	
STAT DSK:	Вывести номер текущей области пользователя и номера всех непустых (содержащих файлы) областей пользователя.
STAT x: DSK:	Вывести характеристики всех активных дисков.
STAT DEV:	Вывести характеристики диска с именем "x".
STAT VAL:	Вывести список текущих назначений физических устройств логическим.
STAT log: = phy:	Вывести список допустимых назначений и других допустимых форматов команды STAT. Назначить физическое устройство "phy:" логическому устройству "log:", где допустимыми назначениями являются:

Логическое устройство Физическое устройство

CON: = TTY:, CRT:, BAT: или UC1:
 RDR: = TTY:, PTR:, UR1: или UR2:
 PUN: = TTY:, PTP:, UP1: или UP2:
 LST: = TTY:, CRT:, LPT: или UL1.

PIP (Программа обмена с периферийными устройствами) :

PIP

PIP y: = x: filename.ext
PIP y: = x: filename.*
PIP y: = x: *.ext
PIP y: = x: *.*

PIP y: newfile.ext = x: oldfile.ext

Загрузить программу "PIP" в оперативную память и перейти в режим диалога.

Копировать файл (файлы) с диска "x:" на диск "y:".

PIP y: newfile.ext = x: oldfile1.ext, oldfile2.ext, oldfile3.ext

Копировать в указанном выше порядке файлы, расположенные на диске "x:", и поместить копии в том же самом порядке (объединить) в файл "newfile.ext", расположенный на диске "y:".

PIP CON: = x: filename.ext
PIP TTY: = x: filename.ext
PIP UC1: = x: filename.ext
PIP CRT: = x: filename.ext

PIP PUN: = x: filename.ext
PIP PTP: = x: filename.ext
PIP UP1: = x: filename.ext
PIP UP2: = x: filename.ext
PIP OUT: = x: filename.ext

Копировать файл "filename.ext", расположенный на диске "x:", на указанное консольное устройство вывода.

PIP LST: = x: filename.ext
PIP LPT: = x: filename.ext
PIP UL1: = x: filename.ext
PIP PRN: = x: filename.ext

Копировать файл "filename.ext", расположенный на диске "x:", на указанное устройство печати.

PIP x: filename.ext = CON:
PIP x: filename.ext = TTY:
PIP x: filename.ext = CRT:
PIP x: filename.ext = UC1:
PIP x: filename.ext = RDR:
PIP x: filename.ext = PTR:
PIP x: filename.ext = UR1:
PIP x: filename.ext = UR2:
PIP x: filename.ext = INP:

Записывать данные, вводимые с указанного консольного устройства, в файл "filename.ext", расположенный на диске "x:", до ввода команды CTRL/Z.

PIP PUN: = NUL:
PIP PUN: = NUL;
x: filename.ext
PIP PUN: = NUL;
x: filename.ext, NUL:
PIP PUN: =
x: filename.ext, NUL:

Записывать данные, вводимые с указанного устройства, работающего в режиме "только ввод", в файл "filename.ext", расположенный на диске "x:", до ввода команды CTRL/Z.

Вывести начальные и хвостовые промежутки при копировании файла "filename.ext", расположенного на диске "x:", на устройство, работающее в режиме "только вывод".

PIP PUN: = x: filename.ext, EOR	Вывести маркер конца файла при копировании файла "filename.ext", расположенного на диске "x:", на устройство, работающее в режиме "только вывод".
Дополнительные параметры команды PIP:	
[V]	Проверить правильность копирования путем сравнения результирующего и исходного файлов.
[E]	Отображать копируемый текст на консоли.
[Sstring ^Z]	Начать копирование после того, как в исходном файле будет обнаружена строка "string" (для завершения строки используется команда CTRL/Z).
[Qstring ^Z]	Завершить копирование после того, как в исходном файле будет обнаружена строка "string" (для завершения строки используется команда CTRL/Z).
[L]	В процессе копирования преобразовывать все символы верхнего регистра в символы нижнего регистра.
[U]	В процессе копирования преобразовывать все символы нижнего регистра в символы верхнего регистра.
[Dn]	В процессе копирования удалять в каждой строке все символы, расположенные после позиции с номером "n".
[Tn]	В процессе копирования преобразовывать все символы табуляции кода ASCII в "n" пробелов.
[F]	В процессе копирования удалить все символы перевода формата.
[Pn]	В процессе копирования через каждые "n" строк вставлять символы перевода формата.
[N]	В процессе копирования дополнить строки соответствующими порядковыми номерами без ведущих нулей.
[N2]	В процессе копирования дополнить строки соответствующими порядковыми номерами с лидирующими нулями.
[R]	Копировать "системный" файл (устанавливается с помощью команды STAT).
[W]	Удалить файл со статусом защиты "R/O" (устанавливается с помощью команды STAT), имеющим то же имя, что и результатирующий файл.
[Gn]	Копировать файл (файлы) из области пользователя с номером "n" (от 0 до 15) в файл, расположенный в текущей области пользователя.
[O]	Рассматривать содержимое копируемого файла (файлов) как объектный/машинный код.
[B]	Установить "блочный" режим копирования.
[H]	Использовать шестнадцатеричный формат фирмы Intel при передаче данных на (с) устройство, работающее в режиме "только вывод" ("только ввод").
[I]	При передаче данных в шестнадцатеричном формате фирмы Intel игнорировать все нулевые (NULL) записи.
[Z]	В процессе копирования символов в коде ASCII устанавливать бит четности в ноль.

ED (Текстовый редактор) :

ED filename.ext Загрузить программу "ED" в оперативную память и открыть файл "filename.ext" для создания и/или редактирования.

Команды программы "ED":

nA	Добавить "n" строк текста из исходного файла в буфер редактирования ("nA") до тех пор, пока буфер не будет заполнен наполовину ("0A") или полностью ("#A").
I	Вставлять текст до тех пор, пока не будет получена команда CTRL/Z.
Itext	Вставить указанный текст и остановиться в режиме вставки.
Itext^Z	Вставить указанный текст и выйти из режима вставки.
U	Установить ("U") или отменить "--U") преобразование символов в коды верхнего регистра.
nC	Переместить указатель СР на "n" символов вправо ("nC") или влево ("–nC").
nL	Переместить указатель СР на "n" строк вперед ("nL") или назад ("–nL").
V	Вызвести ("V") или прекратить вывод ("–V") номеров строк.
–V	
n:	Переместить указатель СР на строку с номером "n".
B	Переместить указатель СР в начало (B) или в конец (–B) буфера.
T	Вызвести текст, расположенный в текущей строке до (T) или после (OT) указателя СР, или всю строку полностью (OTT).
OTT	
nT	Вызвести "n" строк, расположенных после ("nT"), до ("–nT") указателя СР, или все строки, расположенные после ("#T") или до ("–#T") указателя СР.
–nT	
#T	
–#T	
nP	Переместить указатель СР на "n" страниц вперед ("nP") или назад ("–nP") и вызвать текст.
–nP	
nD	Удалить "n" символов, расположенных в текущей строке после ("nD") или до ("–nD") указателя.
–nD	
nK	Удалить "n" строк, расположенных после ("nK") или до ("–nK") позиции указателя СР.
–nK	
F text^Z	Найти в буфере строку "text" ("F text ^Z ") или найти в буфере "n"-е вхождение строки "text" ("nF text ^Z "). Для завершения строки "text" необходимо использовать команду CTRL/Z.
nF text^Z	
N text^Z	Аналогична предыдущей за исключением того, что здесь при необходимости выполняется автоматическая загрузка текста из исходного файла.
nN text^Z	
S find^Z replace^Z	Найти в буфере строку "find" и заменить ее строкой "replace". Выполнить команду поиска—замещения "n" раз. Символы " ^Z " обозначают команду CTRL/Z.
nS find^Z replace^Z	
J find^Z insert^Z end^Z	Найти в буфере строку "find", вставить после нее строку "insert" и удалить расположенный далее текст вплоть до строки "end".
nJ find^Z insert^Z end^Z	Выполнить операцию поиска—цепления "n" раз. Символы " ^Z " обозначают команду CTRL/Z.

nM commands	Выполнить "n" раз последовательность команд "commands".
OV	Определить размеры свободного и занятого пространства в буфере.
W	Запись из буфера редактирования во временный файл первую ("W") или "n" первых ("nW") строк.
nW	Записать из буфера редактирования во временный файл половину ("OW") или все ("#W") строки.
OW	Записать из буфера редактирования во временный файл половину ("OW") или все ("#W") строки.
#W	Записать "n" строк (начиная с текущей) из буфера во временный библиотечный файл.
nX	Записать "n" строк (начиная с текущей) из буфера во временный библиотечный файл.
R	Записать в буфер редактирования полное содержимое временного библиотечного файла (созданного по команде "nX").
Rfilename	Записать в буфер редактирования содержимое библиотечного файла "filename" (с расширением ".LIB").
H	Записать результаты редактирования в выходной файл без завершения работы программы "ED".
E	Завершить работу программы "ED" (вернуть управление ОС CP/M) и сохранить результаты редактирования в выходном файле.
O	Очистить буфер редактирования и выходной файл без завершения работы программы "ED".
Q	Завершить работу программы "ED" (вернуть управление ОС CP/M) и очистить буфер и выходной файл.

SUBMIT (Запуск на выполнение последовательности команд ОС CP/M):

SUBMIT filename a b c Запустить на выполнение командный файл "filename" (с расширением ".SUB"), заменив предварительно в тексте параметр "Q1" значением "a", параметр "Q2" – значением "b", параметр "Q3" – значением "c" и т. д.

XSUB (Расширение программы "SUBMIT"):

Эта команда размещается в командных файлах (с расширением ".SUB") для передачи команд и данных другим программам, которые запущены на выполнение программой "SUBMIT". Самостоятельно эта команда не используется.

ASM (Ассемблер ОС CP/M для микропроцессора Intel 8080):

ASM filename.123 Ассемблировать исходный файл "filename" (с расширением ".ASM"). При этом дополнительные параметры "123" трактуются следующим образом:

"1" – в данной позиции указывается имя диска, на котором находится исходный файл.

"2" – в данной позиции указывается имя диска, на который необходимо записать результирующий файл с объектным кодом (файл с расширением ".HEX") или символ "Z", если вывод файла с объектным кодом необходимо запустить.

"3" – в данной позиции указывается имя диска, на который необходимо записать результирующий файл печати (файл с расширением ".PRN"), или символ "X", если отчет об ассемблировании необходимо вывести на консоль.

DDT (Инструментальное средство динамической отладки программ) :

DDT filename.ext Загрузить программу "DDT" и файл с отлаживаемой программой "filename" в оперативную память.

Команды программы "DDT":

D	Вывести в шестнадцатеричном формате содержимое следующих 192 ячеек памяти, начиная с текущей ("D") или с ячейки, расположенной по заданному адресу ("Dstart").
Dstart	
Dstart, end	Вывести содержимое ячеек памяти, расположенных начиная с адреса "start" и кончая адресом "end".
Estart, end, data	Записать во все ячейки памяти с адреса "start" по адрес "end" шестнадцатеричное значение "data".
Ifilename.ext	Создать блок управления файлом и записать в него имя "filename.ext" с тем, чтобы загружать файл с таким именем в оперативную память по команде "R".
R	
Rstart	Загрузить содержимое файла, имя которого указано в команде I, в оперативную память, начиная с текущей ячейки ("R") или с ячейки, расположенной по адресу "start" ("Rstart").
L	
Lstart	Реассемблировать и вывести на консоль содержимое следующих 192 ячеек памяти, начиная с текущей ячейки ("L") или с ячейки, расположенной по адресу "start" ("Lstart").
Lstart, end	Реассемблировать и вывести на консоль содержимое ячеек памяти, начиная с адреса "start" и кончая адресом "end".
Sstart	Вывести в шестнадцатеричном формате и, возможно, изменить содержимое ячейки памяти, расположенной по адресу "start". Продолжать обработку следующих ячеек до ввода символа ". .".
Astart	Вставить, начиная с адреса "start", оператор языка Ассемблер. Продолжать обработку последующих ячеек памяти до ввода символа ". .".
Mstart1, end1, start2	Копировать содержимое области памяти, расположенной с адреса "start1" по адрес "end1", в другую область памяти, расположенную начиная с адреса "start2".
G	Выполнить всю программу без прерываний.
Gstart	Выполнить программу, начиная с адреса "start".
Gstart, end	Начать выполнение в точке с адресом "start" и прервать выполнение в точке с адресом "end".
Gstart, end1, end2	Начать выполнение программы в точке с адресом "start" и прервать выполнение либо в точке с адресом "end1", либо в точке с адресом "end2".
G, end	Начать выполнение программы с текущей ячейки и прервать выполнение в точке с адресом "end".
G, end1, end2	Начать выполнение программы с текущей ячейки и прервать выполнение либо в точке с адресом "end1", либо в точке с адресом "end2".
Unumber	Выполнить "number" машинных инструкций и по завершении вывести содержимое регистров центрального процессора.

Tnumber	Трассировать выполнение "number" машинных инструкций, распечатывая содержимое регистров центрального процессора после выполнения каждой инструкции.
X	Вывести текущее содержимое всех регистров центрального процессора.
Xregister	Распечатать содержимое регистра "register".
DUMP (Вывести в шестнадцатеричном формате содержимое файла с машинным кодом) :	
DUMP filename.ext	Вывести в шестнадцатеричном формате содержимое файла "filename.ext".
LOAD (Преобразование файла с объектным кодом (файла с расширением ".HEX") в файл с исполняемым машинным кодом программы (файл с расширением ".COM")) :	
LOAD filename	Преобразовать объектный код из файла "filename.HEX", созданного программой "ASM", в исполняемый машинный код и записать его в файл "filename.COM".

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

Ассемблер (ASM)*	Гл. 13, с. 157
Ввод с устройств	Гл. 9, с. 95
Встроенные команды	Гл. 1, с. 13
Вывод на устройства	Гл. 9, с. 91 – 94, 96, 97
Динамическая отладка программ (DDT)	Гл. 15, с. 173
Диски	Гл. 2, с. 22, 61, 62
Дополнительные транзитные программы	Приложение А, с. 234
Загрузка ОС CP/M	Гл. 2, с. 17
– начальная	Гл. 2, с. 18
Изменение	
– адресного пространства (MOVCPM)	Гл. 20, с. 213
– областей пользователя (USER)	Гл. 7, с. 58
– статуса защиты (STAT)	
– – – диска	Гл. 8, с. 69
– – – файла	Гл. 8, с. 68
Интерпретатор языка Бейсик	Приложение Б, с. 260
Использование	
– метасимволов	Гл. 2, с. 27
– ОС CP/M	Гл. 2, с. 16
Команды	
– ASM	Гл. 13, с. 157
– DDT	Гл. 15, с. 173
– DIR	Гл. 3, с. 45
– DUMP	Гл. 17, с. 196
– ED	Гл. 10, с. 112
– ERA	Гл. 5, с. 51
– FORMAT	Гл. 18, с. 198
– LOAD	Гл. 14, с. 171
– MOVCPM	Гл. 20, с. 213
– PIP	Гл. 9, с. 82
– REN	Гл. 4, с. 48
– SAVE	Гл. 16, с. 194
– STAT	Гл. 8, с. 60
– SUBMIT	Гл. 11, с. 147
– SYSGEN	Гл. 19, с. 206
– TYPE	Гл. 6, с. 54
– USER	Гл. 7, с. 57
– XSUB	Гл. 12, с. 153
– автоматического запуска (SUBMIT)	Гл. 11, с. 147
– – – с передачей текста (XSUB)	Гл. 12, с. 153
– встроенные	Гл. 1, с. 13
– общие сведения	Гл. 1, с. 13
– ОС CP/M версии 3.0	Гл. 21, с. 219
– транзитные	Гл. 1, с. 14
– управляющих символов	Гл. 2, с. 29

* В скобках указывается соответствующая команда ОС CP/M. – Прим. перев.

Макроассемблер	
– MAC	Приложение Б, с. 253
– MACRO-80	Приложение Б, с. 257
– RMAC	Приложение Б, с. 255
Метасимволы	Гл. 2, с. 27
Обращение	
– к встроенным командам	Гл. 2, с. 13
– к транзитным командам	Гл. 2, с. 14
– к файлам	Гл. 2, с. 25
Операционная система	
– <i>CP/M</i>	Гл. 1, с. 10
– – – сравнение версий	Гл. 2, с. 34
– – – версия 3.0	Гл. 21, с. 219
– <i>CP/M-86</i>	Приложение В, с. 273
– Concurrent <i>CP/M-86</i>	Приложение В, с. 273
– <i>CP/NET</i>	Приложение В, с. 274
– <i>MP/M</i>	Приложение В, с. 272
– <i>MP/M-II</i>	Приложение В, с. 272
– <i>MP/M-86</i>	Приложение В, с. 274
Печать	
– оглавления диска (DIR)	Гл. 3, с. 45
– содержимого файлов	
– – – объектных (DUMP)	Гл. 17, с. 197
– – – текстовых (TYPE)	Гл. 6, с. 54
– статуса защиты (STAT)	
– – – диска	Гл. 8, с. 63
– – – файла	Гл. 8, с. 65
– фоновая (DESPPOOL)	Приложение А, с. 235
Переименование файлов (REN)	Гл. 4, с. 48
Построение перекрестных ссылок (XREF, CREF-80)	Приложение А, с. 248, 249
Преобразование	
– шестнадцатеричных файлов (LOAD)	Гл. 14, с. 171
– программ для микропроцессора Intel 8080 в программы для микропроцессора Intel 8086/8088 (XLT86)	Приложение А, с. 251
Программа	
– "Монитор"	Гл. 2, с. 18
– обслуживания библиотек	Приложение А, с. 245
– сборщик	Приложение А, с. 239
– CREF-80	Приложение А, с. 249
– DESPOOL	Приложение А, с. 235
– LIB-80	Приложение А, с. 245
– LINK-80	Приложение А, с. 239
– SID	Приложение А, с. 236
– XREF	Приложение А, с. 248
– XLT86	Приложение А, с. 251
– ZSID	Приложение А, с. 238
Промпты ОС <i>CP/M</i>	Приложение Ж, с. 288
Редактирование текстов (ED)	Гл. 10, с. 112
Реинициализация системы	Гл. 2, с. 20

Сборка программ (LINK-80)	Приложение А, с. 239
Символический отладчик	
– SJD	Приложение А, с. 236
– ZSID	Приложение А, с. 238
Создание резервных копий	Гл. 2, с. 19
Сообщение, выдываемое при загрузке	Гл. 2, с. 19
Сохранение содержимого оперативной памяти на диске (SAVE)	Гл. 16, с. 194
Стандартные типы файлов	Гл. 2, с. 31
Транзитные команды	Гл. 1, с. 14
Транслятор с языка	
– Бейсик	
– – – BASCOM	Приложение Б, с. 262
– – – CBASIC	Приложение Б, с. 263
– – – CB-80	Приложение Б, с. 265
– Кобол	
– – – COBOL-80	Приложение Б, с. 270
– Паскаль	
– – – Pascal/MT +	Приложение Б, с. 267
– ПЛ/1	
– – – PL/I-80	Приложение Б, с. 259
– Фортран	
– – – FORTRAN-80	Приложение Б, с. 268
Устройства	
– логические	Гл. 8, с. 77
– специальные	
– – – входные	Гл. 9, с. 94
– – – выходные	Гл. 9, с. 91
– физические	Гл. 8, с. 79
Файлы	
– копирование	Гл. 9, с. 84
– обращение	Гл. 2, с. 23
– стандартные типы	Гл. 2, с. 31

ОГЛАВЛЕНИЕ

	Стр.
Предисловие	5
Г л а в а 1. Введение	7
Назначение книги	7
Как пользоваться книгой	7
Операционные системы для микро-ЭВМ	8
Операционная система <i>CP/M</i>	10
Операционные системы, построенные на базе ОС <i>CP/M</i>	12
Краткое описание команд ОС <i>CP/M</i>	13
Г л а в а 2. Использование ОС <i>CP/M</i>	16
Общие сведения	16
Загрузка ОС <i>CP/M</i>	17
Использование дисков	22
Обращение к командам и файлам	23
Использование метасимволов	27
Использование метасимвола "*"	28
Использование метасимвола "?"	29
Команды управляющих символов	29
Стандартные типы файлов	31
Сравнение версий ОС <i>CP/M</i>	34
Встроенные команды	34
Транзитные команды	36
Команды управляющих символов	38
Спецификации диска	40
Новые или измененные функции секции BDOS	40
Г л а в а 3. DIR	45
Общие сведения	45
Обращение к команде DIR	45
Вывод на консоль полного оглавления диска	45
Вывод на консоль оглавления для одного файла или группы файлов	46
Г л а в а 4. REN	48
Общие сведения	48
Обращение к команде REN	48
Переименование файла, расположенного на текущем диске	48
Переименование файла, расположенного на произвольном диске	50
Г л а в а 5. ERA	51
Общие сведения	51
Обращение к команде ERA	51
Удаление одного файла	51
Удаление группы файлов	52
Удаление всех файлов на диске	53

Г л а в а 6. TYPE	54
Общие сведения	54
Обращение к команде TYPE	55
Вывод содержимого файла при помощи команды TYPE	55
Г л а в а 7. USER	57
Общие сведения	57
Обращение к команде USER	57
Создание или изменение областей пользователя	58
Г л а в а 8. STAT	60
Общие сведения	60
Обращение к команде STAT	63
Способы применения команды STAT	63
Вывод объема свободного пространства и статуса защиты для всех дисков	63
Вывод объема свободного пространства и статуса защиты для конкретного диска	64
Вывод на консоль статуса заданных файлов	65
Вывод на консоль характеристик всех файлов	67
Защита файлов	68
Временная защита всего диска	69
"Скрытые" файлы	70
Определение кодов пользователей диска	72
Вывод на консоль характеристик диска	73
Вывод на консоль списка назначений устройств	77
Вывод всех возможных имен устройств и других возможных параметров команды STAT	78
Назначение логических имен физическим устройствам	80
Г л а в а 9. PIP	82
Общие сведения	82
Обращение и использование утилиты "PIP"	83
Копирование файлов с диска на диск	84
Загрузка утилиты "PIP" и выполнение операций копирования	84
Спецификация операций утилиты "PIP" в одной командной строке	89
Копирование нескольких файлов в один	90
Использование стандартных и специальных выходных устройств	91
Вывод данных на консоль (videoterminal или телетайп)	92
Вывод данных на устройства, работающие в режиме "только вывод"	93
Вывод данных на печатающее устройство	94
Использование стандартных и специальных входных устройств	94
Ввод данных с консоли в файл на диске	95
Ввод данных с устройств, работающих в режиме "только ввод", в файл на диске	95
Пропуск участков перфоленты в начале и конце вывода данных на устройства, работающие в режиме "только вывод"	96
Вывод маркера конца файла при выводе данных на устройства, работающие в режиме "только вывод"	97
Дополнительные параметры команды PIP	97
Сравнение копии и оригинала файла для проверки правильности копирования	97
Вывод всего текста на консоль в процессе копирования	98
Копирование файла, начиная с указанной строки текста	99
Копирование файла до указанной строки текста	100
Совместное использование параметров "начало копирования" и "конец копирования"	101
Преобразование символов верхнего регистра в символы нижнего регистра при копировании	101

Преобразование символов нижнего регистра в символы верхнего регистра при копировании	102
Удаление в каждой строке всех символов, расположенных после указанной позиции	103
Преобразование символов табуляции в указанное число пробелов	104
Удаление в процессе копирования всех символов перевода формата	105
Вставка символа перевода формата после каждого указанного числа строк при копировании	106
Внесение в текст порядковых номеров строк при копировании	107
Копирование "системных" файлов (только в ОС <i>CP/M</i> версии 2.0 и выше)	108
Удаление защищенного файла, имеющего то же имя, что и выходной файл (только в ОС <i>CP/M</i> версии 2.0 и выше)	108
Копирование файлов из областей других пользователей (только в ОС <i>CP/M</i> версии 2.0 и выше)	109
Копирование файлов, содержащих объектный/ машинный код	109
Установка блочного режима копирования	110
Указание шестнадцатеричного формата передачи данных	111
Игнорирование любой нулевой записи при передаче данных в шестнадцатеричном формате фирмы Intel	111
Установление в ноль бита четности каждого копируемого байта	111
Г л а в а 10. ED	112
Общие сведения	112
Алгоритм работы	113
Указатель текущего символа	117
Описание команд программы "ED"	118
Добавление в буфер указанного числа строк из исходного файла	118
Добавление всего текста исходного файла или его частей до того, как буфер будет заполнен наполовину или до конца	120
Вставка текста в буфер	121
Установление или отмена режима перекодировки символов в коды верхнего регистра	124
Перемещение указателя СР на один или несколько символов вперед или назад	125
Перемещение указателя СР на одну или несколько строк вперед или назад	126
Установление или отмена режима вывода номеров строк	127
Перемещение указателя СР в строку с указанным номером	127
Перемещение указателя СР в начало или конец буфера	128
Вывод текста, расположенного до или после указателя СР	128
Перемещение указателя СР на одну или несколько страниц и выдача текста	130
Удаление одного или нескольких символов, расположенных до или после указателя СР	131
Удаление одной или нескольких строк, расположенных до или после указателя СР	132
Поиск строки текста в буфере редактирования (одно- и многократный)	133
Поиск строки текста в буфере редактирования (одно- и многократный) и выполнение при необходимости операций добавления и записи	134
Поиск строки текста в буфере редактирования и замена ее на указанный текст (одно- и многократно)	134
Поиск строки текста в буфере редактирования и сцепление ее с указанным текстом (одно- и многократно)	135
Комбинирование команд перемещения указателя СР	136
Многократное выполнение групп команд	137
Определение размеров свободного и занятого объема в буфере редактирования	138
Запись нескольких строк из буфера редактирования во временный результирующий файл	139
Запись группы строк из буфера редактирования во временный библиотечный файл	140

Чтение в буфер редактирования содержимого временного файла, созданного командой pH	141
Чтение в буфер содержимого библиотечного файла	142
Сохранение отредактированного текста в результирующем файле без завершения программы "ED"	143
Завершение работы программы "ED" с сохранением отредактированного текста в результирующем файле	144
Очистка буфера редактирования и результирующего файла без завершения программы "ED"	144
Прекращение работы программы "ED" с очисткой буфера редактирования и уничтожением результирующего файла	145
Сообщения об ошибках программы "ED"	146
Г л а в а 11. SUBMIT	147
Общие сведения	147
Запуск и использование команды SUBMIT	148
Использование переменных в командных файлах	150
Использование комментариев в командных файлах	152
Специальные возможности команды SUBMIT	153
Г л а в а 12. XSUB	153
Общие сведения	153
Обращение к команде и алгоритмам ее работы	155
Использование команды XSUB	155
Г л а в а 13. ASM	157
Общие сведения	157
Обращение к программе "ASM"	158
Использование команды ASM	159
Ассемблирование файла, расположенного на текущем диске	160
Сообщения об ошибках, выдаваемых на консоль	163
Сообщения об ошибках в исходной программе	164
Формат исходного текста программы	165
Константы и выражения	166
Директивы программы "ASM"	169
Г л а в а 14. LOAD	171
Общие сведения	171
Обращение к программе "LOAD"	171
Г л а в а 15. DDT	173
Общие сведения	173
Обращение к программе "DDT"	174
Загрузка программы "DDT" в память без указания отлаживаемой программы	174
Загрузка программы "DDT" с указанием отлаживаемой программы	175
Описание команд	176
Вывод на консоль содержимого областей памяти	176
Заполнение памяти константами	179
Вставка в управляющий блок FCB имени загружаемого файла	180
Коды имен дисков, используемые в блоке управления файлом (FCB)	181
Чтение файла с диска и загрузка файла в память	181
Реассемблирование и вывод части кода программы	182
Печать и изменение данных, расположенных по некоторому адресу	184
Вставка команд ассемблера по указанному адресу	185
Копирование содержимого области памяти по указанному адресу	186
Исполнение части программы и установление точек прерывания	187

Исполнение определенного числа команд и вывод содержимого регистров	189
Трассировка исполнения определенного числа команд	190
Вывод и изменение содержимого одного или нескольких регистров.	191
Использование встроенного калькулятора для сложения и вычитания шестнадцатиричных чисел.	193
Г л а в а 16. SAVE	194
Общие сведения	194
Обращение к команде	194
Г л а в а 17. DUMP	196
Общие сведения	196
Обращение к программе "DUMP"	197
Г л а в а 18. FORMAT	198
Общие сведения	198
Обращение к программе форматирования диска	199
Форматирование гибкого программно-секторированного диска	201
Г л а в а 19. SYSGEN	206
Общие сведения	206
Обращение к программе "SYSGEN"	207
Способы использования команды SYSGEN	208
Копирование ОС <i>CP/M</i>	208
Запись ОС <i>CP/M</i> в файл	210
Копирование ОС <i>CP/M</i> из файла	212
Г л а в а 20. MOVCPCM	213
Общие сведения	213
Обращение к команде MOVCPCM	214
Способы использования команды MOVCPCM	214
Настройка ОС <i>CP/M</i> на использование всей доступной оперативной памяти и передача ей управления.	214
Настройка ОС <i>CP/M</i> на использование указанного объема оперативной памяти и передача ей управления.	215
Настройка ОС <i>CP/M</i> на использование всей доступной оперативной памяти и подготовка ее к записи в файл.	217
Настройка ОС <i>CP/M</i> на использование указанного объема оперативной памяти и подготовка ее к записи в файл	218
Г л а в а 21. Обзор ОС <i>CP/M</i> версии 3.0 (<i>CP/M Plus</i>)	219
Введение	219
Операционная среда	220
Использование ОС <i>CP/M Plus</i>	220
Обзор команд ОС <i>CP/M Plus</i>	224
Приложение А. Дополнительные транзитные команды	234
Введение	234
Программа фоновой печати ("DESPOLL")	235
Общие сведения	235
Обращение к программе	236
Символический отладчик для микропроцессоров Intel 8080, Intel 8085 ("SID")	236
Общие сведения	236
Обращение к программе	237
Символический отладчик для микропроцессора Zilog Z80 ("ZSID")	238
Общие сведения	238
Обращение к программе	238

Программа-сборщик ("LINK-80")	239
Общие сведения	239
Описание программы "LINK-80" фирмы Digital Research	239
Обращение к программе "LINK-80" фирмы Digital Research	240
Описание программы "LINK-80" фирмы Microsoft	242
Обращение к программе "LINK-80" фирмы Microsoft	243
Программа обслуживания библиотек ("LIB-80")	245
Общие сведения	245
Описание программы "LIB-80" фирмы Digital Research	245
Обращение к программе "LIB-80" фирмы Digital Research	245
Описание программы "LIB-80" фирмы Microsoft	246
Обращение к программе "LIB-80" фирмы Microsoft	247
Программа построения отчета о перекрестных ссылках для программ, написанных на языке Ассемблер ("XREF")	248
Общие сведения	248
Обращение к программе	248
Программа построения отчета о перекрестных ссылках ("CREF-80")	249
Общие сведения	249
Обращение к программе	250
Конвертор программ на языке Ассемблер для микропроцессора Intel 8080 в программы на языке Ассемблер для микропроцессора Intel 8086 ("XLT86")	251
Общие сведения	251
Обращение к программе	251
Приложение Б. Трансляторы с принятых в ОС CP/M версий языков программирования высокого уровня	252
Введение	252
Макроассемблер "MAC"	253
Общие сведения	253
Обращение к макроассемблеру "MAC"	254
Макроассемблер "RMAC"	255
Общие сведения	255
Обращение к макроассемблеру "RMC"	256
Макроассемблер "MACRO-80"	257
Общие сведения	257
Обращение к макроассемблеру "MACRO-80"	258
Транслятор "PL/I-80" (компилятор с языка ПЛ/1 для микропроцессора Intel 8080)	259
Общие сведения	259
Обращение к транслятору "PL/I-80"	260
Интерпретатор "BASIC-80" (интерпретатор с языка Бейсик для микропроцессора Intel 8080)	260
Общие сведения	260
Обращение к интерпретатору "BASIC-80"	260
Транслятор "BASCOM" (компилятор с языка Бейсик фирмы Microsoft)	262
Общие сведения	262
Обращение к транслятору "BASCOM"	262
Транслятор "CBASIC" (компилятор с языка Бейсик)	263
Общие сведения	263
Обращение к транслятору "CBASIC"	264
Транслятор "CB-80" (компилятор чистого кода с языка Бейсик для микропроцессора Intel 8080)	265
Общие сведения	265
Обращение к транслятору "CB-80"	265
Транслятор "PASCAL/MT+" (компилятор с языка Паскаль)	267

Общие сведения	267
Обращение к транслятору "PASCAL/MT+"	267
Транслятор "FORTRAN-80" (компилятор с языка Фортран для микропроцессора Intel 8080)	268
Общие сведения	268
Обращение к транслятору "FORTRAN-80"	269
Транслятор "COBOL-80" (компилятор с языка Кобол для микропроцессора Intel 8080)	270
Общие сведения	270
Обращение к транслятору "COBOL-80"	270
Приложение В. Операционные системы, построенные на базе ОС <i>CP/M</i>	271
Введение	271
Операционная система <i>MP/M-II</i>	272
Операционная система <i>CP/M-86</i>	273
Операционная система <i>Concurrent CP/M-86</i>	273
Операционная система <i>MP/M-86</i>	274
Операционная система для управления сетью ЭВМ <i>CP/NET</i>	274
Приложение Г. Список литературы	275
Книги по ОС <i>CP/M</i>	275
Журнальные статьи по ОС <i>CP/M</i>	275
Книги по программированию на языке ассемблера	282
Приложение Д. Таблица символов кода ASCII	282
Обозначения управляющих символов кода ASCII	286
Приложение Е. Промпты ОС <i>CP/M</i>	288
Приложение Ж. Сообщения об ошибках ОС <i>CP/M</i>	289
Приложение З. Сводный список команд ОС <i>CP/M</i>	291
Встроенные команды	291
Команда управляющих символов	291
Транзитные команды	292
Предметный указатель	300

Производственное издание
Митчелл Уэйт, Джон Ангермейер
ОПЕРАЦИОННАЯ СИСТЕМА CP/M

Заведующая редакцией О. В. Толкачева
Редактор С. Н. Удалова
Художественный редактор Т. В. Бусарова
Переплет художника С. Г. Студениковой
Технический редактор А. Н. Золотарева
Корректор Т. С. Власкина

ИБ № 1152

Подписано в печать 4.04.86 Формат 60x90/16 Бумага офс. № 2 Гарнитура
"Пресс-роман" Печать офсетная Усл. печ. л. 19,5 Усл. кр.-отт. 19,5 Уч.-изд. л. 21,7
Тираж 20 000 экз. Изд. № 21292 Зак. № 1635 Цена 1 р. 80 к.
Издательство "Радио и связь". 101000, Москва, Почтамт, а/я 693

Московская типография № 4 Союзполиграфпрома при Государственном комитете СССР
по делам издательств, полиграфии и книжной торговли. 129041, Москва, Б. Переяславская
ул., д. 46

УВАЖАЕМЫЙ ЧИТАТЕЛЬ!

В 1987 г. в издательстве "Радио и связь" выйдут книги

Э л ф р и н г Г. Программирование на языке ассемблера для микро-ЭВМ: Пер. с англ. — 13 л., ил. — 90 к.

В книге американского автора рассмотрен весь процесс создания программ: от разработки проекта до оформления документации. Особое внимание удалено не описанию командного языка, а вопросам идеологии программирования для микро-ЭВМ. Материал книги насыщен практическими рекомендациями по реализации предлагаемых решений и проиллюстрирован конкретными примерами. В примерах использован язык микропроцессора 8080 фирмы Intel.

Для широкого круга программистов.

У о л ш Б. Настоящий Бейсик: Пер. с англ. — 17 л., ил. — 1 р. 50 к.

В книге известного английского автора содержатся сведения о языке программирования Бейсик, предназначенные как для начинающих (основные правила программирования, а также более сложные конструкции языка с примерами их применения при программировании алгоритмов сортировки), так и для опытных программистов (расширение языка для работы с внешними устройствами ЭВМ, технология разработки программ и методы их отладки). Приведено большое число детально разработанных примеров и задач с решениями.

Для широкого круга программистов.

Приобрести эти книги Вы можете в книжных магазинах, распространяющих научно-техническую литературу.

**СПИСОК МАГАЗИНОВ – ОПОРНЫХ ПУНКТОВ
ИЗДАТЕЛЬСТВА "РАДИО И СВЯЗЬ"**

111024 Москва, шоссе Энтузиастов, 24/43, магазин № 15
197198 Ленинград ПС, Большой пр., 34, магазин № 55
226000 Рига, бульвар Падомью, 17, магазин "Гайсма"
630000 Новосибирск, Красный пр., 60, магазин № 7
443000 Куйбышев, ул. Красноармейская, 62, магазин № 16
700070 Ташкент, ул. Шота Руставели, 43, магазин № 21
390000 Рязань, ул. Циолковского, 1, магазин № 7
173016 Новгород, Ленинградская ул., 13, магазин № 2 "Прометей"
634000 Томск, ул. Нахимова, 15/1, магазин № 15
603000 Горький, пр. Гагарина, 110, магазин № 9

