MS-DOS操作系统高等教程

·汇编语言和C语言程序员手册·

李沐荪　编译

北京科海总公司培训中心

一九八七年九月

前　言

这本书是为熟悉Intel 8086／8088／80286系列微机结构的有经验的汇编或C语言程序员编写的。它提供必要的详尽资料，以便在MS-DOS或派生系列下编写优质程序。

书中详细地探讨了MS-DOS的各项功能和特色，还对各种不同的现存版本作了比较。根据经验，作者相信，利用已经通过的，作为正规文献的程序来说明问题，要胜于一般性的叙述和图表，因此在书中前后列举了大量的详细的编程例子——包括完成特定功能的孤立片段，以及完整的应用程序。本书所有例子都是使用Microsoft宏汇编版本4.00或Microsoft C编译版本3.00和IBM PC硬件开发的。

MS-DOS为在它控制下运行的程序提供大量的各方面的操作系统服务。它们将应用程序与硬件环境隔离开来，大致可分以下几大类，在第一编中论述：

- 字符I/O：键盘和串行端口输入，视频显示，串行端口和行式打印机输出。
- 海量存储：在可动或固定磁盘上维持目录及文件。
- 内存分配及管理。
- 在另外一个程序控制下，装入并执行一个程序任务。

第一编中还讨论了过滤器，设备驱动器及中断处理器的结构—这是编写系统工具和扩充MS-DOS本身功能的程序员特别感兴趣的。此外，还涉及如何为IBM系列个人计算机编写与硬件有关的程序，在某些情况下（特别是视频显示）为使性能可以被接受，这样的程序是必要的。

第二编为MS-DOS中断提供了[illegible]参考资料。[illegible]看清楚调用某一特定功能时需要的寄存器内容，以及执行该功能后—不论成功或失败—所返回的数值。在有关的地方，对一个功能的不易理解之处，或对于它在不同MS-DOS版本下的差别，均加以注释。每一栏均附有简短的汇编语言程序例子，便于读者编写调用时借鉴。

第三编是最常用的IBM PC BIOS中断的参考资料。

本书在编译时有所删节，由于水平有限，不到之处欢迎批评指正。

目　录

第一编　用MS-DOS编程

第一章 MS-DOS的发展沿革

MS-DOS是一种迅速发展的操作系统。过去三年中，每年至少推出一个新的或大或小改动的新版本，已知正在编写更多的版本。主要由于为IBM所采用，以及随着IBM成功而来的巨大的第三者软件冲击，MS-DOS已成为使用Intel 8086系列微处理器的个人计算机的主要操作系统。MS-DOS有许可证的用户多达几百万，所有它的竞争者(包括CP/M-86，Concurrent DOS，P-系统，iRMX-86，XENIX和UNIX) 的用户加在一起也无法相比。

从程序员的观点来看，MS-DOS的当前版本（版本2和3）是博大精深而强有力的开发环境。Microsoft和其他软件公司提供大量的高质量的编程工具以资选择。将现有应用程序转入MS-DOS环境是比较简单的，因为程序员可将MS-DOS看成是CP/M的超集或UNIX的子集。

MS-DOS的原本是一个称为86-DOS的操作系统，由Tim Paterson在八十年代中期为西雅图计算机产品公司编写。其时，Digital Research公司的CP/M-80是微机最常用的操作系统，有相当好，虽然还不够广泛的应用软件（字处理器，数据库等）可配合使用。为便于将8位CP/M-80应用转入新的16位环境，86-DOS原来设计得在可供调用的功能和操作风格上模仿CP/M-80。因而86-DOS的文件控制块，程序段前缀和可执行文件的结构几乎和CP/M-80的一致。现存的CP/M，程序可以机械地变换（将它们的源码通过一个专用翻译程序），变换之后，或是可以立即在86-DOS下运行，或是稍加手编即可。

1980年10月，IBM到各主要微机软件公司去找一个操作系统，以供它正在设计的新个人计算机使用。Microsoft自己没有操作系统可以提供（除一个独立的Microsoft BASIC之外），但向西雅图计算机产品公司购得销售Paterson的86-DOS的权利，（其时，西雅图计算机产品公司收到一张许可证，可以使用和出售Microsoft语言和所有Microsoft操作系统的8086版本）。1981年7月，Microsoft购入86-DOS所有权利，对它作了实质性的改变，并重新命名为MS-DOS。当1981年秋天推出第一台IBM PC时，IBM提出MS-DOS（称为PC-DOS1.0）作为它的原始操作系统。

IBM也选取Digital Research的CP/M-86和Softech的P-system作为PC机的替换操作系统。然而，两者在IBM PC销售商处出现得很慢，外加还有价格高，没有可供编程的语言等缺点。IBM在PC-DOS后面投入相当的力量，推出所有IBM的PC应用软件和开发工具在PC-DOS下运行。因此，大多数第三软件开发者一开始就将他们的产品瞄在PC-DOS上，CP/M-86和P-system在IBM PC兼容市场上从未成为重要因素。

尽管表面上与上代CP/M-80有相似之处，MS-DOS 1.0版本含有很多对CP/M的改进，包括：

- 改进的磁盘目录结构，包括了关于文件属性的信息（例如是否为系统文件或隐藏文件），它的确切大小（以字节计），以及文件建立或上次修改的日期。
- 优越的磁盘空间分配及管理方法，允许极端快速的顺序或随机记录访问和程序装

入。

- 一套扩充的操作系统服务，包括与硬件无关的功能调用以设定或读出日期与时间，一个文件名分析器，多块记录I/O，以及可变记录大小。
- 一个AUTOEXEC批文件，在系统上电或复位时执行用户定义的命令系列。

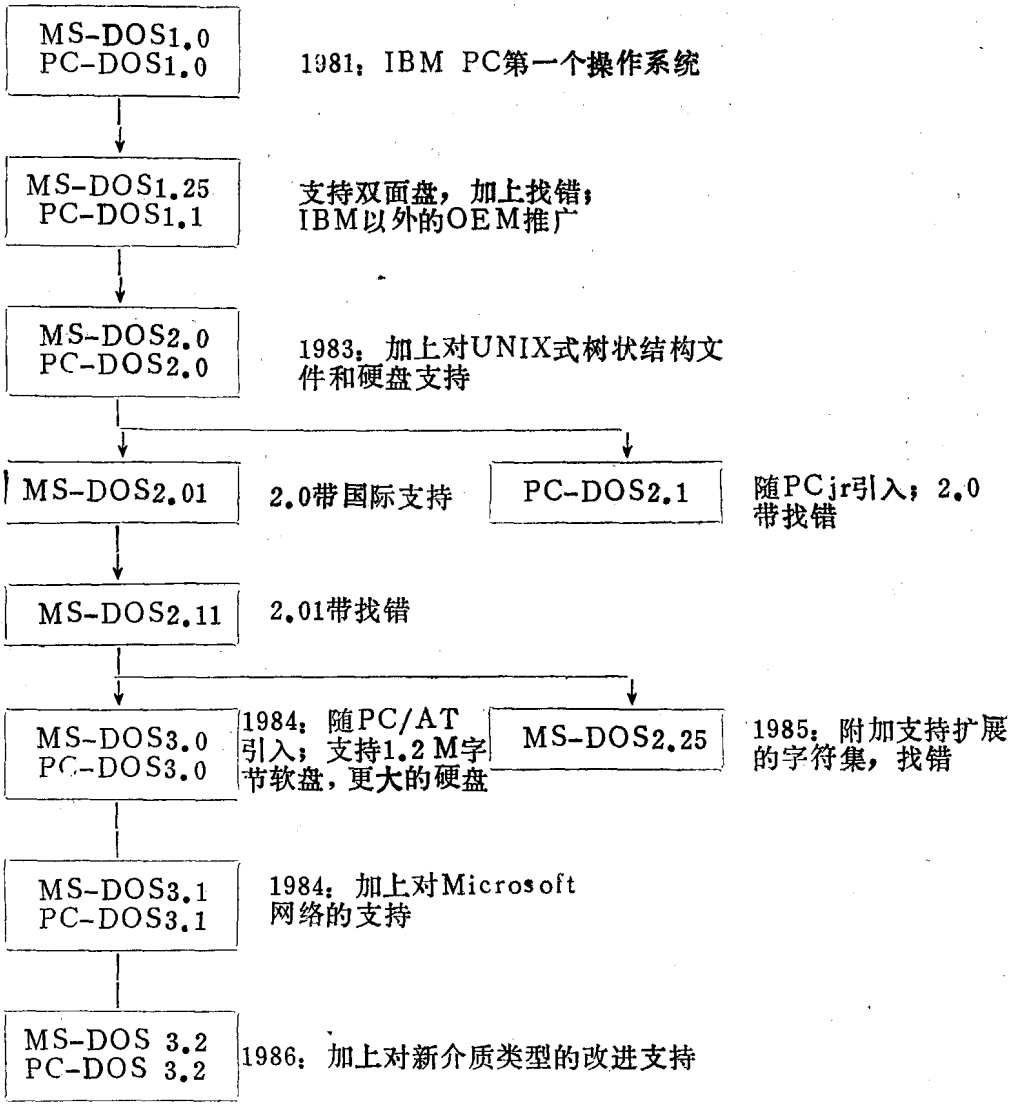

图1.1 MS-DOS的发展

IBM是唯一的将MS-DOS版本1.0（作为PC-DOS1.0）随同其产品销售的主要计算机制造者（有时称为OEM，原文为original equipment manufacturer）。MS-DOS版本1.25（相当于IBM PC-DOS1.1）于1982年6月推出，改正了一些错误，支持双面盘并改进了DOS内核的硬件独立性。这个版本也被其他厂商销售。今天，主要由于硬盘系统日见优势，MS-DOS版本1已不再常用。

MS-DOS版本2.0（相当于PC-DOS2.0）首先于1983年3月推出。回顾起来，这是一个全新的操作系统（虽然作了重大的努力来保持与MS-DOS版本1的兼容性）。它含有

许多重要的革新和改进，包括：

- 支持较大容量的软盘和硬盘。
- 许多UNIX式的特色，包括树状文件结构，文件把柄，I/O再定向，命令衔接和过滤程序
- 背景打印（假脱机打印）。
- 卷标和附加的文件属性。
- 可装入设备驱动器。
- 一个可由用户剪裁的系统配置文件，用来控制附加设备驱动器的装入，系统磁盘缓冲器的数目等。
- 维护程序环境块，它们可以用来在程序之间传递信息。
- 一个可供选用的ANSI显示驱动器，允许程序以与硬件无关的方式调整光标位置和控制显示特性。
- 支持应用程序对内存块的动态分配、修改和释放。
- 支持用户自编的命令解释器（外壳）。
- 备有系统表格，协助应用软件修改它的货币符号，时间及日期格式(称为国际支持)。

MS-DOS版本2.11跟着被推出以改进国际支持（表格方式的货币符号，日期格式，小数点符号，货币分隔符等），加上对整个16位汉字的支持，并修改了一些错误。

在本书写作期间，几乎所有主要OEM厂商，包括Hewlett—Packard，Wang，DEC，Texas Instruments，Compaq和Tandy都以MS-DOS版本2.11作为8086/8088型个人计算机的基础版本。因此，应用程序应当设计得在这一版本来运行。

在1985年10月推出的MS-DOS版本2.25中，国际支持甚至于更进一步被扩充到日文和朝鲜文字符，进一步改正了差错，许多系统公用程序编制得与MS-DOS3.0兼容。

MS-DOS3.0由IBM在1984年8月，随着80286 PC/AT机的推出而首先引入，在本书写作期间，逐渐也由其他OEM提供。它包括下列主要新特色：

- 由应用软件直接控制假脱机打印器。
- 在版本2.11的基础上进一步扩充了国际支持（但不如2.25那样广泛）。
- 扩展错误报告，包括一个向调用程序提供策略对等的代码。
- 支持文件和记录的锁定和共享，便于建立网络应用。
- 支持更大的硬盘。

在MS-DOS版本3.1中又增加了对Microsoft的网络的支持和改正了某些小错，版本3.1是在版本3.0之后不久，于1984年11月推出的。

考察一下MS-DOS从它简陋的开始如何稳步增长是很有趣的。操作系统的版本1占大约16KRAM，可以在64K机器上很好地运行应用程序。MS-DOS版本2约耗费24K RAM（或更多，如果有装入的设备驱动器）需要128K机器才能做出有用的事情。MS-DOS版本3占36KRAM，如果装入文件共享支持和某些用户装入的驱动器，可能需要更多（通常运行于至少有512KRAM的机器上）。

展望未来，据报告有一种MS-DOS的特殊版本，是一种完全多任务操作系统，而工业谈论家期望另一种版本，能够在80286处理上以保护模式运行，同时对绝大多数现存MS-DOS应用程序提供向上兼容性。这样的一种技术上复杂的操作系统将打开充分开拓80286能力的大门，可以访问16兆字节物理内存和1千兆虚拟内存。

第二章 MS-DOS的运行

本章讨论MS-DOS的组成和计算机接通电源时，MS-DOS是怎样装入的。熟悉MS-DOS的一般结构对理解系统的整体表现是很有帮助的。

2.1 MS-DOS的结构

MS-DOS分为若干层，用来将它所赖以运行的硬件与操作系统的核心逻辑，以及用户对系统的感知分隔开来。这些层次是：

- BIOS (Basic I/O system即基本输入输出系统)
- DOS内核
- 命令处理器外壳

下面对这些层的功能分别加以讨论。

2.1.1 BIOS模块

BIOS专属于所用的计算机系统，由系统制造者提供。它含有下列设备的自备常驻硬件驱动程序：

- 控制台 (CON)
- 行式打印机 (PRN)
- 辅助设备 (AUX)
- 日期与时间 (CLOCK)
- 磁盘导引设备

MS-DOS内核通过I/O请求软件包与这些设备驱动程序交往；然后驱动程序将这些请求翻译成各个硬件控制器的相应命令。在许多MS-DOS系统中，包括IBM PC，硬件驱动器的最原始部分位于只读存储器 (ROM) 中，因能够被独立的应用，诊断和系统引导 (自举) 程序使用。

常驻和可装这两个词用来区分写在BIOS中的驱动器和在系统自举过程中，由CONFIG.SYS文件中DEVICE命令装入的驱动器。

BIOS在系统启动时作为文件IO.SYS的一部分装入随机存取内存 (RAM) 中 (在PC-DOS2中，文件称为IBMBIO.COM)。该文件标有特殊属性符：隐藏和系统 (hidden and system)。

2.1.2 DOS内核

DOS内核实现的是应用程序所见到的MS-DOS。内核是Microsoft公司提供的专有程序，它提供一系列称为“系统功能”的与硬件无关的服务例程。系统功能包括：

- 文件与记录管理

·内存管理
·字符设备输入／输出
·其它程序的“派生”
·访问实时钟

程序可以访问这些系统功能，只需将某一功能所特定的参数装入特定寄存器，然而通过一个调用，或软件中断而转送到操作系统。

DOS内核在系统启动时，由引导盘上的MSDOS.SYS文件读入内存（在PC-DOS系统中，文件称为IBMDOS.COM）。该文件标志有属性隐藏和系统

2.1.3 命令处理器

命令处理器，或外壳，是用户与操作系统的接口界面。它承担分析和执行用户命令的任务，包括由磁盘或其他海量存储设备装入和执行其他程序的任务。

MS-DOS的自备外壳可在文件COMMAND.COM中找到。虽然COMMAND.COM的提示和响应构成一般用户对MS-DOS的全部感知，但很重要的是应当理解，COMMAND.COM并不是操作系统，而只不过是在MS-DOS控制下运行的一类专门程序。

COMMAND.COM可以被用户自己设计的外壳所替代，只需在引导盘上的系统配置文件（CONFIG.SYS）中加上一行即可。例如，惠普公司的MS-DOS计算机（触屏式HP-150，便携式HP-110及Vectra）出售时带有一个功能很强的屏幕指向性专用外壳称为Personal Applications Manager（个人应用管理程序）。许多HP微机所有者从未看到过IBMPC用户所熟悉的MS-DOS提示符A>。

2.1.4 关于COMMAND.COM的进一步说明

MS-DOS的自备外壳COMMAND.COM分为三部分：

- 常驻部分
- 初始化部分
- 暂存模块

常驻部分装入内存低部，在DOS内核及其缓冲区和列表区之上，含有一些例程，用来处理Ctrl-C和Ctrl-Break，危急错误和其他暂存程序的终止（最后退出）。COMMAND.COM的这一部分发出出错信息，以及大家熟知的提示：Abort，Retry，Ignore?（放弃，再试，硬性执行？）它还包含在必要时用来重新装入COMMAND.COM暂存部分的代码。

COMMAND.COM的初始化部分在系统被引导时，装在常驻部分之上。它在AUTOEXEC批文件存在时对该文件进行处理，然后初始化部分即被舍弃。AUTOEXEC文件是用户所列出的一系列命令，在系统被引导时执行。

COMMAND.COM的暂存部分装在内存的高端，它的这部分内存也可以被其他应用程序用于其他目的。暂存模块发出用户提示符，由键盘或批文件读入命令，并使这些命令得到执行。当应用程序终止时，COMMAND.COM的常驻部分对暂存模块求检查和，以确定它是否被破坏，并在必要时由磁盘复制新模块。

COMMAND.COM接受的用户命令分为三类：

- 内部命令
- 外部命令
- 批文件

内部命令，有时称为固有命令，由写在COMMAND.COM本身中的代码执行。这类命令包括COPY，REN（AME），DIR（ECTORY）和DEL（ETE）这些内部命令的例程包含在COMMAND.COM的暂存部分中。

外部命令，有时称为外征命令或暂存程序是存储在磁盘文件中的程序名。在这些程序能被执行之前，必须先装入内存的暂存程序区（TPA）中（见本章“MS-DOS是如何装入的”一节）。外部命令的熟知例子为CHKDSK，BACKUP和RESTORE。一旦外部命令执行任务完毕，即由内存中舍弃；因此每次引用时，必须由磁盘重新装入。

批文件是一种文本文件，含有其他内部、外部或批命令。这些文件由一个特殊的解释程序处理，该解释程序装在COMMAND.COM的暂存部分中，它每次读出批文件中的一行，并按顺序执行每一条特定的操作命令。

为解释用户命令，COMMAND.COM首先查看是否为内部命令，这是可以直接执行的。否则，就按照该命令名去搜索外部命令（可执行程序文件）或批命令。首先在当前磁盘驱动器的当前目录中进行搜索，然后在环境PATH字符串中规定的每个目录中搜寻。在寻找的每个目录中，COMMAND.COM首先试找带有扩展名。COM，然后是.EXE，最后是.BAT，的文件。如果在所有可能的位置上寻找这三种文件类型均告失败，COMMAND.COM将显示下列熟悉的信息：

Bad command or file name

如果找到了一个COM文件或EXE文件，则COMMAND.COM利用MS-DOS的EXEC功能装入该文件并执行。EXEC功能于暂存程序区中COMMAND.COM常驻区之上建立起一个专门的数据结构，称为“程序段前缀”（PSP）。PSP含有应用程序所需的各种连接与指针。其次，EXEC功能将程序自身装到紧贴于程序段前缀区之上，并且完成任何需要的再定位。最后，它相应地设定各寄存器，并将控制转移到程序的入口点。（PSP和EXEC功能将于第三及第十章中作更详细的讨论）。当过渡程序完成其作业时，它调用专门的MS-DOS终止功能，并将控制返回到原来使暂存程序被装入的程序中（本情形中，为COMMAND.COM）

在MS-DOS2和3的控制下，外部命令在执行时占用系统的全部资源，唯一例外得以完成的任务是那些由中断处理器（例如键盘输入驱动器和实时钟）而执行的任务，以及暂存程序向DOS请求的操作。这两个MS-DOS版本不允许多个同时执行的任务共享中央处理器，也不能在程序破坏或执行时间过长时夺回控制。

2.2 MS-DOS是如何装入的

当系统复位或上电时，程序在地址OFFFFOH开始执行。这是8086系列微处理器的一个特点，与MS-DOS无关。以这类处理器构成的系统设计时使地址OFFFFOH位于ROM区内，并含有一条跳转机器指令，以便控制转移到系统测试代码及ROM自举例程（图2-1）。

ROM自举例程由磁盘的第一扇区（引导扇区）读入磁盘引导程序到内存某一地址，然后将控制转给导引程序（图2-2）（导引扇区还含有关于磁盘格式的信息表。）

磁盘引导程序检查磁盘是否含有MS-DOS的拷贝。具体做法是读取根目录的第一扇区并确定开始的两个文件是否按顺序为IO.SYS和MSDOS.SYS。如果这两个文件不存在，将提示操作员更换磁盘，并任击一键重新尝试。如果找到这两个系统文件，则磁盘引导程序将它们读入内存，并将控制转移到IO.SYS的初始入口（图2.3）（在某些实现方式中，磁盘导引程序只将IO.SYS读入内存，再由IO.SYS负责装入MSDOS.SYS文件。）

由磁盘装入的IO.SYS文件实际上包括两个模块。第一块是BIOS，它包含连接的常驻设备驱动器，用于键盘显示器、辅助端口、打印机及成块设备等，还包含一些硬件专用初始化代码，这些代码仅在系统引导时运行。第二模块称为SYSINIT。由Microsoft提供，由计算机制造厂家随同BIOS连接到IO.SYS文件。

SYSINIT由制造厂家的BIOS初始化码调用。它确定系统中现有的相邻内存量，然后将自身定位于高内存区。接着，它将DOS内核，MSDOS.SYS，由它的原装入位置移到内存中的最终位置上，覆盖原来的SYSINIT代码，以及IO.SYS文件中所含的其它扩

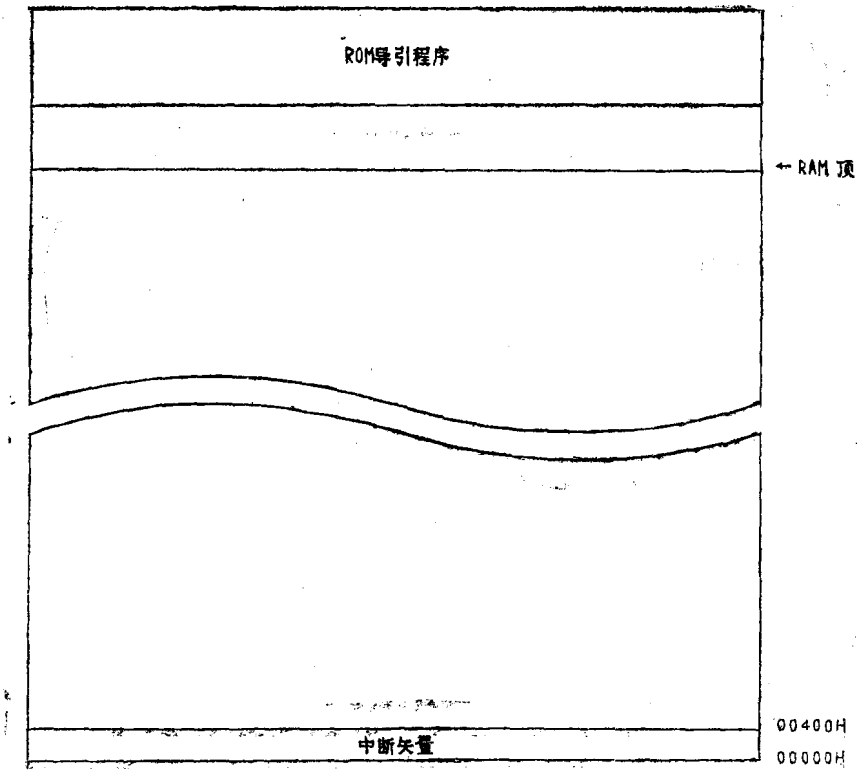

图 2.1 典型的8086/8088计算机系统在上电或复位后的内存现状，执行开始于地址0FFFF0H。该处有一条跳转指令，将程序控制导向ROM自举例程。

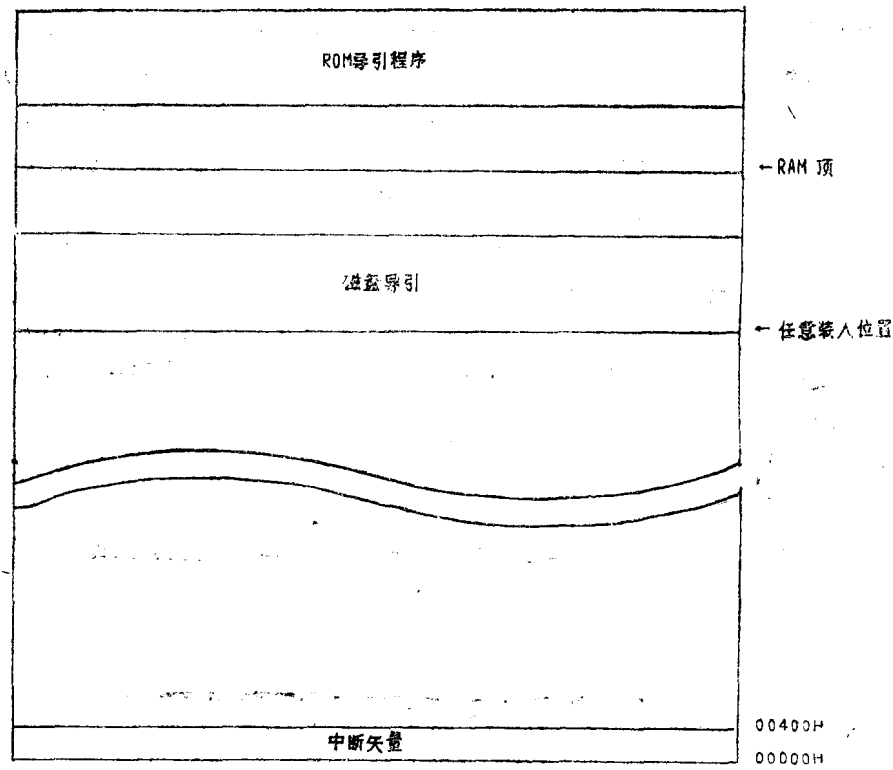

图2.2 ROM自举例程，将磁盘引导程序由系统盘的第一个扇区读入内存，并将控制转移给它。

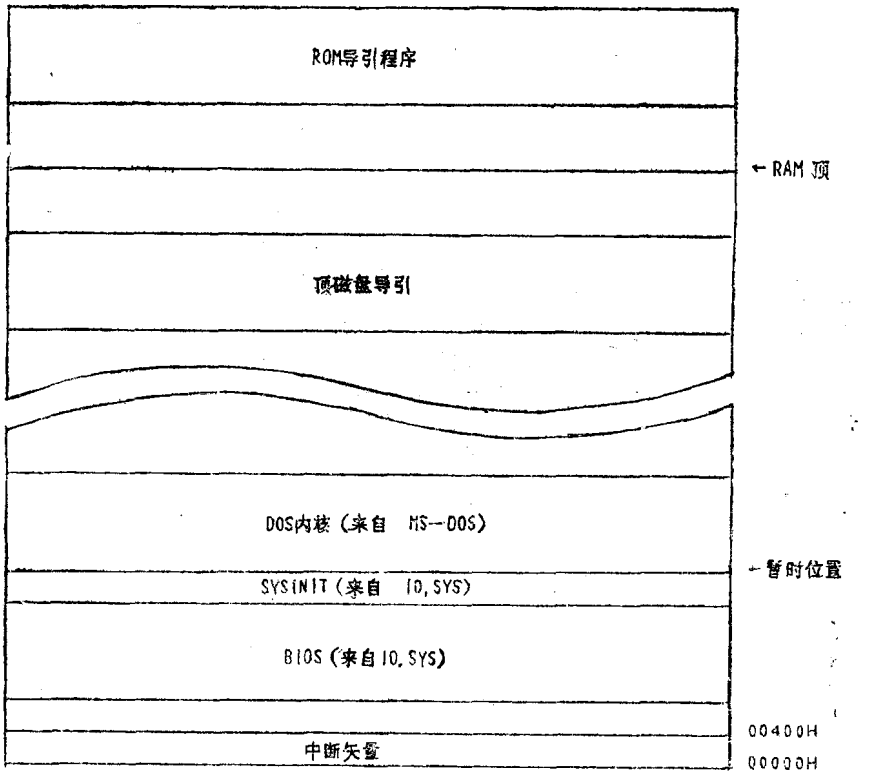

图 2.3 磁盘引导程序将IO.SYS文件读入内存。该文件包含MS-DOS BIOS（常驻设备驱动器）和SYSINIT模块。然后，或是磁盘引导程序或是BIOS（取决于制造者的实现方案）将DOS内核由MSDOS.SYS读入内存。

展初始化码（图2.4）

其次，SYSINIT对MSDOS.SYS中的初始化码作一次调用。DOS内核初始化它的内部表格和工作区，建立20H到2FH中断矢量，然后顺次通过已连接好的常驻设备驱动器表，调用每个驱动器的初始化功能（见第12章）。这些驱动器的功能是确定设备状态，完成必需的硬件初始化，以及设定驱动器为之服务的任何外围硬件的中断矢量。

作为初始化序列的一部分，DOS内核考察由常驻成块设备驱动器返回的磁盘参数块，确定可用于系统的最大扇区容量，建立一定的驱动参数块组，并分配一组磁盘扇区缓冲区。然后，显示MS-DOS版权信息，控制返回到SYSINIT。

现在DOS内核已被初始化，所有常驻设备驱动器均可供使用。SYSINIT可以调用正常的MS-DOS文件服务来打开CONFIG.SYS文件。这一备选文件可以含有各种不同的命令，使用户能够定制MS-DOS环境。

例如，用户可以规定外加的硬件设备驱动器，磁盘缓冲器数目，同一时间可打开的最大文件数，以及命令处理器（外壳）的文件名。

如果发现有CONFIG.SYS，则整个文件被装入内存以备处理。所有小写字符均被变换成大写字符，文件被逐行解释，以对命令进行处理。内存被分配给磁盘快速缓冲器，和

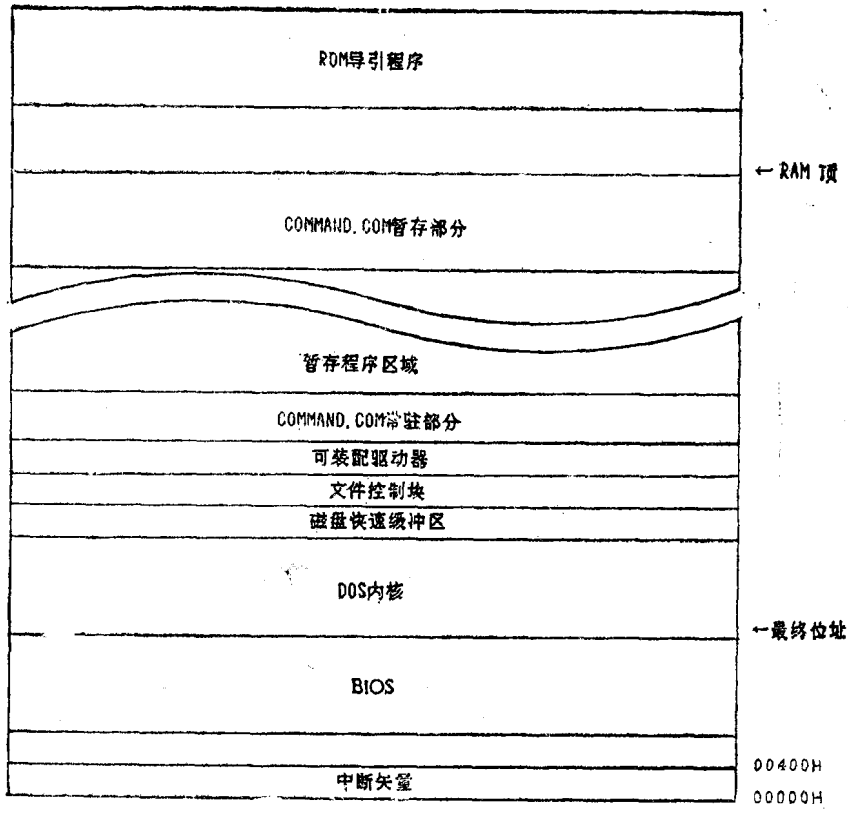

图 2.4 SYSINIT将本身移到内存高区，并将DOS内核，MSDOS.SYS重新定位，向下到最终位置。MS-DOS磁盘快速缓冲区及文件控制块区也被定位，然后CONFIG.SYS文件中规定的可装设备驱动器被装入，并与系统连接。

扩展文件，把柄文件以及记录系统功能所用的内部文件控制块（见第六章）。CONFIG.SYS文件列出的任一设备驱动器均被顺序装入内存，调用它们的init模块实现初始化，并连接到设备驱动器清单中。各个驱动器的init模块告诉SYSINIT需要为该驱动器备留多少内存。

在所有可装设备驱动器被装入之后，SYSINIT关闭所有文件处理把柄，并重新打开控制台（CON），打印机（PRN）以及辅助设备（AUX）作为标准输入，标准输出、标准错误、标准列表和标准辅助设备。这允许用户装置的字符设备驱动器覆盖BIOS的常驻标准设备驱动器。

最后，SYSINIT调用MS-DOS EXEC功能，以装入命令解释器或外壳（应当还记得自定外壳为COMMAN.COM，但是，可以通过CONFIG.SYS文件置换其他外壳）。一旦外壳被装入后，它将显示提示符，并等候用户送入命令。MS-DOS现在已准备好处理业务，SYSINIT模块即被舍弃（图2.5）

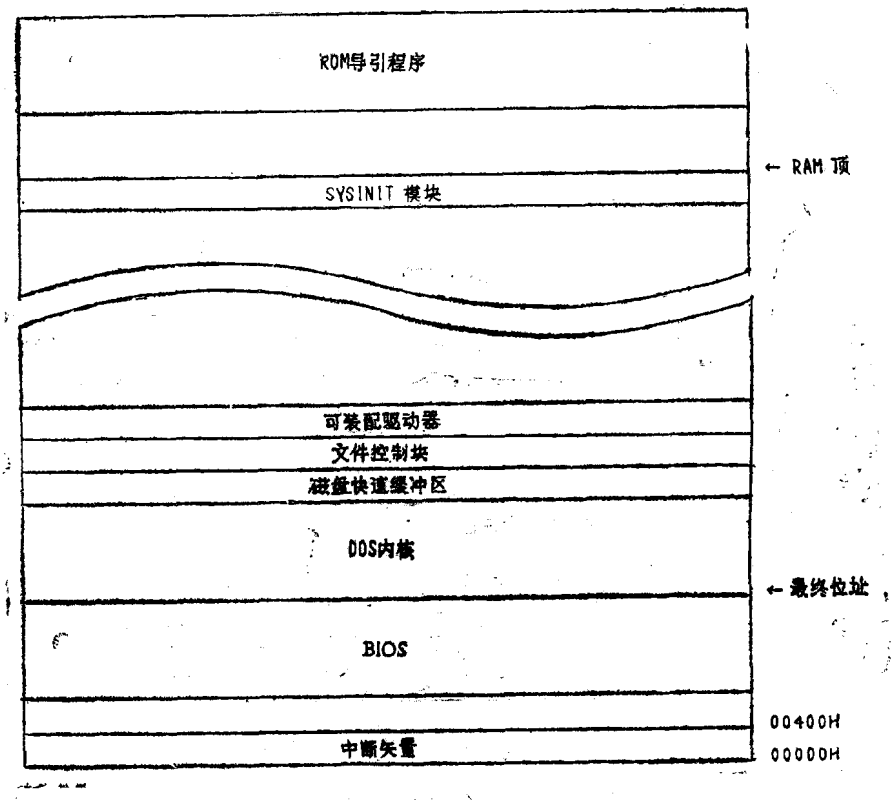

图 2.5 某一典型系统MS-DOS导引过程的最终结果。COMMAND.COM的常驻部分位于内存低部。DOS内核之上。含有批文件处理器和内部命令的暂存部分放在内存高部，可以被在暂存程序区中运行的外部命令和应用程序所覆盖

第三章　在MS-DOS环境下编程

在MS-DOS下运行的程序以两种基本形式出现：一种是COM程序，其最大容量约为64k，一种是EXE程序，其容量可达到全部可用的内存。用Intel 8086的语言来说，COM程序适合于“小模式”，所有段寄存器含有同值；就是说代码与数据混在一起。EXE程序适合“中型”或“大模式”，段寄存器的数值各不相同；就是说代码、数据和堆栈位于不同段内。甚至可以有多重代码和数据段，EXE程序通过长调用来访问代码段，通过对数据段（DS）寄存器的操作访问数据段。

COM型程序在磁盘上以绝对内存映象形式驻留于扩展名为.COM的文件中。文件不具有头段或任何其他识别信息。另一方面，EXE程序以专门的文件类型驻留在磁盘上，带有独一的头段，再定位映象，一个检查和，以及其他被（或可以被）MS-DOS使用的信息。

COM和EXE程序由相同的机制，即所谓EXEC功能，装入内存而执行。EXEC功能构成MS-DOS的装入器。EXEC可以由COMMAND.COM（常规MS-DOS命令解释器）装入的程序文件名所调用，或者由其他外壳或用户接口所调用，或者由EXEC以前装入的另一程序所调用。如在第二章中所讨论的，如果在暂存程序区有足够的自由内存，EXEC将分配一块内存以存放新程序，在其底部建立程序段前缀，然后将程序读入紧接在PSP上方的内存中。最后EXEC设定段寄存器与堆栈，并将控制转交程序。

当被引用时，EXEC可被赋予附加信息的地址，例如命令尾段，文件控制块及环境块，这些信息将被传递到新程序中去。（第十章中将举例讨论如何在自己的程序中使用EXEC的详细步骤）。当程序终止时或者由于被操作系统所抛弃，或者是由于完成任务，有步骤地最后撤出到MS-DOS——内存块将被释放（这就是“暂存”一词的来由），并且可被下一个被装入的程序所使用。

§3.1　程序段前缀

对程序段前缀的彻底了解是在当前的MS-DOS版本下编程是否成功的关键。它是一个保留区，256字节长，由MS-DOS设定在分配给暂存程序的内存区底部。PSP含有一些与MS-DOS的连接信息，可被暂存程序使用，还含有MS-DOS为自身需用而存放的一些数据，以及由MS-DOS传递给暂存程序的信息——根据程序需要可用或不用（图3-1）。在MS-DOS的最初的一些版本中，程序段前缀被设计得与CP／M操作系统下暂存程序区方的一块控制区兼容，因而程序不必在逻辑上大事修改就可转为MS-DOS所用。虽然MS-DOS已经有了大量的演变，仍然可看得出来PSP的结构与CP/M的对应部分相似。熟悉CP/M文件控制块和调用常规的编程人员应当在这里感到自如。

PSP的0000H偏移量含有与MS-DOS过程终止处理程序的连接，它在程序完成作业时将控制完成最终撤出。与此类似，PSP的0005H偏移量含有与MS-DOS功能分派器的

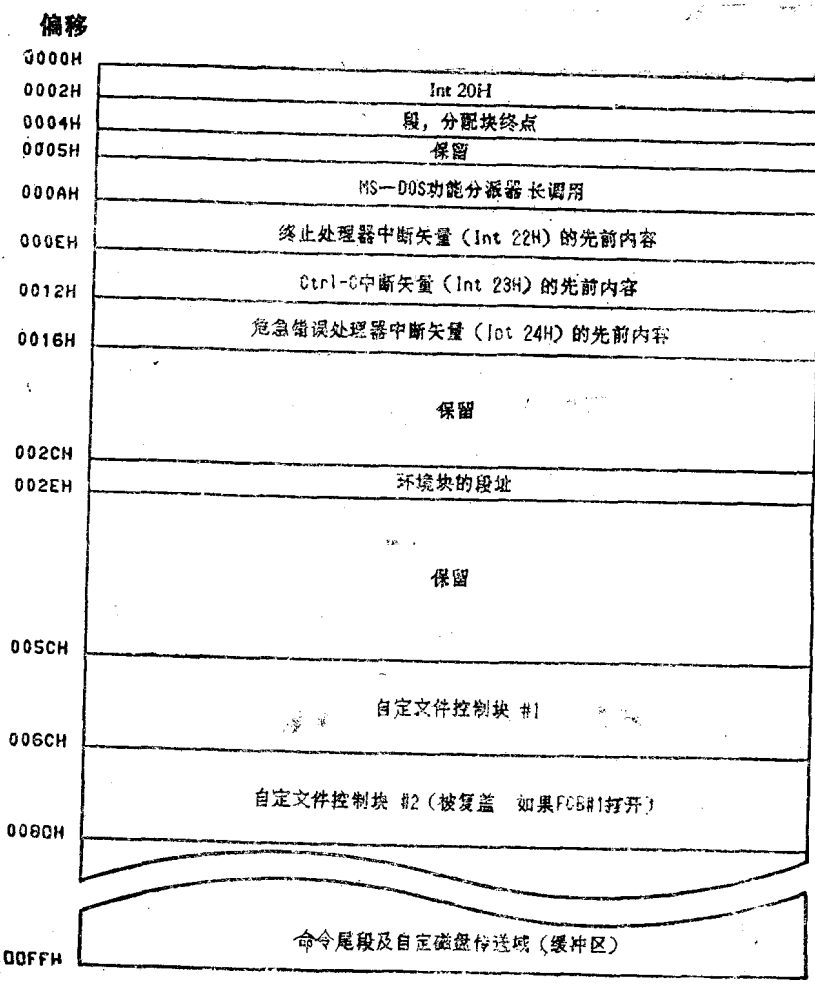

图 3.1 程序段前缀的结构

连接，它完成磁盘操作、控制台输入／输出，以及其他这一类由暂存程序提出的服务。因此，为便于机械地翻译应用程序，调用PSP:0000与PSP:0005与CP/M下CALL0000和CALL00005的效果是一样的（然而这种连接方式不是取得这类服务的“公允”方式）。

PSP偏移量0002H所在的字含有分配给暂存程序的内存块顶部的段地址。程序可利用该值以确定是否需要申请更多的内存来完成任务，或者它的内存有余，可释放出供其他过程使用。

PSP偏移量000AH到0015H含有终止、Ctrl-C及危急错误把柄中断矢量的先前的内容。如果暂存程序为自身目的改变了这些矢量，则当程序最终撤离时，MS-DOS可利用存在PSP中的原始值进行恢复。

PSP偏移量002CH所在字含有环境块的段地址，它含有一串ASCIIZ字符串（一串以零字节终止的ASCII字符）。环境块是由调用EXEC功能（以装入当前执行程序）的程序继承得到的。它包含的信息有诸如COMMAND.COM用以寻找可执行程序的当前搜索路径，COMMAND.COM本身在磁盘上的位置，以及COMMAND.COM所用的

用户提示符格式。

命令尾段——指启用暂存程序的命令行中程序名后面的剩余部分——被拷贝到程序段前缀中，偏移量0081H开始处。命令尾段的长度，不包括最后的回车符，放在偏移量为0080H的字节内。转向或衔接参数及相应文件名不出现在命令行被传送到暂存程序的那一部分（命令尾段）中，因为假定转向对于应用程序是透明的。

为提供与CP/M的兼容性，MS-DOS将命令尾段的前两个参数分解为两个自定的文件控制块（FCB），位于PSP:005CH和PSP:006CH，假定它们可以是文件名。然而，如果参数是包括指定路径的文件名，则在这些自定文件控制块中，只有驱动码是有效的，因为FCB型的文件与记录访问功能并不支持树状文件结构。虽然自定FCB在早年更多地考虑与CP/M的兼容性时是有帮助的，但在近代的必须提供全部路径支持的MS-DOS应用程序中，它们基本上是没有用的。（文件控制块在第六章中详细讨论而树状文件结构在第七章中讨论。）

PSP中0080H到00FFH中128字节区域也用作自定磁盘传送区（DTA），由MS-DOS在将控制转交暂存程序之前设定。如果程序不明确更改DTA，则任何用FCB功能调用提出的文件读写操作将自动使用DTA作为数据缓冲区。这很少有用，也是MS-DOS仅仅为了与CP/M兼容而对PSP作的一种处理。

警告：程序不得改动PSP在偏移量005CH以下的任何部分

§3.2 COM程序

COM型程序存储于磁盘文件中，具有待执行机器码的绝对映象。由于文件不含再定位信息，因此比较简练，装入执行时，要比相应的EXE文件略快一些。注意，MS-DOS并不去核实COM文件是否确实含有可执行码（不象EXE文件那样有标记或检查和）；它只

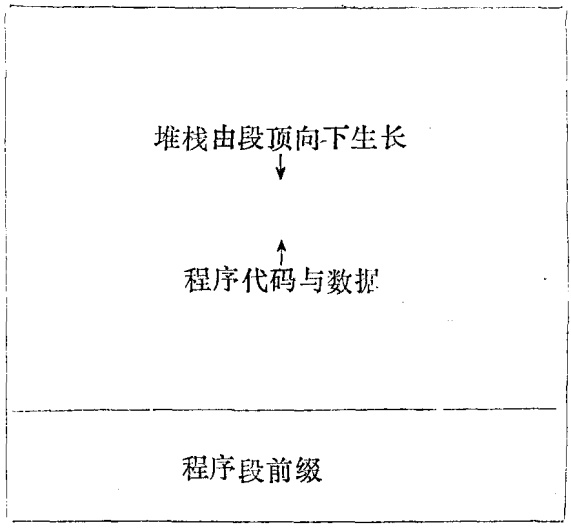

图 3-2 典型COM型程序装入后的内存映象。COM文件的内容被带入内存中程序段前缀的紧接上方。程序、代码和数据混合放置于相同段中，所有段寄存器含有相同数值。

是简单地将任何带有COM扩展名的文件装入内存，并跳转到该文件。

由于COM程序直接装在程序段前缀上方，并且不具有能够指定其他入口点的头段，它们的初始地址永远是0100H，这是程序段前缀的长度。地址0100H必须含有一条可执行指令。COM程序的最大长度为65536字节，减去PSP长度（256字节）以及必不可少的堆栈字（二字节）。

当控制由MS-DOS转移到COM程序时，所有的段寄存器均指向程序段前缀（图3-2）。堆栈指针（SP）寄存器在内存容许时含OFFFEH；否则它被设定到尽可能高的内存位置减去二字节（MS-DOS在进入时在堆栈上压入一个零字）。

虽然可执行COM文件的容量不可能超出64k，MS-DOS的现行版本在装入时将所有暂存程序区分配给COM程序。因为许多这类程序是在MS-DOS早期偏写的，操作系统只作最坏情况打算，并将可供利用的一切都交给COM程序。如果COM程序要使用EXEC功能去启用另一过程，它必须首先将它所占的内存缩小到维持运行的最小量，并注意保护其堆栈。（将在第十章中详细讨论）。

当COM程序执行完毕时，它可以通过几种不同方式将控制返回MS-DOS。可取的方法是Int21H功能4CH，它允许程序送一个返回代码到启用该程序的程序、外壳或批文件。然而，如果程序是在MS-DOS版本1之下运行，它必须经Int20H、Int21H功能，或近程返回（NEAR RETURN）而撤离。（因为进入时在栈顶压入了一个零字，近程返回导致转向PSP:0000，其中含有Int20H指令。）

COM型应用可以由许多分散的目标模块连接而成。所有模块必须使用同一代码段名和类别名(class name)，入口点为段内偏移量0100H的模块必须首先接入。此外，COM程序中的所有过程必须具有NEAR(近程)属性，因为所有可执行代码均位于同一个段内。

当连接COM程序时，连接器（Linker）将显示下列信息：

警告：无堆栈段（Warning:no stack segment）这可以不予理睬。连接器的输出为EXE文件，在执行之前必须用MS-DOS EXE2BIN实用程序转换成COM文件，然后可抹去EXE文件。（第4章中将举例说明这一过程）。

3.2.1 COM程序的例子

图3.3中列出的HELLO.COM程序说明了打算用作COM文件的简单汇编语言程序结构。（将这张清单与本章以后的HELLO.EXE程序比较是有好处的）。因为这个程序过于简短，大部分的源码实际上是汇编指示符，并不产生可执行代码。

第一行的NAME语句只不过是提供在连接过程中使用的模块名。如果源文件中没有NAME命令，则使用TITLE语句中正文的前六个字符作为模块名。如果二者都没有，连接器就将使用文件名。因此NAME语句远非必要；但为养成良好习惯，还是使用为好，不仅有助于整理文件和理解连接器产生的列表，也可以避免模块自定名在以后造成某种麻烦。

PAGE命令，如第二行所示有两个操作数时，规定了页的长度与宽度。自定为66行长，80字符宽。如果使用PAGE命令而无操作数，则向打印机送出换页命令，并打印文件头。在大程序中，可随意使用PAGE命令，以便将各个子程序分放在各页上，便于阅读。