

IBM PC DOS 3.X 版本 技术参考手册

夏东涛 朱芒大 译编 周明德 审校

清华大学出版社

科 目： 159—72

统一书号： 15235·347

定 价： 1.40 元

IBM PC DOS 3.x 版本

技术参考手册

夏东涛 朱芒大 译编

周明德 审校

清华大学出版社

内 容 简 介

本书详细介绍了 IBM PC DOS 的工作原理和体系结构,重点阐述了 PC DOS 3.x 版本的有关技术,详尽列出了该版本的中断调用和系统调用的原理与使用方法。本书是对 IBM PC (0520) 微机进行深层开发的有力工具,尤其适用于 IBM PC (0520) 微机系统软件的开发、汇编语言和 C 语言的编程以及了解 IBM PC (0520) 系统的内部构造。

本书可供广大的应用 IBM PC (0520) 微机的工程技术人员和大专院校师生使用。

IBM PC DOS 3.x 版本 技 术 参 考 手 册

夏东涛 朱芒大 译编

周明德 审校

*

清华大学出版社出版

北京 清华园

北京丰华印刷厂印刷

新华书店北京发行所发行

*

开本: 787×1092 1/32 印张: 6 9/16 字数: 147千字

1987年9月第1版 1987年9月第1次印刷

印数: 00001—35000

统一书号: 15235·347 定价: 1.40元

前 言

随着 IBM PC、PC/XT、PC/AT、长城 0520、浪潮0520 和东海0520以及0530等微型计算机的广泛使用，广大技术人员迫切需要了解和掌握 IBM PC DOS 高版本的内部结构和详细的系统技术资料。

我们以愉快的心情，将最新的 IBM PC DOS 3.x 版本奉献给广大读者，期望它能将蕴藏在 IBM PC一类微机中的潜力开发出来，以便能在国民经济各个领域充分发挥效益。

本书包含有与DOS 2.10, 3.00和3.10版本有关的技术信息。其中某些信息是与 DOS 版本有关的，并不适用于所有版本。

本书内容适合于有经验的 DOS 用户，系统程序设计者以及应用程序开发者阅读。假定读者已熟悉8088体系结构。

在有“与版本有关的信息”一节的各章中，其信息与DOS的版本有关；而在没有此节的各章中，所包含的信息对于DOS2.10, 3.00和3.10版本是通用的。

本书共有十一章：

第一章为关于DOS的一般技术资料。

第二章为关于设备驱动程序的详细资料。

第三章为关于使用扩展屏幕功能和键盘功能以控制光标定位和重新定义键盘上键的作用的详细资料。

第四章介绍正确执行文件管理时的注意事项。

第五章介绍磁盘空间的分配。

第六章介绍系统中断和功能调用。

第七章介绍控制块和工作区，包括内存图，程序段前缀
和文件控制块。

第八章介绍如何从应用程序内执行 DOS 命令。

第九章为有关 DOS 对于硬盘支持的技术资料。

第十章为有关 EXE 文件结构的详细资料。

第十一章为有关 DOS 内存管理的资料。

由于时间仓促和水平有限，书中难免有不妥或错误之处，请读者指正。

夏东涛 朱芒大

一九八七年三月十五日

于北京

目 录

前言	i
第一章 DOS 技术资料	1
第一节 引言	1
第二节 与版本有关的信息	1
第三节 DOS 结构	1
第四节 DOS 初始化	4
第五节 提供使用的 DOS 功能	5
第六节 磁盘传递区 DTA	5
第七节 出错捕获	6
第二章 可安装的设备驱动程序	7
第一节 引言	7
第二节 版本的特殊说明	7
第三节 设备驱动程序的格式	8
第四节 设备的类型	9
第五节 设备标题	10
第六节 建立一个设备驱动程序	13
第七节 安装设备驱动程序	14
第八节 请求标题	17
第九节 设备驱动程序的功能	20
第十节 设备驱动程序实例	33
第三章 扩展屏幕和键盘控制的使用方法	34

第一节	引言	34
第二节	控制序列	34
第三节	控制序列语法	35
第四节	光标控制序列	36
第五节	删除	39
第六节	操作方式	40
第七节	键盘键重定义	41
第四章	文件管理说明	45
第一节	引言	45
第二节	版本专用信息	45
第三节	文件管理功能	45
第四节	FCB 功能调用	46
第五节	句柄功能调用	48
第六节	专用文件的句柄	49
第七节	ASCII 和二进制方式	50
第八节	允许打开文件的个数	53
第九节	FCB 使用的限制	53
第十节	句柄使用的限制	54
第十一节	文件空间的分配	54
第五章	DOS 磁盘分配	56
第一节	引言	56
第二节	版本的特殊说明	56
第三节	DOS 区域	56
第四节	自举记录	57
第五节	DOS 文件分配表 (FAT)	57

第六节	DOS 磁盘目录	60
第七节	数据区	63
第六章	DOS 中断和功能调用	65
第一节	引言	65
第二节	与版本有关的信息	65
第三节	DOS 寄存器	66
第四节	扩展 ASCII 代码	63
第五节	中断	69
第六节	功能调用	83
第七章	DOS 控制块和工作区	177
第一节	引言	177
第二节	DOS 内存图	177
第三节	DOS 程序段	179
第四节	程序段前缀 PSP	182
第五节	文件控制块	183
第八章	在应用程序内执行命令	187
第一节	引言	187
第二节	引用命令处理程序	187
第九章	磁盘信息	189
第一节	引言	189
第二节	硬盘结构	189
第三节	系统初始化	190
第四节	自举记录/分区表	191
第五节	硬盘技术说明	192
第六节	磁盘位置的确定	194

第十章 EXE 文件的结构及其加载	196
第一节 引言	196
第二节 EXE 文件结构	196
第三节 重定位表	197
第十一章 DOS 内存管理	199
第一节 引言	199
第二节 控制块	199

第一章 DOS技术资料

第一节 引言

本章内容有：

- DOS 结构
- DOS初始化
- DOS 功能
- 磁盘传送区 DTA
- 出错捕获

第二节 与版本有关的信息

本章中下面部分是与 DOS 的版本有关的。

对于 DOS 2.10，命令处理程序的暂存覆盖部分包含有子程序 EXEC，该子程序可用于装入和执行外部命令；而对于 DOS 3.00和 DOS3.10 命令处理程序的常驻内存部分包含有子程序 EXEC。

第三节 DOS结构

DOS 由以下四个部分组成：

- 引导记录
- ROMBIOS 接口
- DOS 程序文件(IBM DOS.COM)
- 命令处理程序(COMMAND.COM)

一、引导记录

引导记录放在经 DOS FORMAT 命令格式化后的磁盘的 0 面、0 道、1 扇区。将引导记录安放在磁盘上，则使用非常系统盘在 A 驱动器中启动系统时会产生出错误信息。对于硬盘，引导记录驻留在 DOS 分区的第一个扇区上。每种由 DOS 支持的记录介质 (medium) 都使用了一个扇区存放引导记录。

二、ROMBIOS 接口

隐式文件 IBMBIO.COM 是与 ROMBIOS 的接口模块。IBMBIO.COM 提供了对于 ROMBIOS 设备子程序的低级接口。

三、DOS 程序文件

DOS 程序存放在隐式文件 IBMDOS.COM 中，它对于用户程序提供了高级接口。IBMDOS.COM 由许多子程序组成，包括：文件管理子程序；用于磁盘操作的数据组块/解决子程序；以及各种可由用户程序能方便地访问调用的内部功能。

当用户程序调用这些功能子程序时，这些子程序借助于寄存器和控制块的内容来接收来自高层的信息。如果是要求

设备操作，则子程序将需要翻译成 一个或若干个对于 IBMBIO.COM 的调用来完成该请求。

四、命令处理程序

命令处理程序 COMMAND.COM 由三部分组成：

1. 常驻部分，驻留在内存中，跟在 IBMDOS.COM 及其数据区之后面。这部分包括如下的子程序：中断 22H（终止地址），23H（Ctrl-Break 处理程序），24H（严重错误处理）以及在需要时能够重新装入暂存部分的子程序。

对于 DOS3.00 和 3.10，这部分还包括有一个子程序，能够用于装入和执行一个外部命令；例如带有 .COM 和 .EXE 扩展名的文件。

注：一个程序执行结束后，使用检查和(Checksum)方法确定该程序是否使暂存部分被覆盖掉了，如果暂存部分被覆盖掉了，就需要重新装入。

所有标准的 DOS 错误在 COMMAN.COM 的这部分内进行处理。这包括有：显示指错信息，解释对于 Abort, Retry or Ignore 的答复。

2. 在常驻部分后面便是初始化部分，此部分控制程序的建立启动过程。它包含有 Autoexec.bat 文件处理子程序。初始化部分决定了程序可被装入的段地址。由于运行过程中不需要此部分，因此该部分被 COMMAND.COM 装入的第一个程序所覆盖掉。

3. 暂存部分装入在内存的高端，它便是命令处理程序本身，包含有所有的内部命令的处理程序和一个批文件处理程序。对于 DOS2.10，此部分还包含有一个用于装入和执行

外部命令（例如带有.COM和.EXE扩展名的文件）的子程序。

COMMAND.COM的这一部分还产生DOS提示符（例如A>），从键盘上（或批文件中）读取命令并执行该命令。如果是外部命令，这部分建立一个命令行，并发出EXEC功能调用装入及将控制传送给该程序。

第六章将详细介绍在各种条件下由EXEC将控制交给一个程序及其结果。

第四节 DOS初始化

系统复位或开电源均引起DOS的初始化。ROMBIOS首先检查驱动器A的引导记录。如果没找到引导记录，ROMBIOS搜索硬盘的当前活动分区，如果还是没找到，ROMBIOS便调用ROMBASIC。在系统初始化过程中产生下列动作：

1. 将引导记录读入内存并将控制交给它。
2. 引导记录检查根目录以确定头两个文件是IBMBIO.COM和IBMDOS.COM。这两个文件必须是头两个文件，并且顺序必须是IBMBIO.COM在前，IBMDOS.COM在后，且扇区还要连续安排。
3. 引导记录将IBMBIO.COM装入内存。
4. IBMBIO.COM中的初始化代码将IBMDOS.COM装入内存，确定设备状态，将磁盘系统复位，初始化附加的装置，装入可安装的设备的驱动程序，建立低编号的中断向量，将IBMDOS.COM向下定位，然后调用DOS的第一个

字节。

5. DOS 初始化自己的内部工作表，初始化中断 20H 到 27H 的中断向量，将 COMMAND.COM 的程序段前缀建立在最低的可用段处。对于 DOS3.10 版本，DOS 初始化中断 0FH 到 3FH 的中断向量。

6. IBMBIO.COM 使用 EXEC 功能调用装入并启动最高级别的命令处理程序，默认的命令处理程序是 COMMAND.COM。

第五节 提供使用的DOS功能

DOS 为用户程序提供了大量的功能，它们均通过发生中断和功能调用来使用。有许多的子程序，可用于：键盘输入（带或不带回送信号和 Ctrl-Break 检测）；控制台显示和打印机输出；构造文件控制块；内存管理；日期和时间功能；各种磁盘、目录和文件处理功能等。

DOS 提供了两种类型的可用于文件管理的功能调用，它们是：

- 利用文件控制块 FCB 功能调用
- 扩展的（句柄）功能调用

第四章介绍 FCB 和句柄功能调用。第六章给出了每个调用的细节。

第六节 磁盘传送区DTA

DOS 使用内存中的一个区域来容纳由 FCB 功能调用完

成的文件读、写的数。这个内存区域称作磁盘传送区 DTA。磁盘传送区也被称为缓冲区(buffer)。这个区域可处于用户应用程序数据区中的任意位置,并应由用户程序来建立。

一次只有一个 DTA 是有效的,因此用户应用程序在使用磁盘读、写功能之前必须通知 DOS 所使用的内存位置。使用功能调用 1AH (建立磁盘传送区地址) 来建立磁盘传送区地址。使用功能调用 2FH (得到磁盘传送区地址) 来获取磁盘传送区地址。一经建立后, DOS 便一直使用该区域用于所有的磁盘操作直到发出新的功能调用 1AH 定义了新的 DTA。当 COMMAND.COM 将控制交给一个应用程序时,一个足以容纳 128 个字节的默认的 DTA 建立在应用程序的程序段前缀的 80H 处。

对于扩展功能调用执行的文件读、写操作,没有必要去建立 DTA 地址。而是在用户发出读、写调用时规定一个缓冲区地址。

第七节 出错捕获

DOS 提供了一种方法,使得当发生磁盘或其它设备读/写错误,或检测到文件分配表 FAT 的内存映象出错时,使一个程序接收控制。即这些错误发生时, DOS 执行中断 24H (严重错误处理程序中中断向量),将控制传送给错误处理程序。默认的错误处理程序驻留在 COMMAND.COM 中,但任何用户应用程序可由重新建立中断 24H 的向量值以指向新的错误处理程序。DOS 使用寄存器提供出错信息,并由返回码来提供 Abort, Retry, or Ignore 选择。

第二章 可安装的设备驱动程序

第一节 引言

本章向你介绍：

- 怎样格式化一个设备驱动程序。
- 怎样建立一个设备驱动程序。
- 怎样安装一个设备驱动程序。

本章还介绍各种类型的设备驱动程序、访问标识和时钟(CLOCK\$)设备。

DOS设备接口和设备驱动程序是用键连接在一起的。这样便于增加新的设备驱动程序和选择设备驱动程序与DOS的连接。

第二节 版本的特殊说明

在本章中下面的内容仅适用于DOS某一种版本：

属性字段(Attribute Field)：第11位(可移介质)仅适用于DOS3.00和DOS3.10版本。

命令字段(Command Code Field)：命令代码字段的值13、14和15仅适用于DOS3.00和DOS3.10版本。

状态字字段(Status Word Field): 只有在用 DOS 3.00 和DOS3.10版本时, 才能返回指错代码 0DH、0EH和0FH。

设备驱动程序的功能:

- DOS3.00和3.10版本支持可移动介质。
- 如果你正在使用DOS3.00和3.10版本, 介质检验设备驱动程序在遇到问题时能给出“Error”(出错)以示警告; 同样, 对于DOS3.00和3.10版本, 如果更换磁盘, 介质检验设备驱动程序能使DWORD指针指向卷标识符。

- DOS 3.00和3.10版本有介质描述符, 对于 $5\frac{1}{4}$ 英寸15

扇区的介质来说, 描述符是F9H字节。

- 对于DOS 3.00和3.10版本, 如果没有更换磁盘, 输入/输出设备驱动程序也能返回一个指向卷标识符的 DWORD 指针。

- 打开或关闭设备驱动程序能够用于DOS 3.00 和 3.10 版本。

- 可移介质设备驱动程序能够用于DOS 3.00和 3.10 版本。

第三节 设备驱动程序的格式

设备驱动程序是一种包含了所有用于驱动设备所需代码的内存映象文件或EXE文件。这种文件在开头有一个特殊的头标, 用来表示这种文件是设备驱动程序。该文件定义了使用的决策和中断入口点, 以及设备的各种属性。

注意: 上面所述的设备驱动程序内存映象文件不能使用

OGR100H, 因其没有用程序段前缀, 设备驱动程序仅是简单装入的。因而, 内存映象文件必须有零(0)起始地址 (ORG0或没有ORG语句)。

第四节 设备的类型

基本的设备类型有两种:

- 字符型设备
- 块型设备

一、字符型设备

字符型设备设计成用CON、AUX和PRN等串行方式进行字符输入/输出操作的设备。这些字符型设备都有类似CON、AUX、CLOCK\$这样的名字, 你能够打开通道(通过句柄或FCB)对这些设备进行输入和输出。如果字符型设备只有一个名字, 那末这些设备只能支持一种设备。

二、块型设备

块型设备是指系统中的“硬盘或磁盘驱动器”。这些设备可以一段一段地随机进行输入/输出操作, 这些段称为“块”。通常块的大小与磁盘上物理扇区一样。块型设备不能象字符型设备那样命名, 也不能直接打开, 只能通过使用驱动器字母A、B、C等来映射块型设备。块型设备可以映射整个设备。换句话说, 一个块驱动程序可以响应一个或多个磁盘或软盘驱动器。例如, 第一个块型设备驱动程序能够响应驱动器A、B、C和D, 其意味着, 第一个块型设备驱动程序有预

先定义四个设备，因此有四个驱动字母。在所有驱动程序的链中驱动程序的位决定了驱动器和相应的驱动字母的对应关系。例如，如果设备驱动程序在设备链中是第一个块驱动程序，并且定义有四个设备，则那些设备A、B、C和D，如果第二个块驱动程序定义有三个设备，则那些设备是E、F和G。最多可以用26个字母A~Z定义26个设备。

第五节 设备标题

设备驱动程序要求在文件的开头有一个设备标题。下表给出了设备标题的内容：

字 段	长 度
指向下一个设备标题的指针	DWORD
属性	WORD
指向设备决策例行程序的指针	WORD
指向设备中断例行程序的指针	WORD
名称/单元字段	8字节

一、指向下一个设备标题字段的指针

设备标题字段是指向下一个设备驱动程序的设备标题的指针，设备标题字段是双字段，它是在设备驱动程序被装入后由DOS提供的。第一个字是偏移量（称为区距），第二个字是段。

如果你正在装一个设备驱动程序，则将设备标题置为

-1, 然后再装入设备。如果你正在装一个以上（即多个）的设备驱动程序，则将设备标题字段的第一个字定为下一个设备驱动程序标题的偏移量。给最后一个设备驱动程序的设备标题字段置-1。

二、属性字段

属性字段是在系统中描述设备驱动程序属性的字段。属性如下：

位15 = 1	字符型设备
= 0	块型设备
位14 = 1	支持IOCTL
= 0	不支持IOCTL
位13 = 1	非IBM格式（仅指块型设备）
= 0	IBM格式
位11 = 1	支持可移动介质
= 0	不支持可移动介质
位10~4 = 0	这些位必须是关闭状态，因为它们 依靠DOS系统来保存
位 3 = 1	现行时钟设备
0	非现行时钟设备
位 2 = 1	现行NUL设备
= 0	非现行NUL设备
位 1 = 1	现行标准输出设备
= 0	非现行标准输出设备
位 0 = 1	现行标准输入设备
	非现行标准输入设备

1. 位15

位15是设备类型位。使用位15告诉系统设备驱动程序是块型设备还是字符型设备。

2. 位14

位14是IOCTL位。它既用于字符型设备也用于块型设备。使用位14告诉DOS系统设备驱动程序是否能通过IOCTL功能调用处理控制字符串。

如果设备驱动程序不能处理控制字符串，则它必须置“0”。此时，如果试图通过IOCTL功能调用来对设备发送或接收控制字符串，则DOS就能给出一个错误信息。如果设备驱动程序能够处理控制字符串，则应将位14置“1”，这样，DOS就能调用IOCTL输入和输出功能来发送和接收IOCTL字符串。

IOCTL功能允许在非实际做正常读写操作的情况下从设备接收或发送数据，这样，该设备驱动程序可按自己的目的来使用数据（例如，设定传输率或停止位、改变格式长度等等）。可由IOCTL来解释设备传给它的信息，但此信息必不可看作是正常的I/O要求。

3. 位13

位13是非IBM格式化位。它只能用于块型设备。该位影响GetBPB(BIOS Parameter Block)设备调用的操作。

4. 位11

位11是打开/关闭可移介质位。使用位11告诉DOS设备驱动程序是否能够处理可移介质。

5. 位3

位3是时钟设备位。它只能用于字符型设备。使用位3告

诉DOS字符型设备驱动程序是否是新的CLOCK\$设备。

6. 位2

位2是NUL属性位，它只能用于字符型设备。使用位2告诉DOS字符型设备驱动程序是否是NUL型设备。尽管存在NUL型设备的属性位，但用户并不能再分配NUL型设备。本属性位是为DOS系统准备的，目的是为了告诉DOS NUL型设备是否处于使用状态。

7. 位0和1

位0和1是标准输入/输出位。它们只能用于字符型设备。使用这些位告诉DOS字符型设备驱动程序是否是新的标准输入或输出设备。

三、指向设备决策和中断程序的指针

这两个字段是决策和中断程序入口点的指针。它们是字值，它们必定与设备标题在同一段中。

四、名称/单元字段

这是一个包含字符型设备的名字或块型设备的单元的8字节字段。对于字符型设备，名称以左对齐，余下部分用空白填满8个字节。对于块型设备，单元数能够放在第一个字节中。因为DOS可以通过驱动程序INIT代码的返回值填入此单元，所以本字段内容是任选的。

第六节 建立一个设备驱动程序

为了建立一个DOS能够安装的设备驱动程序，按下列步

骤执行：

- 用在文件开头的设备标题建立一个内存映象文件。
- 在地址0而不是100H产生代码（包括设备标题代码）。
- 设置下一个设备标题字段。详见“指向下一个设备标题字段”。
- 设置设备标题的属性字段。详见“属性字段”。
- 设置中断和决策例行程序入口点。
- 用字符型设备的名字或块型设备的单元数填入名称/单元字段。

DOS系统总是在处理系统默认设备以前来处理可安装的字符型设备驱动程序。所以为了安装一个新的CON设备，只要简单地把设备定名为CON。但是在新的CON设备的属性字段中一定要建立标准输入和输出设备位，设备表的扫描在第一次匹配时就停止，因此可安装设备的驱动程序能获得优先权。

注：由于DOS系统能在内存任何地方安装驱动程序，因此须认真理解FAR内存的使用。你不能期望驱动程序总能在每个时刻在同一地方被写入。

第七节 安装设备驱动程序

DNS在自举时依靠读取和处理CONFIG文件中的DEVICE命令来动态地安装新的设备驱动程序。例如，如果你已写入了一个名叫DRIVER1的设备驱动程序，则通过CONFIG·SYS文件中的DEVICE命令就可以安装设备驱动程序。

device = driver1

通过访问标题和DOS要设备驱动程序做什么的信息描述，调用设备驱动程序到预定入口点。

决策程序不是执行上述请求，而是将请求排队等候，或是保存请求标题的指针。第二个入口点是中断程序，在决策程序返回之后，立即由DOS调用。调用中断程序不需要任何参数，而是根据排队请求执行操作和建立返回信息。

DOS传送指针至在ES:BX中的请求标题。这一种指针的结构是由一个定长标题（请求标题）和紧跟着的要执行操作的有关数据组成。

注意：这时设备驱动程序响应以保存机器状态。例如，可以在进入时保存所有寄存器的状态，并在出口处恢复。

DOS使用的堆栈具有足够的空间保存所有的寄存器中的数据。如果需要更多的堆栈空间，设备驱动程序重新分配和修改另一个堆栈。

所有对设备驱动程序的调用都是FAR调用。为返回DOS，必须执行FAR返回。

一、安装字符型设备

对于每一个设备所定义的功能中有一个功能是INIT(初始化)。初始化程序仅当设备被安装的时候调用一次，然后不再调用了。INIT程序返回过程如下：

- 在内存中设备驱动程序之后的第一个空字节单元，就象在结束地址字段中存入结束和驻留命令。这样，为节省空间，初始化代码被使用过一次以后就可以从内存中清除掉。

- 在建立结束地址字段后，字符型设备驱动程序就能够

设置状态字并返回。

二、安装块型设备

块型设备的安装方法与字符型设备相同，不同的是，块型设备安装时会产生附加的信息。块型设备安装时返回如下信息：

- 块型设备的单元数。该数决定设备将具有的逻辑名称。例如，如果在安装调用时现行最大的逻辑设备名称字母是F，并且块型设备驱动程序INIT占用三个逻辑单元，则这些设备的逻辑名称为G、H和I。映象过程是由设备表中的驱动程序的位置和设备的单元数所决定的。由INIT返回的单元数超越设备标题的名称/单元字段的值。

- 指向BPB(BIOS参数块)的指针组。这是一个指向n个字的指针组的指针，这里的n是所定义的单元数。这些字指针指向BPB指针组。这样，如果所有的单元相同，则为节省空间，所有指针组能够指向同一个BPB。BPB包含了与设备有关的信息，诸如扇区大小、每个分配单元的扇区数、等等。BPB扇区的大小不能大于DOS初始化时所分配的最大容量。

注意：该指针组必须在由返回设置的空指针之后保护起来。

- 介质描述符字节。该字节在执行过程中传送给设备，以便了解当使用实际的驱动单元时DOS使用什么样的参数。

块型设备可以有n种方式。它们可以是迟钝(dumb)或敏捷(smart)方式。迟钝块型设备为每个可能的介质驱动组合定义一个单元(即定义BPB)。这种情况下，单元0=驱动

器0单面盘；单元1 = 驱动器0双面盘等等。对于这种方式，介质描述符字节意味着虚设。敏捷块型设备允许每个单元有多个介质，在这种情况下，在初始化时返回的BPB表必须定义有足够大的空间，以便能安置可能的最大的介质使用空间（这时BPB中的扇区大小一定要与DOS当前使用的扇区大小一样）。敏捷驱动程序将用“介质字节”来传递有关当前在单元中有何种介质的信息。

第八节 请求标题

请求标题能传递描述DOS要设备驱动程序去做什么事情的信息。

字 段	长 度
请求标题字节加请求标题末端的数据的长度	字 节
用较小设备操作的子单元对于字符型设备可以是无意义的	字 节
命令代码	字 节
状态	字
DOS保留区域	8-字节
数据相应于操作	可 变

一、单元代码字段

单元代码字段能识别在块型设备驱动程序中哪一个单元被请求。例如，如果一个块型驱动程序有三个已定义的单元，则单元代码字段的可能值就是0、1和2。

二、命令代码字段

请求标题的命令代码字段可以具有下列数值:

代码	功 能
0	初始化
1	介质检验 (仅适用于块型设备, 对字符型设备不操作)
2	建立 BPB(仅适用于块型设备, 对字符型设备不操作)
3	IOCTL输入 (仅当IOCTL位是1才调用)
4	输入 (续)
5	不等待非破坏性输入 (仅适用于字符型设备)
6	输入状态 (仅适用于字符型设备)
7	输入清除 (仅适用于字符型设备)
8	输出 (写)
9	检验输出 (写)
10	输出状态 (仅适用于字符型设备)
11	输出清除 (仅适用于字符型设备)
12	IOCTL输出 (仅当IOCTL位是1才调用)
13	设备打开(仅当打开/关闭/RM置位才被调用)
14	设备关闭 (仅当打开/关闭/移动置位才被调用)
15	可移动的介质 (仅当打开/关闭/移动置位和设备是块型时才被调用)

注: 命令代码13、14和15仅适用于DOS版本3.00和

3.10。

三、状态字段

请求标题的状态字段包含如下内容：

15	14—10	9	8	7—0
指错位	保留位	忙位	完成位	指错代码 (在15位为1时)

状态字字段在进入时是0，在返回时由驱动中断程序设置。

位15是指错位。如果该位被设置，则状态字（7—0）的低8位就是指错代码。

位14—10是保留位。

位9是忙位。由状态调用和可移动介质调用设置该位。有关详细情况参看本章“状态”和“可移动介质”的说明。

位8是完成位。若置位，则意味着完成操作。当驱动程序退出时在该位置1。

位7—0是状态字的低8位。若位15置位，则位7~0包含指错代码，这些指错代码和指错内容是：

指错代码	说 明
00	违反写保护规定
01	未知单元
02	设备没有准备好
03	未知命令
04	CRC错误
05	设备请求结构长度有错
06	查找错误
07	未知介质
08	找不到扇区

续表

指错代码	说 明
09	打印机无纸
0A	写错
0B	读错
0C	一般性失效
0D	保留
0E	保留
0F	无效的磁盘更换

第九节 设备驱动程序的功能

所有决策程序都是用ES:BX 指向请求标题来调用。中断程序从存储的决策程序中获得指向请求标题的指针。请求标题的命令代码告诉驱动程序执行哪个功能。

注：所有的DWORD 指针都有偏移量和段，存储时偏移量在前。

功能调用参数说明如下：

- INIT (初始化)
- MEDIA CHECK (介质检查)
- BUILD BPB (BIOS Parameter Block) (建立BPB(BIOS参数块))
 - MEDIA DESCRIPTOR BYTE (介质描述符字节)
 - INPUT or OUTPUT (输入或输出)
 - NONDESTRUCTIVE INPUT NO WAIT (不等待非破坏性输入)
- STATUS (状态)
- FLUSH (清除)

- OPEN or CLOSE (打开或关闭)
- REMOVABLE MEDIA (可移动介质)

一、初始化 (INIT)

命令代码 = 0

ES:BX

字 段	长 度
请求标题	13-字节
单元数(并非由字符型设备设置)	字 节
常驻程序代码末端地址	DWORD
指向 BPB 指针组的指针(并非由字符型设备设置)/指向自变量余项的指针	DWORD
对于 DOS 3.10 版本, 该字段包含驱动器个数	字 节

驱动程序必须做下列事情:

- 设置单元数(仅适用于块型设备)
- 将指针指向BPB指针组(仅适用于块型设备)
- 执行初始化代码(对于调制解调器、打印机等等而言)
- 设置常驻程序代码的末端地址
- 设置请求标题中的状态字

为了在初始化时使设备驱动程序获得从 CONFIG·SYS 传送来的信息,BPB指针字段指向一个缓冲器,它包含了在CONFIG·SYS中紧跟在一个等号“=”之后的信息。在文件说明之后的初始化时,DOS传送给驱动程序的缓冲器含有用于打开文件的ASCII码字符串。用一个回车符(0DH)和一个换行符(0AH)来结束ASCII码字符串(以0H结尾)。如果文件说明之后没有变量信息,则文件说明之后紧跟一个换行符(0AH)。

该信息是只读信息，且此信息只能在系统调用 01H—0CH 和 30H 时由驱动程序的初始化代码发出。

最后一个字节参数包含了块型驱动器字符作为块驱动程序的第一个单元，例如，0 = A，1 = B 等等。

如果初始化程序决定不能设置设备和在不使用内存的情况下中止，可按下列步骤进行：

- 将单元数置 0
- 将末端地址偏移量置 0
- 将末端地址字段置于码段 (CS)

注：如果在单个内存映象文件中有多个设备驱动程序，则由最后一个初始化调用所返回的末端地址为一个 DOS 所使用。故建议在单个内存映象文件中的所有设备驱动程序返回同一个末端地址。

二、介质检查 (MEDIA CHECK)

命令代码 = 1

ES: BX

字 段	长 度
请求标题	13-字节
从DOS送来的介质描述符	字 节
返回	字 节
如果使用 DOS 3.00 或 3.10 版本，则该调用返回一个指向前一个卷标识符的指针（如果位 11 = 1，要更换磁盘）	DWORD

当命令代码字段是 1 时，DOS 为驱动单元调用介质检查程序并给出当前介质描述符字节。有关该字节的说明详见本

章后面“介质描述符字节”部分。介质检查程序能给出下列之一的信息：

- 未改变介质
- 改变介质
- 不确定
- 指错代码

驱动程序必须执行下列功能：

- 在请求标题中设置状态字
- 设置返回字节

-1：介质已经改变

0：不知道介质是否改变

1：介质尚未改变

对于DOS 3.00和3.10版本：如果驱动程序将设备标题属性字的可移动介质位11置1，且驱动程序返回-1，则驱动程序必须将DWORD指针指向前一个卷标识字段(ID)符。如果DOS认为介质改变有错，则DOS为设备给出一个指错代码0FH(磁盘改变无效)。如果驱动程序不执行给出的卷标识符，而是将位11置1，则驱动程序应将指针指向字符串“NO NAME”，即0。

三、介质描述符字节 (Media Descriptor Byte)

假定当前的介质描述符字节已经被定义成几种介质类型。如果设备驱动程序断开了非IBM格式位，则该字节应与介质字节相同。介质描述符字节的值定义如下：

介质描述符字节 →

1	1	1	1	1	×	×	×
7	6	5	4	3	2	1	0

位 意义

0 1 = 2面, 0 = 非两面

1 1 = 8扇区, 0 = 非8扇区

2 1 = 可移动, 0 = 不可移动

3~7 必须置1

DOS介质描述符字节实例:

磁盘类型	井面	井扇区/磁道	介质描述符
硬盘	—	—	F8H
5 ¹ / ₄ 英寸	2	15	F9H
5 ¹ / ₄ 英寸	1	9	FCH
5 ¹ / ₄ 英寸	2	9	FDH
5 ¹ / ₄ 英寸	1	8	FEH
5 ¹ / ₄ 英寸	2	8	FFH
8英寸	1	26	FEH
8英寸	2	26	FDH
8英寸	2	8	FEH

注: 在8英寸磁盘栏目中有两栏的介质描述符是相同的(均为FEH), 这并非是误印。为了确定所用的磁盘是单面还是双面, 不妨试读一下第二面, 如发生错误, 则可确认磁盘是单面。

对于8英寸软盘:

FEH(IBM3740格式)——单面单密度, 每个扇区128字节, 软划分扇区, 每单元4个扇区, 1个保留区, 2个FAT, 68

个目录项，77×26个扇区。

FDH(IBM3740格式)——双面单密度，每个扇区128字节，软划分扇区，每个单元4个扇区，4个保留区，2个FAT，68个目录项，77×26×2个扇区。

FEH——双面双密度，每个扇区1024字节，软划分扇区，每个单元1个扇区，1个保留区，2个FAT，192个目录项，77×8×2个扇区。

四、建立BPB(BIOS参数块)(BUILD BPB (BIOS Parameter Block))

命令代码 = 2

ES:BX

字 段	长 度
请求标题	13-字节
DOS介质描述符	字节
转移地址(缓冲器地址)	DWORD
指向BPB表指针	DWORD

DOS在下列两个条件下调用 BUILD BPB:

- 如果返回“Media Changed”(介质已改变)
- 如果返回“Not Sure”(不确定)，没有使用缓冲器。用过的缓冲器中有尚未写入磁盘的更改过的数据。

驱动程序必须执行下列步骤:

- 设置指向BPB的指针
- 在请求标题中设置状态字

驱动程序必须确定当前在驱动器中的介质是否完好，以

便返回指向BPB表的指针。由在设备标题的属性字段内的非IBM格式位决定缓冲器(由DOS传送指针)的使用方法。如果位13=0(设备与IBM格式兼容),则缓冲器包含了FAT(最重要的是FAT ID字节)的第一扇区。在这种情况下,驱动程序不能更换该缓冲器。如果位13=1,则该缓冲器是一个可以用于存储任何信息的扇区暂存区域。

对于支持卷识别和磁盘更换的驱动程序来说,这项调用应产生一个从磁盘读出的新卷标识符。这项调用表示磁盘已被合法地更改。

如果该设备与IBM格式是兼容的,则下述事实一定成立,即对于所有可能的介质来说,第一个FAT的第一个扇区均被分配在同一扇区中。这是因为在介质被真正确认之前,FAT扇区已被读取。

对于一种具体介质来说,与BPB有关的信息保存在该介质的引导扇区内,需要强调的是,该扇区的引导格式是:

对于 DOS2.10 版本,为 3 字节的段内转移指令 JUMP(E9H),或对于 DOS3.00 和 3.10 版本,为 2 字节的短转移令 JUMP(EBH)紧跟着一个 NOP(90H)指令。

8 字节 OEM 名称和版本

每扇区字的字节数

每个分配单元的字节扇区数(必须是 2 次方幂)

字保留扇区数(始于逻辑扇区 0)

FAT 的字节数

根目录项字数(允许的最大值)

在逻辑映象中扇区的字数（介值的总扇区数，包括引导扇区目、录扇区等等

介值描述符字节

单个 FAT 所占据的扇区字数

每磁道的字扇区数

开头字数

隐藏扇区字数

上表中最后三行是用来帮助驱动程序了解介质用的。开头的字数对于存储容量相同、表面数量不同的各种各样功能的驱动程序是有用的。保密扇区数对于支持驱动程序分配方案是有用的。

五、输入/输出(INPUT or OUTPUT)

命令代码 = 3、4、8、9和12

ES:BX

字 段	长 度
请求标题	13-字节
介质描述符字节	字节
传送地址（缓冲器地址）	DWORD
字节/扇区计数	WORD
起始扇区数（对于字符型设备无意义）	WORD
对于 DOS3.00和3.10版本，如果返回指错代码 0FH，指针指向卷标识符	DWORD

驱动程序必须执行下列步骤：

- 在请求标题中设置状态字

- 执行请求功能
- 设置传送扇区(或字节)的实际数

注：对于IOCTL I/O调用不能进行错误检查。然而，驱动程序必须设置返回扇区(字节)数为实际传送的字节数。

以下内容适用于块型设备驱动程序：在某一特定环境中，可以要求设备驱动程序执行一项64K字节的写操作，这似乎是在设备驱动请求包中转移地址的卷绕。出现这种情况是由于增加了DOS写代码的优化所致。这仅仅发生在写操作时，这时写操作是在64K字节的一个扇区内，主要是对于那些正在越过当前文件的末端而扩展的文件而言的。如果选择卷绕处理，则设备驱动程序可允许不考虑写操作的平衡。例如，10000H字节的写操作具有 $\times \times \times \times : 1$ 转移地址最后两个字节忽略不计的扇区。

记住：一个用了DOS功能调用的程序，决不能请求超过FFFFH字节的输入输出操作。因此，在传送(缓冲器)段中不能发生环绕处理。由于这个原因，可以忽略在传送段中发生环绕的字节。

如果驱动程序送回错误代码0FH(磁盘改变无效)，必须将DWORD指针指向一个正确的卷识别符的ASCII码字符串以便要求用户重新插入磁盘。

DOS 3.00和3.10版本：在磁盘上的打开的文件的参考计数(由“开”和“关”调用语句保留)允许驱动程序决定何时返回指错代码0FH。如果没有打开文件(即参考计数=0)，且磁盘已更换，则I/O是完全正确的，不会返回指错代码0FH。如果有打开的文件(即参考计数>0)且磁盘已经改变，则指错代码0FH可能退出。

六、不等待非破坏性输入 (NON DESTRUCTIVE INPUT NO WAIT)

命令代码 = 5

ES: BX

字 段	长 度
请求标题	13-字节
从设备读取	字节

驱动程序必须执行如下步骤:

- 从设备取回一个字节
- 在请求标题中设置状态字

如果字符型设备返回忙位 = 0 (在缓冲器中有字符), 则将读取下一个字符返回。该字符不是从输入缓冲器中取出的 (因此称为非破坏性输入)。该调用允许DOS向前寻找一个输入字符。

七、状态 (STATUS)

命令代码 = 6和10

ES: BX

字 段	长 度
请求标题	13-字节

驱动程序必须执行如下步骤:

- 完成所请求的功能
- 设置忙位

• 在请求标题中设置状态字

忙位的设置步骤如下：

对于字符型设备的输出——若忙位返回1，则写请求等待当前请求完成后才执行；若忙位是0，则表示没有当前请求，因此，如有写请求，应立即启动。

对于具有缓冲器的字符型设备的输入——若忙位返回1，则读请求就进入物理设备；若忙位是0，则表示在设备缓冲器中有数个字符，能很快返回一个读请求，这也表示用户已经用键盘输入了信息。DOS已假定所有字符型设备都具有一种超前输入缓冲器。不具有这种缓冲器的设备总是使忙位=0。因此，DOS不会挂起等待实际上并不存在的缓冲器输入的信息。

八、清除(FLUSH)

命令代码 = 7和11

ES:BX

字 段	长 度
请求标题	13-字节

该调用告诉驱动程序清除(结束)所有挂起来的已知道的请求。它的主要用途是清除在字符型设备中的输入队列。

驱动程序必须执行：

在返回请求标题中设置状态字。

九、打开或关闭 (DOS3.00和DOS3.10)(OPEN or CLOSE(DOS3.00和3.10))

命令代码 = 13和14

ES: BX

字 段	长 度
静态请求标题	13-字节

所设计的这些调用是为了属性字的位11置位的条件，给出有关当前文件在设备中活动的设备信息。对于块型设备，这些调用可以用来管理局部缓冲过程。这种设备可以保留一个参考计数，每一次打开都使设备参考计数增量，每一次关闭都使设备参考计数减量。当参考计数为0时，则表示在设备中没有打开的文件。因此，该设备应该清除缓冲器中已写入的内容，因为此时用户可以改变在一个可移动介质的驱动器中的介质。如果介质已更换，最好不清除缓冲器而将参考计数重新置0。这可以视为“最终关闭产生清除”。这些调用对字符型设备更有用。“打开”调用可以用来向一个设备发送初始化学字符串。对于打印机，这样可以发送一个有待输入的规定了字型及打印宽度等等的字符串。因此，打印机在一系列I/O指令启动下总是处于准备状态。类似上述情况，“关闭”调用能够用来在一系列I/O指令结束时发送一个结束字符串（类似输入一个格式符）。使用IOCTL来建立预定和结束的字符串，向I/O设备控制系列提供一种灵活的机构。

注：由于所有处理已包含在程序STDIN、STDOUT、STDERR、STDAUX和STDPRN(句柄0、1、2、3、4)中，所以CON、AUX和PRN设备总是处在打开状态。

十、可移动介质(DOS 3.00和3.10版本)(REMMO-V

ABLE MEDIA (DOS 3.00 and 3.10)

命令代码 = 15

ES: BX

字 段	长 度
静态请求标题	13-字节

为了使用该调用，将属性字段位11置1。对于块型设备来说，只能通过IOCTL功能调用(44H)的子功能来使用该调用。该调用是非常有用的，因为它能使实用程序判断正在使用的驱动器是可移动介质驱动器还是不可移动介质驱动器。例如，FORMAT实用程序需要知道驱动器是可移动介质的还是不可移动介质的，因为需要显示不同形式的提示符。

在状态字处于忙位情况下，就能返回信息。如果忙位是1，介质是不可移动的；如果忙位是0，介质是可移动的。

注：无须执行错误位检验，因为已假定了调用总是正确的。

十一、时钟设备(The CLOCK\$ Device)

通常情况下，系统中有一个“实时时钟”板。为了能使该板组合到系统中，以便产生时间和日期，有一个特殊设备(即时钟设备)(由属性字决定)。该设备定义和执行的功能类似于其它字符型设备(大多数功能是设置完成位、重置错误位、返回位)。当对该设备进行读写时，实际上传输6个字节。首先2个字节是一个字，该字从1-1-80开始计算日期。第三个字节是分。第四个字节是小时。第五个字节是1/100秒。第

六个字节是秒。在读取时钟设备时，可以得到日期和时间。在给时钟设备输入时，可以设置日期和时间。

第十节 设备驱动程序实例

DOS 3.00和3.10 版本增加了一个软盘，该软盘含有一个叫VDISK.LST的设备驱动程序实例清单，可以用 PRINT 命令将清单打印出来，以便参考使用(中国计算机技术服务公司备有“VDISK.LST设备驱动程序实例”软盘)。

第三章 扩展屏幕和键盘 控制的使用方法

第一节 引言

本章介绍如何发出特殊控制的字符序列：

- 控制光标位置
- 删除屏幕上的文本
- 设置操作方式
- 重新定义键盘键的意义

第二节 控制序列

如果使用标准输入、标准输出或标准指错信息输出装置，通过DOS功能调用01H，02H，06H，07H，09H，0AH和40H，就能产生有效的控制序列。

扩展屏幕和键盘控制设备驱动程序ANSI.SYS的安装，必须通过设置在配置文件CONFIG.SYS中设置下列语句来实现：

```
device = a[d:][path]nsi.sys
```

DOS系统在主存中所占容量的大小随着ANSI.SYS所占容量的大小而改变。

第三节 控制序列语法

每个光标控制序列的格式如下：

ESC [parameters COMMAND

ESC	ESC为一字节(1byte) ASCII 代码, 并不是三个字节代码。
[字符“[”。
Parameters (参数)	参数是用“#”表示的。给“#”指定数字值。“#”代表一个数字型参数。数字型参数是一个用 ASCII 字符表示的整数。如果不指定参数值, 或指定参数值为零, 则系统自动使用系统默认的参数值。
COMMAND	代表命令的字母串。该字母串有特定的含义。

例如：

ESC[2; 10H

该命令建立如下的BASIC 程序：

The IBM Personal Computer Basic

Version 3.00 Copyright IBM Corp.1981, 1982,

1983, 1984

× × × × × Bytes free

OK

Open "Sample" for Output as 1

OK

print #1, CHR\$(27); "[2;10H" ; "× row 2col 10"

OK

Close **##1**

OK

注：这里“CHR\$(27)”即为ESC控制序列。

第四节 光标控制序列

下表给出怎样控制光标位置来产生光标控制序列。

一、光标位置

光标位置	功 能
ESC[##;##H	把光标移动到由参数指定的位置。第一个参数(##)是指行号，第二个参数(##)是指列号。系统默认值为1。如果不给出参数，光标回到初始位置(屏幕左上角)。

上面提到的例子是将光标设置在屏幕的2行10列位置，将前例中提到的SAMPLE文件内容拷贝到CON文件中：

```
type sample
```

二、光标上移

光标上移	功 能
ESC[##A	将光标上移1行或多行，不改变列的位置。“##”的值决定了上移行的数目。系统默认值为1。如果光标已在顶行，本指令不被执行。

三、光标下移

光标下移	功 能
ESC[##B	将光标下移1行或多行，不改变列的位置。“##”的值决定了下移行的数目。系统默认值为1。如果光标已在最底行，本指令不被执行。

四、光标前移

光标前移	功 能
ESC[# C	将光标前移一列或多列,不改变行的位置。“#”的值决定了前移列的数目。系统默认值为1。如果光标已在最右列位置,本指令不被执行。

五、光标后移

光标后移	功 能
ESC[# D	将光标后移一列或多列,不改变行的位置。“#”的值决定了后移列的数目。系统默认值为1。如果光标已在最左列位置,本指令不被执行。

六、水平和垂直位置

水平和垂直位置	功 能
ESC[# ; #	将光标移至参数指定的位置。第一个参数指定行号,第二个参数指定列号。系统默认值为1。如果不指定参数,则光标回到初始位置(屏幕左上角)。

七、光标位置报告

光标位置报告	功 能
ESC[# ; # R	标准输出设备给出光标的当前位置:第一个参数指出光标当前所在的行,第二个参数指出光标当前所在的列。

八、设备状态报告

设备状态报告	功 能
ESC[6n	控制台驱动程序在收到设备状态报告指令后立即输出一个光标位置报告。

下面的例子是让ANSI.SYS将当前光标位置(行和列)存入键盘缓冲器中。然后, ANSI.SYS将其从键盘缓冲器中读出并显示在屏幕上。

```
PROGRAM dsr(INPUT, OUTPUT);
```

```
VAR
```

```
  f:FILE OF CHAR;
```

```
  key:CHAR;
```

```
FUNCTION inkey:CHAR;           {read character}
```

```
VAR                             {from the}
```

```
  ch:CHAR;                       {keyboard buffer}
```

```
BEGIN
```

```
  READ(f, ch);
```

```
  inkey := ch
```

```
END;
```

```
BEGIN
```

```
  ASSIGN(f, 'user');
```

```
  RESET(f);
```

```
    WRITE(CHR(27), '[6n');    {issue a DSR}
```

```
    key := inkey;              {read up to}
```

```
    key := inkey;              {first digit}
```



```

key:=inkey;                                {of the row}
WRITE('row', inkey, inkey, 'column');
key:=inkey;                                {skip to column}
WRITE(inkey,inkey)                          {write column}

```

END.

九、保存光标

保存光标	功 能
ESC[S	保存当前光标的位置。这个光标位置可以用恢复光标位置序列指令来重新恢复（见下述）。

十、恢复光标位置

恢复光标位置	功 能
ESC[u	将光标恢复到控制台驱动程序接收执行保存光标位置指令时所保存的相应的位置。

第五节 删除

下列表给出删除屏幕文本的控制序列。

一、显示器屏幕删除

显示器屏幕删除	功 能
ESC[2J	删除全屏幕，光标回到原始位置（左上角）。

二、行删除

行删除	功 能
ESC[K	删除从光标位置（包括光标所在位置）到该行末尾的所有字符。

第六节 操作方式

下列表给出了设置操作方式的控制指令序列。它们是：

- 设置图象显示方式(SGR)
- 设置方式(SM)
- 重设置方式(RM)

设置图象显示方式 (SGR)

SGR方式	功 能																																														
ESC[<i>n</i> ; ... ; <i>n</i>]	<p>设置由参数规定的字符属性。一经设定，直到下一个SGR序列指令产生前，所有的字符都具有由此参数规定的同一属性。</p> <table border="1"> <thead> <tr> <th>参数</th> <th>意义</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>去掉全部属性 (正常的黑底白字)</td> </tr> <tr> <td>1</td> <td>粗体字 (高明暗度)</td> </tr> <tr> <td>4</td> <td>下划线 (仅适用于IBM单色显示器)</td> </tr> <tr> <td>5</td> <td>闪烁</td> </tr> <tr> <td>7</td> <td>反向显示</td> </tr> <tr> <td>8</td> <td>隐显示 (不可见显示)</td> </tr> <tr> <td>30</td> <td>黑字</td> </tr> <tr> <td>31</td> <td>红字</td> </tr> <tr> <td>32</td> <td>绿字</td> </tr> <tr> <td>33</td> <td>黄字</td> </tr> <tr> <td>34</td> <td>蓝字</td> </tr> <tr> <td>35</td> <td>深红字</td> </tr> <tr> <td>36</td> <td>蓝绿字</td> </tr> <tr> <td>37</td> <td>白字</td> </tr> <tr> <td>40</td> <td>黑色衬底</td> </tr> <tr> <td>41</td> <td>红色衬底</td> </tr> <tr> <td>42</td> <td>绿色衬底</td> </tr> <tr> <td>43</td> <td>黄色衬底</td> </tr> <tr> <td>44</td> <td>蓝色衬底</td> </tr> <tr> <td>45</td> <td>深红衬底</td> </tr> <tr> <td>46</td> <td>蓝绿衬底</td> </tr> <tr> <td>47</td> <td>白色衬底</td> </tr> </tbody> </table>	参数	意义	0	去掉全部属性 (正常的黑底白字)	1	粗体字 (高明暗度)	4	下划线 (仅适用于IBM单色显示器)	5	闪烁	7	反向显示	8	隐显示 (不可见显示)	30	黑字	31	红字	32	绿字	33	黄字	34	蓝字	35	深红字	36	蓝绿字	37	白字	40	黑色衬底	41	红色衬底	42	绿色衬底	43	黄色衬底	44	蓝色衬底	45	深红衬底	46	蓝绿衬底	47	白色衬底
参数	意义																																														
0	去掉全部属性 (正常的黑底白字)																																														
1	粗体字 (高明暗度)																																														
4	下划线 (仅适用于IBM单色显示器)																																														
5	闪烁																																														
7	反向显示																																														
8	隐显示 (不可见显示)																																														
30	黑字																																														
31	红字																																														
32	绿字																																														
33	黄字																																														
34	蓝字																																														
35	深红字																																														
36	蓝绿字																																														
37	白字																																														
40	黑色衬底																																														
41	红色衬底																																														
42	绿色衬底																																														
43	黄色衬底																																														
44	蓝色衬底																																														
45	深红衬底																																														
46	蓝绿衬底																																														
47	白色衬底																																														

设置方式(SM)																			
SM	功 能																		
ESC[=#h 或ESC[=h 或ESC[=0h 或ESC[?7h	<p>根据指定的参数，确定屏幕宽度或打印格式。</p> <table border="1"> <thead> <tr> <th>参数</th> <th>意义</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>40×25黑白显示</td> </tr> <tr> <td>1</td> <td>40×25彩色显示</td> </tr> <tr> <td>2</td> <td>80×25黑白显示</td> </tr> <tr> <td>3</td> <td>80×25彩色显示</td> </tr> <tr> <td>4</td> <td>320×200彩色打印</td> </tr> <tr> <td>5</td> <td>320×200黑白打印</td> </tr> <tr> <td>6</td> <td>640×200黑白打印</td> </tr> <tr> <td>7</td> <td>在行尾接下一行打（打印到行末端时能自动地转下一行开头再打）</td> </tr> </tbody> </table>	参数	意义	0	40×25黑白显示	1	40×25彩色显示	2	80×25黑白显示	3	80×25彩色显示	4	320×200彩色打印	5	320×200黑白打印	6	640×200黑白打印	7	在行尾接下一行打（打印到行末端时能自动地转下一行开头再打）
参数	意义																		
0	40×25黑白显示																		
1	40×25彩色显示																		
2	80×25黑白显示																		
3	80×25彩色显示																		
4	320×200彩色打印																		
5	320×200黑白打印																		
6	640×200黑白打印																		
7	在行尾接下一行打（打印到行末端时能自动地转下一行开头再打）																		

重设置方式(RM)	
RM	功 能
ESC[=#1 或ESC[=1 或ESC[=01 或ESC[?71	除参数7外其余参数及其功能与SM（设置方式）相同。此处，参数7是将截止符设在行末尾，所有超长字符都被丢失。

第七节 键盘键重定义

下表列出了重新定义键盘键含义的控制序列。

控制序列	功 能
ESC[# ;... ;#p 或ESC[# ;"String"; # ;# ;"String" ; #p 或其它字符串和数字的 任意组合。	控制序列的第一位ASCII码指出要处理的那个码。其余的数码表示当键被按下时，产生出一组特定的ASCII码序列。然而，如果第一位码是0(NULL)，则第一和第二位产生出一组扩展的ASCII重定义码（见第六章扩展ASCII码一览表）。

下面是一些实例。

通过作实例练习，你可以学会如下内容：

- 生成一个包括下述语句的文件，然后用 TYPE 命令显示包含该语句的文件。

- 在DOS系统提示下执行命令。

1. 将Q和q键重新指定为A和a(其它键亦可)

①生成文件：

```
ESC [65;81p    A becomes Q
```

```
ESC [97;113p   a becomes q
```

```
ESC [81;65p    Q becomes A
```

```
ESC [113;97p   q becomes a
```

②在DOS系统提示下：

```
prompt $e[65;81p    A becomes Q
```

```
prompt $e[97;113p   a becomes q
```

```
prompt $e[81;65p    Q becomes A
```

```
prompt $e[113;97p   q becomes a
```

2. 将功能键F10定义为DIR命令，然后回车。

①生成一个文件：

```
ESC [0; 68; "dir"; 13p
```

②在DOS系统提示下：

```
prompt $e[0; 68; "dir"; 13p
```

其中：\$e是有关ESC的提示命令字符。0; 68是F10键指定的扩展ASCII码。13(十进制数)是回车的数字信号。

3. 下例是让系统提示将当前目录显示在屏幕顶部，将当前磁盘驱动器号显示在当前行上。

```
prompt $e[s$e[l; 30]$e[k$p$e[ u$n$g如
```

当前目录是C:\FILES, 现行的驱动器是C, 则显示:

C:FILES

C>

4. 下例是一段汇编语言程序, 将F10键定义为DIR B:命令, 然后回车。

```
TITLE SETANSI.ASM-SET F10TO STRING
FOR ANSI.SYS
CSEG SEGMENT PARA PUBLIC 'CODE'
ASSUME CS:CSEG.DS:CSEG
ORG 100H
ENTPT:JMP SHORT START
STRING DB 27,'10;68;
"DIRB: ";13p' ;REDEFINE F10 KEY
STRSIZ EQU $-STRING ;LENGTH OF ABOVE
MESSAGE
HANDLF EQU 1 ;PRE-DEFINED FILE
;HANDLE FOR STAND-
ARD OUTPUT
START PROC NEAR
MOV BX,HANDLE ;STANDARD OUTPUT
DEVICE
MOV CX,STRSIZ ;GET SIZE OF TEXT
TO BE SENT
MOV DX,OFFSET ;PASS OFFSET OF
STRING STRING
;TO BE SENT
```

```
MOV AH,40H ;FUNCTION: "WRITE  
TO DEVICE"  
INT 21H ;CALL DOS  
RET ;RETURN TO DOS  
START ENDP  
CSEG ENDS  
END ENTPT
```

第四章 文件管理说明

第一节 引言

本章介绍以下两个方面的内容：

- 如何使用文件管理功能（包括 FCB 功能调用和句柄功能调用）。
- 如何在 ASCII 和二进制方式下进行文件的输入/输出。

第二节 版本专用信息

在本章中如下信息专用于 DOS 版本：

FCB 使用的范围：对于 DOS 3.00 和 DOS 3.10，如果 SHARE 被装入并且对 FCBS 命令作出了规定，那么用 FCB 打开的文件个数是有限制的。

第三节 文件管理功能

用 DOS 功能调用来建立、打开、关闭、读出、写入、重新定名、查找和删除文件，有以下两类功能调用支持DOS

的文件管理：

- 文件控制块(FCB)功能调用(FCB功能调用 0FH—24H)。

- 扩展功能调用(句柄功能调用 39H—62H)。

句柄功能调用比 FCB 功能调用易于使用而且更有效。

下表将 FCB 功能调用和句柄功能调用作出了比较。

FCB 功能调用	句柄功能调用
只能访问当前目录中的文件。	可以访问任何目录中的文件。
应用程序必须保留文控制块，这样才能打开、建立、重新定名或删除文件。对于输入/输出请求，应用程序也需要FCB。	不需要保留FCB，但需要包含一个打开驱动器、路径和文件名的字符串，这样才能打开、建立、重新定名或删除文件。对于文件的输入/输出请求，应用程序只需要保留一个由DOS提供的16位的字(文件句柄)。

应用程序使用 FCB 功能调用的唯一原因是为了能在 DOS 1.1 上运行。为达到这一目的，程序可以只用 DOS 1.10 提供的功能调用 (00H—2EH)。

第四节 FCB功能调用

FCB 功能调用要求每一个打开的文件用一个文件控制块。应用程序和 DOS 都要求保留这个文件控制块。应用程序提供了一个指向 FCB 的指针并且将该指针写入相应的字段以提供给特殊的功能调用。FCB功能调用可以对系统中有效的驱动器进行文件管理，但只限于当前目录中指定的驱动器。顺序输入/输出是通过使用顺序读写功能调用和使用FCB

中当前的块、当前的记录和当前的记录长度字段来实现的。随机输入/输出是通过填写随机记录号和随机记录长度字段来实现的。参见第七章中“文件控制块”一节中有关 FCB 结构说明。

在程序设计时要考虑的，有几种 FCB 功能调用在某些环境中是不能运行的。这是为了避免出现文件的共享和兼容的问题。一种程序设计错误就是使用同一个 FCB 结构存取几个打开的文件。用一个 FCB 对打开的文件执行 I/O，在文件控制块的文件名字段中填入新的文件名，然后，程序再用同一个 FCB 打开第二个文件，这样做是非法的。因为 DOS 将文件的控制信息写在 FCB 的保留字段中，用同一个 FCB 打开第二个文件时这一信息改变了，如果程序再用原来的文件名填入文件名字段中，然后对该文件执行 I/O，这时 DOS 将拒绝执行，因为控制信息已经改变了。一个 FCB 在当前打开的文件没有关闭之前是不能再用来打开第二个文件的。如果要同时打开几个文件，必须分别使用几个 FCB。

尽管 DOS 保留字段是随着 DOS 版本改变而不同，但程序决不能左右 FCB 中的 DOS 保留字段。注意对一个文件所有 I/O 完成之后就关闭该文件，这是编程的一种良好习惯。为了避免可能出现的文件共享问题，要限制用 FCB 功能调用可同时打开文件的个数。用应用程序对一个当前已被打开的文件进行删除或重新定名，系统也认为是错误的，应用程序不应该做这类事。

第五节 句柄功能调用

较好的文件管理方法是使用扩展的句柄集功能调用。句柄功能调用的使用不限于当前目录中的文件。另外，文件管理调用的句柄集允许应用程序定义在文件共享时能同时对同一个文件进行其它处理的访问类型。

为了建立或打开文件，应用程序提供了一个指向ASCIIZ字符串的指针，该字符串给文件确定了名和位置。ASCIIZ字符串包含了以00H（1个字节）为结尾的可选的驱动器名、路径和文件说明。

下面是一个 ASCIIZ 字符串的例子：

```
DB"a:\path\filename.ext", 0
```

如果正在建立文件，应用程序还能给文件确定属性。属性是一组定义文件是否是只读、保密、系统、目录或卷标头的值。详见第五章“DOS 磁盘目录”一节中有关文件属性的说明。

如果正在打开一个文件，应用程序还能定义文件被打开后的共享和访问方式。应用程序定义的访问方式能通知DOS对打开的文件执行什么样的操作（只读、只写或读/写）。共享方式决定了其它进程可以同时对该文件执行的操作类型。程序还能控制子进程是否继承父进程打开的文件。只有在文件被打开后对文件共享时，共享字段才有意义。

为了对一个文件重新定名或删除，应用程序只需要提供一个指向ASCIIZ字符串的指针，该字符串包含要重新定名或删除的文件的名字和位置，如果要重新定名的话，还要一个

新文件名字符串。

打开或建立文件的功能调用给出一个16位的字作为文件句柄。对文件做任何I/O，程序都用该句柄指向该文件。一旦文件被打开，程序就不再需要保留指向该文件的 ASCIIZ 字符串，也不要求指针一定指向同一个目录，因为DOS保存了文件位置的自动跟踪功能，而无需考虑当前目录是哪一个。

顺序 I/O 可以用句柄读 (3FH) 或句柄写 (40H) 功能调用来实现。对文件的数据区执行I/O时，句柄功能调用能自动地移到刚刚读或写过的内容的末尾。如要求执行随机 I/O 时，可以使用 LSEEK (42H) 功能调用将数据区置于要执行 I/O 的文件中。

第六节 专用文件的句柄

DOS 设置了五个专用文件句柄供应用程序使用，这五个句柄是：

0000H 标准输入设备 (Stdin)

0001H 标准输出设备 (Stdout)

0002H 标准指出设备 (Stderr)

0003H 标准辅助设备 (Stdaux)

0004H 标准打印设备 (Stdprn)

这些句柄是由 DOS 预先定义的，且任何应用程序都可以使用。尽管程序可以关闭这些句柄，但使用时都不需要打开。Stdin 是只读文件句柄。Stdout 和 Stderr 是只写文件句柄。Stdin 和 Stdout 句柄能够用来重新改变文件的传输方向。所有由进程固定控制的句柄都能够被改变传输方向，但

不能用程序的命令行来改变。

用这些句柄对控制台设备执行 I/O 是非常有用的。例如,你能够用读功能调用 (3FH) 和文件句柄 000H(Stdin)从键盘上读入数据,还可以用写功能调用 (40H) 和文件句柄 0002H(Stdout)将数据输到控制台屏幕。如果你不要改变输出方向,你可以使用文件句柄0002H(Stderr)来进行输出。为便于用户能看到实际作用的效果,指错信息和提示符是非常有用的。

文件句柄0003H (Stdaux) 和0004H(Stdprn)都可以读和写。Stdaux 一般是用于串行设备。而 Stdprn通常是用于并行设备。

第七节 ASCII和二进制方式

文件 I/O 是在二进制方式下进行的,这意味着从文件读取数据或将数据写入文件时不用变换。然而, DOS也能在 ASCII 方式下对设备进行读写。在 ASCII 方式下, DOS 读写时,需要对字符串进行处理和对字符进行变换,在DOS进行初始化时就要预先定义文件句柄是在 ASCII 方式下,所有其它不涉及读写设备的文件句柄仍在二进制方式下。程序可以设置IOCTL(44H) 功能调用方式对设备进行I/O。如果预先定义的文件句柄都是对设备而言,就可以用 IOCTL 将 ASCII 码方式转变成二进制方式。不是用于设备的正规的文件句柄总是在二进制方式下,在这种情况下不能转换成 ASCII 方式。

预先定义的句柄 Stdin (0000H)、Stdout (0001H) 和

Stderr (0002H) 都是互联的句柄。用 IOCTL 功能调用改变三个句柄中任一个句柄的方式时, 其它两个句柄方式也随之改变。如果用 IOCTL 改变 Stdout 为二进制方式, 那么 Stdin 和 Stderr 也自动地随之改变成二进制方式。

一、二进制方式下的文件 I/O

1. 在二进制方式下读一个文件时:

- 如果在读的过程中检测不到字符 $\backslash S$ (Scroll lock)、 $\backslash P$ (Print Screen)、 $\backslash C$ (Control Break)时, 即使是读了 $\backslash S$ 或 $\backslash P$ 时, 打印机也不发生反弹。

- 对 Stdout (0001H) 没有反弹。

- 在读取一个规定字节的数时, 在接收到最后一个字节或遇到文件结束时, 系统立即返回到开始状态。

- 如果以 Stdin (0000H) 方式输入, 不允许用功能键编辑输入行。

2. 在二进制方式下写一个文件时:

- 在写操作过程中检测不到字符 $\backslash S$ 、 $\backslash P$ 和 $\backslash C$ 时, 打印机就没有反弹。

- 对 Stdout (0001H) 没有反弹。

- 写出规定字节数的数据。

- 不分离控制字符。例如, 控制符 $\backslash D$ 是按一个字节 04H发送而不是以两个字节 \backslash 和 D发送的。

- 不会将制表符扩展为空格。

二、ASCII 方式下的文件 I/O

1. 在ASCII方式下读一个文件时:

- 检测字符 $\backslash S$ 、 $\backslash C$ 和 $\backslash P$ 。

- 在 ASCII 方式下读取一个文件时有两种情况，一种是要求取出的字符数与设备输入缓冲区中的字符数一样多；另一种情况是要求取出的字符数少于设备缓冲区中的字符数。如果要求取出的字符数小于设备输入缓冲区中字符数，则先取出少于设备输入缓冲区中字符数的那部分字符，下一次再读取缓冲区中剩下的那部分字符。

- 如果设备输入缓冲区已空，则可读入一行（结束符用 $\backslash M$ ）到缓冲区中。该行可以用功能键编辑。取出的字符以 $0DH$ 、 $0AH$ ($\backslash M$ 、 $\backslash J$) 为结尾（要求取出的字符数要足够地包括这些结尾字符）。例如，如果要求取出5个字符，在 DOS 接收到控制台输入回车 ($0DH$ 或 $\backslash M$) 之前只输入3个字符，那么该3个字符和 $0DH$ 和 $0AH$ 将被取出。但是，如果要求取出5个字符，而在回车之前却输入了7个字符，那么在这种情况下，只取出前5个字符， $0DH$ 、 $0AH$ 将取不到。如果在接收到回车时，输入的字符少于要求取出的字符，那么就能取到所有输入的字符和 $0DH$ 、 $0AH$ 结尾符。将 $0AH$ ($\backslash J$) (单字节) 加到取出字符中是为了使许多设备将取出的文件看作是文本文件。

- 如果检测到 $1AH$ ($\backslash Z$)，则从该点结束输入。 $0DH$ 和 $0AH$ 加不到字符串中。

- 完成反弹。

- 在反弹制表符时，能将制表符扩展为空格。在输入缓冲区时，使用制表符字节 ($09H$)。

2. 在 ASCII 方式下写一个文件：

- 在写操作过程中，检测字符 $\backslash S$ 、 $\backslash P$ 和 $\backslash C$ 。

- 把制表符扩展到8个字符宽并填上空格(20H)。
- 能够分离控制符。例如，^D可写成两个字节^和D。
- 在输出字节时，或者按规定的字节数输出，或者输出检测到^Z以前的字节，这就是用户实际上取出的字节数。

第八节 允许打开文件的个数

DOS对可以同时打开的文件个数是有限制的。打开文件的个数取决于文件是如何打开和建立的（FCB或句柄功能调用），以及CONFIG.SYS文件中的FCB和FILES命令所指定的数字。允许FCB功能调用打开文件的个数和由句柄集功能调用能够打开的文件个数是互不相干的。

第九节 FCB使用的限制

如果使用了SHARE命令但实际上不需要对文件共享时，那么用FCB功能调用可以同时打开文件的个数是不受限制的。但是，在需要对文件共享时，FCB打开文件的最大个数则受CONFIG.SYS文件中FCBS命令设置的值限制。有关FCBS命令的说明请参看本书第四章。FCBS命令可以规定m、n两个值，m规定了可由FCB打开的文件总数，n规定了由FCB打开后防止关闭的文件数。

当FCB打开的文件数超过最大值时，DOS自动关闭最近使用的那个文件，此时任何试图访问该文件都会产生中断24H临界指错信息，即“FCB not available”。如果应用程序运行时出现这种情况，由FCBS命令所规定的m值应该

增大。

当 DOS 决定关闭最近使用的那个文件时，并不包括起初打开的 n 个文件，所以防止了起初打开的 n 个文件的关闭。

第十节 句柄使用的限制

同时可以被所有进程打开的文件句柄数取决于 CONFIG.SYS 文件的 FILES 命令(见 DOS 参考手册的 FILES 命令)。单个进程可以打开的文件数由 FILES 命令规定。如果 FILES 命令规定可以打开文件数是大于或等于 20，则单个进程可以打开 20 个文件；如果 FILES 命令规定的数值小于 20，则单个进程可以打开的文件数也小于 20。这个数值包括三个预先定义的句柄，一个句柄用于标准输入/输出/指错，一个用于标准辅助设备，还有一个用于标准打印机。这就是说单个进程最多可以再打开 17 个句柄(20 减 3)。

第十一节 文件空间的分配

在磁盘上写文件没有必要按顺序进行，可以根据需要为文件分配空间。在磁盘上正在写入的文件决定了下一个文件有效位置的分配。因而，若有许多文件需要建立和删除，新建立的文件可能不是写在按顺序排的扇区内。但是，通过文件分配表(FAT)在对文件分配空间(通过链接)以及进行有效的功能调用时，任何文件都可以采用顺序方式或随机方式。

空间的分配以簇为单位。簇的大小是随盘的种类而不

同，单面单密软盘每一个簇是一个扇区，而其它软盘每一个簇可能有几个扇区。硬盘簇的大小取决于 DOS 分区的大小和用FORMAT命令对硬盘的格式化划分。例如，一个10M字节的硬盘全部分给一个 DOS 分区，这时簇的大小为 8 个扇区。

应用程序并不关心 DOS 为文件分配磁盘空间的方式。簇的大小只对决定每次为文件分配的最小空间值有重要意义。例如，对一个每簇2扇区、每扇区512字节的软盘而言，分配给文件的空间是以1024字节为单位的。因此，即使一个文件的长度小于一个簇，也要给该文件分配一个簇的空间。若需要更多的磁盘空间，则分配给文件更多的簇。当所有存在的簇都分配完了则认为磁盘已满。

第五章 DOS磁盘分配

第一节 引言

本章介绍与 DOS 有关的如下内容:

- 自举记录
- 12位或16位的 DOS文件分配表(FAT)
- DOS磁盘目录
- 数据区域

第二节 版本的特殊说明

本章中如下的内容是 DOS 版本专用的:

DOS文件分配表(FAT):

- 12位 FAT 用于DOS2.10、3.00 和 3.10
- 16位 FAT 用于DOS3.00 和 3.10

另外,对于DOS文本3.00和3.10,文件分配表指针即F9H用来标识每磁道15个扇区的软盘。

第三节 DOS区域

由DOS格式化的所有磁盘和软盘都生成每个扇区512字

节的空间，DOS区（对于软盘片而言是整个盘片，对于硬盘而言是DOS分区）格式化如下表：

自举记录——1扇区
文件分配表(FAT)的第一份拷贝——大小可变
文件分配表(FAT)的第二份拷贝——小同第一份拷贝
自举目录——大小可变
数据区

下面分别介绍每一个区的分配。

第四节 自举记录

自举记录写在由DOS FORMAT命令格式化过的磁盘的0面、0道、1扇区上。如果你试图起动在驱动器A中没有系统盘的系统（无硬盘），就会产生指错信息。硬盘的自举记录驻留在DOS分区的第一个扇区上。

第五节 DOS文件分配表 (FAT)

本节阐述DOS如何使用文件分配表(FAT)将文件的簇转换成逻辑扇区号。建议在系统实用程序中使用DOS句柄功能调用而不是去翻译FAT。DOS使用FAT以一次一簇的方式为文件分配磁盘空间。

如果硬盘容量大于20740个扇区（即10M字节以上），FAT在磁盘上每一簇由一个12位（1.5字节）项或由一个16位项（2字节）组成。

开始两个 FAT 项映象到一段目录，这两个项包含磁盘容量标记和磁盘格式。FAT 可以是12位或16位。DOS 根据磁盘上分配单元总数确定是否用12位还是16位FAT。对所有小于20740个扇区的软盘或硬盘 DOS 分区用12位FAT映象一个簇，如多于20740个扇区，则采用16位FAT。

第二、三和四（若用16位FAT）字节总是包含FFFFH。第一个字节内容如下：

<u>16进制值</u>	<u>意义</u>
FF	双面每道8扇区软盘
FE	每道8扇区单面软盘
FD	每道9扇区双面软盘
FC	每道9扇区单面软盘
F9	每道15扇区双面软盘
F8	硬盘

第三个 FAT 项映象到数据区的开头（002簇）。注意：这些值仅供参考，不能用在程序中。

每一个项包含3个16进制字符（16位FAT为4个字符）。

() 表示16位FAT的高位值。

<u>16进制值</u>	<u>意义</u>
(0)000	如果簇存在且未被使用
(F)FF8—(F)FFF	指明文件的最后一簇
(X)XXX	任何其它16进制字符都表示文件下一个簇的簇号。文件第一个簇的簇号存放在文件的目录项中
值(F)FF0—(F)FF7	用来表示保留簇。(F)FF7若不是定

位键的一部分则表示坏簇。(F)FF8—(F)FFF用作文件结束标志。

文件分配表总是放在自举目录之后，若 FAT 大于一个扇区，则占用紧挨着的扇区，两个 FAT 拷贝连在一起存放，一个挨一个，连成一个整体。需要使用 FAT（如打开、分配更多的空间等等）时，将其读到 DOS 的一个缓冲区中。

一、如何使用 12 位 FAT 项文件分配表

首先从目录项中得到文件的起始簇。现在来看怎样确定文件每一个子序列簇：

1. 将刚刚用过的簇号乘以 1.5（FAT 项长 1.5 字节）；
2. 积作为 FAT 的偏移量，该积既指明了刚用过的簇号，又包含了文件的下一个簇的簇号；
3. 用 MOV 指令把计算出的 FAT 偏移量作为字送到寄存器；
4. 若最后一个簇是偶数，那么保留寄存器的低 12 位，否则保留高 12 位；
5. 如果得到 12 位值是 (FF8—FFF)H，则表明文件结束；否则该 12 位值是文件下一个簇的簇号。

为了将簇号转换成逻辑扇区号（即相对扇区号，如 INT 25H、INT 26H 和 DEBUG 所用的扇区号）：

1. 从簇号中减 2；
2. 将结果再乘以每簇的扇区号；
3. 加上数据区开始的逻辑扇区号。

二、如何使用 16 位 FAT 项文件分配表

首先从目录项中获得文件的起始簇。现在来看怎样确定文件每一个子序列簇：

1. 把刚刚用过的簇号乘以 2（每一个 FAT 项是 2 个字节长）；

2. 用 MOV 指令把计算出的 FAT 偏移量作为字送到寄存器；

3. 若得到的 16 位值是 (FFF8—FFFF)H，表示文件已结束，否则这 16 位值给出了文件下一个簇的簇号。

第六节 DOS 磁盘目录

FORMAT 命令一开始就为整个磁盘建立根目录。根目录分配文件空间（逻辑扇区号），并且通过设备驱动程序接口确定有效的最大项目号。

目录项

由于除根目录外的所有目录实际上是文件，所以其包含的项数是没有限制的。

所有目录项均为 32 字节长，格式如下所述（字节偏移量均为十进制），下面分段描述目录项字节数。

1. 字节 0—7

字节 0 到 7 表示文件名。文件名的第一个字节表示文件名的状态，文件名的状态可以是以下值：

00H——表示文件名没有被使用过。为便于操作，使用该字节来限制查找目录的长度。

05H——表示文件名的第一个字符确含有 E5H 字符。

E5H——文件名被使用过，但已被删除。

2EH——该项为目录。如果第二个字节是2EH，则其簇域内(字段内)含有其父目录的簇号(若父目录为根目录时，簇号为0000H)。

任何其它字符都看作为文件名的第一个字符。

2. 字节8—10

这三个字节为文件扩展名。

3. 字节 11

该字节指明文件的属性。属性字节(十六进制值)定义如下：

(注意：属性 08H 和 10H 不能由功能调用 43H(CHMOD)来修改。系统文件(IBMIO.COM和IBMDOS.COM)只能是只读、保密和内部文件。这些文件生成时可以定为保密文件。另外，只读、保密、系统和卷宗属性都可以用 CHMOD 功能调用来修改。)

01H——表示文件具有只读属性。如果企图用功能调用 3DH 打开文件进行输出，则将会给出指错信息。

02H——文件具有保密属性。在正常目录查找时将找不到该文件。

04H——文件具有系统文件属性。在正常目录查找时也找不到该文件。

08H——目录项包含有前 11 个字节的卷宗号。该目录不包含其它信息且只能存于根目录中。

10H——表示该目录定义了一个子目录。在正常目录查找时找不到该目录。

20H——表示卷宗位。该位不论文件是关闭的还是被写的都被置位。BACKUP 和 RESTORE 使用该位确定文件最后

一次备份之后是否修改过。该位可以与其它属性位一起使用。

其它位为保留位，必须置0。

4. 字节12—21

这些字节是DOS的保留区。

5. 字节22—23

这两个字节包含了文件建立或最后一次修改后的时间。
时间与位对应如下：

```

<      23      > <      22      >
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
h  h  h  h  h  m  m  m  m  m  m  x  x  x  x  x

```

其中：

hh——二进制小时（0—23）。

mm——二进制分（0—59）。

xx——以2秒为增量的二进制值。

注意：时间按低有效字节首先存贮。

6. 字节24—25

这个字节范围定义了文件生成或最后修改的日期。

mm/dd/yy（月/日/年）与位对应如下：

```

<      25      > <      24      >
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
y  y  y  y  y  y  y  m  m  m  m  d  d  d  d  d

```

其中：

mm——1—12

dd——1—31

yy——0—119（1980—2099）

注意：日期是以最低有效字节首先存储的。

7. 字节26—27

这个字节范围给出了文件第一个簇的起始簇号。所有硬盘和软盘上数据区的第一个簇都是 002 簇。簇号是以最低有效字节首先存储的。

注意：系统程序设计员可以参看本章“DOS 文件分配表 (FAT)”有关簇号和逻辑扇区号转换的详细内容。

8. 字节28—31

该字节范围给出了文件的字节数，第一个字包含了字节数的低部。两个字都是以最低有效字节首先存储的。

第七节 数据区

只有当文件需要时才给文件分配数据区空间（而不是预先分配）。空间是按每次一个簇的方式来分配的（簇是分配的最小单位）。一个簇总是由一个或多个连续的扇区组成的，文件所有的簇在文件分配表中都是链接在一起的。

在磁盘上簇的安排原则是使多面介质上的磁道头移动距离最小。在移到下一磁道之前，先将本磁道上的所有空间都分配完。分配方法是先按扇区顺序号从最小号数的磁道头开始分配，然后再给下一个磁道头分配所有扇区，直到这个磁道上所有磁道头上的扇区都分配完毕为止。那么，下一个要分配的扇区就是下一道 0 号磁道头的 1 扇区。

对于硬盘，在用 FORMAT 命令初始化时，文件分配表和目录的大小取决于 DOS 分区的大小。

对于软盘，使用下表来分配空间：

面 号	扇区/磁头	FAT 大小扇区	DIR 扇区	DIR 扇区	扇区/簇
1	8	1	4	64	1
2	8	1	7	112	2
1	9	2	4	64	1
2	9	2	7	112	2
2	15	7	14	224	1

数据区的文件没有必要顺序地写在磁盘上。数据区空间的分配是以一次一簇方式，分配时可以跳过已分配的簇。第一个遇到的空簇就是下一个将要被分配的簇，并不考虑簇在磁盘上的物理位置。这样使得磁盘空间得到有效的利用，因为删除文件后剩下的簇可以分配给新的文件。详见本章有关“DOS文件分配表”的说明。

第六章 DOS中断和功能调用

第一节 引言

本章内容为：

- DOS使用的寄存器表
- 扩展 ASCII 代码表
- 所有中断和功能调用的详细说明

第二节 与版本有关的信息

本章中下列信息与 DOS 版本有关：

• 中断：

DOS版本 2.10 支持中断 20H 到 27H。

DOS版本 3.00 支持中断 20H 到 2FH。

DOS版本 3.10 支持中断 20H 到 2FH。

• 功能调用：

DOS版本 2.10 支持功能调用 00H 到 57H。

DOS版本 3.00 支持功能调用 00H 到 5CH 和 62H，DOS

3.00 包括有下列新的和经改变过的功能调用：

- 3DH 打开文件，支持文件共享

- 59H 获得扩展错
- 5AH 生成临时文件
- 5BH 生成新文件
- 5CH LOCK/UNLOCK文件访问
- 62H 获得程序段前缀地址

DOS版本 3.10 支持功能调用 00H 到 62H 还包括有下列新的功能调用

- 5E00H 获得机器名
- 5E02H 建立打印机设置
- 5E03H 获得打印机设置
- 5F02H 获得重定向表格项
- 5F03H 设备重定向
- 5F04H 撤消重定向

对于 DOS 3.00 和 3.10, 中断 24H (致命错误处理程序向量), AH 的第 3—5 位指出出错后哪些响应是合法的。另外 DOS 3.00 和 3.10 处理非法响应的方式也与 DOS 2.10 不同。本章中“非法响应的处理一节有更多的内容提及。

第三节 DOS 寄存器

DOS 在执行中断和功能调用时使用下面的寄存器、指针和标志。

通用寄存器	
寄存器	定义
A X	累加器 (16位)
A H	累加器高字节 (8位)
A L	累加器低字节 (8位)
B X	基址 (16位)
B H	基址高字节 (8位)
B L	基址低字节 (8位)
C X	计数器 (16位)
C H	计数器高字节 (8位)
C L	计数器低字节 (8位)
D X	数据 (16位)
D H	数据高字节 (8位)
D L	数据低字节 (8位)
标志	OF, DF, IF, TF, SF, EF, AF, PE, CF,
指 针	
寄存器	定义
S P	堆栈指针 (16位)
B P	基址指针 (16位)
I P	指令指针 (16位)
段 寄 存 器	
寄存器	定义
C S	代码段 (16位)
D S	数据段 (16位)
S S	堆栈段 (16位)
E S	附加段 (16位)

变 址 寄 存 器

寄 存 器	定 义
DI	目的变址(16位)
SI	源变址(16位)

第四节 扩展ASCII代码

扩展ASCII代码	意 义
3	NUL (空字符)
15	tab上档
16-25	Alt-Q, W, E, R, T, Y, U, Z, O, P
30-38	Alt-Z, X, C, V, B, M, N
59-68	功能键F1到F10
71	Home
72	光标向上
73	Page UP
75	光标向左
77	光标向右
79	End
80	光标向下
81	Page down
82	Insert
83	Delete
84-93	F11—F20 (Shift F1—F10)
94-103	F21—F30 (Ctrl F1—F10)

对于某些标准 ASCII 代码无法表示的键或组合键，返回扩展的 ASCII 代码。扩展的 ASCII 代码作为一个 2 字节字符串的第二个字节被返回。因此，如果返回的 ASCII 值为 0，应检查其第二个字节以获得扩展的 ASCII 代码。

上表列出扩展的 ASCII 代码及其意义。

第五节 中 断

在打算查看或设置中断向量内容的程序时，建议使用 DOS 的功能调用 (35H 和 25H) 来实现目的，而不要直接引用中断向量地址。

DOS 将中断类型 20H 到 3FH 保留为己用。这意味着绝对内存地址 80H 到 FFH 由 DOS 所保留。所定义的中断如下 (所有值均为 16 进制)。

一、20H 程序结束

发出中断 20H 从程序中退出。此向量传送给 DOS 以恢复以前输入给该程序的结束地址，Ctrl-Break 地址，致命错误退出地址。所有的文件缓冲区被刷新，并关闭所有句柄。在发出此中断之前应关闭所有长度改变了的文件 (见功能调用 10H 和 3EH)。如果改变过的文件未关闭，则在目录中不能正确记录其长度、日期和时间。

如果一个程序在结束时要传出一个完成代码或出错代码，则必须使用功能调用 4CH (结束一个过程) 或 31H (结束过程并驻留)。这两种方法较之使用中断 20H 更方便，由它们返回的代码可用于批处理中。功能调用 4CH 一节讲述

了批处理中 ERRORLEVEL子命令的用法。

[注]: 在发出中断 20H 之前, 程序必须保证 CS 寄存器包含有程序段前缀的段地址。

二、21H功能请求

见“功能调用”一节。

三、22H结束地址

当程序结束时控制传送给此中断所在的地址。该地址在建立程序段时拷贝到程序的程序段前缀中。不要直接发出此中断, EXEC功能调用帮助实现此功能。

四、23H Ctrl-Break退出地址

如果用户在标准输入, 标准输出, 标准打印机或异步通信适配器工作期间打入Ctrl-Break, 则执行中断23H。如果BREAK为on, 在大多数功能调用中(除了调用06H和07H外)进行中断23H检查。如果用户所写的Ctrl-Break子程序存储了所有寄存器的值, 则可以用中断返回指令(IRET)结束以继续程序的执行。如果用户所写的中断程序从段间返回, 进位标志用来确定程序是否被流产。若进位标志置位, 则该程序流产, 否则继续执行(与IRET返回时一样)。如果用户写的Ctrl-Break中断使用了功能调用09H或0AH, 则^C, 回车和换行被输出。如果IRET来继续执行, I/O便从行首开始继续。此中断发生时, 所有寄存器设置成原始对DOS的功能调用时所具有的值。Ctrl-Break处理程序允许作些什么是没有限制的, 甚至也可以包括DOS功能调用, 只要使用IRET时不改变

寄存器。

如果程序建立了新段并装入了第二个程序，第二个程序又改变了Ctrl-Break地址，则第二个程序结束并返回到第一个程序时将引起Ctrl-Break地址被恢复到第二个程序执行前所具有的值，从第二个程序的程序段前缀中恢复，并不直接发出此中断。

五、24H致命错误处理程序向量

在DOS内发生致命错误时，控制由中断24H传送。如果错误是磁盘错(最常发生)，则AH度其比特7=0(高比特)，

出错代码	出错名称
0	试图对带有写保护的磁盘写
1	不认识的单元
2	驱动器未准备好
3	不认识的命令
4	数据出错(CRC)
5	请求结构长度不正确
6	寻道错
7	不认识的介质类型
8	扇区未找到
9	打印机无纸
A	写失败
B	读失败
C	一般失败

如果不是则比特7 = 1。

BP:SI包含有设备标题控制块的地址，从中可检索到附加的信息。

诸寄存器设置为重试操作，出错代码在DI寄存器的低半部，高半部未定义。出错代码如上表：

用户堆栈有效并且自顶向下的内容为：

IP 发出INT24H的DOS寄存器

CS

FLAGS

AX 原始INT24H请求时的用户寄存器

BX

CX

DX

SI

DI

DI

BP

DS

ES

IP 来自用户到DOS的原始中断21H

CS

FLAGS

如果执行IRET，寄存器便设置成，DOS按照(AL)内容作如下响应：

(AL) = 0 不理睬出错

= 1 重试此操作

= 2 通过中断23H结束该程序

= 3 正在进行的系统调用失败

注：选择“不理睬出错”作为一种响应时应小心，因为它将使DOS认为操作已完成，而实际上并未成功。

从致命出错处理程序中，将控制返回给用户的出错子程序，必须满足下列条件：

在INT24H发生前：

(1) 用户的应用程序初始化代码应保存INT24H向量，并用一个指向用户出错子程序的向量来取代之。

在INT24H发生时：

(2) 用户出错程序接收到控制后，它将标志寄存器推入堆栈，然后执行一个长调用 CALL FAR到步骤(1)保存的原始INT24H向量。

(3) DOS给出适当的提示，并等待用户输入(Abort, Retry, or Ignore)。用户输入后，DOS将控制返回给用户出错子程序中长调用后面的指令。

(4) 用户出错子程序可执行任何操作。如要在错误发生处返回到原始应用程序，则出错子程序需执行一条IRET指令。否则，用户出错子程序将从堆栈中去除IP，CS和标志寄存器。然后控制传送给所要求的点。

1. 磁盘出错

如果是磁盘硬错误(AH比特7=0)，则AL包含有失败的驱动器号(0=驱动器A，等等)。AH比特0—2指出受到影响的磁盘区域以及是读操作还是写操作，方式如下：

比特0 = 0是读操作

1是写操作

比特2—1 (受影响的盘区)

00 DOS区域

01 文件分配表

10 目录

11 数据区域

AH比特3—5指出哪些响应是合法的，它们是：

比特3 = 0 不允许FAIL

1 允许FAIL

比特4 = 0 不允许RETRY

1 允许RETRY

比特5 = 0 不允许IGNORE

1 允许IGNORE

2. 非法响应的处理 (DOS 3.00和3.10)

如果指定的是IGNORE (AL = 0)，但IGNORE 又是不允许的 (比特5 = 0)，则响应为FAIL (AL = 3)。

如果指定的是RETRY (AL = 1)，但RETRY 又是不允许的 (比特4 = 0)，则响应为FAIL (AL = 3)。

如果指定的是FAIL (AL = 3)，但FAIL 是不允许的 (比特3 = 0)，则响应为ABORT (AL = 2)。

3. 其他错误

如果AH的比特7 = 1，说明在某个字符设备上发生了错误，或是由于文件分配表FAT的内存映像不正确而引起的。可由检查BP:SI寄存器所传送的设备标题确定是什么原因导致退出的。如果属性字节比特位高，指示是一个块设备，则该错误是由于不正确的FAT引起的，否则，错误便是出在某个字符设备上。

如果涉及到的是一个字符设备，则AL的内容是不可预知的，错误代码如上述在DI中。

注：

(1) 由于磁盘错误，将控制交给出错处理子程序之前，DOS要重试三次。

(2) 对于磁盘错误，此类退出只当错误是在中断21H功能调用期间发生时执行。如果是在中断25H或26H期间出错不执行此种退出。

(3) 此子程序是从禁止中断方式进入的。

(4) 必须保留所有寄存器内容。

(5) 此中断处理程序应尽量避免使用DOS功能调用。如果必要，可使用功能调用01H到12H。其它功能调用的使用可能破坏DOS的堆栈并使DOS处于无法预知的状态。

(6) 此中断处理程序不得改变设备标题的内容。

(7) 如果此中断处理程序自行处理错误而不返回到DOS，则应当从堆栈中恢复应用程序的寄存器，去除最后三个字以外的堆栈中所有内容，然后发出IRET。这样在INT21H遇到出错后便可立即返回到该程序。注意，如果这样做的话，DOS将处于不稳定状态，直到发出一个较之12H更高的功能调用，因而是推荐的。

(8) 对于DOS3.00和3.10，当在FAT或DIR扇区发生致命错误时，IGNORE请求（AL=0）转变为FAIL。

(9) 欲得到更多的有关出错处理的信息，见本章“出错返回信息”和“扩展出错代码”两节。

(10) 对于DOS3.10，发生网络致命错误时，IGNORE请求（AL=0）转变为FAIL请求。

由BP:SI指出的设备标题格式如下:

对于下一个设备的双字指针 (如果是最后一个设备, 为FFFFH)

字属性:

比特15 = 1是字符设备
= 0是块设备

如果比特15 = 1:

比特0 = 1当前为标准输入
比特1 = 1当前为标准输出
比特2 = 1当前为NULL空设备
比特3 = 1当前为CLOCK时钟设备
比特14为IOCTL位

对于设备驱动程序方式 (strategy) 入口点的字指针

对于设备驱动程序中断入口点的字指针

对于块设备的8字节字符设备命名字段, 第一个字节是部件号

欲得知错误是发生在块设备还是字符设备上, 查看属性字段的比特15 (在BP:SI + 4处的字)。

如果得到字符设备的名字, 查看由BP:SI + 10开始的8个字节。

六、25H绝对磁盘读

它将控制直接传送给设备驱动程序。返回时, 原来的标志仍然在堆栈中 (由INT指令安放在那里的)。这是必要的, 因为返回时, 是把这些信息送回当前标志的。应确实保证这些信息从堆栈中弹出以防止失去控制。请求格式如下:

(AL) 驱动器号 (例如, 0 = A, 1 = B)

(CX) 欲读的扇区数

(DX) 起始逻辑扇区号

(DS:BX) 传送地址

在给定的驱动器和传送地址之间进行传送所指定的扇区数。逻辑扇区号是从0道、0磁头、1扇区（逻辑扇区0）开始顺序编号，在同一磁头上往下编，接着到下一磁头，直到磁道上最后一个磁头的最后一个扇区，然后再编下一磁道。由此，逻辑扇区1为磁道0，磁头0，扇区3，等等。当一个磁道上的所有磁头都编完后，编号继续从下一个磁道的磁头0的扇区开始。注意，虽然扇区是顺序编号的（例如，上例中磁道0的扇区2和扇区3），但由于交叉访问(Interleave)，它们在磁盘上的物理位置可能并不是相邻的。还要注意双面磁盘所用的映像与DOS版本1.10所用的也不同。

此调用破坏除段寄存器以外的所有寄存器。如果传送成功，进位标志(CF)为0。如果传送未成功，CF=1，并且(AX)按如下格式指出错误。(AL)为DOS出错代码，与发出INT24H时返回给DI低字节的出错代码相同，(AH)包含：

- 80H 设备(Attachment)响应失败
- 40H 寻道(SEEK)操作失败
- 08H 磁盘读时CRC不正确
- 04H 所请求的扇区未找到
- 03H 企图对写保护的磁盘作写操作
- 02H 其他错误

七、26H绝对磁盘写

此向量与上述中断25H相对应，不同的是这是写操作，其余相同。

八、27H程序结束但驻留

此向量提供给当COMMAND.COM重新获得控制后仍需维持驻留内存的程序用。

DOS功能调用31H是使程序驻留的更好的方法，因为它允许返回的信息被传送，且允许驻留比较之64K更大的程序。该程序初始化后，必须将DX设置为程序的最后地址加上1，（相对于程序的初始DS或ES值，加上一个偏移量使其他程序可被装入），然后执行中断27H。DOS便将此程序看作是DOS的扩展，从而当其他程序被执行时此程序不会被覆盖掉。这种概念对于要装入用户所编写的这类必须维持驻留的中断处理程序是非常有用的。

注：

(1) 此中断不可为装入内存高端地址的EXE程序所使用。

(2) 此中断以与INT20H相同的方式恢复中断22H，23H和24H向量。因此，不可用来安装永久性驻留的Ctrl-Break或致命错误处理子程序。

(3) 用此方法可实现驻留的最大内存量是64K。

(4) 如果在程序结束前，包含有环境副本的块被解除分配，则可以更有效地使用内存。这可由包含在PSP的2CH的段装入ES，然后发出功能调用49H(Free Allocated Memory)来实现。

(5) DOS功能调用用4CH允许打算结束的程序向DOS传送一个完成代码（或出错代码），传送的代码可在批处理过程中解释（见功能调用31H）。

(6) 结束但驻留程序并不关闭文件。

九、28H—2EH为DOS保留

这些中断保留给DOS使用。

十、2FH多路中断

中断2FH为多路中断,为在两个进程之间定义通用接口。具体使用中断2FH的应用程序来定义具体的功能和参数。

每个多路中断处理程序分配给一个具体的多路处理号。多路处理号在AH寄存器中指定。该处理程序执行的具体功能在AL寄存器中指定。其他参数按需要安排在其他寄存器中。处理程序链接到中断2FH中断向量中,检查多路处理号看是否有其他应用程序正在使用相同的多路处理号。没有预先指定的给处理程序分配多路处理号的方法,必须只使用一个。为了避免由于两个应用程序选择了相同多路处理号而产生的冲突,某个应用程序所使用的多路处理号应当是可“修补”(patchable)的。

多路处理号AH=0到AH=7FH为DOS所保留。应用程序应使用多路处理号80H到FFH。

注:在中断24H“链”中,如果你的代码调用了DOS或借助中断执行,则代码必须是可重入(reentrant)/递归(recursive)的。

1. 功能代码

下表中所包含的功能代码可由你在AL中指定以执行一个具体功能。

功能代码	说 明
0	获得已安装的状态
1	提交 (submit) 文件
2	撤消文件
3	
4	状态 (status)
5	状态结束

2. 2FH 出错代码

下表所包含的出错代码是由 INT 2FH 返回的。

出错代码	说 明
1	非法功能
2	文件未找到
3	路径未找到
4	打开的文件太多
5	拒绝访问
8	队列满
9	忙
12	名字太长
15	非法驱动器

AH = 1 为 PRINT 的驻留部分，它具有如下功能：

(1) AL = 0 获得已安装状态

此调用必须被所有中断 2FH 处理程序所定义。处理程序的调用者用其来确定该处理程序是否存在。进入时 AL = 0；

返回时AL包含有已安装的状态:

AL = 0 未安装, 准备好去安装

AL = 1 未安装, 未准备好安装

AL = FF 已安装

(2) AL = 1 提交文件

进入时, AL = 1, AH = 1, DS = DX指向提交报文分组。提交报文分组包含有级别(一个字节)和一个指向ASCII字符串的指针(段、段内偏移量形式的双字)。该ASCII字符串必须包含有驱动器、路径、打算打印的文件的文件名等。此文件名不可包含全局文件名字符。

(3) AL = 2 撤消文件

进入时, AL = ?, DS:DX指向打算撤消之打印文件的ASCII字符串, 此文件名中允许出现全局文件名字符。

(4) AL = 3 撤消所有文件

进入时, AL = 3, AH = 1。

(5) AL = 4 状态(status)

此调用维持住打印队列中的作业, 使你能够扫视该队列, 发出任何其他代码均释放作业。进入时, AL = 4。返回时, DX包含出错计数器。DS:SZ指向打印队列。该打印队列由一连串文件名登记项组成。每个登记项为64字节长。队列中的第一个登记项为当前正在被打印的文件。队列末尾由一个以空作为第一个字符的队列登记项来标记。

(6) AL = 5 状态结束(End of status)

发出此调用以释放来自调用4的队列。进入时, AL = 5, AH = 1。返回时, AX包含有出错代码, 有关出错代码的解释见“2FH出错代码”一节。

(7) AL2F8—FF为DOS保留

3.2FH处理程序列

```
MYNUM EQU X ; X = 指定的AH多路处理号
INT_2F_NEXT DD ? ; 链接位置
INT_2F: ;
ASSUME DS:NOTHWG,ES:NOTHING,SS:NOTHING
CMP AH, MYNUM
JE MINE
JMP INT_2F_NEXT ; 链接到下一个2FH处理程序
MINE:
CMP AL, 0F8H
JB DO_FUNC
IRET ; IRET到保留的功能
DO_FUNC:
OR AL, AL
JNE NON_INSTALL ; 不是“获得已安装状态”请求
MOV AL, 0FFH ; 通知“我在这里”
IRET ; 结束
NON_INSTALL ;
```

4. 安装处理程序

```
MOV AH, MYNUM
```

```

XOR AL, AL
INT 2FH ; 询问是否已安装?
OR AL, AL
JZ OK INSTALL
BAD INSTALL; ; 处理程序已安装
OK-INSTALL; ; 安装自己的处理程序
MOV AL, 2FH
MOV AH, GET_INTERRUPT VECTOR
INT 21H ; 获得多路处理号
MOV WORD PTR INT_2F_NEXT + 2, ES
MOV WORD PTR INT_2F_NEXT, BX
MOV DX OFFSET INT_2F
MOV AL, 2FH
MOV AH, SET_INTERRUPT_VECTOR
INT 21H ; 设置多路处理向量
.
.
.

```

十一、30H—3F为DOS所保留

这些中断保留给DOS使用。

第六节 功能调用

DOS提供了各种各样的功能调用用于字符设备输入/输出, 文件管理, 内存管理, 日期和时间功能, 执行其他程序

等等。这些功能调用可分为以下的组别（调用号为十六进制数）：

十六进制数	意 义
0	程序结束
1 - C	原有字符设备I/O
D - 24	原有文件管理
25 - 26	原有非设备功能
27 - 29	原有文件管理
2A - 2E	原有非设备功能
2F - 38	扩展功能组
39 - 3B	目录组
3C - 46	扩展文件管理组
47	目录组
48 - 4B	扩展内存管理组
4C - 4F	扩展功能组
54 - 57	扩展功能组
59 - 5C	扩展功能组
5E - 5F	网络功能组
62	扩展功能组

一、功能调用清单

00H	程序结束
01H	键盘输入
02H	显示输出
03H	辅助输入
04H	辅助输出

05H	打印机输出
06H	直接控制台I/O
07H	不带反弹的直接控制台输入
08H	不带反弹的控制名输入
09H	打印字符串
0AH	带缓冲的键盘输入
0BH	检查标准输入状态
0CH	清键盘缓冲区, 调用键盘功能
0DH	磁盘重置
0EH	选择磁盘
0FH	打开文件
10H	关闭文件
11H	搜寻第一个目录项
12H	搜寻下一个目录项
13H	删除文件
14H	顺序读
15H	顺序写
16H	建立文件
17H	文件换名
18H	DOS保留
19H	当前磁盘
1AH	设置磁盘传送地址
1BH	分配表信息
1CH	专用设备的分配表信息
1DH	DOS保留
1EH	DOS保留

1FH	DOS保留
20H	DOS保留
21H	随机读
22H	随机写
23H	文件大小
24H	设置相对记录字段
25H	设置中断向量
26H	生成新程序段
27H	随机块读
28H	随机块写
29H	文件名语法分析
2AH	获得日期
2BH	设置日期
2CH	获得时间
2DH	设置时间
2EH	设置/重设置校验开关
2FH	获得磁盘传送地址
30H	获得DOS版本号
31H	结束过程并驻留
32H	DOS保留
33H	Ctrl-Break检查
34H	DOS保留
35H	获得向量
36H	获得磁盘自由空间
37H	DOS保留
38H	设置或获得与国籍有关的信息

- 39H 生成子目录(MKDIR)
- 3AH 除去子目录(RMDIR)
- 3BH 改变当前目录(CHDIR)
- 3CH 建立文件(CREAT)
- 3DH 打开文件
- 3EH 关闭文件句柄
- 3FH 从文件或设备读
- 40H 向文件或设备写
- 41H 从指定目录删除文件(UNLINK)
- 42H 移动文件读/写指针(LSEEK)
- 43H 改变文件方式(CHMOD)
- 44H 设备的I/O控制(IOCTL)
- 45H 复制文件句柄(DUP)
- 46H 强使复制文件句柄(FORCDUP)
- 47H 获得当前目录
- 48H 分配内存
- 49H 释放已分配的内存
- 4AH 修改已分配的内存块(SETBLOCK)
- 4BH 装入或执行一个程序(EXEC)
- 4CH 结束一个过程(EXIT)
- 4DH 获得子过程的返回代码(WAIT)
- 4EH 寻找第一个匹配的文件
- 4FH 寻找下一个匹配的文件
- 50H DOS保留
- 51H DOS保留
- 52H DOS保留

53H	DOS保留
54H	获得校验设置
55H	DOS保留
56H	文件换名
57H	获得/设置文件的日期和时间
58H	DOS内部使用
59H	获得扩展错
5AH	建立临时文件
5BH	建立新文件
5CH	LOCK/UNLOCK文件访问
5DH	DOS保留
5E00H	获得机器名
5E02H	建立打印机设置
5E03H	获得打印机设置
5F02H	获得重定向清单表项
5F03H	设备重定向
5F04H	撤消重定向
60H	DOS保留
61H	DOS保留
62H	获得PSP地址

二、DOS内部堆栈

DOS接受控制后，便转接到一个内部堆栈。保留下用户的寄存器，除非按特殊请求将信息传送回请求者。用户堆栈必须足够大，以容纳下中断系统。最好使用户堆栈比用户所需要的再多80H字节。

三、出错返回信息

许多功能调用，在操作成功后，所返回的进位标志为CF=0。如果遇到有出错条件，则进位标志CF=1。

如果使用的是DOS版本2.10。应检查所返回出错代码，详细解释见本章中“DOS2.10出错代码”一节。

如果使用的是DOS版本3.00或3.10，使用“获得扩展错”功能调用便返回有关错误代码的附加信息。详细内容参见“获得扩展错”一节。

四、DOS2.10出错代码

如果使用DOS版本2.10的功能调用38H—57H，检查是否遇有错误，可检查AX寄存器中的下列出错代码。

功能调用号	出错代码	功能调用号	出错代码
38H	2	44H	1, 3, 5, 6
39H	3, 5	45H	4, 6
3AH	3, 5, 15	46H	4, 6
3BH	3	47H	15
3CH	3, 4, 5	48H	7, 8
3DH	2, 3, 4, 5, 12	49H	7, 9
3EH	6	4AH	7, 8, 9
3FH	5, 6	4BH	1, 2, 3, 5, 8, 10, 11
40H	5, 6	4EH	2, 3, 18
41H	2, 3, 5	4FH	18
42H	1, 6	56H	2, 3, 5, 17
43H	1, 2, 3, 5	57H	1, 6

五、获得扩展错(DOS2.00和3.10)

“获得扩展错”功能调用(59H)用于提供一组公用的 出错代码,并为有关应用程序的出错提供更为广泛的信息。从功能调用59H返回的信息,除了出错代码外,还有出错类别(error class),出错地点(locus),以及建议的动作。出错类别提供有关出错类型(硬件的、内部的、系统的等等)的信息。出错地点提供有关失败所涉及到的区域(串行设备、块设备、网络或内存)。建议的动作为不了解具体出错代码的 程序提供一种默认动作。

从现在起所写的程序可由中断24H硬错误处理程序和 在任何中断21H 功能调用之后得到扩展错的支持。

FCB功能调用由在AL中返回FFH报告出错。句柄(Handle)功能调用由设置进位标志CF = 1并在AX中返回出错代码来报告出错。DOS2.00和DOS2.10的中断21H句柄(Handle)功能调用连续地返回出错代码1—18。中断24H句柄(Handle)功能调用连续地返回出错代码0—12。而应用程序可由发出功能调用59H得到扩展出错代码表中列出的任何出错代码。DOS3.00和3.10的句柄(Handle)功能调用能够返回任何出错代码。然而,推荐用功能调用59H以得到出错类别、地点和建议的动作。

为了主成一个公用的出错表,来自中断24H的出错代码0—12对应于扩展出错代码表中出错代码19—31。当中断24H 出错处理程序中指定的是FAIL选择时,发出功能调用59H返回出错代码83(中断24H的FAIL)。

扩展错代码如下分组:

- 0 无错
- 01—18 DOS2.00/2.10INT21H错误的映像
- 19—31 DOS2.00/2.10INT24H错误的映像
- 32—88 DOS3.00/3.10的错误

注：如果遇到了你不认识的扩展出错代码，应执行本章中建议的操作以获得更多的信息。

1. 扩展出错代码

许多功能调用，当操作成功时返回进位标志CF = 0。如果遇到了出错条件，进位标志CF = 1。欲得到有关错误类别、地点和建议动作等信息，应发出获得扩展错功能调用59H。

代码	意 义
1	非法功能号
2	文件未找到
3	路径未找到
4	打开的文件太多(无剩余句柄)
5	拒绝访问
6	非法句柄
7	内有控制块被破坏
8	内存不够
9	非法内存块地址
10	非法环境
11	非法格式
12	非法访问代码
13	非法数据
14	保留

- 15 指定非法驱动器
- 16 试图删除当前目录
- 17 非同一个设备
- 18 没有更多的文件
- 19 试图对写保护磁盘作写操作
- 20 不知道的部件
- 21 驱动器未准备好
- 22 不知道的命令
- 23 数据出错(CRC)
- 24 不正确的请求结构长度
- 25 寻道错
- 26 不知道的介质类型
- 27 扇区未找到
- 28 打印机无纸
- 29 写失败
- 30 读失败
- 31 一般性失败
- 32 共享犯规
- 33 加锁犯规
- 34 非法磁盘改变
- 35 FCB无法使用
- 36 共享缓冲区溢出
- 37—49 保留
- 50 不支持网络请求
- 51 远程计算机不在倾听
- 52 网络上有重名

- 53 网络名未找到
- 54 网络忙
- 55 网络设备已不存在
- 56 超过3NetBIOS命令的限制
- 57 网络适配器硬件出错
- 58 网络响应不正确
- 59 未料到的网络
- 60 远程适配器不兼容
- 61 打印队列满
- 62 没有足够空间供打印文件用
- 63 打印文件已被删除
- 64 网络名已被删除
- 65 拒绝访问
- 66 网络设备类型不正确
- 67 网络名未找到
- 68 超过网络名限制
- 69 超过3NetBIOS对话的限制
- 70 暂停
- 71 网络请求不接收
- 72 打印或磁盘重定向暂停
- 73—79 保留
- 80 文件退出
- 81 保留
- 82 无法生成目录项
- 83 INT24失败
- 84 重定向过程过多

85	重定向重复	33
86	非法通行字	33
87	非法参数	33
88	网络设备失败	33

2. 错误类别

这里的值提供了有关错误类型的信息。

值	说 明
1	没有资源：没有空间、通道等
2	临时情况：有时可指望“情况”随时间推移而消失。注意这不是一个错误条件，而是一个“情况”，如文件上了锁等。
3	合法性：允许问题。
4	内部：系统软件的内部错误，经判断是系统软件中的问题而不是用户或系统故障。
5	硬件故障：非用户程序出错的严重故障。
6	系统故障：系统软件的严重故障，不直接导致用户失败。例如，配置文件丢失或出错。
7	应用程序出错：不一致的请求等等。
8	查找不到：文件/项目查找不到。
9	格式错：非法格式、类型或其它方面非法。不适当的文件/项目。
10	死锁：文件/项目互锁。
11	介质：介质失败(磁盘错，CRC 错……)。驱动器中磁盘错，介质上有底点等等。
12	已存在：与已存在的项目冲突，例如声明了一个

已经存在的机器名。

13 不知道：分类不存在或不合理。

3. 动作

注意这些是推荐的动作。在一些很重要的场合下，应用程序将分析出错代码并采取特别的动作，这里的默认动作用于不了解特殊出错代码的程序。

值	说 明
1	重试：重试若干次，然后提示用户已确定程序是应当继续还是流产。
2	延迟重试：暂停一段时间后重试，然后提示用户应继续还是流产。
3	用户：请用户重新输入。例如在系统调用中存在着不正确的驱动器符或文件名。当然，如果此值是写在程序中而不是由用户直接从键盘上输入的，该程序就不会请求用户重新作输入。此动作说明，如果数据来自于某个用户，最好的动作是通知该用户重新提供。
4	流产：使应用程序清空后流产。如果应用程序无法继续下去，而系统又是很正常的，则该应用程序应当遵循一定次序进行关闭。
5	立即退出：不作清空，使应用程序立即流产。一般我们不建议应用程序应试着去关闭文件，更新索引等。而应当尽可能快地退出。
6	不理睬(Ignore)
7	用户干预后重试：用户需要执行某些动作(如取出

磁盘片，放入另一张盘片等)，然后重试该操作。

4. 出错地点(Locus)

此值给出更多的信息以帮助确定失败所涉及到的区域。

值	说	明
---	---	---

- | | | |
|---|-------|---------------------|
| 1 | 不知道 | |
| 2 | 块设备： | 与随机访问大容量存储器(如磁盘)有关。 |
| 3 | 网络： | 与网络有关 |
| 4 | 串行设备： | 与串行设备有关。 |
| 5 | 内存： | 与RAM有关。 |

六、ASCIIZ字符串

有几个功能调用用接收一个ASCIIZ字符串作为输入。所谓ASCIIZ字符串由一个包含有可选的驱动器，冒号和空的终止符。

注：凡是接收路径名的功能调用都能接收符号“\”或“/”作为路径分隔字符。

七、网络路径

对于DOS3.10,如果装上了IBMPC Network网络，则有几个功能调用可接收一个网络路径作为其输入。所谓的网络路径由一个包含有计算机名，后面装一个目录路径和一个文件名的ASCII字符串组成，该字符串不可包含有一个驱动器标识符，并以一个字节的二进制0结束，例如：

\\SERVER1\LEVEL1\LEVEL2\FILE1

凡是接收ASCIIZ路径作为输入的功能调用也都接收网

络路径作为输入。不接收网络路径作为输入的两个功能调用是“改变当前目录(3BH)”和“查找第一个匹配文件(4EH)”。

八、网络访问权限

某些功能调用的解释中包含有一段称为“网络访问权限”的详注。在“网络访问权限”下的信息告知你网络上的计算机在使用DOS3.10时要执行该功能调用对于目录有什么样的访问和权限要求。例如，假定你打算执行功能调用5BH(建立新文件)，则你必须对于能够建立该文件的目录具有读/写/建立(Read/Write/Create)或写/建立(Write/Create)访问权限，如果你只有只读(Read Only)或只写(Write Only)访问权限，(无建立Create访问权限)，则你便不能在该目录中建立文件。

九、文件句柄

扩展功能调用(3CH—62H)使用了一个称之为“句柄”(handle)的标识符。当使用这些功能调用生成或打开一个文件或设备时，在AX中返回一个16位的二进制值。这便是在该文件后需引用时要用到的句柄(有时也称之为令牌token)。

下面的句柄是DOS预先定义好的并可被你的程序所使用。在使用之前无需打开它们：

十六位进制值	意 义
0000	标准输入设备。输入可被重定向。
0001	标准输出设备。输出可被重定向。
0002	标准错误输出设备，输出不可被重定向。
0003	标准辅助设备。
0004	标准打印机设备。

十、使用DOS功能

大多数功能调用都需要通过寄存器将输入传送给它们。设置合适的寄存器值之后，此功能可通过下述方式之一来使用：

(1)将功能号置于AH内，并执行一个长调用到你的程序段前缀的偏移量50H处。

(2)将功能号置于AH内，发出中断类型21H。这是使用DOS功能调用较好的方法。

(3)还有一种方法可用于使用另外的调用规矩写成的现成程序。新程序应避免使用这种方法。功能号置于CL寄存器中，而其他寄存器则根据功能的要求来设置。然后执行段内调用至当前码段的地址5处。该地址处含有一个长调用到达DOS功能调度程序(dispatcher)。如果使用这种机构总要破坏寄存器AX。此种方法仅对于功能调用(00H—24H)是合法的。

注：

(1)所用FCB功能调用均不允许有非法字符(0DH—29H)

(2)设备名不可以冒号(:)结尾

(3)AX寄存器的内容可能被任一个功能调用所改变。即使没有出错代码返回到AX，用户也不能保证AX未经改变。

(4)功能调用01H到0CH使用在“文件句柄”一节所列的标准设备。

• 00H程序结束

用途：结束一个程序的执行。

进 入	寄 存 器 内 容
AH	00H
CS	指向PSP
返 回	寄 存 器 内 容
	无

注释：此调用结束时，Ctrl-Break 和致命错误退出地址都恢复为进入此调用时的值，这些值是从程序段前缀中取出来的。所有文件缓冲区被刷新，由此过程打开的句柄均关闭。任何长度改变过的和未关闭的文件在目录中不能正确记录。控制传送给结束地址。此功能调用执行与中断20H 相同的功能。在调用此功能之前，程序应保证CS寄存器包含着程序段前缀控制块的段地址。

• 01H 键盘输入

用途：等待从标准输入设备读入一个字符(如果已准备好则不用等待)，然后将此字符反弹(echo)到标准输出设备，并将该字符返回AL。

进 入	寄 存 器 内 容
AH	01H
返 回	寄 存 器 内 容
AL	来自标准输入设备的字符

注释：检查此字符是不是Ctrl-Break，如果是Ctrl-Break，

则执行中断23H。

注：对于功能调用01H，一个扩展的 ASCII 代码需执行两次功能调用。第一个功能调用返回值00H作为指示，表明下一个调用返回的是扩展代码。

• 02H 显示器输出

用途：将DL中的字符输出到标准的输出设备。

进 入	寄 存 器 内 容
AH	02H
DL	字符

返 回	寄 存 器 内 容
	无

注释：如果DL中的字符是退格(08)，则光标向左移动一个位置(不删除)。如果在输出后检测到 Ctrl-Break，则执行中断23H。

• 03H辅助输入

用途：等待标准辅助设备送来字符，并将其返回到AL。

进 入	寄 存 器 内 容
AH	03H

返 回	寄 存 器 内 容
AL	辅助设备来的字符

注释：辅助设备(AUX, COM1, COM2)是无缓存且非

中断驱动的。

启动时, DOS将第一个辅助端口初始化为2400波德率, 无奇偶校验, 一个停止位, 并为8比特字。

辅助功能调用(03H和04H)不返回状态或出错代码。若要获得更强的控制, 建议使用ROM BIOS子程序(中断14H)或编写一个AUX设备驱动程序并使用IOCTL。

• 04H 辅助输出

用途: 将DL中的字符输出到标准辅助设备。

进 入	寄 存 器 内 容
AH	04H
DL	字符

返 回	寄 存 器 内 容
	无

• 05H 打印机输出

用途: 将DL中字符输出到标准打印机设备。

进 入	寄 存 器 内 容
AH	05H
DL	字符

返 回	寄 存 器 内 容
	无

• 06H 直接控制台I/O

用途: 如果标准输入设备中有字符已准备好, 则等待它

到来。

进 入	寄 存 器 内 容
AH	06H
DL	FFH, 控制台输入 00H—FEH 控制台输出
返 回	寄 存 器 内 容
AL	见下面叙述

注释：如果 DL 为 FFH，且字符已准备好，则 AL 中为标准输入设备来的输入字符，且零标志清 0。如果该字符未准备好，则零标志置位。

如果 DL 为非 FFH，则假定若 DL 中有一个要输出给标准输出设备的合法字符，且执行输出。此功能调用不检查 Ctrl-Break，或 Ctrl-Prtsc。

注：对于功能调用 06H，一个扩展 ASCII 代码需执行两次功能调用。第一个调用返回 00H 指示下一个调用返回的是一个扩展代码。

• 07H 不带反弹 (Echo) 的直接控制台输入

用途：等待从标准输入设备读入一个字符，然后将该字符返回给 AL。

进 入	寄 存 器 内 容
AH	07H
返 回	寄 存 器 内 容
AL	标准输入设备来的字符

注释：与功能调用06H一样，对此字符不作检查。

• 08H 不带反弹的控制台输入

用途：等待从标准输入设备读入一个字符，并将此字符返回给AL。

进 入	寄 存 器 内 容
AH	08H

返 回	寄 存 器 内 容
AL	标准输入设备来的字符

注释：检查该字符是否为Ctrl-Break。如果是Ctrl-Break，执行中断23H。

注：对于功能调用08H，一个扩展 ASCII 代码需要二次功能调用。

• 09H 打印字符串

用途：将指定的字符串中的字符输出到标准输出设备。

进 入	寄 存 器 内 容
AH	09H
DS : DX	指向该字符串

返 回	寄 存 器 内 容
	无

注释：内存中的该字符串必须以\$ (24H)结束。该字符

串中的每个字符以与功能调用02H 相同的形式输出到标准输出设备。

• 0AH带缓冲的键盘输入

用途：从标准输入设备读入字符，并将它们从缓冲区第三个字节开始存放。

进 入	寄 存 器 内 容
AH	0AH
DS:DX	指向输入缓冲区

返 回	寄 存 器 内 容
	无

注释：输入缓冲区的第一个字节指出该缓冲区能够容纳的字符数。此值不可为0。该调用不断地对标准输入设备读入字符并填入缓冲区，直到读到Enter。如果缓冲区已填至距最大可容纳字符数还少一个时，多余的字符将不再接纳，并响铃，直到读到Enter。在此调用结束后，缓冲区的第二个字节设置为已接收到的字符数，但不包括作为最后一个输入字符的回车(0DH)等。

• 0BH检查标准输入状态

用途：检查标准输入设备是否有字符要输入。

进 入	寄 存 器 内 容
AH	0BH

返 回	寄 存 器 内 容
AL	FFH 如果标准输入设备有字符要输入
	00H 如果标准输入设备没有字符要输入

注释：如果标准输入设备有字符要输入，AL为FFH；否则AL为00H。如果检查到 Ctrl-Break，执行中断 23H。

• 0CH 清键盘缓冲区并调用键盘功能

用途：清除标准输入缓冲区，然后执行 AL 中的功能调用号（只允许是01H，06H，07H，08H和0AH）所规定的功能调用。

进 入	寄 存 器 内 容
AH	0CH
AL	功能号

返 回	寄 存 器 内 容
	无

注释：迫使系统等待到字符被打入。

• 0DH 磁盘重置

用途：刷新所有文件缓冲区。

返 回	寄 存 器 内 容
	无

注释：文件长度已经改变过但又未关闭的文件在磁盘目

录中不能正确被记录。

• 0EH 选择磁盘

用途：选择DL中所指定的驱动器（0 = A, 1 = B 等）作为默认驱动器。

进 入	寄 存 器 内 容
AH	0EH
DL	驱动器号 (0 = A, 1 = B 等)

返 回	寄 存 器 内 容
AL	总的驱动器数

注释：AL 中返回总的驱动器数（总的软盘和硬盘驱动器数）。对于DOS3.00和3.10, AL中返回的最小值是5。如果系统只有一个软盘驱动器，被计数为 2 以符合系统相当于有两个逻辑驱动器A和B的想法。

• 0FH 打开文件

用途：搜寻当前目录中的命名文件，如果未找到，则AL中返回FFH。如果找到了，AL返回00H，且FCB按如下格式填写。

进 入	寄 存 器 内 容
AH	0FH
DS : DX	指向未打开的FCB

返 回	寄 存 器 内 容	
AL	00H	如果文件已打开
	FFH	如果文件未打开

注释：如果驱动器代码是0（默认驱动器），则改变到实际所有的驱动器而不打扰对此文件的后续操作。当前块字段（FCB字节C—D）设置为0，所用的记录大小（FCB字节E—F）设置为系统默认值80H。FCB中的文件大小和日期根据目录中得到的信息进行设置。可改变记录大小的默认值（FCB字节E—F），或设置为随机记录大小及当前记录字段。上述的这些改变要在文件打开后，但在磁盘操作前进行。文件从兼容方式打开。所谓兼容方式的解释见功能调用3DH。

• 10H 关闭文件

用途：对文件作写操作后关闭该文件。

进 入	寄 存 器 内 容	
AH	10H	
DS:DX	指向打开的FCB	

返 回	寄 存 器 内 容	
AL	00H	如果文件被找到
	FFH	如果在当前目录中未找到该文件

注释：此功能调用必须在对文件作完写操作以保证所有目录信息被更新并且该文件不再需要的情况下使用。如果文件在当前目录的正确位置处未找到，假定已更换了磁盘片，

且在AL中返回FFH。否则便修改目录以反映FCB中的状态，刷新用于该文件的缓冲区，AL中返回00H。

• 11H 搜寻第一个匹配的目录项

用途：搜寻第一个匹配的文件名。

进 入	寄 存 器 内 容
AH	11H
DS:DX	指向未打开的FCB

返 回	寄 存 器 内 容
AL	00H 如果找到了匹配文件 FFH 如果未找到匹配文件

注释：在当前磁盘目录中搜寻第一个匹配文件。如果未找到，在AL中返回FFH。对于DOS2.10，文件名中允许出现“?”符号；对于DOS3.00和3.10，允许出现全程文件名字符。如果找到了匹配的文件名，AL中返回00H，并且磁盘传送地址处的位置按如下设置：

(1) 如果为搜寻所提供的FCB是一个扩展的FCB，则磁盘传送地址的第一个字节被设置为FFH，后面跟着5个字节的0，然后是搜寻FCB的属性字节，所用的驱动器号(1=A, 2=B等)，最后是32个字节的目录登记项。这样，磁盘传送地址处包含有一个具有与要搜寻FCB相同属性的合法的未打开扩展FCB。

(2) 如果用于搜寻的FCB是一个标准FCB，则第一个字节被设置为所用的驱动器号(1=A, 2=B)，后面32个

字节包含有匹配的目录登记项。这样，磁盘传送地址包含有一个合法的未打开的标准FCB。

注：如果使用的是扩展FCB，则使用下面的搜寻模式：

① 如果FCB的属性字节是0，仅找到了普通文件登记项。不返回有关卷标、子目录、隐式文件和系统文件的登记项。

② 如果属性字段被设置为隐式或系统文件，或目录登记项，则被看作是一个“或”搜寻。所有的普通文件登记项加上与指定属性相匹配的登记项均被返回。欲查看除卷边外的所有目录登记项，属性字节可被设置为“隐式 + 系统 + 目录”（这3个位均为ON）。

③ 如果属性字段被设置为卷标，则被看作是排他性搜索，仅返回卷标登记项属性位在“DOS 磁盘目录一节定义”。

• 12H 搜寻下一个匹配的目录项

用途：在当前目录中搜寻下一个匹配的文件名。

进 入	寄 存 器 内 容
AH	12H
DS:DX	指向由上一次功能调用11H或功能调用12H所指定的未打开的FCB

返 回	寄 存 器 内 容
AL	00H 找到了匹配的文件
	FFH 未找到匹配的文件

注释：使用功能调用 11H 找到了一个匹配的文件名之后，可调用功能12H以寻找下一个匹配的目录项。对于DOS

2.10, 在文件名中允许使用“?”符号; 对于DOS3.00和3.10文件名中允许出现全程文件名。磁盘传送地址DTA中包含有上一次“搜寻第一个(11H)”或“搜寻下一个(12H)”得到的信息。FCB中除名字/扩展名字段外全部用于存放继续搜寻所需要的信息, 故在上一次功能调用11H或12H与本次功能调用之间不可执行磁盘操作。

• 13H 删除文件

用途: 删除所有与指定文件名相匹配的当前目录中的目录项。所指定的文件名不可以是只读的。

进 入	寄 存 器 内 容
AH	13H
DS:DX	指向一个未打开的FCB

返 回	寄 存 器 内 容
AL	00H 文件被删除
	FFH 未找到匹配的目录登记项

注释: 删除所有匹配的当前目录登记项。文件名中允许出现全程文件名字符“?”。如果没有匹配的目录登记项, AL中返回FFH; 否则AL中返回00H。如果指定文件为只读方式, 则不删除。

注: 在删除文件前应先关闭打开着的文件。网络访问权限: 要求有建立(create)访问权限。

• 14H 顺序读

用途：根据当前块（FCB字节C—D）和磁盘传送地址DTA的当前记录（FCB字节IF）提供的地址装入记录，然后使记录地址加1。

进 入	寄 存 器 内 容	
AH	14H	
DS:DX	指向一个打开的FCB	

返 回	寄 存 器 内 容	
AL	00H	读操作成功完成
	01H	遇到EOF（无数据可读）
	02H	DTA太小（撤消读操作）
	03H	读入部分记录然后填以0

注释：记录长度由FCB记录大小字段确定。网络访问权限：需要读（Read）访问权限。

• 15H 顺序写

用途：根据当前块和DTA的记录字段（由FCB记录大小字段决定其大小）提供的地址写记录。如果记录小于一个扇区的大小，则该记录进入缓冲区等待扇区中的数据被累加。然后使记录地址加1。

进 入	寄 存 器 内 容	
AH	15H	
DS:DX	指向一个打开的FCB	

返 回	寄 存 器 内 容
AL	00H 写操作成功完成
	01H 磁盘已满(撤消写操作)
	02H DTA太小(撤消写操作)

注释：如果所指定的文件为只读方式，则不执行顺序写操作。网络访问权限：要求有写(Write)访问权限。

• 16H 建立文件

用途：在指定驱动器的当前目录中搜寻匹配的登记项。

进 入	寄 存 器 内 容
AH	16H
DS:DX	指向一个未打开的FCB

返 回	寄 存 器 内 容
AL	00H 建立了文件(找到了匹配的登记项或找到了完了登记项)
	FFH 未建立文件(目录或磁盘或未找到匹配的目录登记项)

注释：找到了一个匹配的登记项，但重新使用它。如果未找到，便搜索目录以找查空登记项。如果找到了一个匹配的登记项，但将它初始化为一个0长度的文件，该文件被打开(见功能调用0FH)，且AL返回00H。

文件在建立时可由使用包含有适当属性字节的扩展FCB使之标记为隐式文件。网络访问权限：要求有建立(Create)

访问权限。

• 17H 文件换名

用途：将指定驱动器当前目录中与第一个文件名相匹配的每一个文件改变为第二个文件名（限制是这两个文件不可以具有相同的名字和扩展名）。

进 入	寄 存 器 内 容
AH	17H
DS:DX	指向修改过的FCB

返 回	寄 存 器 内 容
AL	00H 文件换了名（找到了匹配的文件名）
	FFH 未找到匹配的文件名或如果企图换成为一个已存在的文件名

注释：修改过的FCB在通常位置上具有驱动器代码和文件名，在第一个文件名后面6个字节处(DS:DX + 11H)，此处通常为保留区域，开始第二个文件名。如果第二个名字中间出现有“?”符号，则原名字相应位置处的字符不改变。

如果文件是以只读方式指定的，则该文件不换名。

网络访问权限：要求有“建立”访问权限。

• 19H 当前磁盘

用途：确定当前的默认驱动器。

进 入	寄 存 器 内 容
AH	19H

返 回	寄 存 器 内 容
AL	当前默认驱动器 (0 = A, 1 = B等)

注释: AL中返回当前默认驱动器的代码 (0 = A, 1 = B等)。

• 1AH 设置磁盘传送地址

用途: 将磁盘传送地址设置到DS:DX。

进 入	寄 存 器 内 容
AH	1AH
DS:DX	磁盘传送地址

注释: DOS不允许磁盘传送程序在段内“回绕”(wrap-around)或溢出到下一段。如果没有设置DTA,则默认DTA在程序段前缀中偏移量80H处。

注: 可由使用功能调用2FH来获得DTA。

• 1BH 分配表信息

用途: 返回有关默认驱动器分配表的信息。

进 入	寄 存 器 内 容
AH	1BH

返 回	寄 存 器 内 容
DS:DX	指向默认驱动器的介质描述符字节
DX	分配单元数
AL	扇区/分配单元数
CX	实际扇区大小

注释：欲了解更多的有关 DOS 磁盘分配的信息，可参考“DOS 磁盘目录”一节，还可以参考功能调用 36H(“获得磁盘自由空间”)。

• 1CH 特殊设备的分配表信息

用途：返回有关特殊设备的分配表信息。

进 入	寄 存 器 内 容
AH	1CH
DL	驱动器号
返 回	寄 存 器 内 容
DS : DX	指向DL中指定驱动器的介质描述符字节
AL	扇区/分配单元数
DX	分配单元数
CX	实际扇区大小

注释：此调用与调用 1BH 相同，只是DL中包含有含有所需信息的驱动器的号码(0 = 默认, 1 = A等)。欲了解更多的有关 DOS 磁盘分配的信息，请参考“DOS 磁盘目录”一节，也可参考功能调用36H。

• 21H 随机读

用途：将当前块和当前记录字段确定地址的记录读入内存中当前磁盘传送地址。

进 入	寄 存 器 内 容
AH	21H
DS : DX	指向一个已打开的FCB

返 回	寄 存 器 内 容
AL	00H 读操作成功完成
	01H EOF (无数据可读)
	02H DTA 太小 (撤消读操作)
	03H EOF (读入了部分记录但均为0)

注释：当前块和当前记录字段设置与随机记录字段相一致。然后由这些字段确定地址的记录被读入到内存的当前磁盘传送地址。

网络访问权限：需要有“读”访问权限。

• 22H 随机写

用途：从当前磁盘传送地址的当前块和当前记录字段确定地址作记录写操作。

进 入	寄 存 器 内 容
AH	22H
DS:DX	指向一个已打开的FCB

返 回	寄 存 器 内 容
AL	00H 写操作成功完成
	01H 磁盘已满 (写操作撤消)
	02H DTA 太小 (写操作撤消)

注释：当前块和当前记录字段被设置为与随机记录字段一致。然后从磁盘传送地址处对由这些字段确定地址的记录作写操作(如果记录与扇区大小不同，则缓冲)。

如果该文件以只读方式被指定，便不执行随机写。

网络访问权限：要求“写”访问权限。

• 23H 文件大小

用途：在磁盘目录中搜录与指定文件相匹配的登记项并将FCB随机记录字段设置为文件中的记录数。

进 入	寄 存 器 内 容
AH	23H
DS:DX	指向一个未打开的FCB

返 回	寄 存 器 内 容
AL	00H 如果找到了目录登记项
	FFH 如果找到目录登记项

注释：在磁盘目录中搜寻匹配的登记项。如果找到了匹配的登记项，则随机记录字段设置为文件中的记录数(依据记录大小字段四舍五入)如果未找到匹配的登记项，AS 返回 FFH。

注：如果在使用此功能之前未设置过 FCB 记录大小字段，返回的是不正确的信息。

• 24H 设置相对记录字段

用途：将随机记录字段设置为与当前块和当前记录字段相同的文件地址。

进 入	寄 存 器 内 容
AH	24H
DS:DX	指向一个打开的FCB

返 回	寄 存 器 内 容
	无

注释：在执行随机读、写和随机块读、写前必须调用此功能。

• 25H 设置中断向量

用途：设置中断向量表中的中断号。

进 入	寄 存 器 内 容
AH	25H
DS:DX	中断处理程序地址
AL	中断号

返 回	寄 存 器 内 容
	无

注释：在 AL 中指定的中断向量表设置为 DS:DX 中所包含的地址。使用功能调用 35H (Get Vector“获得向量”) 以获得该中断向量的内容。

• 26H 建立新的程序段

用途：建立一个新的程序段

进 入	寄 存 器 内 容
AH	26H
DX	新程序段的段号

返 回	寄 存 器 内 容
	无

注释：当前程序段位置0处整个100H 区域被拷贝到新程序段中地址0处。新段中位置6处的内存大小信息被更新，并且中断向量表中有关中断22H，23H，24H处的当前结束 Ctrl-Break 退出和致命错误地址被存放到新程序段0AH 开始处。在程序结束时，它们从此区域中恢复出来。

注：应避免使用此调用，而代之以使用 EXEC 功能调用4BH。

• 27H 随机块读

用途：从随机记录字段所指定的文件地址将指定数量的记录(按照记录大小字段内容)读到磁盘传送地址处。

进 入	寄 存 器 内 容
AH	27H
DS:DX	指向一个已打开的FCB
CX	欲读出的记录数

返 回	寄 存 器 内 容
AL	00H 如果读操作完成
	01H 如果EOF (无数据读)
	02H 如果DTA 太小 (撤消读操作)
	03H 如果EOF (读入记录被读入但均为0)
CX	实际读入的记录数

注释：设置随机记录字段和当前块/记录字段为下一个

记录(即：第一个未读记录)的地址。

网络访问权限：要求读访问权限。

• 28H 随机块写

用途：从磁盘传送地址中，把指定数量的记录写入到随机记录字段所指定的文件地址。

进 入	寄 存 器 内 容
AH	28H
DS:DX	指向一个已打开的FCB
CX	欲写出的记录数

返 回	寄 存 器 内 容
AL	00H 写操作完成
	01H 磁盘已满(写操作取消)
	02H DTA太小(写操作取消)
CX	

注释：如果磁盘上空间不够，AL 返回01H 且无记录被写。如果 CX 为0，则无记录被写，但该文件按照随机记录字段设置文件长度，而不管是比当前文件大小还是长短(按情况进行单元的分配和释放)。

网络访问权限：要求“写”访问权限。

• 29H 文件名语法检查

用途：对指定的文件名进行语法检查。

进 入	寄 存 器 内 容
AH	29H
DS:SI	指向要进行语法检验的命令行
ES:DI	指向将以一个未打开FCB填写的内存部分
AL	其各位控制语法检验操作

返 回	寄 存 器 内 容
AL	00H 命令行中没有全程文件名字符
	01H 命令行中使用了全程文件名字符
	FFH 驱动器标识符非法
DS:SI	指向语法检验后文件名的第一个字符
ES:DI	指向格式化后FCB的第一个字节

注释: AL 的内容用于确定将采取的动作:

位<必须=0>

7 6 5 4 3 2 1 0

如果位0=1, 则 DS:SI 处命令行上的前导分隔符被检出, 否则不检出前导分隔符。

如果位1=1, 则仅当在语法检查之命令行上的驱动器被指定时, 结果 FCB中的驱动器 ID 字节被置位(被改变)。

如果位2=1, 则仅当命令行上包括有一个文件名时, FCB中的文件名才被改变。

如果位3=1, 则仅当命令行上包含有一个文件扩展名

时，才改变 FCB 中的文件扩展名。

文件名分隔符包括下列字符：“，，= +”和TAB,SPACE。文件名结束符包括所有上述字符以及“<,>,|,/,”,[,],”和所有控制字符。对命令行进行语法检查看文件名形式是否为“d:文件名.扩展名”，如果找到了，则在 ES:DI 处建立一个未打开的 FCB。如果没有驱动器标识符，则假定为全空。如果字符“*”在文件名或扩展名中出现，则文件名或扩展名中该字符以及其后所有字符设置成“?”。

如果文件名或扩展名中出现了“?”或“*”，AL 返回 01H；如果有驱动器标识符，AL 返回 FFH；否则返回 00H。

返回时 DS:SI 指向文件名后第一个字符，而 ES:DI 指向格式化后FCB的第一个字节。如果不存在合法的文件名，ES:DI+1 包含一个空格。

• 2AH 获得日期

用途：返回星期几、年、月和日期。

进 入	寄 存 器 内 容
AH	2AH

返 回	寄 存 器 内 容
AL	星期几 (0 = SUN, 6 = SAT)
CX	年 (1980—2099)
DH	月 (1—12)
DL	日期 (1—31)

注释：如果日历时钟走到了下一天，则应相应调整日

期、计及每个月的天数以及是否为闰年。

• 2BH 设置日期

用途：设置日期

进 入	寄 存 器 内 容
AH	2BH
CX	年 (1980—2099)
DH	月 (1—12)
DL	日期 (1—31)

返 回	寄 存 器 内 容
AL	00H 如果日期合法 FFH 如果日期不合法

注释：进入时，CX:DX 必须具有与功能调用2AH所返回的相同格式的合法日期。

返回时，如果日期合法，且设置操作成功，AL 返回 00H；如果日期不合法，则 AL 返回 FFH。

• 2CH 获得时间

用途：返回时间、小时、分、秒和百分秒。

进 入	寄 存 器 内 容
AH	2CH

返 回	寄 存 器 内 容
CH	小时 (0—23)
CL	分 (0—59)
DH	秒 (0—59)
DL	百分秒 (0—99)

注释：进入时，AH 中包含有 2CH。返回时，CX:DX 包含有一日时间。时间实际上从四个 8 位的量进行表示的。CH 具有小时 (0—23)，CL 具有分 (0—59)，DH 具有秒 (0—59)，DL 具有 1/100 秒 (0—99)。这种格式可被很快地转换成为一种可打印格式，然后用于计算，例如两个时间值相减。

• 2DH 设置时间

用途：设置时间。

进 入	寄 存 器 内 容
AH	2DH
CH	小时 (0—23)
DH	秒 (0—59)
CL	分 (0—59)
DL	百分秒 (0—99)

返 回	寄 存 器 内 容
AL	00H 如果时间合法
	FFH 如果时间不合法

注释：进入时，CX:DX 的时间格式与功能2CH 所返回的一样。返回时，如果时间的任一分量不合法，设置操作便流产，且AL返回FFH。如果时间合法，AL返回00H。

• 2EH 设置/重置验证开关

用途：设置验证开关。

进 入	寄 存 器 内 容
AH	2EH
AL	00H 设置“验证”off
	01H 设置“验证”on
返 回	寄 存 器 内 容
	无

注释：进入时，AL必须包含01H以打开 Verify(验证)，或00H以关闭Verify。Verify为ON时，DOS在每次作磁盘写操作时都要执行一次验证操作以确保数据记录正确。虽然磁盘记录操作很少出错，但对关键数据有时应用程序还是希望能进行一下验证。可通过功能调用54H来获得验证开关的当前设置。

注：对网络上磁盘作写操作时不支持对于数据的验证。

• 2FH 获得磁盘传送地址(DTA)

用途：返回当前磁盘传送地址。

进 入	寄 存 器 内 容
AH	2FH

返 回	寄 存 器 内 容
ES : BX	当前DTA

注释：进入时，AH包含有2FH。返回时，ES:BX包含有当前磁盘传送地址。可以使用功能调用1AH来设置DTA。

• 30H获得DOS版本号

用途：返回DOS版本号。

进 入	寄 存 器 内 容
AH	30H

返 回	寄 存 器 内 容
BX	0000H
CX	0000H
AL	主版本号
AH	辅版本号

注释：进入时，AH中包含有30H。返回时，BX和CX设置为0，AL包含有主版本号，AH包含辅版本号。例如，对于DOS3.10，主版本号是03H，辅版本号为0AH。

注：如果AL所返回的主版本号为0，则可假定DOS是DOS2.00之前的。

• 31H 结束过程并驻留

用途：结束当前过程并试图将初始分配块设置为以段落表示的内存大小。

进 入	寄 存 器 内 容
AH	31H
AL	返回代码
DX	以段落(paragraphs)表示的内存大小
返 回	寄 存 器 内 容
	无

注释：进入时，AL 包含有一个二进制返回代码。DX 包含有从段落表示的内存大小值。这功能调用不释放任何属于该过程的其他分配块。由该过程打开的文件在调用被执行时不被关闭。在 AL 中传送的返回代码可通过 Wait(功能调用ADH)进行检索，可通过 ERRORLEVEL 批处理子命令进行测试。

注：如果包含有“环境”(environment)的副本的块在结束前被解除分配，则可更有效地使用内存。这可由将包含在程序段前缀 PSP 之2C 处的段落入 ES 寄存器，并发出功能调用49H(释放已分配内存)来实现。

• 33H Ctrl-Break 检查

用途：设置或获得 BREAK(Ctrl-Break检查)的状态。

进 入	寄 存 器 内 容
AH	33H
AL	00H 请求当前状态
	01H 设置当前状态
DL	00H 设置当前状态为OFF
	01H 设置当前状态为ON

返 回	寄 存 器 内 容
DL	当前状态, 00H = OFF 01H = ON

注释：进入时，AL包含00H 请求Ctrl-Break检查的当前状态，01H用于设置状态。如果设置状态，DL包含00H表示OFF，或01H表示ON。返回时，如果请求当前状态，DL包含有当前状态(00H = OFF, 01H = ON)。

• 35H 获得向量

用途：指向中断处理子程序。

进 入	寄 存 器 内 容
AH	35H
AL	中断号

返 回	寄 存 器 内 容
ES : BX	指向中断处理子程序

注释：进入时，AH 包含35H。AL包含一个十六进制的中断号；返回时，ES:BX 包含有指定中断的CS:IP 中断向量。使用功能调用25H(设置中断向量)去设置该中断向量。

• 36H 获得磁盘自由空间

用途：返回磁盘自由空间(可用的盘束，每驱动器有多少盘束，每扇区有多少字节)。

进 入	寄 存 器 内 容
AH	36H
DL	驱动器 (0 = 默认, 1 = A)

返 回	寄 存 器 内 容
BX	可用盘束
DX	盘束/驱动器
CX	字节/扇区
AX	如果DL中驱动器为非法则FFFFH, 否则是每个盘束的扇区数

注释: 如果DL中的驱动器号是合法的, 则BX中包含有可分配单元(盘束cluster)的个数, DX包含该驱动器上总的盘束个数, CX包含每个扇区的字节数, AX包含每个盘束的扇区数。

注: 此功能调用与“获得FAT指针”调用(1BH)在相同的寄存器中返回相同的信息(只是FAT指针除外)。

• 38H(DOS 2.10) 返回与国籍有关的信息

用途: 返回与国籍有关的信息。

进 入	寄 存 器 内 容
AH	38 H
DS : DX	指向32字节内存区
AL	功能代码

返 回	寄 存 器 内 容
AX	如果进位标志置位, 则为错误代码。
DS : DX	如果进位标志未置值, 则为国籍数据。

注释：进入时，DS:DX 指向一个 32 字节的内存块，所返回的信息传送到那里，AL 包含一个功能代码。在 DOS 2.10 中，此功能代码必须是 0。下列信息与国际上应用有关：

WORD	日期/时间格式	
BYTE	ASC II Z	钱币符号字符串，后面跟有一个字节的 0
BYTE	ASC II Z	千分隔符字符串，后面跟有一个字节的 0
BYTE	ASC II Z	小数点符字符串，后面跟有一个字节的 0
保留 24 个字节		

时间和日期格式具有如下的值和意义：

- 0 = 美国标准 时:分:秒 月/日/年
- 1 = 欧洲标准 时:分:秒 日/月/年
- 2 = 日本标准 时:分:秒 日:月:年

• 38H(DOS3.00和3.10) 获得或设置有关国籍的信息
用途：返回有关国籍的信息。

(1) 获得当前国籍

进 入	寄 存 器 内 容
AH	38H
DS:DX	指向数据将被返回处的内存缓冲区
AL	00H；获得当前国籍信息 国籍代码，<255 FFH；获得的国籍代码≥255
BX	如果 AL=FFH，则为 16 位的国籍代码

返 回	寄 存 器 内 容
AX	如果进位标志置位, 则为出错代码
DS:DX	填以国籍信息(下面所描述的)
BX	国籍代码

(2) 设置当前国籍

进 入	寄 存 器 内 容
AH	38H
DX	FFFFH
AL	国籍代码 < 255 FFH, 代码 ≥ 255
BX	如果 AL = FFH, 则为 16 位国籍代码

返 回	寄 存 器 内 容
AX	如果进位标志置位, 则为出错代码

(3) 国籍信息

- 1个字 日期格式
- 5个字节 钱币符号, 以空结束
- 2个字节 千分隔符, 以空结束
- 2个字节 小数点分隔符, 以空结束
- 2个字节 日期分隔符, 以空结束
- 2个字节 时间分隔符, 以空结束
- 1个字节 比特场钱币格式

位0=0 钱币符号在值前面
=1 钱币符号在值后面

- 位1 值与钱币符号之间的空格数 (0或1)
- 1个字节 钱币中有效小数位数
- 1个字节 时间格式
- 位0 = 0 是12小时时钟
- = 1 是24小时时钟
- 2个字 大小写变换调用地址
- 2个字节 数据列表分隔符, 以空结束
- 5个字 保留

(4) 大小写变换调用地址: 用于大小写变换时的寄存器内容为:

进 入	寄存器内容
AL	要被转换成大写的字符ASCII代码

返 回	寄存器内容
AL	大写输入字符的ASCII代码

大小写变换调用地址的格式适合于直接发出长调用。

日期格式具有下列的值和意义:

代 码	日 期
0=美国	月 日 年
1=欧洲	日 月 年
2=日本	年 月 日

注释: 出错代码在AX中返回。发出功能调用59 H“获得扩展错”可得到有关出错类别、建议动作和出错地点的更多

的信息。调入另一个键盘处理程序，则该键盘程序被装入到用户内存（从用户可用内存最低地址开始装）。此子程序改变了BIOS键盘服务中断向量，使CPU重定向到新的键盘子程序所驻留的用户内存部分。每个键盘子程序约占用1.6K字节的读/写内存，并有一张ASCII值与每种语言的对照表，参见DOS参考手册中的KEYBxx命令。

一旦键盘中断向量被DOS键盘程序所改变，该中断便由读/写内存中该子程序所服务。同时按下Ctrl, Alt和F1键，则返回到美国英语键盘格式。这并不将中断向量改回到BIOS位置，此时中断仍由读/写子程序处理，但将扫描码转换为ASCII代码是在ROM中实现的。类似地有，同时按下Ctrl, Alt和F2，则返回到读/写对照表。

• 39H生成子目录(MKDIR)

用途：生成指定的目录。

进 入	寄 存 器 内 容
AH	39H
DS:DX	指向一个ASCII字符串

返 回	寄 存 器 内 容
AX	如果进位标志置位，则为出错代码

注释：进入时，DS:DX包含有一个带有驱动器和目录路径名的ASCII字符串地址。如果该目录路径某个部分不存在，则不生成此目录路径。返回时，在指定路径末端生成一个新目录。

出错代码在AX中返回。发出功能调用59H“获得扩展错”可得到有关出错类别、建议动作和出错地点的更多的信息。

网络访问权限：需要“建立”访问权限。

• 3AH删除子目录(RMDIR)

用途：删除指定的目录。

进 入	寄 存 器 内 容
AH	3AH
DS:DX	指向一个ASCII字符串

返 回	寄 存 器 内 容
AX	如果进位标志置位，则为出错代码

注释：进入时，DS:DX包含有一个带有驱动器和目录路径名的ASCII字符串地址。所指定的目录将从此结构中删除。不可删除当前目录。

AX中返回出错代码。发出功能调用59H“获得扩展错”可得到更多有关出错的信息。

网络访问权限：要求有“建立”访问权限。

• 3BH改变当前目录(CHDIR)

用途：将当前目录改变到指定的目录。

进 入	寄 存 器 内 容
AH	3BH
DS:DX	指向一个ASCII字符串

返 回	寄 存 器 内 容
AX	如果进位标志置位, 则为出错代码

注释: 进入时, DS:DX包含有一个带有驱动器和目录路径名的ASCII字符串的地址。此字符串被限制为64个字符, 且不可包含有网络路径。如果此目录路径的任一部分不存在, 则不改变目录路径。否则将当前目录设置为此ASCII字符串。

出错代码在AX中返回。发出功能调用59H, 可得到更多有关出错的信息。

• 3CH建立文件(CREAT)

用途: 建立一个新文件或将一个老文件改变为0长度以作好写操作准备。

进 入	寄 存 器 内 容
AH	3CH
DS:DX	指向一个ASCII字符串
CX	文件属性

返 回	寄 存 器 内 容
AX	如果进位标志置位, 则为出错代码 如果进位标志未置位, 则为16位的句柄

注释: 如果此文件不存在, 则在相应目录中建立此文件并给予此文件读/写访问代码。文件被打开作读/写时, 句柄在AX中返回。注意改变方式功能调用(43H)可用于改变文件的属性。

AX中返回出错代码。发出功能调用59H可得到更多的有关出错的信息。

网络访问权限：要求建立访问权限。

• 3DH(DOS 2.10) 打开文件

用途：打开指定的文件。

进 入	寄 存 器 内 容
AH	3DH
DS:DX	指向一个ASCII路径名
AL	访问代码

返、回	寄 存 器 内 容
AX	如果进位标志位置，则为出错代码 如果进位标志未置位，则为16位文件句柄

注释：此调用打开一个与所指定名字相匹配的普通文件或隐式文件。以冒号结束的文件不被打开。

读/写指针设置在该文件的第一个字节处，且该文件的记录大小为1字节(读/写指针可通过功能调用42H来改变)。所返回的文件句柄必须被后面对于该文件的输入和输出来使用。可通过功能调用57H来获得或设置此文件的日期和时间，且其属性可由调用43H获得。

访问代码：

AL = 0 打开文件做读操作

AL = 1 打开文件做写操作

AL = 2 打开文件做读、写操作

• 3DH(DOS 3.00和3.10) 打开文件

用途：打开指定的文件。

进 入	寄 存 器 内 容
AH	3DH
DS:DX	指向一个ASCIIZ路径名
AL	打开方式

返 回	寄 存 器 内 容
AX	如果进位标志置位，则为出错代码 如果进位标志未置位，为16位文件句柄

注释：读/写指针设置在文件的第一个字节处，并且文件的记录大小为1字节（可通过功能调用42H来改变读/写指针）。以后该文件的输入和输出必须用到所返回的文件句柄。可通过调用57H获得或设置文件的日期和时间，通过调用43H获得文件的属性。

出错代码在AX中返回。发出功能调用59H可得到更多有关出错的信息。

网络访问权限：如果打开方式字段(AL)中的访问字段(A)等于：

- 000 要求“读”访问权限
- 001 要求“写”访问权限
- 010 要求“读/写”访问权限

注：

(1) 此调用打开任何与指定名字相匹配的普通文件或隐式文件。不打开以冒号结尾的文件。

(2) 一个文件关闭后，由打开时施于其上的共享限

制便被撤消。

(3) 共享文件必须以共享方式加载。请参见 DOS 手册中的 SHARE 命令。

(4) 使用扩展 FCB 建立文件时，可将文件设置为只读属性，使用 CHMOD 中断 21H 功能调用，或 DOS 的 ATTRIB 命令建立句柄时，可在 CX 中指定属型。

(5) 如果该文件被“子女”过程所继承，则同时继承下所有共享限制和访问限制。

(6) 如果打开文件的句柄由两个 DUP 功能调用中任何一个进行了复制 (duplicate)，则所有共享权限和访问权限也被复制。

打开方式

打开方式在 AL 中定义，由四个面向比特的域组成，它们是：

- 继承性标志
- 共享方式域
- 保留域
- 访问域

继承性标志指出已打开文件是否被某个子女过程所继承。访问域定义了此过程可对该文件作哪些操作。共享方式域定义了其它过程可对该文件作哪些操作。

位域

	(I)	(S)	(R)	(A)
打开方式位	7	654	3	210

I: 继承标志(Inheritance flag):

如果I = 0; 文件被子女过程继承。

如果I = 1; 文件为当前过程专用。

S: 共享方式(Sharing Mode):

如果S = 000兼容方式

S = 001拒绝读/写 (排它)

S = 010拒绝写方式

S = 011拒绝读方式

S = 100不拒绝方式

任何其它组合均是非法的。

当打开一个文件时, 通知DOS 其它过程可对此文件执行哪些操作(共享方式)是很重要的, 默认(兼容方式)拒绝所有其它过程访问文件。你的过程对该文件操作时, 若允许其它过程对该文件作读操作, 此时, 应指定“拒绝写”方式, 从而允许其它过程读, 但不允许写。

同样, 指出你的过程将执行哪些操作(访问方式)也很重要。默认的访问方式(读/写)使得当其它过程不是以“不拒绝”共享方式打开该文件时, 打开请求便失败。如果你只是对文件作读操作, 则只要其它所有操作指定的都是“不拒绝”或“拒绝写”(从而增加了对该文件的访问), 你的文件打开操作是会成功的。文件共享要求两个共享过程的合作, 这种合作是通过共享方式和访问方式进行联络的。

R: 保留(Reserved将控制字的第三位设置为0)。

A: 访问(Access)

A = 000; 读访问

A = 001; 写访问

A = 010; 读/写访问
任何其它组织均是非法的。

共享方式

(1) 兼容(Compatibility)方式

文件以下列方式被打开时被认为是处于兼容方式:

- ① 任一种CREATE功能调用。
- ② 一种FCB功能调用。
- ③ 带有指定为兼容方式的一种句柄功能调用。

文件可由某一个过程以兼容方式打开任意多次, 只要该文件当前不是以其它四种共享方式之一打开的。如果该文件标记为只读, 并且当前是以“读访问”权限“拒绝写”共享方式打开的, 则该文件可以“读访问”权限兼容方式来打开。如果该文件已经由其它共享方式成功地打开了, 则试图再以兼容方式对其进行打开, 会产生中断24H通知出错。中断24H的基本错将会标志“Drive not ready”, 扩展错进一步指出“Sharing violation”。

以兼容方式打开之文件的共享方式可由DOS根据文件属性是否为只读来改变, 从而允许只读文件被共享。

上 读

文件打开方法	访问权限	共享方式
FCB	上读	拒绝写
句柄读	上读	拒绝写
句柄写	出错	
句柄读/写	出错	

非 上 读

文件打开方法	访问权限	共享方式
FCB	读/写	兼容
句柄读	读	兼容
句柄写	写	兼容
句柄读/写	读/写	兼容

(2) 拒绝读/写方式(排它)

如果一个文件是以“拒绝读/写”方式成功打开的,则对该文件的访问是排它的。当前以此种方式打开的文件在此文件被关闭之前不能再由任何过程以任何共享方式来打开(包括当前过程)。

(3) 拒绝写方式

以“拒绝写”共享方式成功打开的文件,在该文件被关闭前是不允许其它写访问来打开此文件的(A=001或010)。如果文件当前是以写访问权限打开的,则试图以“拒绝写”方式打开此文件是不成功的。

(4) 拒绝读方式

以“拒绝读”共享方式成功地打开的文件,在此文件被关闭前是不允许其它读共享访问去打开该文件的(A=000或010)。如果文件当前是以兼容方式或以读访问权限打开的,则试图以“拒绝读”共享方式打开此文件是不成功的。

(5) 不拒绝方式

以“不拒绝”方式成功打开的文件,对该文件的读/写访问能力没有限制。如果该文件当前是以兼容方式打开的,则试图以“不拒绝”方式打开该文件是不成功的。

注：如果访问的文件处于网络磁盘上，则当文件以如下共享方式之一打开时，不进行本机缓冲：

拒绝读

不拒绝

读/写访问权限和拒绝写方式打开

第二次,第三次,...打开

第一次打开

	拒绝读/写			拒绝写			拒绝读			可以读/写			
	I	IO	O	I	IO	O	I	IO	O	I	IO	O	
拒绝读/写	I	N	N	N	N	N	N	N	N	N	N	N	N
	IO	N	N	N	N	N	N	N	N	N	N	N	N
	O	N	N	N	N	N	N	N	N	N	N	N	N
拒绝写	I	N	N	N	Y	N	N	N	N	N	Y	N	N
	IO	N	N	N	N	N	N	N	N	N	Y	N	N
	O	N	N	N	N	N	N	Y	N	N	Y	N	N
拒绝读	I	N	N	N	N	N	Y	N	N	N	N	N	Y
	IO	N	N	N	N	N	N	N	N	N	N	N	Y
	O	N	N	N	N	N	N	N	N	Y	N	N	Y
可以读/写	I	N	N	N	Y	Y	Y	N	N	N	Y	Y	Y
	IO	N	N	N	N	N	N	N	N	N	Y	Y	Y
	O	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y

Y: 允许打开第二次, 第三次,

N: 不允许打开第二次, 第三次,

I: 只读访问权限

O: 只写访问权限

IO: 读/写访问权限

上写和拒绝方式打开

因此，在网络环境下，“拒绝读/写”共享方式、兼容共享方式以及“输入拒绝写”打开时是在本机缓冲的。

上面的共享矩阵给出了打开文件和以后使用各种访问、共享方式的组合重新打开同一文件的结果。

• 3EH 关闭一个文件句柄

用途：关闭指定的文件句柄。

进 入	寄 存 器 内 容
AH	3EH
BX	由open或create返回的文件句柄

返 回	寄 存 器 内 容
	如果进位标志置位，则为出错代码。 如果进位标未置位，则无

注释：进入时，BX包含有由“open”或“create”返回的文件句柄。返回时，该文件被关闭，目录被更新，并刷新用于该文件的所有内部缓冲区。

在AX中返回出错代码，发出功能调用59H可得到有关出错的更多信息。

• 3FH 从文件或设备中读

用途：从一个文件中将指定数量的字节传送到缓冲区中。

进 入	寄 存 器 内 容
AH	3FH
BX	文件句柄
DS:DX	缓冲区地址
CX	欲作读操作的字节数

返 回	寄 存 器 内 容
AX	读入的字节数 如果进位标志置位，则为出错代码

注释：进入时，BX包含有该文件的句柄。CX包含有欲被读入的字节数。DS:DX包含有缓冲区地址。返回时，AX包含有读入的字节数。

此功能调用试图从文件中传送(CX)个字节到缓冲区中，不能保证读入所有的字节。例如，从键盘读入时最多读入一行文本。如果是从标准输入设备作读操作，则输入可被重定向。参见DOS手册中“标准输入和输出的重定向”一节。如果AX中的值为0，则程序曾试着从文件尾读入。出错代码在AX中返回。发出的功能调用59H可得到更多有关出错的信息。

网络访问权限：要求“读”访问权限。

- 40H写入文件或设备

用途：将一个缓冲区中指定数量的字节传送到一个指定文件中。

进 入	寄 存 器 内 容
AH	40H
DS:DX	文件句柄
CX	被写数据的地址
BX	被写字节数

返 回	寄 存 器 内 容
AX	写入的字节数 如果进位标志置位，则为出错代码

注释：进入时，BX包含有文件句柄。CX包含有被写的字节数。DS:DX包含有被写数据的地址。

此功能调用试图将(CX)字节从缓冲区传送到一个文件中，AX返回实际写完的字节数。如果此值与请求的数不同，被看作是出错(不返回出错代码，但应用程序可比较这些值)，一般的原因是磁盘满。如果是对一个标准输出设备作写操作，输出可被重定向。参见DOS参考手册中“标准输入和输出的重定向”一节。如果打算使文件指针当前位置所指的文件清空，可在发出中断21H之前将(CX)字节数设置为0。由读、写和执行功能调用42H(移动文件读/写指针)可将文件指针移至所要求的位置。

如果文件是只读的，不执行向该文件或设备的写操作。

出错代码在AX中返回，发出功能调用59H可得到更多有关出错的信息。

网络访问权限：要求“写”访问权限。

- 41H从指定目录中删除一个文件 (UNLINK)

用途：去除与某个文件名相关联的目录登记项。

进 入	寄 存 器 内 容
AH	41H
DS:DX	ASCII字符串的地址
返 回	寄 存 器 内 容
	如果进位标志置位，则为出错代码 如果进位标志未置位，则无意义

注释：该ASCII字符串的任何部分不可以是全程文件名字符。由此调用不可删除上读文件。欲删除一个上读文件，首先应调用43H将文件上读属性改变为0，然后删除之。出错代码在AX中返回，发出功能调用59H可得到更多有关出错的信息。

网络访问权限：要求“建立”访问权限。

• 42H移动文件读写指针 (LSEEK)

用途：按照所指定向方法移动读/写指针。

进 入	寄 存 器 内 容
AH	42H
CX:DX	移动距离(偏移量)以字节计
AL	移动方法(0,1,2)
BX	文件句柄
返 回	寄 存 器 内 容
AX	如果进位标志置位，则为出错代码
DX:AX	如果进位标志未置位，则为新的指针置位

注释：进入时，AL 包含有一个方法值，BX 包含有文件句柄，CX:DX 包含以字节计的要求偏移值（CX 包含高位部分）。返回时，DX:AX 包含有指针的新位置（DX 包含高位部分）。出错代码在 AX 中返回，发出功能调用 59H 可得到有关出错的更多信息。

此功能调用按照如下方法移动读/写指针：

AL	说 明
0	指针从文件开始移动 CX : DX 个字节（偏移值）
1	指针移至当前位置加偏移值
2	指针移至文件末尾加偏移值，此法可用于确定文件大小

注：如果对驻留在网络磁盘上，并且是以“拒绝读”或“拒绝写”共享方式打开的文件执行 LSEEK 操作，则在文件实际存在的计算机来调整读/写指针信息。如果该文件以任何其它共享方式来打开的，则在远程计算机上保存读/写指针信息。

• 43H 改变文件模式 (CHMOD)

用途：改变指定文件的模式(属性)。

进 入	寄 存 器 内 容
AH	43H
DS : DX	指向一个 ASCII 路径名
CX	属性
AL	功能代码

注释：进入时，AL中包含有一个功能代码，DS:DX中包含一个带有驱动器，路径和文件名的 ASCII 字符串。

如果AL包含01H，则文件属性被设置成CX寄存器中的属性。“DOS磁盘目录”一节给出了关于属性字节的说明。如果AL为00H则在CX中返回文件的当前属性。

出错代码在AX中返回，发出功能调用59 H可得到更多有关出错的信息。

注：属性08H和10H不能够使用CHMOD来改变。如果企图用它们来改变文件的属性则返回出错代码。

网络访问权限：欲改变案卷(archive)位(AL=20H)，无需访问权限。欲改变其它位，要求有“建立”访问权限。

• 44H 设备的I/O控制(IOCTL)

用途：设备或得到与打开设备句柄有关的设备信息，或向该设备发送/接收控制字符串。

进 入	寄 存 器 内 容
AH	44H
DS:DX	数据或缓冲区
CX	作读或写操作的字节数
BX	文件句柄
BL	驱动器号(0=默认, 1=A等)
AL	功能值

返 回	寄 存 器 内 容
AX	如果进位标志未置位，则为传送的字节数 如果进位标志置位或如果AX=0FFH，则为出错代码

注释：进入时，AL包含有功能值。BX包含该文件的句柄。返回时，AX包含功能2, 3, 4, 5的传送的字节数；功能6, 7的状态值(00H = 没准备好, FFH = 准备好)；或一个出错代码。

出错代码在AX中返回，发出功能调用59H可得到有关出错的更多信息。

AL中允许出现下列功能值：

(1) AL = 00H获得设备信息(在DX中返回)。

(2) AL = 01H设置设备信息(由DX决定)。作此调用时通常DH必须为0。

(3) AL = 02H从设备控制通道向DS = DX读入CX个字节数。

(4) AL = 03H从DS:DX向设备控制通道写CX个字节数。

(5) AL = 04H同(2)，但使用BL中的驱动器号(0 = 默认, 1 = A等)。

(6) AL = 05H同(3)，但使用BL中的驱动器号(0 = 默认, 1 = A等)。

(7) AL = 06H获得输入状态。

(8) AL = 07H获得输出状态。

(9) AL = 08H该块设备是否可改变？

(10) AL = 09H是一个逻辑设备还是一个远程设备？

(11) AL = 0AH句柄为本机的还是远程的？

(12) AL = 0BH改变共享重试次数。

IOCTL可被用于获得有关设备通道的信息。可对普通文件执行这些调用，但此时仅功能值0, 6, 7有意义。其它调用将返回“非法功能”错。

在网络设备上不支持功能值00H到08H。

功能值0BH要求装载文件共享命令(SHARE)。

调用AL = 00H和AL = 01H

DX中的各位按如下定义：

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	C							I	E	B	R	I	I	I	I
E	T							S	O	I	E	S	S	S	S
S	R	网络保留						D	F	N	S	C	N	C	C
	L							E		A		L	U	O	I
								V		B		K	L	T	N

ISDEV = 1 该通道是一个设备。

= 0 该通道是一个磁盘文件(此情况下位8—15=0)。

如果 ISDEV = 1

EOF = 0 输入时文件结束。

BIN = 1 以二进制方式进行操作(不检查Ctrl-Z)。

= 0 ASCII方式进行操作(检查 Ctrl-Z 作为文件结束)。

ISCLK = 1 此设备是时钟设备。

ISNUL = 1 此设备是空设备。

ISCOT = 1 此设备是控制台输出。

ISCIN = 1 此设备为控制台输入。

CTRL = 0 此设备不能通过调用AL = 02H, AL = 03H, AL = 04H, AL = 05H处理控制字符串。

CTRL = 1 此设备可通过调用AL = 02H和AL = 03H来处理控制字符串。注意此位不可由功能调用44H来设置。

如果ISDEV = 0

EOF = 0 已对通道作了写操作。位0—5是该通道的块设备号(0 = A, 1 = B, ...)。位15, 8—13, 4保留且不改变内容。

注: 对于调用AL = 01H, DH必须为0。

调用AL = 02H, AL = 03H, AL = 04H, AL = 05H

这四个调用允许对设备设置属性控制字符串或从设备接到属性控制字符串。调用语法与“Read”和“Write”调用相同(除了调用4,5是在BL中接到一个驱动器号而不是接到BX中的句柄)。如果CTRL位为0,则返回“非法功能”错误。如果04H和05H调用时的驱动器非法则返回“拒绝访问”代码。

调用AL = 06H和AL = 07H

这两个调用可用于检查文件句柄是否已准备好输入或输出。如果用于文件,则在到达文件尾之前AL中一直返回FFH,然后总是返回00H(除非当前文件位置通过调用42H进行了改变)。如果用于设备,则“准备好”时AL返回FFH,“没准备好”时AL返回00H。

调用AL = 08H(DOS3.00和3.10)

此调用可用于确定一个设备是否能支持“可移出”(removable)介质。如果AX中返回的值是0,则该设备是可移出的。如果值是1,则该设备是固定(fixed)的。设备号应安置在BL中;如果BL中的值为非法,则返回出错代码0FH。对于网络设备,返回“非法功能”出错信息。

调用AL = 09H(DOS3.10)

此调用可用于确定一个逻辑设备与一个网络目录是否有关联。进入时, BL包含有你打算检查的块设备的驱动器号

(0 = 默认, 1 = A, 2 = B等等)。本机设备上DX中返回的值为来自于设备标题的属性字。对于远程设备, 位12被置位(1000H)。DX中的其它位被保留。如果磁盘重定向暂停, 该功能为本机设备返回属性字, 并且位12不置位。

注: 不要写测试位12的程序, 所写的应用程序应当与块设备的地点(本机的或是远程的)无关。

调用AL = 0AH (3.10)

此调用可用于确定句柄是用于本机设备还是用于网络上的远程设备。DX中返回的值是来自于该设备标题的属性字。对于远程设备, 位15被置位(8000H), 句柄应放在BX中。

注: 不要写程序去测试位15, 应用程序的编写应当与句柄的地点(本机或远程)无关。

调用AL = 0BH (DOS3.00和3.10)

所有共享冲突和加锁冲突在作为DOS错误或致命错误返回之前, 均自动地重试若干次。你可选择重试次数和重试之间的延迟时间。输入时, CX 包含执行一个延迟循环的时间数, DX 包含有重试次数。延迟循环由下列语句构成:

```
XOR CX, CX
```

```
LOOP $      , 持续64K次
```

如果从未发出此调用, DOS使用“延迟 = 1”和“重试次数 = 3”作为CX和DX的默认值。如果估计自己的应用程序可能在一个短时间中发生共享冲突或加锁冲突, 可增加CX和DX的值以减少实际返回到你的应用程序中的出错次数。

• 45H复制文件句柄(DUP)

用途: 为一个要求在同样位置处的同样文件返回一个打

开文件的新文件句柄。

进入	寄存器内容
AH	45H
BX	文件句柄

返回	寄存器内容
AX	如果进位标志未置位, 则为新文件句柄 如果进位标志置位, 则为出错代码

注释: 进入时, BX 包含该文件的句柄。返回时 AX 包含返回的文件句柄。

出错代码在 AX 中返回, 发出功能调用 59H 可得到有关出错的更多信息。

注: 如果使用读、写或 LSEEK 功能调用移动了任一个句柄的读/写指针, 则另一个句柄的指针也被改变。

• 46H 强迫句柄复制 (FORCDVP)

用途: 强迫 CX 中的句柄相联系, 与在 BX 中的句柄相同位置处的相同文件。

进入	寄存器内容
AH	46H
BX	已存在的文件句柄
CX	第二个文件句柄

返回	寄存器内容
AX	如果进位标志置位, 则为出错代码 如果进位标志未置位, 则无意义

注释：进入时，BX包含有读文件句柄。CX包含有第二个文件句柄。返回时，CX句柄联系于BX文件句柄相同位置处的相同文件。如果CX文件句柄已是一个打开文件，则它首先关闭。如果你移动了任一个句柄的读/写指针，另一个句柄的指针也改变。出错代码在AX中返回，发出功能调用59H可得到有关出错的更多信息。

• 47H得到当前目录

用途：在DS:SI所指向的区域中，安排一个指定驱动器之当前目录的完整路径名（从根目录开始）。

进入	寄存器内容
AH	47H
DS:SI	指向一个64字节的用户内存区
DL	驱动器号（0=默认，1=A等）

返回	寄存器内容
DS:SI	如果进位标志未置位，则填充以自根目录开始的完整路径名
AX	如果进位标志置位，则为出错代码

注释：驱动器字母并不是返回字符串的一个部分。该字符串并不以反斜杠\开始，但从包含00H的一个字节结束。

出错代码在AX中返回，发出功能调用59H可得到有关出错的更多信息。

• 48H分配内存

用途：分配请求数量的内存段落(paragraph)。

进 入	寄 存 器 内 容
AH	48H
BX	请求的内存段落数

返 回	寄 存 器 内 容
AX:0	指向已分配的内存块
AX	如果进位标志置位, 则为出错代码
BX	如果分配失败, 则为段落计数可用内存最大块的大小

注释: 进入时, BX中包含有请求的段落数。返回时, AX:0指向已分配的内存块。如果分配失败, BX返回以段落计数的最大可用内存块的大小。

出错代码在AX中返回, 发出功能调用59H可得到有关出错的更多信息。

• 49H释放已分配的内存

用途: 释放指定的已分配内存。

进 入	寄 存 器 内 容
AH	49H
ES	被返回的块所在段(segment)

返 回	寄 存 器 内 容
AX	如果进位标志置位, 则为出错代码 如果进位标志未置位, 则无意义

注释: 进入时, ES包含欲返回到系统内存的块所在段的地址。当调用返回时, 该内存块返回给系统内存(pool)。

出错代码在AX中返回，发出功能调用59H可得到有关出错的更多信息。

• 4AH 修改已分配了的内存块 (SETBLOCK)

用途：修改已分配的内存块以容纳新指定块的大小。

进 入	寄 存 器 内 容
AH	4AH
ES	该块所在段 (segment)
BX	包括以段落计数新请求的块之大小

返 回	寄 存 器 内 容
AX	如果进位标志置位，则为出错代码 如果进位标志未置位，则无意义
BX	如果进位标志置位，则在“增大 (grow) 请求”时，此调用失败，BX 的值是最大可请求的值

注释：DOS试图“增大 (grow)”或“收缩 (shrink)”指定的块。

出错代码在AX中返回，发出功能调用59H可得到有关出错的更多信息。

• 4BH 装入或执行程序 (EXEC)

用途：允许一个程序将另一个程序装入内存，并可任选开始执行第二个程序。

进入	寄存器内容
AH	4BH
DS:DX	指向欲装入文件的带有驱动器路径, 文件名的 ASCIIZ 字符串
ES:BX	为加载指向一个参数块
AL	功能值(见说明)
返回	寄存器内容
AX	如果进位标志置位, 则为出错代码 如果进位标志未置位, 则无意义

注释: 出错代码在AX中返回, 发出功能调用59H可得到有关出错的更多信息。

AL中允许使用下列功能值:

功能值	说明
00H	装入并执行该程序, 为该程序建立一个程序段前缀; 结束地址和 Ctrl-Break 地址被设置到 EXEC 系统调用之后的一条指令。 注: 控制返回时, 包括堆栈在内的所有寄存器被改变。此时必须在继续处理之前恢复SS, SP和其它要用到的寄存器。
03H	加载, 但不建立程序段前缀, 也不开始执行, 这对于装入覆盖程序是有用的。

对这些功能值, ES:BX所指向的块具有下列格式:

AL = 00H装入/执行程序

WORD 欲被传送之环境字符串段地址

DWORD指向将放置在PSP + 80H处的命令行

DWORD指向将在PSP + 5CH处被传送的默认FCB

DWORD指向在PSP + 6CH处传送的默认FCB

注：DWORD（双字）指针的格式是段和偏移量。

AL = 03H装入覆盖模块

WORD文件将被装入的段地址

WORD向该映像(image)施用的重定位因子(factor)

在EXEC后，在新建立的过程中，将复制调用过程的所有打开文件（除非该文件打开时文件的“继承性”位置位），这意味着父过程对于标准输入、输出、辅助和打印机设备仍具有控制权。例如，父过程向一个文件写入了一组记录，将该文件作为标准输出，然后执行一检索程序从标准输入中取出输入内容，并写到标准输出中。“继承”（或父过程的一个副本）也是一个“环境”。它是一块文本字符串（总数小于32k字节），它传送了各种组合(Configuration)参数。该环循的格式如下（总是处于一个段落边界处）：

Byte ASCII Z字符串1

Byte ASCII Z字符串2

...

Byte ASCII Z字符串n

Byte 0

典型的环境字符串具有如下形式:

参数 = 值

在环境的 0 字节之后是指出后面的其它字符串个数的一
个字。再后面是传送给子过程之 DS:DX 处 文件名的一个副
本。例如, 可传送字符串 VERIFY = ON。环境地址为 0 值将
使得新建立的过程继承父过程的环境而不改变。该环境的段
地址放置开始调入程序的程序段前缀之偏移量 2CH 处。

注:

(1) 当你的程序接收控制时, 所有可用内存已分配给
它。在 EXEC 能够装入你打算调用的程序之前, 你必须释放某
些内存 (见调用 4AH)。通常应压缩到你需要的最小内存量
并将其余的释放。

(2) EXEC 调用使用 COMMAND.COM 的装载程序来执
行装入。

• 4CH 结束一个过程 (EXIT)

用途: 结束当前过程并将控制传送给调用过程。

进入	寄存器内容
AH	4CH
AL	返回代码

返回	寄存器内容
	无

注释: 可设置返回代码, 该返回代码可由批处理子命令
IF 和 FRRORLEVEL 以及“等待 (WAIT)”功能调用 4DH 来查

询。由该过程打开的所有文件均被关闭。

• 4DH获得子过程的返回代码(WAIT)

用途：获得通过功能调用4CH或功能调用31H处理的另一过程所给出的返回码。上返回一次EXIT代码。

进入	寄存器内容
AH	4DH

返回	寄存器内容
AX	返回码

注释：退出码的低字节包含着由退出例行程序所发送的信息。退出码的高字节可以包含：

00H 正常结束

01H Ctrl-Break 结束

02H 致命设备错引起的结束

03H 调用31H产生的结束

• 4EH找寻第一个匹配文件(FINDFIRST)

用途：找寻与指定文件标识符相匹配的第一个文件名。

进入	寄存器内容
AH	4EH
DS:DX	指向一个包含有欲找寻的文件的驱动器, 修理和文件名的ASCII字符串
CX	用于搜寻此文件的属性

返 回	寄 存 器 内 容
AX	如果进位标志置位，则为出错代码

注释：DS:DX 中的文件名可包含全程文件名字符。该 ASCII 字符串不能包含有网络路径。搜寻时如何使用属性位请参看功能调用 11H 的说明。

出错代码在 AX 中返回，发出功能调用 59H 可得到有关出错的更多信息。

如果找到了一个与指定驱动器、路径、文件名和属性相匹配的文件，则当前 DTA 填充以如下内容：

21 个字节——保留给 DOS 用于以后的查找调用

1 字节——文件属性

2 字节——文件时间

2 字节——文件日期

2 字节——文件大小低字 (Word)

2 字节——文件大小高字

13 字节——找到文件的名字和扩展名，后面跟着一个 0 字节。名字和扩展名中的空格被去除，如果存在有扩展名，则以一个句号为前导。故所返回的名字看起来与作为命令参数打开时一样，例如 TREE，COM 后跟一个 0 字节。

• 4FH 寻找下一个匹配文件 (FINDNEXT)

用途：寻找与在上一次 4EH 和 4FH 调用中已指定的名字相匹配的下一个目录登记项。

进 入	寄 存 器 内 容
AH	4FH
DTA	包含有取自于上一次4EH, 4FH功能调用的信息。

返 回	寄 存 器 内 容
AX	如果进位标志置位, 则为出错代码

注释: 如果找到了匹配文件, DTA的设置同调用4EH所述。如果没有找到匹配文件, 则返回出错代码。

出错代码在AX中返回, 发出功能调用59H可得到有关出错的更多信息。

• 54H 获得验证设置

用途: 返回验证标志的值。

进 入	寄 存 器 内 容
AH	54H

返 回	寄 存 器 内 容
AL	当前验证标志值 00H, 如果验证为off 01H, 如果验证为ON

注释: 返回时, 如果验证为OFF, AL返回00H; 如果验证为ON, 则AL返回01H。注意验证开关可通过调用2EH设置。

• 56H 文件换名

用途：给指定文件换名。

进 入	寄 存 器 内 容
AH	56H
DS : DX	指向包含有欲被换名文件的驱动器，路径和文件名的ASCII字符串
ES : DI	指向包含有新的路径和文件名的ASCII字符串

返 回	寄 存 器 内 容
AX	如果进位标志置位，则为出错代码 如果进位标志未置位，则无内容

注释：如果ASCII字符串中包含有驱动器，则必须与所指定的驱动器相同，或隐含在第一个字符串中。目录路径则不必相同，即允许一个文件移至另一目录中并在此过程中换名。文件名中不许出现全程文件名字符。

出错代码在AX中返回，发出功能调用59H可得到更多有关出错的信息。

网络访问权限：要求“建立”访问权限。

• 57H 获得/设置文件的日期和时间。

用途：获得和设置文件的日期和时间。

进 入	寄 存 器 内 容
AH	57H
AL	00H 获得日期和时间 01H 设置日期和时间
BX	文件句柄
CX	如果 AL = 01H, 为被设置成的时间
DX	如果 AL = 01H, 为被设置成的日期

返 回	寄 存 器 内 容
AX	如果进位标志置位, 则为出错代码
DX	如果获得日期, 则为句柄内部表格中的日期
CX	如果获得时间, 则为句柄内部表格中的时间

注释: 日期和时间的格式与本书第五章中所描述的目录登记项一样, 不同的是, 在寄存器中传送时, 字节被保留 (即DH包含有日期的低阶部分等)。

出错代码在AX中返回, 发出功能调用 59H 可得到更多有关出错的信息。

• 59H (DOS3.00和3.10) 获得扩展错

用途: 返回如错误类别, 出错地点以及建议动作等附加的出错信息。

进 入	寄 存 器 内 容
AH	59H
BX	0000H (版本号, 0用于3.00和3.10)

返 回	寄 存 器 内 容
AX	扩展错
BH	错误类别
BL	建议动作
CH	出错地点

注释：此功能调用返回的除了出错代码外，还返回出错类别、地点和建议动作。可从以下情况下使用此功能调用。

- (1) 中断24H出错处理程序
- (2) 中断21H功能调用返回时进位标志出错
- (3) FCB功能调用返回FFH

返回时，DX，SI，DI，ES，CL和DS 寄存器内容被破坏。

进位位出错的返回

对于由设置进位标志来指示出错的功能调用，执行功能调用59H的正确方法是：

- (1) 给寄存器赋值。
- (2) 发出中断21H。
- (3) 如果进位位未置位，则继续操作。
- (4) 不理睬该出错代码，发出功能调用 59H 以获得更多的信息。
- (5) 使用 BL 中的值以确定要作的建议动作。

在AL中的出错状态

对于由设置AL为FFH来指示出错的功能调用，执行功能调用59H的正确方法是：

- (1) 给寄存器赋值。

(2) 发出中断21H。

(3) 如果AL中没有报告出错，继续操作。

(4) 不理睬出错代码，发出功能调用59H以获得更多的有关信息。

(5) 使用BL中的值以确定应采取的建议动作。

• 5AH (DOS3.00和3.10) 建立唯一性(unique)文件

用途：产生一个没用过的文件名，并在指定目录中建立此文件

进 入	寄 存 器 内 容
AH	5AH
DS:DX	指向以反斜杠\结尾的ASCIIZ路径
CX	属性

返 回	寄 存 器 内 容
AX	如果进位标志置位，则为出错代码
DS:DX	带有追加新文件的文件名的ASCIIZ路径

注释：进入时，AX 包含 5AH。如果没有出错，则文件以兼容方式及读/写访问权限打开，AX 包含有文件句柄，文件名追加到DS:DX指定的路径。

此功能调用产生一个没用过的名字，并试图在指定目录中建立一个新文件。如果该目录中已有此文件存在，则产生另一个没用过的名字并重复此过程。需用到临时文件的程序，可用此功能调用去生成唯一的文件名。

出错代码在AX中返回，发出功能调用59H可得到更多有

关出错的信息。

注：使用此功能调用建立的文件在程序结束时并不自动删除。

网络访问权限：要求“建立”访问权限。

• 5BH (DOS3.00和3.10) 建立新文件

用途：建立一个新文件。

进 入	寄 存 器 内 容
AH	5BH
DS:DX	指向一个ASCII路径名
CX	文件属性

返 回	寄 存 器 内 容
AX	如果进位标志置位，则为出错代码 如果进位标志未置位，则为句柄

注释：此功能调用与功能调用3CH（建立）相同，差别在于如果文件名已存在，则此功能调用将失败。文件以兼容方式建立，可作读、写操作。

出错代码在AX中返回，发出功能调用59H，可得到更多有关出错的信息。

网络访问权限：要求“建立”访问权限。

• 5CH (DOS3.00和3.10) Lock/Unlock 文件访问

用途：对一个已打开文件中部分字节加锁(lock)或解锁(unlock)。

进 入	寄 存 器 内 容
AH	5CH
AL	00H, 加锁 01H, 解锁
BX	文件句柄
CX	偏移量高字
DX	偏移量低字
SI	长度高字
DI	长度低字

返 回	寄 存 器 内 容
AX	如果进位标志置位, 则为出错代码

注释: 仅当文件是使用“拒绝读”或“不拒绝”共享方式打开, 或当文件为读/写访问权限、只写访问权限及“拒绝写”共享方式打开时, 才能使用lock/unlock功能调用。这些方式在访问网络磁盘上的文件时不在本机作数据缓冲。

AL = 00H 加锁(lock)

提供了排除其他进程对该文件作读/写访问操作的简单机构。如果有另一个进程试图对同一区域进行读或写, 其系统调用将根据 IOCTL 设置的系统重试若干次。

如果经重试后未能成功, 则产生出错信息。由使用功能调用 44H 中 AL = 0BH (IOCTL 中“改变共享重试次数”) 可改变重试的次数和两次重试之间的时间长度。发出功能调用 59H 可得到出错代码、出错类别、出错地点以及推荐的动作。加锁区域可处于逻辑文件的任何地方。最好被加锁区域

的锁定时间能短一些。复制句柄便重复对加锁区域的访问，对加锁区域的访问不能通过 EXEC 系统调用来重复。文件处于打开状态，并且对该文件已加上锁以便退出可能产生无法预料的结果。由使用 INT23H 或 INT24H 可能流产的程序，在退出之前应将锁释放。正确地使用加锁并不是要“拒绝读”或“拒绝写”，而是要对所要求区域上锁及对于出错代码的检查。

AL = 01H 解锁(Unlock)

解锁是将在加锁的系统调用中所加的锁释放掉。所指定的区域必须与上一次加锁时所指定的区域完全相同。在加锁状态下关闭一个文件会导致无法预料的结果。文件打开着并已对该文件上了锁使退出也会导致无法预料的结果。使用 INT23H 或 INT24H 可能流产的程序，在退出前应先释放锁。正确地使用上锁并不是要“拒绝读”或“拒绝写”，而是要对所需要的区域上锁及对于出错代码的检查。

出错代码在 AX 中返回。发出功能调用 59H 可得到更多有关出错的信息。

• 5E00H(DOS3.10) 得到机器名

用途：返回本机的字符标识符。

进 入	寄 存 器 内 容
AX	5E00H
DS:DX	指向返回计算机名的 ASCII 串的内存缓冲区

返 回	寄 存 器 内 容
DS:DX	填写计算机名 ASCIIZ
CH	名字/号码指示标志 0 = 名字未定义 非0 = 定义了名字/号码
CL	该名字的NETBIOS名字是号码
AX	如果进位标志置位, 则为出错代码

注释：“得到机器名”将当前计算机名返回给调用者。所谓计算机名由15个字符组成，不足时填充以空格，最后是00H字节。如果该计算机名从未设置过，寄存器CH中返回00H，且CL寄存器中的值为非法。此功能调用要正确执行则须装入IBM PC Network网络软件。

• 5E02H(DOS3.10) 建立打印机设置

用途：为打印机文件指定一个初始字符串。

进 入	寄 存 器 内 容
AX	5E02H
BX	重定向表索引
CX	设置字符串长度（最大长度为64字节）
DS:SI	指向打印机设置缓冲区

返 回	寄 存 器 内 容
AX	如果进位标志置位, 则为出错代码

注释：所指定的字符串放在与特定网络打印机有关的所有文件前面。打印机设置允许一台打印机的多个用户指定打

印机运行的各自方式。BX所设置的索引与功能调用5F02H（获得重定向表登记项）所用的索引一样。如果打印重定向被暂停或没有装入IBMPC Network网络软件，则返回出错代码。

出错代码在AX中返回，发出功能调用59H可得到更多有关出错的信息。

• 5E03H(DOS3.10) 获得打印机设置

用途：返回打印机文件所用的打印机设置字符串。

进 入	寄 存 器 内 容
AX	5E03H
BX	重定向表索引
ES:DI	指向打印机设置缓冲区（最大长度为64字节）

返 回	寄 存 器 内 容
AX	如果进位标志置位，则为出错代码
BX	返回数据的长度
ES:DI	填写打印机设置字符串

注释：此功能调用返回使用功能调用5E02E（建立打印机设置）所指定的打印机设置字符串。该设置字符串供所有与特定打印机有关的文件用。BX中的值设置成与功能调用5F02H（获得重定向表登记项）相同的索引。如果未装入IBM PC Network网络则返回出错代码1（非法功能号）。

出错代码在AX中返回，发出功能调用59H可得到更多有关出错的信息。

• 5F02H(DOS3.10) 获得重定向表登记项

用途：返回非本机的网络重定向表登记项

进 入	寄 存 器 内 容
AX	5F02H
BX	重定向索引(0为基底)
DS:SI	指向本机设备名的128字节缓冲区地址
ES:DI	指向网络名的128字节缓冲区地址

返 回	寄 存 器 内 容
AX	如果进位标志置位, 则为出错代码
BH	设备状态标志 位0=0如果设备合法 =1如果设备非法 位1-7保留
BL	设备类型
CX	存储的参数值
DX	被破坏
BP	被破坏
DS:SI	ASCII本机设备名
ES:DI	网络名ASCII

注释：“获得重定向表登记项”功能调用返回，通过功能调用5F03H（设备重定向）所建立的网络重定向表。每次调用返回一个重定向，则BX的值每次增加1。在调用之前可改变表的内容。根据出错代码18（无更多文件）来检测是否为表末。如果没有装入IBM PC Network网络程序，则返回出错

代码 1 (非法功能号)。

不管是磁盘还是打印机的重定向被暂停, 本功能不受影响。

出错代码在 AX 中返回。发出功能调用 59H 可得到更多关于出错的信息。

• 5F03H (DOS3.10) 设备重定向

用途: 引起重定向器(Redirector)与服务器(Server)的连接实现。

进 入	寄存器内容
AX	5F03H
BL	设备类型 03 打印机设备 04 文件设备
CX	为调用者所保存的值
DS:SI	源 ASCIIZ 设备名
ES:DI	带口令的目的 ASCIIZ 网络路径

返 回	寄存器内容
AX	如果进位标志置位, 则为出错代码

注释: 此调用是为网络定义当前目录及定义网络打印机重定向的接口。

★如果 BL = 03, 源是一台打印机, 目标是网络路径, CX 寄存器中的字是 DOS 为编程者所保存的。为了与 IBM PC Network 网络程序兼容, CX 应被设置为 0。非 0 值是为 IBM PC Network 网络程序所保留的。该字可通过功能调用 5F02

H (获得重定向表) 来检出。所有向指定打印机的输出被缓冲后发送到远程打印机作该设备的假脱机打印。打印机在 INT17H 级别上被重定向。源字符串必须是 PRM, LPT1, LPT2 或 LPT3, 均须以 00H 结尾。目标字符串必须指向一个如下形式的网络名字符串:

[`\\计算机名\{短名\打印设备}`]

目标字符串必须以 00H 结尾。

在网络字符串后应紧跟访问远程设备所需要的 ASCII 口令 (0 到 8 个字符)。口令亦必须以 00H 结尾。空口令 (零长度) 被认为是无口令

★如果 BL = A, 源指出一个驱动器字母和冒号, 以 00H 结尾, 目标指出一个网络路径, 以 00H 结尾, CX 寄存器中的字是 DOS 为编程者所保存的。为了与 IBM PC Network 程序兼容, CX 应设置为 00H。非 00H 以外的值是 IBM PC Network 程序所保留的。此值可由功能调用 5F02H (获得重定向表) 进行检索。如果源是一个驱动器字母, 则建立起驱动器字母和网络路径之间联系。对于该驱动器字母以后的引用, 转换为对该网络路径的引用。如果源是一个空字符串, 则系统试图对带有指定口令的目标进行访问, 而不是重定向什么设备。

访问远程路径所用的 ASCII 口令应直接跟在网络字符串后面。以 00H 结尾的空口令 (0 长度) 被看作是无口令。

出错代码在 AX 中返回, 发出功能调用 59H 可得到更多有关出错的信息。

注:

(1) 通过此功能调用, 重定向的设备不能用 NET USE 命令显示出来。

(2) 如果在磁盘重定向暂停时试图重定向一个文件设备或如果在打印重定向暂停时试图重定向一台打印机则返回出错。

• 5F04H (DOS 3.10)撤消重定向

用途：撤消前面的重定向。

进 入	寄存器内容
AX	5F04H
DS:SI	ASCII设备名或路径
返 回	寄存器内容
AX	如果进位标志置位，则为出错代码

注释：由“设备重定向”功能调用(5F03H)所产生的重定向，通过“撤消重定向”调用所撤消。如果缓冲区指向一个驱动器字母，而该驱动器与一个网络名相关联，则该联系被终止，且该驱动器恢复为原来的实际意义。如果缓冲区指向PRN, LPT1, LPT2或LPT3, 该设备与一个网络设备相关联，则此关系被终止，且该设备恢复为基原有的实际意义。如果缓冲器指向一个以00H结尾的网络路径和一个以00H结尾的口令，则本机机器与网络目录之间的关联被终止。如果在磁盘重定向暂停时试图撤消一个已重定向了的文件设备，或如果在打印重定向暂停时试图撤消一个已重定向了的打印机，则返回出错代码。

如果没有装入 IBM PC Network 程序则返回出错代码1(非法功能号)。

出错代码在 AX 中返回，发出功能调用59H 可得到有关出错的更多信息。

- 62H (DOS 3.00和3.10)获得程序段前缀地址(PSP)
用途：返回程序段前缀地址。

进 入	寄存器内容
A H	62H

返 回	寄存器内容
BX	当前执行过程的段地址

注释：当前执行过程的内部 PSP 地址在 BX 中返回。

第七章 DOS控制块和工作区

第一节 引 言

本章包含有：

- DOS 内存图诸地址及其用途的说明。
- 程序段前缀的详细描述和图示。
- 文件控制块(标准的和扩展的)的详细描述和图示。

第二节 DOS 内存图

地 址	用 途
0000 : 0000	中断向量表
0040 : 0000	ROM 通信区
0050 : 000	DOS 通信区
XXXX : 0000	IBMBIO.COM——DOS 对于 ROM I/O例行程序的接口
XXXX : 0000	IBMDOS.COM——DOS中断句柄, 服务子程序(INT21H功能);DOS缓冲区, 控制区及已装配的设备驱动程序

续

XXXX:0000	COMMAND.COM 的驻留部分——中断22H(程序结束), 23H(Ctrl-Break), 24H(致命错误)的中断句柄, 以及重装入暂存部分的代码
XXXX:0000	外部命令或实用程序——(.COM或.EXE文件)
XXXX:0000	.COM文件的用户堆栈(256字节)
XXXX:0000	COMMAND.COM的暂存部分

注:

1. 内存图地址以段地址、段内偏移量的格式给出。例如, 0070:0000的绝对地址为00700H。

2. DOS 通信区使用情况如下:

0050:0000 存储屏幕打印状态标志。

0 未进行屏幕打印或屏幕打印操作成功

1 屏幕打印正在进行

255 屏幕打印操作期间出错

0050:0001 由BASIC 使用

0050:0004 单驱动器方式状态字节

0 最后一次用的是驱动器 A 中的磁盘

1 最后一次用的是驱动器 B 中的磁盘

0050:0010——0021 BASIC使用

⁰0050:0022——002F DOS用于作磁盘初始化

0050:0030——0033 MODE命令用

0050:0000 处开始的 256 字节中的其它地址为 DOS 保留。

3. 用户内存从可提供内存最低端开始分配。

第三节 DOS程序段

打入一条外部命令,或通过 EXEC 功能调用调用一个程序, DOS 确定出最低的可用地址作为程序可用内存的开始地址。这一区域称之为程序段。

程序段偏移量0处, DOS 建立起程序段前缀控制块。EXEC 在偏移量100H处装入该程序并给予它控制权。

该程序以 EXEC 返回时,须发出 INT20H,或发出一个 INT21H(寄存器 AH = 00H 或4CH),或调用程序段前缀中地址50H(寄存器 AH = 00H 或4CH),以便跳转到程序段前缀偏移量0处。

注:使用上述任何方法(除了调用4CH外)终止程序时,所有程序都必须保证 CS 寄存器中包含有程序段前缀的段地址。

上述各种方法的结果均是返回到发出 EXEC 的程序。在返回过程中,中断向量 22H, 23H, 24H (程序终止, Ctrl-Break, 致命错误退出地址)均从终止程序的程序段前缀所保留的值恢复,然后将控制交给程序终止地址。

一个程序接到控制后,下列条件便起作用:

对于所有程序

- 所传送的环境的地段地址包含在程序段前缀偏移量 2CH处。

所谓环境(environment)为一串 ASCII 字符串(总数小于32k字节)格式如下:

名字 = 参数 00 00 01 00

每个字符串均以 0 字节结束，整个一组字符串以另一个 0 字节结束。在终止一组环境字符串的 0 字节之后，是一组初始化变量，它们传送给包含有 ASCIIZ 字符串后跟有字计数的程序。该 ASCIIZ 字符串包含有该可执行程序的驱动器、路径、文件名[.扩展名]。程序可使用此区域确定该程序是从何处装入的。由命令处理程序所建立的环境(以及传送给它所调用的所有程序)至少包含有一条 COMSPEC = 字符串 (COMSPEC 的参数是 DOS 用于查找磁盘中 COMMAND.COM 的路径)。该环境中还将有最后一条发出的 PATH 和 PROMPT 命令，以及通过 SET 命令输入的其它环境字符串。

所传送的环境实际上是调用过程环境的一个副本。如果应用程序使用了“程序结束并驻留”的方法，则将发现传送的环境副本是静态的。即使后面发出了 SET, PATH 或 PROMPT 命令也不发生改变。

- 程序段的前缀中偏移量 50H 包含有调用 DOS 功能调用程序所要求的代码。因此将所要求的功能号放在 AH 中，程序可发出长调用至 PSP + 50H 以调用 DOS 功能，而不用发出中断号 21H。

- 磁盘传送地址 (DTA) 设置在 80H 处，此为 DTA 在 PSP 中的默认位置。

- 命令发出后输入的头二个参数用于格式化 5CH 和 6CH 处的文件控制块。注意，即使某个参数包含有路径名，相应的 FCB (文件控制块)，也只包含有一个合法的驱动器号，文件名部分无效。

- 在81H处的无格式参数区包含有命令名后输入的所有字符(包括前导的和插入的分界符),80H处设置为字符个数。如果在命令行上打入了<, >, 或!参数, 这些参数将不出现在参数区中(与这三个参数有关的文件名也不出现), 因为标准输入、输出的重定向对于应用程序是透明的。

- 对于.COM文件, 偏移量6(一个字)包含有本段中可用的字节数。

- 寄存器AX反映出由前二个参数输入的驱动器标识符的合法性:

AL = FFH, 如果第一个参数包含有非法的驱动器标识符(否则AL = 00H)。

AH = FFH, 如果第二个参数包含有非法的驱动器标识符(否则AH = 00H)。

1. 对于.EXE程序:

- DS 和 ES 寄存器指向程序段前缀。

- CS, IP, SS和SP寄存器为链接程序传送的值。

2. 对于.COM 程序

- 所有四个段寄存器包含初始分配块的段地址, 以PSP控制块开始;

- 所有用户内存分配给该程序, 如果该程序打算通过EXEC 功能调用去调用另一个程序, 必须首先通过 Setblock (4AH)功能调用释放一些内存, 为要调用的程序提供部分空间。

- IP 指针设置为00H。

- SP 寄存器设置到程序段结尾。偏移量6处的段大小折算成段落大小。

• 堆栈顶部放一个0字。

程序段前缀 PSP格式如下节。

第四节 程序段前缀PSP

0		1		2		3		4		5		6		7	
INT20H		内存顶部		保留		对DOS功能调用入口的									
8		9		A		B		C		D		E		F	
长调用		程序结束地址(IP)		程序结束地址 (CS)		Ctrl-Break退出地址(IP)									
10		11		12		13		14		15		16		17	
Ctrl-Break退出地址(CS)		致命错误退出地址 (IP)		(CS)		保留									
18		19		...2A		2B		2C		2D		2E		2F	
保		留		环境块段地址								保		留	
30			4F	
保		留		保								留			
50		51		52		53		54		55		56		57	
DOS调用		保								留					
58		59		5A		5B		5C		5D		5E		5F	
保		留		标准FCB1											
60		61		62		63		64		65		66		67	
标准FCB1 (续)															
68		69		6A		6B		6C		6D		6E		6F	
标准 FCB1 (续)				标准 FCB2											
70		71		72		73		74		75		76		77	
标准FCB2 (续)															
78		79		7A		7B		7C		7D		7E		7F	
标准 FCB2 (续)															
80		81		82		83		84		85		86		87	
程序长度		左留空的命令参数													
88			FF	
默认的磁盘传送区 DTA															

1. 可用内存的第一段以段形式给出，例如1000H 表示64K。
2. 偏移量6处的字包含有段中可用字节数。
3. 偏移量2CH 处包含环境的段地址。
4. 程序不得改变偏移量5CH 下面 PSP 的唯一部分。

第五节 文件控制块

	FFH	全0	属性	FCB扩展
0	驱动器	文件名(8字节)或保留的设备名		
8		文件扩展名	当前块	记录大小
16	文件大小(低)	文件大小(高)日期		
24	为系统保留			
32	当前记录	随机记录号(低)	随机记录号(高)	

一、标准文件控制块

标准文件控制块(FCB)按如下定义，偏移量为十进制：

字节	功	能
0	驱动器号。	例如
	打开前：	0——默认驱动器 1——驱动器A 2——驱动器B，等等；
	打开后：	0——驱动器A 1——驱动器A

2——驱动器B, 等等;

打开过程中由实际驱动器号来代替0。

- 1—8 文件名, 左对齐, 填以空格。如果此处为一个保留的设备名(如LPT1), 则不要包括可选的冒号。
- 9—11 文件扩展名, 左对齐, 填以空格(可以全是空格)
- 12—13 从0开始的与文件开始有关的当前块号(由Open功能调用设置为0)。一个块由128个记录组成, 每一个的记录大小在逻辑记录大小域指出。当前块号与当前记录域一起提供给以后的读、写用。
- 14—15 以字节计的逻辑记录大小。由Open功能调用设置为80H, 如果不正确, 必须设定此值, 因为DOS用它来确定文件中的适当位置作为磁盘读、写用。
- 16—19 以字节计的文件大小。此域为2个字, 第一个字为低级部分。
- 20—21 注明文件建立成最后一次更新的日期。mm/dd/yy与诸位的关系为:

21								20							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
y	y	y	y	y	y	y	m	m	m	m	d	d	d	d	d

其中: mm为1—12, dd为1—31。

yy为0—119(1980—2099)

- 22—31 为系统保留。
- 32 当前块中的当前相对记录号(0—127)。在作后面磁盘读/写操作前必须先设置此域, 此域不由Open功

能调用所初始化。

33—36 以文件开始处为0算起的相对记录号。在对磁盘作随机读/写之前必须设置此域。此域未由Open功能调用所初始化。

如果记录大小小于64字节，两个字都使用。否则，使用前三个字节。注意：使用了程序段中5CH处的FCB，FCB的最后一个字节将与无格式参数区的第一个字节相重叠。

注：

1. 未打开的FCB，由FCB前缀(若用的话)，驱动器号，正确填写的文件名/扩展名组成。打开的FCB，其未填写域已被 Create 或 Open 功能调用所填写。

2. 字节0—15和32—36必须被用户程序所设置。字节16—31由DOS设置，并且用户程序不得改变其内容。

3. 字域存储时先存低位。例如，一个记录长度为128，存储时在偏移量14处为80H，偏移量15处为00H。

二、扩展文件控制块

扩展的FCB用于建立或搜索磁盘目录中具有特殊属性的文件。

它为FCB加了一个7字节的前缀，格式内容为：

字节

功 能

FCB—7 内容为FFH的标志字节以标明这是一个扩展的FCB。

FCB—6到
FCB—2

保留

FCB—1 属性字节。见本书“DOS磁盘目录”中关于属性位的定义。此功能使应用程序可将自己的文件定义为隐式(hidden)(从而在目录查找时看不到)。

DOS功能调用对于FCB的引用(无论是打开的还是未打开的),既可以是普通的也可以是扩展的FCB,如果使用了扩展的FCB,某个寄存器应设置为程序段前缀的第一个字节,而不是设置为驱动器号域。

第八章 在应用程序内执行命令

第一节 引言

应用程序可以引用命令处理程序的二级副本。所编写的程序可将DOS命令作为参数来传递，二级命令处理程序处理这项命令时如同它们是从标准输入设备上输入的一样。

第二节 引用命令处理程序

引用命令处理程序的步骤如下：

1. 确保存在有足够的自由内存(DOS2.10和3.00要求17K字节，DOS3.10要求23K字节)可容纳命令处理程序的副本(二级命令处理程序)和欲执行的命令。这由执行功能调用4AH来实现。下一步执行功能调用48H，使BX = FFFFH。由此返回可用内存的大小。

2. 按如下形式为二级命令处理程序建立一个参数字符串：

1个字节 = 参数字符串长度

xx个字节 = 参数字符串

1个字节 = 0DH (回车)

例如，下面的汇编程序所建立的字符串引起DISKCOPY命令的执行：

DB19, "/CC:DISKCOPY A:B:", 13

3. 使用EXEC功能调用（4BH，功能值为0）引起命令处理程序二级副本的执行，命令处理程序所在之驱动器、目录和名字可在程序段前缀PSP中PSP + 2CH处“环境域”中的语句“COMSPEC = 参数”传送得到。应记住设置EXEC控制块的偏移量2以指向上面所建立的字符串。

第九章 硬盘信息

第一节 引言

IBM PC硬盘（固定盘）结构的设计满足以下要求：

- 允许多种操作系统使用硬盘，在更换操作系统时不需保留与重新恢复。
- 允许用户选择的操作系统从硬盘上启动。

第二节 硬盘结构

硬盘结构如下：

• 为了使各操作系统共享硬盘，可以将磁盘在逻辑上分为1至4个分区。每个分区的空间是连续的，可以为某个操作系统专用，而且每个操作系统只能占用一个分区。分区数和分区的大小可以由用户用硬盘实用程序选择。DOS实用程序是FDISK·COM。分区信息存放在磁盘第一个扇区中的固定盘的自举记录中。

• 任何操作系统必须将其分区看作整个盘，并确保其使用，操作时不涉及盘上的其它分区。

• 每个分区可以在它的第一个扇区包含一个自举记录和

你所选取的程序或数据，包括操作系统副本。例如，DOS FORMAT命令用于格式化（并放入DOS副本）DOS分区，方法与软盘格式化相同。可以用FDISK应用程序将分区指定为“可自举的”（激活），当系统启动或重新启动时，主硬盘自举记录使得该分区的自举记录得到控制。

第三节 系统初始化

系统初始化（或系统自举）顺序如下：

1. 系统初始化首先从软盘驱动器A引导操作系统。如果驱动器没有准备好或者发生读错误，则再从系统中读取第一个硬盘的第一个扇区的主硬盘自举记录。如果还不能成功地引导，说明主硬盘自举记录不存在，则必须启动ROM BASIC。

2. 如果自举成功，主硬盘的自举记录将给出控制信号并自动地检查分区表。如果有一个项目表示分区“可自举的”（激活），则自举记录从分区的第一个扇区读出并给与控制。

3. 如果没有可自举的分区，则需启动ROM BASIC。

4. 只要有一个自举指示灯失效，哪怕是大多数指示灯表示可自举，则会显示“Invalid partition table”（非法分区表），系统将进入死循环，此时你只要在驱动器A中插入系统盘，使用系统重置（或总清）重新起启动软盘。

5. 如果有五次不能成功地读出分区自举记录，将显示指错信息“Error loading operating system”（操作系统引导错），此时系统进入死循环。

6. 如果分区自举记录中无有效的“签字”，系统将给出信

息“Missing operating system”（未发现操作系统）并进入死循环。详见本书“自举记录分区表”中有关自举记录的说明。

注意：当改变分区的大小或位置时，必须确认盘上的所有数据已有备份，因为在对硬盘分区过程中将失去以前分区存入磁道的信息。

第四节 自举记录/分区表

硬盘的自举记录必须写在硬盘的第一个扇区中，包括如下内容：

1. 装入代码和将控制转给（四个可能的操作系统的一个）自举记录的代码。

2. 分区表在自举记录的末尾，每个表项有16个字节长，每个表项包含起始和结束的柱面、扇区和头号，也即四个可能分区中每个分区之前的扇区数和分区占用的扇区数。自举记录用“自举指示器”字节来决定一个分区中是否装有操作系统。FDISK 初始化程序将相应分区的自举指示器字节填入80H值以表明该分区是可自举的，同时把其它分区的指示器置成0值。80H告诉标准自举程序引导自举扇区，扇区的位置包含随后的三个字节中，该扇区是用户所选择的操作系统的真正的自举记录，其作用是将操作系统的其余部分引导进来（象从软盘中引导一样），所有的自举记录均引导到绝对地址0:7C00。

下面给出了分区表及其在自举记录中的偏移量。

偏移量	用途
1BE	分区 1 开始
1C2	分区 1 结束
1C6	分区 1 相对扇区
1CA	分区 1 扇区数
1CE	分区 2 开始
1D2	分区 2 结束
1D6	分区 2 相对扇区
1DA	分区 2 扇区数
1DE	分区 3 开始
1E2	分区 3 结束
1E6	分区 3 相对扇区
1EA	分区 3 扇区数
1EE	分区 4 开始
1F2	分区 4 结束
1F6	分区 4 相对扇区
1FA	分区 4 扇区数
1FE	标记

头扇柱

自举指针	H	S	CYL
系统指针	H	S	CYL
低	字	高	字
低	字	高	字
自举指针	H	S	CYL
系统指针	H	S	CYL
低	字	高	字
低	字	高	字
自举指针	H	S	CYL
系统指针	H	S	CYL
低	字	高	字
低	字	高	字
自举指针	H	S	CYL
系统指针	H	S	CYL
低	字	高	字
低	字	高	字
55H	AAH		

第五节 硬盘技术说明

一、自举指示器(Boot Ind)

自举指示器为0，表明分区不能自举；自举指针为80H则可以自举。只能有一个分区可举。

二、系统指示器(Sys Ind)

系统指示器指明“拥有”该分区的操作系统。系统指示器为 00H——无定义

01H——DOS12位FAT

04H——DOS16位FAT

三、柱(CYL)和扇区(S)

标有CYL的一个字节字段含有低八位的柱面号，高二位是在扇区字段中的高二位，这与ROM BIOS中断13H（磁盘I/O）的要求相对应，可以是10位柱面号。这样定义柱面和扇区字段是为了使ROM BIOS引导相应的自举记录只用两条MOV指令就可以设置DX和CX，硬盘自举只能从系统的第一个硬盘上实现，它的BIOS驱动号（80H）对应于自举指示器字节。所有分区都按柱面的倍数分配，且从0头1扇区开始。有一种情况例外，在磁盘开头分配的分区是从扇区2开始的，因为1扇区是主自举记录。

四、相对扇区(Rel Sect)

磁盘每个分区之前的扇区号存放在标有“rel sect”的4字节字段中，该值是从磁盘的0柱面、0头、1扇区开始计数得到的，扇区、头、磁道的序号递增直至读分区开始。因此，若磁盘共4个头，每个磁道有17个扇区，且第二个分区从1柱面、1扇区、0头开始，则该分区的相对扇区为68（十进制）。该字段先存储最低有效字。

五、扇区数(#Sects)

分配给分区的扇区号保存在“#of sects”字段中，该4字

节字段先存储最低有效字。

六、标记(Signature)

自举记录的最后两个字节(55AAH)用作表示自举记录有效的标记,该记录和分区自举记录都包含偏移量(1FEH)的标记。

如果没有指示器字节指示“可自举的”系统,则主磁盘自举记录将起动ROM BASIC。

当分区自举记录获得控制,就将其分区表入口地址送到DS:SI寄存器。

系统程序员设计的初始化/管理硬盘应用程序至少要提供如下功能:

1. 将主磁盘自举记录/分区表写到硬盘的第一个扇区。
2. 进行磁盘分区时,也就是当用户想建立分区时,能建立或更新分区表的信息(分区所有的字段),但只限于在每个分区建立一种操作系统,且允许整个磁盘重新分区或者增加一个分区而不影响已有的分区(由用户选择)。
3. 提供一种能使用户指定的分区可以自举的方法,同时能将所有其它分区的自举指示器字节清除。
4. 应用程序不会改变和移动其它操作系统的任何分区信息。

第六节 磁盘位置的确定

DOS用如下公式确定磁盘位置:

$$SPF = \frac{TS - RS - \frac{D * BPD}{BPS}}{CF + \frac{BPS * SPC}{BPC}}$$

其中:

TS——磁盘的总扇区数。

RS——磁盘开始处留给自举记录的扇区数。DOS保留1个扇区。

D——根目录中的目录项数。

BPD——每个目录项的字节数，通常为32字节。

BPS——每个逻辑扇区的字节数，一般为512字节。可以用VDISK指定不同的值。

CF——每个磁盘的FAT数，对于多数磁盘来说CF为2。可以用VDISK指定CF为1。

SPF——每个FAT的扇区数，最大值为64。

SPC——每个分配单元的扇区数。

BPC——每个FAT表的字节数。12位FAT的BPC为1,2,16位FAT的BPC为2。

第十章 EXE文件的结构及其加载

第一节 引言

本章内容有：

- EXE文件结构
- 重定位表

第二节 EXE文件结构

连接程序 Linker 所产生的 EXE 文件由两个部分组成：

- 控制信息和重定位信息
- 装入模块本身

控制和重定位信息处于文件开始的一个称为“文件头”(header)的域中，装入模块紧接在文件头后面，以 Linker 构造之模块的内存映像开始。

文件头格式如下：

十六进制偏移量	内 容
00—01	4DH, 5AH——此为链接程序的标记, 表示该文件是一个合法的 .EXE 文件。
02—03	模为 512 的映像长度 (即装入模块映像大小除以 512 后得到的余数)。
04—05	以 512 字节 (一页) 为增量的文件大小 (包括文件头)。
06—07	重定位表项的个数。
08—09	以 16 字节 (段落) 为增量的文件头大小。可用于定位文件中装入模块的起始处。
0A—0B	在装入程序尾端上方需要的 16 字节段落的最少数目。
0C—0D	在装入程序尾端的内存上方需要的 16 字节段落的最大数目。
0E—0F	堆栈段段落在装入模块中的位移。
10—11	控制交给模块时, SP 寄存器中的偏移量。
12—13	字的检查和——文件中所有字和的负值, 不管是否溢出。
14—15	控制交给模块时, IP 寄存器中的偏移量。
16—17	代码段段落在装入模块中的位移。
18—19	第一个重定位表的字节在文件中的位移。
1A—1B	覆盖号 (程序的驻留部分则用 0)。

注: 使用 16 进制偏移量 18—19 处的值去定位重定位表中的第一表项。

第三节 重定位表

在 18H 处的字定位了重定位表中第一个表项。重定位表由数目可变的重定位表项组成。表项数目包含在偏移量

06—07处。每个重定位表项包含有两个域——一个2字节的偏移量值和一个2字节的段地址值。这两个域表示了模块中的一个字在模块得到控制前需加以修改时，相对于该装入模块的位移。这一过程称之为重定位，由下列步骤完成：

1. 在执行加载操作的程序的驱留部分后面建立起程序段前缀PSP。

2. 将文件头的格式化部分(其大小在偏移量08—09处)读入内存。

3. 装入模块的大小由文件大小减去文件头大小得到。偏移量04—05及08—09可用作此运算。实际的大小则根据偏移量02—03的内容往下调整，应注意到1.10版之前的链接程序所生成的文件在该处总是放一个为4的值，而不管程序的实际大小。因此如果该位置的值为4，建议不要理睬它。根据高/低装入程序开关的设置情况，可决定装入该装入模块的合适的段地址。此段称之为起始段。

4. 装入模块读入内存，从起始段开始。

注：重定位表是诸重定位表项的一个无顺序组合。第一个重定位表项是文件中最低偏移量的那个。

5. 重定位表项被读入工作区(一次一个或一次多个)。

6. 每个重定位表项的段值与起始段值相加，由此而计算得到的重定位表项段值及偏移量值，指向装入模块中的一个字，所得结果送回到装入块中该字处。

7. 所有重定位表项处理后，SS和SP寄存器根据文件头中的值进行设置，并将起始段值加到SS上。ES和DS寄存器设置为程序段前缀的段地址。起始段值加到文件头CS寄存器上。此结果与文件头IP值一起用于给出对模块的控制。

第十一章 DOS内存管理

第一节 引 言

DOS对已分配了的和尚未分配的内存块进行追踪，并为应用程序把它的内存需求通知DOS提供了三个功能调用。它们是：48H用于分配内存块；49H用于解除以前已分配了的内存块；4AH(SETBLOCK)用于改变已分配内存块的大小。

第二节 控制块

DOS按如下方式管理内存：

DOS为每个内存块(不管是自由的还是已分配的)建立一个控制块。例如，假定一个程序要求给予内存分配，DOS查找能够满足该请求的自由内存块，然后将所请求的内存大小从该块中“划”出。将该块的第一个字节传送给请求内存的程序，然后建立起二个内存管理控制块。第一个用于描述已分配块；第二个内存管理控制块描述原来自由内存块中所剩余的空间大小。如果发出“Setblock”以压缩一个已分配了的块，DOS为将成为自由的区域建立一个内存管理控制块，并将它加入到控制块“链”中去。因此，如果一个程序要改变不

是分配给它的内存的内容，便可能破坏 DOS 内存管理控制块。由此将引起无法预料的后果，而且只有当DOS使用到它的控制块“链”时，才会表现出来(一般情况下会产生内存分配出错，此时唯一的修复办法是重新启动系统)。

一个程序(命令或应用程序)被装入后，DOS使用 EXEC 功能调用(4BH) 执行加载，此功能调用亦可用于在应用程序中加载另一个程序。此功能调用有两个选择项：

- 功能0，装入并执行一个程序(命令处理程序用此功能来装入并执行外部命令)；
- 功能3，装入一个覆盖程序但并不执行。

虽然这两个功能调用以相同方式实行加载(为EXE文件实行重定位)，但它们对内存管理的方式都不同。

一、功能0

使用功能0装入和执行一个程序，EXEC首先分配出最大的自由空间块(新程序的程序段前缀PSP将处于该内存块的0偏移量处)。然后EXEC装入该程序。这样在多数情况下，新的程序“拥有”了从它的PSP到内存最高端之间的所有内存，包括由COMMAND.COM暂驻部分所占用的内存。如果这个程序又发生了自己的EXEC功能调用以装入并执行另一个程序，则由于已无自由空间可用来装入新程序将使该请求失败。

注：对于.EXE程序，如果有足够的内存供分配，则所分配给的内存量为程序内存映象的大小加上文件头 MAX ALLOC 域中的值(偏移量0CH处)；如果没有足够的内存可供分配，则 EXEC 分配给程序内存映象的大小加上文件头

MIN ALLOC 域中的值(偏移量0AA处)。这两个域的值是由链接程序 Linker 设置的。

“规矩”(Well-behaved)的程序使用 SETBLOCK 功能调用接收控制将分配给它的内存块压缩到它实际需要的大小。一个.COM 程序在实行 SETBLOCK 之前会建立起自己的堆栈,因为由 DOS 所提供的默认堆栈有可能处于将要变成自由的内存区中。这样释放了不再需用的内存后便可用于装入后面的程序。

如果该程序在处理过程中需要附加内存,可使用要求分配的功能调用得到附加内存,以后再释放内存的功能调用释放它。

使用 EXEC 功能0装入的程序退出后,其起始分配块(从其PSP开头的块),在调用程序重新得到控制之前自动被释放。所有程序在退回到调用程序之前都有责任释放它们所分配的内存。

二、功能3

功能3用于装入一个覆盖模块,不建立程序段前缀PSP,EXEC假定调用程序已经分配了内存用于装入新程序,故不用为其再分配内存。因此,调用程序或是在发出SETBLOCK调用以确定其所需内存量时考虑到要装入覆盖模块,或是在初始化时取得尽可能多的自由空间。然后调用程序应分配给一个块(基于欲被装入程序的大小)以容纳使用“load overlay”调用装入的程序。注意到“load overlay”并不查看调用程序是否实际地拥有已指明要装入的内存块,只是假定调用程序会遵循这一规则。如果调用程序并不拥有覆盖模块将被装入的内

存，则便有可能被装入的该程序覆盖掉 DOS 用于监控内存块的控制块之一。

使用功能3装入的程序并不发出 SETBLOCK 调用，因为它并不拥有运行所用的内存（此内存被调用程序所拥有）。

因为使用功能3装入的程序直接由调用程序交给某控制（并也是直接返回控制给调用程序），而不用DOS干预，故被调用程序退出时无内存被自动释放，由调用程序决定退出程序所占用内存的重新布置。应注意到，如果退出程序曾分配过内存，它便应负责在退出前将那些内存释放。