

HOPE

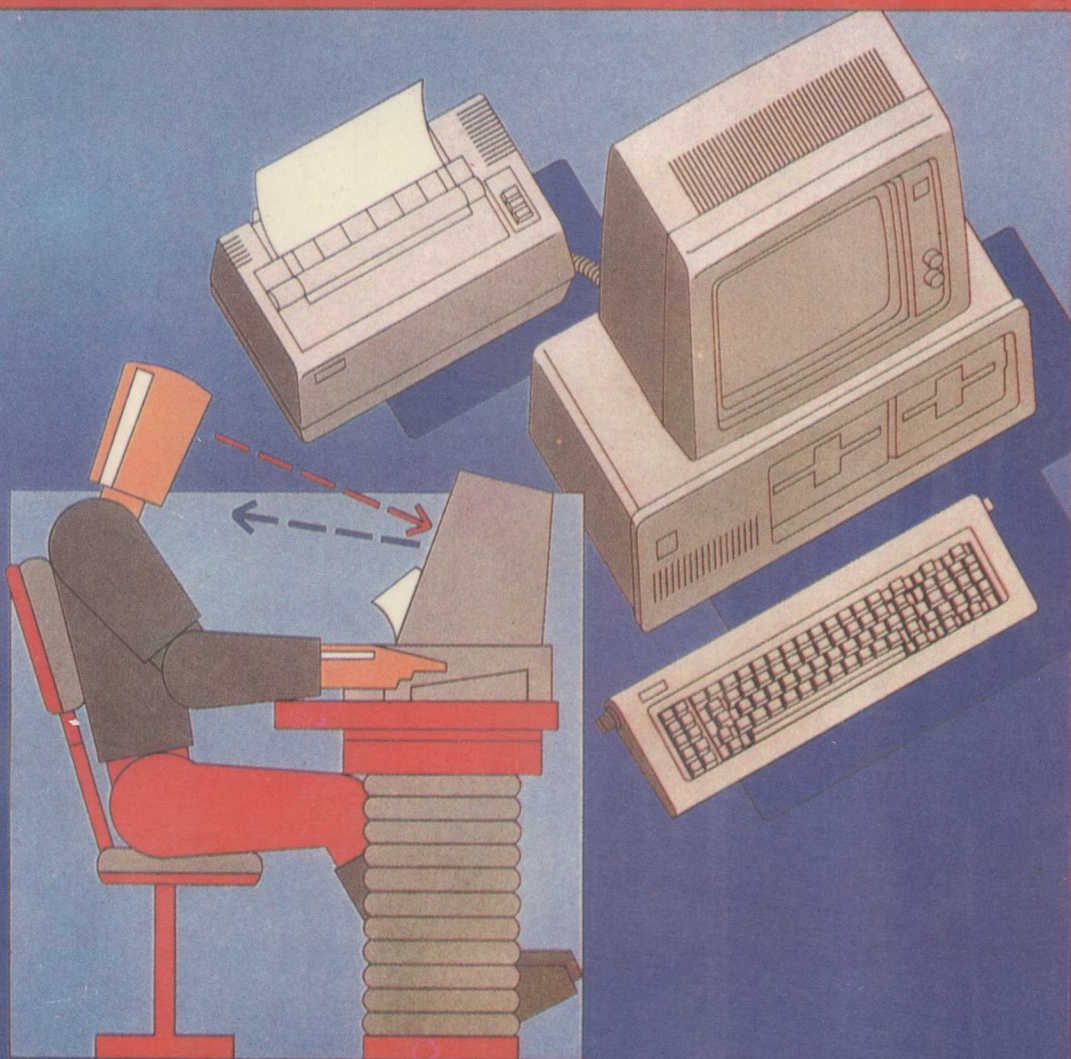
(共二册)

IBM PC 8086、8088、80286

微机原理

与宏汇编语言的DOS BIOS 程序设计教程

(下册)



北京希望电脑公司



72.5764
170/3
IBM PC 8086、8088、80286

微机原理

与宏汇编语言的 DOS BIOS 程序设计教程

(下册)

钟应木 章忆文 编译



北京希望电脑公司

一九九一年十月

前 言

本书以已普及的 80286 为基础,介绍 IBM PC 的系统结构及对各个外设的控制。本书还以最流行的操作系统 DOS 3.3(或以上版本)和标准的 BIOS 为背景,以宏汇编语言(MASM)为工具,介绍对外设的控制编程。内容兼容于 8086、8088 和基础的 80386。

第一部分首先介绍 IBM PC 80286 的数据表示法以及硬件结构。然后说明宏汇编语言的编写、汇编、连接、执行的基础知识。按功能分成数据的传送,简单的四则运算,分支与循环,位运算指令,字串指令,子程序和宏调用,分别介绍汇编语言指令。

第二部分围绕 DOS 系统资源与应用,按不同设备分别介绍 DOS 的系统调用,说明基本字符的输入/输出、日期与时间的管理、驱动器目录管理、使用文件句柄的文件管理、驱动器的管理、进程管理、设备的管理、使用文件控制块的文件管理以及 DOS 其它资源的管理。

第三部分以 BIOS 服务为主线,介绍声音的产生,文本模式下的屏幕处理,图形模式下的屏幕绘图,键盘的 BIOS 服务,打印机的 BIOS 服务,磁盘的 BIOS 服务,时间与日期的 BIOS 服务以及硬盘等其它 BIOS 服务。

在第一部分介绍汇编指令时用丰富的例子说明用法,在第二、第三部分介绍 DOS 和 BIOS 对外设的控制时定义了每个调用的宏,并列举了精巧的宏调用示例。

本书言简意赅,循序渐进,示例丰富,只要稍有一点 PC 或汇编语言知识的人,就可以做为以系统资源控制和应用编程的参考书。它又是一本教程,可以作为授课和培训之教材。

编者

第十三章 磁盘驱动器目录管理系统

本章学习目标

- 1: 让读者彻底了解, 磁盘文件及磁盘目录结构。
- 2: 读者研习本章内容后, 应可配合各功能程序, 编写与目录操作有关的系统程序。

本章内容

- 13-1: 磁盘结构的说明
- 13-2: AH=39H, 建立子目录
- 13-3: AH=3AH, 删除子目录
- 13-4: AH=3BH, 改变当前的工作目录
- 13-5: AH=41H, 从目录中删除一个文件
- 13-6: AH=43H, 文件属性的获得/设定
- 13-7: AH=47H, 取得工作目录的路径
- 13-8: AH=4EH, 寻找第一个匹配的文件
- 13-9: AH=4FH, 寻找下一个匹配的文件

13-1: 磁盘结构的说明

每个磁盘在使用前, 一定要先使用 **FORMAT** 指令将磁盘表面划分成磁道和扇区(tracks and sectors)。一个标准的 360k 单面软磁盘在可用面上有 40 个磁道, 而双面磁盘则每面有 40 个磁道, 共计是 80 个磁道。每个磁道都有一个编号, 最外圈的是第 0 道, 依此类推, 可知, 最内圈的是第 39 道。

而每磁道又被细分成许多扇区, 早期的 DOS 1.X 版本每道分成 8 个扇区, 而在 DOS 2.X 和 3.X 版本以后, 每道有 9 个扇区, 由于早期的 DOS 1.X 版本现已不多见, 所以本书将重点放在 2.X 和 3.X 版本的磁盘。由于每磁盘有 80 道, 而每道有 9 扇区, 所以可知:

$$80 \times 9 = 720 \dots \dots \dots \text{每片磁盘有 720 个扇区}$$

和磁道一样, 我们也将扇区予以编号。其中, 第 0 面第 0 道第 1 扇区称为第 0 扇区, 第 0 面第 0 道第 2 扇区称为第 1 扇区, $\dots \dots \dots$ 第 0 面第 0 道第 9 扇区称为第 8 磁区, 第 1 面第 0 道第 1 扇区称为第 9 扇区, $\dots \dots \dots$, 其余依此类推, 如图 13-1 所示。

扇区编号	面	磁道	扇区
0	0	0	1
1	0	0	2
2	0	0	3
.	.	.	.
8	0	0	9
9	1	0	1
.	.	.	.
719	1	39	9

图 13-1 扇区编号的说明

由于每个扇区可容纳 512 个字节数据，所以可知：

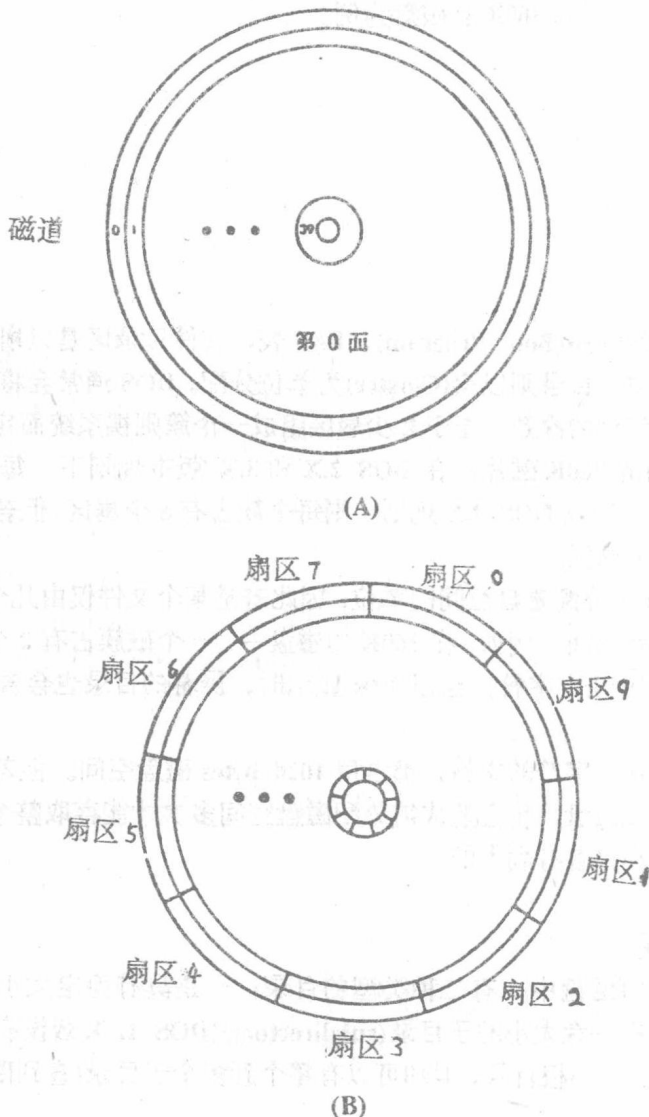
$$720 \times 512 = 368640 \text{ (360K)} \dots\dots \text{每个磁盘的容量(字节)}$$

磁道和扇区的产生是由 FORMAT 完成的，然而此指令除了将磁盘做适当的规划之外，还会在每一磁盘上建立两个与文件管理息息相关的结构：文件分配表 (File Allocation Table, FAT) 和文件目录 (File Directory)。每个磁盘拥有两个 FAT，第一份存在第 0 面第 0 道第 2 和第 3 扇区内，第二份则存在第 4 和第 5 扇区内。而文件目录则存在第 0 面第 0 道第 6 到第 9 扇区与第 1 面第 0 道第 1 到第 3 扇区内。13-2 图是目前常见的软磁盘格式说明。图 13-3 则是磁盘结构的说明。

容量	面	磁道	扇区	总扇区数	*目录	*FAT
360K	2	40	9	720	7	4
1.2M	2	80	15	2400	14	14

*目录和 FAT 表示所占用的扇区数

图 13-2 目前常见软盘格式的说明



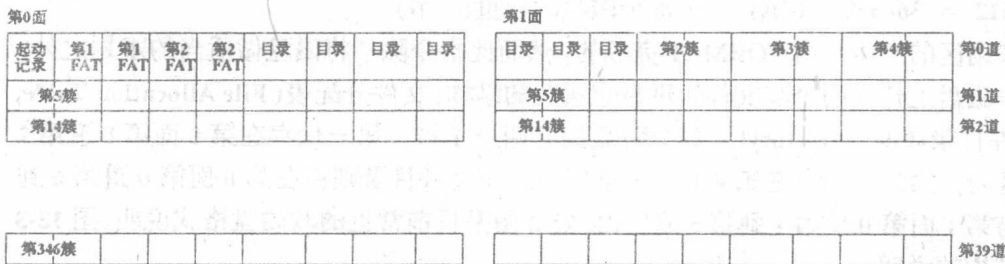


图 13-3 (A)每片磁盘有 40 个磁道(0~39)
 (B)每个磁道有 9 个扇区(1~90)
 (C)FAT, 文件目录与簇的配置情形
 上图以 360K 软磁盘为例。

然而磁盘,除了引导程序(Boot Program), FAT 表,文件目录区是以扇区为单位分配外,其他正式分配文件或子目录则以簇(Cluster)为单位分配。DOS 通常会将连续的扇区组合成簇,以减少读写头移动的次数。至于多少扇区组成一个簇则视系统而定。

例如,市面上常用的 360K 磁片,在 DOS 2.X 和 3.X 版本规划下,每个簇有 2 个扇区。而 20 Mega 的硬盘,若以 DOS 2.X 规划,则每个簇占有 8 个扇区,但若改以 DOS 3.X 规划,每个簇只占有 4 个扇区。

刚刚已说过簇是 DOS 分配磁盘空间的单位,因此若是某个文件仅由几个字符所组成,此文件也将会占据整个簇空间。例如:在 360K 的磁盘中,一个磁簇占有 2 个扇区,即 1024 bytes,若有一个文件只有几个字符,当以 DIR 显示时,屏幕的目录也会告诉你,此文件大小为 1024 bytes。

从上段叙述可知,几个字符的文件,都占用 1024 bytes 磁盘空间。但若文件很大,而簇空间很小,则数据存取磁盘,将会造成需分配磁盘空间多次才能存取整个文件,如此速度会很慢。因此,簇大小是各有利弊的。

13.1.1 磁盘目录的说明

在 IMB PC DOS 的磁盘中,有二种类型的目录:一是具有固定大小的根目录(root directory),另一是可具有任意大小的子目录(subdirectory)(DOS 1. X 版没有子目录)。而每一个磁盘有一个(也只有一个)根目录,但却可以有零个到多个子目录(直到磁盘装满为止)。

而子目录之间又可以有其它的子目录，这也是所谓的层次目录结构(又称树状目录)，如图 13-4 所示。

磁盘内文件目录就像是磁盘内容的表格，磁盘内每个文件在目录内都有一个对应的目录项。每个目录项长度是 32 bytes，从图 13-2 可知 360K 磁盘有 7 个目录扇区，所以：

$7 \times 512 = 3584$ bytes……目录所占空间数

$3584 / 32 = 112$ ……每个磁盘的目录容量

对 1.2Mega 的磁盘而言，每个目录项长度和 360k 磁盘一样是 32bytes，从图 13.2 可知 1.2Mega 磁盘有 14 个目录扇区，所以：

$14 \times 512 = 7168$ ……目录所占空间数

$7168 / 32 = 224$ ……每个磁盘的目录容量

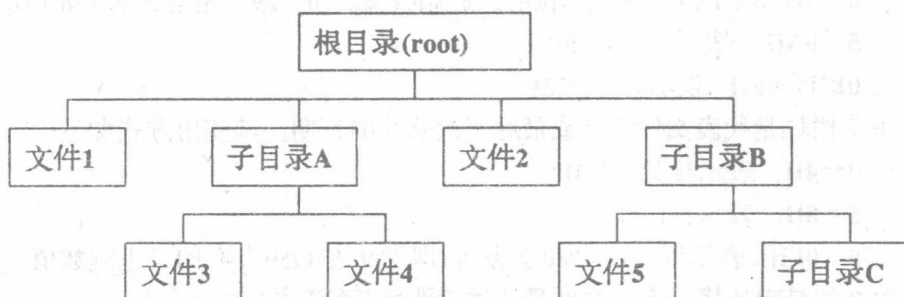


图 13-4 DOS 的层次目录结构

而 32 bytes 长的目录项，又可分成 8 个位，如图 13-5 所示。

00	08	0B	0C	16	18	1A	1C	1F
文件名称	扩展	属性	保留	时间	日期	起始簇	文件大小	

图 13-5 文件目录项目的结构图

图 13-5 中的每个位是文件名称，这个位占有 8 个字节，如果文件名称少于 8 个字符则其余的以空白填补，然而目录项中，文件名称位的第一个字节，有其特殊意义：

00H: 代表这个目录项从未被使用过，

05H: 文件名称的第一个字符实际上是 E5H，但 E5H 有其特殊的意义——代表某文件已被删除。因此 DOS 会将 E5H 开头的文件名称，更改成 05H。

2EH: 表示该目录所对应的文件是一个亲代目录。此时起始簇会指向目录本身所在的簇中。若是下一个字节也是 2EH，则起始簇会指向父目录所在的簇，若父目录是根目录，则其值为零。

E5H: 表示文件已被删除。其实删除，并不是将整个文件删除，只是把第一个字节设成此值，所以要挽回被删除的文件，可利用此特性。

目录项目的第 2 个名称是扩展名，如果扩展名少于 3 个字符，则以空白填补。

第 3 位是文件属性，其中每个位代表一个属性，若某位为 1 表示该文件具有此属性，反之则否。下列假设各位元值是 1。

位 0: 只读文件。只能读取, 无法执行写入和删除。

位 1: 隐含文件。使用 Dir 无法看到此文件

位 2: 系统文件。启动 DOS 系统时所使用的文件。

位 3: 磁盘卷标。这个项目一定要在根目录内, 且一个磁盘仅能有一个这种项目。

位 4: 子目录。设定此位时, 该目录项目表示是子目录。

位 5: 处理位。只要文件被修改, 这个位都会被设定。

位 6: 保留

位 7: 保留

第 4 位是保留, 预备给较新版本的 DOS 使用。

第 5 位是代表文件建立或最后一次修改的时间, 其使用方式如下:

位 0~4H: 以 2 秒为增量单位的二进制数值。(0~29, 相当于 0~58 秒)。

位 5~0AH: 表示分(0~59)

位 0BH~0FH: 表示时(0~23)

每个栏位是代表文件建立或最后一次修改的日期。其使用方式如下:

位 0~4H: 表示日其(1~31)

位 5~8H: 表示日份(1~)

位 9~0FH: 表示年, 以 1980 为基值(即以 0 表 1980 年) 的二进制数值

第 7 位是起始簇, 表示文件是从磁盘那里开始存放。

第 8 位是文件大小, 这个位有 4 个字节长, 储存时先存入低阶的两个字节, 放在低地址处, 然后再存入高阶的两个字节, 放在高地址处。

13.1.2 FAT 的说明

文件分配表(FAT) 主要是记录磁盘空间(簇)使用的情形。如图 13-3 所示, 最常用的 360K 磁盘, 每个 FAT 占用二个扇区, 对大多数磁盘格式而言, DOS 都拥有两个一模一样的 FAT, 用以预防一个损坏或不可读时当备份用, 但在一般情况下, DOS 并不会使用多余的那一个 FAT 的。

FAT, 从字义上可知道是一个表格, 表格上则被划分成许多位, 每个位恰有一个簇与之配合, 如图 13-6 所示。

FAT 位有 12 位和 16 位两种, 12 位的 FAT 格式, 较常用且较复杂(例 360K 磁盘使用此格式), 至于 16 位的 FAT 格式则一般用在硬盘上。

由于 FAT 位的前二个位(3 个字节)代表特殊意义, 所以并不指向簇编号, 这也是为什么, 我们从 2 开始进行簇编号。FAT 第一位的第一个字节又被称为是磁盘的 ID 主要用于辨别磁盘格式。其所代表意义如下:

FDH: 一般常用 360K 的磁盘

F9H: AT 用 1.2Mega 的磁盘

F8H: 硬盘

其它 FAT 内容值的意义如下:

000H: 表示此位所对应的簇尚未使用。

FF7H: 表示此位所对应的簇已损坏, 不可使用。

FF8H~FFFH: 表示这是文件的最后一个簇。

其他值：此值为下一个簇的编号。

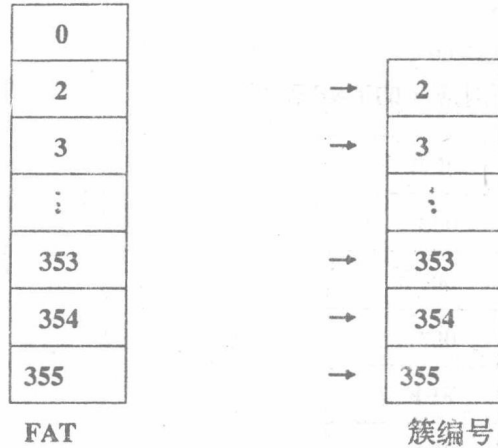


图 13-6 FAT 与互相对应的簇

先前已知，目录项 1AH~1CH 栏位会指向该文件起始簇的编号。假设有一文件，目录项，FAT，簇关系如图 13-7 所示。则磁盘在读取该文件时，首先会从目录项得知文件 A 起始地址在第 3 簇，由于第 3 个 FAT 是 4，因此可知此文件另一部分内容放在第 4 磁簇内，由于第 4 个 FAT 值是 5，所以第 5 个簇也存在文件 A 的内容，因为第 5 个 FAT 值是 FFE (介于 FF8H~FFFH 间)表示这是最从一个簇，所以可知文件 A 是由 3,4,5 簇所组成。

由于每个 FAT 位是由 12 个位所组成，所以要特别注意 FAT 位各字节所代表的意义，如图 13-8 所示：

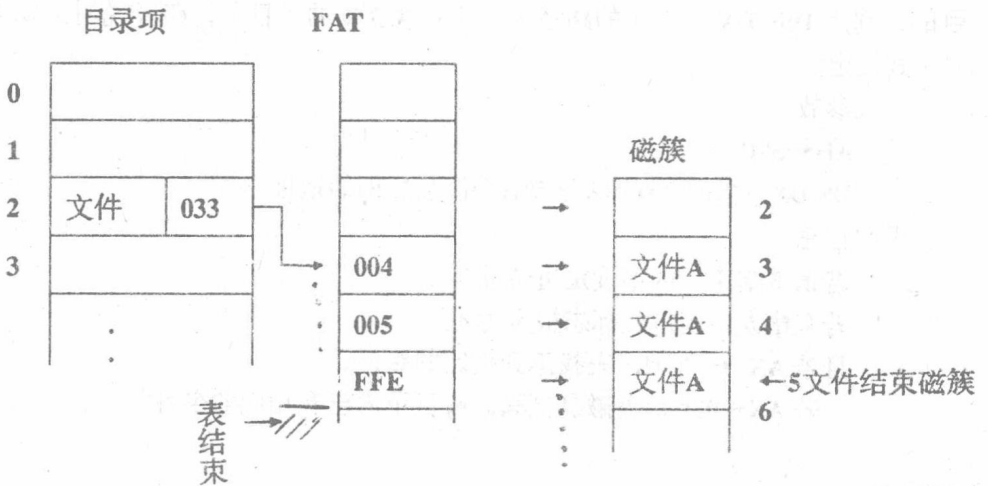


图 13-7

至于 FAT 位为 16 位型式的硬盘，只要按上表推算即可很容易得知，硬盘 FAT 表各字节所代表的意义。

在 FAT 讨论结束前，我想要解释一下磁道损坏的工作情形。通常，当磁盘进行格式化(FORMAT)时，若碰上有簇损坏情形，系统会将之记录在 FAT 内，此时损坏簇所对应的 FAT 内容值为 FF7H，而 DOS 在使用簇时会自动避开这些坏簇区域，所以在用磁盘，若碰上磁道损坏，并不是严重的问题，不必担心文件会被写到坏的磁道上，因为这是不可能的。

以上所述是磁盘的基本概念，至于如何使用 DOS 功能编写与文件目录有关字节：

0 1 2 3 4 5 6 7 8 ……

内容: FD 00 00 01 20 00 FF 0F 00

所对应↓的 FAT 表

0	0FD
1	000
2	001
3	002
4	FFF
5	000

图 13-8. 存储器内存和 FAT 表

的汇编语言程序，则留待各小节再一一说明。

另外有一点要注意的是，在进行层次文件处理时，常会用到一种 ASCIIZ 字串，所谓 ASCIIZ 字串就是以零字符为结尾的字串，此字串常被用以描述层次路径名，详情则请参阅各小节内容。

13-2: AH=39H, 建立子目录

目的: 利用 DS: DX 所表示的路径名说明，建立新的子目录，路径说明是以 ASCIIZ 字串的形式设定。

输入参数

AH←39H

DS:DX←指向 ASCIIZ 字串(即路径说明)的地址

返回信息

若正常结束，则不设定进位标志。

若有错误，则进位标志被设为 1。

且若 AX ← 03H，表找不到指定的路径。

若 AX←05H，表磁盘已满，或目录内已有相同的名称。

宏定义:

```
;
; Function Request 39H
; Make Directory
;
MKDIR MACRO PATH
    MOV    DX,OFFSET PATH
    MOV    AH,39H
    INT    21H
```

ENDM

程序范例 CH13_1.ASM:

建立一个子目录(Subdirectory)的应用程序。本程序在执行的时候,会在 B 磁盘驱动器中建立一个 ABC 的子目录(第 12 行中设定此一子目录)。

```
01 ;=====PROGRAM DESCRIPTION =====
02 ;   PROGRAM NAME : CH13_1.ASM
03 ;   Function Request 39H Application.
04 ;       FOR : 8086/8088/80286
05 ;=====
06 ;
07 STACK SEGMENT STACK
08     DW     1024 DUP(7)
09 STACK ENDS
10 ;
11 DATA SEGMENT
12 PATH  DB     'B:\ABC',0
13 ERROR DB     'CREATE ERROR','$'
14 DATA ENDS
15 ;
16 ;=====
17 ;   Function Request 39H
18 ; Make Directory
19 ;
20 MKDIR MARO    PATH
21     MOV     DX,OFFSET PATH
22     MOV     AH,39H
23     INT     21H
24     ENDM
25 ;
26 CODE SEGMENT
27 PROG  PROC    FAR
28     ASSUME  CS:CODE,DS:DATA,SS:STACK
29 ;INITIAL THE PROGRAM
30     PUSH   DS
31     MOV    AX,0
32     PUSH  AX
33     MOV    AX,DATA
34     MOV    DS,AX
35 ;
```

```

36      MKDIR    PATH
37      JC       ERR
38      RET
39 ERR:
40      LEA      DX,ERROR      ;PRINT ERROR MESSAGE
41      MOV      AH,09H
42      INT      21H
43      RET
44 PROG  ENDP
45 CODE  ENDS
46      END      PROG

```

执行结果:

```

A>DIR B:
Volume in drive B is CH8-12
Directory of B:\

CH13_3  EXE  2637  4-22-88  9:36a
PT      COM  154  11-04-87  9:43a ←执行前目录表
2File(s) 358400 bytes free

```

A>CH13_1 ←————程序的执行

```

A>DIR B:
Volume in drive B is CH8-12
Directory of B:\

ABC <DIR> 4-21-88 12:03a ←————新建的子目录
CH13_3  EXE  2637  4-22-88  9:36a ←执行前目录表
PT      COM  154  11-04-87  9:43a
3 File(s) 357376 bytes free
A>

```

13-3: AH=3AH, 删除子目录

目的:删除 DS:DX 所表示路径说明的子目录, 路径说明是以 ASCII 字串的形式设定。

输入参数

AH←3AH

DS:DX←指向 ASCII 字串(即路径说明)的地址

返回信息

若正常结束, 则不设定进位标志。若有错误, 则进位标志被设为 1。

AX←03H, 表示找不到指定的路径。

AX←05H, 表示拒绝执行。

AX←16H, 表示要除去的目录是
目前所处的目录。

附注: 若执行不成功, 且 AX=05H, 此功能会拒绝执行删除子目录, 此拒绝的原因不外乎是, 这不是一空目录, 或欲删除根目录, 或所欲删除的不是目录。

宏定义:

```
;
; Function Request 3AH
; Remove directory
;
RMDIR MACRO PATH
    MOV  DX,OFFSET PATH
    MOV  AH,3AH
    INT  21H
ENDM
```

程序范例 CH13_2.ASM:

删除一个子目录的应用程序。本程序在执行的时候, 会删除 B 驱动器内的 ABC 子目录(在第 12 行, 设定的要删除的子目录)。

```
01 ;=====PROGRAM DESCRIPTION =====
02 ;   PROGRAM NAME : CH13_2.ASM
03 ;   Function Request 3AH Application.
04 ;   FOR : 8086/ 8088/80286
05 ;=====
06 ;
07 STACK SEGMENT STACK
08     DW      1024 DUP(?)
09 STACK ENDS
10 ;
11 DATA SEGMENT
12 PATH  DB      'B:\ABC ',0
13 ERROR DB      'REMOVE ERROR','$'
14 DATA ENDS
15 ;
16 ;=====
17 ;   Function Request 3AH
18 ;   Remove Directory
19 ;
20 EMDIR MACRO  PATH
21     MOV      DX,OFFSET PATH
```

```

22     MOV     AH,3AH
23     INT     21H
24     ENDM
25 ;
26 CODE  SEGMENT
27 PROG  PROC FAR
28     ASSUME  CS:CODE,DS: DATA,SS:STACK
29 ;INITIAL THE PROGRAM
30     PUSH   DS
31     MOV    AX,0
32     PUSH  AX
33     MOV   AX,DATA
34     MOV   DS,AX
35 ;
36     RMDIR  PATH           ;REMOVE DIRECTORY
37     JC    ERR
38     RET
39     ERR:
40     LEA   DX,ERROR       ;PRINT ERROR MESSAGE
41     MOV   AH,09H
42     INT   21H
43     RET
44 PROG  ENDP
45 CODE  ENDS
46 END   PROG

```

执行结果:

```

A>DIR B:
Volume in drive B is CH8_12
Directory of B:\

```

```

ABC      <DIR>    4-21-88    12:03a
CH13_3  EXE 2637  4-22-88    9:36a    ←执行前的目录表
PT      COM 154  11-04-87   9:43A
3File(s) 357376 bytes free

```

```

A>CH13_2 ←程序的执行
A>DIR B:
Volume in drive B is CH8_12
Directory of B:\

```

```

CH13_3  EXE  2637  4-22-88  9:36a
PT      COM  154  11-04-87  9:43a←执行后的目录表
        2 File(s) 358400 bytes free ABC子目录已被删除
A>

```

13-4: AH=3BH, 改变当前的工作目录

目录: 这个功能会将当前工作目录, 移到 DS:DX 所表示的路径说明内。此一路径说明必须是 ASCII 字符串形式, 并且限定在 64 个字符以内。

输入参数

AH←3BH

DS:DX←指向 ASCII 字符串(即路径说明)的地址

返回信息

若正常结束, 则不设定进位标志。

若有错误, 则进位标志等于 1。

AX←03H, 表示路径说明无效。

附注: 路径说明无效的原因, 大概有下列几种, 找不到指定的路径, 或所指定的路径是一文件名, 而不是目录。

宏定义:

```

;
; Function Request 3BH
; Change current directory.
;
CHDIR MACRO PATH
    MOV  DX,OFFSET PATH
    MOV  AH,3BH
    INT  21H
ENDM

```

程序范例 CH13_3.ASM:

改变当前的工作目录。这个范例会将当前的工作目录, 转移到 ABC 的子目录中(子目录在第 12 行设定)。

```

01 ;=====PROGRAM DESCRIPTION =====
02 ;          PROGRAM NAME : CH13_3.ASM
03 ;          Function Request 3BH Application.
04 ;          FOR : 8086/ 8088/80286
05 ;=====
06 ;
07 STACK SEGMENT STACK

```

```

08      DW      1024 DUP(?)
09      STACK   ENDS
10 ;
11      DATA   SEGMENT
12 PATH  DB      'B:\ABC
13 ERROR DB      'CHANGE DIRECTORY ERROR', '$'
14 DATA   ENDS
15 ;
16 ;=====
17 ;   Function Request 3BH
18 ;   Change Current Directory
19 ;
20 CHDIR MACRO   PATH
21      MOV      DX,OFFSET PATH
22      MOV      AH,3BH
23      INT      21H
24      ENDM
25 ;
26 CODE   SEGMENT
27 PROG   PROC   FAR
28      ASSUME  CS:CODE,DS:DATA,SS:STACK
29 ;INITIAL THE PROGRAM
30      PUSH   DS
31      MOV    AX,0
32      PUSH  AX
33      MOV   AX,DATA
34      MOV   DS,AX
36      CHDIR PATH      ;CHANGE CURRENT DIRECOTRY
37      JC   ERR
38      RET
39      ERR:
40      LEA  DX,ERROR    ;PRINT ERROR MESSAGE
41      MOV  AH,09H
42      INT  21H
43      RET
44 PROG   ENDP
45 CODE   ENDS
46      END   PROG

```

执行结果:

A>DIR B:

Volume in drive B is CH8_12

Directory of B:\ ←原先的工作目录

```

=====
ABC          <DIR>    4-21-88    12:07a
CH13_3  EXE    2637    4-22-88    9:36a
PT          COM     154    11-04-87    9:43a
3 File(s) 357376 bytes free
=====

```

A>CH13_3 ←程序的执行

A>DIR B:

Volume in drive B is CH8_12

Directory of B:\ABC ←改变后的工作目录

```

.          <DIR>    4-21-88    12:07a
..         <DIR>    4-21-88    12:07a
2 File(s) 357376 bytes free

```

A>

13-5: AH=41H, 从目录中删除一个文件

目的: 删除 DS:DX 所表示路径名称的文件, 此路径名称是以 ASCII 字符串方式表示。

输入参数

AH ← 41H

DS:DX ← 预定删除文件的路径名称起始地址。

返回信息

若正常结束, 则进位标志等于零

若删除失败, 则进位标志等于 1

AX ← 02H, 表示找不到路径名称或文件名称。

AX ← 05H, 表示拒绝执行(例, 所设定的路径是目录, 或是只读文件)。

附注: 这项删除文件功能, 不可使用匹配符 “*” 及 “?” 等, 但 DOS 另一功能 AH=13H 可使用匹配字符。

宏定义:

```

;
; Function Request 41H
; Delete file
;
DELETE MACRO PATH
    MOV DX,OFFSET PATH ; FILE NAME
    MOV AH,41H
    INT 21H
    ENDM

```