第一章　操作系统的使用技巧

第一节　DOS 操作系统

DOS 的版本

在 PC 系列机及兼容机上运行的操作系统 PC-DOS(PC Disk Operating System)至今已正式发表了 10 个版本，除 DOS 4.00 和 DOS 5.00 支持多任务的并发功能外，其余的 8 个版本(DOS 1.00～3.30)均属于单用户单任务系统。尤其是 DOS 2.00 及以上各版本拥有上万个用户，其数量远远超过使用其他各类个人计算机操作系统用户的总和。

80 年代初，IBM 公司为其正在设计的 16 位 PC 机向一些大软件公司寻求配套的操作系统。当时，Microsoft 公司向 Seattle Computer Products 公司购买了 DOS 的前身 86-DOS 的专利权，并对其作了较大的改进，命名为 MS-DOS 作为该机的基本操作系统，并改名为 PC-DOS 1.00，这就是 PC-DOS 的第一个版本。为使当时运行于 8 位微机上的软件能方便地移植到 16 位的 PC 机上，DOS 的结构及内核都尽力模仿 8 位机的著名操作系统 CP／M，以后为了支持不断改进的硬件设备，PC-DOS 不断地升级版本。

版本	发表时间	支持硬件
1.00	1981 年 8 月	单面软驱 PC 机
1.10	1982 年 5 月	双面软驱 PC 机
2.00	1983 年 3 月	硬驱 PC／XT 机
2.10	1983 年 10 月	半高软驱 PCjr 手提式 PC
3.00	1984 年 8 月	高密软驱 PC／AT 机
3.10	1985 年 3 月	网络硬驱的 PC 服务器
3.20	1985 年 12 月	3.5 英寸软驱的轻便型 PC
3.30	1987 年 4 月	大容量硬驱的 PS／2

在 DOS 的 8 个单用户单任务的版本中，从 DOS 2.00 至 3.30，其核心功能未做大的更改，只是在 CP／M 的传统功能上，增添了类似于著名的多用户操作系统 UNIX 的许多特色，因此 DOS 2.00～3.30 的各个版本，其核心功能是两大操作系统——CP／M 和 UNIX 的完美结合，但相互之间还是存有一定的差异。只有清楚了解它们间的区别，才能更好地适应不同 DOS 运行的环境，充分利用高版本中增设的新命令和增强了的旧命令，以及所提供的系统功能调用，从而满足不同层次的应用需求。

一般来讲，随着 PC 机的普及应用，8 位操作系统 CP／M 烙印很深的 DOS 1.x 版本，使用得愈来愈少，应用软件的开发大都基于 DOS 2.00 以上。使用 10MB 容量硬盘的 XT 机用户，最好选用 DOS 2.x 版本，而使用 20MB 以上容量硬盘的 PC 机用

户，最好选用 DOS 3.x 版本。

DOS 各版本的兼容使用

PC-DOS 不断推出新版本以支持硬件技术的发展，虽然 DOS 有很好的向下兼容能力，但用户还是不时碰到低版本和高版本不能匹配使用的问题，以下我们从几个方面谈谈 DOS 各版本的兼容使用。

一、取消对 *DOS* 版本的检验

当外部命令所要求的版本号比所使用的 DOS 版本高时，屏幕会显示"Incorrect DOS version"而拒绝执行进一步操作。这种版本检验有时是必要的，但多数情况下却大可不必。对外部命令稍做修改，就可取消它对 DOS 版本的检查，使之能适用于任何版本的 DOS。这样，有高版本外部命令文件的用户能在低版本 DOS 下使用高版本 DOS 新增的外部命令，同时可顺利地使用任何低版本 DOS 的外部命令文件。

通用的方法是用 DEBUG 调入要修改的文件，用 S 命令寻找版本检查功能 4B 的功能号赋值地址，然后反汇编这一地址，找到"JZ xxxx"将其用 a 命令修改成"JMP xxxx"，再用 W 写回磁盘。下面是修改 DOS 3.30 的 DISKCOPY.COM 的例子。

```
A>DEBUG DISKCOPY.COM
-S100 FFFE B4 30
CS:1650
-u 1650
1184:1650 B430          MOV   AH,30
1184:1652 CD21          INT   21
1184:1654 3D031E        CMP   AX,1E03
1184:1657 740B          JZ    1664
1184:1659 BAE90F        MOV   DX,0FE9
1184:165C B409          MOV   AH,09
1184:165E CD21          INT   21
1184:1660 CD20          INT   20
1184:1662 EB03          JMP   1667
1184:1664 E80100        CALL  1668
1184:1667 C3            RET
1184:1668 51            PUSH  CX
1184:1669 C6060903FF    MOV   BYTE PTR [0309],FF
1184:166E 33C9          XOR   CX,CX
-a 1657
1184:1657               JMP   1664
1184:1659
-W
Writing 1897 bytes
-q
```

注意：修改后的"JMP xxxx"字节数不能超过"JZ xxxx"的字节数。

二、降低硬盘 *DOS* 版本

在高版本(3.00)以上分区硬盘后，硬盘的 FAT 被指定为 16 位，而低版本却不支持 16 位的 FAT。经分析试验，只要将分区信息表中说明 FAT 位数的系统指示符 04 改为

01，即可用软盘启动硬盘。

具体方法如下：

1. 把系统指示符 04 修改成 01

a 分区信息表：

```
MOV  DX,80
SUB  AX,AX
INT  13
MOV  BX,1000
MOV  CX,1
MOV  AX,0201
INT
```

b 查找引导指示符80

S 1080 1100 80

c 把引导指示符80位移04处的内容改为01

d 将主引导程序存盘，其编程与 a 读取程序相似，只是将 MOV AX,0201 改为 MOV AX,0301

以上过程在 DEBUG 中进行。

2. 用低版本启动机器

3. 逻辑格式化硬盘

进行到第 2 步后，硬盘就对低版本 DOS 开放了，这时就可以对硬盘进行操作，但此时硬盘的各种参数还是高版本所设置的，所以有必要对硬盘进行逻辑格式化。

三、换装软磁盘引导程序

1. 格式化软盘。如欲在 CCDOS 2.10 下使用 3.00 以上版本的命令，则用 3.00 以上版本的 FORMAT 命令格式化一张盘，将 CCDOS 2.10 的文件拷入这张盘，再重新用它启动机器即可，其格式化命令如下：

A>FORMAT B:/S

2. 修改引导数据。如在 DOS 2.00 版本下读写经 DOS 3.10 格式化并存有文件的软盘，但又没有相应的高版本系统盘，就必须利用 DEBUG 或 PCTOOLS 对其引导数据进行修改。

首先去掉软盘的写保护，把软盘插入 A 驱动器中，然后按如下步骤修改：

```
C>DEBUG
-l 0100 0 0 1
-D 100
0C1D:0100 E9 D5 01 49 42 4D 20
20-33 2E 31 00 02 04 01 00.....
.....
-E 108
0C1D:0108 33. 32 2E. 2E 31. 30
-w 0100 0 0 1
-q
```

斜体部分表示修改部分，L 和 W 命令后的格式必须一致，否则会破坏软盘的内容。

利用此方法，在 COMPAQ 与 PC 系列机之间互用软盘十分方便。

COMMAND.COM 的使用

COMMAND.COM 是 DOS 的基本模块之一，负责分析执行命令，并从存贮介质中加载程序到内存中运行。系统在自举时 COMMAND.COM 已被装入到内存中，但它仍可以作为一个外部的用户命令来使用，其调用格式如下：

COMMAND [d:][path][/p][/c string][/e:xx]

/p 把新命令处理器永久地复制到内存中，除非重新引导 DOS，否则无法返回原来的命令处理器；/C string 传递一个字符串给命令处理器，对它进行解释并执行，然后返回到主命令处理器，如同时调用 /P，/C 两个参数，/P 将被忽略；/E:xx 用以指定环境空间的大小，xx 以字节为单位在 DOS 3.0 和 3.1 版中，它是 16 进制数 10～62，在 DOS 3.20 以上版本中，是 10 进制数 160～32768，若未指定该参数，则隐含值为 10 进制数 160。

dBASE,FOXBASE 中的 RUN(;)，BASIC 中的 SHELL，LOTUS 1-2-3 中的 /S 等命令的运行都是调用 COMMAND.COM 来实现的。

在 DOS 3.2 以前版本中，批处理文件不允许嵌套使用，为了实现嵌套使用，就可以运用调用 COMMAND.COM 的方法，如在 BATCH1 中调用 BATCH2，执行完后再返回到 BATCH1 中继续执行，则在 BATCH1 中加入 COMMAND /C BATCH2 即可。

了解了 COMMAND.COM 的使用方法后，读者就可以在开发应用程序时加以灵活运用，值程序具有更好的完整性和封闭性。

DOS 环境变量的自动修改

DOS 环境块是由环境字符串——ASCII 字符组成，存放在 DOS 系统中称为主环境块的特定区域中。环境字符串的结构是：

＜环境变量名＞＝＜环境变量值＞

DOS 允许用户自定义，修改或删除任意的环境变量，但系统有三个标准环境变量：COMSPEC、PATH 和 PROMPT。其中变量 PATH 用于引导 DOS 到特定的目录中去依次查找具有扩展名 COM、EXE 和 BAT 的文件。

若应用程序需要在环境变量 PATH 中包括了某个指定路径的环境里运行，则一般要求用户在运行程序之前先设置这个环境变量，这既可以通过直接键入 DOS 命令实现，也可以通过预先提供一个专用批处理文件实现。但是这都要求用户输入该应用程序之外的命令，对于普通用户来说是不方便的。较好的方法是在应用程序中自动将所需要的路径名加到环境变量 PATH 中，以下的过程将能实现这一目的，该过程用 Turbo PASCAL 编写。

```
PROCEDURE Addpath(NewPath:string);
    VAR ESeg,Len,i,j,k:integer;
        PROCEDURE Fined;
        BEGIN
              Inc(i,4);
              IF Mem[ESeg:i]=Byte('=')
                    THEN BEGIN k:=i+1;j:=0;END;
```

```
            Inc(i);
        END;
BEGIN
        ESeg:= MemW[PreFixSeg: $ 2C];
        Len:= Length(NewPath);
        i:= 0;
        WHILE MemW[ESeg:i]<> $ 0 DO
            IF (Mem[ESeg:i]= $ 50) AND (Mem[ESeg:i+1]= $ 41) AND
                (Mem[ESeg:i+2]= $ 54) AND (Mem[ESeg:i+3]= $ 48)
            THEN Fined
            ELSE BEGIN Inc(i);Inc(j);END;
        Move (Mem[ESeg:k],Mem[ESeg:k+Len+1],j+2);
        FOR i:= 0 TO Len-1 DO
            Mem[ESeg:k+1]:= Byte(NewPath[i+1]);
        Mem[ESeg:k+Len]:= byte(';');
END.
```

获取 DOS 命令的入口地址

获取 DOS 命令的入口地址的目的在于对 DOS 进行分析，研究，完善和改进。

在 DOS 的命令解释器 COMMAND.COM 中有一张 DOS 内部命令入口码表，所有 DOS 内部命令均编排在这张码表中，每条命令均由四部分组成：1) DOS 命令字长(3.00 以下版本中为字长加 1)，占一字节；2) DOS 命令；3) 标志，占一字节；4) 入口地址，占两字节。

内部命令入口码表的搜寻方法如下：

```
C>DEBUG COMMAND.COM
-S0 FFFE "NOT"
4126:4CF5
-D4CF0
4126:4CF0 24 C0 11 00 03 4E 4F 54-C1 07 0A 45 52 52 4F 52   $ @...NOT...ERROR
4126:4D00 4C 45 56 45 4C 85 08 05-45 58 49 53 54 18 08 00  LEVEL...EXIST...
4126:4D10 03 44 49 52 03 91 0D 06-52 45 4E 41 4D 45 01 C0  .DIR....RENAME.@
4126:4D20 0F 03 52 45 4E 01 3C 0F-05 45 52 41 53 45 01 68  ..REN.@..ERASE.h
4126:4D30 0F 03 44 45 4C 01 E4 0F-04 54 59 50 45 01 38 10  ..DEL....TYPE...
4126:4D40 03 52 45 4D 02 04 01 04-43 4F 50 59 03 61 26 05  .REM....COPY.a&.
4126:4D50 50 41 55 53 45 02 D7 0F-04 44 41 54 45 02 C9 1D  PAUSE.[..DATE.[
4126:4D60 04 54 49 4D 45 00 CC 1E-03 56 45 52 00 2E 11 03  .TIME....VER....
```

现以 DIR 命令为例说明。由上可知，DIR 命令的入口地址为 0C91。但这还不是最终的入口地址，因为在 COMMAND.COM 中，文件的指针作了移动，在其代码常驻部分中有如下一段指令：

```
MOV   DX,XXXX
XOR   CX,CX
MOV   AX,4200
INT   21
```

其中的 XXXX 即为文件指针的移动量。要想获取其值，需用 s 命令搜索到 XOR CX,CX 的起始地址，在此地址前若干字节用 U 命令反汇编，即可得到 XXXX 的值，具体方法如下：

```
-s 100 L5AAA 33 C9 B8 00 42 CD 21
4BB2:043C
-u 430
   MOV  CH,FD
   CALL 03DD
   JMP  041C
   MOV  BX,AX
   MOV  DX,1340
   XOR  CX,CX
   MOV  AX,4200
   INT  21
```

从中可知其移动量为 1340H。

最后，用码表中的地址加上移动量，就是真正的 DOS 内部命令入口地址，对于 DIR，其实际入口地址为：0C91+1340＝1FD1。

以上的例子均针对 DOS 3.10 版而言。下面列出了几种不同 DOS 版本的有关地址，供参考。

DOS 版本号	COMMAND.COM 长度	命令码表地址	基址
PC DOS 2.00	17664(4500H)	3AAD	10C0
GW DOS 2.10	17792(4580H)	3AC0	10C0
MS DOS 2.11	17888(45E0H)	3B76	1110
PC DOS 3.10	23210(62DBH)	4CF4	1340
Olivetti 3.20	25307(62DBH)	4BE2	1440
PC DOS 3.20	23791(5CEFH)	4F14	14B0

DOS 的设备文件功能及应用

DOS 的设备文件不同于数据文件，既可以先打开后读写，也可以直接读写。系统为每一外设赋予一个文件名，同时分配给每个外设一个文件处理号，使用户可灵活有效地对这些设备进行 I／O 操作。常用外设及设备文件名和文件处理号为：

设备名	设备文件名	文件处理号
键盘	CON	0
显示器	CON	1
异步通讯器	AUX 或 COM1	3
打印机	PRN 或 LPT 1	4

对于以上设备文件，可用 DOS 系统功能请求里的文件管理功能进行读写操作，即 3DH(打开数据或设备文件)，3FH(读数据或设备文件)，40H(写数据或设备文件)。

一、从键盘读字串

1. 得用 DOS 提供的文件号进行读操作。

```
   MOV  AH,3FH
   MOV  BX,0                    ;键盘文件号送 BX
   MOV  CX,10                   ;字串长度
```

```
MOV   DX,OFFSET BUF          ;指向缓冲区
INT   21H                    ;读键盘
JC    ERR                    ;C=1，转出错提示
MOV   CHAR_JS,AX             ;保存实际输入字符数
.....
BUF   DB 13 DUP(?)
CHAR_JS DW ?
```

2. 先打开该设备，再进行读操作

```
MOV   AH,3DH
MOV   AL,0                   ;文件读方式
MOV   DX,OFFSET NAME         ;指向设备文件名
INT   21H                    ;打开设备文件
JC    ERR                    ;C=1，转出错提示
MOV   FILE_HEAD,AX           ;保存文件号
.....
MOV   AH,3FH
MOV   BX,FILE_HEAD           ;文件号送 BX
MOV   CX,10                  ;字串长度
MOV   DX,OFFSET BUG          ;DX 指向缓冲区
INT   21H                    ;读键盘
JC    ERR                    ;C=1，转出错提示
MOV   XHAR_JS,AX             ;保存实际输入字符数
.....
BUF   DB 13 DUP (?)
CHAR_JS DW ?
NAME  DB 'CON',0
FILE_HEAD DW ?
```

二、向打印机输出字符串

1. 利用 DOS 提供的文件号进行操作

```
MOV   AH,40H
MOV   BX,4                   ;打印机的文件号送 BX
MOV   CX,10                  ;字串长度
MOV   DX,OFFSET BUF          ;DX 指向缓冲区
INT   21H                    ;输出字串
JC    ERR                    ;C=1，转出错提示
MOV   CHAR_JS,AX             ;保存实际输出字符数
.....
BUF   DB 13 DUP (?)
CHAR_JS DW ?
```

2. 先打开该设备，再进行写操作

```
MOV   AH,3DH
MOV   AL,1                   ;文件写方式
MOV   DX,OFFSET FILE_NAME    ;DX 指向设备文件名
INT   21H                    ;打开设备文件
JC    ERR                    ;C=1，转出错提示
MOV   FILE_HEAD,AX           ;保存文件号
.....
MOV   AH,40H
MOV   BX,FILE HEAD           ;文件号送 BX
```

```
MOV   CX,13                    ;字串长度
MOV   DX,OFFSET BUF            ;DX 指向缓冲区
INT   21H                      ;输出字串
JC    ERR                      ;C=1，转出错提示
MOV   CHAR_JS,AX               ;保存实际输出字符数
MOV   AH,0
MOV   DX,0
MOV   AL,0DH
INT   17H                      ;打印输出数据后回车
MOV   AL,0AH
INT   17H                      ;换行
.....
BUF   DB   13  DUP (?)
CHAR_JS    DW ?
FILE_NAME  DB   'PRN',0
FILE_HEAD  DW ?
```

向异步通讯口或显示器输出字串类似于向打印机输出字串，所不同的是在不打开设备文件直接输出时，送 BX 的文件号不同，异步通讯口的文件号为 3。

DOS 重定向功能的应用

DOS 的输入输出，通常是在标准设备上进行的，但也可以把它定义为对文件进行的操作，该文件可以是磁盘文件或是设备文件，这就是输入输出重定向。

一、输出改向

在命令行中打入一个大于号“>”加上输出文件引用名或设备名即可实现输出改向，如：A>TYPE READ > PRN

将 READ 送到打印机去，而不是显示器，又如

A>DIR C:>A: HDDIR

将 C 盘的根目录送到 A 盘中的 HDDIR 文件中保存起来，而不是立即显示出来。

使用连续的两个大于号“>>”，可用来添加一个文件。如文件已存在，则输出将置于原有文件之后，而不会破坏原文件，一个大于号“>”则会重写已有的同名磁盘文件。

二、输入改向

在命令行中使用小于号“<”，可进行输入改向，如执行：

A>BASIC MAX<INFILE> OUTFILE

则表示运行 BASIC 程序 MAX 时，从 INFILE 中读取数据，而结果输出到 OUTFILE 中保存。

三、重定向到设备

重定向为使用系统中的其他设备提供了一种方便的途径，在调用程序执行的命令中，只需给出重定向命令，就可以把程序的输入输出转换到其他设备上去。如执行：

A>PROGRAM <INFILE> PRN

表示执行 PROGRAM 程序时，由 INFILE 中输入需要的数据，执行结果送到打印机输出。

有关 DOS 的重定向功能的一些具体应用，读者还可参看本节的“DEBUG 反汇编程

序的存盘与压缩"。

系统复制的简便方法

若硬盘中未带有系统文件或系统文件遭到破坏，则无法从硬盘引导系统，一般的对策是先将硬盘中的内容拷出，再将硬盘重新格式化，装入系统文件，最后装回盘中的原文件。显然这是一件极其麻烦的事。

操作系统的成功引导是由引导程序 IBMBIO.COM,IBMDOS.COM 和 COMMAND.COM 共同完成的。凡是格式化的磁盘，其 0 道已装有引导程序，所以需要的只是后三个文件，其中 IBMBIO.COM 和 IBMDOS.COM 必须放在根目录下。因此可采用以下的方法：

1. 将一系统软盘上的 IBMBIO.COM 和 IBMDOS.COM 文件的属性改为读写属性；

2. 查看硬盘根目录第一、二个文件，其长度若大于 IBMBIO.COM,IBMDOS.COM 两文件之和，则将其拷出备份，再将前两个文件名依次换为 IBMBIO.COM、IBMDOS、COM；若其长度小于 IBMBIO.COM,IBMDOS.COM，则删除下面的文件，直到大于为止；

3. 用 COPY 命令将软盘上的 IBMBIO.COM,IBMDOS.COM 和 COMMAND.COM 拷入硬盘。

至此系统复制完成，系统已能从该盘引导了，此法亦适用于软盘或磁盘上 DOS 的升级。应注意的是，引导程序的版本最好与三个系统文件版本一致。

恢复 DOS 控制的方法

用户常常会碰到这样的情况：运行某个程序时，出现了死循环，而无法退回到 DOS 状态下，只好重新引导系统。这样不仅浪费时间，有时还会造成数据的丢失。下面提供的程序就是针对这种情况提出的解决方法，即按下特定的键，就强行退出，恢复 DOS 的控制，而不需重新启动系统.当然只有在键盘中断未能禁止时，它们才起作用。

```
CODE        SEGMENT
            ASSUME CS:CODE, DS:CODE,ES:CODE,SS:CODE
            ORG    0100H
START:      JMP    INIT
_ALT        EQU    08H
_ESC        EQU    01H
HOLD        EQU    THIS DWORD
HOLDIP      DW     00H
HOLDCS      DW     00H
MSG         DB     'Please strike <Alt>+<Esc> key when DOS will be hitched !',0DH,0AH,24H
EXTENSION   LABEL  NEAR
            PUSH   ES
            PUSH   DS
            PUSH   AX
            PUSH   BX
            PUSH   CX
            PUSH   DX
```

```
        PUSH    BP
        PUSHF
        PUSH    CX
        POP     DS
        MOV     AX,0040H
        MOV     ES,AX
        IN      AL,60H
        CMP     AL,1BH          ;敲 <ESC> 键?
        JNZ     NOMAL_KEY
        MOV     BH,ES:[0017H]
        TEST    BH,08H          ;敲 <ALT> 键?
        JZ      NOMAL_KEY
        IN      AL,61H
        OR      AL,80H
        OUT     61H,AL
        AND     AL,7FH
        OUT     61H,AL
        MOV     AL,20H
        OUT     20H,AL
        MOV     AX,4CFFH
        INT     21H
NOMAL_KEY: POPF
        POP     BP
        POP     DX
        POP     CX
        POP     BX
        POP     AX
        POP     DS
        POP     ES
        JMP     CS:HOLD
INIT    LABEL   NEAR
        PUSH    CS
        POP     DS
        MOV     DX,OFFSET MSG
        MOV     AH,09
        INT     21H
        MOV     AX,3509H
        INT     21H
        MOV     HOLDIP,BX
        MOV     HOLDCS,ES
        MOV     AX,2509H
        MOV     DX,OFFSET EXTENSION
        INT     21H
        MOV     DX,OFFSET INIT
        INT     27H
CODE    ENDS
        END     START
```

扩展内存的充分利用

一、建立磁盘高速缓冲区

一般而言，缓冲区开辟得越大，主机与磁盘的信息交换速度就越快。如果在配置文件 CONFIG.SYS 中设置较大的磁盘缓冲区，就会占用较多的基本内存。PC TOOLS 4.0 以上版本提供了一个 PC-CACHE.COM 文件，可将高速缓存开辟在扩展内存中。

C>PC-CACHE／SIZE XT=64K

以上命令将在扩展内存中建立一个 64K 的高速缓存，代替了磁盘缓冲区的作用。程序在运行中基本不再读盘，为了减少对基本内存的占用，可在 CONFIG.SYS 中设置 BUFFERS=1。

二、建立"软汉字卡"

16 点阵汉字库一般要占用 240～260K 内存，如果将其放入扩展内存中，则既可提高速度又可充分利用扩展内存。CCBIOS 2.13F 提供了这样的功能，它将显示用的汉字库全部放入扩展内存中，大大节省了基本内存的开销，相当于建立了一块"软汉字卡"。对于没有 2.13F 的用户也可建立类似的"软汉字卡"，其方法请参看本章第二节中的有关介绍。

三、建立虚拟盘

DOS 3.00 以上版本中提供了 VDISK.SYS 设备驱动程序，包含在 CONFIG.SYS 文件中，在系统启动时在内存中开辟一块区域做为虚拟盘使用，其调整用格式如下：

DEVICE=[d:][path] VDISK.SYS [comment][bbb][comment][sss][comment][ddd][／E[:m]]

其中 ／E 参量通知 VDISK 将虚拟盘装在扩展内存中。多个 DEVICE=VDISK.SYS 命令，可在扩展内存中建立多个虚拟盘。m 是 VDISK 一次传送数据的最大扇区数目，可选值为 1～8，隐含值为 8。此命令仅适用于 CPU 为 80286 以上并有扩展内存的机器。

DBASE、WS、FOXBASE 等软件都带有覆盖模块，如将其装入虚拟盘中启动运行，则可明显地提高速度。

对于 80386 机器，如果使用 DOS 4.00 版中提供的设备驱动程序 XMAEM.SYS，则可将扩展内存仿真为扩充内容，使 DOS 随意访问这一部分内存，使用更为方便。

DOS 保留的中断调用与应用

MS-DOS(PC-DOS)提供了许多很强的软，硬中断，有些中断功能调用是 DOS 系统保留的，这些功能调用主要供 DOS 内部使用，用于处理 DOS 内部的低级的数据结构。

下面介绍的是 DOS 2.xx 和 DOS 3.xx 中保留的中断调用功能。版本更新时，这些功能可能会被修改，但实际上，大部分保留功能都会被保留，因此对它们的调用也有相当的可信度。

一、*INT 21H* 中保留的功能调用

INT 21H
功能 34H
获得 DOS 临界区单元地址

调用：AH=34H

返回：ES:BX＝DOS 临界区单元地址

INT 21H
功能：50H
设置新的 PSP 段地址

调用：AH＝50H
BX＝新 PSP 段地址
返回：无

```
;(50H) Set PSP
NEW_PSP DW ?
.....
        MOV AH,50H
        MOV BX,NEW_PSP
        INT 21H
.....
```

INT 21H
功能：51H
获得当前 PSP

调用：AH＝51H
返回：BX＝当前 PSP 段地址

```
;(51H) Get PSP
OLD_PSP DW ?
.....
    MOV AH,51H
    INT 21H
    MOV CS:OLD_PSP,BX
    .....
```

INT 21H
功能：52H
取 DOS 内部缓冲区指针

调用：AH＝52H
返回：ES:BX＝内部缓冲区段：地址
说明：此功能返回一个指向 DOS 内部缓冲区的指针，不同的 DOS 版本缓冲区结构有所不同。表 1 为 DOS 2.xx 缓冲区结构，表 2 为 DOS 3.xx 缓冲区结构。

```
;(52H) Get an internal MS-DOS buffer pointer
Buffer_Ptr  DD  ?
DOS_Version DB  ?
    .....
    MOV   AH,52H
    INT   21H
    MOV   WORD PTR Buffer_Ptr,BX
    MOV   WORD PTR Buffer_Ptr[2],ES
    CMP   DOS_Version,2
    JZ    Handle_DOS2
Handle_dos3:
    .....
Handle_dos2:
    .....
```

表一 DOS 2.xx 缓冲区结构

偏移(H)	大小	解 释
-2	字	通过内存分配功能调用分配的第一块可用空间的段地址值
0	双字	指向第一个 MS-DOS 磁盘参数数据块(DPB)的指针
4	双字	指向 MS-DOS 打开文件链接表指针
8	双字	指向时钟驱动程序的(CLOCK $)的指针
0C	双字	指向控制台(CON,)驱动程序的指针
10	字节	系统支持的当前逻辑驱动器号
11	字	系统支持的最大扇区尺寸
13	双字	指向逻辑磁盘使用第一个扇区缓冲区结构的指针。每个缓冲区的大小为每个缓冲区的大小加上十六个字的头。缓冲区的数量由 CONFIG.SYS 文件中的"BUFFERS=nn"语句来确定。
17		第一个设备驱动程序的开始 (NUL)

表二 DOS 3.xx 缓冲区结构

偏移(H)	大小	解 释
10	字	最大扇区尺寸
12	双字	指向逻辑盘使用的第一个扇区缓冲区的指针
16		指向驱动器路径表
1A	双字	指向第一个 FCB 表的指针，只有 CONFIG.SYS 含有"FBS=nn"命令时，此表才有效
1E	字	FCB 表的大小
20	字节	系统支持的当前驱动器号
21	字节	CONFIG.SYS 中"LASTDRIVE=nn"定义的值,缺省为 5
22		第一个设备驱动程序的开始(NUL)

二、*INT 28H* 中断

DOS 中的 INT 8H 和 INT 1CH 两个时钟中断每秒产生 18.2 次，并且可以通过软件调整，如设计一个时钟中断服务程序，希望在前台有任务时，尽量减少由于频繁进入该中断而影响前台任务，则可将时间间隔设得长一些，但是在前台没有任务时也间隔同样的时间，又将浪费主机时间，因此希望当前台无任务时，能尽快地执行时钟中断服务程序。然而在时钟中断服务程序中判断前台是否有任务是很困难的，频繁修改系统时钟也是不现实的。DOS 保留的 INT 28H 中断提供了解决的方法。

DOS 在空闲时不断产生 INT 28H 中断，如将时钟中断服务程序也作为 INT 28H 的中断服务程序，则在 DOS 空闲时便可不断进入该服务程序。

三、*INT 21H* 中的 *34H* 号功能调用和中断 *INT 28H* 的具体应用

DOS 系统提供的假脱机打印程序 PRINT.COM 利用了 INT 21H 中的 34H 功能调用和中断 INT 28H。下面是 DOS 2.1 中 PRINT.COM 的有关内容。

该程序在初始化部分建立了 INT 1CH 和 INT 28H 两个中断服务程序，并利用了 INT 21H 的 34H 功能调用，将 DOS 临界区标志的地址保存了起来.其代码如下：

```
0C54  PUSH  CS
0C55  POP   DS
0C56  MOV   DX,0557
0C59  MOV   AL,28
0C5B  MOV   AH,35
0C5D  INT   21          ;取得原 INT 28H 服务程序入口地址
0C5F  MOV   [0193],ES   ;保存该地址
0C63  MOV   [0191],BX
0C67  MOV   AL,28
0C69  MOV   AH,25
0C6B  INT   21          ;设置新的 INT 28H 服务程序入口地址为 CS:557H
      ......
0D23  MOV   AH,34
0D25  INT   21          ;取得 DOS 临界区单元地址
0D27  MOV   [0185],ES   ;保存该地址
0D2B  MOV   [0183],BX
0D2F  MOV   AL,1C
0D31  MOV   AH,35
0D33  INT   21          ;取得原 INT 1CH 服务程序入口地址
0D35  MOV   [0189],ES   ;保存该地址
0D39  MOV   [0187],BX
0D3D  MOV   DX,04FD
0D40  MOV   AL,1C
0D42  MOV   AH,25
0D44  INT   21          ;设置新的 INT 1CH 服务程序入口地址为 CS:4FDH
```

……

在时钟中断服务程序中，首先检查是否在 DOS 临界区。若是，则转原时钟中断服务程序；若不是，则先调整用执行输出打印子程序，然后才转原时钟中断服务程序，代码如下：

```
04FD CS:
04FE INC   BYTE  PTR[018D]
0502 CS:
0503 INC   BYTE  PTR[018E]
0507 CS:
0508 CMP   BYTE  PTR[018F],00  ;计数减为零了吗?
050D JZ    0516
050F CS:
0510 DEC   BYTE  PTR[018F]
0514 JMP   0552
0516 CS:
0517 CMP   BYTE  PTR[018B],00
051C JNZ   0552
051E PUSH  DS                  ;保存 DS,SI
051F PUSH  SI
0520 CS:
0521 LDS   SI,[0183]           ;使 DS:SI 指向 DOS 临界区标志
0525 CMP   BYTE PTR[SI],00     ;该标志单元是零吗?
0528 POP   SI                  ;恢复 DS,SI
0529 POP   DS
052A JNZ   0552                ;不为零，转原 INT 1C 程序
052C CS:                       ;为零，先执行本身的服务程序
052D JNC   BYTE  PTR[018B]
0531 CS:
0532 MOV   BYTE  PTR[018D],00
0537 CS:
0538 MOV   BYTE  PTR[018E],00
053D STI
053E MOV   AL,20
0540 OUT   20,AL               ;发出 EOI 命令
0542 CALL  0585
0545 CLI
0546 CS:
0547 MOV   BYTE  PTR[018F],08  ;每隔 8*1/18.2 秒中断一次
054C CS:
054D MOV   BYTE  PTR[018B],00
0552 CS:                       ;转原 INT 1CH 入口
0553 JMP   FAR[0187]
```

……

其中 0585 子程序是输出打印，在 INT 28H 中也用它，[018B]单元是标志，以防止重入。当它不为零时，表示已在中断服务程序中，再产生 INT 28H 和时钟中断都不应再进入。[018F]是计数器，当它减少为零时才能进入时钟中断服务程序，以增加时钟中断的时间间隔，减少对前台任务的影响。

在 INT 28H 中断服务程序中，不判断是否在临界区。因为此时该单元总是一。但此

时中断 DOS 相对来说比较安全。其部分代码如下：

```
0557  CS:
0558  CMP   BYTE  PTR[018B],00   ;已在中断服务程序中吗?
055D  JNZ   0580                ;是，转原 INT  28H 入口
055F  CS:                       ;否，执行相应服务
0560  INC   BYTE  PTR[018B]
0564  CS:
0565  INC   BYTE  PTR[018C]
0569  STI
056A  CALL 0585
056D  CLI
056E  CS:
056F  MOV   BYTE  PTR[018C],00
0574  CS:
056F  MOV   BYTE  PTR[018F],08
057A  CS:
057B  MOV   BYTE  PTR[018B],00
0580  CS:                       ;转原 INT 28H 入口
0581  JMP   FAR[0191]
      ......
```

INT 28H 不是硬中断，所以程序中没有发 EOI 指令。

虽然这两个中断的应用很普遍(特别是在象 3+网的进程管理程序 PRO.SYS 等系统软件中)，但高版本的 DOS 可能会对其进行部分修改，如 DOS 3.1 的 PRINT.COM 程序中就是判断一个字(CMP WORD PTR[SI-01])，而不是一个字节。

实时中断处理程序调用 DOS 内核的方法

因为受 DOS 单用户性的限制，使得用户在开发中断处理程序时，无法直接使用 DOS 的内核——系统功能调用 INT 21H 部分，仅能使用 DOS BIOS 的部分调用来进行编程。这给用户中断处理程序的编制，带来了极大的不便，增加了编程的复杂性。

由于 DOS 内核的 INT 21H 处理程序具有很好的模块性，完全可以通过对该部分的复制来重新设计新的系统功能调用。这样就可以在中断处理进程中使用复制后的系统功能调用，而在用户进程中仍然使用原来的 DOS 系统功能调用，避免了由于 DOS 内核同时被多个进程调用而发生的冲突。

附后的程序，是在 DOS 3.1 环境下，设置 INT 0AH 实时中断处理程序，其中参数 Count 0 和 Count 1 分别表示复制 INT 21H 程序的缓冲区长度和复制部分的入口地址，INT 51H 为系统保留的中断向量，用来作为复制后的系统功能调用的中断向量。在 INT 0AH 处理程序中，完成用 INT 51H 的 09 号功能来显示字符串信息。对于不同的 DOS 版本，其内核处理程序有所不同，下表给出了现有的几个 DOS 版本复制新的系统功能调用时，复制部分的入口地址参数和所用缓冲区的长度。

DOS 版本号	IBMDOS.COM 长度(字节)	缓冲区长度(字节)	复制部分入口参数
2.00	17024	4300	0BABH
2.10	17664	4300	0BABH
3.00	27920	27920	0004H
3.10	27760	27760	0004H
3.20	28477	28477	0004H
3.30	30159	30159	0004H

上述方法对 DOS 内核的复制，并不是所有的 INT 21H 调用都可在中断处理程序中使用，象内存管理，EXEC 等系统功能调用由于涉及到内存的定位与分配，都不能在中断处理程序中使用，否则将会出错。另外，同样中断处理程序也是不可重入的，在设计时必须采取相应的加锁控制，防止多重中断的发生。

```
COUNTO   EQU   27760                    TINT:   PUSH  DS
COUNT1   EQU   04H                              PUSH  ES
CSEG     SEGMENT                                SUB   AX,AX
         ASSUME ES:CSEG,DS:CSEG                 MOV   DS.AX
DIR_P    DB    "GOOD!",                         MOV   BX,86H
               07,0DH,0AH,24H                   MOV   AX,DS:[BX]
TINT51   DB    COUNTO   DUP(?)                  MOV   DS.AX
BEGIN    PROC  FAR                              MOV   BX,COUNT1
         PUSH  DS                               MOV   DX,DS:[BX]
         XOR   AX,AX                            MOV   AX,DS:[BX+2]
         PUSH  AX                               MOV   DS.AX
         MOV   AX,CSEG                          MOV   SI,0
         MOV   DS,AX                            MOV   AX,ES
         MOV   ES,AX                            MOV   BX,OFFSET TINT51
         JMP   TINT                             MOV   CL,4
10A      PROC  NEAR                             SHR   BX,CL
         PUSH  AX                               INC   BX
         PUSH  BX                               ADD   AX,BX
         PUSH  CX                               MOV   ES,AX
         PUSH  DX                               MOV   DI,0
         PUSH  DS                               MOV   CX,COUNT0
         PUSH  ES                               REP   MOVSB
         MOV   AX,CS                            PUSH  ES
         MOV   DS,AX                            POP   DS
         MOV   AH,09                            MOV   AL,51H
         MOV   DX,OFFSET DIR_P                  MOV   AH,25H
         INT   51                               INT   21H
         POP   ES                               POP   ES
         POP   DS                               POP   DS
         POP   DX                               MOV   DX,OFFSET 10A
         POP   CX                               MOV   AL,0AH
         POP   BX                               MOV   AH,25H
         POP   AX                               INT   21H
         IRET                                   IN    AL,21H
10A      ENDP                                   AND   AL,0FBH
```