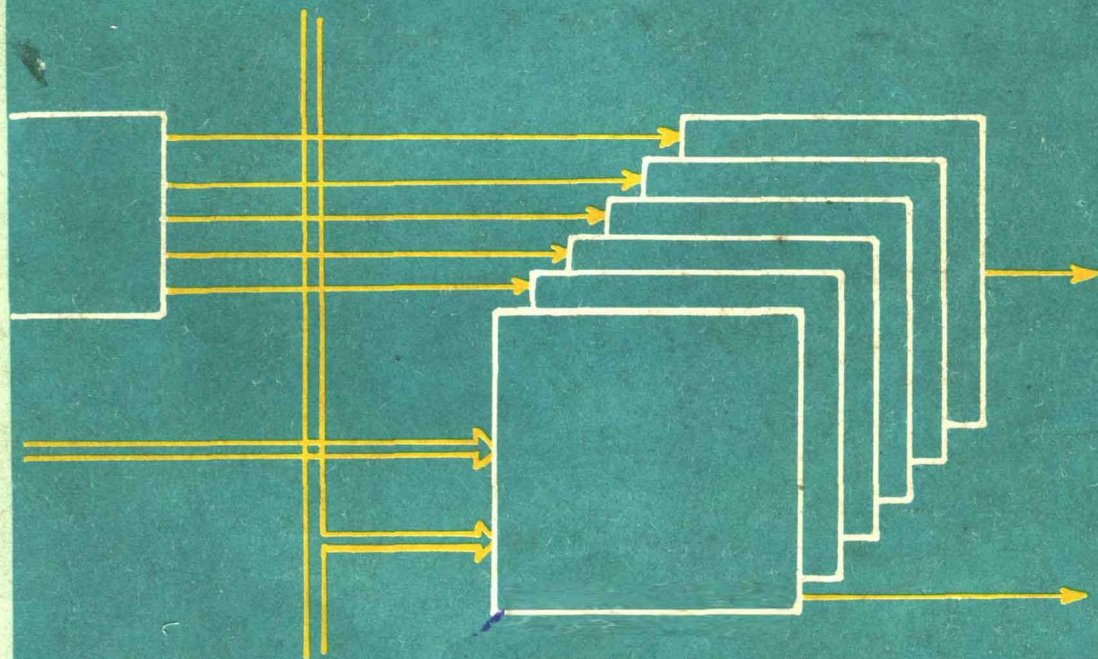


IBM—PC、PC/XT、PC/AT、长城

# DOS、BIOS功能调用手册

戴水贵 编著



陕西电子杂志社

IBM—PC. PC/XT. PC/AT.

长城 DOS. BIOS

# 功能调用手册

戴水贵 编著

# 前 言

系统功能调用是DOS为用户提供的—组常用子程序，汇编语言程序除了可以以软中断（10H—1FH）的方式调用BIOS中的子程序外，还可以通过软中断（DOS的INT21H已经标准化，可以在任何DOS系统上使用，即可以在任何支持DOS的计算机上运行）直接调用DOS所提供的有关输入输出操作、目录操作、文件操作等70多种功能，使编制程序极为方便有效。

功能调用的一般过程为：把调用号放入寄存器AH中，设置入口参数，然后执行软中断INT N，最后分析出口参数。

由于调用的功能繁多，汇编程序员很难记住每个功能调用的入口参数设置和返回参数的状态。这给汇编程序员带来不便。为此，我们收集整理了这本手册。

手册分三部分，第一部分为DOS功能调用，第二部分为BIOS、CCBIOS和长城0520机功能调用，第三部分为各类功能调用的程序实例。

为了便于查阅，手册第一部分将20H—2FH软中断按功能进行了分类。第二部分按软中断号进行了分类。第三部分则按程序的不同应用目的进行了分类（见目录）。用户在进行程序设计时，很快就能找到所需要的功能入口和相应的程序实例。

为了便于初学者，手册中的功能调用绝大部分给出了具体使用的例子，尤其是手册的第三部分提供了各种类型的程序实例。这对掌握每个功能调用的正确使用是极为有帮助的，这使得本手册具有教科书的性质，是汇编语言高级程序设计员的一本极有参考价值的书。

手册中提供的汇编语言程序实例，都是在PC/XT或长城0520CH机上调试通过的完整程序。这些程序中的一些程序是极有实用价值的，其中的一些曾在国内计算机刊物上发表过。

由于80286/80386提供的调用入口是向下兼容的，即长城和PC机上运行的调用入口也可以在80286或80386机上运行。故本书提供的调用入口也可在80286和80386机上运行。

本手册由孙毅君副教授主审，在编写过程中他提了许多宝贵意见，在此特予致谢。由于时间仓促和编者水平有限，手册难免有许多错误和不当之处，请用户批评指教。

编 者

1989年5月

# 目 录

## 第一部分 DOS功能调用

程序结束	( 3 )
字符输入	( 5 )
字符输出	( 9 )
盘控制	( 12 )
文件操作	( 20 )
记录操作	( 43 )
目录操作	( 55 )
盘传送区地址	( 58 )
系统日期和时间	( 59 )
动态存储分配	( 61 )
网络功能	( 65 )
其它系统功能	( 73 )

## 第二部分 BIOS功能调用

显示驱动部分 (INT 10H)	( 82 )
软盘服务 (INT 13H)	( 96 )
串行口服务 (INT 14H)	( 100 )
键盘服务 (INT 16H)	( 104 )
打印机控制服务 (INT 17H)	( 105 )
EMS扩展的内存规则服务 (INT 67H)	( 107 )
长城0520机GWBIOS 3.00功能调用	( 118 )
CCBIOS功能调用	( 143 )
CGE400高分辨率图形显示卡BIOS功能调用	( 148 )

## 第三部分 应用程序实例

一、各类常用编程技巧程序例	( 152 )
二、磁盘文件的创建与读写程序例	( 164 )
三、利用键盘绘图程序例	( 173 )
四、屏幕图形的存贮与调用程序例	( 181 )
五、字符放大程序例	( 195 )
六、DOS的EXEC功能调用程序例	( 201 )
七、磁盘文件加、解密程序例	( 208 )
八、PC/XT机之间的通讯程序例	( 213 )
九、中断截取和驻留程序例	( 217 )
十、图形和文本打印程序例	( 226 )

## 如何在汇编语言中使用功能调用

由于汇编语言兼有机器语言和高级语言的一些主要优点,因此在软件开发中十分有用。对于从事计算机软件开发工作者来说,掌握汇编语言程序设计将使你进入更高层的程序设计境界。尤其是DOS、BIOS为用户提供了功能丰富的调用入口和子程序,使得汇编语言更加锦上添花,使用也变得更加方便。

在汇编语言中使用DOS功能调用的一般格式为:

```
mov ah, 功能号
    :
    :
int 21H
```

即置AH寄存器为功能号,其它的寄存器为指定的调用参数,随后执行INT 21H。由于DOS的int 21H已经标准化,并且可以在任何DOS系统上使用,所以由这些功能完成所有I/O的程序,将可在任何支持DOS的机器上运行。

ROM BIOS显示驱动子程序的一般调用格式为:

```
mov ah, 功能号
    :
    :
int 10H
```

即置AH寄存器为功能号,其它的寄存器为指定的调用参数,随后执行int 10H。其余功能调用的调用格式如上类似。从以上看出,实现功能调用是比较简单的。这样,很短的程序将完成功能复杂的操作。

指令助记符	转移条件	指令含义	比较类型					
JA/JNBE JAE/JNB JB/JNAE JBE/JNA	CF=0且ZF=0 CF=0或ZF=1 CF=1且ZF=0 CF=1或ZF=1	高于转移(A>B) 高于等于转移(A>=B) 低于转移(A<B) 低于等于转移(A<=B)	两无符号比较					
JG/JNLE JGE/JNL JL/JNGE JLE/JNG	ZF=0且SF=0F SF=0F或ZF=1 SF≠0F且ZF=0 ZF=1或SF≠0F	大于转移(A>B) 大于等于转移(A>=B) 小于转移(A<B) 小于等于转移(A<=B)	两有符号数比较					
JC JNC	CF=1 CF=0	进位、借位转移 无进位、借位转移	进、借位					
JE/JZ JNE/JNZ	ZF=1 ZF=0	等于0转移 不等于0转移	等于、不等于					
JO JNO	OF=1 OF=0	溢出转移 不溢出转移	溢出、不溢出					
JS JNS	SF=1 SF=0	符号位为1转移 符号位为0转移	符号位					
JP/JPE JNP/JPO	PF=1 PF=0	偶校验转移 奇校验转移	奇、偶					
JCXZ	CX=0	CX为零转移	CX为零					
-RF命令可改变标志的设置								
设置	OV是	DN↓	EI允许	NG为负	ZR是	AC是	PE偶	CY是
清除	NV否	UP↑	DI屏蔽	PL为正	NZ否	NA否	P0奇	NC否
符号	OF	DF	IF	SF	ZF	AF	PF	CF
	中断	方向	中断	符号	零	辅进位	奇偶位	进位



## 程 序 结 束

---

int 20H (32)

程序结束

**调用:** CS = 程序段前缀 (PSP) 的段地址 (e×e文件不能使用)。

**例子:** 完成退出。

int 20H, 转到DOS。

---

int 21H (33)。

**功能:** 00H (0)。

程序终止

**调用:** AH = 00

CS = 程序段前缀的段地址

**返回:** 无

**例子:** 终止退出。

mov ah, 0; 功能号

int 21H, 转到DOS。

---

int 21H

**功能:** 31H (49)

结束并驻留 (保持进程)

**调用:** AH = 31H

AL = 返回码

DX = 保留的内存大小 (节数)。

**返回:** 无

**例子:** 带返回码 1 结束, 但驻留内存, 保留从程序段前缀开始的16K内存。

mov AH, 31H; 功能号。

mov AL, 1; 返回码。

mov dx, 0400H; 保留长度为节数。

int 21H, 转DOS。

---

int 21H

**功能:** 4CH (76)

带返回码结束。

**调用:** AH = 4CH。

AL = 返回码。

返回：无

例子：完成最后退出，传送一返回码1。

mov AH, 4CH; 功能号。

mov AL, 1; 返回码。

int 21H; 转DOS。

---

int 27H (39)。

结束并驻留内存（终止当前程序的执行，但保留部分或全部它的内存，使得它不被另一装入的程序覆盖）。

调用：DX = 要保护程序的最后一字节加1的位移（包括程序段前缀）。

CS = 程序段前缀段。

返回：无。

例子：退出并驻留，保留足够大的内存以保护程序代码和数据。

start,

∴  
∴

mov dx, offset pgm—end; 保存字节DX = 结束驻留

int 27H; 转DOS。

∴  
∴

pgm—end equ 'Y'

end start

注释：该中断一般允许用户写驱动程序或中断处理的COM或EXE程序，使它们驻留内存。顺序进入该码的是一硬件或软件中断。

\* ① 版本中用该中断能保留内存的最大数目是64K字节。因此，int 27H调用只应用在DOS 1.0以下程序。②、③版本中用前述的int 21H功能31H，它允许你保留任意大的内存。

\* 该中断不应由装入内存高端的EXE程序调用（以/hight开关连接）。因为这部分由command.com的暂有部分使用。如果command.com不能重装入，则系统将失败。

\* 由于执行该中断的结果改变结束（int 22H），ctrl—c（int 23H）和严重错误（int 24H）向量，它不能用于永久安装用户写的严重错误处理程序。执行int 27H时，打开的文件不自动关闭。

\* 当DX包含从0ff1H到0ffffH的值时，该中断不能正确工作。此时，DOS舍掉DX的高位，结果实际保留在内存中的是32K字节，少于程序的请求。



## 字 符 输 入

---

int 21H (33)

功能 01H (1)

带回显的字符输入(无字符输入时将等待输入)。

调用: AH = 01

返回: AL = 8位数据(输入)

例子: 从键盘上读一字符到寄存器AL, 显示在屏幕上, 并存入变量input-char中。

```
mov ah, 1; 功能号。
```

```
int 21H; 转DOS。
```

```
mov input-char, AL; 保存字符。
```

input-char db 0

注释: \*如果读的字符是ctrl-c, 则执行int 23H。

\*为了在pc和它的兼容机上读扩充的ASCII码(如移动功能键F1到F10), 需要调用两次功能。头一次返回0, 表明是扩充码。

---

int 21H (33)。

功能: 03H (03)。

辅助输入(从第一串行口读一个字符)。

调用: AH = 03

返回: AL = 8位数据(输入)

例子: 从标准辅助设备读一字符到寄存器AL, 并存入变量input-char中。

```
mov ah, 3; 功能号。
```

```
int 21H; 转DOS。
```

```
mov input-char, al; 保存字符。
```

```
⋮
```

```
input-char db 0
```

注释: 若DOS检测到ctrl-c或ctrl-break, 则执行int 23H。

---

int 21H (33)

功能: 06H (6)。

直接控制台I/O(不通过操作系统接口, 读写所有可能的字符和控制码)。

调用: AH = 06

DL = 请求功能

00h—0feh输出请求（d1输出字符）。

0ffh—输入请求。

**返回：**对输出请求——无

对输入请求并且一字符准备好；

零标志 = 清除

AL = 8位数据（输入）。

对输入请求并且无字符准备好；

零标志 = 置

**例子：**送字符 \* 到标准输出设备。

mov ah, 6; 功能号。

mov al, '\*'; 字符输出。

int 21H; 转DOS。

**例子：**从标准输入设备读一字符。若无字符则等待。

wait;

mov ah, 6; 功能号。

mov al, 0ffh; 读参数。

int 21H; 转DOS。

jz wait; 等待直到准备好。

**注释：**\* 如果检测到ctrl-c或ctrl-break，使用该功能调用时将不产生任何影响。

\* 为在pc和其兼容机上读扩充的ASCII（如特殊功能键F1—F10），需要调用该功能两次。头一次调用返回0表示为扩展代码。

int 21H (33)

**功能：**07H (7)

无面显、无过滤的字符输入（若无字符输入则等待）。

**调用：**AH = 07

**返回：**AL = 8位数据（输入）。

**例子：**从键盘无面显地读一个字符到寄存器AL，并且存到变量input-char中。

mov ah, 7; 功能号。

int 21H; 转DOS。

mov input-char, al; 保存字符。

⋮

⋮

input-char db 0

**注释：**\* 在调用此功能时，如果检测到ctrl-c或ctrl-break将不产生影响。

\* 为读扩充的ASCII码（例如特殊功能键F1—F10），要两次调用该功能。头一次调用返回0，表明是扩充代码。

int 21H (33)

功能: 08H (8)

无面显的字符输入 (如果没有字符准备好则等待)。

调用: AH = 08

返回: AL = 8位数据 (输入)。

例子: 从标准输入设备读一字符到寄存器AL, 允许检测ctrl-c或ctrl-break, 并存字符在变量input-char中。

```
mov ah, 8; 功能号。
```

```
int 21H; 转DOS。
```

```
mov input-char, al; 保存字符。
```

```
⋮
```

```
⋮
```

```
input-char db 0
```

注释: \* 如果检测到ctrl-c或ctrl-break则执行int 23H。

\* 为读扩充的ASCII (例如特殊功能键F1到F10), 要两次调用该功能。头一次调用返回0, 表明是扩充代码。

int 21H (33)

功能: 0AH (10)。

缓冲输入 (从键盘读一行并放入用户定义的缓冲)。

调用: AH = 0AH。

DS: DX = 缓冲区段; 位移。

返回: 无

例子: 从标准输入设备读最大长度为80个字符的字符串 (加上回车), 并放入my-buffer中。

```
mov ah, 0ah; 功能号。
```

```
mov dx, seg my-buffer; 输入缓冲区地址。
```

```
mov ds, dx
```

```
mov dx, offset, my-buffer
```

```
int 21H; 转DOS。
```

```
my-buffer db 81H; 最大串长度包括CR。
```

```
db 0; 实际长度。
```

```
db 81H dup (0); 串缓冲区。
```

注释: \* 缓冲区的头一个字节说明它能保存的最大字符数 (1至255), 该值由用户设置。第二个字节由DOS返回实际读入的字符数 (回车除外)。

\* 扩充ASCII码 (如功能键), 在缓冲区存储为两个字节, 头一个字节为0。

- \* 该调用有缓存超前打字的能力，同时可以执行全部标准键盘编辑命令
- \* 如果检测到ctrl-e和ctrl-break，则执行int 23H。

int 21H ( 33 )

功能: 0BH ( 11 )

取输入状态 ( 检查是否可以从键盘取一字符 )。

调用: AH = 0BH

返回: AL = 00若无字符可用。

= 0FFH若字符可用。

例子: 检测是否可以从标准输入设备得一字符。

mov AH, 0BH, 功能号。

int 21H, 转DOS。

or al, al, 字符等待吗?

jnz char-ready, 字符准备好则转?

⋮

⋮

char-ready:

注释: \* 如果一个输入字符正在等待, 该功能不断返回一真标志, 直到有功能调用01H, 06H, 07H, 08H或0AH的程序实际地输入。

\* 如果检测到ctrl-c或ctrl-break, 则执行int 23H。

int 21H ( 33 )

功能: 0CH ( 12 )

重置输入缓冲区后输入 ( 清超前打字缓冲区, 接着调用键盘输入功能 )。

调用: AH = 0CH

AL = 重置缓冲区后调用的功能号必须是01H, 06H, 07H, 08H或0AH。

对功能0AH,

DS, DX = 输入缓冲区段, 位移。

返回: 对功能0AH——无。

对功能01H, 06H, 07H或08H,

AL = 8位数据 ( 输入 )。

例子: 清超前打字缓冲, 接着等一字符进入, 回显并返回到AL寄存器。

把字符存到变量input-char中。

mov ah, 0ch, 功能号。

mov al, 1, 子功能调用 = 字符输入。

int 21H, 转DOS。

mov input-char, al, 保存字符。

input—char db 0

**注释：** \* 该功能调用的目的是丢掉在DOS中超前打字已输入的字符，使程序等待它所需要的输入字符（通常从键盘）。

---

## 字符输出

---

int 21H ( 33 )

**功能：** 01H ( 2 )

字符输出（输出一字符到当前活动的显示器上）。

**调用：** AH = 02H

DL = 8位数据（通常ASCII代码）。

**返回：** 无。

**例子：** 传送字符 \* 到标准输出设备。

mov ah, 2; 功能号。

mov dl, '\*' ; 输出字符。

int 21H; 转DOS。

**注释：** \* 当显示时，一退格码（08H）使光标左移一个位置，并留在此位。

\* 请求输出后，如果有ctrl—break，则执行int 23H。

---

int 21H。

**功能：** 04H ( 4 )

辅助输出。

**调用：** AH = 04H

DL = 8位数据（输出）。

**返回：** 无

**例子：** 输出字符 \* 到辅助设备。

mov ah, 4; 功能号。

mov dl, '\*' ; 输出字符。

int 21H; 转DOS。

**注释：** \* 如果输出设备正忙，该功能调用等待，直到设备准备接收一个字符。

\* 用该功能无法得到辅助输出设备的状态。在pc中可调用 BIOS中的int 14H，或直接驱动通讯控制器完成。

\*如果DOS检测到ctrl-c或ctrl-break, 则执行int 23H。

---

int 21H (33)

功能: 05H (5)

打印输出 (送一字符到标准排队设备 (PRN或LPT 1))

调用: AH = 05

DL = 8位数据 (输出)。

返回: 无。

例子: 送\*字符到排队设备。

mov ah, 5; 功能号。

mov dl, '\*'; 输出字符。

int 21H转DOS。

注释: \*如果打印机正忙, 该功能将等待, 直到打印机准备接收该字符。

\*如果DOS检测到ctrl-c或ctrl-break, 则执行int 23H。

---

int 21H (33)

功能: 06H (6)

直接控制台I/O (不通过操作系统接口, 读写所有可能的字符和控制码)。

调用: AH = 06

DL = 请求功能。

00H-0FEH输出请求 (DL = 输出字符)。

0FFH请求输入。

返回: 对输出请求——无。

对输入请求, 并且一字符准备好;

零标志 = 清除

AL = 8位数据 (输入)。

对输入请求, 并且无字符准备好;

零标志 = 置

例子: 送字符\*到标准输出设备。

mov ah, 6; 功能号。

mov dl, '\*'; 字符输出。

int 21h; 转DOS。

从标准输入设备读一字符, 若无字符则等待。

wait;

mov ah, 6; 功能号

mov dl, 0ffh; 读参数。

int 21h; 转DOS。

jz wait; 等待直到准备好。

**注释:** \* 如果检测到ctrl-c或ctrl-break,使用该功能调用时,将不产生任何影响。  
\* 为在pc和其兼容机上读扩充的ASCII码(如该功能键F1到F10),需调用该功能两次。头一次调用返回0,表示为扩展代码。

---

int 21H (33)

**功能:** 09H (9)

输出字符串(送一字符串到标准输出设备)。

**调用:** AH = 09

DS, DX = 段: 串位移。

**返回:** 无

**例子:** 送字符串hello here和回车换行到标准输出设备。

```
cr equ 0dh
```

```
if equ 0ah
```

```
⋮
```

```
mov ah, 9; 功能号。
```

```
mov dx seg my string; 串地址。
```

```
mov ds, dx
```

```
mov dx, offset my string
```

```
int 21H, 转DOS
```

```
⋮
```

```
mystring db 'hello here', cr, lf, '⋆'
```

**注释:** \* 字符串必须以'⋆'字符(24H)结尾。其它的ASCII码,如控制码,不能出现在该字符串中。

---

int 2fh (47)

假脱打印机控制(提供一文件供假脱机打印,从假脱打印队列中删除或得到打印机的状态)。

**调用:** AL = 00H取安装状态。

01H送文件打印。

02H从打印文件中删除文件。

03H终止队列中的所有文件。

04H为读状态挂起打印任务。



05H为读状态结束挂起。

DS: DX = 对功能01H为包地址段段: 位移。

= 对功能02H为ASCIIZ文件说明: 位移。

**返回: 成功:**

进位标志 = 清

对功能00H:

AL = 状态

00H如果未安装, 则现在可以安装。

01H如果未安装, 则现在不可安装。

0FFH已安装。

对功能04H:

DX = 错误计数。

DS: SI = 打印队列的段: 位移。

**失败:**

进位标志 = 置

AX = 错误码

1 功能无效。

2 文件未找到。

3 路径未找到。

4 打开的路径太多。

5 拒绝存取。

8 队列满。

9 假脱机忙。

0 ch名字太长。

0 fh驱动程序无效。

**注释:** \* 对功能01H, 信息包有5个字节。头一字节包含级, 后四个字节包含ASCII Z文件说明(文件名不能包含任意字符)的双字地址(段和位移)。如果文件存在, 它被加到打印队列尾部。

\* 对功能02H, 在文件说明中允许任意字符(\*和?), 使得可用一个调用从打印队中删除多余文件。

\* 对功能04H, 打印队的返回地址指向一系列的文件名项, 在队列中每一项为64字节长, 并包含文件说明的ASCIIZ串。队中的头一文件说明是当前正被打印的文件。队列中的最后一项首字为null(0)。

---

## 盘 控 制

---

int 21H(33)

**功能:** 0DH (13)

重启动磁盘。

**调用:** AH = 0DH

**返回:** 无

**例子:** 刷新DOS的全部盘缓冲区。

```
mov ah, 0dh
```

```
int 21H
```

**注释:** \* ①版本缺省时选驱动器A, 置盘传送区(DTA)地址为DS: 0080H, 刷新磁盘文件缓冲区,

\* ②、③版本刷新全部的文件缓冲区, 例如, 程序要求写全部的数据, 但它们仍然在DOS缓冲区中, 则将把它们写到盘上。

\* 对任一打开的文件, 该功能不能修改磁盘上的目录。如果在退出盘之前, 程序未能关闭所有的文件, 同时文件改变了大小, 则由于目录项不正确, 存到磁盘上的数据将不可存取。

int 21H (33)

功能0EH (14)

置缺省盘驱动器(选择一说明的驱动器, 为当前或缺省的盘驱动器, 返回系统中逻辑驱动器总数)。

**调用:** AH = 0EH

DL = 驱动器码(0 = A, 1 = B, 等)。

**返回:** AL = 系统中逻辑驱动器数。

**例子:** 使逻辑驱动器B为当前驱动器, 存逻辑驱动器数在变量drives中。

```
mov ah, 0eh; 功能号。
```

```
mov dl, 1; 1 = b
```

```
int 21H; 转DOS。
```

```
mov drives, al; 存驱动器数。
```

```
⋮
```

```
⋮
```

```
drives db 0
```

**注释:** \* 在单驱动器pc兼容机中, AL返回2, 这是因为即可作A驱动器又可作B驱动器存取, 因此, 系统中有两个逻辑驱动器, 对IBM-PC, 想知道系统中实际的物理软盘驱动器数, 可调用BIOS中的int 11h

int 21H (33)

功能19H (25)

取缺省盘驱动器(返回当前或缺省盘驱动器的驱动器代码)。