

# MICROSOFT MOUSE

## 鼠标器程序员参考手册

许志平  
吴 双 编译

本书介绍了鼠标器程序设计的方法和技巧。同时提供了大量的各种语言的实例，可以作为开发鼠标器驱动程序的模板。

本书还附加了样本软盘，包括了书中引用的所有源程序，能为读者节省大量时间。



中国科学院希望高级电脑技术公司



HOPE

Microsoft

# 鼠标器程序员参考手册

许志平

编译

吴双

- 鼠标器的发展历史
- 鼠标器菜单语言及菜单设计
- 鼠标器程序设计接口及实例
- EGA卡的鼠标器接口
- 鼠标器与BASIC, C  
FORTRAN, PASCAL和MASM语言的接口

北京希望电脑公司

一九九二年三月

1.3.2

## 译 者 的 话

本书是由Microsoft公司出版的。它不仅介绍了鼠标器的发展史，也详细介绍了鼠标器程序设计的方法和技巧。同时，本书提供了大量的、各种语言的实例，可以作为开发鼠标器驱动程序的模板。本书还附加了样本磁盘，包括了书中引用的所有源程序，能为读者节省大量时间。

本书分为四个部分。第一部分介绍了Microsoft公司进入鼠标器领域的历史。第二部分介绍了鼠标器菜单程序，其中详细介绍了菜单语言的概念，语言结构及程序设计技巧，同时给出了多种多样的实例。掌握这种语言可以为原来不是以鼠标器为基础的应用程序加上鼠标器接口。第三部分介绍了鼠标器程序设计接口，这是本书的核心。其中包括如何把鼠标器加入你的已写好程序中，鼠标器驱动程序提供的各种功能及其使用方法。你可以从大量的实例和解释中学到很多方法。这里既提供了库函数MOUSE.LIB接口，也提供了中断33H接口；既提供了汇编语言接口，也提供了Microsoft的QuickBASIC，解释BASIC，C，QuickC，FORTRAN和Pascal语言接口。同时对于EGA卡，还提供了寄存器接口函数库。第四部分包括几个有用的附录供读者参考。

国内目前流行多种鼠标器，其中绝大部分都是与Microsoft鼠标器兼容的（尤其是功能0到功能20）。如果你使用的不是Microsoft鼠标器，本书提供的一般性的鼠标程序设计知识也是非常有用的，当然，读者还要详细研读所使用的鼠标器提供的资料。

本书还附加了两张磁盘，上面有多种语言编写的源程序。这些程序将是开发鼠标器程序的基础和范例。如需要此软件的朋友，请自带盘到希望公司资料部拷贝，拷贝费每张20元。

译 者

1990.1.2

7551710801

# 目 录

## 第一部分 概 论

### 第一章 鼠标器的发展

- 1.1 早期的鼠标器..... ( 1 )
- 1.2 Microsoft 鼠标器..... ( 1 )
- 1.3 展望..... ( 3 )

### 第二章 鼠标器程序设计概述

- 2.1 鼠标器驱动程序..... ( 4 )
- 2.2 鼠标器菜单..... ( 5 )
- 2.3 鼠标器程序设计接口..... ( 6 )

## 第二部分 鼠标器菜单

### 第三章 构造你自己的鼠标器菜单

- 3.1 鼠标器菜单语言的命令..... ( 8 )
- 3.2 语句格式..... ( 8 )
- 3.3 菜单程序结构..... ( 11 )
- 3.4 编写一个菜单程序..... ( 19 )

### 第四章 鼠标器菜单语言中的语句

- 4.1 语句的语法约定..... ( 21 )
- 4.2 ASSIGN 语句..... ( 21 )
- 4.3 BEGIN语句..... ( 22 )
- 4.4 EXECUTE语句..... ( 23 )
- 4.5 MATCH 语句..... ( 24 )
- 4.6 MENU...MEND 语句..... ( 26 )
- 4.7 NOTHING语句..... ( 27 )
- 4.8 OPTION语句..... ( 28 )
- 4.9 POPUP...PEND 语句..... ( 29 )
- 4.10 SELECT语句..... ( 31 )

4.11	TEXT语句	( 31 )
4.12	TYPE 语句	( 32 )

### 第五章 鼠标器菜单程序的实例

5.1	菜单程序 SIMPLE	( 34 )
5.2	菜单程序 DOSOVRLY	( 34 )
5.3	其它菜单程序的实例	( 36 )

## 第三部分 鼠标器的程序设计接口

### 第六章 鼠标器的程序设计接口

6.1	鼠标器驱动程序	( 39 )
6.2	显示卡和显示器	( 40 )
6.3	虚拟屏幕	( 43 )
6.4	图形光标和文本光标	( 45 )
6.5	内部光标标记	( 49 )
6.6	读入鼠标器状态	( 49 )

### 第七章 鼠标器程序设计的注意事项

7.1	设置你的系统	( 50 )
7.2	高级课题	( 52 )
7.3	鼠标器功能	( 54 )

### 第八章 鼠标器功能调用

8.1	鼠标器功能简介	( 56 )
8.2	功能0 鼠标器重置和取状态	( 58 )
8.3	功能1 显示光标	( 63 )
8.4	功能2 隐藏光标	( 64 )
8.5	功能3 取按钮状态和鼠标器位置	( 65 )
8.6	功能4 设置光标位置	( 68 )
8.7	功能5 取按钮压下状态	( 69 )
8.8	功能6 取按钮放开状态	( 71 )
8.9	功能7 设置水平位置极限	( 73 )
8.10	功能8 设置垂直位置极限	( 74 )
8.11	功能9 设置图形光标	( 75 )
8.12	功能10 设置文本光标	( 81 )
8.13	功能11 取鼠标器移动的计数值	( 83 )
8.14	功能12 设中断程序掩码和地址	( 84 )

8.15	功能13	打开光笔模拟	( 91 )
8.16	功能14	关闭光笔模拟	( 92 )
8.17	功能15	设置mickey与象素的比	( 93 )
8.18	功能16	条件关闭	( 94 )
8.19	功能19	设置速度加倍的下限	( 96 )
8.20	功能20	交换中断程序	( 98 )
8.21	功能21	取驱动程序存贮要求	( 102 )
8.22	功能22	保存驱动程序状态	( 103 )
8.23	功能23	恢复驱动程序状态	( 105 )
8.24	功能24	设辅助程序掩码和地址	( 106 )
8.25	功能25	取用户程序地址	( 111 )
8.26	功能26	设置分辨率	( 112 )
8.27	功能27	取分辨率	( 114 )
8.28	功能28	设置中断速度	( 115 )
8.29	功能29	设置 CRT页号	( 116 )
8.30	功能30	取CRT 页号	( 117 )
8.31	功能31	关闭驱动程序	( 118 )
8.32	功能32	打开驱动程序	( 120 )
8.33	功能33	软件重置	( 121 )
8.34	功能34	选择语言	( 123 )
8.35	功能35	取语言编号	( 124 )
8.36	功能36	取版本号及鼠标器类型	( 125 )

## 第九章 鼠标器接口程序设计实例

9.1	解释BASIC 程序	( 129 )
9.2	QuickBASIC 程序	( 133 )
9.3	C和QuickC 程序	( 149 )
9.4	MASM 程序	( 157 )
9.5	FORTRAN 程序	( 165 )
9.6	PASCAL 程序	( 168 )

## 第十章 在IBM EGA下编写鼠标器程序

10.1	EGA 寄存器接口函数库	( 177 )
10.2	使用EGA 寄存器接口函数库的限制	( 183 )
10.3	EGA 寄存器接口功能	( 184 )

## 附录

附录A:	鼠标器命令行开关	( 195 )
------	----------	---------

附录B: 鼠标器驱动程序提示信息.....	( 197 )
附录C: 鼠标器菜单提示信息.....	( 199 )
附录D: 把MOUSE·LIB 连接到鼠标器程序中.....	( 201 )
附录E: 在TURBO PASCAL程序中调用鼠标器功能.....	( 201 )
附录F: 在Hercules 卡上使用鼠标器.....	( 202 )
附录G: ASCII字符集.....	( 203 )

# 第一部分 概 论

## 第一章 鼠标器的发展

鼠标器是一个控制计算机屏幕上光标移动的小型手控设备。它的第一次问世是在25年前。开始时，它是一个粗劣的，带有一个按钮的木制品，不过到目前为止，它已经发展成为一个复杂而精巧的输入设备，对今天的计算机用户来说，它的重要性不亚于键盘。

在鼠标器的25年发展历史中，Microsoft公司在鼠标器的发展中所起的作用是非常重要的。Microsoft鼠标器从1983年问世以来立刻就成为人们与计算机通讯的新标准。虽然鼠标器不是Microsoft公司发明的，但该公司在改进鼠标器上做了大量工作。为了了解这一点，先让我们回顾一下鼠标器最初设计的情况。

### 1.1 早期的鼠标器

鼠标器的发明者Doug Engelbart曾经说过，我们所用的许多设备在鼠标器出现之后就都显得相形见绌了。除非有更好的替代物出现，鼠标器将会一直是计算机用户最好的定点设备。

Doug Engelbart于1963年在斯坦福研究所制造出第一个木制鼠标器。他的思想极大地影响了很多计算机设计师，之后25年中所发生的一切甚至连Doug Engelbart本人也未曾预料到。

最初的鼠标器只是一个简单的模拟设备，它把鼠标器的移动状态通过传送信号的方式送给控制软件，再由这个软件移动屏幕上光标的位置。在鼠标器的木制结构中有两个金属轮，它们通过轴连接在两个可变电阻上。

70年代早期，Xerox公司建立了第一个数字化鼠标器，这提高了鼠标器的知名度。这种新的鼠标器本质上是原始的模拟鼠标器的数字化版本。当时，Xerox公司为它制造的Alto计算机配备了鼠标器。尽管这种计算机的销售量不过百台，并没有取得市场上的成功，但它却为个人计算机和鼠标器的进一步发展铺平了道路。

Xerox公司在1975年宣布了鼠标器的规范，这个规范被很多制造商所采用，并且持续到80年代。

### 1.2 Microsoft鼠标器

当Xerox鼠标器受到越来越广泛的注意时，Microsoft公司开始考虑设计一种鼠标器。



Xerox 公司的雇员加入Microsoft公司，想为该公司的新软件Microsoft Word 提供鼠标器支持。几乎在同时，Microsoft 也在考虑生产这种硬件。

### 1.2.1 从一块粘土开始

80年代初期，Microsoft还是个小公司。在很大程度上该公司都依赖于西雅图的一位图形设计师 David Strong (他设计了Microsoft 公司的商标和颜色)。因此，在鼠标器设计上Microsoft 公司也要得到Strong的帮助。

Microsoft 公司说明了他们所需要的设备应该是容易抓握的小型设备，刚好能装下必要的内部电路。这时，Strong 就开始了设计工作。最后的结果是得到了一个  $2.5 \times 4 \times 1.25$  英寸的粘土模型，下面装了三个图钉来模拟滚轮。

### 1.2.2 第一代产品

1983年6月，Microsoft 公司为IBM PC机配置了一个新的产品——总线型鼠标器。这是一个双按钮的机械式鼠标器，它有两个转轮和一个钢球，用它们来记录鼠标器滑动时的状态。鼠标器的电源来自一块装有Intel 8255芯片的插件板上。这种鼠标器的主要优点是，它的机械编码器只消耗很少的能量。

在总线型鼠标器推出一年后，Microsoft 设计了串行口鼠标器。这是技术上的巨大进步，这种鼠标器可以直接连在RS-232串行口上。它既不需要总线型插件板，也不需要独立的电源系统，鼠标器中的CMOS处理器可以从RS-232口中得到足够的动力。

第一代鼠标器（总线型和串行口型）都带有一个独立的控制软件（鼠标器驱动程序）和一个用来与高级语言程序相连接的函数库（Mouse. LIB）。为了帮助用户更方便地使用鼠标器，Microsoft公司同时还提供了三个程序：

- Notepad, 一个基于鼠标器的正文编辑程序
- Piano, 一个用鼠标器操纵的屏幕钢琴键盘
- Life, 一个模拟微生物成长和死亡的游戏程序

后来的鼠标器软件版本提供了功能更强的Notepad程序，还提供了名为Doodle的绘图程序，而且进一步提供了鼠标器菜单。使用鼠标器菜单，用户可以在原来不是为鼠标器设计的软件中使用鼠标器。使用visiCalc, Multiplan, Wordstar和Lotus 1-2-3的用户可以通过一个特殊菜单用鼠标器控制程序运行。此外，Microsoft公司提供了MENU.COM程序来装入菜单，MAKEMENU.EXE编译程序帮助用户生成自己的菜单。

1983年，MS-DOS 2.0开始发行，鼠标器利用了新版MS-DOS的可安装设备驱动程序的特点，可以很方便地配置在使用MS-DOS的计算机上。

1985年，鼠标器的软件版本升级到3.0和4.0，这个版本主要是为IBM PC/AT和高分辨率显示设备开发的。用户可以把鼠标器软件安装在大多数显示器上，包括Hercules卡，CGA卡，EGA卡 and 很多新型的高分辨率显示设备上。鼠标器驱动程序可以在安装时自动检测

硬件设备。

1985年5月，在4.0版的鼠标器软件中，Doodle程序被换成了最新式的图形软件PC paintbrush。

### 1.2.3 第二代产品

1985年10月，鼠标器已经进化到一个新的阶段，它已经相当复杂，而且可以根据使用人员的意愿进行重新配置。很多改变都是可见的，灰色的按钮，重新设计的外形，原来的钢球外面覆盖上了一层橡胶。但是更重要的改变不是所看见的，而是被感受的，它的分辨率提高到原来的两倍（每英寸200点），这样鼠标器就更容易使用，不必在桌上到处推动。

这种新型鼠标器只需要方圆4至5英寸的桌面就可以使用。大多数操作都可以利用手腕和手的动作来完成。而早期的鼠标器则相反，它需要大约8至10英寸的使用面积。

1986年，Microsoft 公司把总线型鼠标器接口替换成用特制的Inport 芯片驱动，这大大地改善了鼠标器的性能，因为驱动程序可以利用该芯片的可编程中断速率。

鼠标器硬件性能的改变使使了新版本软件的推出。1986年9月，Microsoft 公司推出6.0版鼠标器软件，它具有下列特性：

- PC paint brush 升级并改名为Microsoft Paintbrush
- 增加了鼠标器的配置程序，还增加了图形软件Show Partner（后者在6.1版本中已被取消）
- 专业化的鼠标器菜单可以为Lotus 1-2-3, Display Write III和Multimate 3.31服务
- 提供了一个教程软件（6.1版中取消）
- 提供了一个鼠标器控制台，用户可以调整鼠标器的灵敏度，以适应不同的应用程序。

除此之外，这个版本提供了带有九种语言提示信息的国际版本，软件设计师可以很容易地提供多种语言支持。

### 1.2.4 第三代产品

第三代鼠标器是1987年9月问世的，它有一个小巧精制的外型和容易使用的按钮，按钮在按下时会发生轻微的喀嗒声。

这一代鼠标器的内部结构基本上与上一代的鼠标器相同，主要的改变是把滚动球移到了鼠标器的前面，把左键改得比右键更大，这是为了使鼠标器更容易控制。

这一次鼠标器的软件也有了进一步的改进，它可以支持多种总线型和串行口型硬件。总线型版本仍然使用自己的插件板，串行口型版本可以直接连在计算机的串行口上或IBM PS/2机器的鼠标器接口上。

## 1.3 展望

1988年夏，在Doug Engelbart 设计他的木制模型25年以后，Microsoft 公司庆祝它销售的第一百万个鼠标器。

今天，带有图形接口的应用软件大大超出了人们的预料，鼠标器的使用也随着这种接口

的广泛采用而得到扩大。随着 OS/2, Presentation Manage, Microsoft Windows 和其它一些图形接口软件的使用, 鼠标器已经开始从奢侈品走向了必需品的行列。

## 第二章 鼠标器程序设计概述

鼠标器是一个向计算机传送信号的设备。对你的软件来说, 这些信号代表着光标的移动和按钮的状态。不过这种原始的数据很难被利用, 而且不同的鼠标器(总线型, Inport型, 串行口型和 PS/2型)产生的信号各不相同。为了使程序员能得到一致的, 容易使用的接口, Microsoft 公司提供了鼠标器驱动程序。

### 2.1 鼠标器驱动程序

鼠标器驱动程序是这样—个软件, 它解释鼠标器传送的数据并给操作系统提供统一的接口。Microsoft 鼠标器驱动程序为此提供了35个功能调用, 使程序可以了解当前的状态, 并采取特定行动, 例如, 检查鼠标器按钮的状态。这些功能调用是与你使用的硬件无关的。

Microsoft 公司提供了三种与鼠标器驱动程序打交道的方法: 鼠标器菜单, 鼠标器函数库和直接调用MS-DOS的中断33H。每种方法都有其优点和缺点, 并分别用于不同的目的。例如, 你可以把鼠标器菜单用于已有的应用程序。在编写你自己的程序时, 可以用鼠标器函数库和中断33H。

#### 2.1.1 使用鼠标器菜单

鼠标器菜单是用来给那些原来不是为鼠标器设计的处理正文的软件加入鼠标器功能的, 你可以给该应用程序配上一个菜单, 也可以模拟键盘输入。你也可以把鼠标器的移动和按钮的状态对应于原来的键盘操作。

#### 2.1.2 使用鼠标器函数库

鼠标器函数库是用来为你自己编写的应用程序配备鼠标器功能的。由于鼠标器功能已经成为你自己程序的一部分, 这时, 鼠标器提供的能力要比在鼠标器菜单中更为丰富。函数库可以允许在高级语言中使用35个鼠标器函数, 这些高级语言包括: 解释 BASIC, QUICK BASIC, C, QUICKC, FORTRAN和PASCAL。在汇编程序 MASM中也可以使用这些函数。

#### 2.1.3 使用MS—DOS中断33H

你可以通过MS-DOS的中断33H直接访问鼠标器驱动程序, 这时你使用的35个功能与鼠标器函数库所提供的完全相同。由于函数调用的过程被省略了, 所以直接使用中断33H的程序会比使用函数库的同样程序更小更快。很多专业化的程序都是直接使用中断33H的。实际上, 只要你的程序设计语言允许利用MS-DOS中断, 就应该使用这种方法与鼠标器驱动

顺序通讯。

## 2.2 鼠标器菜单

鼠标器菜单在屏幕上显示一组你可以选择的菜单。选择其中一项后，可以对当前的应用程序模拟键盘输入，也可以执行其它菜单命令。

使用鼠标器菜单同应用程序通讯时所提供的功能仅仅是移动鼠标器和按动鼠标器按钮，把一串字符送入键盘缓冲区。

键盘缓冲区是内存中专为键盘输入字符开辟的一小块区域。应用程序从这个缓冲区中读入字符，把它作为输入并根据它的意义执行不同操作。当你移动鼠标器或按动其中的某些按钮时，鼠标器菜单程序就把一些字符送入键盘缓冲区，好象它们是从键盘上输入的一样。

鼠标器菜单软件填写键盘缓冲区要比你直接在键盘上输入快得多。键盘输入缓冲区的速度取决于每个计算机BIOS规定的限制，但鼠标器菜单软件则没有这个限制。因此，用鼠标器模拟方向键时，光标的移动速度要比用键盘操作快得多。

某些软件不是从缓冲区读入数据，而是直接从键盘读入，这时，鼠标器菜单程序就不能正常工作了。此外，该程序也不能模拟某些组合键，如Ctrl—Alt—Del。这些键的名字将在第四章介绍TYPE语句时给出。

### 2.2.1 键盘位图

鼠标器菜单程序可以识别七种鼠标器的动作，它们是：

- 按下左钮
- 按下右钮
- 同时按下两钮
- 向右移动
- 向左移动
- 向上移动
- 向下移动

你可以把这些动作对应到一个或多个菜单命令上。例如，约定俗成的鼠标器动作与键盘缓冲区的对应关系为：

- 鼠标器的四个移动方向对应于四个方向键
- 按下鼠标器的一个按钮相当于按〈Enter〉键或〈Esc〉键
- 按下鼠标器的一个按钮可以让菜单程序显示一个预先设置好的菜单，用它来向应用程序或MS-DOS输入命令

下面我们给出一个简单的鼠标器菜单程序清单，它提供了一些基本的对应关系：

```
BEGIN lb,rb,bb,lm,rm,um,dm,48,48
```

```
lb: EXECUTE f1 : Left button emulates F1 key
rb: EXECUTE entkey : Right button emulates Enter key
bb: EXECUTE escape : Both buttons emulate Esc key
```

lm;	EXECUTE left	: Left movement emulates left-arrow key
rm;	EXECUTE right	: Right movement emulates right-arrow key
um;	EXECUTE up	: up movement emulates up-arrow key
dm;	EXECUTE down	: Down movement emulates down-arrow key
f1;	TYPE 0.59	: These commands perform the
entkey;	TYPE enter	: actual work when you move
escape;	TYPE 27	: the mouse or press one or
left;	TYPE 0.75	: both mouse buttons. Refer
right;	TYPE 0.77	: to Chapter 4 for detailed
up;	TYPE 0.72	: explanations of each of
down;	TYPE 0.80	: these commands.

### 2.2.2 建立鼠标器菜单

菜单程序语言包括一组命令，可以用来设计弹出式菜单。你既可以设计简单的只有一个功能的菜单，也可以设计出多级菜单系统，其中某一个菜单项可以调用另一个菜单。

下面我们给出建立鼠标器菜单的基本步骤：

1. 设计和编写源程序
2. 编译源文件
3. 运行鼠标器菜单程序
4. 调试这个程序

关于建立鼠标器菜单的详细说明请参看第三章。

### 2.3 鼠标器程序设计接口

对于原来不是为鼠标器设计的现有应用程序，可以用鼠标器菜单提供一些支持。不过，对应用程序提供鼠标器支持的更有效方法是把鼠标器功能编写到你的源程序中。这时鼠标器就成为一个独立的输入设备，而不只是一个键盘的模拟者了。鼠标器提供的最重要的功能是一个自由移动的光标，很多通用软件，如 Microsoft Word, Microsoft Works, AutoCAD, Microsoft Paintbrush和Microsoft Windows都利用了这一特点。这样，程序就变得非常直观，友好，容易掌握。

作为鼠标器硬件和应用软件之间的媒介，鼠标器驱动程序记录了鼠标的移动和按钮的状态，当应用程序需要了解鼠标器状态时，它就向驱动程序发出请求，后者把必要的信息返回给应用程序。

#### 2.3.1 鼠标器功能

驱动程序可以执行35个输入输出操作。每个操作（或者叫功能）都是通过给驱动程序发出一个特殊指令，使应用程序和鼠标器硬件进行通讯。有些函数是向鼠标器查询状态的，如按钮状态，相对光标位置和相对移动。另一些功能控制鼠标器接口的特性，如光标的分辨率，光标形状和光标移动范围。应用程序只是把所需功能告诉驱动程序，而由驱动程序做具

体的工作。

### 2.3.2 与鼠标器驱动程序通讯

在一个程序中，有两种方法同鼠标器驱动程序通讯：一是使用MOUSE.LIB函数库，这是使用程序设计语言调用驱动程序的方法，二是用MS-DOS中断33H。这两种方法都能使用所有的鼠标器功能。虽然它们各有优缺点，但本质上是一致的。

注意，鼠标器驱动程序只控制鼠标器本身，至于设置显示方式这些操作要通过其它手段来完成。

#### 2.3.2.1 使用MOUSE.LIB函数库

你可以使用软件包中提供的MOUSE.LIB函数库为Microsoft高级语言服务。这些库函数可以把鼠标器功能变为PASCAL语言中的过程，QUICK BASIC和FORTRAN语言中的子程序，或者C和QUICK C中的函数。库函数所使用的参数是与所使用的语言一致的。所以对鼠标器进行程序设计并不需要特殊的技巧。调用鼠标器功能就像调用其它子程序一章。

要使用鼠标器函数库，在程序设计语言中必须遵循Microsoft库函数的约定。如果一种语言支持这种约定，你就可以使用函数库。关于把鼠标器连接到应用程序的细节请参看第九章。

你也可以查对一下你所使用的语言的说明书，看看它是如何与外部函数库连接的。如果该语言不支持Microsoft函数库的约定，你就无法使用MOUSE.LIB，不过你可以使用下面所介绍的中断33H方法。

#### 2.3.2.2 使用中断33H

当MS-DOS启动时，一般使用Autoexec.Bat或Config.sys文件装入鼠标器驱动程序。该程序把自己作为操作系统的一部分，并把入口地址作为中断向量设置到中断33H上。这样你就能用软中断33H访问鼠标器驱动程序。当你的软件执行这个中断时，DOS会在中断向量表中查到入口地址，使程序进入鼠标器驱动程序并执行相应的操作。

当应用程序使用鼠标器功能时，鼠标器驱动程序(MOUSE.COM或者MOUSE.SYS)必须已经在内存中。

你可以把寄存器AX, BX, CX和DX设置成相应的值来执行不同的功能，有时寄存器ES, SI和DI也将被使用。驱动程序的返回值也是通过这些寄存器返回的。有关变量传送时寄存器使用的详细信息请参看第八章。

中断33H优于函数库的主要原因是执行速度更快。这是因为直接调用中断会省去函数调用过程中的操作。中断33H的另外一个优点是当无法和函数库相连接时，仍然能够通过寄存器调用鼠标器功能。

#### 2.3.2.3 EGA寄存器接口

尽管鼠标器驱动程序可以和EGA/VGA同时工作，但是当程序员直接操作EGA/VGA

硬件时，鼠标器驱动程序必须记录EGA/VGA寄存器的状态的变化，这时要通过某些特殊方法通知驱动程序有关信息，尤其在EGA/VGA的图形方式下。

关于EGA寄存器级接口的详细信息请参看第十章。

## 第二部分 鼠标器菜单

### 第三章 构造你自己的鼠标器菜单

菜单程序设计语言是为了本来不支持鼠标器的应用程序提供鼠标器支持而设计的。菜单是通过一组命令把键盘缓冲区作为界面同应用程序通讯的。本章描述如何使用这些命令来设计和执行你自己的菜单。

#### 3.1 鼠标器菜单语言的命令

菜单语言包括十三个命令。你可以用这些命令给鼠标器指定功能，建立菜单或模拟键盘输入。

下面我们列出菜单语言的所有命令：

ASSIGN	给鼠标器指定动作或改变分辨率
BEGIN	设置鼠标器的初始动作和指定初始分辨率
EXECUTE	执行标号所指定的命令
MATCH	如果屏幕指定位置出现了所要求的字符串，就执行指定操作
MENU	开始菜单子程序
MEND	结束菜单子程序
NOTHING	空操作
OPTION	当选择菜单的指定项目时，执行OPTION所规定的动作
POPUP	启动弹出式子程序
PEND	结束弹出式子程序
SELECT	当选定弹出式菜单的项目时，执行指定操作
TEXT	定义弹出式菜单的标题或项目
TYPE	规定放入缓冲区中的字符串

#### 3.2 语句格式

在菜单程序设计语言中，语句既可以是大写的，也可以是小写的。很多语句具有下列格式：

[ 标号： ] 命令 [ 参数；注解 ]

语句BEGIN和处于弹出式子程序中的语句可以不使用这种格式，因为它们不需要标号。下面分别介绍语句的各个部分。

### 3.2.1 标号

标号是每个菜单语句的名字，除了处于弹出式子程序或菜单中的语句外，所有的语句标都应该有标号。程序实际上是通过语句标号来访问指定语句的，当这个语句执行完后，控制权就返回到调用它的语句中。换句话说，控制权不会转移到下一条语句中。在下面的例子中，mat1是MATCH语句的标号：

```
mat1; MATCH 23, , inverse, 'Format', execl, exec2
```

当你使用一个标号时，请注意以下几点：

- 标号要以字母开头，后面紧跟一个冒号。
- 在冒号和后面的命令之间至少要有一个空格。
- 不能用命令名或保留字（backsp, enter, esc或tab）做标号。
- 标号可以使用除冒号外的任何可打印ASCII字符。
- 尽量用有意义的名字做标号。如menu1表示程序中的第一个菜单。

### 3.2.2 参数

参数是影响语句动作的变量。当你使用一个语句时，必须把参数变成你所希望的值。除了NOTHING, MEND和PEND以外，所有的语句都有参数。

在语句中，参数紧跟在命令之后。语句的参数与命令之间要有一个空格，参数和参数之间要用逗号分开。

语句EXECUTE和TYPE允许有1至15个参数，而其它语句的参数数目则是固定的。当你使用这些语句时，也可能不希望为某个特定参数指定值，这时应该在该参数的位置上什么也不写。实用程序MAKEMENU会自动地用缺省值填充这个位置。

例如，下面的语句中，MATCH命令需要六个参数，但这里只给出五个。MAKEMENU会把第一个逗号和第二个逗号之间的位置补成缺省值：

```
mat1; MATCH 23, , inverse, 'Format', execl, exec2
```

菜单程序设计语言的参数类型分为三种：数字型，字符串型和属性型。

#### 3.2.2.1 数字型参数

数字型参数用来表示数字信息，如屏幕坐标或鼠标器移动的分辩率。

上面例子中的23表示MATCH命令要在第23列执行，它属于数字型参数。

#### 3.2.2.2 字符串型参数

大多数字符串型参数用来显示菜单或提示信息。一个字符串型参数可以包括数字，字母，特殊字符或空格。

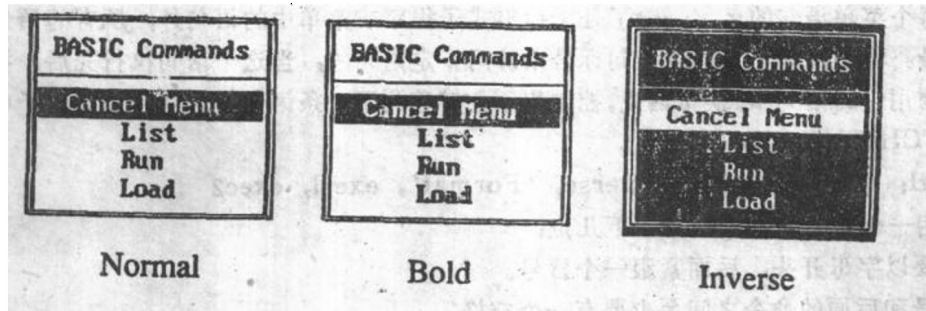
字符串必须放在双引号之内。在字符串内部不能再使用双引号。

#### 3.2.2.3 属性型参数

属性型参数决定显示器的属性，它刻划了菜单或提示信息的外观。这类参数可以有四类



值, normal, bold, inverse, 或数字值(后者在彩色显示器上标明前景和背景颜色)。下图表示normal, bold和inverse是如何影响显示结果的。



如果你没有说明属性型参数, 就使用缺省属性。第四章中会介绍每个语句中的缺省属性。

### 3.2.2.4 彩色菜单

如果你使用彩色显示器和CGA卡, 可以在MATCH, MENU或POPUP等语句中使用属性型参数, 用来刻划菜单和提示信息的前景和背景颜色。正文以前景颜色显示, 其它部分则以背景颜色显示。

下面我们给出前景和背景颜色及其对应的数值的列表。特殊的颜色是由不同的前景和背景值组合而成的。假如你要在蓝色背景下显示绿字, 绿色前景的值为2, 蓝色背景的值16, 那么所要的属性值为18,

颜色	前景	背景
黑	0	0
蓝	1	16
绿	2	32
青	3	48
红	4	64
品红	5	80
棕	6	96
白	7	112
灰	8	128
浅蓝	9	144
浅绿	10	160
浅青	11	176
浅红	12	192
浅品红	13	208
黄	14	224
高亮白	15	240