

鼠标器程序设计指南

BUILDING A BETTER MOUSE TRAP

A Programmer's Guide to the mouse

余雪丽 杨宇宁 段立维 译

电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

- 配有实例源代码的3.5"软盘
 - 全局事件管理和处理程序
 - 新的创建鼠标器光标的子画面驱动程序
 - 在DOS应用程序中管理Windows光标形状的源代码
- 用Borland和Microsoft编译的源代码
- 适用于DOS和Windows环境的鼠标器编程序



383099.

Building a Better Mouse
Trap: A Programmer's
Guide to the Mouse

鼠标器程序设计指南

[美] Jeffrey S. Donovan 著
余雪丽 杨宇宁 段立维 译

電子工業出版社

Publishing House of Electronics Industry

(京)新登字 055 号

133/2
内 容 提 要

本书以循序渐进的方式介绍鼠标器编程的基本技术和 C 语言源程序代码,使读者对鼠标器功能有高水平的理解,并且能够把鼠标函数与自己的应用程序成功地结合起来。该书分为四部分:第一部分阐述事件处理,菜单管理,CAD 系统和光标形状管理等鼠标编程的基本内容及一些实际的编程技术;第二部分使读者学会创建自己的具有新特点的图形鼠标器光标,利用 Sprite 动画,光标机构以及在 800×600 的 16 位彩色图形方式下用超 VGA 编程的方法;第三部分介绍在 Windows 环境下编程与鼠标器进行通信;第四部分附录提供了 52 种 MS-DOS 鼠标函数的文档文件和详细参考文献。

本书可以做为计算机专业高年级学生的教材或教学参考书,以及广大科技人员的技术参考书。

Copyright © 1992 by McGraw-Hill, Inc.

本书英文版由 McGraw-Hill, Inc. 出版,版权为 McGraw-Hill, Inc. 公司所有,中文版 1994 年 10 月经 McGraw-Hill, Inc. 授权予电子工业出版社独家出版。未经出版者书面许可,不得以任何形式或任何手段复制或抄袭本书内容。

Building a Better Mouse Trap:
A Programmer's Guide to the Mouse

鼠标器程序设计指南

[美] Jeffrey S. Donovan 著

余雪丽 杨宇宁 段立维 译

责任编辑:应月燕

*

电子工业出版社出版
北京市海淀区万寿路 173 信箱(100036)
电子工业出版社发行 各地新华书店经销
北京科技大学 印刷厂印刷

开本:787×1092 毫米 1/16 印张:28.75 字数:736 千字

1995 年 6 月第一版 1995 年 6 月北京第一次印刷

印数:6000 册 定价:55.00 元

ISBN 7-5053-3120-5/TP·1104

著作权合同登记号 图字:01-1995-151

前 言

对于《鼠标器程序设计指南》一书,用一句话概括是“太令人高兴了”。“Jeff Donovan 已经成功地把相当枯燥(有时是令人气馁的)鼠标器接口编程任务转换成有趣的鼠标器及其软件驱动程序的开发。使得这本书如此珍奇,不仅因为它概括了鼠标器接口编程的基本内容,而且还增加了一些提示和技术,这些提示和技术甚至对于熟练的程序员都将感到是新鲜的、很有用的。

在一个很愉快的偶然机会中我接触了这本书。当我在 Osborne/McGraw-Hill 的编辑 Frances Stack 考虑出版此书时,她问我是否能大概地看看这本书,并告诉她我的印象。我开始是浏览,继而完整地阅读,我热心地建议出版这本书。因为它写得很好,并且为有关鼠标器的某些重要性日益渐增的题目提供了很多必需的参考资料。进而,我对该书的材料非常感兴趣,以致于自愿成为该项目的技术编辑。由于我的时间安排太紧,只能做很少的编辑工作,但这是不能拒绝的一件事。因此我相信,你将会发现这本书的材料既准确又有趣味性。

《鼠标器程序设计指南》以传统的方式描述在 DOS 环境中的鼠标器接口的基本内容开始本书的章节,然后介绍如何编制用户定义的鼠标器接口库程序。许多其它书仅仅到此为止,Jeff 却完整地讨论了在编写 Windows 程序时如何与鼠标器进行通信。由于对鼠标器的支持是建立在 Windows 环境下的,你可能会认为对于 DOS 环境下的编程没有什么值得讲述的,但这种想法是错误的。Jeff 说明如何避免某些缺陷,并对于在 Windows 环境下编程时遇到的这些典型的鼠标器麻烦问题提出了创造性解决办法。

这似乎成了陈词滥调,但这本书的确是既为初学者,又为有经验的程序员所写。尽管该书包含了对于基本内容的完整讨论,它却超越了基本内容。简单地讲,你将会发现,这本书提供的信息在别的地方找不到。所以,如果鼠标器对于你和你的编程工作是重要的,这本书将是你所需要的。

Herbert Schildt
Mahomet, IL
1992、12、1

原 序

在当今的计算机行业中,鼠标器是一种最常使用的键盘输入设备,并且越来越多地被集成到新的计算机应用程序中。作为一个程序员,我常常发现自己由于缺乏鼠标器编程的原始材料而感到沮丧。写作本书的目的是提供一种对鼠标器的高水平理解,使读者能够把鼠标器的函数与自己的应用程序成功地结合起来。无论是新手还是熟练的程序员,本书中的信息、循序渐进的介绍及源程序代码都将帮助读者编制有效的、有价值的鼠标器程序。

鼠标器的历史简介

我们使用的计算机点入式设备鼠标器,是1963年由斯坦福研究所的 Douglas Engelbart 发明的。Engelbart 是一位计算机幻想家,如果没有他,个人计算机、GUIs、虚拟现实(virtual reality),以及许多今天我们已经了解或者将来将要了解的技术可能都不会存在。他设计的第一个鼠标器用于 ARC(Augmentation Research Center)计算机上。它基本上是一个底部带有两个金属轮子,顶部有一个单按键的木制匣子作成的模拟装置。

在七十年代初期,Xerox 公司的 Palo Alto 研究中心委托 Jack S. Hawley 为他们新的 Alto 计算机设计一个数字型鼠标器。当 Hawley 结束了在 Xerox 的工作之后,他在加州的 Berkeley 开了一个鼠标器车间,继续设计和制造鼠标器。

今天存在的若干种鼠标器设计及鼠标器标准,都在一定程度上受到 Engelbart 和 Hawley 的影响。

Microsoft 公司在1983年推出了他们的第一代总线鼠标器。一年以后,串行鼠标器被推出。从那以后,Microsoft 公司已经连续生产了几代新型鼠标器,并不断改进设计,使设计更合理,更功能化。随着1984年增强型图形适配器(EGA)的开发和1987年视频图型阵列(VGA)的出现,鼠标器被广泛使用,并发展成为新型的高性能图形应用产品。1988年夏季之前,Microsoft公司已销售了100万只鼠标器。且不说键盘,鼠标器已经变为最常用的输入设备。

关于这本书

通过这本书将能以最直接的方式学习到鼠标器编程的技术。所有的代码程序是简单扼要的,不必定义目标和复杂的结构(第18章除外)。

本书由四部分组成。每部分都围绕特定的中心教读者进行鼠标器编程,下面是本书的概要。

第一部分 本部分教读者掌握成功地与鼠标器进行联络所必需的所有技术,该部分包括第1章到第16章。当你读完这些章节以后,将能学到鼠标器编程的所有基本内容,并能掌握实际编程的一些技术,例如事件处理、菜单管理、CAD系统、光标形状管理等。你可以认为自己是一个羽翼丰满的鼠标器程序员。除此而外,在第16章中,还可以把先前章节中编制的程序结合

产生一个完整的鼠标器函数库,以便应用于你自己的应用程序。

第二部分 本部分教读者如何创建自己的图形鼠标器光标,这种光标具有许多新的令人兴奋的特点,这部分由第 17 章到第 21 章组成。读者可以学到子画面动画、光标机构,以及与 Super VGA(特别是在 800×600 16 位彩色图形方式下)有关的编程问题。读完这些章节以后,读者会真正感到自己似乎了解了图形鼠标器光标的内部工作情形,并且将拥有一个全新的高级鼠标器函数库,以便供将来使用。该函数库允许使用缺省的图形鼠标光标,也允许使用自己的子画面/鼠标器光标。这种光标可以工作于 EGA 640×200 16 位彩色,EGA 640×350 16 位彩色和单显方式,也可工作于 VGA 640×480 16 位彩色和彩显方式,以及 Super VGA 800×600 16 位彩色图形方式下。

第三部分 本部分教读者正确地在 Windows 环境下与鼠标器联络,该部分由第 22 章到 26 章组成,将详细讲述和开发适当的 Windows 鼠标器技术、规则和指南。

第四部分 该部分为附录,提供了所有 50 种 MS-DOS 的鼠标器函数的详细参考文献,所有 Windows 鼠标器和光标函数,以及所有与 Windows 鼠标器有关的消息。

有关 DOS 章节的概述

如果你首先打算在 DOS 应用程序中与鼠标器联络,作者建议从第 1 章开始连续读下去。你将通过后续章节逐步学习鼠标器编程,一点也不觉得意外。无论你是鼠标器程序员生手、中等程度者、或者熟练者,下面章节的内容都不应忽略,也许会有特殊的兴趣。

注:如果要查询与鼠标器有关的题目可参考目录表。

第 1 章 第 1 章介绍鼠标器驱动程序、中断服务程序、寄存器以及怎样通过中断 0x33 与鼠标器通信。此外,还讨论了 Borland 和 Microsoft 编译程序的不同之处,并且为处理这些不同创建了一个名为 COMPILER.H 文件。

第 2 章 在本章,读者可以创建视频函数库,该库可以在所有的 DOS 编程实例中使用(第 20 章的例子除外)。这个函数库是在本书中唯一分别建立 Borland C 和 Microsoft C 文件的代码程序。

第 3 章 该章介绍创建鼠标器初始化函数库。读者至少应当浏览一下,以便熟悉这些函数,并了解它们是如何执行的。

第 7 章和第 8 章 阅读和理解第 7 章、第 8 章是最基本的要求。在第 7 章,读者可以创建事件管理和事件处理程序;在第 8 章,学习如何使用它们。第 7 章以后的所有编程例子都使用理件管理和事件处理程序,读者必须理解二者的基本机制。同时,还有一些在使用事件管理和处理程序时必须遵守的特殊规则和指南。

第 12 章 第 12 章介绍那些不属于必需的、作为可选的鼠标器函数。尽管读者可以不使用这些函数,同样能控制鼠标器,但是使用这些函数将在你的应用中增加专业化色彩。除了第 12 章,第 13、14 和 15 章的程序实例都是基于这些可选函数的。

第 16 章 把所有的鼠标器函数组合成一个文件,并编译产生一个实用库(.lib),同时该库做一个主要的头文件,这个头文件成为在读者应用程序中完全可以再次使用的源程序文件。

第 17 章 本章作为第 18、19、20 和 21 章的介绍。读者将学习关于 Super VGA 编写鼠标

器程序的问题(特别是在 800×600 16 位彩色 VGA/Super VGA 图形方式下)。为何 800×600 16 位彩色方式应当被看作是标准方式,而来自各种视频卡和鼠标器制造商生产的 Super VGA 方式却不是标准方式。然后,在第 18 章,读者将开始作一个通用的方案,建造一个子画面驱动程序,用于产生自己的图形鼠标器光标。

关于 Windows 章节的概述

如果你是一个 Windows 编程生手,或者对 Windows 3.1 不熟悉,建议首先阅读并理解第 22 章。在第 22 章中,读者可以创建一个通用的 Windows 程序,并学习如何使该程序与 Windows 相互作用。

如果读者是一个中等程度或有经验的 Windows 程序员,并熟悉 Windows 3.1 的标准,可以从任何希望的部分开始阅读。

注:如果要查询与鼠标器有关的题目可参考目录表。

OEM 鼠标器

读者按本书指导创建的 DOS 鼠标器函数库是商品化的源程序,如果不是 Microsoft 的鼠标器,必须当成一般的设计来考虑。

当编写商品化应用程序时,必须要保持与所有的 OEM 鼠标器的兼容性,以保证该应用程序在任何环境下都能适当地运行。尽管大多数 OEM 鼠标器制造商声称他们的鼠标器与 Microsoft 的鼠标器 100% 的兼容,他们并没有提供 Microsoft 的高级鼠标器函数。这些高级函数是在 Microsoft 的鼠标器驱动程序 6.26 版本中推出的。从那以后,Microsoft 公司已经连续在新的版本中推出更多高级函数,到目前为止把它们范围由鼠标器函数 37 扩大到鼠标函数 52。据我所知,OEM 制造商执行全部高级鼠标器函数的仅有 Logitech,使用版本 6.1 及更新版本的鼠标器驱动程序,Krafft 使用版本 8.2 及更新版本的鼠标器驱动程序。

因此,为了维持与大多数 OEM 鼠标器的兼容性,本书重点讨论鼠标器函数 37 以下的函数。附录 A 提供了所有 52 个鼠标器函数的文档,以及描述它们执行功能的程序代码段落。

要求

MS-DOS 和 Windows 章节的要求是明显不同的。首先列出 MS-DOS 的要求,然后是 Windows 的要求。

MS-DOS 资源

对于 MS-DOS 编程要求下列硬件和软件环境:

硬件

MS-DOS 硬件要求如下:

- 一台 IBM8086、8088、80286、80386、80486 或 100% 的兼容机。
- MS-DOS 或 PC-DOS2.0 或更新版本。
- 一个鼠标器(Microsoft 或 100% 兼容)以及适当的鼠标器驱动程序(MOUSE.COM 和 MOUSE.SYS)。
- 从第 1 章到第 16 章的例子将在 CGA、EGA 和 VGA 视频系统下运行。
- 第 18 章到第 21 章的例子要求 EGA 和 VGA 视频系统。

软件

MS-DOS 软件要求如下:

- Borland 程序员需要 Turbo C 2.0 版本, Borland C++ 版本 1.0 或更新。
- Microsoft 程序员需要 Microsoft C 6.0 版本, 或 Microsoft Quick C 2.5 版本或更新。
- 两个源文件都要求与汇编语言的程序共同编译。尽管如此, 不要求具备汇编程序, 本书的软盘上提供了所需程序的目标代码。假如读者想改变汇编代码, 需要准备 Borland Turbo 汇编程序(版本 1.0)或者 Microsoft 宏汇编(版本 5.0 或更新)。Borland 公司已经把 Turbo 汇编程序与 C、C++ 编译程序综合到一起, 读者应查阅有关的 Borland 文件。而 Microsoft 的宏汇编并未包含在 Microsoft C 或 C++ 编译程序之中, 因此必须要分开购买。

Windows 资源

对于 Windows 编程实例, 要求下列硬件和软件:

硬件

- 一台 IBM 80286、80386、80486 或 100% 兼容机。
- MS-DOS 或 PC-DOS3.1 或更新版本。
- 一只 Windows 支持的鼠标器。
- 一台 Windows 支持的监视器。

软件

Windows 环境要求下列软件:

- Windows 编程实例是在 Windows3.1 的新标准下编写的, 因此要求运行 Windows 3.1。
- Borland C++ 程序员需要 Windows 3.1 下的 Turbo C++3.1 版本, Borland C++ 3.1 或更新版本。
- Microsoft 程序员需要 Windows 3.1 SDK 编译程序下的 7.0 版本或更新版本的 Microsoft C/C++。
- Windows 代码将直接由 Borland 编译程序转成 Microsoft 编译程序, 不使用 Turbo OWL 库或者 Microsoft 类库, 只使用 C 代码。因此, Windows 程序代码应当转换成任何一种编译程序, 以便维持 Windows 3.1 SDK 的兼容性。

测试

考虑到鼠标器编程的复杂性、可使用的 OEM 鼠标器数目、视频监视器的配置和兼容性,以及所支持的编译程序,本书中使用的每一个程序实例都彻底地被测试过,以保证它们的功能性和可靠性。下面部分将详细讲述如何进行测试。

MS-DOS 测试

对于 DOS 的章节,将对每一个程序实例进行下面的测试:

- 使用 Borland Turbo C 2.0、Borland C/C++ 2.0、Microsoft C/C++ 7.0 和 Microsoft Quick C 2.5 进行编译和运行。使用的汇编程序是 Borland Turbo 汇编 1.0 和 Microsoft 宏汇编 5.0。
- 在 SMALL、MEDIUM、COMPACT、LARGE 和 HUGE 存储模式下用前面讲到的编译程序进行编译和运行。
- 若提供了文本方式,则在文本和图形两种方式下编译和运行程序。
- 在 EGA 和 VGA 彩色监视器上,以彩显和单显两种方式运行。CGA 方式是用 EGA 显示器仿真运行的。
- 80286、80386、80486 机器上在 MS-DOS 3.2、4.0 和 5.0 版本下运行。
- 在 80? 86 机器上,使用 Windows 3.1,在 MS-DOS 5.0 环境下运行 DOS 的应用程序实例。
- 第 18 章到 21 章的例子只在纯 EGA、VGA 和 Super VGA 系统下运行,并且在 640×200、640×350、640×480 和 640×600 16 位彩显方式下进行测试。除此而外,测试还包括 640×350 和 640×480 的单显方式。

测试要求包括:

- Microsoft 的鼠标器驱动程序版本 7.00、7.04、8.16 和 8.20a,还要求测试 Logitech 的鼠标器驱动程序 4.0 和 6.1 版本、Krafft 跟踪球及驱动程序 3.0、IMSI 鼠标器驱动程序 3.03、SUN Crystal 鼠标器驱动程序 3.81 和 ATI InPort 鼠标器驱动程序 2.00。
- 使用的图形卡包括 Everex EGA 卡、Compaq EGA 卡、Everex VGA 卡、ATI Wonder Plus 16 VGA 卡、Paradise Plus 16 VGA 卡、Video-7 VRAM II ERGO 卡和 ATI Graphic Ultra Accelerator 卡。
- 使用的监视器包括 NEC Multisynch 5FG、Multisynch 3-D、Multisynch II、Hyundai VGA、Compaq VGA 和 Compaq EGA。

Windows 测试

所有的 Windows 编程例题都在 Windows 3.1 下使用 Borland C/C++ 3.1 编译程序和 Microsoft 7.0 C/C++ 编译程序进行编译和运行。这些例子也在 80486 机器上在 DOS 5.0 下进行过测试。

目 录

第一部分 鼠标器接口程序

第 1 章 如何与鼠标器通信:33H 中断	(3)
1.1 鼠标器驱动程序	(3)
1.2 33H 中断	(3)
1.2.1 中断服务函数	(4)
1.2.2 工作寄存器	(4)
1.2.3 说明	(4)
1.2.4 通用寄存器	(5)
1.2.5 做一个鼠标器驱动程序的函数调用	(5)
1.3 Borland 和 Microsoft 编译的兼容性	(6)
1.3.1 修改 COMPILER. H	(8)
1.3.2 图形和文本 I/O	(8)
第 2 章 视频函数库	(9)
2.1 视频函数库的介绍	(9)
2.1.1 对于图形和文本使用相同的调用格式	(9)
2.1.2 视频函数库的内部工作情况	(10)
2.1.3 全局变量	(11)
2.2 视频函数库的源代码	(11)
2.2.1 Borland 视频函数库	(11)
2.2.2 Microsoft 视频函数库	(17)
2.3 有关视频函数库的注意事项	(23)
2.4 测试视频函数库	(24)
2.4.1 编译测试程序	(24)
2.5 运行测试程序	(25)
第 3 章 十个必需的鼠标器函数	(27)
3.1 鼠标键	(27)
3.2 鼠标器函数	(27)
3.3 全局变量	(28)
3.4 关于图形方式和文本方式	(29)
3.5 鼠标器函数库	(29)
3.6 鼠标器函数 0:复位和取状态	(36)
3.6.1 确定鼠标器存在与否	(36)

3.6.2	鼠标键的数目	(3 7)
3.6.3	文本单元的大小	(3 7)
3.6.4	鼠标器光标的大小	(3 8)
3.7	第一个鼠标器程序	(3 8)
3.8	使鼠标器光标消隐	(4 0)
3.8.1	根据鼠标器光标大小限制鼠标器移动范围	(4 1)
3.8.2	警告:鼠标器函数 38	(4 2)
第 4 章	如何防止鼠标器转换为老鼠	(4 3)
4.1	黄金规则	(4 3)
4.2	问题所在	(4 3)
4.3	解决办法	(4 6)
4.3.1	方法 1:消隐和显示鼠标器光标	(4 6)
4.3.2	方法 2:设置一个禁止区	(5 0)
4.4	EGA 寄存器、兼容性和 OEM 鼠标器	(5 3)
4.4.1	EGA.SYS	(5 4)
4.4.2	EGA_REG_READ 变量	(5 5)
4.4.3	EGA 的解决办法	(5 5)
第 5 章	跟踪鼠标器光标和用键盘模拟光标的移动	(5 6)
5.1	跟踪鼠标器的坐标值	(5 6)
5.2	在图形方式下跟踪文本坐标	(5 8)
5.3	用键盘模拟鼠标器光标的移动	(6 0)
5.3.1	移动鼠标器光标	(6 0)
第 6 章	确定鼠标键状态和限制移动范围	(6 3)
6.1	鼠标键函数	(6 3)
6.2	捕获单击键:按下及释放	(6 4)
6.2.1	相反的捕获	(6 6)
6.3	鼠标键的双击	(6 7)
6.3.1	为双击键定时	(6 7)
6.3.2	确认一个双击键	(6 7)
6.4	顺序应用程序中的范围限制	(7 0)
第 7 章	编制事件管理程序	(7 5)
7.1	有关定时的问题	(7 5)
7.2	什么是事件管理程序	(7 5)
7.3	安装事件管理程序	(7 6)
7.3.1	调用掩码	(7 6)
7.3.2	事件管理程序地址	(7 6)
7.3.3	用于安装事件管理程序的鼠标器函数	(7 6)
7.4	鼠标器函数 12:设置事件管理程序	(7 7)
7.4.1	使用限制	(7 9)
7.4.2	克服汇编语言中的局限性	(7 9)

7.5	编译事件管理程序	(85)
7.5.1	不同的存储模式	(86)
7.5.2	如果没有汇编程序	(86)
第8章	使用事件管理程序	(87)
8.1	编译一个使用事件管理程序的应用程序	(87)
8.1.1	文件顺序	(87)
8.1.2	堆栈检测	(87)
8.1.3	集成环境编译	(88)
8.1.4	命令行编译	(88)
8.1.5	新的头文件	(90)
8.2	使用事件管理程序的第一个程序	(90)
8.2.1	终止应用程序和事件管理程序	(92)
8.2.2	重置全局状态变量	(92)
8.3	对事件管理与事件处理程序的全面调试	(93)
8.3.1	重新访问触摸光标函数(touch_cursor())	(96)
8.3.2	使用鼠标器光标的坐标	(97)
8.4	其它事件管理函数	(97)
8.4.1	鼠标器函数 20: 交换事件管理程序	(97)
8.4.2	鼠标器函数 24: 设置交替的事件管理程序	(98)
8.5	spawn()和exec()函数	(98)
8.6	驻留程序(TSRs)	(98)
8.7	继续前进	(99)
第9章	菜单管理	(100)
9.1	浮动菜单	(100)
9.1.1	保持操作简单	(100)
9.2	关于横向菜单	(101)
9.2.1	横向菜单的范围限制	(101)
9.3	关于纵向菜单	(102)
9.3.1	纵向菜单的范围限制	(102)
9.4	关于组合式菜单	(102)
9.5	浮动菜单程序	(103)
9.6	细节详解	(113)
9.6.1	存储范围	(113)
9.6.2	恢复鼠标器光标的位置	(114)
9.6.3	重置事件状态变量	(115)
9.7	键盘输入	(116)
第10章	把鼠标器作为十字准线使用	(120)
10.1	定义十字准线	(120)
10.2	使用鼠标器光标进行联络	(121)
10.3	十字准线算法	(122)

10.4	十字准线的限制	(122)
10.5	十字准线实例	(123)
10.6	程序细节	(128)
10.6.1	定义十字准线	(128)
10.6.2	关闭鼠标器光标	(128)
10.6.3	打开鼠标器光标	(129)
10.6.4	定义图表坐标系	(129)
10.7	为什么使用十字准线	(131)
第 11 章	在 CAD 类应用中使用鼠标器	(132)
11.1	伸展还是拖曳	(132)
11.2	图形目标的伸展	(132)
11.2.1	确定端点	(133)
11.2.2	Undo 选项	(133)
11.2.3	XOR 和 COPY_PUT 画图方式	(133)
11.2.4	直线伸展实例	(134)
11.2.5	矩形伸展实例	(137)
11.2.6	椭圆伸展实例	(138)
第 12 章	可选的鼠标器函数	(143)
12.1	六个新的鼠标器函数	(143)
12.2	使用可选函数	(147)
12.3	有关返回的信息	(149)
12.3.1	鼠标器类型和中断请求排队(IRQ)号	(149)
12.3.2	CRT 页	(150)
12.3.3	主版本号和副版本号	(150)
第 13 章	速度的设置: Mickeys 和像素	(151)
13.1	Mickeys	(151)
13.1.1	Mickey 像素比	(151)
13.1.2	放大因子	(152)
13.1.3	倍速阈值	(152)
13.2	改变速度	(152)
13.2.1	速度灵敏度	(152)
13.2.2	速度灵敏度实例	(153)
13.2.3	倍速阈值实例	(156)
13.2.4	最大速度	(158)
第 14 章	改变和管理图形鼠标器光标	(159)
14.1	图形鼠标器光标机构	(159)
14.1.1	屏幕掩码和光标掩码	(159)
14.1.2	位扩展	(161)
14.1.3	热点	(162)
14.2	设置鼠标器光标形状	(162)

14.3	DOS 下的 Windows 方式光标管理程序	(167)
14.3.1	边界上的尺寸箭头	(167)
14.3.2	缺省的鼠标器光标形状	(167)
第 15 章	鼠标器的文本光标	(184)
15.1	两类文本光标	(184)
15.1.1	软件文本光标	(184)
15.1.2	硬件文本光标	(185)
15.2	使用设置文本光标函数(set_text_cursor())	(185)
15.2.1	修改软件文本光标	(185)
15.2.2	修改硬件文本光标	(188)
第 16 章	把函数组合到库文件中	(191)
16.1	一个真实的鼠标器函数库	(191)
16.1.1	库文件	(191)
16.1.2	创建库文件	(192)
16.2	使用库	(196)
16.2.1	在用户应用程序中使用函数库	(198)

第二部分 建立你自己的鼠标器光标和难以捉摸的 800×600 16 位彩色显示方式

第 17 章	建立你自己的鼠标器光标	(203)
17.1	为什么要建立自己的鼠标器光标	(203)
17.1.1	800×600 16 位彩色 Super VGA 方式	(204)
17.1.2	已有的解决办法	(205)
17.1.3	通用的解决办法	(206)
第 18 章	编制一个子画面驱动程序	(208)
18.1	图形鼠标器光标	(208)
18.1.1	EGA/VGA 读/写方式 0	(208)
18.1.2	光标/子画面机构	(210)
18.2	子画面驱动程序	(211)
18.2.1	低级图形函数	(211)
18.2.2	子画面函数	(213)
18.2.3	子画面驱动程序源代码	(213)
18.2.4	编译 SPRITELL.C	(232)
18.2.5	关于新的鼠标器函数库	(234)
第 19 章	新的鼠标器函数库和子画面光标	(235)
19.1	修改鼠标器函数库	(235)
19.1.1	显示和消隐函数	(235)
19.1.2	设置一个禁止区域	(235)
19.1.3	事件处理程序	(236)

19.1.4	光标的位置	(236)
19.2	新的鼠标器函数库	(236)
19.2.1	编译 MOUSEDRV.C	(246)
19.3	将事件管理程序、鼠标器函数库和子画面驱动程序组合在一起	(247)
19.3.1	新的函数库头文件	(248)
19.4	使用子画面光标	(251)
19.4.1	初始化子画面光标	(254)
19.4.2	销毁子画面光标	(256)
19.4.3	缺省的鼠标器光标	(256)
第 20 章	800×600 16 位彩显方式下的子画面光标	(257)
20.1	新的视频函数	(257)
20.2	在任意 16 位彩显方式下运行	(261)
20.3	设置 800×600 16 位彩显方式	(264)
20.4	确定适当的操作	(265)
第 21 章	子画面的使用规则和特点	(270)
21.1	子画面光标的使用规则	(270)
21.1.1	全局变量	(270)
21.1.2	合法的取值范围	(270)
21.1.3	显示边界	(270)
21.1.4	消隐子画面光标	(271)
21.1.5	切换光标	(271)
21.1.6	使事件管理程序处于激活状态	(272)
21.1.7	销毁子画面光标	(272)
21.2	子画面光标的特点	(272)
21.2.1	修改子画面光标的颜色和写方式	(273)
21.2.2	改变子画面光标掩码	(279)
21.2.3	使用其它子画面光标	(282)
21.2.4	双子画面光标	(283)
21.2.5	双子画面光标 CAD 镜像程序	(284)

第三部分 在 Windows 环境下与鼠标器通信

第 22 章	通用的 Windows 应用程序	(299)
22.1	Windows, 大的事件处理程序	(299)
22.1.1	在 Windows 环境下与鼠标器通信	(300)
22.2	基本的 Windows 应用程序	(300)
22.2.1	模块定义文件	(302)
22.2.2	编译 GENERIC.C	(303)
22.2.3	运行 GENERIC.EXE	(304)
22.2.4	理解 GENERIC.C	(307)

22.2.5 理解 GENERIC.DEF	(310)
第 23 章 Windows 环境下的鼠标键消息	(312)
23.1 鼠标键的压下和释放消息	(312)
23.2 来自鼠标键消息的附加信息	(318)
23.2.1 用户区的信息细节	(319)
23.2.2 非用户区的信息细节	(325)
23.3 双击键定时	(331)
第 24 章 改变 Windows 的光标	(337)
24.1 标准的 Windows 光标	(337)
24.1.1 设置缺省的已注册光标	(337)
24.1.2 装入 Windows 光标 on the Fly	(339)
24.1.3 鼠标器移动消息(WM_MOUSEMOVE)	(343)
24.1.4 设置光标消息(WM_SETCURSOR)	(343)
24.2 创建和使用自己的光标	(348)
24.2.1 资源光标	(349)
24.2.2 动态光标	(355)
第 25 章 其它与 Windows 鼠标器有关的题目	(365)
25.1 在 Windows 环境下跟踪光标的坐标值	(365)
25.2 用键盘模拟鼠标器移动	(370)
25.2.1 确定鼠标器是否存在	(371)
25.2.2 利用键盘移动光标	(371)
25.2.3 内部显示计数	(371)
25.2.4 光标移动的范围限制	(372)
25.2.5 键盘模拟鼠标器程序	(373)
25.3 鼠标器消息发送范围	(379)
25.4 非用户区按键测试消息(WM_NCHITTEST)	(380)

第四部分 附录

附录 A 52 个鼠标器函数的文档	(383)
附录 B Windows 鼠标器和光标函数	(429)
附录 C Windows 鼠标器消息	(437)

第一部分

鼠标器接口程序

