

夏东涛等译

MS - DOS

设备驱动程序剖析与实现

北京科海培训中心

序

本书用于指导大家设计、编制用户MS-DOS下的设备驱动程序。为了便于大家理解和掌握，本书采用先易后难，循序渐进的方法，开始几章用一个非常简单的框架驱动程序做为导引，说明设备驱动程序的工作过程，然后介绍较复杂的驱动程序，最后两章提供给大家一套完整的，用于编制和调试驱动程序的技术。

大家知道，1982年微机的MS-DOS刚开始使用时，对于在BIOS或DOS上如何工作略知一二，但对如何最佳地使用微机的硬件和外设了解甚少。因此，大家可能遇到这样的问题，就是你编的程序不能总能在其它的MS-DOS支持的机器上或所有的IBM PC兼容机上使用。那么能否找出一个方法来使得一个版本的软件应用程序可运行在任何基于MS-DOS或PC-DOS的微计算机上呢？

我们发现最通用的方法是用设备驱动程序从应用程序中将那些与机器直接有关的特征取出，然后以一个可安装的设备驱动程序的形式将它们装进DOS中去。这些相对较小的驱动程序可为每个机器编写一个，使用这种方法编制较大应用程序就可以做成一个单一的标准化版本，而且可以在所有的MS-DOS和PC-DOS支持的微机上使用。

学习时，为了易于掌握，你可以将书中给的例子全部打入机器进行试验，当然你也可以向我们订购带有所有设备驱动程序的盘片。

问题 { 抄写, 学习, 思考 }
G28110101
干部素质
组织令传递加
7:10
7:30
招兵买马
不能2000
海堤
自

序

第一章 引言.....	(1)
1.1 本书介绍.....	(1)
1.2 阅读对象.....	(2)
1.3 使用本书需要什么?	(2)
1.4 本书的目的.....	(3)
1.5 本书中使用的约定.....	(3)
1.6 如何使用本书.....	(3)
1.7 本书各章节介绍.....	(4)
1.8 问题.....	(4)
第二章 基本概念.....	(5)
2.1 通过软件控制设备.....	(5)
2.1.1 设备基本原理概述.....	(5)
2.1.2 设备的程序控制	(8)
2.2 DOS 概述.....	(15)
2.3 DOS 服务功能.....	(17)
2.3.1 DOS 中断.....	(17)
2.3.2 DOS 服务程序.....	(19)
2.3.3 使用DOS服务程序.....	(21)
2.4 DOS 设备管理.....	(21)
2.5 DOS设备驱动程序.....	(22)
2.5.1 新设备的设备驱动程序.....	(22)
2.5.2 替换 驱动程序 (ANSI·SYS)	(22)
2.5.3 新旧设备驱动程序的关系.....	(22)
2.5.4 为标准设备定制驱动程序.....	(23)
2.5.5 无设备之驱动程序.....	(24)
2.5.6 驱动程序结构框架.....	(24)
2.5.7 DOS怎样与驱动程序通信.....	(24)
2.6 块设备和字符设备.....	(28)
2.7 设备驱动程序命令.....	(28)
2.7.1 初始化命令.....	(30)
2.7.2 Media Check和获得BIOS参数块命令.....	(30)
2.7.3 IOCTL 输入命令	(30)
2.7.4 输入命令	(31)
2.7.5 无破坏输入命令.....	(31)

2.7.6	输入状态命令	(31)
2.7.7	输入废除命令	(31)
2.7.8	输出命令	(31)
2.7.9	带验证的输出命令	(31)
2.7.10	输出状态命令	(31)
2.7.11	输出废除命令	(32)
2.7.12	IOCTL 输出命令	(32)
2.7.13	设备打开命令	(32)
2.7.14	设备关闭命令	(32)
2.7.15	可移动存储介质命令	(32)
2.7.16	输出至忙命令	(32)
2.7.17	其它命令	(32)
2.8	跟踪程序对设备的请求	(32)
2.9	建立设备驱动程序的过程	(34)
2.9.1	编写设备驱动程序	(34)
2.9.2	汇编设备驱动程序	(34)
2.9.3	连接设备驱动程序	(35)
2.9.4	转换 EXE 为 COM 格式	(35)
2.9.5	将设备驱动程序装入 DOS	(35)
2.10	本章小结	(36)
2.11	问题	(36)
第三章	简单的设备驱动程序	(37)
3.1	设备驱动程序的基本框架	(37)
3.2	简单设备驱动程序的各个部分	(38)
3.3	对汇编程序的指示	(39)
3.4	主过程代码	(40)
3.5	DOS 要求的设备标题	(41)
3.6	设备驱动程序的工作空间	(43)
3.7	STRATEGY 过程	(44)
3.8	INTERRUPT 过程	(45)
3.9	局部过程	(46)
3.10	DOS 命令处理	(47)
3.11	错误出口过程	(48)
3.12	公共出口过程	(49)
3.13	程序结束部分	(50)
3.14	完整的 Simple 设备驱动程序	(50)
3.15	建立 Simple 设备驱动程序	(53)
3.16	使用 Simple 设备驱动程序	(53)

3.17	假如不工作怎么办?	(53)
3.18	本章小结	(53)
3.19	问题	(53)
第四章	控制台设备驱动程序	(55)
4.1	控制台设备驱动程序	(55)
4.2	设计自己的控制台设备驱动程序	(55)
4.3	编写控制台设备驱动程序概述	(59)
4.3.1	程序说明	(60)
4.3.2	汇编程序伪指令	(60)
4.3.3	主过程代码	(62)
4.3.4	设备标题	(62)
4.3.5	控制台设备驱动程序的工作区	(64)
4.3.6	STRATEGY过程(策略过程)	(64)
4.3.7	INTERRUPT过程(中断过程)	(65)
4.3.8	局部过程	(66)
4.3.9	DOS命令处理	(66)
4.3.10	错误和公共退出	(72)
4.3.11	程序结束	(73)
4.4	完整的控制台设备驱动程序	(74)
4.5	关于DOS版本的注解	(81)
4.6	建立更替的控制台设备驱动程序	(81)
4.7	本章小结	(81)
4.8	问题	(81)
第五章	打印机设备驱动程序	(82)
5.1	打印机类型	(82)
5.2	I/O控制和IOCTL调用	(83)
5.3	IOCTL程序	(84)
5.4	建立和使用IOCTL程序	(87)
5.5	用于串行适配器和并行适配器的BIOS服务程序	(88)
5.6	打印机设备驱动程序内部构造	(90)
5.6.1	设备标题	(92)
5.6.2	设备驱动程序的工作空间	(94)
5.6.3	STRATEGY, INTERRUPT过程及局部过程	(94)
5.6.4	DOS命令处理	(95)
5.6.5	出错退出	(103)
5.6.6	公共退出	(103)
5.6.7	程序结束	(103)

5.7	建立打印机设备驱动程序.....	(10)
5.8	使用打印机设备驱动程序.....	(113)
5.9	本章小结.....	(113)
5.10	问题.....	(113)
第六章	时钟设备驱动程序.....	(114)
6.1	时钟/日历芯片.....	(114)
6.2	时钟设备驱动程序的功能.....	(114)
6.3	PC时钟及定时信号概述.....	(115)
6.4	对MM58167A时钟芯片编程.....	(116)
6.4.1	二进制编码的十进制值.....	(117)
6.5	寻找时钟芯片.....	(117)
6.6	驻留程序.....	(118)
6.7	使用定时器中断显示时间.....	(118)
6.8	认识时钟设备驱动程序.....	(119)
6.8.1	开始.....	(119)
6.8.2	设备标题.....	(121)
6.8.3	时钟设备驱动程序的工作空间.....	(122)
6.8.4	策略和中断过程.....	(123)
6.8.5	用于时钟设备驱动程序的局部过程.....	(124)
6.8.6	DOS命令处理.....	(128)
6.8.7	错误退出部分.....	(135)
6.8.8	公共退出部分.....	(135)
6.8.9	程序结束.....	(136)
6.9	建立时钟设备驱动程序.....	(139)
6.10	时钟设备驱动程序的工作.....	(155)
6.11	本章小结.....	(156)
6.12	问题.....	(156)
第七章	磁盘内部结构介绍.....	(157)
7.1	磁盘的物理性质.....	(157)
7.1.1	磁盘类型.....	(157)
7.2	磁盘驱动器上的数据组织.....	(157)
7.2.1	磁道.....	(157)
7.2.2	存储容量估算.....	(157)
7.2.3	数据组织成扇区.....	(158)
7.2.4	扇区编号和划分.....	(158)
7.2.5	扇区划分的DOS标准.....	(159)
7.2.6	磁盘格式化.....	(159)

7、3	DOS磁盘支持的技术细节.....	(159)
7.3.1	DOS支持的磁盘.....	(159)
7.3.2	磁盘组织方式.....	(160)
7、4	引导区/保留区,文件分配表和盘簇.....	(160)
7.4.1	文件存储单元——盘簇 (cluster)	(162)
7.4.2	文件分配表 (FAT)	(162)
7.4.3	在FAT中记录盘簇.....	(163)
7.4.4	盘簇、链和FAT.....	(163)
7.4.5	文件分配表的数目 (几乎) 总是 2	(165)
7、5	文件目录.....	(165)
7、6	磁盘容量:	(168)
7.6.1	FAT表目: 12或16位?	(168)
7.6.2	DOS磁盘容量限制.....	(169)
7、7	重要的磁盘参数.....	(169)
7.7.1	启动区 (引导区)	(169)
7.7.2	BIOS参数块 (BPB)	(169)
7.7.3	使用BPB查找信息.....	(172)
7、8	DOS磁盘设备驱动程序.....	(173)
7.8.1	DOS和磁盘设备驱动程序.....	(173)
7.8.2	磁盘分类.....	(173)
7、9	磁盘设备驱动程序命令.....	(175)
7.9.1	初始化命令.....	(176)
7.9.2	Media Check (介质检查) 命令.....	(177)
7.9.3	Get BPB (获得BIOS参数块) 命令.....	(178)
7.9.4	IOCTL Inpnt命令.....	(178)
7.9.5	Input (输入) 命令.....	(178)
7.9.6	Output (输出) 命令.....	(178)
7.9.7	Output With Verity (带验证输出) 命令.....	(178)
7.9.8	IOCTL Output命令.....	(179)
7.9.9	Device Open (设备打开) 命令.....	(179)
7.9.10	Device Close (设备关闭) 命令.....	(179)
7.9.8	Removable Media (可移动介质) 命令.....	(179)
7、10	本章小结.....	(179)
7、8	问题.....	(180)

第八章	RAM磁盘设备驱动程序.....	(181)
8、1	使用RAM磁盘设备驱动程序.....	(181)
8、2	RAM磁盘及其工作原理.....	(181)
8、3	RAM磁盘设备驱动程序.....	(182)

8.3.1	指定RAM磁盘的内部格式.....	(183)
8.3.2	一些RAM磁盘设备驱动程序的设计选择.....	(184)
8、4	RAM磁盘设备驱动程序使用的命令.....	(184)
8、5	RAM磁盘设备驱动程序清单.....	(185)
8.5.1	该设备驱动程序的工作空间.....	(189)
8.5.2	RAM磁盘的STRATEGY和INTERRUPT部分.....	(190)
8.5.3	局部过程.....	(191)
8.5.4	DOS命令处理.....	(193)
8.5.5	错误出口和公共出口.....	(200)
8.5.6	程序结束部分.....	(200)
8、6	RAM磁盘设备驱动程序整体.....	(201)
8、7	建立RAM磁盘设备驱动程序.....	(209)
8、8	修改RAM磁盘设备驱动程序.....	(209)
8、9	本章小结.....	(210)
8、10	问题.....	(210)
第九章	建造一个完整的全功能设备驱动程序.....	(211)
9、1	要求的工具.....	(211)
9.1.1	完美的编辑程序.....	(211)
9、2	设备驱动程序概述.....	(214)
9.2.1	对汇编程序的指示.....	(215)
9.2.2	设备标题.....	(222)
9.2.3	STRATEGY和INTERRUPT过程.....	(227)
9.2.4	DOS命令处理.....	(227)
9.2.5	从设备驱动程序退出.....	(249)
9.2.6	未实现命令的状态字.....	(251)
9、3	本章小结.....	(251)
9、4	问题.....	(251)
第十章	编程技巧.....	(253)
10、1	编写设备驱动程序的核对清单.....	(253)
10、2	设备驱动程序调试纠错技巧.....	(255)
10、3	编制样板设备驱动程序.....	(256)
10、4	设备驱动程序在内存中位置的确定.....	(266)
10、5	加入调试纠错程序.....	(267)
10、6	新堆栈.....	(271)
10、7	特殊比特.....	(272)
10、8	机器不兼容性.....	(272)
10、9	DOS的差别.....	(272)

10、10	DOS 版本的差别	(273)
10、11	本章小结	(275)
10、12	问题	(275)
附录 A	8086/8088体系结构概述	(276)
A、1	Intel 8086/8088微处理器芯片	(276)
A、2	8086/8088存储器结构	(276)
A.2.1	8086/8088存储器中的数据存储	(276)
A.2.2	段落 (paragraph)	(277)
A.2.3	存储器分段	(278)
A.2.4	存储器分段与段寄存器	(279)
A、3	8086/8088体系结构中的硬件寄存器	(279)
A.3.1	通用寄存器	(280)
A.3.2	指针和变址寄存器	(280)
A.3.3	段寄存器	(280)
A.3.4	指令指针寄存器	(280)
A.3.5	标志	(281)
A、4	8086/8088输入/输出 (I/O) 结构	(281)
A、5	软件中断	(282)
附录 B	BIOS中断	(283)
附录 C	DOS初始化	(295)
C、1	DOS系统盘	(295)
C.1.1	建立系统盘	(295)
C、2	初始化DOS	(295)
附录 D	硬盘的特殊功能	(296)
D、1	分区 (partition)	(296)
D、2	FDISK程序	(296)
D.2.1	生成分区	(296)
D.2.2	删除分区及显示分区信息	(296)
D.2.3	使分区成为当前活动	(296)
D、3	分区扇区	(296)
D.3.1	分区表	(297)
附录 E	问题解答	(299)

第一章 引言

在PC机上,MS-DOS控制设备的核心部分是设备驱动程序。如果没有设备驱动程序,所执行每个程序都需要适配于当前所用的PC机。设备驱动程序遵循一组一致的编程规则,从而在MS-DOS和PC之间提供了标准的接口,这些规则对MS-DOS和PC-DOS、IBM PC及兼容机都是通用的。在本书中,DOS是指MS-DOS和PC-DOS,PC是指IBM-PC及其兼容机。

大家都知道,在DOS的支持下可以运行很多应用程序,如实用程序、数据库、字处理程序、表格处理程序等。DOS已提供了一些内部“服务程序”用于存储数据、绘图、存取磁盘、控制外部硬设备。但很少人知道,DOS本身还有一些内部的设备驱动程序来控制硬件。

标准的DOS可以管理和控制一组标准的PC设备,包括键盘、显示器、磁盘以及串、并行适配板。这些标准的设备驱动程序一般都是操作系统中设备管理程序的一部分,对用户是不可见的。在DOS 2.0版本之前,没有提供给一个一致的访问外部硬件的方法。如果要加一个新设备给PC,就需要用户对用户程序本身进行修改,即重新编制一个专门对新增设备的程序对设备管理程序进行改动。这样,DOS对新增设备的支持就显得非常不方便。

从版本2.0开始,DOS允许用户可安装的设备驱动程序。这些用户的设备驱动程序补充到DOS中去,就可使DOS支持一外部设备。

编写设备驱动程序必须按照Microsoft指定的规则进行,否则就不能安装到DOS中去。这些规则提供了一个一致的连到DOS的接口,它会使DOS对新增设备,用对原有设备的同样方法对待。这些规则对设备驱动程序有特别的格式要求。比如,要求程序必须用一个表开头,向DOS标明所要控制之设备的属性和类别。在表中要有一个条款,告诉DOS如何来控制或调用设备驱动程序。程序的最后要包括一些处理DOS要求该设备驱动程序提供的标准命令的程序代码。这些规则在“DOS技术参考手册”中没有做明确的介绍。

可安装设备驱动程序,使得我们能够向PC增加新的硬件设备并用标准的DOS服务程序去访问这些设备。如果没有设备驱动程序,使用每一个新设备就要修改你的程序。一个DOS新版本出来,就需要修改用户程序来适应新的版本,因此对同样的一台新设备要有几种不同的程序,就显得非常麻烦。而如果没有驱动程序的标准,则要求每个程序都不相同。

虽然很多用户认为用PC去读、写数据是非常容易的事,但对数据怎样从键盘获得,又怎样写到一个磁盘文件中去的过程了解很少,因为这是一个较长和较复杂的过程,它与软件驱动程序有关,通过这本书的学习,大家就会了解的更多一些。

1.1 本书介绍

通过学习本书的内容,使大家能够学会编写设备驱动程序,在系统上联接使用任何一种硬设备。通过分析、解剖几个实例来了解DOS的设备驱动程序的各个组成部分。另外我们还要研究设备驱动程序的工作原理,它是怎样连接DOS的以及它与各种设备相互之间是什么关系。这要研究对新增设备如何写它的设备驱动程序以及对标准的DOS驱动程序如何编写替换程序。

编写设备驱动程序是PC机上软件开发中最有挑战性的任务之一。掌握这一复杂的编程技

术是非常有用的。对驱动程序深入研究的付产品,使你能够对DOS系统调用和内部细节有一个较彻底的了解,并对汇编语言会有更新理解。这是因为要懂得驱动程序,需要使用BIOS中的绝大多数的服务程序。

在了解了设备驱动程序的工作原理后,就可以开始修改本书中提供的设备驱动程序,可以修改RAM DISK设备驱动程序以适应你的需要,并可在PC上写一个新的设备驱动程序。这种可能性是无限的。

1.2 阅读对象

学习这本书要求读者对DOS和PC机要有基本的了解。对那些在PC机上做工作的人会有很大的帮助,因为它提供了一个有价值的工具,使他们能够在PC上新增功能较强的设备,以扩大、增强PC机的能力。

本书对计算机科学系的教师非常有用,近年来出现了不少有关DOS和PC的教科书,但很少有关于设备驱动程序的教材,本书可用于填补此空白。

读者最好要懂得用汇编语言在PC上编程,另外,还要对DOS和BIOS服务程序要有一定的了解。

读者应能够读懂简单的8086/8088汇编程序。必要时可以找一本介绍8086/8088汇编语言编程的基本概念的书看看,例如,Robert Lafore写的Assembly Language Primer for IBM PC and XT (New York: Plume/Waite, New American Library, 1984),最好是能有一定的用汇编编程的经验。对8086/8088CPU,内存(存储器)结构、段编址技术、I/O结构和寄存器结构要了解。在本书的附录A中介绍了这些。

另外,读者应了解DOS所提供的中断和功能调用。这方面的知识可通过读“DOS Technical Reference Manual”来了解。还有一本书就是Steven Armtzust和Ted Forgezon写的Programmer's Reference Manual for IBM Personal Computers (Illinois: Dow Jones-Irwin, 1986)。

读者还应熟悉驻留在ROM中的BIOS程序代码,读者必须了解BIOS功能及如何使用这些功能的基本知识。对于IBM系统,可参见硬件技术参考手册来查找BIOS调用。

由于本书主要介绍设备驱动程序,因此要求读者对各类PC上所连接的设备要有些了解,这包括对设备的类型(例如键盘、打印机、磁盘驱动器等)以及对其它功能(输入、输出)的了解。

1.3 使用本书需要什么?

首先需要的是对了解设备驱动程序和DOS有较大的兴趣。另外,为了更好地使用本书,需要准备的硬件、软件如下:

- IBMPC、XT、AT或兼容机。
- MS-DOS或PC-DOS操作系统,2.0版本或更高的版本。
- Microsoft或IBM的MASM(8086/8088/80286宏汇编)
- LINK(MASM的连接程序)
- EXE2BIN(用于将Linker的输出转换为DOS对驱动程序要求的形式)。
- 文本编辑或字处理程序(用于输入设备驱动程序的源程序)。

关键是两点,一是使用的计算机系统的CPU一定是Intel 8088/8086/80286/80386芯片,二是使用的操作系统必须是MS-DOS版本2.0或更高的版本。

本书是以8086/8088CPU为基础编写的,虽然有些系统用的是30286/80386CPU,象IBMPCAT,但本书讲的内容同样适用于80286/80386系统。如果使用所举例子,操作系统必须是MS-DOS或PC-DOS(2.0版本或更高的版本)的等价系统(因为MS-DOS和PC-DOS在功能上是等价的)。有时我们也会引用到DOS 3.0版本,特别是要涉及到DOS 3.2版本,因为3.2版本为在物理控制、联网及设备共享方面的设备驱动程序提供了更多的功能。

在本书所举的例子中,使用了一些实际代码,因为这些代码都是以8086/8088汇编语言所编写的,主要需用三个DOS实用程序:MASM、LINK和EXE2BIN。

LINK和EXE2BIN是标准的实用程序,一般随DOS一同提供给用户。这些实用程序在编制设备驱动程序时是需要的。

MASM(Microsoft Macro Assembler)是在MS-DOS/PC DOS系统上使用的汇编程序。是由IBM和Microsoft公司提供的。我们认为这两个公司所提供的MASM是完全相同的,即便你使用了一个不同的汇编程序,请你放心,在本书所举的例子中也是可用的,至少可以转换成等价特性进行使用。

除了以上所需要的三个DOS实用程序,还需要一些文本编辑或字处理程序,用来建立ASCII文本文件。EDLIN是一种,一般随DOS提供给你,但EDLIN比较适合用于编较短的程序,它的功能有限,因此最好使用功能强又灵活的字处理程序,这样的程序在市面上可以买到。

1.4 本书的目的

自IBMPC机1981年推出以来,为其已编写了大量的程序。並出版了大量有关的刊物,杂志和书籍。但有关DOS如何作用于程序及设备的信息不是太复杂就是不完整。各种文章和书籍均不能提供有关设备驱动程序的必要信息。

本书用于满足编写和了解设备驱动程序的需要。对于PC初学者可提供DOS内部工作原理的知识;对于PC高级用户可提供为何需要设备驱动程序,又如何使用那样的规定格式的知识;对于PC的高级程序员,本书提供了有关DOS服务程序与设备驱动程序及设备的相互作用关系。本书的主要目的是提供编写设备驱动程序的基本框架,亦讨论支持这种程序的理论。从实际出发,本书还提供了若干多数PC机均可使用的有用的设备驱动程序。

1.5 本书中使用的约定

本书遵循多数PC用户所熟悉的约定。本书中使用的数字如果具有后缀h表示为十六进制,否则为十进制格式。如没有特别指出,则DOS指的是PC-DOS及MS-DOS, Disk(磁盘)指的是硬盘和软盘。Diskettes指的是可移动的磁盘类型。routine(子程序)是一组程序代码,它可以执行某个功能,並且没有特殊要求的格式。Procedure(过程程序)是一组具有指定格式的程序代码,並可以用一个调用(call)来调用它。

1.6 如何使用本书

本书采用循序渐进的方法,开始的几章介绍一些基本概念,随后的几章继续介绍一些编程的方法,每一章都举一些典型的例子供大家学习。在本书的最后几章中,全面介绍编写设备驱动程序的方法及调试技术、技巧。

此书亦可作为一本非常有用的参考书籍,在阅读时,要特别注意书中提供的各种图、表和清单,这对编写设备驱动程序的程序员非常有用。在这方面,第9章(建立一个功能齐全的设备驱动程序)和第10章(编程的技巧)特别有用。

1.7 本书各章节介绍

第二章简单介绍一下在PC环境的结构中,设备驱动程序所起的作用。学习这章你可以了解到如何对设备编程、DOS由哪些部分组成以及设备驱动程序是如何装入的。本章末尾给出设备驱动程序的有关规则。

第三章推出第一个设备驱动程序,虽然这是一个很短的程序,不能解决什么实际问题,但介绍了编写设备驱动程序的基本概念。

第四章给出了几个实际设备驱动程序中的第一个:控制台设备驱动程序,此程序控制着屏幕输出设备和键盘输入设备。为了区别与其它同类程序,为此设备驱动程序增加了一个在键盘上敲键能够发音的功能。

第五章介绍一个打印机设备驱动程序,与标准打印机设备驱动程序不同,它可以控制五台打印机工作,由DOS I/O控制服务程序来选择使用五台工作中的哪一台工作。

第六章介绍一个时钟设备驱动程序,它需要一个硬件时钟/日历设备,这种设备虽不是PC的标准设备,但它却是绝大多数在PC上使用的多功能卡的选件。

此时钟设备驱动程序能保存DOS的时间和日期完整(在机器关机至下一次打开之前)。这使得每次DOS启动时重新打入时间和日期的麻烦被消除。

第七章使读者了解有关磁盘和磁盘驱动器的基本知识。

第八章介绍RAM DISK设备驱动程序,基于第七章中所介绍的信息,本章中将不是针对物理磁盘设备,而是针对用存储器存数据的方法来建立一个磁盘设备驱动程序。大家能够知道设备驱动程序怎样在内存中存数据文件的。

第九章一般地讨论了怎样编写设备驱动程序,详细讨论了这种程序的每一部分,包括DOS要求设备驱动程序所提供的信息等。

第十章介绍一些编程的技术和技巧以及调试方法作为本书结尾。还有一些在DOS各种版本下如何使设备驱动程序工作等较深入的题目也在此章讨论。

1.8 问题

- 〈1〉使用MS-DOS或PC-DOS是否有影响?
- 〈2〉应选择什么样的DOS版本?
- 〈3〉本书中的例子是否能够同时用于PC机和AT机上?

第二章 基本概念

本章中介绍一些DOS的硬件、软件的基本概念，包括：对PC硬设备的编程、DOS的内部操作以及DOS与设备之间的相互关系。

本章的第一部分介绍PC上的各种设备和怎样通过ROM BIOS设问这些设备。第二部分使大家了解PC是用什么方法将程序连接到DOS上去的，并讨论怎样通过DOS来访问设备。第三部份对设备驱动程序做一个概述，介绍有关设备驱动程序的一些基本概念。第四部分介绍建立一个设备驱动程序所需的步骤。

2.1 通过软件控制设备

2.1.1 设备基本原理概述

▲PC上的设备

PC所接的标准硬件设备一般有键盘、显示器、打印机、磁盘，除此之外，PC还可以接很多种硬设备，表2—1列举了一些设备，是按输入设备、输入输出设备和输出设备三大部分分类的。

表2—1 PC机上所用的各种设备

类别	设 备	用 途
输入设备	图像数字化仪 (Image digitizer)	通过摄象机获得视频图象，然后转换成计算机使用的数字图象
	<u>CD—ROM</u>	压缩磁盘系统使其能存储巨库数据和视频图象
	条形码阅读机	使用光学扫描仪设备读入计算机化的超级市场用标签
	图形数字化仪 (Graphics digitizer)	由跟踪印刷的图形图象来获得复杂的图象
	鼠标器/跟踪球 (Mouse/Track ball)	提供一个较好的人—机环境，机械定位设备
输入输出设备	A/D	将模拟信号转换成数字信号，用在测试中
	局域网 (<u>LAN</u>)	将多个PC连接起来可以共享设备和数据
	磁带驱动器 (Tape drives)	将数据后备到磁带上

	录像带记录器 (Video Cassette Recorder)	可同时存贮影视和数据的设备
	Bisync 接口 (Bisync interface)	使用一种特殊的通讯规程可以与大型计算机通讯
	多功能板 (Multifunction Board)	有串、并接口, 可接打印机和调制解调器
	磁盘驱动器 (disk drive)	软、硬盘存数据
输出设备	绘图仪 (plotter)	使用垂直定位的绘图笔, 绘制高分辨的图形
	PROM写入器	对ROM编程
	激光打印机 (Laser printer)	用激光技术实现快速、高分辨打印
	合成器 (Synthesizer)	人工声音产生器, 可以由计算机演奏音乐。
	D/A	可将数据信号转换为模拟信号

▲控制器、适配器和接口

如果一个设备要加到PC机上, 首先PC要承认这个设备。设备常常带有一些插在PC机内硬件插槽上的特殊印刷电路板。这些印刷电路板有几种名字: 控制器、适配器或接口板。实际上, 这些板的作用是在设备和PC机之间提供一个接口。这样就允许PC在总线上通过信号来控制这些设备。这些信号通常叫做“I/O总线信号”, 有多种功能。用I/O端口或地址来识别接到PC上的设备。在总线上传送数据使用的是“数据总线”信号, 总线上的其它控制信号是用来协调所有外接设备的。

较典型的是控制器板直接插在PC母板上的I/O总线上, 变成了PC的组成部分。PC机设计时考虑到允许控制器和设备很容易地加入, 因此而得名“开放式结构 (open architecture)”。正因为采用了开放式结构, PC机才有这样广泛的可用设备。

共有64 k的I/O地址, 每个设备用一组I/O地址, 这组地址是唯一的, PC用这组I/O地址选择作为数据传送的设备。对一个给定的设备, 每个I/O地址履行唯一的功能。例如, 打印机为传送数据有一个I/O地址, 对数据传送的状态也有一个I/O地址, 对打印机的控制还有一个I/O地址, 每一个I/O地址只执行一个功能。

当PC向设备传送数据时, 用OUT指令去选择一个I/O地址和一个被发送的字符。例如,

要送一个 ASCII “A” 到打印机，可以用这样一指令：

```
Mov dx, 0378h ; 对打印机的 I/O 地址
Mov al, 41h ; ASCII A
Out dx, al ; 送字符到打印机
```

当 PC 执行 OUT 指令时，I/O 地址出现在总线上（378 h 安排在地址总线上）；AL 寄存器中的值也出现在总线上（值 41 在数据总线上）。该设备的控制器一直在监视着地址总线，一旦看到 378 h 在总线上，控制器将抓住数据总线上的这个值，然后把它送到设备上去。

由在地址和数据总线上识别 8086/8088 送出的信号，控制器执行控制设备和在 PC 与这些设备之间传送数据的基本功能。

▲PC 的标准设备

设计通用控制器或多功能卡用于控制一组设备：输出设备、时钟、日历、扩充存储器，游戏 I/O 等。目前典型的 PC 系统通常包括有两个标准设备做为通用控制器：串行设备控制器和并行设备控制器。早期的 PC 机有一个游戏接口用于连接游戏棒。

为了使外面的设备能够接触到控制器，一般在控制器卡机壳外面提供一个接头。这些接头通常称之为端口，用作硬件联接机构。串行和并行设备控制器都使用这些接头，分别称之为串行口和并行端口。

注意，上面所说的端口与我们前面 8088/8086 体系结构中所讨论的 I/O 端口不是一回事，8088/8086 的 I/O 端口是内部端口，它是使用 CPU 的特殊指令（IN 或 OUT）通过数据总线访问设备控制器的。这里所说的外部设备端口是指总线外面和接口板上的接口，用来联接设备到各自的控制器上。

将一个新设备连接到 PC 上，简单的方法是将其连接到 PC 的串行端口，并行端口或游戏端口上。这三个端口的主要差别在于传送的电气信号类型及设备与控制器之间数据传送的方式。

▲串行设备

以上说的三种接口中，串行口是用途最多的，它可以联接调制解调器、鼠标器和条形码阅读器。对 8086/8088 而言，数据的传输是双向的，传输速度可高达 9600 波特。打印机也可以接在串行口上，但更多地是接到并行口上使用。用于将设备插入串行端口的接头定义为 RS—232—C 接头，这是因为用于与端口上所连接设备通信的协议接近于 EIA RS—232—C 标准。因此我们把串行口也叫做 RS—232 端口。

▲并行设备

并行接口开始只是为输出设备设计的，主要是为打印机做接口。这个接口的特点是速度较快，数据传输速度可以达到每秒 10 k 字节以上。因为并行接口是为只做输出的设备设计的，因此限制较之串行端口更多些。因此有一种趋势即只在并行接口上接打印机，打印机缓冲器设备和特殊的输出设备。由于最早开发出这个接口的是 Centronics 公司，因此并行口也叫 Centronics 接口。

▲游戏接口

这个接口是最受限制的，因为它只有很简单的输入信号和一个很小的数据缓冲器，它只是用于接一些简单的输入设备，例如游戏棒和跟踪球。

▲高速和 DMA 控制器

有些设备像磁盘驱动器要求传送数据的速度非常高。串、并行和游戏设备使用通用控制

器便无法控制。

串行控制器每秒处理数据可达9600bits (近1200字符/字节), 而硬盘要求每秒要传送100,000字节以上。除了速度要求以外, 还要求很多的控制信号。这些都是串行控制器不能提供的。

由于这些原因, 硬盘驱动器所要求的控制器要能够直接访问PC机的数据总线, 而数据必须以最大效率在设备与内存之间直接传输。这种方式叫做直接存储器访问 (DMA) 方式。其它的一些设备也需要这种高速DMA方式, 如监视器、磁带驱动器、时钟/日历等。这些设备与硬盘一样都要求有专用的控制器, 其接口设备与常见的打印机等设备所用的不一样。

▲字符设备和数据块设备

在PC的环境下, 设备一般分成两种类型: 字符设备和数据块设备, 它们的区别在于设备与PC之间传输数据的方式。

字符设备一次传送一个字符。这样的设备有: 打印机、调制解调器、键盘和鼠标器设备。

数据块设备则不同, 它管理数据和传输数据是以一组若干个字节为单位, 一次传出一组数据, 如512或1024个字节。这样的设备有磁盘和磁带设备。用这些设备, 可以高数据传送速度传送一组数据。当要求高数据传输速度时选用块设备。假想磁盘作为一台字符设备, 则数据传输速度将大大降低。因为磁盘在高速旋转, 传送完一个字符后, 磁盘已不在读第二个字符的位置了, 如果传送一组字符, 磁盘要转许多圈。但将磁盘作为块设备, 则磁盘可一次获得读写头下的一块数据。

所有类型控制器在PC上都可以用。对那些特殊的控制器, 在写设备驱动程序时, 要对设备的工作方法有所了解。

▲控制台设备

这种设备由两部分组成, 一是键盘, 二是显示器, 两部分组成后叫做控制台。控制台设备作为与PC的主要接口这是一个老概念。早年的操作台包括一个键盘和一个代替了显示器的打印机, 是与计算机通讯的唯一途径。在PC上, 可以从键盘将命令输入到计算机, 也可以在显示器的屏幕上看到键入的内容和命令执行的结果。

我们已经概述了PC的标准设备, 第七章中我们要更详细地介绍磁盘设备, 因为它是一个很重要的设备。

2.1.2 设备的程序控制

如果想在你的程序中使用一个特定的设备, 就需要在程序中包括有一个复杂的软件控制子程序。幸而可借助于两种方法完成: 一是使用PC机ROM中的子程序; 二是使用DOS提供的服务程序。

▲ROM—BIOS与DOS服务程序的不同之处

ROM—BIOS (基本输入输出系统) 是由一些子程序组成。通过这些子程序可以控制串行口、并行口、键盘、显示器和磁盘。子程序提供了基本的访问功能, 如读、写。但它们不能以某种形式组织数据, 比如, 通过ROM子程序可以将数据写到磁盘的扇面上去, 但它不能把数据组成一个数据文件。

DOS提供了一些高层软件, 並有很强的处理能力。不必每个设备用BIOS服务程序编写单独的子程序。可以用DOS服务程序在程序中由名字来指定设备。对磁盘数据, 程序可以让DOS将数据组织到一些文件中, 而不是去管理磁盘的扇面。如果要把数据传到串或并行口,