

MS-DOS

编程技巧

朱传乃 主编

人民邮电出版社



MS-DOS 编程技巧

朱传乃 主编

人民邮电出版社

内 容 提 要

本书从基本的计算机系统启动和简单的 MASM 编程开始,全面详实地介绍了 MS-DOS 的编程技术与技巧,主要内容包括 MS-DOS 的存储器管理机构及其应用、磁盘管理、设备驱动程序、中断和中断处理程序、典型的实用程序、保护模式下的编程方法、EMS 和 XMS 的编程方法等。

本书内容通俗易懂、深入浅出,适合计算机应用人员和计算机爱好者阅读,也可作为高等院校教材。

JS169/14

MS-DOS 编程技巧

朱传乃 主编

责任编辑 刘君胜

*

人民邮电出版社出版发行
北京市崇文区夕照寺街 14 号
北京顺义向阳胶印厂印刷
新华书店总店北京发行所经销

*

开本:787×1092 1/16 1997 年 3 月 第 1 版
印张:32 1997 年 3 月 北京第 1 次印刷
字数:805 千字 印数:1—4000 册

ISBN 7-115-06315-X/TP·373

定价:40.00 元

前 言

在微型计算机中,MS-DOS 是使用最为广泛的操作系统。目前,有关介绍 MS-DOS 的资料和书籍,大多侧重说明 MS-DOS 的命令功能(其分类有 MS-DOS 的命令,BATCH 的命令,CONFIG.SYS 的命令,DEBUG 的命令,以及 EDLIN 的命令等),而对于作为操作系统核心部分的系统中断功能调用,特别是 INT 21h 的一些功能调用,在编程中的使用技巧,则缺乏深入实际的分析。本书的目的,是要通过介绍 MS-DOS 的运行机制,结合典型的程序范例,着重介绍通常不易掌握和使用的系统功能调用,以求加深读者对操作系统的理解和进一步提高编程的能力。

该书第一章详细介绍系统启动和 MS-DOS 建立的过程,以及 .COM 文件和 .EXE 文件的编程方法与加载执行的步骤。第二章着重说明 MS-DOS 对存储器的管理机构,并且通过程序实例,特别介绍 MS-DOS 未公开的 INT 21h 的 52h 功能,该功能具有广泛而实际的用途。

第三章介绍 MS-DOS 的磁盘管理,以及与磁盘管理有关的各种重要信息。然后,通过应用程序的例子,具体说明这些信息的实际用途。

第四章讲述设备驱动程序的编写方法。在这一章提供有实用的取代 RS-232-C ROM BIOS 的常驻程序(TSR),以及与 RS-232-C 接口有关的常用样板驱动程序。最后,给出了可供参考的块型设备驱动程序的编写实例。

第五章系统介绍中断和中断处理程序的编写技巧,列举 DOS 通过中断监视应用程序的方法。另外,还简要论述了 DOS 的 CTRL 键,以及对 CTRL 键的编程控制方法。

第六章为典型的实用程序,它包括 DOS 对应用程序访问文件进行追踪的程序,检索、修改、设置和使用环境变量的程序,以及同时打开 20 个以上文件的程序等。通过这些程序,可以学习到更多的编写实用程序的技术,加深对 MS-DOS 内在功能的理解。

第七章为保护模式下的编程方法,具体介绍了 80386/80486 CPU 的内部结构,工作模式,以及在 MS-DOS 下如何直接编程访问保护模式的存储器。

第八章介绍扩充存储器(EMS)和扩展存储器(XMS)及其编程方法,详细叙述系统存储器空间的分配及有效利用,并提供大量的程序实例,说明访问扩充存储器和扩展存储器的编程步骤。

书后附录列出了 MS-DOS 的系统功能调用,特别是 INT 21h 的 DOS 功能,包括公开的和未公开的(未公开的功能今后有可能要进行修改,但是,其中的一些已被许多应用程序所使用),并且对它们进行了详细说明,这是目前最为完整的资料,极具参考价值。此外,附录部分还列出了调用 DOS 功能的 C 函数,供使用 C 语言编程的读者查阅。

本书由朱传乃主编并统稿,是编写组 28 位成员集体劳动的成果,书中的大部分实用程序已在微机上运行通过。但是,由于涉及面广,编者水平有限,书中如有错误或不当之处,敬请读者批评指正。

目 录

第一章 从计算机系统启动到简单的 MASM 编程	1
1.1 系统复位以后 MS-DOS 的建立过程	1
1.1.1 CPU 的复位和系统初始化的步骤	1
1.1.2 引导扇区的读取步骤	3
1.1.3 MS-DOS 的三个文件	6
1.1.4 从磁盘读取引导扇区后加载 OS 的步骤	7
1.1.5 执行 COMMAND.COM	9
1.1.6 内部命令	9
1.2 程序的启动原理	10
1.2.1 COM 程序的启动过程	10
1.2.2 批处理文件的启动	18
1.2.3 EXE 程序的启动步骤	18
1.3 编写简单的程序	22
1.3.1 编写简单的 COM 程序	22
1.3.2 编写简单的 EXE 程序	26
第二章 MS-DOS 的存储器管理机构及其应用	28
2.1 MCB(存储器控制块)	28
2.1.1 MCB 的结构	28
2.1.2 由 DOS 检索和分配存储器块的方法	29
2.2 常驻程序	32
2.2.1 常驻程序的种类	32
2.2.2 程序常驻的方法	33
2.2.3 由正常结束的常驻	38
2.2.4 常驻时的堆栈区域和环境变量区域	41
2.3 常驻程序的检索	43
2.3.1 常驻程序的检索方法	43
2.3.2 MS-DOS 对常驻程序的检查功能	45
2.3.3 常驻程序常驻后的调用方法	46
2.4 编程举例	47
2.4.1 编写 TSR 程序的注意事项	47
2.4.2 TSR 程序的实例	49

第三章 MS-DOS 的磁盘管理	60
3.1 MS-DOS 的磁盘形式	60
3.1.1 软磁盘的管理项目	60
3.1.2 MS-DOS 的各磁盘区域的功用	61
3.2 目录区域和 FAT 区域的检索	66
3.2.1 目录区域的检索	66
3.2.2 FAT 区域的检索	68
3.3 与磁盘有关的各种信息	70
3.3.1 DPB(磁盘参数块)和磁盘的自由空间	70
3.3.2 与磁盘有关的其它操作	74
3.3.3 恢复被删除文件的程序举例	78
第四章 设备驱动程序	94
4.1 设备驱动程序的功用和种类	94
4.1.1 设备驱动程序的功用	94
4.1.2 设备驱动程序的种类	96
4.1.3 设备驱动程序的安装	97
4.2 设备驱动程序的结构	97
4.2.1 设备头	98
4.2.2 请求头(命令包)	104
4.2.3 I/O 请求命令的种类	105
4.3 取代 RS-232-C ROM BIOS 的常驻程序(TSR)	115
4.3.1 ROM BIOS 和 INS 8250 的内部寄存器	115
4.3.2 程序的编写方法	121
4.3.3 RS-232-C 接口信号和样板驱动程序	145
4.4 块型设备驱动程序举例	153
第五章 中断和中断处理程序	159
5.1 关于中断的基本知识	159
5.1.1 中断的种类	159
5.1.2 中断控制器 8259A	161
5.2 硬中断处理程序	166
5.2.1 用汇编语言编写中断用的样板程序	166
5.2.2 结束中断和利用原有中断向量的方法	172
5.3 DOS 通过中断监视程序	175
5.3.1 常驻和释放的步骤	224
5.3.2 命令说明和操作方法	227
5.4 MS-DOS 的 CTRL 组合键和中断处理程序	229
5.4.1 MS-DOS 的 CTRL 组合键	229

5.4.2 对 CTRL 组合键的编程控制方法	230
第六章 典型的实用程序	236
6.1 对应用程序访问文件进行追踪的 FAT.ASM	236
6.1.1 FAT 程序的使用方法	236
6.1.2 Log 文件的显示格式	260
6.1.3 关于 FAT 程序的说明	264
6.2 环境变量的使用方法	265
6.2.1 MS-DOS 的环境变量	265
6.2.2 环境变量的取得方法	266
6.2.3 环境变量的检索、修改和设置	270
6.3 同时打开 20 个以上文件的方法	276
6.3.1 MS-DOS 文件句柄的结构	276
6.3.2 同时打开 20 个以上文件的测试程序	284
第七章 保护模式下的编程方法	291
7.1 80386 CPU 的内部寄存器	291
7.1.1 寄存器的种类	291
7.1.2 标志寄存器	293
7.1.3 控制寄存器	295
7.1.4 工作模式	297
7.2 存储器物理地址的形成	299
7.2.1 虚拟存储器(虚拟地址)和物理存储器(物理地址)	299
7.2.2 描述符表和描述符	300
7.2.3 段选择器和描述符表	309
7.3 进入保护模式的编程方法	322
7.3.1 在保护模式下访问 VRAM 的程序	322
7.3.2 把保护模式的存储器作为数据区使用的程序	329
7.3.3 从 MS-DOS 访问保护模式存储器的样板程序	338
第八章 EMS 和 XMS 的编程方法	346
8.1 概述	346
8.1.1 IBM PC 和 PC/XT 机存储器空间的分配和扩充	346
8.1.2 IBM PC/AT 机存储器空间的分配和扩充	349
8.2 EMM 的编程方法	351
8.2.1 EMM 编程概要	351
8.2.2 EMS 应用程序实例	372
8.3 XMS 的编程方法	392
8.3.1 XMS 的编程步骤	392
8.3.2 XMS 编程实例	398

8.3.3 EMM386.EXE 设备驱动程序	403
附录	409
附录 A-1 系统调用一览表	409
附录 A-2 系统调用说明	413
附录 B-1 INT 21H DOS 功能一览表	419
附录 B-2 INT 21H DOS 功能说明	433
附录 B-3 MS-DOS 2.11 以后版本的 DOS 功能调用所提供的错误代码	462
附录 C 直接调用 DOS 功能的 C 函数一览表	464
附录 D-1 EMM INT 67h 的功能	485
附录 D-2 结束状态一览表	502

第一章 从计算机系统 启动到简单的 MASM 编程

1.1 系统复位以后 MS-DOS 的建立过程

作为理解计算机系统操作的基础,这里首先说明加电之后计算机系统开始操作的过程。

1.1.1 CPU 的复位和系统初始化的步骤

对 8086/88 CPU 来说,当对其进行加电复位之后,则 CPU 的 CS 寄存器的内容为 FFFFH, IP 寄存器的内容为 0000H, CPU 从 1MB 存储器空间的最高位地址 FFFF:0000(此为逻辑地址,物理地址为 FFFF0H)取指令代码(一般 CPU 复位之后都是进入取指令周期)加以执行。通常在 8086 系统中,如图 1.1 所示,存储器地址的高端是分配给 ROM,存储器地址的低端是分配给 RAM。从 FFFF0H 开始执行的地址是指向 ROM 区域,通常在这里写入的是远跳转指令(FAR JMP),跳转到系统专用的初始化处理程序(在 PC/AT 兼容的 286、386 和 486 微机中,其复位进入方式同采用 8088 CPU 的 PC 机是兼容的)。

如图 1.1 所示,CPU 存储器地址空间的低端为中断向量表区域,高端为 ROM BIOS 区域。现在先对这两个区域加以简单地说明。

大家知道,所谓中断,是在 CPU 进行某项处理(不仅包含有意义的处理,也包含无意义的处理)时,中断该处理,转而进行另外的处理。在计算机系统中,预先设置了各种中断原因及其处理方法,如果产生中断,CPU 即跳转到预先设置的程序(称作中断处理程序),执行该程序。中断向量表的区域,是根据中断的原因设置程序地址的表,称作中断向量表。在 8086 CPU 中,按照表 1.1 所列的中断的原因,可以对中断分类如下:

- (1) CPU 的内部中断;
- (2) 硬中断;
- (3) 软中断。

内部中断是 CPU 处于某种困难状态时自身引起的中断。如 CPU 执行用零去除的程序时产生的中断就属于内部中

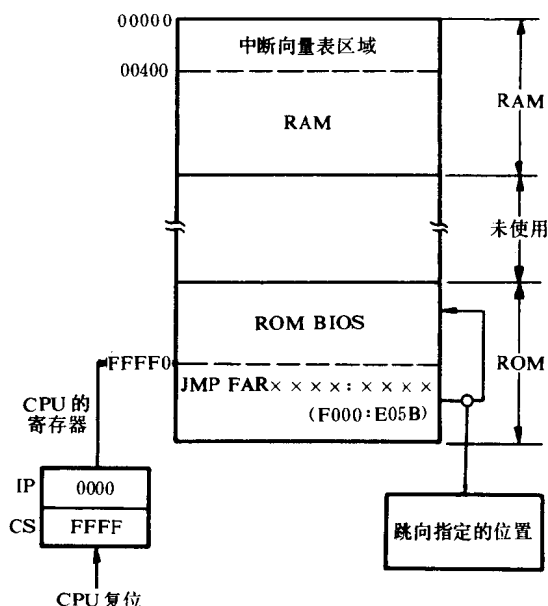


图 1.1 CPU 的存储器地址空间的分配

表 1.1

8086 CPU 的中断

向量号	地 址	说 明	备 注
0	00~03	除法错	DIV、IDIV 指令
1	04~07	单步中断	
2	08~0B	NMI	
3	0C~0F	断点中断	INT 3 指令
4	10~13	溢出中断	INTO 指令
0~255	0~3FF	软中断	
0~255	0~3FF	硬中断	

断。硬件中断，是从 CPU 的外部通过信号线接收中断请求时产生的中断，如键盘输入等即属于硬件中断。若发生硬中断，CPU 暂时中止正在处理的程序，转去执行另外的程序，待该程序执行完了之后，再返回原来的处理程序。

所谓软中断，是执行中的软件要求产生的中断，从用户的角度来看，这是一种子程序的调用功能。对计算机来说这种功能很重要，应用软件可以通过软中断使用操作系统(OS)本身具有的功能(子程序)。例如，在 MS-DOS 中，DOS 的功能 INT 21H 就是软中断。

在 8086 系统中，通过在存储器空间的最低端 0~3FFH 单元的中断向量表登记中断处理程序的地址，可以调用这三类中断。0~3FFH 的地址区域，以 2 个字(4 个字节)为一组，从起始位置开始，对应中断向量号 0~255，总计 256 组。中断向量表的低位字登记中断处理程序的偏移地址，高位字登记中断处理程序的段地址。

在微机系统设计时，一般不对硬中断和软中断分配相同的中断向量，但是在微机系统中可以对中断向量重复使用。IBM PC 微机，从初期开始把 8088 中 Intel 预约的中断向量 00H~1FH 中的一部分分配给硬中断和软中断。在 80286/386 CPU 中，其中的一些中断向量 Intel 进行了新的分配。在微机系统中，兼容性是优先要考虑的，不能修改以前的中断向量，只允许重复使用。例如，中断向量 05H，在 80286/386 CPU 中，是通过边界检查指令 BOUND 执行的中断，但是在 IBM PC 微机中是用于对屏幕进行硬拷贝的软中断，不限于这个例子，微机中的主要硬中断和 ROM BIOS 的软中断是分配给中断向量 08H~1FH，这同 80286/386 CPU 本身的中断相重复。在 MS-DOS 中，和 8086 CPU 的兼容性居于优先的位置，所以忽略 CPU 本身的中断，按微机分配的中断进行处理。但是，在 DOS 扩展器中，多采用对中断向量进行重新分配的办法。另外，也有硬中断和软中断使用相同中断向量的情况，不过这时中断处理程序有必要判别中断的种类。

微机中的 ROM 部分主要包含如下的内容：

- (1) 硬件的初始化程序；
- (2) BIOS (Basic Input Output System：基本的输入/输出系统)；
- (3) ROM BASIC。

其中，BIOS 通常称作 ROM BIOS，是微机提供的可以简单利用硬件功能的输入/输出程序，主要包含如下的一些程序：

- ① 键盘输入；
- ② 屏幕输出；

- ③ 磁盘的输入/输出；
- ④ RS-232C 的输入/输出；
- ⑤ 打印机的输出。

也可以用 FAR CALL 直接调用 ROM BIOS(实际上一部分特殊的应用程序是用 FAR CALL 调用 ROM BIOS 的),但是,通常与 DOS 的功能 INT 21H 一样,是使用软中断调用 ROM BIOS 的功能(例如,在 IBM PC 机中是通过 INT 13H 调用磁盘的 BIOS)。采用这种方法,在推出新的机种时,即使硬件有少量的修改,ROM BIOS 的内容也可能会有所不同,由于这些软中断的中断向量是用 ROM 的初始化程序设置,所以如果用软中断调用 ROM BIOS,即使程序的地址和内容发生变化,OS 和应用程序仍可照样运行,图 1.2 所示为 ROM BIOS 和软中断的关系。

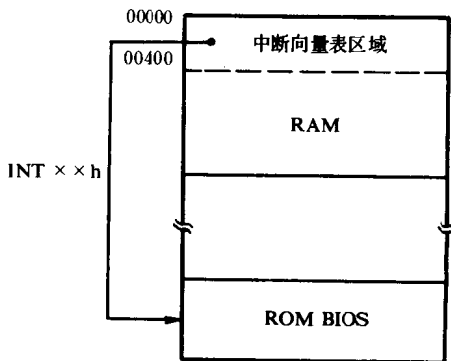


图 1.2 ROM BIOS 和软中断的关系

1.1.2 引导扇区的读取步骤

当 CPU 通过远跳指令从 FFFF : 0000H 地址把控制转移到 ROM 内的初始化程序时,通常按图 1.3 所示的顺序进行处理。

1. 对系统存储器和硬件进行检查

加电之后,对内部存储器和硬件进行检查,如果检查的结果有错误,则产生音响提示信息,或显示出错提示信息及出错代码。

2. 设置中断向量

为了管理中断,如图 1.4 所示,必须对地址 00000H~003FFH 的中断向量表进行设置。

在 IBM PC/AT 兼容机中,ROM BIOS 使用的系统数据区为 400H~4FFH 的区域,也有微机,其 ROM BIOS 使用的系统数据区为 400H~5FFH。大多数微机其自由 RAM 区的最后为 9FFFFH,但是,也有一些机种其自由 RAM 区的最后为 0AFFFFH 或以上的地址空间。另外,有些机种的硬盘适配器卡和显示适配器卡的 BIOS 可能是写扩展 ROM 区。在进行初始化处理时,要检查扩展的 ROM 区域,如果设置有适配器卡,则修改中断向量表,用软中断调用扩展的 ROM BIOS。根据机种,ROM BIOS 的地址会有所不同,IBM PC/AT 机及其兼容机是从 F0000H 开始。早期的 IBM PC 机,基本的 40kB 的 ROM 中包含有 32kB 的 ROM BASIC,ROM BIOS 为 8kB。

3. 硬件初始化

要对键盘和打印机等标准化的硬件进行初始化。一般的微机系统都要使用硬中断,因此还要对中断控制器如 8259A 等进行初始化。如前所述,在检查扩展 ROM 空间时如果有适配器卡,也要对其进行初始化。

4. 读取引导扇区

ROM 中的初始化处理结束以后,读取磁盘的引导扇区。所谓引导扇区,即是磁盘的起始扇区,其内容因操作系统的种类(如 CP/M、OS/2、MS-DOS 等)而有所不同,图 1.5 所示为 MS-DOS 在高容量磁盘(1.2MB)上的存在格式。

如图 1.5 所示,引导扇区是磁盘的起始扇区,位于图中的 IPL(Initial Program Loader)。引

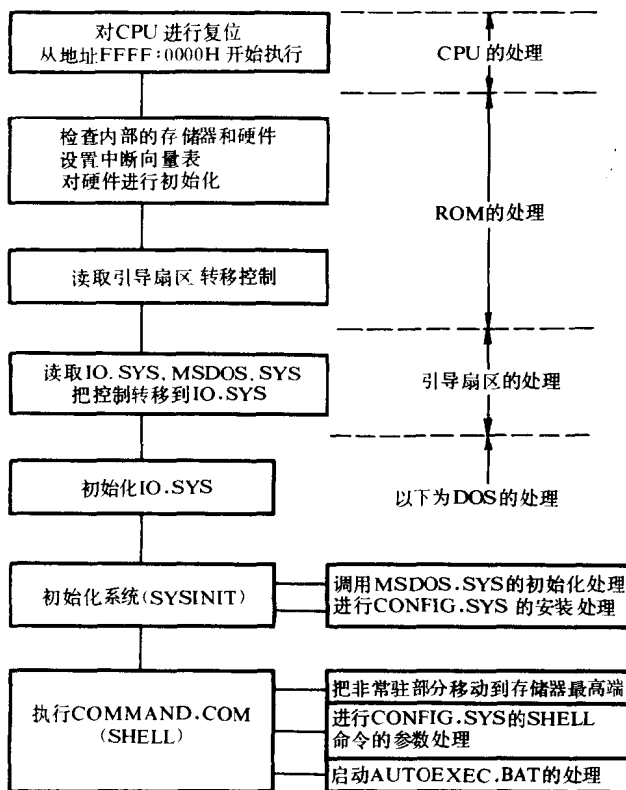


图 1.3 从复位到 MS-DOS 建立的过程

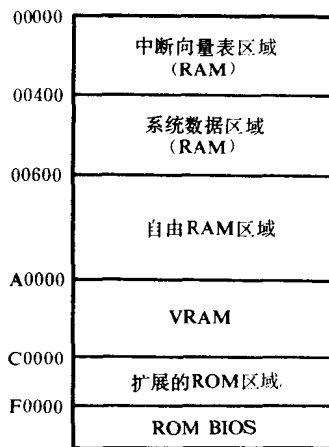


图 1.4 IBM PC 机系列存储器空间的分配

面	0															1														
扇区	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
磁道	0	I R L	第1个 FAT	第2个 FAT	根目录							IO.SYS				IO.SYS														
1	IO.SYS															IO.SYS														
2	IO.SYS															MSDOS.SYS														
3	MSDOS.SYS															MSDOS.SYS														
4	MSDOS.SYS															数据														
	数据															数据														
79	数据															数据														

图 1.5 MS-DOS 在磁盘上的存在格式

导扇区的后面是文件分配表 FAT (File Allocation Table) 的区域，共有 2 组。FAT 的后面是根目录区域。如果是系统盘，根目录的后面是 IO.SYS 和 MSDOS.SYS。IO.SYS 和 MSDOS.SYS 区域的大小因 DOS 的版本不同而不同。图中是以 1.2MB 的高容量磁盘为例，对于其它种类的磁盘，只是磁道数和扇区数不同，IPL, FAT 和根目录等的排列顺序是完全相同的。

从哪一个磁盘驱动器读取引导扇区，按微机系统的设置而有所不同，通常是按软盘、硬盘的顺序进行查询。

在 IBM PC 机(无硬盘)中,没有插入系统盘片时,进入 ROM BASIC,没有 ROM BASIC 的系统,则输出提示信息,等待插入系统盘。

图 1.6 所示为 MS-DOS 的引导扇区中前 256 个字节的内容。256 个字节中的前 3 个字节为 EBH、39H、90H,这个代码是短跳指令和空操作指令(JMP SHORT XXXX, NOP, 这里也有可能是 3 字节的近跳指令, JMP NEAR XXXX: 0E9H XXH XXH)。在这个跳转指令的后面写入的是 OEM 厂家名,和对磁盘进行控制所需要的 BPB(BIOS Parameter Block: BIOS 参数块)。BPB 的后面写入的是 IPL,表 1.2 列出了 MS-DOS 的引导扇区的内容。

09BF: 0000	EB 39 90 49 42 4D 20 20 - 33 2E 33 00 02 01 01 00 . 9 . IBM 3 . 3
09BF: 0010	02 E0 00 60 09 F9 07 00 - 0F 00 02 0F 00 00 EA 25 . . . \ %
09BF: 0020	A3 00 F0 B8 01 02 B9 0E - 00 BA 00 01 BB 00 06 CD
09BF: 0030	13 C6 06 1D 7C 00 EA 4D - 06 00 00 FA 33 C0 8E D0 . . . ! . M . . . 3 . . .
09BF: 0040	BC 00 7C 8E D8 8E C0 50 - 50 50 FB EB D6 B9 0B 00 . . ! P P P
09BF: 0050	FC AC 26 80 3D 00 74 03 - 26 8A 05 AA 8A C4 E2 F1 . . & . = . t . &
09BF: 0060	06 1F 89 47 02 C7 07 2B - 7C FB CD 13 72 67 A0 10 . . . G . . . - ! . . . rg . .
09BF: 0070	7C 98 F7 26 16 7C 03 06 - 1C 7C 03 06 0E 7C A3 3F ! . . & . ! . . . ! . . . ! . ?
09BF: 0080	7C A3 37 7C B8 20 00 F7 - 26 11 7C 8B 1E 0B 7C 03 ! . > ! . . & . ! . . . ! .
09BF: 0090	C3 48 F7 F3 01 06 37 7C - BB 00 05 A1 3F 7C E8 9F . H . . . > ! . . . ? ! . .
09BF: 00A0	00 B8 01 02 E8 B3 00 72 - 19 8B FB B9 0B 00 BED6 r
09BF: 00B0	7D F3 A6 75 0D 8D 7F 20 - BE E1 7D B9 0B 00 F3 A6 } . . u }
09BF: 00C0	74 18 BE 77 7D E8 6A 00 - 32 E4 CD 16 5E 1F 8F 04 t . . ω } . j . 2
09BF: 00D0	8F 44 02 CD 19 BE C0 7D - EB EB A1 1C 05 33 D2 F7 . D . . . } 3 . .
09BF: 00E0	36 0B 7C FE C0 A2 3C 7C - A1 37 7C A3 3D 7C BB 00 6 . ! . . . < ! . > ! . = ! . .
09BF: 00F0	07 A1 37 7C E8 49 00 A1 - 18 7C 2A 06 3B 7C 40 38 . . > ! . I . . . ! * . ; ! @ 8

图 1.6 MS-DOS 的引导扇区的内容

表 1.2 MS-DOS3. X 版本的引导扇区的内容

偏 移 量	长 度	内 容
00H	3 字节	跳向 IPL 的近跳/短跳指令
03H	8 字节	OEM 厂家名和版本
0BH	1 字	每个扇区的字节数
0DH	1 字节	每个分配单位(簇)的扇区数
0EH	1 字	预约的扇区数
10H	1 字节	FAT 的个数
11H	1 字	根目录的项数
13H	1 字	逻辑扇区数
15H	1 字节	介质描述符
16H	1 字	每个 FAT 的扇区数
18H	1 字	每个磁道的扇区数
1AH	1 字	磁头数
1CH	1 字	隐含扇区数
1EH		IPL

BPB 集中设置了管理磁盘所需的信息, IPL 是用于加载 OS 的初始化处理程序, 跳转指令就是要跳到这里 (MS-DOS 的情况下, 在引导扇区的开始是跳转指令, 而 CP/M 在引导扇区的开始写入的是 IPL)。通过 ROM 中的初始化处理, 如果执行位于磁盘上的 IPL 的内容, 则从磁盘把 OS 加载到系统存储器, 再把控制转移给 OS。MS-DOS, CP/M 和 OS/2 等各种各样的 OS 都可以在相同的微机系统中进行启动。

硬盘被分为多个区域时, 从哪个区域启动, 由写入到硬盘引导扇区的 IPL 程序确定。

1. 1. 3 MS-DOS 的三个文件

如图 1. 7 和表 1. 3 所示, MS-DOS 是由 IO. SYS, MS-DOS. SYS 和 COMMAND. COM 等三个文件组成 (IO. SYS 和 MSDOS. SYS 是用 DIR 命令看不到的隐含文件)。在说明从磁盘读取 MS-DOS 的步骤之前, 先对这三个文件加以说明。

表 1. 3 MS-DOS 的组成

文件名	功 能
IO. SYS	设备驱动程序 (CON, AUX, CLOCK, ..., 驱动器 A:, B: ...)
MSDOS. SYS	系统功能 INT 20H~3FH
COMMAND. COM	内部命令 (COPY, TYPE, DEL, ...), 启动批文件

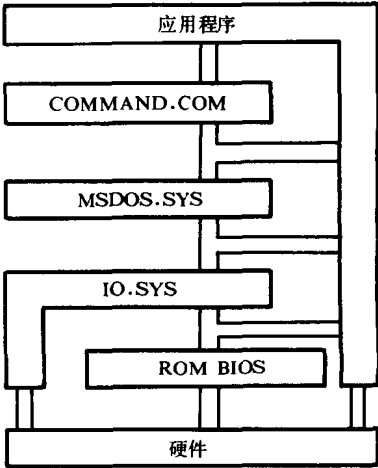


图 1. 7 MS-DOS 的各组成部分的关系

1. 作为设备驱动程序的 IO. SYS

作为设备驱动程序的 IO. SYS 用于担当 MS-DOS 的输入/输出控制, 相当于 MS-DOS 中的 BIOS, 不同机种这一部分会有所不同。各微机厂家在移植 MS-DOS 时, 要开发本公司微机使用的 IO. SYS。该 IO. SYS 主要包含如下的功能:

- ① 进行键盘输入和屏幕显示输出的 CON 设备的驱动程序;
- ② 管理时钟 CLOCK 设备的驱动程序;
- ③ RS-232C 的 AUX 设备等的字符类设备驱动程序;
- ④ 对磁盘进行输入/输出的块设备驱动程序。

IO. SYS 是设备驱动程序, 至少应包含 CON 设备驱动程序, CLOCK 设备驱动程序和一个磁盘驱动器 (块设备) 的设备驱动程序。设备驱动程序也可以由 CONFIG. SYS 文件进行安装, 但是由 IO. SYS 安装和由 CONFIG. SYS 安装, 其含义有所不同。例如, 在有的系统 (如 PC-9801 系列) 中, 从 MS-DOS 3. 3 起支持 SCSI 接口, 可以从 SCSI 的硬盘进行启动。但是, 使用 SCSI 硬盘的 MS-DOS 3. 1 也可以把硬盘的设备驱动程序安装在 CONFIG. SYS 中, 如果是这样, 就不能从 SCSI 硬盘建立系统, 若要从硬盘建立系统, 就必须将其驱动程序安装在 IO. SYS 中。

另外, 一部 RAM 盘, 如果要设置同软盘兼容, 可以从 RAM 盘进行启动, 这时一般需要改写引导扇区。RAM 盘的驱动程序如果是安装在 IO. SYS, 即使同软盘不兼容也可以从 RAM 盘进行启动。

如上所述,如果把设备驱动程序全部安装在 IO.SYS 中,由于用户不能简单地修改 IO.SYS,基于这一点考虑,全部安装在 IO.SYS 是必要的。但是这样做,可能因为 IO.SYS 中包含不必要的设备驱动程序,而使系统可以使用的存储器空间变小。

有些系列的微机(如 PC-9801 系列),初期的 DOS 是把打印机的 PRN 驱动程序和 RS-232C 的 AUX 驱动程序都安装在 IO.SYS 中,而 MS-DOS 3.3 版本却将其安装在 CONFIG.SYS 中。这样做,不致因 DOS 版本的升级而使 IO.SYS 过度地增大。对于不使用 PRN 和 AUX 驱动程序的用户来说,可以加大存储器的有效使用空间,这是把设备驱动程序安装在 CONFIG.SYS 中的可取之处。所以,对 DOS 来说,应把常用的最低限度的设备驱动程序安装在 IO.SYS,把其它不常用的驱动程序安装在 CONFIG.SYS 中。可以看到,把设备驱动程序安装在 CONFIG.SYS 的优点既有利于设备,也便于 DOS 版本的升级。

2. 作为 OS 核心部分的 MSDOS.SYS

MSDOS.SYS 作为 OS 的核心部分,如果版本相同,对所有机种来说基本上是一样的。MSDOS.SYS 通过调用 IO.SYS 的功能,对设备的输入/输出和时刻等进行管理。另外,存储器和磁盘等的管理也是由 MSDOS.SYS 进行的。MSDOS.SYS 是 OS 的功能中最基本的功能,应用程序可以通过系统调用(在 MS-DOS 中是 INT 21H)使用这些功能。因此,所谓的本体就是用软中断 INT 20H~3FH 进行调用的功能。其中,DOS 的 INT 21H 功能是 MSDOS.SYS 的最主要功能。

3. 作为用户接口的 COMMAND.COM

作为用户接口程序的 COMMAND.COM,又称作 SHELL,通常在用户输入命令的状态下,是用户同这个 COMMAND.COM 正在进行对话。DIR 和 TYPE 等内部命令是 COMMAND.COM 的功能。另外,批文件的启动也是由 COMMAND.COM 进行的。

批文件和 COM/EXE 文件不同,COM 文件和 EXE 文件是由汇编语言程序翻译成机器代码的二进制文件,批文件则是可以用 TYPE 命令在屏幕上显示的文本文件。

COM 文件和 EXE 文件,通常是用编辑程序建立源文件,在这个阶段它是文本文件,但是由于 DOS 不能直接解释汇编语言或 C 语言的文本程序,所以有必要使用汇编程序或编译程序将其变换成机器代码(MSDOS.SYS 可以执行的 COM 和 EXE 文件)。

这就是说,批文件不是 MSDOS.SYS 执行的文件,而是由 COMMAND.COM 直接进行解释并加以执行的文件。COMMAND.COM 对批文件的每一行进行解释,如果是自身具有的内部命令,则加以执行;如果是 COM 文件和 EXE 文件,则调用 MSDOS.SYS 的功能(INT 21H)加以执行。

1.1.4 从磁盘读取引导扇区后加载 OS 的步骤

1. 读取 IO.SYS 和 MSDOS.SYS

如图 1.8 所示,DOS 的 IPL 把位于磁盘数据区域前面的 IO.SYS 和 MSDOS.SYS 连续地读取到系统存储器,再把控制转移到 IO.SYS。DOS 的文件在磁盘上即使是不连续的存放,在进行输入/输出时也没有问题。但是,IO.SYS 和 MSDOS.SYS 是一个例外,必须在磁盘数据区域的前面连续进行存放。不能用 SYS.EXE 命令把已经存放有文件的磁盘设置成系统磁盘,实际上就是由于这个原因。

IO.SYS 和 MSDOS.SYS,除了带有文件的隐含属性和系统属性以外,是同其它的文件用完全相同的形式在磁盘上进行存放,只是为了简化引导扇区的处理才有连续存放的约定(这是

由于引导扇区的字节数有限，不能有复杂的引导程序)。

如果操作有错，插入了数据磁盘进行复位，这时会产生音响信息，并显示 “No system files” 等字样，有的系统还要催促用户插入系统盘。

2. IO.SYS 中的 SYSINIT 的作用

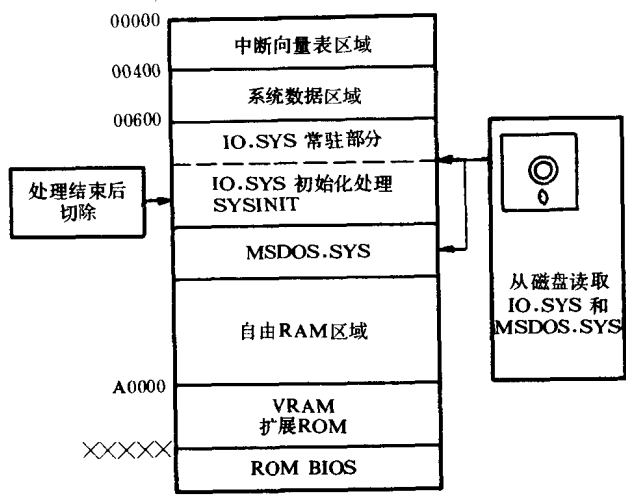


图 1.8 从磁盘中读入 IO.SYS 和 MSDOS.SYS

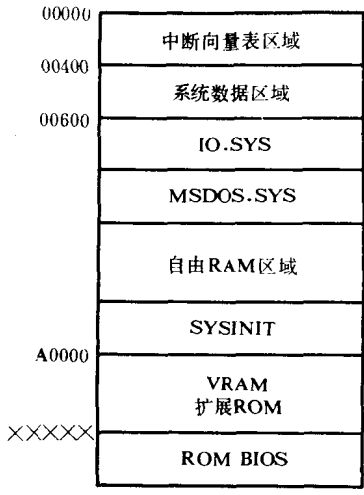


图 1.9 由 SYSINIT 进行的重新分配

在 IO.SYS 中，除了前述的设备驱动程序以外，还包含对自身进行初始化处理的程序，以及名为 SYSINIT，用于对系统进行初始化处理的程序。这两个初始化处理的程序，在系统建

立之后要进行切除(即从存储器上消失)。IO.SYS 先对自身进行初始化处理，然后再执行 SYSINIT 程序对系统进行初始化。

SYSINIT 把自身移到存储器的最高端，再把 MSDOS 移到 IO.SYS 常驻部分的后面，然后如图 1.9 所示，调用 MSDOS.SYS 的初始化处理程序。用 MSDOS.SYS 的初始化处理程序建立从 INT 20H 到 INT 3FH 的系统功能，并对要安装的设备驱动程序进行初始化等处理(这时才可以使用磁盘驱动器，利用 DOS 的功能)。

其后，如果在根目录上有 CONFIG.SYS 文件，SYSINIT 则将其打开，在 MSDOS 的后面分配磁盘缓冲区，和追加设备驱动程序等(如果没有 CONFIG.SYS 则进行缺省的设置)。

最后，如图 1.10 所示，如果有用 CONFIG.SYS 指定的 SHELL 命令，则 SYSINIT 按照指定的 SHELL 启动 COMMAND.COM，如果没有指定的 SHELL，则按缺省的

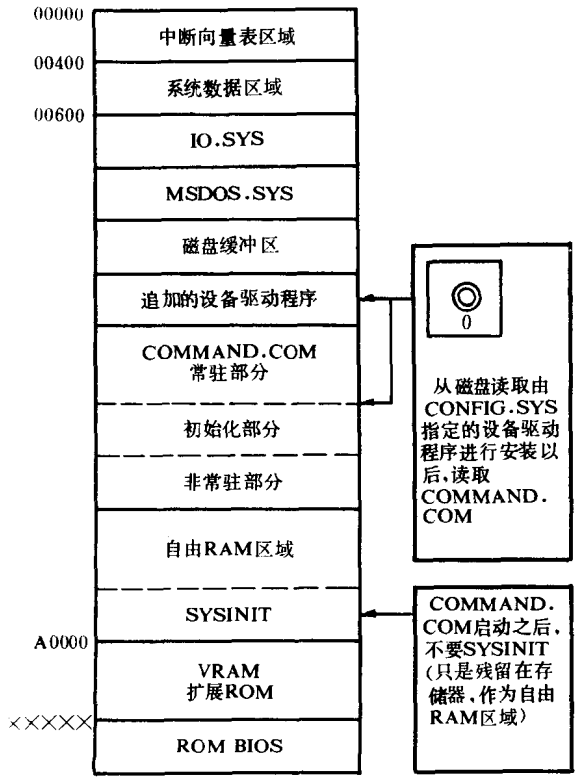


图 1.10 COMMAND.COM 的读入

SHELL 启动 COMMAND.COM。

1.1.5 执行 COMMAND.COM

COMMAND.COM 由常驻部分, 初始化部分和非常驻部分组成。下面对这三部分分别加以说明。

1. 初始化部分

初始化部分只调用一次, 完成下述操作以后即把控制返回到常驻部分:

- (1) 把非常驻部分移动到存储器的最高端;
- (2) 处理 CONFIG.SYS 的 SHELL 命令的选择参数;
- (3) 进行 AUTOEXEC.BAT 的启动处理。

这时, 可以看到 DOS 启动后的状态, 是显示如下的提示符, 等待命令的输入:

A>

图 1.11 所示为 DOS 启动以后的存储器的状态。

2. 非常驻部分

为了有效地利用存储器, 把 COMMAND.COM 分为常驻部分和非常驻部分。早期的微机系统, 由于安装的存储器容量较小(例如 IBM PC 机为 256KB), 把平时不常用的部分划归为非常驻部分, 如果启动的命令要请求使用自由 RAM 区域以上的存储器区域时, 可以使用非常驻部分驻留的区域。从现在的微机系统安装有 640KB 的主存储器的角度来看, 这种做法是不可取的, 应该把非常驻部分也归为常驻部分。

3. 常驻部分

常驻部分在命令启动以后对非常驻部分进行累加和检查, 以确认非常驻部分是否被破坏, 如果非常驻部分已被破坏, 则通过由环境变量 COMSPEC 指定的路径把 COMMAND.COM 的非常驻部分再一次装入。

COMMAND.COM 的常驻部分是作为 DOS 使用中的存储器块加以识别的, 而把非常驻部分作为未使用的存储器块的一部分进行对待(加载到最高位自由 RAM 区的最后)。因此, 如果 DOS (MSDOS.SYS)有存储器请求, 可以把非常驻部分作为未使用的存储器块分配给请求存储器的过程(或程序)。

1.1.6 内部命令

DOS 的命令有内部命令、外部命令和批文件三类。内部命令是包含在 COMMAND.COM 的非常驻部分内, 通常可以在显示 DOS 提示符的状态下加以执行。

外部命令如 DISKCOPY.COM 和 FORMAT.EXE 等是 COM 文件和 EXE 文件, 批文件是由这些内部命令和外部命令并行编写的文本文件。

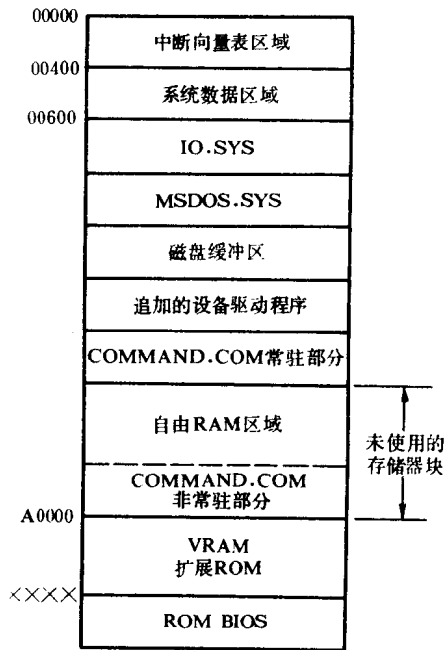


图 1.11 DOS 启动后的存储器的状态