MS-DOS 5.0
内核剖析

李凤华　周利华　赵丽松　编著

（中册）

西安电子科技大学出版社

MS—DOS 5.0 内核剖析

（中册）

李凤华　周利华　赵丽松　编著

西安电子科技大学出版社
1992

(陕)新登字 010 号

内 容 简 介

作者对 DOS 操作系统的最新版本 MS－DOS 5.0 的核心程序进行了完整、详细的分析，同时结合从事 DOS 操作系统开发和改造的实践经验，全面地介绍了 DOS 操作系统的设计思想及 MS－DOS 5.0 的具体实现，为读者提供了 MS－DOS 5.0 内核的完整信息。

《MS－DOS 5.0 内核剖析》分上、中、下三册出版，全书分为十一个章节。它主要介绍了 DOS 操作系统的结构和设计思想、配置系统、引导过程、设备管理、内存管理、文件系统、进程管理、DOS 功能调用、DOS 核心文件的编程环境，并提供了 MS－DOS 5.0 的 IO.SYS 和 MSDOS.SYS 两个系统文件的源程序注释清单。第八章给出了包括内部保留(未公开)功能调用在内的所有 DOS 功能调用的详细信息，同时为了方便读者阅读 IO.SYS 和 MSDOS.SYS 源程序，书中还提供了这两个核心文件的源程序索引。

本书适合于计算机系统软件开发人员、微机开发和应用人员参考，也可作为大专院校“操作系统”、“系统程序设计”、“计算机外设联机技术”等有关课程的教学参考书，本书也可供从事微机的加/解密和计算机病毒防治等方面的技术人员参考。

MS－DOS 5.0 内核剖析(中册)
李凤华　周利华　赵丽松　编著
责任编辑　殷咸安

西安电子科技大学出版社出版发行
西安电子科技大学印刷厂印刷
新华书店经销
开本 787×1092　1/16　印张 27 4/16　字数 834 千字
1992 年 10 月第 1 版　1992 年 10 月第 1 次印刷　印数 1—4 000

ISBN7－5606－0213－4/TP・0076　　全套定价：80.00 元

目　录

第十章　IO.SYS 源程序

本章详细地介绍 MS－DOS 5.0 的 IO.SYS 系统文件的各个组成部分的内容和功能，并给出了 IO.SYS 源程序注释清单。这些资料是读者深入理解和掌握 MS－DOS 5.0 核心所需要的。

IO.SYS 系统文件的主要功能是进行 DOS 的系统引导和初始化、处理 CONFIG.SYS 文件和配置系统、提供系统基本配置的设备的设备驱动程序，它是 MSDOS.SYS 系统文件的基础。

§10.1　IO.SYS 的编程基础

IO.SYS 系统文件是建立在 ROM BIOS 的基础之上，它的运行环境是以下几个方面：

(1)硬盘分区信息表；

(2)引导记录提供的 BPB 参数块(对于 MS－DOS 5.0，引导记录提供扩充 BPB 参数块)；

(3)由 ROM BIOS 提供的软盘参数表和硬盘参数表；

(4)由 ROM BIOS 提供的中断服务程序——INT 10H～INT 1AH。

§10.2　IO.SYS 的组成

IO.SYS 系统文件由 Loader 模块、IO1 模块、IO2 模块和 IO3 模块组成(如表 10.1 所示)，有关这四个模块的各自作用将在下一节详细介绍。IO.SYS 系统文件的这种结构是根据如下思想编程的：

表 10.1

模块名	长度(字节数)	相对起始段内偏移	在源程序中的位置(行号)
Loader	05A6H	0000H	1～518
IO1	2570H	0000H	519～3817
IO2	1A60H	0030H	3818～7889
IO3	3D20H	0000H	7890～15417
IO.SYS	33430(字节)		

(1)Loader 是 IO.SYS 系统文件的装入程序；

(2)IO1 模块由驻留在常规内存低端的接口程序和数据区、以及系统初始化程序 SysInt－I 组成；

(3)IO2 模块是设备驱动程序集，它允许安装在常规内存(如图 5.7)或安装在 HMA(如图 5.9)；

(4)IO3 模块负责处理系统配置文件 CONFIG.SYS。

§10.3 IO.SYS 的各个模块的功能

10.3.1 Loader 模块

Loader 模块长 05A6H 字节，它由引导记录读入并存放在常规内存 0000:0070H 处。有关引导盘的信息(如引导盘的物理驱动器号、介质描述符和数据区的起始扇区号、IO.SYS 和 MSDOS.SYS 系统文件的目录项等)是由引导记录提供的。当 CPU 的执行控制权由引导记录转到 Loader 模块时，Loader 模块负责装入 IO.SYS 系统文件的其它三个模块——IO1 模块、IO2 模块和 IO3 模块，装入的过程如图 10.1 所示。

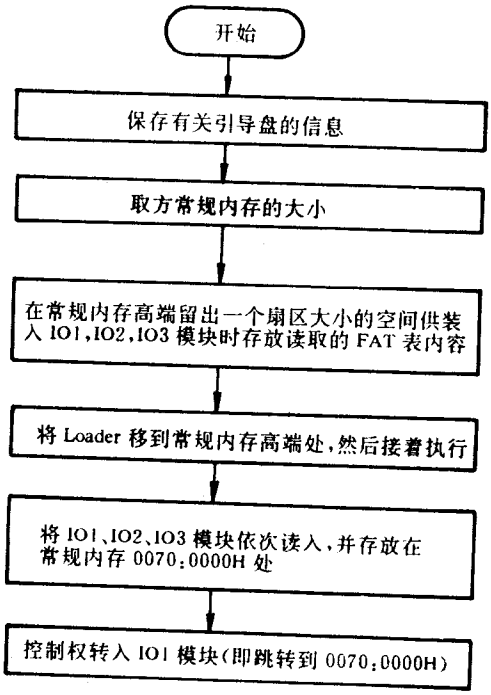

图 10.1 Loader 模块的工作过程

Loader 模块的作用如下：

(1)通过 Loader 模块安装 IO.SYS 系统文件的主体程序，使得 IO.SYS 的文件长度对引导记录透明，从而使得 MS—DOS 5.0 的引导记录能在以后的版本中保持不变；

(2)MS—DOS 5.0 允许 IO.SYS 和 MSDOS.SYS 系统文件的内容存于引导盘数据区的任意处，而不是引导盘数据区的起始位置。MS—DOS 5.0 只要求根目录的头两个目录项分别对应于 IO.SYS 和 MSDOS.SYS 即可。Loader 模块使得安装 MS—DOS 5.0 更方便。

MS—DOS 5.0 允许在不破坏系统盘数据的情况下安装，它的这一新特点正是由 Loader 模块来保证的。Loader 模块在 IO1 模块获得控制之后就失去了作用，并终止存在和被覆盖。

10.3.2 IO1 模块

IO1 模块长 2570H 字节，它由 Loader 读入并存放在常规内存 0070:0000H 处。由引导记录提供的有关引导盘的信息通过 Loader 模块转传给 IO1 模块。

一、IO1 模块的组成

IO1 模块由以下 7 部分组成：

(1)DOS 提供的所有设备驱动程序对应的设备驱动程序标题(如 CON、AUX、PRN、CLOCK $、块设备、COM1、LPT1、LPT2、LPT3、COM2、COM3、COM4，如图 4.1 所示。在源程序中的行号为 604～708)和调用 IO2 模块提供的设备驱动程序的接口程序(在源程序中的行号为 940～1014)；

(2)提供了 INT 1BH、INT 29H、INT 6CH 中断服务程序(在源程序中的行号分别为 931～939、1015～1034、1682～1939)；

(3)INT 13H 中断接管程序(在源程序中的行号为 1035～1071、1365～1590)；

(4)A20 地址线管理程序(在源程序中的行号为 1073～1140)；

(5)INT 19H 中断接管程序(在源程序中的行号为 1141～1191)，它是为保证系统重启动的环境而设置的(因为当 DOS 启动后，它接管或更改了 INT 10H、INT 13H、INT 15H、INT 19H、INT 1BH 中断服务程序；以外，当 CONFIG.SYS 文件中指定了 stacks 系统配置命令后，它还接管了 INT 02H、INT 08H～INT 0EH、INT 70H、INT 72H～INT 77H 中断服务程序，以便给它们提供动态堆栈)；

(6)IO1 模块的驻留部分及 IO2 模块的数据区和 BDPB 数据结构所需的内存；

(7)系统初始化程序 SysInt－Ⅰ(在源程序中的行号为 1941～3817)。

其中，1～6 项为 IO1 模块的驻留部分，但它们的驻留长度与具体的机器相关(即与微机系统配置的软硬盘数、硬盘分区数、机器型号、ROM BIOS 版本号等有关)，具体计算方法如下(参见源程序 2531～2625)。

二、SysInt－Ⅰ的功能

SysInt－Ⅰ是 DOS 操作系统的初始化程序，它具体建立了 DOS 内核(设备驱动程序和 MSDOS.SYS 系统文件的基本运行环境，分析 DOS 操作系统的读者从此开始逐步地深入将会了解 DOS 操作系统是如何启动的、以及如何管理磁盘等设备的。当 SysInt－Ⅰ执行完后，CPU 的控制权就转到 IO3 模块的系统初始化程序 SysInt－Ⅱ。SysInt－Ⅰ具体地完成以下五个方面的工作：

(1)确定并保存系统配置(包括保护和重新设置中断向量)；

(2)初始化串并端口；

(3)建立软硬盘的 BDPB 数据结构；

(4)确定 IO1 模块的驻留长度和 MSDOS1 模块在常规内存低端的最终位置；

(5)读取 MSDOS.SYS 系统文件内容到常规内存 083F:0000H 处(083FH＝0070H＋7CF0H/16，这里 7CF0H 是 IO.SYS 系统文件的 IO1、IO2 和 IO3 模块的长度和，即 MSDOS.SYS 系统文件暂时存于 IO3 模块之后)。

三、IO1 模块的驻留长度

IO1 模块的驻留长度(KL)是按如下算法计算的：

(1) 若机器配置了硬盘，则转到(3)。

(2) 若 ROM BIOS 不支持软盘的 Change Line 功能，则 KL＝08FEH，否则，KL＝0918HH。

(3) 若机器为 AT 档次机器、ROM BIOS 版本号为“01/10/84”，则 KL＝KL＋(140CH－12DEH)＝KL＋012EH；若机器为 COMPQA，ROM BIOS 是 1986 年 8 月 4 号以前开发的，则 KL＝KL＋(142DH－140CH)＝KL＋0021H。

(4) 若机器有实时时钟，则 KL＝KL＋(1509H－142EH)＝KL＋00DCH

(5) 若 INT 15H 中断服务程序支持 AX＝4101H 功能，则 KL＝KL＋(16D4－150A)＝KL＋01CBH。

根据上述算法，IO1 模块的驻留长度范围为 08FEH～(12DEH＋012EH＋00DCH＋01CBH)，即 08FEH～16B3H。

10.3.3 IO2 模块

IO2 模块长 1A60H 字节，它是 DOS 操作系统提供的设备驱动程序集，由 CON、PRN、AUX、CLOCK $ 和块设备等设备驱动程序组成(如表 10.2 和表 10.3a～10.3e 所示，其中表 10.3a 对应于 CON 设备驱动程序、表 10.3b 对应于 PRN 设备驱动程序、表 10.3c 对应于 AUX 设备驱动程序、表 10.3d 对应于 CLOCK $ 设备驱动程序、表 10.3e 对应于块设备驱动程序)；此外，它包含了 INT 13H 中断接管程序(在源程序中的行号为 6995～7344)和 INT 2FH 中断服务程序(在源程序中的行号为 6686～6840)。IO2 模块全部驻留内存，它的编程起始地址偏移为 0030H，这是因为当 IO2 模块安装在 HMA 起始位置时，在 8086/8088 指令体系下，访问 HMA 是通过使能 A20 地址线并利用段地址 0FFFFH＋段内偏移来实现的，因而 0FFFFH:0010H 才是 HMA 的起始地址；此外，HIMEM.SYS 等扩充内存管理程序使用 HMA 的起始 32 个字节来存放它们的安装标志和使用扩充内存的管理信息等。

表 10.2

设备名	起始行号	结束行号
CON	4002	4145
PRN	4146	4324
AUX	4325	4465
CLOCK $	4466	4638
块设备	4640	7882

表 10. 3a

命令码(H)	起始行号	结束行号
4	4002	4056
5	4057	4109
17	4130	4145
8 或 9	4110	4129

表 10. 3b

命令码(H)	起始行号	结束行号
4	4146	4156
8 和 9	4157	4195
0A	4196	4236
10	4237	4272
13	4273	4302
19	4303	4324

表 10. 3c

命令码(H)	起始行号	结束行号
4	4325	4347
5	4366	4389
7	4418	4429
8 和 9	4430	4454
0A	4390	4404

表 10. 3d

命令码(H)	起始行号	结束行号
4	4562	4617
8 和 9	4509	4561

表 10. 3e

命令码(H)		起始行号	结束行号
0		7346	7356
1		4687	4745
2		4775	4809
4、8 和 9		5186	5210
0D		5129	5143
0E		5144	5160
0F		5161	5173
13（子功能码(H)）	40	5837	5927
	41	6073	6171
	42	5928	6028
	46	6487	6531
	47	6597	6612
	60	5797	5836
	61	6073	6171
	62	6029	6072
	66	6457	6486
	67	6580	6596
	68	6643	6684
17 和 18		6391	6427
19		6613	6642

CON 设备驱动程序是标准的显示器和键盘的驱动程序，所有键盘输入和以电传方式显示服务都是由它提供的。但是显示器和键盘是两个独立的设备，应用程序（包括系统程序）是由 CON 设备驱动程序从键盘上输入一个字符，然后将该输入字符输出到 CON 设备驱动程序，显示在显示器上。用户所见到的“输入并回显”就是分上述两步完成的。

PRN 设备驱动程序是标准并行打印机设备驱动程序，它在程序控制下发送单字节数据，不支持中断驱动式的打印。设备名 LPT1 等于 PRN，它们都对应于并行端口 1；LPT2 和 LPT3 分别对应于并行端口 2 和 3。

AUX 设备驱动程序是标准异步通讯（即串行通讯）设备驱动程序，它只能在程序控制下发送或接收字节数据，也不支持中断驱动式的数据传输和 XON/XOFF 这样的协议。设备名 COM1 等于 AUX，它们都对应于串行端口 1；COM2、COM3 和 COM4 分别对应于串行端口 2、3 和 4。

CLOCK $ 设备驱动程序提供了标准时间服务，DOS 内核通过它取出或设置当前日期或时间。

块设备驱动程序提供了磁盘服务，用户通过逻辑驱动器符来访问具体的磁盘介质。

10.3.4 IO3 模块

IO3 模块长 3D20H 字节，它的主要内容是系统初始化程序 SysInt－Ⅱ。SysInt－Ⅱ负责处理 CONFIG.SYS 系统配置文件，为 DOS 设置各种数据结构和其它工作环境。当位于 IO1 模块里的 SysInt－Ⅰ执行完后，CPU 的控制权就转到这里。

一、IO3 模块的组成

IO3 模块由以下十个部分组成：

(1)INT 02H、INT 08H～INT 0EH、INT 70H、INT 72H～INT 74H、INT 76H～INT 77H 的中断接管程序，它们为各自的中断提供动态堆栈。当 CONFIG.SYS 中有 stacks 系统配置命令，那么 SysInt－Ⅱ将驻留这部分程序(在源程序中的行号为 7943～8292)；

(2)系统初始化程序 SysInt－Ⅱ(在源程序中的行号为 8293～8819)；

(3)负责安装 IO2 模块和 MSDOS2 模块的程序(在源程序中的行号为 8820～9103)；

(4)DOS 启动时使用的 HMA 管理程序(在源程序中的行号为 9104～9263)；

(5)建立逻辑驱动器的当前目录结构(CDS)的子程序(在源程序中的行号为 9297～9375)；

(6)在作为 DOS 数据段的第一个内存块中为 DOS 内核建立 SFT 表数组、盘缓冲区、CDS 数组、动态堆栈等各种数据块的子程序(在源程序中的行号为 9376～9773)；

(7)处理 install 系统配置命令，并在 DOS 启动过程中安装 DOS 内存驻留程序的程序(在源程序中的行号为 9774～9925)；

(8)建立动态堆栈的控制块，并安装 INT 02H、INT 08H～INT 0EH、INT 70H、INT 72H～INT 74H、INT 76H～INT 77H 的中断接管程序(在源程序中的行号为 10013～10481)；

(9)取系统配置命令的参数(开关参数、整数参数、开关状态、关键字串、文件名说明串)程序，它在源程序中的行号为 10482～11752；

(10)CONFIG.SYS 允许的系统配置命令的参数说明信息和处理 CONFIG.SYS 的程序。它在源程序中的行号分别为 11753～12069、12071～15417。

二、SysInt－Ⅱ的执行过程

在 MS－DOS 5.0 中，IO2 模块、MSDOS2 模块和盘缓冲区可以安装在 HMA；在 80386 或更高档微机系统中，允许在 UMB 中安装某些设备驱动程序和其它一些应用程序；还允许使用 install 系统配置命令在 DOS 启动时安装 DOS 内存驻留程序。MS－DOS 5.0 的这些特点决定了对 CONFIG.SYS 的处理必须按一定的先后次序，CONFIG.SYS 的处理过程与先前旧的 DOS 版本有很大的差别。SysInt－Ⅱ的执行过程如图 10.2 所示，它反映了 SysInt－Ⅱ所完成的工作。

在分析 SysInt－Ⅱ程序时，应注意下列几个方面的问题：

1. 转换 CONFIG.SYS 的文件内容

完成 CONFIG.SYS 文件内容转换的程序为 14309～14537，它具体的转换方法和所完成的工作如下：

(1) 小写字符转换成大写字符；

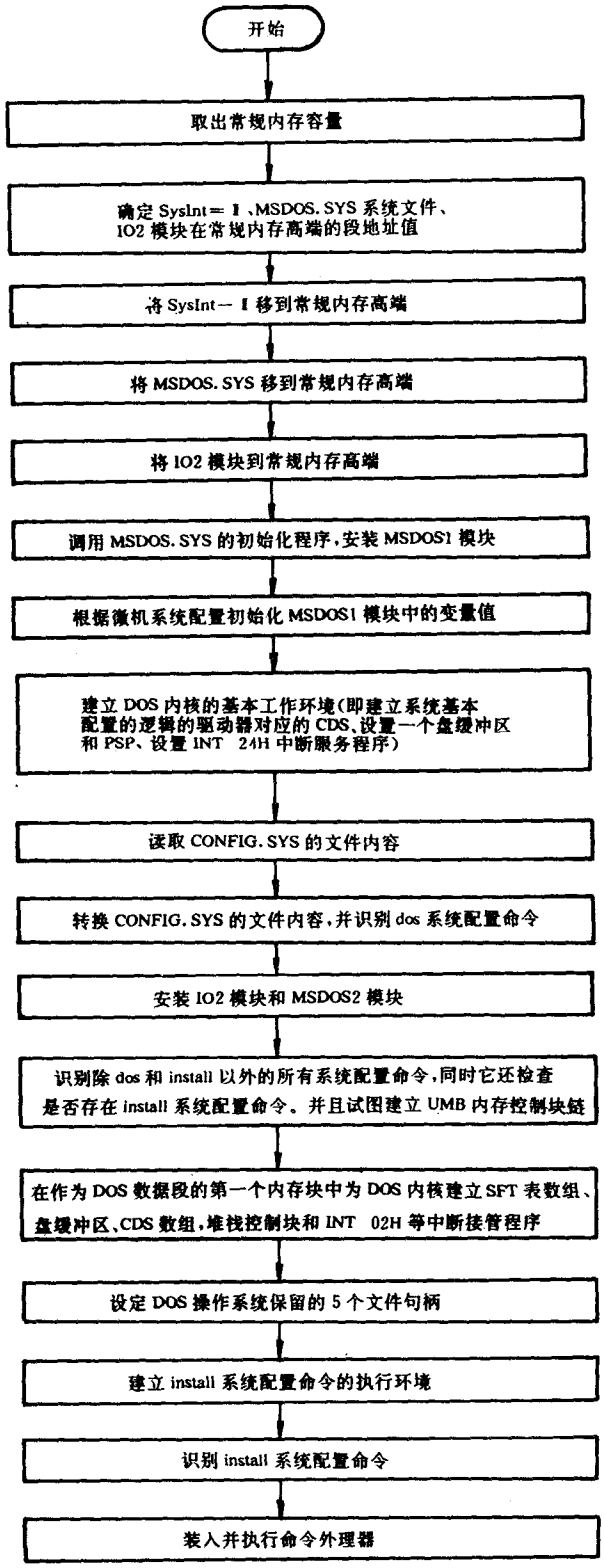

图 10.2 SysInt— Ⅱ 的执行过程

(2) 丢掉每个 CONFIG.SYS 命令行的所有起始参数分隔符；

(3) 系统配置命令名用对应的缩写关键字符取代，如表 10.4 所示：

表 10.4

系统配置命令名	缩写字符	系统配置命令名	缩写字符	系统配置命令名	缩写字符
break	C	dos	H	multitrack	M
buffers	B	drivparm	P	rem	0
comment	Y	fcbs	X	shell	S
country	Q	files	F	stacks	K
device	D	install	I	switches	1
devicehigh	U	lastdrive	L		

(4) 识别 comment 系统配置命令，并丢掉在它之后的所有注释字符串；

(5) 命令行的结束符只使用换行符(0AH)，而回车符(0DH)被丢掉；

(6) 结束符(00H)代替文件名说明串后的参数分隔符；

(7) 用"Z"字符和换行符(0AH)代替整个非法的命令行；

(8) 给 devicehigh 命令行追加一个空参数项，以确保能正确地处理 devicehigh 系统配置命令的两种格式。

2. dos 系统配置命令

dos 系统配置命令的使用已在第二章中进行了介绍。由于 dos 系统配置命令是用于指定 DOS 的 IO2 模块、MSDOS2 模块和盘缓冲区的安装位置(即它们是被 SysInt－Ⅱ安装在常规内存低端或 HMA 中)；同时，它还指定设备驱动程序是否被安装在 UMB 中。鉴于 dos 系统配置命令的特殊作用，它必须首先被识别和处理。

3. 扩充内存管理程序 HIMEM.SYS

HIMEM.SYS 是扩充内存管理程序，它管理对扩充内存(包括高内存区 HMA)的使用。当 DOS 的 IO2 模块、MSDOS2 模块和盘缓冲区欲安装在 HMA 时，除 CONFIG.SYS 文件中有"dos=high"语句之外，还应有安装 HIMEME.SYS 的 device 命令(它必须安放在其它所有使用扩充内存的程序或设备驱动程序的 device 命令之前)。有关 HIMEM.SYS 的使用方法请参见第二章。

4. 高内存块管理程序 EMM386.EXE

EMM386.EXE 使得 DOS 操作系统和应用程序能够使用 80386 或更高档微机的高内存块(UMB)，同时，EMM386.EXE 还允许 SysInt－Ⅱ将设备驱动程序装入 UMB。使用 EMM386.EXE 之前必须安装 HIMEM.SYS。

EMM386.EXE 允许存取 UMB，MS－DOS 5.0 通过它可以分配和使用 UMB。有关 EMM386.EXE 的使用方法请参见第二章、第五章。

5. install 系统配置命令

DOS 操作系统的许多系统扩充(如 FASTOPEN、KEYB、NLSFUNC 和 SHARE 等)是通过 DOS 内存驻留程序提供的，它允许用户根据使用的需要有选择地过安装和使用。若用户不需要某一功能，就可以不安装对应的系统扩充命令，从而为用户留出更多的常规内存空间。

DOS 操作系统的这种功能分配使得 DOS 的核心文件尽可能地短小，使用户可以得到最佳的存贮——性能比或存贮——功能比。

在 DOS 4.0 以前，DOS 的这些系统扩充由 AUTOEXEC.BAT 或由用户通过命令行装入。在 DOS 启动时，这种处理方式使得设备驱动程序无法使用这些扩充，为了克服这个缺点，从 DOS 4.0 开始增加了 install 系统配置命令。

install 系统配置命令在 SysInt－Ⅱ执行过程中装入 DOS 内存驻留程序（如 SHARE.EXE、KEYB.COM 等），它不为装入程序生成环境块，因而它比从 AUTOEXEC.BAT 批文件或命令行下装入程序占用的内存少。

由于 install 系统配置命令装入程序时命令处理器尚未被装入运行，因而它不能装入要求 COMMAND.COM 管理临界错误的程序。另外，它也不能保证需要使用环境变量的装入程序的正常运行。

6. stacks 系统配置命令

stacks 系统配置命令允许用户指定多达 64 个动态堆栈，这些动态堆栈是提供给 INT 02H、INT 08H～INT 0EH、INT 70H、INT 72H～INT 74H、INT 76H～INT 77H 中断服务程序使用的。DOS 操作系统通过接管上述硬中断的中断服务程序方式来提供动态堆栈服务的。当上述硬中断发生时，DOS 就从用户定义的动态堆栈中分配一个堆栈，但当 stacks＝n，s 的 n 值指定为 0 时，DOS 将不接管上述硬中断对应的中断服务程序，此时，每个运行程序必须有足够的栈空间供硬中断各自对应的中断服务程序使用。

（1）stacks 系统配置命令执行后需要的内存容量

①INT 02H、INT 08H～INT 0EH、INT 70H、INT 72H～INT 74H、INT 76H～INT 77H 中断接管程序的长度为 0267H 字节；

②动态堆栈需要的内存字节数＝(8＋s)＊n，其中，8 为每个动态堆栈的控制块所需的字节数；

③堆栈对应的子段控制块的字节数＝16。

整个 stacks 系统配置命令执行后所需的内存是上述三项之和。

（2）堆栈控制块(SCB)

动态堆栈控制块(Stack Control Block，简称为 SCB)的数据结构如表 10.5 所示，它保存了堆栈的使用状态、先前的堆栈指针寄存器值和它对应的动态堆栈的指针值。

表 10.5

```
SCB                 Struc
SCB_UseMark         DB   ?      ;动态堆栈的使用标志，其值的意义如下：
                                    00：动态堆栈空闲
                                    01：动态堆栈已被分配
                                    03：动态堆栈被破坏，该动态堆栈不再被分配
                    DB   ?      ;未使用
SCB_SPValue         DW   ?      ;保存先前的 SP 寄存器值
SCB_SSValue         DW   ?      ;保存先前的 SS 寄存器值
SCB_AddrOfStack     DW   ?      ;指向对应的动态堆栈的栈顶(它的值为新的 SP 寄存器值)
SCB                 Ends
```

§10.4 IO.SYS 源程序注释清单

10.4.1 IO.SYS 源程序的编译和连接方式：

IO.SYS 源程序的编译和连接方法如下

C>MASM LOADER;↙

C>LINK LOADER;↙

C>EXE2BIN LOADER.EXE LOADER.COM;↙

C>MASM IO1;↙

C>MASM IO3;↙

C>LINK IO1+IO3;↙

C>EXE2BIN IO1.EXE IO1.COM↙

C>COPY/B LOADER.COM+IO1.COM IO.SYS↙

10.4.2 BIO.STR 的源程序清单

```
CR                          EQU     0DH
LF                          EQU     0AH
TAB                         EQU     9
SysInt_11_OFS               EQU     267H
SysInt_11_SEG               EQU     46DH
ROM_SEG                     EQU     0F000H
BIO_SEG                     EQU     70H
BDPB_Len                    EQU     64H
BPB_Len1                    EQU     19H
BPB_Len2                    EQU     1FH
DOS_0024                    EQU     24H
DOS_008C                    EQU     8CH
FileNameLen                 EQU     0BH
;
BDPB                        STRUC
  BDPB_NextPtr_OFS          DW      -1          ;0
  BDPB_NextPtr_SEG          DW      BIO_SEG     ;2
  BDPB_DriveNumber          DB      ?           ;4
  BDPB_LogicalNumber        DB      ?           ;5
;Start of Build BPB
  BDPB_BytesPerSector1      DW      512         ;6
  BDPB_SectorsPerCluster1   DB      ?           ;8
  BDPB_ReservedSectors1     DW      1           ;9
  BDPB_NumberOfFAT1         DB      2           ;B
  BDPB_TotalOfRootDir1      DW      ?           ;C
```

```
    BDPB_TotalSector1           DW      ?               ;E
    BDPB_MediaByte1             DB      ?               ;10
    BDPB_SectorsPerFAT1         DW      ?               ;11
    BDPB_SectorsPerTrack1       DW      ?               ;13
    BDPB_NumberOfHead1          DW      ?               ;15
    BDPB_HiddenSectors1         DD      0               ;17
    BDPB_BigTotalSector1        DD      0               ;1B
;End of Build BPB
    BDPB_SizeFlag               DB      0               ;1F
    BDPB_DeviceOpen             DW      0               ;20
    BDPB_DeviceType             DB      3               ;22
    BDPB_Attr_Status            DW      20H             ;23
    BDPB_NumberOfCylinder       DW      28H             ;25
;Start of Default BPB
    BDPB_BytesPerSector2        DW      512             ;27
    BDPB_SectorsPerCluster2     DB      ?               ;29
    BDPB_ReservedSectors2       DW      ?               ;2A
    BDPB_NumberOfFAT2           DB      ?               ;2C
    BDPB_TotalOfRootDir2        DW      ?               ;2D
    BDPB_TotalSector2           DW      ?               ;2F
    BDPB_MediaByte2             DB      ?               ;31
    BDPB_SectorsPerFAT2         DW      ?               ;32
    BDPB_SectorsPerTrack2       DW      ?               ;34
    BDPB_NumberOfHead2          DW      ?               ;36
    BDPB_HiddenSectors2         DD      0               ;38
    BDPB_BigTotalSector2        DD      0               ;3C
;End of Default BPB
    BDPB_ReservedFeild          DB      6 Dup(0)        ;40
    BDPB_CylinderOperated       DB      -1              ;46
    BDPB_Clock1orCylinderFlag   DW      -1              ;47
    BDPB_Clock2orCylinder       DW      -1              ;49
    BDPB_VolumeID               DB      ' NO NAME',0    ;4B
    BDPB_SerialNumber           DD      0               ;57
    BDPB_FileSystemType         DB      ' FAT12' ,0     ;5B
BDPB                            ENDS
;
BPB                             STRUC
    BPB_BytesPerSector          DW      512             ;0
    BPB_SectorsPerCluster       DB      ?               ;2
    BPB_ReservedSector          DW      ?               ;3
    BPB_NumberOfFAT             DB      ?               ;5
    BPB_TotalOfBootDir          DW      ?               ;6
    BPB_TotalSector             DW      ?               ;8
```

```
    BPB_MediaByte           DB      ?               ;A
    BPB_SectorsPerFAT       DW      ?               ;B
    BPB_SectorsPerTrack     DW      ?               ;D
    BPB_NumberOfHead        DW      ?               ;F
    BPB_HiddenSectors       DD      ?               ;11
    BPB_BigTotalSector      DD      0               ;15
BPB                         ENDS
;
FPT                         STRUC
    FPT_ControlByte1        DB      ?               ;0
    FPT_ControlByte2        DB      ?               ;1
    FPT_Motor_DelayTime     DB      ?               ;2
    FPT_SectorSize          DB      ?               ;3
    FPT_SectorsPerTrack     DB      ?               ;4
    FPT_ByteBetweenSector   DB      ?               ;5
    FPT_BytesPerSector      DB      ?               ;6
    FPT_LengthOfGapInFMT    DB      ?               ;7
    FPT_WriteByteInFMT      DB      ?               ;8
    FPT_HeadLoadTime        DB      ?               ;9
    FPT_StartupWaitTime     DB      ?               ;A
FPT                         ENDS
;
FDT                         STRUC
    FDT_FileName            DB      8  Dup(' ')     ;0
    FDT_FileType            DB      3  Dup(' ')     ;8
    FDT_FileAttribute       DB      ?               ;B
    FDT_Reserved            DB      10 Dup(0)       ;C
    FDT_TimeLastUpdated     DW      ?               ;16
    FDT_DateLastUpdated     DW      ?               ;18
    FDT_StartingCluster     DW      ?               ;1A
    FDT_FileLength          DD      0               ;1C
FDT                         ENDS
;
MCB                         STRUC
    MCB_Location            DB      ?               ;0
    MCB_ProcessID           DW      ?               ;1
    MCB_AllocationAmount    DW      ?               ;3
    MCB_Reserved            DB      3 Dup(0)        ;5
    MCB_ProgramName         DB      8 Dup(?)        ;8
MCB                         ENDS
;
DriverRequestHeader         STRUC
    DRH_Length              DB      ?               ;0
```