

计算机实用程序 与使用技巧

上册

茅 晋 李 湍 编
徐天祯 李健伟



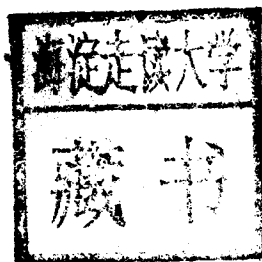
清华大学出版社

TP31
MJ/1-1

计算机实用程序 与使用技巧

上册

茅 晋 李 湍 编
徐天祯 李健伟



清华大学出版社

0033671

(京)新登字 158 号

内 容 简 介

《计算机实用程序与使用技巧》分上、下两册。主要内容包括操作系统、C 语言、数据库、工具软件使用技巧、病毒防治技术、加密与解密、汉字处理、图形与图象、显示与打印、实用程序、软件评测和网络等十二章,上册为前六章。书中所有程序是众多计算机编程人员和用户经验与智慧的结晶,具有很强的实用性。

本书可供广大计算机用户和程序员参阅。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

计算机实用程序与使用技巧 上册/茅晋等编. —北京:清华大学出版社,1996
ISBN 7-302-01959-2

I. 计… II. 茅… III. 程序系统-基本知识 IV. TP31

中国版本图书馆 CIP 数据核字(96)第 18734 号

出版者:清华大学出版社(北京清华大学校内,邮编 100084)

印刷者:北京市清华园胶印厂

发行者:新华书店总店北京科技发行所

开 本:787×1092 1/16 印张:31.75 字数:760 千字

版 次:1996 年 10 月第 1 版 1996 年 10 月第 1 次印刷

书 号:ISBN 7-302-01959-2/TP·903

印 数:0001—5000

定 价:36.00 元

目 录

第1章 操作系统

1.1 MS DOS 重入分析	(1)
1.2 巧用 DOS 中的 I/O 重定向命令	(5)
1.3 DOS 内部命令	(8)
1.4 在批处理文件中巧用 DOS 环境变量	(26)
1.5 DOS 文件的 FCB 管理机制分析	(29)
1.6 DOS 下人机界面的处理技术与磁盘文件管理	(34)
1.7 巧用 DOS 设备名 NUL、PRN、COV	(39)
1.8 在 DOS 状态下随意扩充屏幕和键盘的有效方法	(39)
1.9 固化 DOS 的一种实用技术	(41)
1.10 多路中断 INT 2FH 的分析与应用	(42)
1.11 一个 DOS 尚未公布的 INT 21H 的 52H 号功能	(50)
1.12 MS DOS 5.0 的新增功能	(52)
1.13 MS DOS 5.0 新增命令 DOSKEY	(57)
1.14 利用 DOS 5.0 优化 SPDOS 5.1 的运行环境	(63)
1.15 MS DOS 5.0 系统的备份及恢复	(64)
1.16 DOSKEY 应用技巧	(65)
1.17 MS DOS 5.0 使用经验	(67)
1.18 如何在 MS DOS 5.0 下运行金山汉字系统	(69)
1.19 MS DOS 6.0 透视	(70)
1.20 DOS 6.0 的奥秘	(72)
1.21 关于使用 MS DOS 6.0 中 Double Space 的体会	(75)
1.22 如何在 DR DOS 6.0 下使用 2.13H	(77)
1.23 在 DOS 6.0 中实现 BACKUP 功能	(78)
1.24 DOS 6.0 的 CONFIG 技术	(79)
1.25 安装运行 WMDOS 5.0 和 WMDOS 6.0 的方法	(81)
1.26 在 SPDOS 6.0F 下用 WPS	(84)
1.27 DRIVE.SYS 设备命令的几种用法	(84)
1.28 用 XENIX 系统修复 DOS 系统硬盘与文件	(85)
1.29 一个 XENIX 系统常见问题的处理办法	(86)
1.30 XENIX 下磁带备份和恢复的实用程序	(87)
1.31 XENIX 系统在任意目录下多个文件的加密	(91)
1.32 多操作系统共享硬盘技巧	(93)
1.33 DM 管理下 MS DOS 与 XENIX 共享硬盘的 XENIX 系统的安装	(94)

1.34	XENIX 系统和 DOS 系统共享硬盘	(94)
1.35	用 XENIX 应急引导盘恢复损坏的根文件系统	(95)
1.36	Windows 3.1 使用中的几个技巧	(97)
1.37	Windows 3.1 的主要特点及功能	(101)
1.38	Windows 3.1 中文版的特色	(104)
1.39	Windows 启动位图的修改	(106)
1.40	恢复在 Windows 中被误删的 MS DOS Prompt	(109)
1.41	Windows NT——新一代的操作系统	(109)

第2章 C 语言

2.1	C 语言学习中需注意的几个问题	(115)
2.2	实现 C 语言宏代换功能	(120)
2.3	利用 C 语言完善 DOS 的 TYPE 命令	(123)
2.4	功率谱估计的 C 语言实现	(125)
2.5	C 语言中文件通配符的处理	(126)
2.6	用 C 程序自动调整文本文件	(127)
2.7	利用 C 语言实现音乐简谱的识别与演奏	(129)
2.8	背景音乐的原理及其实现	(132)
2.9	Turbo C 键盘宏定义	(134)
2.10	大数字实时时钟的 C 语言实现	(137)
2.11	Turbo C 和 C++ 两类文件函数的区别和使用	(139)
2.12	用 MS-FORTRAN 直接读写内存	(143)

第3章 数据库

3.1	dBASE ■ 程序逻辑框图的自动生成	(145)
3.2	dBASE ■ 屏幕格式的自动生成	(149)
3.3	dBASE ■ 菜单源程序自动生成方法	(150)
3.4	dBASE ■ 修改文件属性的方法	(153)
3.5	认可方式的程序设计	(155)
3.6	一程序多报表的自动建立、修改与输出方法	(156)
3.7	C 语言如何读取 dBASE 的库文件	(159)
3.8	用 C 语言直接调用 C-dBASE ■ 数据库文件	(160)
3.9	dBASE 数据双重录入校验程序	(162)
3.10	数据通讯的一种简便方法	(163)
3.11	dBASE ■ 应用中的两个小技巧	(165)
3.12	巧用长城 0520EM 运行汉字 dBASE ■	(166)
3.13	不同数据库系统间嵌入式操作——DECnet-PVDB 系统设计与实现	(167)
3.14	巧用 FoxBASE 中 @.....box 命令	(170)
3.15	一种通用菜单主控程序	(171)

3.16	通用快速下拉式菜单在 FoxBASE 中的应用	(173)
3.17	菜单设计方法与弹跳式菜单程序	(177)
3.18	在 FoxBASE 中建立演示程序	(178)
3.19	FoxBASE 编辑调试技巧	(181)
3.20	FoxBASE 全屏幕通用批输入修改程序设计	(183)
3.21	用 C 语言编制类似 FoxBASE 中的 INKEY(n) 函数	(187)
3.22	谈多用户编程中的冲突处理	(188)
3.23	在 DOS 中直接访问数据库记录和结构	(191)
3.24	数据库应用系统钥匙盘制作技术	(195)
3.25	下拉菜单输入数据应用程序	(199)
3.26	充分利用错误陷阱技术	(200)
3.27	巧用 FoxBASE 中的宏命令	(202)
3.28	FoxBASE 多屏幕浏览式检索	(205)
3.29	在 FoxBASE 下的英汉输入转换	(207)
3.30	FoxBASE 2.1 的内存释放技术及其应用	(209)
3.31	如何在库文件中实现数值型字段的小计运算	(211)
3.32	数据输入程序中编码的无记忆录入	(212)
3.33	FoxBASE+ 与汇编语言通用接口技术	(213)
3.34	谈 FoxBASE+ 反编译的技巧	(219)
3.35	防止对 FoxBASE+ 伪编译程序进行反编译的方法	(222)
3.36	FoxBASE+ 弹出式菜单的实现方法	(223)
3.37	C-MFoxBASE+ 提示边框的修正	(225)
3.38	扩充 FoxBASE+ 的查询命令	(226)
3.39	“将错就错”在 FoxBASE+ 中的应用	(230)
3.40	FoxBASE 下一个新型的通用菜单控制程序	(231)
3.41	FoxBASE+ 的通用表格横向计算与核对程序	(236)
3.42	记录在数据库中的移动程序	(239)
3.43	用 C 语言解决 FoxPlus 不能存取汉字屏幕的问题	(240)
3.44	如何在不走纸的情况下使 PROW() 置零	(242)
3.45	FoxPro 2.5 for Windows 全程路径文件名解析	(242)
3.46	FoxPro 2.0~2.5 中 BROWSE 命令的使用方法	(245)
3.47	FoxPro 的名字表达式	(248)
3.48	FoxPro 系统菜单的汉化方法	(249)
3.49	FoxPro 2.5 使用技巧	(252)
3.50	数据交叉组配检索的方法与实现	(265)
3.51	ORACLE 7 的主要特点与新功能	(267)
3.52	汉字 CCDOS 下如何使用 ORACLE	(271)
3.53	ORACLE 的分区及其使用	(274)
3.54	SQL * FORMS 3.0 的体系结构与特点	(276)

3.55	SQL * FORMS 用户出口的实现过程	(280)
3.56	数据库结构的自动比较	(282)
3.57	xBASE 数据库的数据去重与实现	(283)
3.58	一种通用的数据库转换方法	(284)
3.59	通用数据库行间四则运算过程	(286)
3.60	用 C 语言实现 dBASE III (FoxBASE) 数据向 ORACLE 的自动转换	(288)
3.61	dBASE III 与 BASICA 之间的数据传送问题	(293)
3.62	磁盘数据的录入与备份	(297)
3.63	数据自动存盘程序	(299)
3.64	论小写数字到大写金额的转换	(304)
3.65	巧算元、角、分	(306)
3.66	Windows 环境下的第一个动态电子表格 (Lotus Improve for Windows)	(308)
3.67	dBASE/FoxBASE 中文报表程序设计工具	(311)
3.68	dBASE III 或 FoxBASE 菜单式自由组合查询	(314)
3.69	大众化的源程序生成程序	(317)
3.70	巧用数组和 GATHER FROM 命令	(320)
3.71	动态封面程序设计	(321)
3.72	格式文件与 SET FILTER TO 命令的结合方法	(322)
3.73	对 70~80 年代管理软件菜单进行改造的方法	(323)
3.74	通用事物管理信息系统的建立方法	(326)
3.75	可编辑的输入函数实现	(329)
3.76	AT() 函数在选票统计中的应用	(332)
3.77	SAS 统计软件系统配置与数据调用	(334)
3.78	四种新一代数学计算软件	(336)

第 4 章 工具软件使用技巧

4.1	PCTOOLS 高级实用技巧	(338)
4.2	保护硬盘信息的工具软件	(340)
4.3	谈 CPShrink 压缩工具的使用	(343)
4.4	Norton 软件包中 BE 的使用	(344)
4.5	妙用 Nu 工具软件	(348)
4.6	HD COPY——最快捷的复制工具	(349)
4.7	PC 中断调用分析工具	(350)
4.8	PCShell 7.0 中 .FNT 字型库的利用	(354)
4.9	PC 机驱动器清洗工具 Driver Clean	(358)

第 5 章 病毒防治技术

5.1	计算机资料的安全维护与防卫措施	(363)
5.2	如何消除 GenP/GenB 病毒	(365)

5.3	MS DOS 6.0 的防病毒功能	(366)
5.4	一种新的 1990 病毒	(369)
5.5	计算机病毒的检测、消除和预防	(370)
5.6	CPAV 使您的电脑百毒不侵	(375)
5.7	防毒软件 LANProtect	(376)
5.8	DOS 自身的抗病毒软件	(379)
5.9	抗病毒软件 TurboAntiVirus V6.80A 简介	(382)
5.10	防病毒软件检测能力的提高	(384)
5.11	引导型病毒与磁盘逻辑坏簇	(385)
5.12	清除主引导型病毒修复主引导记录	(388)
5.13	对抗“火炬”病毒	(389)
5.14	消除引导型病毒一法	(391)
5.15	WIPE 程序和硬盘初始化加密法	(391)
5.16	检查文件型病毒的简单方法	(396)
5.17	清除 1465 病毒的程序	(398)
5.18	二叉树病毒的检测与清除方法	(400)
5.19	文件型 64 病毒的简析与防治	(407)
5.20	新世纪/XQR 病毒的检测与清除	(408)
5.21	DIR 2 病毒的自我快速解法	(412)
5.22	消除 DIR 2 病毒的简便方法	(413)
5.23	香港病毒的消除和防范	(417)
5.24	Generic 病毒的发现与消除	(420)
5.25	TRAVELLER(C)病毒的发现与清除	(422)
5.26	“中国青蛙”病毒	(426)
5.27	Write Protect 病毒的发现与消除	(429)
5.28	1741 病毒的检测及其清除	(439)
5.29	1741 病毒的防治	(442)
5.30	5978 病毒的检测与清除	(444)
5.31	一种伪病毒现象	(449)
5.32	谈谈几种新病毒的正确认识与清除	(451)
5.33	识别计算机病毒	(452)
5.34	正确认识计算机防病毒卡的作用	(453)

第 6 章 加密与解密

6.1	计算机软件硬加密原理及技术	(459)
6.2	磁盘密写技术及有关加密因子的分析	(462)
6.3	简便实用的磁盘加密方法	(467)
6.4	利用特殊磁道为软磁盘加密	(470)
6.5	反跟踪加密实用技术	(472)

6.6	一个文件加密程序	(475)
6.7	磁盘文件的加密与解密	(476)
6.8	汉化 FoxGraph 加密及解密要点	(478)
6.9	子目录加密与解密的一种方法	(480)
6.10	隐含方式加密子目录的解密程序.....	(481)
6.11	加密 WPS 文件的解密	(484)
6.12	CCDOS 4.0 解密的方法	(485)
6.13	CMOS 密码处理	(486)
6.14	PKZIP 压缩软件包的技术背景及使用	(487)
6.15	内存驻留程序安全性要素分析.....	(492)
6.16	AST 386 机口令服务	(496)

第1章 操作系统

1.1 MS DOS 重入分析

MS DOS 是一个单任务的磁盘操作系统,随着应用的不断深入,越来越多的要求需要该系统能够支持多任务。事实上,若干新软件如 Windows 已经支持多任务。尤其是越来越多的人掌握 TSR 程序的设计方法,更需要了解 DOS 的重入问题。在若干公开的文献中,大部分都指出 DOS 是不可重入的,但详细的原因都不甚清楚,本文试图通过对 INT 21H 的总控部分的分析,指出可重入的情况和不可重入的情况。

1.1.1 INT 21H 的总控程序结构

INT 21H 是 MS DOS 的核心部分,要想了解 DOS 的重入问题,只要熟悉 INT 21H 的总控程序结构即可。下面以 MS DOS 3.3 为基础来讨论 INT 21H 的总控程序结构。该程序的框图如图 1.1.1 所示。

根据上述框图分述如下:

- (1)进入 INT 21H 的一开始,首先判断 AH 值是否是合法的值,若不合法,退出。
- (2)判断 AH 中是否是重入码 51H,62H,50H,33H,64H,如果是则进入相应程序,并执行,然后直接返回。
- (3)如果不是上述两种情况,进入真正的 INT 21H,首先是寄存器压栈保护,然后是初始参数设置(1)。它的任务是保存 DS 和 BX 的值到内存单元中,之后将 CS 赋给 DS。
- (4)置 DOS 忙标志,DOS 忙标志的信息可由 INT 21H 的 34H 取到。
- (5)保存原 PSP 地址(但不恢复)和原用户栈指针;保存当前进程的用户栈指针并设置到 PSP 的 +2EH~+31H 单元中。
- (6)首先将 SS:SP 指针设置到系统栈的辅助栈(STACK1)上。
- (7)设置相应的初始化参数,然后,根据 AH 中的值,计算相应功能调用子程序的地址偏移量。
- (8)根据 AH 中的功能号,设置相应的系统堆栈,详细情况见图 1.1.1 的右边部分。
- (9)取相应子程序的地址并取执行(CALL)。
- (10)DOS 忙标志被置为闲。
- (11)恢复用户栈指针和原用户栈指针。
- (12)恢复寄存器值。
- (13)中断返回,退出 INT 21H 程序。

从以上分析可以看出 INT 21H 的总控程序功能包括四部分:第一是重入功能码的判断并执行;第二是初始化参数设置和恢复;第三是根据 AH 中的功能号,计算程序偏移量并调用相应子程序;第四是根据 AH 中的功能号,设置相应的系统堆栈。由此可见,该总控程序与一般应用系统的总控程序的不同是,应用系统的总控程序只有第二和第三部分。

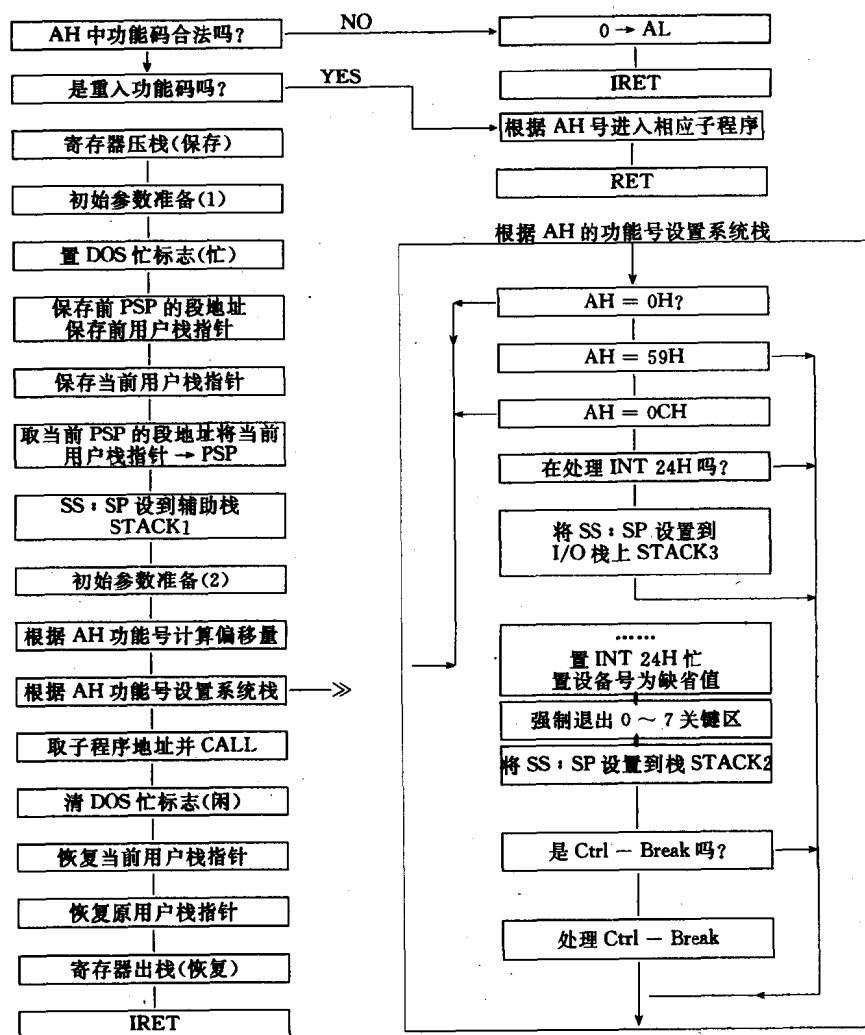


图 1.1.1 INT 21H 的总控程序结构

1.1.2 堆栈切换分析

MS DOS 重入问题的关键是分析 INT 21H 的总控程序,而分析总控程序的核心是了解总控程序对堆栈的使用方法问题。在 INT 21H 中碰到的堆栈有 5 个,图 1.1.2 是 INT 21H 的堆栈及切换状态情况图。

当进入 INT 21H 程序时,先将原用户栈的指针保存到内存单元中,然后,保存当前用户栈的指针再用系统内部的堆栈。系统堆栈有 3 个,分述如下:

(1)辅助栈(STACK1),它是供 INT 24H 和 INT 21H 的 59H 功能调用使用的。系统初始将 SS: SP 指向该栈。

(2)I/O 栈(STACK3),它是专门供 INT 21H 的输入/输出功能调用使用,这些功能调用包括 1H~0CH。

(3)磁盘栈(STACK2),它是供磁盘操作使用的,这些功能调用包括 AH=0, AH>0CH, AH≠59H。

从图 1.1.2 中可以看出,系统在任何时刻 SS : SP 只指向一个系统栈。当系统在一个堆栈上工作时,如果被另一个进程中断,当前堆栈的数据得不到保护,这样系统就崩溃,这就是所谓 DOS 不可重入的原因。

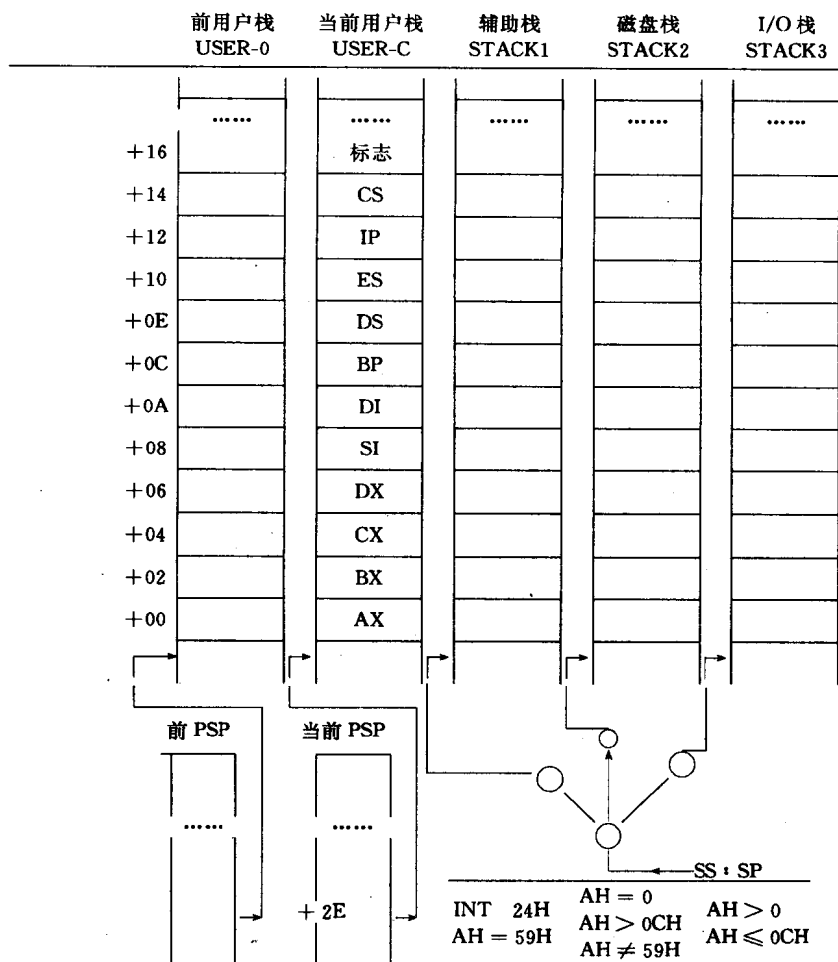


图 1.1.2 INT 21H 的堆栈及切换状态

1.1.3 重入与不可重入分析

重入和不可重入是本文讨论的重点,首先要明白什么叫重入,什么叫不可重入,为清楚起见,定义如下:

不可重入:在执行 INT 21H 的代码时不允许中断它的程序再次调用 INT 21H 的功能,换句话说,在执行 INT 21H 调用时,如又被定时中断 INT 8H 调用 INT 21H,再次进入 INT 21H 的代码(递归),这种情况被称之为 MS DOS 的不可重入。

重入:不可重入的反定义即 INT 21H 的代码可在多个并行进程中,允许一个进程在任何一个时刻中断另一个进程,称之为重入。

通过对 INT 21H 的总控程序和系统堆栈切换分析,我们可以很清楚地了解到 MS DOS 的重入情况分为如下两种情况。

1. 完全可重入的功能调用

从上述分析中可以看到,有一些功能调用是完全可以重入的,它们是:

- (1)50H,设置活动进程的 PSP 段的地址;
- (2)51H,取当前活动进程的 PSP 段地址;
- (3)33H,取/置 Ctrl-Break 状态;
- (4)62H,取当前活动进程的 PSP 段地址;
- (5)64H,设置未知标志(在 PRINT.COM 中使用过)。

这些功能能够重用的原因是它们的操作比较简单和单一,而且不用系统堆栈。

2. 不同堆栈切换的重入

从上述分析中可以很清楚地了解到,SS:SP 任一时刻指向一个系统栈,如果在磁盘栈上工作,被另一个进程中中断到 I/O 栈上工作,这时系统是可以重入的。

不是上述两种情况,一般说来是不可重入的,这是若干公开文献中所指出的。但是,如果是这样,那将大大限制 TSR 的发展,同时也极大地限制了 DOS 的应用范围。DOS 不可重入的最重要的原因是当前进程的堆栈和状态数据得不到保护。事实上 DOS 的真正的重入方式,并不在于它的代码可重入性,而是因为它的数据的可保存性。换句话说,如果当一个进程被切换到另一个进程时,前一个进程的所有状态数据得以保存,当该进程被恢复时,这些数据也被恢复,这就是 DOS 未公开的秘密——DOS 重入方法:DOS 的上下文切换方法。这里简单介绍一下其基本思想。DOS 有一个数据交换区 SDA(Swap Data Area),它包括系统的三个堆栈和其它若干状态数据。它分为两部分:第一部分叫一直交换区(Swap Always Area),大约 18H 字节;第二部分叫交换区,大约 73CH 字节,不到 2KB。后者包括前者。该数据区位于临界区资源 PV 操作数据区的下面。取该区的首地址和大小可通过一个未公开的功能调用来实现,它是:

DOS 3.1~3.3 5 D06H

入口:AX=5D06H

出口:DS:SI 指向 SDA

DX:当 DOS 忙时,一直交换区的大小

CX:交换区的大小

DOS 4.X 5D0BH

入口:AX=5D0B

出口:DS:SI 中返回一个表,其内容如下:

表 1.1.1

偏移量	大小	描述
00H	字	SDA 的数目
SDA 入口:		
02H	双字	SDA 地址
06H	字	数据区长度和类型 第 15 位=1 要交换“交换区” =0 要交换“一直交换区”
		第 4~0 位 数据区的字节数
08H	双字	下一个 SDA 的入口
		

关于什么时候保存一直交换区,什么时候保存交换区,也可用未公开的 INT 2AH 中断来监视。

(龔正科)

1.2 巧用 DOS 中的 I/O 重定向命令

本文介绍 DOS 手册中一个大多数人未予重视的命令,即 I/O 重定向命令,结合其它 DOS 编程命令给出了 4 个应用实例,即使用 I/O 重定向命令修改程序,调试、修改汇编语言程序,显示内存容量,备份硬盘结构信息。

PC DOS 2.0 以上都支持一个类似于 Unix 系统的 I/O 重定向功能,如果用户能熟悉它并结合 DOS 的其它功能加以灵活运用,将会更方便有效地使用微机,提高应用水平。

DOS 的输入、输出通常是在标准设备(键盘和显示器)上进行的,但也可重新定义为另外的设备。此设备可以是磁盘文件或设备文件,即可以改变输入、输出的方向。DOS 承认的常见设备文件有:CON(控制台,即键盘/显示器)、AUX(异步通信口)、PRN(并行打印机)、NUL(虚拟设备)等等。

1. 输出改向

输出改向可以在命令中键入一个大于号“>”加上输出文件名或设备名。如果该输出文件已经存在,则清除该文件并进行复写。格式是:

```
>[d:][path]filename
```

使用两个连续的大于号“>>”,可用来添加一个文件。如果该文件已存在,则输出将附加于原文件之后,不会破坏原文件。若文件不存在,则该命令类似于上述命令。

```
A>dir b:*.exe>b:dirs.txt
```

```
A>dir b:*.com>>b:dirs.txt
```

这两个重定向的列目录命令,可产生一个目录文件 dirs.txt,包括扩展名为.com 和.exe 的文件目录。进而用命令:

```
A>type b:dirs.txt>prn
```

```
A>copy b:dirs.txt>lptl:
```

这两个命令达到的效果是一样的,即重定输出设备为并行打印机。

2. 输入改向

类似于输出改向,在命令行中使用小于号“<”及输入文件名,可进行输入改向。格式是:

```
<[d:][path]filename
```

3. 使用实例

(1) 利用重定向功能修改程序

在软件移植过程中经常要对原有软件进行修改,特别是软件的 I/O 接口部分。这类修改的特点是:难度不高,工作量大,需要非常细心,又很有规律。我单位有一个用 dBASE III 编写的人事工资管理系统,原运行环境为 PC/XT 机,汉字 CC DOS2.1,现需改在 286 机的高分辨率环境下工作。由于制表符区位的差异,使 30 个报表程序中的制表符都要进行修改。如用 EDLIN 程序处理,虽然可以用字符串替换功能,但由于 EDLIN 只提供命令方式而不支

持程序方式,若每个程序有 20 种制表符需要修改,则共需 $30 \times 20 = 600$ 次的字符串替换命令。用重定位功能可以实现修改过程的自动化。方法如下:

①用 EDLIN 或 COPY CON:AA.BAT 建立一个名为 AA.BAT 的批命令文件,其中插入了以下形式的命令:

```
FOR %%FIN(*.PRG)DO EDLIN%%F<XG
```

其中:XG 是一个任取的文件名,其内容是 EDLIN 程序的命令序列。在本例中,可用 EDLIN 自身建立,如下:

```
1,R <原字符串 1>^Z<替换字符串 1>
```

```
1,R <原字符串 2>^Z<替换字符串 2>
```

...

...

```
1,R <原字符串 N>^Z<替换字符串 N>
```

```
E
```

②现在程序修改工作只需在 DOS 提示符下运行 AA.BAT,即可一次修改完所有的程序。用类似的方法可修改各类程序,只要对 XG 文件进行细致的设计,就可获得满意的结果。

(2)用 DEBUG 的 U 命令可以很方便地进行反汇编

如果没有汇编语言源程序,又需要修改源程序,尤其是插入修改,一般是比较麻烦的。但实际上只要有扩展名为 .COM 和 .EXE 的可执行文件,再结合重定向命令就方便多了。如有一个 T.COM 文件,长度为 200H 字节。键入以下命令:

```
A>DEBUG T.COM>T.ASM
```

```
-u 100 3ff
```

```
-q
```

T.ASM 就是 T.COM 的机器码和反汇编结果。你可以用 EDLIN 等编辑软件调用 T.ASM,进行分析和打印,修改时将其中的机器码去掉,借助于编辑软件进行插入、替换等修改。工作完成后,将第一行的 U 命令改为:

```
f 100 ffff 0
```

```
a 100
```

存盘,然后键入命令:

```
A>debug<t.asm
```

在退出 DEBUG.COM 后,用命令方式进入 DEBUG 环境中,再用 D 命令查看文件长度(最后一个非零字节的地址减去 100H),设置寄存器 BX,CX,再将汇编好的内容以文件名 T.COM 存盘即可。

(3)在 AUTOEXEC.BAT 中显示内存容量,加强对病毒的监视能力

许多病毒常修改内存总量(减少 1~2K),使病毒程序常驻内存。PC 机的内存总量参数存放在 0:0413H 和 0:0414H 单元中,只要知道微机的实际容量,对比一下就可以判定内存中是否染上病毒。这种方法对于那些操作系统型病毒、感染 COMMAND.COM 的外壳型病毒以及利用修改内存总量隐藏自己的病毒都很有效。一旦发现参数不符可及时采取措施。由于 0:0413H 和 0:0414H 两单元存放的是基本内存参数,所以对一些扩展了内存的 AT, 286, 386 机同样适用。常见的内存配置和两单元的对应关系如下:

实际内存	0:0413H	0:0414H
640K	80H	02H
512K	00H	02H
384K	80H	01H
256K	00H	01H

在批文件中加入以下命令:DEBUG<DEBUG.DAT,其中 DEBUG.DAT 是 DEBUG 程序的命令序列,建立方法同(1),内容为:

```
D 0:413 L 2
```

```
Q
```

(4)备份硬盘结构信息

微机用户都知道像 DOS 系统盘是很重要的系统软件,需要做好备份。实际上对拥有硬盘的用户来说,硬盘的结构信息(包括硬盘引导区 BOOT,DOS 分区引导区 DOSBOOT,文件分配表 FAT,文件目录 FDT)也是非常重要的系统软件。尤其是现在许多恶性病毒发作时都破坏硬盘的结构信息,从而达到破坏整个硬盘数据的目的。经常备份硬盘的结构信息可以最大限度地减少病毒造成的数据损失,对于引导型的病毒也可以起到消毒的作用。

以 DOS 3.3 管理的 43M 硬盘 C:为例,假设硬盘结构信息的四部分分别保存到文件 BOOT.DOS, DOSBOOT.DOS, FAT.DOS, FDT.DOS 中。实现保存的批处理文件 BACK.BAT 内容如下:

```
@ECHO OFF
IF NOT EXIST *.DOS GOTO A
ATTRIB -R *.DOS
IF NOT EXIST *.BAK GOTO B
ATTRIB -R *.BAK
DEL *.BAK
:B
RENAME *.DOS *.BAK
:A
DEBUG<BACK.DAT>NUL
ATTRIB +R *.DOS
ATTRIB +R *.BAK
ECHO BACKUP SUCCESSFUL!
```

实现恢复的批文件 REST.BAT 内容如下:

```
@ECHO OFF
IF EXIST *.DOS GOTO B
ECHO DOS INFORMATION NOT FOUND!
GOTO END
:B
DEBUG<REST.DAT>NUL
ECHO RESTORE SUCCESSFUL!
:END
```

其中 BACK.DAT 和 REST.DAT 为 DEBUG.COM 的命令序列。它们的内容为:

BACK.DAT	REST.DAT
A 100	A 100
MOV AX,0201	MOV AX,0310
MOV BX,1000	MOV BX,1000

MOV CX,0001	MOV CX,0001
MOV DX,0080	4000
INT 13	W 100
INT 3	Q
G=100	MOV DX,0080
N BOOT.DOS	INT 13
R CX	INT 3
200	N BOOT.DOS
R BX	L 1000
0	G=100
W 1000	N DOSBOOT.DOS
L 100 2 0 1	L 100
W 100	W 100 2 0 1
N DOSBOOT.DOS	N FDT.DOS
L 100 2 1 55	L 100
R CX	W 100 2 1 55
AA00	W 100 2 65 55
N FAT.DOS	L 100
W 100	L 100
L 100 2 AB 20	W 100 2 AB 20
N FDT.DOS	Q
R CX	

注意:使用时需将文件 DEBUG.COM, ATTRIB.COM, BACK.BAT, REST.BAT, BACK.DAT, REST.DAT 都放在同一软盘上。

用户在做好备份后,可用 COMP 命令比较 BOOT.DOS 和 BOOT.BAK,以及 DOSBOOT.DOS 和 DOSBOOT.BAK 是否一致。若原机器无毒,比较结果两文件不一样,则说明有病毒侵入。同时也可结合(3)来判断。发现有毒后,解毒方法很简单:在无毒环境下,将备份好的硬盘结构信息用 REST.BAT 恢复即可。注意:这时恢复要用 *.BAK,因为 BOOT.DOS 和 DOSBOOT.DOS 可能已经包含着病毒的程序。

(缪晓明)

1.3 DOS 内部命令

1.3.1 DOS 内部命令的修改和扩充

DOS 由以下 4 个主要部分组成:引导块, IBMBIO.COM, IBMDOS.COM, COMMAND.COM, 其中 COMMAND.COM 负责接受键盘命令,并执行相应的命令子程序;若是像 DIR, COPY, RENAME 等内部命令,就直接由驻留在内存的代码执行;若是外部命令,则调用相应的可执行程序执行。DOS 内部命令是由 COMMAND.COM 初始完成并驻留内存的,其优先级最高,利用 DEBUG.COM 将 COMMAND.COM 调入内存,可找到一个 DOS