

内存驻留功能设计及程序实现

施 伟, 刘建辉

(辽宁工程技术大学, 辽宁 葫芦岛 125105)

摘 要: 驻留内存程序是解决 PC DOS 单用户单任务局限的一个行之有效的办法。从具体的例子出发, 说明了这种办法的原理和实施步骤。介绍了这种办法在热键冲突、显示、DOS 调用嵌套诸方面所可能遇到的种种问题及其解决办法, 讨论了驻留程序彼此之间以及与其他程序之间的兼容性问题。

关键词: 驻留内存; 进程; 激活标志; 中断

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2007)11-0099-04

Function Design and Procedure Realization of Resident Memory

SHI Wei, LIU Jian-hui

(Liaoning Technical University, Huludao 125105, China)

Abstract: Resident memory procedure solves the limitation of single user and single duty of PC DOS effectively. Embarking from the concrete example, this article introduced its principle and implementation step. Thoroughly introduced all sorts of questions and its solution which possibly meet in the hot key conflict, the demonstration, the DOS transfer nesting and so on. Fully discussed compatible question between resident program each other as well as with other procedure.

Key words: resident memory procedure; advancement; activation symbol; severance

0 引 言

内存驻留程序即 Terminate and Stay Resident Program, 缩写为 TSR。这种程序加载进内存, 执行完成后, 就驻留在内存里, 当满足条件时, 调到前台来执行。

程序常驻内存的目的有两个: 第一, 对操作系统或 BIOS 进行某些修改或补充, 比如转入用户编写的必须恒起作用的中断处理程序; 第二, 解决 PC DOS 不能实现多任务处理的问题, 例如在运行其他程序的过程中, 定时地在屏幕上闪现一些日历、时间和其他提示信息。不过一些可恶的人也利用 TSR 技术制作很多可恶的病毒程序, 几乎所有的病毒程序都是 TSR 程序。

因此利用常驻内存的方法可以为应用程序更深层次的使用创造条件, 达到多任务操作的目的。

1 内存驻留程序设计的一般过程

驻留程序分成两个部分, 即暂驻部分和驻留部分。

第一部分为初始化部分, 这部分只是在常驻内存程序装入时执行一次, 随后被废弃。它主要完成两种工作: 一是如何使 DOS 再次执行常驻内存程序的驻留部分, 这时要进行有关中断向量的修改; 二是常驻内存程序结束及驻留^[1]。一般使用 INT 21H 的 31H 子功能和 INT 27H 来实现。第二部分为驻留部分, 它将在一定条件下被激活而运行。激活驻留内存程序的方法包括以下几种:

(1) 硬件中断: 常用的是键盘中断 INT 9H, 时钟中断 INT 8H, 通讯中断 INT 14H, 磁盘中断 INT 13H 等等。

(2) 软件中断: 常用的是键盘中断 INT 16H, 时钟中断 INT 1CH, DOS 中断 INT 21H, 等等。

(3) 以上两种的结合。

一个完整的驻留内存程序可以抽象成以下几个过程:

- * 取中断向量;
- * 保存旧的中断向量;
- * 设置或恢复中断向量;
- * 中断处理程序的链接;
- * 检测是否已驻留;
- * 执行终止并驻留;

收稿日期: 2007-01-04

基金项目: 辽宁省教育厅基金项目(2004G108)

作者简介: 施 伟(1978-), 男, 辽宁铁岭人, 博士研究生, 研究方向为计算机系统结构、管理信息技术; 刘建辉, 教授, 博士生导师, 研究方向为计算机系统结构、管理信息技术。

* TSR 的删除。

删除 TSR 比较复杂,必须按下列步骤进行^[2]:

a. 检查中断向量是否已经被替换。如果没有替换,就恢复所有的中断向量;如果某个中断向量被替换,则应该跳过下面各步,该 TSR 不能删除。

b. TSR 的 PSP 中偏移量 16H 存放着主进程的 PSP,把这个值改为当前进程的地址。

c. 把当前 PSP 设为 TSR 的 PSP。

d. 执行 INT 21H 的 4CH 功能,释放 TSR 占用的内存,关闭所有文件,并使用 PSP 中存放主进程的地址。

e. 返回到初始进程中,当前 PSP 也指向初始进程,所有寄存器值包括 SS 和 SP 都不确定。

在执行完上述步骤后,要恢复寄存器。如果要无条件地删除 TSR,必须监控每个 TSR 对中断向量表、内存控制块和设备驱动程序链的修改。

2 相关技术及问题

2.1 COM 程序

DOS 之下有两种形式的可执行文件,这两种文件分别是 COM 文件和 EXE 文件。其中,COM 文件可以迅速地加载和执行,但是其大小不能超过 64k 字节,只能有一个代码段,而且起始地址为 100H。EXE 文件可以加载到许多个段中,因此程序的大小没有限制,但是程序加载的过程就比较慢,这对于内存驻留程序来说会造成很大的麻烦。

以下是一个 COM 文件的框架,只要将任何应用部分加在这个文件中,就可以形成一个 COM 格式的内存驻留程序:

```
STEP1
CODE SEGMENT
ASSUME CS:CODE,DS:CODE
ORG 100H
STEP2
START:
RET;可执行程序部分,这里用 RET 代替
STEP3
CODE ENDS
END START
```

上面的程序可以分成三部分:第一部分定义了代码段和数据段分别放在程序中的位置,以及执行代码的起始地址;第二部分是可执行的程序;第三部分是程序段的终结。这个程序在使用汇编连接并使用 EXE2BIN 后生成的 COM 文件只有 1k,说明 COM 文件实际是可以支持很长的程序段的,完全可以满足驻留内存程序的需要^[3]。

2.2 DOS 调用的嵌套问题

编写驻留内存程序时所感到的一个很困难的问题是 DOS 系统调用的嵌套问题。几乎所有的驻留内存程序在涉及到链盘、显示、时间、打印机等设备的处理时,大都绕过 DOS,而直接使用 BIOS 或 directly 对硬件编程。但在涉及到文件操作时,因为编程过于麻烦,所以又几乎都是通过 DOS 的系统调用去实现。这种带有 DOS 文件系统调用的常驻程序有时能工作,有时不能工作,甚至还会毁坏盘上已有的文件。发生这种情况是因为 DOS 是单任务操作系统,是不可重入的。

不可重入的意思就是说,当正在处理一个 DOS 系统调用时,别的程序不能再发出一个 DOS 系统调用。DOS 调用会改变 DOS 唯一的环境块,重复的调用会引起系统崩溃。当然,DOS 的调用中有的没有使用 DOS 环境,它们是可重入的。DOS 重入在驻留内存的程序中是什么情况呢?这就是当前台程序在调用 DOS 的 21H 中断,还没有返回前台程序期间,中断服务程序被某个事件激活。如果程序中调用了 DOS 的 21H 中断,这就出现了 DOS 重入。如果中断服务程度不需要调用 INT21H,那就不会出现 DOS 重入。

解决这个问题的一种办法是使驻留内存程序也截取中断 21H,目的是设立一个标志来表示当前是否正处在 DOS 系统调用的处理过程中。驻留程序在被激活前,应首先查看这个标志。如果这个标志为 0,则表明当前不在 DOS 系统调用的处理过程中,于是就激活驻留程序。如果这个标志已置位,则不激活驻留程序。

DOS 毕竟不是一个多任务系统,不论采取什么办法去解决它的不可重入问题,都不会取得完全彻底的效果。当一个使用了文件输入输出系统调用的驻留程序编写完以后,应该测试一下它在多大程度上解决了不可重入的问题^[4]。

一种可行的办法是使用 DOS 的 TYPE 命令去显示一个比较长的文件,在显示的过程中,去激活一个驻留程序去存取一个文件。因为使用了 TYPE 命令在显示文件的内容时不是采用低号的系统调用,而是采用系统调用 40H,所以此时驻留内存程序不应该被激活。否则,说明驻留程序编写的不够好,需要修改。

2.3 热键冲突问题

许多驻留内存程序都定义了一个特殊的热键,一旦用户打入了这个热键,就激活了常驻程序,开始工作。检测一个热键的办法大体上有三种:通过 DOS 的系统调用;通过 BIOS 中的键盘功能中断 16H;通过修改中断 9H 来设置新的处理程序^[5]。

很显然,热键一般应该是一个很特殊的几乎不用的键或几个键的组合,但即便如此也不能保证驻留程

和其它应用程序的热键没有相同的可能。如果通过已有的 BIOS 调用 16H 或 DOS 的系统调用来检测热键,由于这些键的定义是很通用的,所以产生冲突的可能性就很大。

因此,大部分驻留程序都是通过建立自己的中断 9H 处理程序来检测热键的。这样,驻留程序可以使用正常的 BIOS 中断 9H 所不承认的无定义键或死键组合来作为热键。但这种解决办法仍然是不彻底的。如果同时常驻在内存的程序有许多个,且这些常驻程序都是通过修改中断向量 9H 来定义自己的热键,则在把这些常驻程序装进内存以后,它们的中断 9H 处理程序将按照装入的先后次序连成一条链。链头是最后装入的驻留程序,链尾是由 BIOS 提供的原始的中断 9H 处理程序。如果这些程序所定义的热键是彼此不同的,则各取所需,不会发生冲突。但是,如果有两个驻留程序定义的热键彼此相同,则不可避免地还要发生冲突,最后的结果总是激活最后装入的驻留程序^[5]。

即使同时装入的各驻留程序所定义的热键彼此不同,但如果它们通过中断 9H 所定义的一些别的键彼此相重,还是可能发生另一类型冲突。以键盘宏定义程序为例,它既截取中断 9H 的控制权,又截取中断 16H 的控制权。当用户敲入的键是宏键时,键盘宏定义程序把一个特殊码存入到键盘缓冲区,当应用程序通过中断 16H 来取字符时,如果取出来的是这个特殊码,键盘宏定义程序再以一个字符串代替这个特殊码发回到应用程序。现在,假设用户既装入了上述的键盘宏定义驻留程序,又装入了一个以键来激活的常驻的文本编辑程序。两者定义的热键不同,因此可以同时被激活。

但是,这两个常驻程序是否能协调工作,要示具体情况而定。

(1) 驻留的文本编辑程序是通过中断 16H 来读取字符的,此时键盘宏定义程序和文本编辑程序可以同时起作用。

(2) 驻留的文本编辑程序不是直接通过中断 16H,而是通过调用原来的中断 16H 处理程序来读取字符。在这种情况下,只有先装入键盘宏定义程序后装入文本编辑程序,宏定义才可以起作用。否则宏定义不起作用。

(3) 驻留的文本编辑程序也是通过中断 9H 来读取字符的。在这种情况下,无论装入两个程序的顺序如何,键盘宏定义都不会起作用。而在停止常驻的文本编辑程序的工作后,一旦再通过中断 16H 去读取任一字符时,屏幕上会显示键盘宏定义的字符串。

3 设计实例

下面是一个具体的驻留内存程序,程序驻留后会在计算机运行一个小时候用喇叭报警,报警时间为 1 到 2 秒。

```
CODE SEGMENT
ASSUME CS:CODE,DS:CODE

ORG 100H
START:JMP INTA
OLD DD ?
ALARM PROC FAR
    MOV AL,0B6H
    OUT 43H,AL
    MOV AL,8
WAIT:SUB CX,CX
HERE:LOOP HERE
    DEC AL
    JNZ WAIT
    MOV AL,AH
    OUT 61H,AL
    MOV AX,CS
    MOV DS,AX
    MOV SI,OFFSET OLD
    MOV AX,0
    MOV ES,AX
    MOV DI,4 * 4AH
    MOV CX,4
    REP MOVSB
ALARM ENDP
INTIA:MOV AX,CX
    MOV AX,666
    OUT 42H,AL
    MOV AL,AH
    OUT 42H,AL
    IN AL,61H
    MOV AH,AL
    OR AL,03H
    OUT 61H,AL
    JC STEP
    MOV AX,SEG ALARM
    MOV DS,AX
    LEA DX,ALARM
    MOV AX,254AH
    INT 21H
STEP1:MOV AH,2
    INT 1AH
    JC STEP1
    MOV AL,CH
    ADD AL,1
    DAA
```

```

CMP AL,24H
JL STEP2
MOV AL,00H
STEP2:MOV CH,AL
MOV AH,6
MOV DS,AX
LEA SI,OLD
MOV AX,354AH
INT 21H
MOV WORD PTR [SI],BX
MOV WORD PTR [SI+2],ES
STEP:MOV AH,7
INT 1AH
INT
CODE ENDS
INT 1AH
JC STEP1
MOV CX,3
SETT:MOV AH,2
MOV DL,8
INT 21H
LOOP SETT
MOV BX,CS
MOV DS,BX
MOV DX,OFFSET INTIA
21H END START

```

程序结构分为驻留和非驻留两部分。驻留部分是用户的报警中断程序,除鸣笛外,还要在退出前恢复

INT 4AH 中断向量。非驻留部分完成保护 INT 4AH,用 INT 1AH 的 7 号功能取消原报警时刻,同时设置 4AH 新的中断向量,使之指向用户报警程序,用 INT 1AH 的 6 号子功能设置报警时刻,然后使用中断 INT 27H 中断驻留退出。

4 结 论

驻留内存程序是一种较特殊的程序。由于加载后一直占有系统资源,会影响系统的整体效率。所以,一般情况下尽量不要使用驻留程序。特别在 DOS 这类主要是在单任务下工作的操作系统中如使用不当会使系统工作效率降低或不能正常工作,甚至造成整个系统的崩溃。因此,在开发应用程序时,若必须使用驻留程序,其程序长度要尽量缩短,以减少驻留的内容。

参考文献:

(上接第 95 页)

- [11] Lu Baoliang, Wang Kaian, Utiyama M, et al. A part - versus - part method for massively parallel training of support vector machines[C]//Proceedings of IEEE/INNS Int. Joint Conf. on Neural Networks (IJCNN2004). Budapest, Hungary: [s. n.], 2004: 735 - 740.

- [12] SMOBR - ASMO Program for Training SVMs[EB/OL]. 2001. <http://www.litc.cpdee.ufmg.br/~barros>.
[13] Chang Chih - Chung, Lin Chih - Jen. LIBSVM: a library for support vector machines[EB/OL]. 2001. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

(上接第 98 页)

据流处理的解决方案。采用以上技术的解决方案已成功地应用于基于 RFID 的汽油发动机生产线监控执行系统中,并在应用中取得了较好的效果。但本方案还不完善,未能很好地解决安全性问题,如何在套接字接口实现安全认证、身份验证、授权和审计等还有待进一步研究。

参考文献:

- [1] 周四军,王建宇. 基于条形码技术的车间监控系统的实时信息采集[J]. 电子技术应用, 2000(5): 21 - 23.

- [2] 王璐,秦汝祥,贾群. 基于 RFID 技术的门禁监控系统[J]. 微机发展, 2003, 13(11): 59 - 63.
[3] 秦王景,储方杰,高文. 中间件技术研究[J]. 计算机应用研究, 2006, 23(2): 292 - 296.
[4] 李俊平. 银行大机互连通信中间件的设计[J]. 计算机与数字工程, 2000, 28(6): 58 - 63.
[5] 高建华,沈莹. 一种新的交易事务处理程序设计方法[J]. 计算机工程与科学, 2001, 23(5): 87 - 90.
[6] 周函,周波,董金祥. 对象持久层高性能 Cache 实现技术[J]. 计算机应用研究, 2003, 20(11): 35 - 37.