

一种实用的硬盘双重加锁程序

云南省军区自动化站 李晓华

摘要:本文分析了硬盘的自举原理,讨论了硬盘的主引导扇区记录、分区信息表。介绍了一种实用的硬盘双重加锁技术。最后给出了一个具体的实例程序。

目前微机硬盘容量逐渐增大。如 USER 386 机已达到 100M。对于存在这样大的硬盘上的信息是非常重要的。如何保存好自己的数据,不让别人从硬盘上盗窃信息是众所关心的。

一、硬盘自举原理

在分析硬盘加锁程序以前让我们先看看硬盘的基本结构。通常硬盘被划分成称为分区的 4 个区域。一个专用的操作系统(如 DOS, XENIX)只占用一个在硬盘上所定义的分区空间。但一个操作系统可以有属于它的不止一个分区。按 DOS3.0 至 4.0 的版本, DOS 允许在硬盘上建立 2 个属于 DOS 的分区:一个是主要的 DOS 分区(分区的容量可以到 32M),另一个是扩展的 DOS 分区(分区的大小可以到硬盘所限制的容量)。而扩展 DOS 又可以再划分成多个叫做逻辑驱动器的区域。而逻辑驱动器是可以达到 32M 的容量。当机器启动时,只能从硬盘的主要 DOS 分区的驱动器上启动。

在主要 DOS 分区驱动器 C 盘的 0 柱头 0 磁头第一扇区(相对于 DOS 扇区)中存放有用于系统启动时的 DOS 引导记录。在 PC 家族中,它被分成两部分,一部分固化在 ROM-BIOS 中的 INT19H 自举中断程序;另一部分是保存在引导盘上的 DOS 引导记录。DOS 操作系统启动时,自举例程将 DOS 引导记录读入到 0000:7C00H 处,并由引导例程负责装入 IBMBIO.COM,当该程序得到控制权后,装入 DOS 操作系统的核心部分 IBMDOS.COM 和 COMMAND.COM。

这里要注意的是,若是从硬盘启动,则 INT19H 自举程序首先把主引导扇区(存放在硬盘上的 0 柱头 0 头 1 扇区)的内容读入 0000:7C00H 处,而后根据分区表中

的分区信息,引导相应的活动分区,最后再由主引导程序把对应主要 DOS 分区的第一扇区(DOS 引导记录)的内容读入内存 0000:7C00H 处,并把控制权转给 DOS 引导记录。从而完成 DOS 的引导过程。

二、分区信息表及加锁原理

为了实现多个操作系统共享硬盘,硬盘在逻辑上可以分为 1~4 个分区。每个分区内部的逻辑区号是邻接的。这样在硬盘上最多允许 4 个操作系统共享硬盘。但就每个操作系统而言,分配给它的一个分区均被看作一个“整磁盘”。分区信息表是由 FDISK 命令对它进行硬盘分区的(具体使用见 DOS 命令 FDISK 一节)。分区信息表由 4 个项组成,每区占 1 项,每项由 16 个字节组成。占用 0 柱 0 头 1 扇区 1BEH-1FDH 之间的 16*4 个字节。其主要内容包括各分区的大小和起止柱面号、磁头号 and 扇区号等。其中还有 DOS 系统的标识符(0:未确定 1: DOS_ FAT 表项长 12 位 2: XENIX 3: DOS_ FAT 表项长 16 位 5: 扩展 DOS 6: 保留 DBH: 并发的 DOS), 引导标识符(00: 表示不从本分区引导系统 80H: 表示从本分区引导系统)等重要信息。当系统启动时首先要读入这些信息,判断是从那个分区引导操作系统。

分区扇区总计 512 字节,除主引导程序(包括系统提示信息)和分区信息表外,中间还有一部分没有用到。加锁原理就是利用 DOS 系统标识符来实现对硬盘加锁的。其主要设计思想是:在中间不用的那部分主引导扇区中编写扩展加锁程序。程序首先把各分区的 DOS 系统标识符的内容搬到其它内存单元保存,而后清除 DOS 的系统标识符。当输入正确口令后再恢复其内容。本程

序能对硬盘能实现双重加锁。当机器启动时,首先提示输入主要 DOS 分区驱动器的口令(pass)。当口令正确后(口令不正确,不能操纵硬盘,进入死循环)。又提示输入扩展 DOS 分区驱动器口令(logicla)。口令正确后,方能操纵扩展 DOS 分区所分配的逻辑驱动器,否则只能进入主要 DOS 分区的所指定的驱动器。从而达到双重加锁的目的。

三、实用的硬盘双重加锁程序

本程序全部由 8086 / 8088 汇编语言写成,输入下例程序后,用 MASM PASSWORD; LINK PASSWORD; 得到 PASSWORD.EXE 程序,执行一次 PASSWORD.EXE。尔后启动机器,就可以达到双重加锁。需要说明的是为了使其它用户用软盘启动机器后,不能操作硬盘,必须完成以下操作:

第一种方法:关机前执行一次 PASSWORD. EXE

第二种方法:修改命令处理程序 COMMAND.COM 具体修改读者自己完成。只要在 command.com 结束前把分区信息表 DOS 标识符清除即可。

;硬盘双重加锁程序

;filename pass_cd.asm

;lock in C: D(logicla)

;password [C:pass][D(logical):logicla]

addprog equ 6dah ;加入程序到该处

jmp61daddr equ 618h ;修改主引导扇区的 JMP
0000:061d 到扩展程序

dossysf equ 7c2h ;分区信息表的 DOS 系统标识符

sotredossysf equ 7b0h ;保存 DOS 系统标识符

code segment

assume cs: code, ds: code

msg1 db 'A utility program two time LOCK Fixed
Disk. ',0dh,0ah

msg2 db 'Expanded program is intoed Main BOOT
sector. ',0dh,0ah,'\$'

start: push cs

pop ds

push cs

pop es

mov ah,09

lea dx,msg1

int 21h

mov ax,0201h

mov cs,0001h

mov dx,0080h

mov bx,600h

int 13h ;读取主引导扇

mov di,jmp61daddr

mov si,offset modi

mov cx,nummodi

repnz movsb ;修改 JMP 到扩展程序

mov di,addprog

mov si,offset addpass

mov cx,num

repnz movsb ;把扩展程序加入到主引导扇区中

mov cx,4

mov di,sotredossysf

mov si,dossysf

m1: lodsb

stosb

add si,0fh

loop m1 ;把分区表的 DOS 系统标识符保存

mov al,00

mov di,dossysf

mov cx,04

m2: stosb

add di,0fh

loop m2 ;清除分区表 DOS 系统标识符

mov ax,0301h

mov cx,0001h

mov dx,0080h

mov bx,600h

int 13h ;重写主引导扇

mov ah,09

lea dx,msg2

int 21h

mov ax,4c00h

int 21h ;返回 DOS

```

addpass:          ;扩展程序入口
    mov al,00
    mov di,dossysf
    mov cx,04
re3: stosb
    add di,0fh
    loop re3      ;清 DOS 系统标识符
;
disp: mov ax,1301h
    mov bx,000ah
    mov bp,(offset passmsg)-(offset
addpass)+addprog
    mov cx,24
    mov dx,0a00h
    int 10h      ;显示输入口令 1
;
    mov cl,4
    mov di,(offset password)-(offset
addpass)+addprog
r1: mov ah,0
    int 16h      ;输入口令 1
    cmp al,[di];口令比较
    jnz disp     ;口令不正确死循环
    inc di
    dec cl
    jnz r1       ;比较 4 次
ok:          ;正确执行下面程序
displ: mov ax,1301h
    mov bx,000ah
    mov bp,(offset pass d)-(offset
addpass)+addprog
    mov cx,28
    mov dx,0b00h
    int 10h      ;显示输入口令 2
    mov cl,7
    mov di,(offset pass w)-(offset
addpass)+addprog
r22: mov ah,0
    int 16h      ;输入口令 2
    cmp al,[di];口令比较
    jnz exit__c;不正确转
    inc di
    dec cl
    jnz r22     ;比较 7 次
;
    mov si,sotredossysf;正确,恢复 DOS 系统标识符
    mov di,dossysf
    mov cx,04h
re44: lodsb
    stosb
    add di,0fh
    loop re44
    jmp exit
exit c:          ;只恢复主要 DOS 分区分配的驱动器号
    mov si,sotredossysf
    mov di,dossysf
    lodsb
    stosb
    add di,0fh * 2
    inc si
    lodsb
    stosb
    add di,0fh
    lodsb
    stosb;
exit:
    mov ax,0301h
    mov bx,600h
    mov cx,0001h
    mov dx,0080h
    int 13h      ;重写主引导扇区
    db 0eah
    dw 061dh,0
; jmp 0000:061dh ;转原 JMP 的入口
passmsg db 'Input PRI Disk password:'
password db 'pass'
pass d db 'Input logical Disk password:'
pass w db 'logical'
modi db 0eah
dw addprog
    nummodi equ $ -modi
    num equ $ -addpass
    code ends
end start

```