

磁盘操作安全保护技术及应用实例

王晓武 (解放军总后司令部 100842)

应用软件系统频繁的磁盘读写操作过程中,若不采用可靠的安全保护技术,往往会给用户带来不便或造成数据丢失。本文简要分析现有应用系统存在的问题并介绍一种“主动式”磁盘操作安全保护技术及应用实例。供读者参考。

现有的应用系统,包括市场上流行的一些 MIS 自动生成系统,在进行软盘读写操作过程中,大多没有主动测试软盘当前状态,因此,普通用户在使用这类应用系统时稍有不慎(比如,用户在做数据备份、恢复等磁盘操作时,忘记了在驱动器中插入软盘或无意中碰了键盘以及软盘写保护标签不符合程序要求)应用系统就会发生异常中断, DOS 或 WINDOWS 的系统错误报警提示就会赫然显示在屏幕上,致使普通用户一时不知所措。因为这种错误提示大多是英文,即便弄清了提示的含义,也难以造成应用系统界面被破坏,甚至造成系统中断和数据丢失的严重后果。这种现象绝不是耸人听闻,相信经常使用这类应用系统的用户,都会有深深的感受。

造成上述磁盘操作异常的原因并不复杂,主要是由

于这类应用系统开发过程中,忽略了安全保护环节,在应用系统进行磁盘操作之前,没有主动进行磁盘状态的测试,当然就不能正确地处理磁盘的异常状态。所以一旦发生异常,应用系统就会陷入 DOS 或 WINDOWS 系统的错误处理接口,从而引起界面的混乱。

如何在应用系统中主动测试磁盘当前状态,避免磁盘读写异常呢? 解决这个问题确实要费一番周折,因为流行的支撑软件(如 FOXBASE、FOXPRO 及大部分高级语言)均没有提供现成的有关函数和命令供程序员使用。因此,程序员必须深入了解 PC 机 I/O 接口方面的有关技术规范,编写相应的调用 BIOS 或磁盘端口操作的接口子程序,然后在应用系统中适时地调用自定义的接口子程序,截获磁盘异常状态,主动提示用户正确使用软盘,避免陷入 DOS 或 WINDOWS 操作系统的错误处理接口。所谓“主动式”磁盘操作保护技术就是依据上述原理而采用的一种应用系统安全保护措施。

本文以下分别给出 C 语言程序和 Foxpro for Dos 应用系统中,测试磁盘状态的方法和应用实例。

Notes 另一个无与伦比的开发工具是电子化的“签署”文档并让用户检验该文档中的字段自文档被签名后而不被篡改的性能。电子签名文档涉及到已经讨论过的许多加密系统,同时也涉及 Notes 认证的概念。为了有效地使用电子签名,用户必须有一公共认证或者公共原始认证,以使该电子签名能被接收者认可。签名与加密略微有些不同,不管整个文档被签名,还是一个区中的字段被签名文档有多个签名,都允许一个文档有多个签名,而且每个都与窗体中的一个给定区相连。

Notes 加密主要特点是通过设定属性来对文档、数据库的字段进行加密,从而保证其安全性,它较适用于一些交易和文件签发等方面的应用。

四、使用 Notes 安全机制的建议

尽管 Notes 提供了比较完善的安全机制,一般建议不要将所有的不同种类的安全性都集成到一个应用程序

中。因为设计一个安全性过分复杂的应用程序,不仅使应用程序难以测试和维护,而且如果发生意外时,会过多地限制用户。

根据经验,通过有一个良好的稳固 ACL 策略就可以得到约 75% 的 Notes 安全应用程序。当你需要有更高级的安全性时,Notes 的集成化工具可以使你设计出坚固的安全应用程序。

参考文献

- [1] Lotus Notes Developer's Guide 1994 Simon & Schuster(Asia)Pte Ltd.
- [2] Lotus Notes Administrator's Guide 1994 Lotus Development Corporation.

(来稿时间:1996年7月)

实例程序说明:

C语言实例程序首先调用-bios-disk()函数,写软盘扩展扇区(不会破坏软盘正常扇区的数据),然后测试status变量中软盘状态标志值,将当前软盘状态写入一个供Foxpro应用程序调用的内存变量文件“STATUS.MEM”中,其变量名为DSTATUS,(若只在C语言中应用,可不写此文件)供Foxpro for Dos或FoxBASE程序调用。Foxpro系统的内存变量文件结构如下表所示:

Foxpro 内存变量文件结构(STATUS.MEM)

地址(十进制)	内容及含义
0~10	变量名称(字符串)
11	变量类型(字符“C”或“N”)
16	变量长度(数字值、绝对值)
32	变量值(字符型为串,数字型为数字值)
32+变量长度	文件尾标志(两个字节:00H,1AH)

注: 0~32单元内的空余单元应以00H填充。若有多个变量,每个变量的结构相同并顺序排列

程序员在Foxpro程序中调用上述C程序,然后以增加方式读入内存变量文件STAEUS.MEM,根据DSTATUS变量的值,只接受正确的按键进行软盘读写操作,保证系统安全运行。

本文实例程序在MS-C 6.0A、Foxpro for DOS2.6和Foxbase 2.1环境下运行无误(若要在Foxpro for Windows中使用,则必须将C语言接口程序用C++语言进行改写)。

C语言实例源程序清单:

```
#include <stdio.h>
#include <bios.h>

main(argc,argv)
int argc;
char *argv[];
{
    int i;
    FILE *fd1;
    unsigned status = 0;
    struct diskinfo_t disk_info;
    char stbuf[40];
    disk_info.drive = toupper(argv[1][0])-'A';
    if(disk_info.drive>1 || disk_info.drive<0){
        printf("diskttr number error!");
        return -1;
    }
}
```

```
}
disk_info.head =0;
disk_info.track =80;
disk_info.sector =0;
disk_info.nsectors =1; /* 写软盘扩展扇区 */
for(i=0;i<2;i++){ status = bios_disk(DISK_WRITE,&disk_info);
/*printf("disk status:%x",status);*/
if((fd1=fopen("STATUS.MEM","wb"))==NULL){
    printf("STATUS.MEM file not open!");
    return -1;
}
/* 将当前磁盘状态标志 */
for(i=0;i<40;i++){ stbuf[i]='\x0'; /* 写入FOXPRO程序的内存 */
strcpy(stbuf,"DSTATUS"); /* 变量文件STATUS.MEM */
stbuf[11]='C';
stbuf[16]='\x3';
if(status==0x8000){
    stbuf[32]='E'; stbuf[33]='1'; /* 驱动器未准备好状态 */
}
else
if(status==0x300){
    stbuf[32]='E'; stbuf[33]='2'; /* 软磁盘写保护状态 */
}
else{
    stbuf[32]='0'; stbuf[33]='K'; /* 驱动器可读写正常状态 */
}
stbuf[35]='\x1A';
fwrite(stbuf,sizeof(char),40,fd1);
fclose(fd1);
return 0;
}
```

FOXPRO实例部分源程序清单:

```
do while .not.eof()
    @ 14, 0 clear to 22, 79
    @ 18, 10 say "请在A/B驱动器中插入软盘并按A/B 键开始备份数据"
    clear type
    PPV = inkey(0)
    do case
    case PPV=27
        exit
    case PPV=65.or.PPV=97
        diskchar = "A"
    case PPV=66.or.PPV=98
        diskchar = "B"
    otherwise
        loop
    endcase
    set default to
    run diskst.exe &diskchar /* 调用接口程序检测磁盘状态
    restore from status additive /* 读磁盘状态标志内存变量文件
    if DSTATUS="E1"
        ?? chr(7)
        set color to 4+
        @ 23, 0 say diskchar+"驱动器没准备好! 准备好后继续..."
        set color to 7+
        loop
    endif
    if DSTATUS="E2"
        ?? chr(7)
        set color to 4+
        @ 23, 0 say diskchar+"软盘写保护! 去掉写保护后继续..."
        set color to 7+
        loop
    endif
    @ 23, 0 clear to 23, 79
    set default to &diskchar
```