Windows 环境下的文件输入与输出处理

苗沛淋　(平顶山高压开关厂)

摘要:由于 *Windows* 是一个多任务操作系统,所以应用程序应管好文件以避免和其它程序发生冲突。本文指出了在 *Windows* 中进行文件操作时应利用 *OpenFile* 函数创建、打开、关闭文件,用 *C* 语言的低级 *I/O* 函数执行文件输入输出操作,而不使用 *C* 语言的基于流的 *I/O* 函数,并给出了操作方法,以充分发挥 *Windows* 环境的优势。

一、引 言

C 语言是推荐的 Windows 应用程序的开发语言,Windows 的许多编程特性都充分考虑了 C 语言的特点。那么,关于文件的输入输出,在 Windows 程序与标准的 C 运行程序中就应是相似的.但由于标准 C 语言程序是在 DOS 下运行的,而 DOS 与 Windows 两个环境之间存在着较大的差异,比如 Windows 是一个多任务环境,打开的文件的管理就与 DOS 环境中有所不同。所以说,Windows 中尽管可用 C 语言运行函数库的基于流的输入输出函数,但最好是使用 C 语言运行库中的底级 C 语言 I/O 函数,即 C 语言的 SYS/TYPE.H 和 SYS/STAT.H 两个头文件中所定义的函数。

在 C 语言中,一般用 fOpen 来打开文件,fread/fWrite 来读写文件,但在 Windows 环境下,使用 OpenFile 函数来进行文件操作,并通过它返回的 DOS 文件句柄调用低级函数,管理数据读写等操作。

二、Windows 环境中处理文件的原则

在 Windows 环境中,由于同一时刻可能有多个应用程序在进行文件操作,为了避免发生访问冲突和丢失文件,应在编程中遵守下述原则。

1.应在应用程序有执行控制权时才打开文件。其原因有两个:

(1)防止其它程序引起的磁盘环境变化而对该文件产生影响。在应用程序中调用可能占有执行控制权的函数时,应当首先把正处理的文件关闭。这样可防止暂时夺取控制权的程序改变了磁盘环境而对文件产生不良影响,造成数据丢失。

(2)DOS 对打开文件的数目有限制。在 DOS 的 CONFIG.SYS 中 FILES 参数指定了可以同时打开的文件的数目,如果多个应用程序都想打开和使用文件,就会

很快用完可利用的文件数,造成程序打开文件失败。

2.进行文件操作时应当遵守 DOS 关于文件名的约束。从一定意义上讲,Windows 是 DOS 环境中的一个应用程序。那么,Windows 实现所有的文件输入输出操作都依赖于 DOS 的文件处理函数;所以在进行文件操作时应当而且必须遵守 DOS 关于文件名的约束。同时,在处理中文等双字符串中前后移动,以正确处理中文字符串。

3.应用程序中的每个实例都要使用独立的文件名。这是为了防止在一个应用程序中同一时刻运行多个实例时,临时文件的相互覆盖.可以使用 GetTempFilename 函数来产生独立的临时文件名。

4.使用消息框或非系统模式的错误消息框之前应关闭所有已打开的文件。这是为了防止在使用上述消息框时,用户切换应用程序而引起的文件 I/O 错误。

三、Windows 环境中文件的操作方法

要在 Windows 进行文件操作,必须在程序开头使用下面两个包含语句,以使用 C 语言低级函数。

```
#indude<sys\types.h>
#indude<sys\stat.h>
```

1.创建新文件。创建新文件时,要利用 OpenFile 函数。在指定了文件名、类型为 OFStruct 的缓冲区之后,选择 OF-Create 任选项即可创造一个新文件,并返回该文件的句柄。例如:要创建一个名为 MPL.TXT 的文件,可按如下方式进行:

```
int WJJB;
OFStruct PYJG;
WJJB=OpenFile("MPL.TXT,PYJG,OF-EXIST");
if(WJJB>=0)
  {
   wAction=MessageBox(hWnd,(LPSTR)"File exists,
        Overwrite?",(LPSTR)"File",MB-OKCANCEL);
   if(wAction)=IDCANCEL
    {
     /*结束此过程*/
    }
  }
WJJB=OpenFile("MPL.TXT",PYJG, OF-CREATE);
```

2.打开已存在的文件。可用 OpenFile 函数按读方式/写方式/ 读写方式打开已经存在的文件,它们的任选项分别是 OF-Read/OF-Write/OF-ReadWrite,如

```
WJJB=OpenFile("MPL.TXT",&PYJG,OF-ReadWrite);
```

是用读写方式打开文件 MPL.TXT。

若打开失败,句柄 WJJB 将是-1,可以显示文件找不到的对话框或提示用户输入文件名。

3.文件的读操作.打开文件之后,就可用 C 语言的低级函数库的函数来读文件.如下例:

```
char ch[64];
int string;
  ......
WJJB=OpenFile("MPL.TXT",&PYJG,OF-Read);
if(WJJB>=0)
 {
   string=read(WJJB,ch,64);
   close(WJJB);
 }
```

就是以读方式打开 MPL.TXT,并从中读出 64 个字节放到字符数组缓冲区中。

4.文件的写操作.写操作之前以写方式打开文件,用 write 函数写入文件,如:

```
WJJB=OpenFile("MPL.TXT,&PYJG,OF-WRITE);
if(WJJB>=0)
  {
   write(WJJB,ch,string);
   close(WJJB);
  }
```

5.重新打开文件。在任何时候,只要打开或创建了一个文件,OpenFile 函数就自动把该文件的路径名、文件指针的当前位置等信息保存在类型为 OFStruct 的一个结构中。要重新打开该文件,只需用带 OF-REOpen 选项的 OpenFile 函数,就可用该结构提供的信息打开该文件,如:

```
WJJB=OpenFile((LPSTR)NULL,&PYJG,
     OF-REOpen|OF-Read);
```

6.文件操作时的提示。重新打开文件时,可用 OF-PROMPT 选项提示用户插入正确的软盘。若用户只读方式重新打开,Windows 还将检查文件的日期和时间与文件第一次打开时是否相同,如:

```
WJJB=OpenFile((LPSTR)NULL,&PYJG,OF-PROMPT
      OF-REOpen|OF-Read);
```

7.检查文件状态。要检查当前打开的文件的长度/创建时间等文件信息,可用低级函数 fstat 来获得,它将有关信息写入一个类型为 stat 的结构中,如:

```
stat WJXXJG;
  ......
fstat(WJJB,WJXXJG);
WindowsWindows
```