

限制 Windows 应用程序 多个实例的方法

戴志远 (暨南大学)

在 Windows 中,不仅可以同时运行多个程序,而且可以同时运行一个应用程序的多个实例(拷贝)。Windows 系统允许应用程序有多个实例存在,各个实例共用该程序的代码段,但有各自的数据段。Windows 为每个实例提供一个唯一的实例句柄作标识。这种机制对于书写器这一类应用是很有好处的,用户可以使用它同时编辑几个文件。但是对于程序管理器要求只可能有一个实例的场合,就必须限制它使之只有一个实例在运行。在开发自己的 Windows 的应用程序时,也会遇到这样的情况。而一般有关 Windows 的书籍中给出的例子都是允许多个实例的。所以笔者给出在实际编程中的解决办法。与大家共同探讨。

在 Windows 的应用程序中,必须包括 WinMain()函数,它是 Windows 应用程序的入口,和 C 语言中的 main()类似。WinMain()函数由 Windows 系统调用,调用时共传给四个参数,其基本形式为:

```
int PASCAL WinMain(hinstCurrent,hinstPrevious,Lpsz
CmdLine,nCmd Show)
HINSTANCE hinstCurrent;
HINSTANCE hinstPrevious;
LPSTR lpszCmdline;
int nCmdShow;
```

第一个参数是当前实例的句柄(hinstCurent),第二个参数是前一个实例的句柄(hinstPrevious)。在没有其它实例的时候,Windows 将 hinstPrevious 参数设为 NULL。利用这一点可以用如下的方法阻止用户运行一个以上的实例。方法是启动应用程序时检测 hinstPrevious 参数,如果该参数为 NULL,则返回 Windows。语句如下:

```
if (hinstPrevious)
return NULL;
```

但是这样一来就无法启动应用程序的第一个实例,用户要切换到任务表中去启动第一个实例,很不方便。应用程序应该能够自动切换到第一个实例,只要在 Winain()函数中稍作修改即可,实现方法如下:

在检测到 hinstPrevious 不为 NULL 时,用 Find-

Window()函数查出该应用程序第一个实例的窗口句柄,若该窗口已被最小化(图标),则打开该窗口,然后用 SetActiveWindow()函数把该窗口设置成当前活动窗口,这样就可以转到第一个实例了。

要说明的是,FindWindow()函数查找与给定窗口类名和窗口标题匹配的所有窗口,它需要两个参数:第一个是指定查找窗口类名字符串的地址。如该参数为 NULL,则所有的窗口类名都匹配;第二个是指定查找窗口的窗口标题的字符串地址。如该参数为 NULL,则所有的窗口标题都匹配。比如后面的程序查找窗口类名为 "SampleWclass" 的窗口句柄。一般说来,不同的应用程序有不同的窗口类名,所以用窗口类名在这种情况下就可以区分不同的应用程序。

该方法使用 COMPAQ—425、433 一体机,AST486 等多种机器在 Windows3.1 下均运行正常,对系统没有任何影响。

```
int PASCAL WinMain(hinstCurrent,hinstPrevious,lpsz
Cmdline,nCmdShow)
HINSTANCE hinstCurrent;
HINSTANCE hinstPrevious;
LPSTR lpszCmdline; /* Windows调用入口 */
int nCmdShow; /* Windows调用入口 */
{
MSG msg;
HWND hWndPrev;
if (hinstPrevious){ /* 已有实例存在 */
hWndPrev = FindWindow("SampleWClass",NULL);
/* 查找指定窗口句柄 */
if (hWndPrev!=NULL){
if (isIconic(hWndPrev); /* 最小化? */
OpenIcon(hWndPrev); /* 恢复 */
SetActiveWindow(hWndPrev); /* 设成当前活动窗口 */
return NULL; /* 返回 */
}
}%
if(!InitApplication(hinstCurrent))
/* 主初始化—登记窗口类 */
return FALSE; /* 失败返回 */
if (!InitInstance(hinstCurrent,nCmdShow))
/* 实例初始化 */
return FALSE; /* 失败返回 */
while(GetMessage(&msg,NULL,NULL,NULL)){
/* 消息循环 */
if (!TranslateAccelerator(hWnd,hAccTable,&msg)){
TranslateMessage(&msg);
DispatchMessage(&msg);
}
}
return(msg.wParam);
}
```