

# 移动存储设备防遗忘提醒系统的设计与实现

邹建忠, 乔 榛, 蒯啸玲  
(上海海事大学, 上海 201306)

**摘要:**随着社会的发展,移动存储设备的使用不断向普遍化发展,而与之相伴发生的便是普遍发生的移动存储设备遗失现象。为了避免用户在公共计算机上使用移动存储设备后由于遗忘而造成丢失的普遍现象,基于MFC利用VC++6.0开发平台设计实现了可移动存储设备防遗忘提醒系统。该系统通过捕捉用户习惯性动作而给出相关提醒,以达到从根源上避免遗失情况的发生的目的。当移动存储设备与计算机保持连接状态时,该系统捕捉到计算机关闭程序、关机等“习惯性动作”,或检测到计算机长时间无操作时,会通过弹出对话框、发出提醒语音等方式提醒用户取回自己的移动存储设备。

**关键词:**MFC;计算机;可移动存储设备;提醒系统;系统开发

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2015)01-0159-05

doi:10.3969/j.issn.1673-629X.2015.01.036

## Design and Implementation of Reminder System for Avoiding Forgetting Removable Storage Devices

ZOU Jian-zhong, QIAO Zhen, KUAI Xiao-ling  
(Shanghai Maritime University, Shanghai 201306, China)

**Abstract:** With the development of the society, the use of mobile storage devices is developed to generalization constantly, and accompanied by is the common mobile storage equipment loss phenomenon. In order to avoid the phenomenon of loss caused by forgetting after users use mobile storage equipment on public computer, a reminder system of mobile storage device has been realized by means of a platform of VC++6.0 based on MFC. This system gives the reminding by capturing the users' routine actions, to achieve the purpose of avoiding the loss. When the removable storage device is connected to the computer, the system will capture the routine actions from the computer, such as shutting down programs and powering off, or the system detects that the computer has been under no operation for a long time, it will pop up a message-box or give out a warning voice to remind users of getting their removable storage devices back.

**Key words:** MFC; computer; removable storage devices; reminder system; system development

### 0 引言

随着社会的发展,移动存储设备的使用越来越普遍,而随之发生的U盘等设备的遗忘丢失现象也愈加频繁。多因使用者在电子阅览室、公共机房、网吧、多媒体教室等地方的计算机上使用后,离开时忘记将其取回而导致的丢失现象。由于大部分用户存储设备里面存储有个人信息、研究成果等重要资料,丢失造成的不仅仅是经济损失,更甚的是个人信息、研究成果的外泄。

市面上现存的有很多物品防丢提醒器,大多分为主机和子机两部分,主机装于所要防止丢失的物品上,子机携带在主人身上,当两者超过一定距离时就会有

提醒声提醒。这类防丢器存在携带麻烦、需充电等缺点,并且该方法不适合用于移动存储设备的防遗忘提醒上。

为防止移动存储设备遗忘丢失现象的普遍发生,文中旨在设计并实现基于MFC的移动存储设备的防遗忘提醒系统,使移动存储设备在使用的过程中不被遗失。通过分析遗失原因,归类出发生遗失情况的高频原因。捕捉使用者使用计算机和移动存储设备的常用动作,主要有关闭应用程序,关闭计算机,计算机长时间无操作。当用户关闭应用程序、关机以及长时间无操作时,系统会给使用者相应的提醒以避免此类遗忘丢失现象的发生。该系统可为避免人们财产的损

失,保护个人资料和成果,为移动存储设备的防遗忘提醒做出一定贡献,为人们的生活带来一些方便。

## 1 系统分析与设计

### 1.1 方案分析

为防止用户遗忘丢失移动存储设备,首先分析用户在离开计算机前的“习惯性动作”。一般情况下,使用者在计算机上使用可移动存储设备的工程中,这类情况经常发生:

(1)使用者使用完计算机,关闭某个当前正在使用的程序就离开电脑;

(2)使用者关闭计算机然后离开电脑,忘记移动存储设备仍与计算机保持连接;

(3)使用者没有做任何动作,没有取回自己的移动存储设备就直接离开了计算机。

综上所述,习惯性动作可总结为三种情况:关闭此前使用的程序、关机和无任何操作。该提醒系统就从这三个方面着手给予提醒,避免遗失。

(1)关闭应用程序时,如有可移动存储设备与计算机保持连接,可通过对话框文字提醒、语音提醒、蜂鸣声提醒等方式提醒用户。

(2)执行关机命令时,如检测到有可移动存储设备与计算机保持连接,可暂时强制阻止关机并通过对话框文字提醒、报警声提醒方式提醒用户。

(3)计算机长时间无操作(键盘和鼠标无操作)时,如检测到有可移动存储设备与计算机保持连接,可将相关遗失信息发送给指定 IP,网络管理员的主机服务器程序就会收到相应的提醒信息。

### 1.2 结构设计

(1)关闭应用程序的习惯性动作都会伴随着计算机“关闭一个应用程序窗口”。每个应用程序的窗口都有一个句柄,用 FindWindow (LPCTSTR lpszClassName, LPCTSTR lpszWindow-Name) 可得到指定程序窗口的句柄。窗口未关闭时,句柄值不为 NULL;窗口关闭时,句柄值则为 NULL。通过判断这个句柄值是否为 NULL,就可以断定该应用程序是否关闭。该系统检测到这个句柄值为 NULL 时,就认为该应用程序关闭了,这样程序就实现了捕捉该“习惯性动作”。但一般情况下,一个应用程序往往有多个窗口或者一个应用程序会同时有多个实例,没必要对每一个窗口的关闭都做出提醒响应,只需在用户关闭使用程序的最后一个界面时给出提醒。根据“同一个应用程序所生成的窗口具有相同的类名”就可以实现这一点,如果这一类窗口都不存在了,即可断定程序已关闭最后一个界面,这样的检测及给出的提醒更加有效且人性化。提醒方式为文字提醒和语音提醒。其工作流程图如图

1 所示。

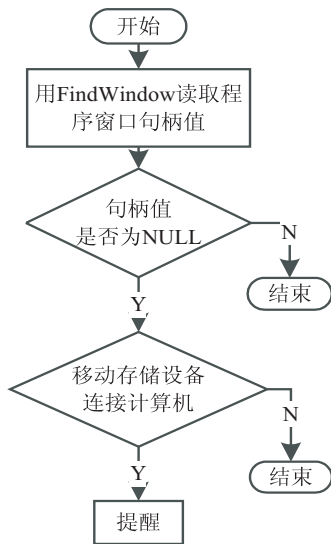


图 1 关闭程序时的工作流程图

(2)关机时,计算机系统会向所有正在运行的应用程序发出“WM\_QUERYENDSESION”消息,若所有程序接收消息后皆返回“TRUE”,则计算机执行关机操作,若有程序返回“FALSE”则不能完成关机。该系统在接收到信息后,若检测到仍有可移动存储设备与计算机保持连接,则会向计算机系统返回“FALSE”以阻止关机。

根据 MFC 的消息响应机制,通过如下函数捕捉消息: WindowProc (UINT message, WPARAM wParam, LPARAM lParam)。其中 Message 即为系统接收到的消息。

该系统接收到此消息时,若检测到有可移动存储设备与计算机连接,只要判断 WM\_QUERYENDSESION 与 message 是否相等即可。如果相等就返回 FALSE 阻止关机,并给出对话框文字提醒,播出语音提醒,强制关机时可设置蜂鸣器发出警告声以提醒。有时候移动存储设备可能是光盘,该系统可选择添加弹出 CD 托盘的提醒方式以提醒用户取出光盘。其工作流程图见图 2。

(3)计算机长时间无操作,通常表征为键盘和鼠标长时间无操作。捕捉长时间无键盘操作,需用钩子(Hook)编程,用 SetWindowsHookEx 函数设置全局的键盘钩子,以此来判断是否有键盘输入。捕捉鼠标长时间没有移动,需利用 GetCursorPos (int x, int y) 来获得当前鼠标屏幕坐标位置。若键盘和鼠标超过设定的时间后皆无任何操作,时间已超出用户设定值,则系统会判定使用者已经离开电脑,即向已设定的 IP 发送遗失消息。时间可以由使用者自行设定,且在弹窗提醒时,可以点选不再提醒,则系统就不会再进行提醒,这样可以避免打扰使用者观看电影等。其工作流程图如图 3

所示。

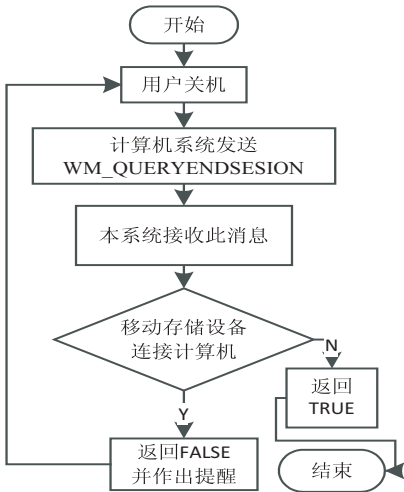


图2 关机时的工作流程图

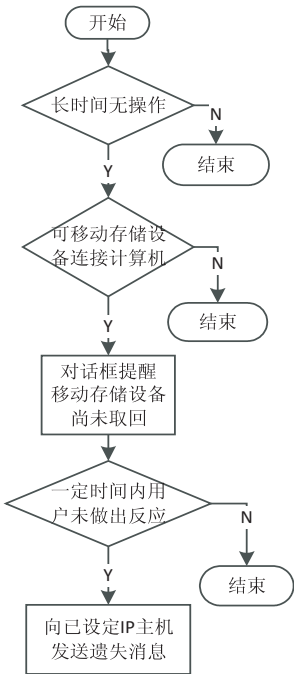


图3 长时间无操作时的工作流程图

2 系统开发与实现

2.1 方案开发

2.1.1 线程同步

由于该软件有多种提醒,每个提醒都要消耗一定的时间,如果串行执行这些提醒的话,各种提醒会依次进行,这样会影响效率,降低提醒效果。利用线程同步<sup>[1]</sup>可使提醒并行执行,各种提醒会在同一时间进行,以此达到增强效果的目的。

利用线程函数创建线程<sup>[2]</sup>:

CreateThread ( LPSECURITY \_ ATTRIBUTES  
lpThreadAttributes, SIZE \_ TdwStackSize, LPTHREAD \_  
START\_ ROUTINE lpStartAddress, LPVOID lpParameter,

DWORD dw- CreationFlags, LPDWORD lpThreadId)  
//lpStartAddress 为线程函数的入口地址

2.1.2 最小化于托盘

利用已经封装好的类 CTrayIcon 实现最小化于托盘后台运行,首先用 CTrayIcon 实例化一个对象, m\_ TrayIcon, 利用其成员如下函数创建对象:

Create ( CWnd \* pWnd, UINT uCallbackMessage, LPCTSTR szToolTip, HICON icon, UINT uID)

用 SetIcon ( IDI\_ MYICON) 定义其图示,最后用 ShowIcon ( ... ) 显示托盘图标。

2.1.3 网络通信

两台计算机之间进行通信(TCP/IP 协议),要指定其中一台为服务器(server),另一台为客户端(client)。通信时服务器首先设定自己的端口,客户端根据服务器的IP和服务器设定的端口进行数据传送。该系统利用MCF里已封装好的类 CSocket 完成两台计算机之间的通信。

服务器的设定:第一步,用 AfxSocketInit( NULL) 初始化 WinSock 环境;第二步,调用 CSocket 的 Create( int nPort) 函数创建对象,其中 nPort 为服务器所设定的端口;第三步,调用 Listen( int nBlocks) 使端口处于监听状态;第四步,调用 Accept( CSocket sockConnect) 等待客户端的连接;第五步,在数据到达时调用 Receive ( void \* lpBuf, int nBufLen, int nFlags=0) 接收数据。

客户端的设定:第一步,用 AfxSocketInit( NULL) 初始化 WinSock 环境;第二步,在调用 CSocket 的 Create ( int nPort) 函数创建对象,其中 nPort 为任意可用端口;第三步, Connect ( LPCTSTR lpszHostAddress, UINT nHostPort); 第四步, Send( const void \* lpBuf, int nBuf- Len, int nFlags=0) 发送数据。

2.1.4 软件的开机自启动与配制信息保存

在注册表项的 HKEY\_ LOCAL\_ MACHINE \ \ Soft- ware \ \ Microsoft \ \ Windows \ \ CurrentVersion \ \ Run 中添加以应用程序完整路径为键值的子键,这样就可以实现开机自动启动。先用 RegCreateKeyEx 创建注册表句柄,然后调用 RegSetValueEx 写入应用程序的路径信息。

用同样的方法可以将部分配制信息写入注册表中相应的位置,还有一部分配置信息用配置文件存储起来, WritePrivate\_ ProfileString 把一串字符写入配置文件,相应的用 GetPrivate\_ ProfileString 从配置文件中得到相关的配置信息。

2.2 系统实现

2.2.1 关闭应用程序时的提醒

关闭应用程序最后一个窗口时,如检测到有可移动存储设备与计算机保持连接,系统将通过对话框文

字提醒、语音提醒、蜂鸣声提醒等方式提醒用户。关键代码<sup>[3]</sup>和功能实现如下:

```
void CWDeviceDlg::MsgWaZqy0212
(WPARAM wp,LPARAM lp)
{
    int t_nDevice=wGetDeviceInfo();
    POSITION pos=(POSITION) wp;
    CString strtmp;
    m_listCareProg.GetAt(pos).m_nLife=0;
    CStatusBarMsgWnd * m_PopMessage = CStatusBarMsgWnd::
CreateObject(
    _T("NULL!!"),
    .....
    if(t_nDevice==DRIVE_CDROM)
    strtmp+="U 盘\\n\\n";
    strtmp+="还没取走\\n 请留意!";
    m_PopMessage->PopMsg(strtmp);
    m_listCareProg.GetAt(pos).m_bExist=0;
}
```

### 2.2.2 关机时的提醒

关闭计算机时,如检测到有可移动存储设备与计算机保持连接,系统阻止关机并通过对话框文字提醒、语音提醒等方式提醒用户。关键代码<sup>[4-5]</sup>和功能实现如下:

```
void CWDeviceDlg::wCopeShutdown()
{ int rst,t_nDevice=wGetDeviceInfo();
  CString str;
  if(t_nDevice==DRIVE_REMOVABLE)
  { if(m_nSndCheck!=0)
  {
    str=m_sPathInstall;
    str+="\\01";
    SendMessage(MSG_SOUND);
    .....
    rst=MessageBox("你要关闭计算机[6-7],但还有(*可移动存盘*)与电脑连接,单击\\是\\继续关机!","温馨提示!",
    MB_YESNO);}
```

### 2.2.3 长时间无操作时的提醒

计算机长时间无操作时,如检测到有可移动存储设备与计算机保持连接,系统会提醒用户。在用户仍无任何操作的情况下,系统会向设定好的 IP 地址发送提醒消息,且从主机可查看遗失信息。由于要捕捉全部(包括客户区以外键盘消息),所以要用到钩子(Hook)编程,用 SetWindowsHookEx(…)函数设置全局的键盘钩子,生成 lib 文件和 dll 文件,为了使所有的进程共同使用一个全局变量,这时就必需创建一个结。程序开始关键代码以及功能实现图如下:

```
#pragma data_seg("sec") //sec 为创建结的名称
HWND g_hWnd=NULL;
```

```
#pragma data_seg()
#pragma comment(linker,"/section:sec,RWS")
```

### 2.2.4 桌面窗口的枚举

每个程序有打开多个窗口的情况,只需要对同一程序的最后一个窗口的关闭作出反应,这就需要对桌面窗口进行枚举。程序段如下:

```
.....
CWDeviceDlg * pThis=(CWDeviceDlg *)lpParameter;
int t_nDevice;
POSITION pos;
int i;
CWinFom winTmp;
while(1)
{
    Sleep(500);
    t_nDevice=pThis->wGetDeviceInfo();//tzb
    int t_bShutdown=pThis->shutdown;
    if(t_bShutdown==1)
    .....
    return 1;
}
```

### 2.2.5 系统的设置

针对捕捉关闭程序的工况,系统可选择关注程序,也可屏蔽掉生命值小于一定时间的程序。关注程序指用户经常使用的程序,且认为是需要程序对其动作进行实时捕捉的程序;生命值小于一定时间指打开的程序未达到设定的时间长度就关闭,该系统就不会向用户提醒。针对长时间无操作的工况,系统可设置多长时间无操作即发送遗失消息,也可设置接收消息的 IP。该程序配有多种提醒方式,分别为文字提醒、语音提醒、蜂鸣器提醒和弹出 CD 托盘提醒。有用户可以按自己喜好以及易发生遗失的使用习惯来选择提醒方式<sup>[7]</sup>。

系统的设置界面如图 4 所示。

## 3 结束语

文中详细介绍了基于 MFC 的移动存储设备的防遗忘提醒系统的开发过程。基于 MFC<sup>[8]</sup>利用 VC++ 6.0<sup>[9-11]</sup>开发平台实现了可视化文字、语音等多种方式的防遗忘提醒,在移动存储设备的使用过程中避免因使用者的疏忽而造成的遗失现象。该系统已在高校公共计算机机房完成试验测试。通过一段时间的使用,经过对使用同学以及遗失 U 盘、手机等同学的调查,得知该系统可大大降低用户使用公共计算机后遗忘移动存储设备现象的发生概率,同学们遗失移动存储设备的情况经常发生,苦于无预防遗失的措施,对这款系统以及同类系统有较强的使用意向和渴望。



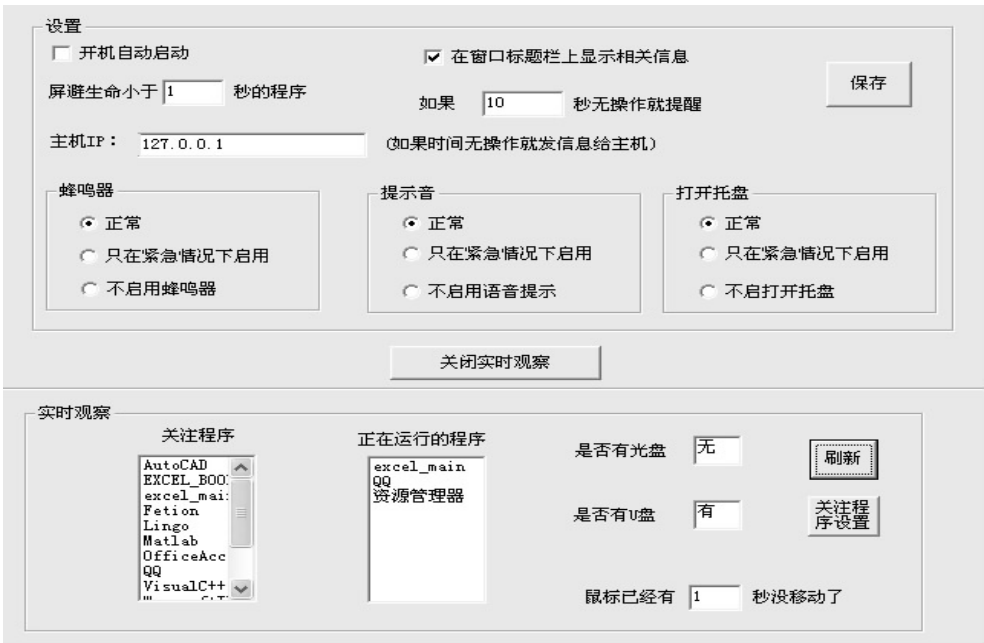


图4 系统的设置界面

该系统实现方法简单、反应敏捷、多样化提醒、目的性强、可靠性高、占用内存小,很适合运用于网吧、多媒体教室、电子阅览室、公共机房等场所。该系统不仅可以单独使用,还可以以插件的形式与杀毒等软件搭接以投入普遍的使用。软件安装包小,安装迅速且占用计算机内存较小。

对于以后的发展动向,该移动存储设备防遗忘提醒系统可通过在现有的 USB 数据插头中植入芯片来进行实现,基于单片机语言来实现一种具有防遗忘提醒功能的新型 USB 数据接头,使用该类插头即可起到防止移动存储设备遗忘丢失的情况发生。同时可将其植入存储设备内部,使其本身就带有防遗忘提醒的功能,可制作出配有该功能的 U 盘和移动硬盘等,以使其使用走向实物化、产品化<sup>[12-14]</sup>。

参考文献:

[1] 卢亮. Java 线程同步原理探析[J]. 高等函授学报:自然科学版,2012,25(4):44-45.  
[2] 李永中. 基于 VC 的多线程编程技术及其实现[J]. 微机计算机信息,2002(7):26-28.

[3] 谭浩强. C++程序设计[M]. 北京:清华大学出版社,2011.  
[4] 谭浩强. C 程序设计[M]. 北京:清华大学出版社,2010.  
[5] 孔垂柳. VC++中 MFC 框架技术探索[J]. 吉林建筑工程学院学报,2009,26(4):82-84.  
[6] Prosis J. MFC Windows 程序设计[M]. 北京:清华大学出版社,2007.  
[7] 徐璇,姜明新,黄静,等. 基于 MFC 的工程软件界面设计[J]. 电子设计工程,2011,19(21):11-13.  
[8] 侯捷. 深入浅出 MFC[M]. 武汉:华中科技大学出版社,2000.  
[9] 辛长安,王颜国. Visual C++权威设计[M]. 北京:清华大学出版社,2007.  
[10] 裘宗燕. Visual C++ 6.0 程序设计与开发技术大全[M]. 北京:人民邮电出版社,2004.  
[11] 方可燕,张双民,徐振明. Visual C++ 6.0 实战与精通[M]. 北京:清华大学出版社,2000.  
[12] 孙权,张显峰. 区域生态环境空间信息服务系统的设计与实现[J]. 计算机应用与软件,2012,29(1):191-195.  
[13] 邵东青,焦文英,陈海飞,等. 一种内容可定制的数据采集系统的设计与实现[J]. 计算机应用与软件,2012,29(1):181-183.  
[14] 张茂军. 虚拟现实系统[M]. 北京:科学出版社,2001.

# 移动存储设备防遗忘提醒系统的设计与实现

作者：[邹建忠](#)，[乔榛](#)，[蒯啸玲](#)，[ZOU Jian-zhong](#)，[QIAO Zhen](#)，[Kuai Xiao-ling](#)  
作者单位：[上海海事大学, 上海, 201306](#)  
刊名：[计算机技术与发展](#)  
英文刊名：[Computer Technology and Development](#)  
年，卷(期)：2015(1)

引用本文格式：[邹建忠](#). [乔榛](#). [蒯啸玲](#). [ZOU Jian-zhong](#). [QIAO Zhen](#). [Kuai Xiao-ling](#) [移动存储设备防遗忘提醒系统的设计与实现](#) [期刊论文] - [计算机技术与发展](#) 2015(1)