网络考试系统中防止U盘作弊方法研究

严 强

(苏州市广播电视大学,江苏 苏州 215004)

摘 要:目前越来越多的高校、教学培训机构和各种社会考试,采用了网络考试、网络在线考试等网络考试形式的无纸化考试。然而许多考试系统自身尚不具备防止考生使用U盘等移动存储设备作弊的功能。本文所涉及的系统能完善这样的功能,考生一旦插入了U盘或其他移动存储设备后,系统立即显示一个特殊的界面锁住屏幕,禁止一切可能的键盘操作。这种状态将一直保持到监考老师在该考生的考试机上解锁为止。

关键词:网络考试;客户端;U盘作弊;USB端口监视

中图分类号:TP319 **文献标识码**:B **文章编号**:1674-7720(2011)02-0012-03

Network examination system to prevent U dish cheating method research

Yan Qiang

(Suzhou Radio and TV University, Suzhou 215004, China)

Abstract: More and more colleges and universities, teaching and training institutions and various social examination, using a network test, online examinations and other forms of network test paper examination. However, many examination candidates to prevent the system itself does not have the use of U disk and other removable storage devices to cheat function. This system is involved in the function to accomplish this, the candidate once inserted U disk or other removable storage device, the system displays a particular interface immediately lock the screen to prohibit all possible keyboard. This state will remain the invigilator in the examination of the candidates until the machine unlocked.

Key words: Network examination; client; U disk cheating; USB port monitoring

无纸化考试系统的模式通常有单机版模式、C/S模式和B/S模式三种,无认哪种模式均要有客户端,而考生必须在客户端上完成其考试内容。目前USB存储设备特别是U盘的存储容量大、存取速度快、体积小、价格低廉,普及程度也越来越高,在考试中使用U盘作弊现象时有发生。由于考生在考试过程中使用U盘的隐蔽性极高,监考老师很难发现。目前大多数考试系统的客户端并没有对USB端口进行管理的功能。“网络考试客户端USB端口监视系统”(后文简称系统)就是实现在考试过程对考试客户端的USB端口进行全程监视,一旦客户端有USB设备接入,系统即刻锁屏、封锁键盘,发出使用USB设备的提示警告。本系统经过我校进行的“试点高校网络教育部分基础课程统一考试”多次使用实验表明,该系统在考试过程中能有效地监控USB端口,杜绝了考生使用U盘的现象,从而保证考试的公正、公平和权威性。

1 系统功能要求

(1)系统要能识别从计算机任何一个USB端口接入的移动存储设备,如U盘等设备。对于非存储类的USB设备,如:USB接口的打印机、扫描仪等,不能误判作移动存储设备。

(2)锁屏。一旦有U盘接入,系统能立即响应,并做出相应的处理。用一个无标题、无边框、无关闭按钮、最大化显示的窗体覆盖在所有窗体和任务栏的前面实现锁屏。

(3)封锁键盘。封锁键盘就是使用户键盘上的键失效,尤其是一些特殊功能组合键,如:Ctrl-Alt-Delete、Alt-Tab、Alt-Esc、Alt-F4、Windows-Key。

(4)关闭任务栏和开始菜单。

(5)封锁状态解锁。系统有提供给系统管理员或监考人员解锁的功能,解锁时要输入特定的解锁密码。

(6)解锁密码管理功能。系统提供由系统管理员使用

的密码维护、修改、重置和忘记密码的处理功能。

2 系统总体设计

2.1 系统组成

系统主要由 USB 端口监视系统和密码维护系统两部分组成。其组成结构如图 1 所示。

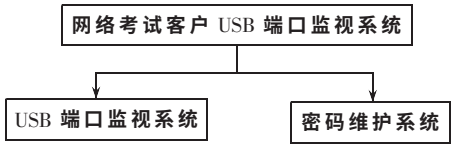

图 1 系统总体架构图

2.2 USB 端口实时监视系统组成结构

USB 端口实时监视系统由核心模块、USB 端口监视模块、屏幕、键盘加锁模块、键盘解锁模块、记录/读取系统 USB 使用状态模块、密码解锁处理模块和重启处理模块组成。其组织结构如图 2 所示。

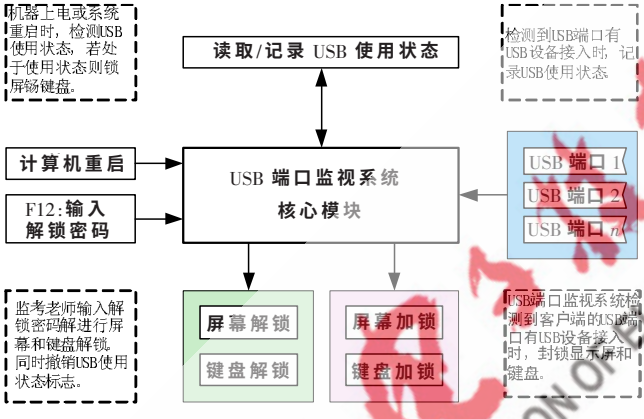

图 2 USB 端口实时监视系统结构图

2.3 密码维护系统组成结构

密码维护系统由密码修改和使用初始密码两部分组成。如图 3 所示。

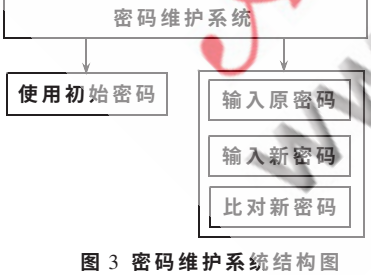

图 3 密码维护系统结构图

密码修改提供给管理员修改维护解锁密码。遗忘解锁密码时，先使用初始密码功能将解锁密码设置为初始密码，然后再修改成指定密码。

该系统为一个相对独立的子系统，由系统管理员掌管使用。

3 USB 端口监视系统设计

为了不和考试系统争抢资源，系统被设计成后台运行方式，以中断方式获取机器 USB 端口状态变化。采用进程保护技术进行系统自我保护，使系统在运行时不被考生强行终止，提高了系统的安全性。当考生在考试过程中插入 U 盘时，系统自动弹出锁屏窗体，并在窗体中显示考生已使用 U 盘的提示信息。锁住键盘，记录 U 盘的状态信息，即使考生拔掉 U 盘，系统自锁也不撤消。若考生用 RESET 开关强行重启机器或强行关机后再开机，当 Windows 系统启动完成后，系统仍然处于自锁状态。只有当监考老师用解锁命令和解锁密码解锁后，系统才恢复到正常状态。

3.1 USB 端口监视模块设计

USB 端口监视模块是系统的核心模块之一，也是系统的重要模块。设计流程如图 4 所示。

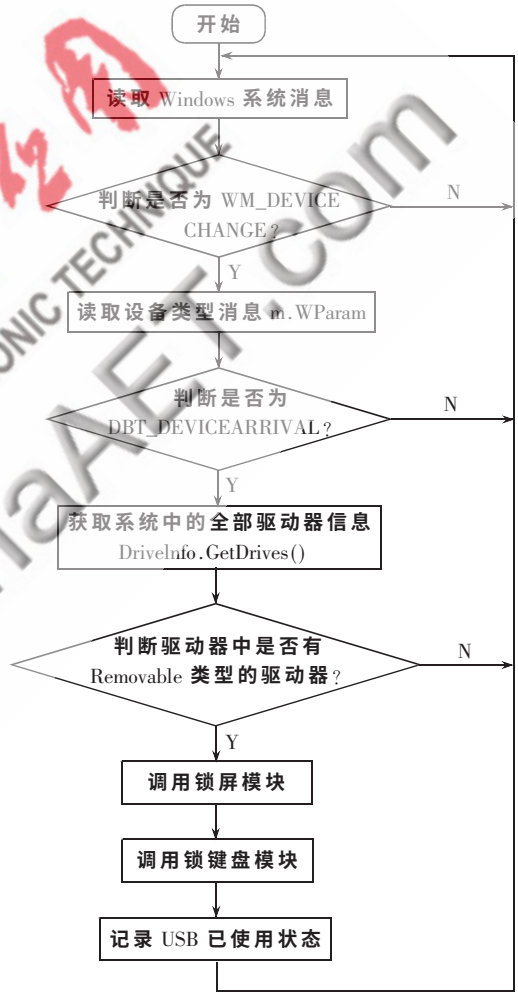

图 4 监视 USB 端口模块流程图

系统运行后，模块即进入工作状态，为了尽量少占用系统资源，模块以中断方式获取机器 USB 端口状态变化，一旦发现端口接入 USB 移动存储设备随即进行处理。

当发现端口有 USB 设备接入后，随即读取此设备的设备类型特征码。根据其设备类型特征码判断此设备是否是 USB 移动存储设备。如果是移动存储设备，则调用

其后续模块。否则，不做处理。

3.2 锁屏设计

锁屏模块是使系统呈现在考生面前的模块。设计流程如图5所示。

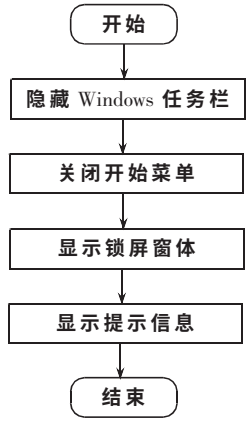

图5 锁屏模块流程图

模块被调用后，首先隐藏任务栏，关闭开始菜单，然后以白色、无边框、无标题栏、无关窗体控制按钮的窗体形式显示在所有窗体的最前面，遮盖住Windows的任务栏，并在窗体中央显示“正在使用USB设备... 系统自锁。请与监考老师联系”字样，提示考生由于插入了U盘从而系统自锁了。

3.3 锁键盘设计

锁键盘模块的执行结果呈现在考生面前的是键盘处于失效。本模块被调用后，将封锁键盘上的所有字母键、功能键和组合键，唯一开放的键是F12键，是输入解锁密码的入口键。

其功能是：

(1)封锁键盘上的所有单键，只留F12键。

(2)封锁Ctrl-Alt-Delete组合键，禁止打开Windows任务管理器。

(3)封锁Windows-Key(微软键)，禁止打开开始菜单。

(4)封锁Alt-F4组合键，禁止用此键关闭本程序。

(5)封锁Alt-Tab、Alt-Esc组合键，禁止切换任务窗体。

在封锁键盘模块设计中使用到了钩子(Hook)技术。系统使用键盘钩子截获键盘消息，在钩子函数中判断键盘消息中的键是否需要屏蔽键，如果需要屏蔽键，则不将此消息发送到窗体，从而达到屏蔽键盘的目的。由于系统需要屏蔽的键中有Windows-Key，而使用普通的键盘钩子不能捕捉到Windows-Key键。所以系统使用了底层键盘钩子，即全局键盘钩子。

安装键盘钩子：

```
[DllImport ("user32", EntryPoint = "SetWindowsHookExA", CharSet = CharSet.Ansi, SetLastError = true, ExactSpelling = true)]
public static extern int SetWindowsHookEx(int idHook, // 钩子的类型，即其处理的消息类型。
GlobalKeyboardProcDelegate lpfn, // 如果dwThreadId参数为0或是一个由别的进程创建的线程的标识，lpfn必须指向DLL中的钩子子程。
int hMod, // 应用程序实例的句柄。标识包含lpfn所指的子程的DLL。
int dwThreadId);
```

其中：GlobalKeyboardProcDelegate是处理键盘钩子截获的键盘消息的代理函数。设计如下：

```
public int GlobalKeyboardProc (int nCode, int wParam, ref KBDLLHOOKSTRUCT lParam)
{
    bool keyEven = false;
    switch (wParam)
    {
    case WM_KEYDOWN:
    case WM_KEYUP:
    case WM_SYSKEYDOWN:
    case WM_SYSKEYUP:
    keyEven = (((lParam.vkCode == 0x09) && (lParam.flags == 0x20)) | // Alt+Tab
    ((lParam.vkCode == 0x1B) && (lParam.flags == 0x20)) | // Alt+Esc
    ((lParam.vkCode == 0x1B) && (lParam.flags == 0x00)) | // Ctrl+Esc
    ((lParam.vkCode == 0x5B) && (lParam.flags == 0x01)) | // Left Windows Key
    ((lParam.vkCode == 0x5C) && (lParam.flags == 0x01)) | // Right Windows Key
    ((lParam.vkCode == 0x73) && (lParam.flags == 0x20)) | // Alt+F4
    );
    break;
}
if (keyEven == true)
{
    return 1;
}
else
{
return CallNextHookEx(0, nCode, wParam, ref lParam);
}
}
```

卸载键盘钩子：

```
[DllImport ("user32", EntryPoint = "UnhookWindowsHookEx", CharSet = CharSet.Ansi, SetLastError = true, ExactSpelling = true)]
public static extern int UnhookWindowsHookEx(int hHook);
```

转到下一个钩子：

```
[DllImport("user32",EntryPoint="CallNextHookEx", CharSet=CharSet.Ansi, SetLastError=true,ExactSpelling= true)]
public static extern int CallNextHookEx (int hHook, int nCode, int wParam, ref KBDLLHOOKSTRUCT lParam);
```

4 系统自身保护及运行

由于考生在考试过程中的作弊现象时有发生，因此监考和作弊考生是一对不可避免的矛盾，相互存在着一

定斗智斗勇现象。系统在一定程度上也充当了监考的角色,因此它就需要具有一定的自身保护能力,而不被考生终止运行。系统采用如下几种保护:

(1)用户终止保护

系统被设计成一个后台运行系统,运行后不显示任何窗体,只在系统的托盘中显示一个图标,而且不提供右击弹出菜单和双击显示主程序窗体的功能。从而使考生不能直接终止本程序,实现终止保护。

(2)进程保护

采用了进程保护技术,使得在 Windows 的任务管理器中无法终止本进程。防止考生在使用 U 盘前先终止本进程。

(3)卸载保护

系统在第一次运行时会自动在添加/删除程序列表中找到自己的列表项,并将其隐藏起来,这样本系统就无法从 Windows 中卸载。

系统目前已经运行在本校进行的"试点高校网络教育部分基础课程统一考试"的客户端上,取得了良好的效果。该系统不但能运行在网络考试客户端上,也可以运行于各种无纸化考试系统的客户端上。系统目前还有一些不足之处需要改进,如目前只是单机版,过程状态数据没有记录,不利于监考老师的集中管理,还有待于升级成网络版。

参考文献

[1] 李英伟.USB2.0 原理与工程开发(第 2 版)[M].北京:国防工业出版社,2007

[2] 陈启美,丁传锁.计算机 USB 接口技术[M]. 南京:南京大学出版社,2003.

[3] 薛园园. USB 应用开发技术大全[M].北京: 人民邮电出版社,2007.

[4] 肖踞雄,翁铁成. USB 技术及应用设计[M].北京:清华大学出版社,2003.

(收稿日期:2010-10-21)

作者简介:

严强,男,1962 年生,工程师,主要研究方向:软件工程。