基于 Socket API 通信技术的计算机实验室管理系统开发*

汪 平
(重庆师范大学 数学与计算机科学学院, 重庆 400047)

摘 要 描述了一套能够监控和管理用户在实验室上机的软件系统的设计思想、设计原则、总体结构和实现技术。系统采用账号和密码结合方式管理用户上机,并在系统中设计了一套新的基于 Socket API 的通信方案和独特的进程防杀技术,克服了目前广泛使用的读卡管理系统存在的弊端,取得了良好的实验室管理效果。

关键词 实验室管理;套接字;API 钩子;消息钩子
中图分类号 :TP393.07 **文献标识码** :A **文章编号** :1672-6693(2005)02-0039-05

Development of Computer Laboratory Management System Based on Socket API

WANG Ping
(College of Mathematics and Computer Science, Chongqing Normal University, Chongqing 400047, China)

Abstract: This paper describes the design method, principle, architecture and implementation technique of a software system that can watch and manage the users who practise in computer laboratory. The system uses account and password to manage the users and in this system a new scheme of network communication and special technique of non-terminating process are designed. It can get rid of the defects of card reader management system, and improve the management efficiency of computer laboratory.

Key words: laboratory management; socket; API hook; message hook

随着高校招生规模的扩大和计算机使用的不断普及,越来越多的人到计算机实验室进行上机实践,这对实验室的管理提出了新的要求。使用传统的人工管理,效率低下且很难实现合理的实验室资源分配,给实验室管理人员带来很大的工作难度的同时,也不方便用户上机。因此,目前很多实验室采用了读卡上机的方式进行管理,虽然提高了工作效率,但也存在不少问题:① 需要购买一定数量的读卡设备和专用于读卡的计算机,增加了实验室费用开销,若没有足够多的读卡设备,将会使大量的用户在上机之前等待读卡,造成"读卡瓶颈"现象,担搁用户上机时间;② 大量用户上机时系统不稳定;③ 对监控程序的保护措施不足,很容易被强行结束或绕过,达不到监控效果。

笔者在总结实验室管理经验的基础上,着眼于解决以上问题,开发了本系统。该系统有以下特色:① 采用账号与密码结合的方式登录系统,不需要读卡,解决了"读卡瓶颈"问题;② 系统通信采用 Socket API,并对通信连接设计了一种新型存储管理方案,大大提高了通信效率和系统的稳定性;③ 设计了独特的进程防杀技术,使系统监控有效可靠。

以下将主要描述本系统的设计思想、总体结构、设计原则和包含在上述特色中的关键技术的实现。

1 系统总体设计

1.1 系统设计思想

实验室管理的基本特点是设计本系统的依据。通过对实验室日常管理经验进行分析,可总结出以下基本特点:(1) 用户上机的方式主要有以下 3 种,第一种方式是用户作为某个班的成员上机,他们必

* 收稿日期:2005-03-10
作者简介:汪平(1977-),男,四川凉山人,助理实验师,硕士研究生,主要研究方向为分布式实时系统、软件开发。

须按照统一的上机安排表上下机,这是最基本的方式;第二种方式是单个用户自由用机,上下机由用户自己安排,不受课表的限制;第三种是在特殊的场合下,用户按照教师或实验室管理员的安排上机。(2)无论以哪种方式上机,只要下机时间到,都必须强迫用户下机。(3)下机时需要对计算机使用情况做记录。这些特点可以通过编程来实现,处理流程如图1所示。

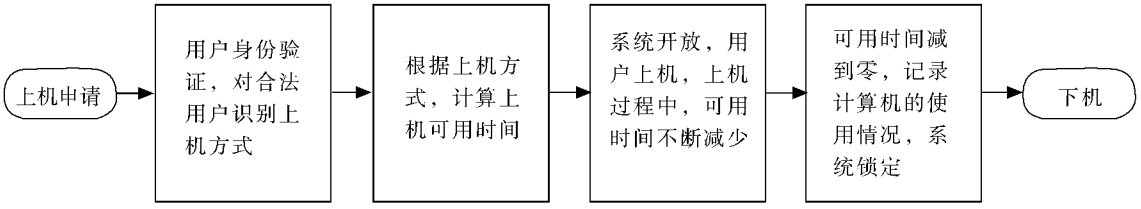

图1 用户上机流程

用户上机时,首先通过系统界面提出上机申请(即输入账号和密码),系统将进行身份验证,禁止非法用户上机并识别用户为哪一种方式上机;然后根据上机的方式计算出本次上机可用时间;接下来系统开放,用户正式开始上机,在上机过程中可用时间随时间的流逝而减少;一旦减到零,将记录本次计算机的使用情况,最后将系统锁定,用户下机。锁定和开放系统是实现上下机的关键,锁定系统是通过隐藏桌面、任务条和锁定键盘、鼠标实现的;相反,开放系统是通过显示桌面、任务条和对键盘、鼠标解锁实现的。

1.2 系统功能模块及相互关系

本系统由4个基本功能模块构成,分别为客户端、服务端、控制端、后台数据库。各个模块功能独立,并能通过网络实现控制信息或数据的交互。各个模块功能和控制信息(或数据)的交互关系如图2所示。

客户端安装在每台用户所使用的每台计算机

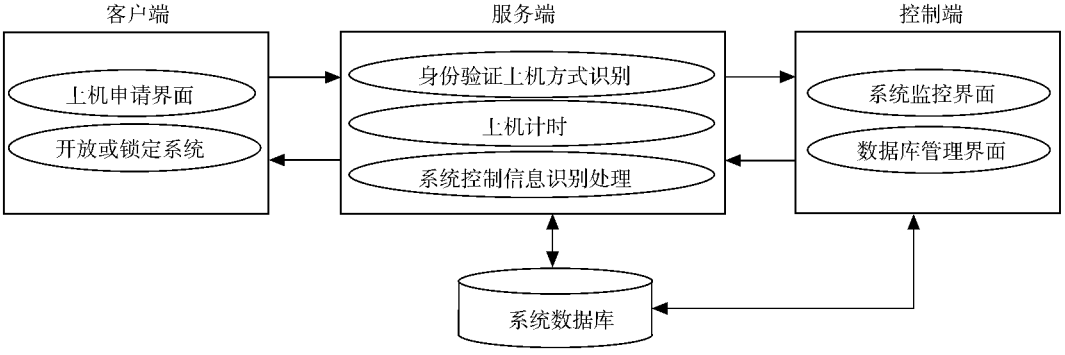

图2 系统功能模块及联系

上,它能根据服务端传来的控制信息执行相关程序段实现开放、锁定、关闭或重启计算机系统,达到用户上下机的目的;它提供了用户上机申请界面,为了方便用户使用系统,设计了3种登录方式而且不需要读卡,用户的上机申请信息通过网络发往服务端。

服务端一方面接收来自客户端的上机请求信息,依据系统数据库对用户做身份验证和上机方式识别,根据上机安排判断用户是否可以上机,然后通过网络向客户端发出开放或锁定系统的控制信息;另一方面也能接收来自于控制端的信息并对信息做识别处理,根据识别的结果向客户端发出控制信息,由此实现管理员对实验室所有计算机的统一控制(如强制用户下机、使计算机重启或关闭等操作);与此同时,服务端记录每个用户上机时间,定时判断是否该下机,如果下机时间到,向系统数据库写入计算机使用情况记录,同时向客户端发出下机信息,客户端接到信息后将锁定系统;服务端还要向控制端发送整个系统的当前状态信息,供管理员进行系统监视。

控制端安装在管理员的计算机上,完成对实验室所有计算机运行状态和用户上机情况的监视;可向服务端发出控制信息,服务端将这些控制信息转发到客户端,达到控制实验室中所有计算机的目的;它也是后台数据库的操作界面,管理员可在其中增删改管理员、用户、班级和计算机的基本信息,也实现排课、用户上机记录查询、计算机使用情况查询等

附加功能。

后台数据库存放了系统运行所必须的数据:管理员、用户、班级和计算机的基本信息、所有客户端的状态信息、用户用机日志和计算机使用情况日志等,供服务端和控制端读取或写入。

1.3　系统的设计原则

(1) 系统的各个模块的设计遵循单一职责(SOP)和开放封闭原则(OCP)[1],因而能够比较独立自由地扩展演化,每个模块独立完成一定的功能,可通过升级其中的一个模块来实现整个系统的升级,为系统的功能扩充带来方便。

(2) 各个模块之间通过网络通信进行数据和控制信息交换,网络通信采用 C/S 设计模式进行设计,采用的协议是目前广泛使用的 TCP/IP 协议,以增强软件的通用性。

(3) 按照面向对象的思想设计程序,核心代码可以重用,为系统的后续设计带来方便。

(4) 充分体现人性化设计要求,全面采用图形界面,界面直观、简洁流畅,对同一种处理设计了多种操作方式,供上机用户或管理员选择他们习惯的方式进行操作。

2　系统关键部分实现

2.1　系统通信实现

在系统的运行过程中,客户端与服务端、控制端与服务端之间要通过网络传送控制信息或数据,系统采用套接字编程实现。为了通信稳定和高效,在系统中没有采用 MFC 提供的 CSocket 类[2],也没有采用 Socket 控件[3],而是直接对 Socket API 以面向对象的方式进行了封装,形成了两个类:服务端套接字类 CLSocket 和客户端套接字类 CCSocket。在客户端定义一个 CCSocket 类对象,在服务端定义一个 CLSocket 对象,调用 Listen 函数使服务端处于监听客户端的连接请求状态,若监听到客户端的请求,则调用 Accept 函数接受请求,并在服务端动态生成一个 CCSocket 类对象与客户端 CCSocket 类对象进行信息交互,通信原理如图 3 所示。

每个连接都要在服务端动态生成一个 CCSocket 套接字对象,若同时有多个客户端连接到服务端,将在服务端产生多个套接字对象,如何存储这些对象是一个值得考虑的问题。常用的方法是采用链表[2],但在通信过程中经常进行链表查找,连接增多时会大大降低服务端的处理效率,因此本系统采用哈希表存储[4]。套接字中包含了 IP 信息,一般地,同一局域网内的计算机的 IP 最后一个字节是不相同的,因此可以构造这样的哈希函数:

H(套接字对象) =

套接字对象的 IP 的最后 1 个字节值

通过此函数产生的哈希地址冲突可能性很小。具体实现过程是:定义一个指针向量

CCSocket * pHashSocket[256];

每个分量的初始状态均为空指针。凡哈希地址为 i 套接字对象均存入 pHashSocket [i] 为头指针的链表中。以这种方式对套接字进行存储,在通信过程中几乎不查找链表,可以加快服务端的通信处理速度,减轻服务端的负荷。

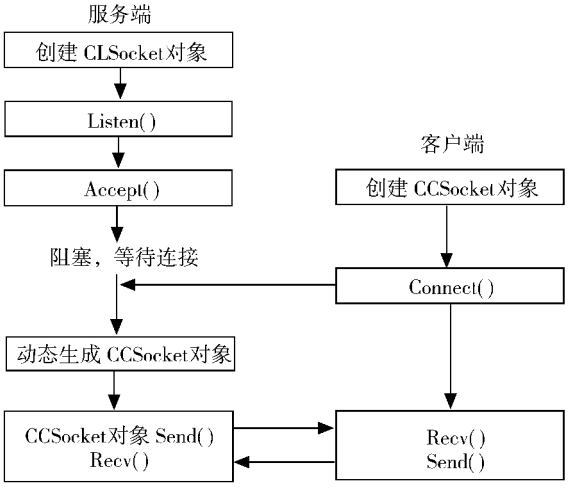

图 3　系统通信原理

2.2　客户端实现

客户端安装在用户使用的计算机上,对用户进行监控,并根据服务端传来的消息控制用户上下机,完善的客户端对于本系统来说具有很重要的意义。客户端应具有以下 3 个特征。

(1) 客户端随计算机的启动而启动。系统通过调用 API 函数 RegCreateKeyEx 将程序启动路径写入系统注册表的 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersions\Run 分支中实现。

(2) 客户端在计算机上一直运行,不被强制结束,否则起不到监视作用。这是本系统开发的一个难点,每个正在运行的程序在操作系统中以进程的方式存在,要使程序能长期在计算机上运行,就必须实现进程的隐藏或进程防杀(在任务管理器中看不见或不能结束进程),否则用户强行结束进程后将起不到监视的作用。在 WIN95 或 WIN98 中,使用

不公开的 API 函数 RegisterServiceProcessFun 可将普通进程设置为服务进程,达到隐藏效果。在 WIN2000 或 WINXP 中的进程是不容易隐藏的,在深入分析 API 钩子原理的基础上,实现了客户端进程防杀。API 钩子又称为 API 拦截,基本含义是:用已在程序中编写好的函数地址去取代系统 API 函数的地址,当对系统函数进行调用的时候,不立即执行系统函数,而是转向编写的函数去执行。API 钩子的实现已有多种成熟的技术,在本系统中采用的是 Jeffrey Richter 所提供的方案:修改 exe 文件的模块输入字节[5]。现在问题的关键是拦截什么系统函数,强行结束进程是由以下程序段完成的。

```
hProcess = OpenProcess(dwDesiredAccess,
        bInheritHandle, dwProcessId);
        TerminateProcess(hProcess,
        uExitCode);
```

真正能起到强行结束进程作用的是 API 函数 TerminateProcess。但拦截此函数达不到目的,因为它的第一个参数为句柄(hProcess)在不同的进程中有不同的值,因而实现不了跨进程拦截,达不到目的。进一步分析,在 TerminateProcess 调用之前,还必须调用 API 函数 OpenProcess 取得句柄,此函数有3个参数,其中第三个参数为进程标识(dwProcessId),它在系统范围内是唯一的,并且可通过 API 函数 FindWindow 查找到。因此,拦截 OpenProcess 函数可以实现进程防杀。具体设计思路是:设计一个新函数 MyOpenProcess,它的所有参数与 OpenProcess 相同,在其中调用 FindWindow 找到客户端进程标识,将其值与参数 dwProcessId 进行比较,若相等就不再执行 OpenProcess,从而 TerminateProcess 函数得不到正确的客户端进程句柄而不能将其强行结束,最后按照 Jeffrey Richter 所提供的方案用 MyOpenProcess 函数地址取代 OpenProcess 函数地址实现 API 拦截。

(3) 能开放或锁定系统,锁定系统的时候还需屏蔽 CTRL + ESC、ALT + ESC、ALT + TAB 等组合键。

该特征用消息钩子技术实现,消息钩子又称为消息拦截,基本含义是:每当特定的消息发出,在没有到达目标窗口之前,钩子函数(一段程序)先捕获消息,对其进行加工处理,然后继续传递,也可强制结束消息传递。消息钩子在使用前要调用 API 函数 SetWindowsHookEx 进行安装,为了能够在系统范围内对消息进行拦截,要将消息钩子写入到一个动态链接库中。在客户端中实现了对键盘消息的拦截,在钩子函数中对消息进行分析,若是 Alt + Esc、Ctrl + Esc、Alt + TAB 等组合键则结束消息传递,达到了屏蔽这些按键的效果。

2.3 后台数据库的设计

后台数据库的合理设计将为程序的设计、维护和升级带来方便。在系统设计之前,对系统所涉及的数据对象及其联系进行了全局性分析,得到如图4所示的 E-R 模型[6]。根据 E-R 模型向关系模式转换原则和关系模式的规范化要求,为系统数据库设计了以下数据表:管理员信息表、用户信息表、计算机信息表、班级信息表、管理员维护记录表、用户上机记录表、管理员管理用户记录表和上机安排表。通过对数据库的规范化设计,降低了控制端设计的复杂性,节约了系统开发时间。

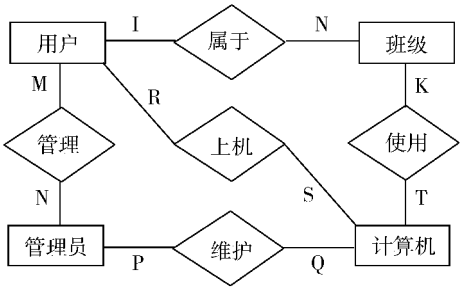

图4 系统数据的 E-R 模型

3 系统应用实例

在解决上述关键技术的基础之上,进一步完成了整套系统的设计并投入使用。图5为客户端的界面,用户可以选择3种方式之一登录,只要用户符合上机要求,系统将解除锁定状态,让用户使用计算机,不需要事先读卡,消除了"读卡瓶颈"现象。上机结束后客户端界面将自动弹出,系统锁定。

图5 客户端界面

图 6　控制端监视界面

图 6 展示了控制端的主界面,它由 4 个子窗口组成,左边窗口以树形方式列出了系统的所有功能,体现在这 4 个结点上:系统监视、用户账号、管理员和课程表及考勤。在左窗口中选中不同的结点会使右上窗口和右下的两个窗口发生改变,以实现不同功能的切换。当选中“系统监视”结点后,系统切换到如图 6 所示的界面,此时右上窗口显示正在上机用户的相关信息和计算机的使用情况,管理员可在其中选中用户或计算机,然后对其进行控制(如让用户下机、关机或重启系统),右下两个窗口用来显示用户上机记录和计算机使用记录。当选中“用户账号”结点后,系统切换到如图 7 所示的界面,此时右上窗口仅显示用户信息,管理员可对选中项的用户信息进行修改,实现用户管理,右下两个窗口用来显示用户上机记录。当选中“课程表及考勤”结点后,系统切换到如图 8 所示的界面,此时右上窗口显示上机安排信息,管理员可在这里增加、修改或删除上机安排,右下两个窗口用来实现上机考勤。

图 7　客户端界面

图 8　客户端界面

4　结束语

本系统在 Window2000 环境下,使用 Visual C + + 6.0 编程实现。系统分为四大模块并以网络通信进行信息交互,实现了各个模块的独立性,为功能扩展带来方便,也便于实现管理员对实验室计算机进行统一控制管理,对 socket API 封装和套接字对象存储方式的改进能使系统高效稳定地运行,在客户端引入进程防杀技术系统监视功能更为可靠。本系统的成功开发,实现了计算机实验室的自动化管理,方便了用户上机,也减轻了实验室管理人员工作强度,提高了工作效率。本系统为计算机实验室引入了技术管理的新理念,现已在重庆师范大学数学与计算机科学学院实验室投入使用,连续稳定地运行了两年,取得了良好的管理效果,现正向其它学校推广。

参考文献:

[1] 孙沛,陈世福. 面向公安部门的人脸识别系统的设计和实现[J]. 计算机科学,2004,31(9):183-185.

[2] 方娟,萧秋水. Visual C + + . NET 网络编程与互联网应用开发[M]. 北京:清华大学出版社,2002.

[3] 艾光利,马燕. 基于 Sockets 的计算机远程监控技术及实现[J]. 重庆师范大学学报(自然科学版),2004,21(1):33-37.

[4] 严蔚敏,吴伟民. 数据结构[M]. 北京:清华大学出版社,1996.

[5] Jeffrey Richter. Windows 核心编程[M]. 王建华译. 北京:机械工业出版社,2000.

[6] 王能斌. 数据库系统教程[M]. 北京:电子工业出版社,2002.

(责任编辑　李若溪)