

计算机远程控制的研究与实现

学 院： _____ 计算机学院 _____

专 业： _____ 计算机科学与技术 _____

姓 名： _____ 纪梓航 _____ 学 号： _____

指导老师： _____ 雷剑刚 _____ 职 称： _____ 讲师 _____

中国·珠海

二〇一四年四月

诚信承诺书

本人郑重承诺：本人承诺呈交的毕业设计《计算机远程控制的研究与实现》是在指导教师的指导下，独立开展研究取得的成果，文中引用他人的观点和材料，均在文后按顺序列出其参考文献，设计使用的数据真实可靠。

本人签名：_____

日期：_____年___月___日

计算机远程控制的研究与实现

摘 要

随着网络技术的飞速开展，远程控制协助软件越来越受到人们的重视。计算机远程控制技术是计算机系统管理人员在异地通过计算机网络连接目标计算机，由本地计算机对远程计算机进行管理和维护的行为。本课题设计是为适应远程控制及协助的要求，使远程控制提高到计算机的实时水平而设计的。远程控制包括多项内容，本课题设计只是承建了其中的一局部即：实时控制。本课题设计为一个通信应用程序，用到了多项技术，诸如：异步模式 socket、面向对象编程、软件工程思想、API HOOK 等。本文主要分析了远程控制系统的一些根本功能和组成情况，包括系统的需求分析、系统结构、功能模块划分分析等，重点对应用程序的实际开发实现作了介绍。到达了实时性和平安性，且应用程序功能完备。同时简单介绍了 Visual C++ 6.0 编程环境和 WIN SOCKET 的功能特点。

本系统采用 Visual C++ 6.0 作为开发工具，整个系统操作简洁、界面友好、功能灵活、实用，根本完成了远程控制中所需要到的主要功能。

关键字：套接字、面向对象、软件工程、远程控制

Research and Realization of computer remote control

Abstract

A web search engine is a software system that is designed to search for information on the World Wide Web, and to summarize and classify the information, then the search results are generally presented in a line of results often referred to as search engine results pages (SERPs). With the development of Internet, people need the search engine to become diversified. According to the different demands, the search engine has developed lots of different functions.

Now, the search engine includes initial content index, photo search engine according to the key words, a genuine full-text index of search engine, which showed the trend of personalization and diversification.

This paper elaborates the working principle of search engine, and to study and analyze.

Keywords : Object-Oriented Software Engineering socket Remote Control

1 前言	1
1.1 课题背景	1
1.2 目的以及意义	2
2 开发平台的理论根底	3
2.1 Microsoft Visual C++及编程模式简介	3
Visual C++的简介	3
2.1.2 MFC 应用程序框架	3
MFC 的消息映射	4
消息	4
消息的映射	4
2.2 系统架构的模式	6
C/S 模式即 Client/Server 结构模式	6
2.2.2 TCP C/S 模式的通信原理	7
C/S 结构的优点	7
2.3 Socket 通信技术介绍	8
2.4 CAsyncSocket 类的简单介绍	8
3 需求分析	10
3.1 系统根本情况描述	10
3.2 系统可行性分析	10
经济可行性	10
技术可行性	10
操作可行性	11
结论	11
3.3 功能需求分析	11
3.4 系统工作流程	12
3.5 系统结构模型	14
4 总体设计	15
4.1 使用工具	15
4.2 系统可行性分析	15
模块设计	15
客户端模块设计	15
效劳器模块设计	16
/FTP 效劳器模块设计	17
API HOOK 模块设计	17

5 系统详细设计.....	18
5.1 屏幕控制模块.....	18
5.2 文件操作模块设计.....	19
5.3 效劳器模块设计.....	21
5.4 客户端与效劳器连接设计.....	22
6 系统测试.....	23
6.1 软件测试根底.....	23
软件测试的目标.....	23
软件测试的常用方法.....	23
6.2 本系统采用的测试方法.....	23
连接测试.....	24
屏幕键盘鼠标传输测试.....	24
6.3 测试结论.....	25
7 结论.....	26
参考文献.....	27

1 前言

1.1 课题背景

远程控制一般支持下面的这些网络方式：LAN、WAN、拨号方式、互联网方式。此外，有的远程控制软件还支持通过串口、并口、红外端口来对远程机进行控制（不过，这里说的远程电脑，只能是有限距离范围内的电脑了）。计算机中的远程控制技术，始于DOS时代，只不过当时由于技术上没有什么大的变化，网络不兴旺，市场没有更高的要求，所以远程控制技术没有引起更多人的注意。但是，随着网络的高度开展，电脑的管理及技术支持的需要，远程操作及控制技术越来越引起人们的关注。传统的远程控制软件一般使用NETBEUI、NETBIOS、IPX/SPX、TCP/IP等协议来实现远程控制。

通常企业内部或者IT公司的客户技术支持部门都有技术支持业务，其任务是通过解答疑难问题，努力减少技术人员到现场效劳或者让用户把设备送到支持中心进行维护。远程控制，顾名思义，就是像操作自己电脑一样，去远距离操控另一台电脑。很显然，这样的系统，一般就是采用C/S模式或者B/S模式。网络应用程序的根本开发模式是C/S模式，其系统结构式指把一个大型的计算机应用系统变为多个能互为独立的子系统。效劳器是整个应用系统资源的存储与管理中心，多台客户机那么各自处理相应的功能，共同实现完整的应用。C/S模式一般基于传输层的TCP协议实现，主要特点是交互性强、具有平安的存取模式、网络通信量低、相应速度快、利于处理大量数据。远程控制软件一般分两个局部：一局部是客户端程序Client，另一局部是效劳器端程序Server，在使用前需要将客户端程序安装到主控端电脑上，将效劳器端程序安装到被控端电脑上。它的控制的过程一般是先在主控端电脑上执行客户端程序，像一个普通的客户一样向被控端电脑中的效劳器端程序发出信号，建立一个特殊的远程效劳，然后通过这个远程效劳，使用各种远程控制功能发送远程控制命令，控制被控端电脑中的各种应用程序运行，我们称这种远程控制方式为基于远程效劳的远程控制。

通常，技术支持必须依赖技术人员和用户之间的口头交流来进行，这种交流既耗时又容易出错。许多商业用户对计算机知之甚少，然而当遇到问题时，他们必须向技术人员提供故障情报及相关操作。在尝试解决问题时，技术人员可能指导用户执行一系列复杂的过程，而这些过程对用户来说或许完全不熟悉；如果用户不能正确的按要求操作，反而使问题恶化。此外，如果通过不能解决问题，那么在技术人员亲自到用户现场解决问题之前，计算机将无法继续使用，导致工作延误。

远程控制具有一下优点：便于用户能够在任何地方通过网络及时，快速的访问，控制自己的主机。特别是对于网络管理员，技术效劳人员来说，远程控制提供了一个便捷，高效的手段。

当然，远程控制软件是一把双刃剑，有优点就会有缺点，它存在比拟严重的平安隐患。一方面给用户和网络管理员工作带了的方便，另一方面也给一些居心不良者留下后门。形成很多木马程序。这就需要对软件在设计方面做一些平安考虑。

1.2 目的以及意义

本文正是在上文提到的背景下提出的，目的就是为了解决计算机的远程操作，降低企业对软件的后期维护本钱，设计出一款远程控制系统。远程控制系统能使技术人员直接操作远程计算机，就像操作本地机器一样，无须用户介入，技术人员技能得到该机器的问题的第一手材料，从而加快了问题的解决。实际上，使用远程控制工具的技术人员能够做到解答疑难问题，安装和配置软件，把软件下载到用户计算机上，配置应用程序和系统软件设置并可通过实际操作培训用户。总之，本课题的设计与实现具有很大的现实意义。

2 开发平台的理论根底

2.1 Microsoft Visual C++及编程模式简介

Visual C++的简介

Visual C++的资源编辑器能以所见即所得 (What you see is what you get) 的形式直接编辑程序的用户界面, 为所有资源分配 ID 标识号。ClassWizard 能把对话框模板与生成的类定义或与已有的类代码连接起来, 为菜单项、控制等资源生成空的处理函数模板, 创立消息映射条目, 并将资源 ID 与处理函数连接起来。通过使用 AppWizard, 程序员的编程工作便简化为用资源编辑器直观的设计界面, 完善对话框类代码, 在空的处理函数模板处填写响应用户操作的代码, 这是一种比拟完善的可视化编程方法。

但产品名“Visual C++”也容易误导人, 让人认为自己使用的是一个与 Microsoft Visual Basic 类似的完全可视化的系统。然而, 使用 Visual C++, 开发人员必须真正地阅读和编写 C++代码。Visual C++向导可以节省时间和提高精度, 但是, 程序员也必须理解向导产生的代码, 并且, 最重要的是, 还必须理解 MFC 库的结构和 Windows 操作系统的内部工作方式[7]。

MFC [Microsoft Foundation Classes] 应用程序框架

应用程序框架的一种定义是: 提供一般应用程序需要的全部面向对象软件组件的集成集合。

C++流行的一个原因是它可以用类库扩充。类库是可在应用程序中使用的有关 C++类的集合。应用程序框架是类库的超集。一般的类库只是一种孤立的类的集合, 用来嵌入在任何程序中, 但是, 应用程序框架却定义了程序的结构。自从 MFC 库发布以来, MFC 已经成为主要的 Windows 类库。使用 MFC 类库构建应用程序具有以下优点:

①MFC 库是 C++的 Microsoft Windows API。

②应用程序框架生成的应用程序使用了标准的结构, 具有标准化的用户接口, 这对具有标准用户界面的 Win32 程序来说, 可以极大的减轻程序员的负担, 使程序员不必过多地考虑界面, 可把主要精力放在程序设计上, 以提高程序设计的效率。

③使用应用程序框架的应用程序不仅小, 而且运行速度快, 具有很大的灵活性。MFC 封装了 Win32 SDK 中的几乎所有函数, 能实现 Win32 系统的任何功能。

④MFC 框架降低了编码的复杂性。

⑤MFC 库应用程序框架有丰富的特性, 如: Windows API 的 C++接口、通用的 (非 Windows 所特有的) 类、“共用根对象”类层次结构、流线式多文档界面 (MDI) 应用程序支持等。

⑥强大的功能。除封装了大局部的 Win32

SDK 函数外，MFC 还提供了应用程序本身的数据和操作及 ActiveX、OLE、Internet、WinSock、DAO (Data Access Objects)、ODBC (Open DataBase Connectivity) 等操作类。

MFC 框架的核心是文档/视图结构 (Document-View Architecture)，这是一个很好用、但又往往较难以入门的功能。简单的说，文档/视图结构就是将数据和对数据的观察或数据的表现 (显示) 相别离。文档仅处理数据的实际读、写操作，视图那么是显示和处理数据的窗口，视图可以操作文档中的数据^[7]。

MFC 的消息映射

在使用 VisualC++ 进行 Win32 程序设计时，消息映射是一个非常重要的概念。Windows 应用程序是消息驱动的，应用程序不能直接得到用户所做的操作，如鼠标按键、键盘输入和窗口移动等。这些操作由操作系统管理，操作系统检测到操作事件后，便向相关的应用程序发送消息，应用程序响应这些消息来完成用户的操作。

.1 消息

Windows 中的消息是操作系统与应用程序之间、应用程序与应用程序之间、应用程序各对象之间相互控制与传递信息的方式。

消息的根本格式如下：

```
Message wParam lParam
```

Message 是消息名称；wParam 是与消息相关的 Word 型参数；lParam 是与消息相关的 Long 型参数。

消息主要有以下 3 类。

Windows 系统消息：Windows 系统向窗口发送的消息，由窗口 (Window) 或视图 (View) 进行响应处理。这类消息包括除 WM_COMMAND 消息之外的名称以 WM_ 开始的其他消息。

控制通知消息：控制或子窗口传给父窗口的 WM_COMMAND 通知的消息。

命令消息：在响应用户接口操作时，将产生 WM_COMMAND 命令消息。其参数指定了用户接口的标识号，如菜单项和按钮等 ID 号。

.2 消息映射过程

在使用 AppWizard 创立应用程序时，MFC 应用程序框架设置了相应的消息处理函数来响应消息，以完成相应的操作。消息处理函数是某些类 (通常是窗口类) 的成员函数和程序员在其中编写响应消息时应进行操作的代码。

框架将消息和它们的处理函数连接起来就是消息映射。消息映射使应用程序在接收到消息时调用对应的消息处理函数来响应和处理消息。

ClassWizard 在创立新类时将其创立一个消息映射，并为每个类能响应的消息和命令增加对应的处理函数。在源代码中，消息映射开始于 BEGIN_MESSAGE_MAP 宏，结束于 END_MESSAGE_MAP 宏，中间由一系列预定义的被称为“条目宏”的宏组成。其根本格式如下：

```
BEGIN_MESSAGE_MAP (classname, parentclassname)
//{{AFX_MSG_MAP (classname)
    条目宏 1
    条目宏 2
    条目宏 3
    .....
//}}AFX_MSG_MAP
END_MESSAGE_MAP ()
```

其中 classname 为拥有消息映射的当前类名，parentclassname 为当前类的父类名。条目宏定义了类所处理的消息与其对应的函数。常用的条目宏类型如表 2.1 所示。

表 2.1 消息映射条目宏

消息类型	宏格式	说明
Windows 消息	ON_WM_XXXX	WM_XXXX 为 Windows 消息名
命令	ON_COMMAND (ID, Function)	ID 为命令标识号，Function 为处理函数名
更新命令	ON_UPDATE_COMMAND_UI (ID, Function)	ID 为命令标识号，Function 为处理函数名
控制通知	ON_XXXX (ID, Function)	ID 为控制标识号，Function 为处理函数名
用户定义消息	ON_MESSAGE (ID, Function)	ID 为消息标识号，Function 为处

		理函数名
	ON_REGISTERED_MESSAGE (ID, Function)	

用户注册消息		ID 为消息标识号，Function 为处理函数名
--------	--	---------------------------

Windows 消息的处理函数在 CWnd 类中进行了预定义，类库以消息名为根底定义这些处理函数的名称，且 MFC 要求所有消息处理函数声明为 `afx_msg` 类型。例如，消息 `WM_PAINT` 的处理函数在 CWnd 类中的声明如下：

```
afx_msg void OnPaint();
```

通过 ClassWizard 在派生类中用同样的原型定义处理函数并为该函数生成消息映射条目，然后由程序员编写处理函数代码，并在派生类中覆盖了其父类的消息处理函数。在有些情况下，必须在派生类的消息处理函数中调用其父类的消息处理函数，使 Windows 和基类能对消息进行处理。ClassWizard 将在生成的处理函数中建议是否应调用父类的消息处理函数及调用的次序。除此之外，用户定义和注册的消息、命令和控制通知都没有默认的处理函数，需要在定义时声明，一般根据其 ID 名称来为函数命名^[7]。

2.2 系统架构的模式

C/S 结构，即 Client/Server (客户机/效劳器) 结构，软件系统体系结构，通过将任务合理分配到 Client 端和 Server 端，降低了系统的通讯开销，可以充分利用两端硬件环境的优势。

2.2.1 C/S 模式即 Client/Server 结构模式

Client/Server 结构，它的开展经历了两个阶段：从两层结构到三层结构。

①两层结构：它由两局部构成：前端是客户机，通常是 PC，主要完成用户界面显示，接受数据输入，校验数据有效性，向后台数据库发请求，接受返回结果，处理应用逻辑；后端是效劳器，运行 DBMS，提供数据库的查询和管理。应用逻辑主要在前端，如在后端那么是存储过程的形式。

②三层结构：利用中间件将应用分为表示层、业务逻辑层和数据存储层三个不同的处理层次。三个层次的划分是从逻辑上来分的，具体的物理分法可以有多种组合。基于三层结构的应用系统不但具备了大型机系统稳定、平安和处理能力高等特性，同时拥有开放系统本钱低、可扩展性强、开发周期短等优点。而中间件作为构造三层结构应用系统的根底平台，提供了以下主要功能：负责客户机与效劳器间、效劳器间与效劳器间的联接和通讯；实现应用与数据库的高效连接。

提供一个三层结构应用的开发、运行、部署和管理的平台。

TCP C/S 模式的通信原理

TCP Client/Server 的通信原理如图 2-1 所示，效劳器端首先监听一个固定端口，客户端再连接到效劳端，此时效劳端执行 Accept 操作，以接受客户端的连接。此时连接创立成功，那么进行数据传输，待数据传输完毕，效劳端和客户端就断开连接。

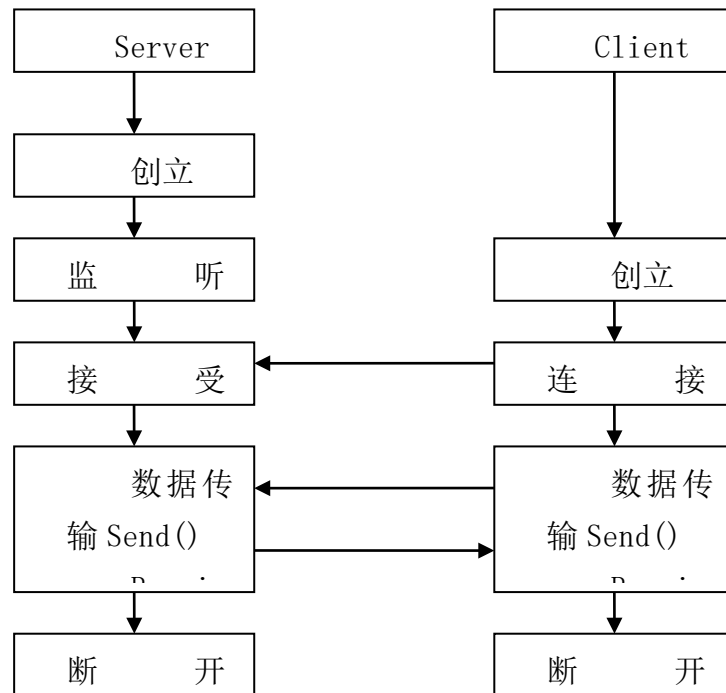


图 2-1 Client/Server 的通信流程

2.2.3 C/S 结构的优点

Client/Server 技术在目前程序开发中得到了广泛的应用，这种技术的优点在于它将处理工作按照一定的比例分配到客户端和效劳器上去执行，这样减少了网络传输的工作量，从而合理地利用了资源，提高了应用程序开发的效率。由于客户端实现与效劳器的直接相连，没有中间环节，因此响应速度快。

2.3 Socket 通信技术介绍

TCP/IP 的 Socket 提供流式套接字、数据报套接字、原始套接字三种。数据报套接字提供一种不可靠的，非连接的数据包(Packet)通信方式。在这里，“不可靠”

是指发送一个数据包不能获得担保,也不能保证数据包按照发送的顺序到达目的地。数据包以独立包形式被发送,不提供无错保证,数据可能丧失或重复,并且顺序混乱。在实现上,同一分组数据报可能不止一次的被发送。对于 Socket 的 TCP/IP 实现,数据报套接字使用用户数据报协议(UDP)。虽然在通常情况下,在同一台计算机或在轻负载的局域网所连接的两台计算机的进程之间进行通信时,可能不会出现数据包不被发送或没按照顺序到达及又重复发送的情况,但在编写应用程序时,应该注意检测意外发生的事件,应具备处理出现这些情况的能力。当然,如果为非常复杂的网络(如 Internet)编写通信应用程序,就应该考虑到数据报套接字的不可靠性。如果我们的应用程序没有处理好这个问题,它就有可能崩溃。尽管如此,数据报套接字在发送数据包或者记录数据时仍然有用。另外,数据报套接字还提供向多个目的地发送播送数据包的能力。

Socket 在计算机中提供了一个通信端口(ip:port)。通过这个端口,一台计算机可以与任何一台具有 Socket 接口的计算机通信。通信的根底是套接字。一个套接字是通信的一端,在这一端上可以找到与其对应的一个名字。一个正在被使用的套接字都有它的类型和与其相关的进程,套接字存在于通信域中。一个套接字通常和同一个域中的套接字交换数据(数据交换也可以穿越域的界限,但这时一定要执行某种解释程序)。应用程序在网络上传输、接收的信息都通过这个套接字来实现。在应用开发中就像使用文件句柄一样,可以对 Socket 句柄进行读写操作。

网络客户/效劳器模式的原理是一台主机提供效劳(效劳器),另一台主机接受效劳(客户机)。作为效劳器的主机翻开一个端口并进行监听,如果有客户机向效劳器的这一端口提出连接请求,效劳器上的相应程序就会自动运行,来应答客户机的请求。而 Socket 的正确使用,给设计者带来了极大的方便,使得各种应用程序能够在 Windows 环境下顺利进行各种网络通讯。

2.4 CAsyncSocket 类的简单介绍

CAsyncSocket 类是 MFC 对 WINSOCK API 的较底层封装,通过类名就知道这是一个异步非阻塞 SOCKET 类。

什么是异步非阻塞呢?举个例子:一位体育老师,需要测验 100 位同学的 400 米成绩。老师每隔 10 秒让一位同学起跑,直到所有同学出发完毕;另一边每有一个同学回到终点就记录成绩,直到所有同学都跑完。老师设计了两个函数,其中一个函数记录起跑时间和学生号,该函数会主动调用 100 次;另一个函数记录到达时间和学生号,该函数是一个事件驱动的 callback 函数,当有同学到达终点时,callback 函数会被动调用。老师主动调用的函数是异步的,因为调用它时,它并不会立刻返回;这个函数也是非阻塞的,因为一旦调用它,它就马上返回,不用等待就可以再次调用它。这就是异步非阻塞模式。

然而，最不容易被初学 Socket 编程的人理解的，也是本课题最要提醒的一点是，客户方在使用 `CAsyncSocket::Connect()` 时，往往返回一个 `WSAEWOULDBLOCK` 的错误（

其它的某些函数调用也如此，如：Send(),Receive()等)，实际上这不应该算作一个错误，它是 Socket 在提醒用户，由于使用了非阻塞 Socket 方式，所以〔连接〕操作需要时间，不能瞬间建立。因此可以在程序中等待，等它连接成功为止，于是许多程序员就在调用 Connect()之后，Sleep(0)，然后不停地用 WSAGetLastError() 或者 CAsyncSocket::GetLastError() 查看 Socket 返回的错误，直到返回成功为止。这是一种错误的做法，不能到达预期目的。事实上，可以在 Connect() 调用之后等待 CAsyncSocket::OnConnect() 事件被触发，CAsyncSocket::OnConnect() 是要说明 Socket 要么连接成功了，要么连接彻底失败了。

3 需求分析

3.1 系统根本情况描述

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。

如要下载或阅读全文，请访问：

<https://d.book118.com/618076071053007010>