

# 《网络与通信》课程实验报告

## 实验 2: Socket 通信编程

姓名	汪雨卿	院系	计算机学院	学号	19120191
任课教师	张瑞	指导教师	张瑞		
实验地点	计 708	实验时间	周三 5-6		
实验课表现	出勤、表现得分(10)	实验报告 得分(40)	实验总分		
	操作结果得分(50)				
实验目的:					
1. 掌握 Socket 编程过程;					
2. 编写简单的网络应用程序。					
实验内容:					
利用你选择的任何一个编程语言, 分别基于 TCP 和 UDP 编写一个简单的 Client/Server 网络应用程序。具体程序要求参见《实验指导书》。					
要求以附件形式给出:					
<ul style="list-style-type: none"><li>● 系统概述: 运行环境、编译、使用方法、实现环境、程序文件列表等;</li><li>● 主要数据结构;</li><li>● 主要算法描述;</li><li>● 用户使用手册;</li><li>● 程序源代码;</li></ul>					
实验要求: (学生对预习要求的回答) (10 分)				得分:	
<ul style="list-style-type: none"><li>● Socket编程客户端的主要步骤<ol style="list-style-type: none"><li>1. Socket(): 创建Socket</li><li>2. connect(): 连接指定的计算机端口</li><li>3. send(): 向socket中写入信息</li><li>4. close(): 关闭socket</li></ol></li><li>● Socket编程服务器端的主要步骤<ol style="list-style-type: none"><li>1. Socket(): 创建Socket</li><li>2. bind(): 绑定socket和端口号</li><li>3. listen(): 监听端口号</li><li>4. accept(): 接受来自客户端的连接请求</li><li>5. recv(): 从socket中读取字符</li><li>6. close(): 关闭socket</li></ol></li></ul>					
实验过程中遇到的问题如何解决的? (10 分)				得分:	

问题 1: 在使用 C++作为编程环境实现编程的过程中,教材中使用的库和 C++使用的库不一致,且部分库只针对了 Linux 系统。

由于对于 socket 编程库的不熟悉,最初并不知道该如何替换所需要的 socket 扩展库。因此搜集了很多资料,最后发现在 windows 系统下,可以有一些同类的库实现 socket 的编程。进而自行寻找了教程,再结合实验指导中的材料,通过自主学习的方式,举一反三的实现了 socket 编程的任务。

问题 2: 在实现 socket 进行通信传输的过程中,虽然服务端和客户端可以连接。但是传输的数据是乱码。

通过打印数据,输出数据的类型,发现不能够直接输出字符串类型的数据,需要将 string 类型的数据转换成为 char 类型的数据。实现读取类型,传输类型和输出类型的统一,这样才能够实现输出不会出现乱码以及报错的现象。

问题 3: 利用 C++完成编程 TCP 和 UDP 的编程之后,由于没有接触过 C++可视化的编程。对于如何实现 GUI 编程遇到困难。

最开始还是希望能够在 C++的基础上实现 GUI 的 Windows 风格编程。通过查阅相关资料,发现这个过程相对也并不是很容易。同时,想到上个学期选修的 python 课程中有学习过基于 python 的 GUI 风格编程。因此决定重新利用 python 库实现基于 python 的 socket 编程,然后再通过 python 的 wx 库实现可视化界面。

本次实验的体会(结论)(10分)

得分:

本次 socket 编程实验让我学习到了很多,无论从编程的角度,还是从计算机网络的层面,甚至是用户友好的 GUI 编程。这次相对综合的实验,让我明白了计算机科学是一门需要融会贯通,不断学习,同时不断自我挑战的学科。

这次实现,最开始虽然对于编程语言没有限制。所以实现最初我选择了相对熟悉的 C++,而由于 C++语言对于指针应用的灵活性以及类型转换的限制。让我在完成实验的过程中,不仅仅要当心 socket 库自身使用的方式,更要思考如何运用好不同的指针类型,以及数据结构。也在最开始的实验过程中遇到了不少麻烦。

从计算机网络层面,socket 编程让我实实在在体验到了什么是 TCP 和 UDP,他们的过程和区别。让我了解了套接字的概念,以及网络数据传输的方式。在完成实验之初,我通过资料的学习,了解到了三次握手的过程。也从而窥探到了一些如今互联网传输数据的方式方法。此外,尤其在进行 GUI 编程的过程中,对于每一个步骤需要如何分割,如何展示的思考,让我对 TCP 和 UDP 如何交互数据有了更深入的理解。

最后,从 GUI 编程角度。这次实验也帮助我温习了上个学习所学习的 python 可视化的实现。或许作为一个程序员,编写好一段程序是重要的,而能够赋予他一个可以被用户简单上手使用的界面也是同样重要的。

思考题:(10分)

思考题 1:(4分)

得分:

你所用的编程语言在 Socket 通信中用到的主要类及其主要作用。  
本次实验使用了 C++和 python 的 socket 库实现了 socket 编程及可视化

#### C++:

1. SOCKET 类: socket 可以使一个应用从网络中读取和写入数据
  - a. 创建套接字。
  - b. 设置端口地址, 和访问的 ip 地址。
  - c. 建立客户端和服务端的连接。
  - d. 实现套接字的监听。
  - e. 服务端和用户段的数据传输。
  - f. 套接字的关闭。
2. string 类/ cstring 类 和 iostream 类 :
  - a. 从键盘读入数据
  - b. 实现 string 和 char 的转换

#### Python:

1. socket 类: socket 可以使一个应用从网络中读取和写入数据
  - a. 创建套接字。
  - b. 设置端口地址, 和访问的 ip 地址。
  - c. 建立客户端和服务端的连接。
  - d. 实现套接字的监听。
  - e. 服务端和用户段的数据传输。
  - f. 套接字的关闭。
2. sys 类: 设置一些系统操作, 例如关闭窗口。关闭程序等。
3. wx 类: 实现 GUI 编程。
  - a. 初始化窗口, 设置标题, 窗口大小, 状态。
  - b. 绑定提示语, 设置按键, 并绑定相应的事件。

思考题 2: (6 分)

得分:

说明 TCP 和 UDP 编程的主要差异和特点。

#### 1. TCP 协议的主要特点

- (1) TCP 是面向连接的运输层协议;
- (2) 每一条 TCP 连接只能有两个端点 (即两个套接字), 只能是点对点的;
- (3) TCP 提供可靠的传输服务。传送的数据无差错、不丢失、不重复、按序到达;
- (4) TCP 提供全双工通信。允许通信双方的应用进程在任何时候都可以发送数据, 因为两端都设有发送缓存和接受缓存。
- (5) 面向字节流。虽然应用程序与 TCP 交互是一次一个大小不等的数据块, 但 TCP 把这些数据看成一连串无结构的字节流, 它不保证接收方收到的数据块和发送方发送的数据块具有对应大小关系, 例如, 发送方应用程序交给发送方的 TCP10 个数据块, 但就受访的 TCP 可能只用了 4 个数据块久保收到的字节流交付给上层的应用程序, 但字节流完全一样。

#### 2. UDP 协议特点

- (1) UDP 是无连接的运输层协议;

- (2) UDP 使用尽最大努力交付，不保证可靠交付；
- (3) UDP 是面向报文的，对应用层交下来的报文，不合并，不拆分，保留原报文的边界；
- (4) UDP 没有拥塞控制，因此即使网络出现拥塞也不会降低发送速率；
- (5) UDP 支持一对一，一对多，多对多的交互通信；
- (6) UDP 的首部开销小，只有 8 字节。

### 3. TCP 和 UDP 的比较

- (1) TCP 是可靠传输,UDP 是不可靠传输;
- (2) TCP 面向连接,UDP 无连接;
- (3) TCP 传输数据有序,UDP 不保证数据的有序性;
- (4) TCP 不保存数据边界,UDP 保留数据边界;
- (5) TCP 传输速度相对 UDP 较慢;
- (6) TCP 有流量控制和拥塞控制,UDP 没有;
- (7) TCP 是重量级协议,UDP 是轻量级协议;
- (8) TCP 首部较长 20 字节,UDP 首部较短 8 字节。

指导教师评语：

日期：