**计算机网络课程设计**

**实验报告**

**班级：07111606**

**学号：1120161873**

**姓名：孙颢源**

**北京理工大学**

**计算机学院**

**2019年11月**

**Socket 编程**

1. 实验目的

WinSock 是 Windows 操作系统下的 Socket 编程接口，通过 WinSock 函数库 可以实现基于 TCP/IP 协议的进程之间通信。

1.理解基于 WinSock 的客户/服务器概念

2.掌握使用 WinSock 进行编程的方法

3.了解常见 WinSock 开发模式的使用

1. 实验内容

编写一个TCP或UDP客户端和一个TCP或UDP服务器。实现从远程服务器检索温度、湿度和光数据的客户机。客户机应该连接到服务器，保持连接，并检索一些读数。客户机可以像您希望的那样多次从服务器请求读数。

服务器负责接受客户端连接，并根据客户发送的命令发送正确的数据。

服务器返回的数据如下：

温度= 60 湿度= 0 光= 1

1. 编程/开发环境

表1 计算机基本配置

|  |  |
| --- | --- |
| 操作系统 | Windows 10 家庭中文版 |
| CPU | Intel® Core™ i5-6300HQ |
| 内存 | PyCharm 2018.2.4 x64 |

表2 编程及开发环境

|  |  |
| --- | --- |
| 编程语言 | 开发环境 |
| C/C++ | Visual Studio 2017 |
| Python | PyCharm 2018.2.4 x64 |
| Java | IntelliJ IDEA 2018.2.4 x64 |

1. 系统设计详细

编写服务器端和客户端程序。

服务端指定自己的ip地址以及端口号，同时负责读取本地的.dat文件，接收来自客户端的连接后，服务端首先接收来自客户端的消息，再将对应的数据发回给客户端。

代码部分，以python为例。

设定ip地址、端口等并创建套接字，开始监听是否有客户端接入：

|  |
| --- |
| HOST = '127.0.0.1'  PORT = 8080  BUFSIZ = 1024  ADDR = (HOST,PORT)  tcpSerSock = socket(AF\_INET,SOCK\_STREAM)  tcpSerSock.bind(ADDR)  tcpSerSock.listen(5) |

读取本地文件：

|  |
| --- |
| def getdat(filename):  with open(filename,"r" ) as f:  tmplist=f.readlines()  return tmplist  H=getdat("Humidity.dat")  L=getdat("Light.dat")  T=getdat("Temperature.dat") |

等待客户端连接，连接后开始接收客户端消息并发送相应数据：

|  |
| --- |
| while True:  print('waiting for connection...')  tcpCliSock, addr = tcpSerSock.accept()  print('...connnecting from:', addr)  while True:  data = tcpCliSock.recv(BUFSIZ)  if not data:  break  print("客户端请求："+data.decode('utf-8'))  data1=data.decode('utf-8')  num=int(data1)  message=" TEMPERATURE = "+T[num-1]+" HUMIDITY = "+H[num-1]+" LIGHT = "+L[num-1]  tcpCliSock.send(message.encode())  tcpCliSock.close()  tcpSerSock.close() |

客户端指定要连接的ip地址和端口进行连接，连接后开始接收标准输入的文本，并发送给服务端，同时将收到的服务端消息打印出来。

代码部分，以python为例。

设定要连接的ip地址和端口，并创建套接字：

|  |
| --- |
| HOST = '127.0.0.1'  PORT = 8080  BUFSIZ =1024  ADDR = (HOST,PORT)  tcpCliSock = socket(AF\_INET,SOCK\_STREAM) |

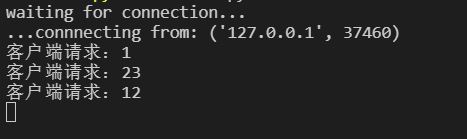
开始连接，并在连接后发送和接收消息：

|  |
| --- |
| tcpCliSock.connect(ADDR)  while True:  data1 = input('>')  if not data1:  break  tcpCliSock.send(data1.encode())  data1 = tcpCliSock.recv(BUFSIZ)  if not data1:  break  print(data1.decode('utf-8'))  tcpCliSock.close() |

1. 程序测试与验证

服务端实验结果：

实验结果中首先接收了来自客户端的连接，然后分别接收了三次请求，后面数字显示的为请求内容。



客户端实验结果：

客户端连接后发送了三次请求，并将收到的结果打印出来。

