|  |  |
| --- | --- |
| **计算机网络**  **实验课**  **实验报告** | |
| **第1次** | |
| 图片包含 标牌  已生成极高可信度的说明 | |
| **姓名** | 隋霖 |
| **班级** | 软件62 |
| **学号** | 2161700078 |
| **电话** | 13572510120 |
| **Email** | 401569798@qq.com |
| **日期** | 2019-6-2 |

目录

1. 实验内容
2. 实验环境
3. 实验原理
4. 实验模块说明
5. 源代码展示
6. 实验结果
7. 实验总结
8. 实验内容

内容:

1) 实现一个简单的客户机/服务器程序,基于 TCP 和 UDP 协议分别实现。

2) 应用场景为一个验证用户登录的程序。

目的:

1) 掌握 Sockets 的相关基础知识，学习 Sockets 编程的基本函数和数据类型

2) 掌握 UDP、TCP Client/Server 模式的通信原理。

3) 掌握 socket 编程命令

实现内容:

基于Python实现一个屏幕实时分享的应用其包含了如下的功能:

1. 登陆
2. 服务端进行UDP广播进行屏幕数据的实时发送
3. 客户端进行UDP数据的接收并显示画面
4. 实验环境

|  |  |
| --- | --- |
| **环境** | **版本** |
| **操作系统环境** | MacOS High Sierra 10.13.6 |
| **Python** | Python 3.6.2 |
| **PyCharm** | PyCharm 2017.1.5 |

1. 实验原理

首先简单介绍互联网的 Client/Server 模式的工作原理，以 TCP 服务器为例说明，

UDP 服务器略有不同。客户端也是如此。

1)服务器

服务器先创建一个套接字(Socket)，并将该套接字和特定端口绑定，然后服务器 开始在此套接字上监听，直到收到一个客户端的连接请求，然后服务器与客户端建立 连接，连接成功后和该客户端进行通信(相互接收和发送数据)，进行用户信息验证， 并返回验证信息。最后，服务器和客户端断开连接，继续在端口上监听。

2)客户端

客户端创建一个套接字，里面包含了服务器的地址和端口号，客户端的端口号由 系统自动分配，不需要指明。和服务器建立连接，如果连接成功则 socket 创建成功。 然后客户端发送用户名和密码，等待验证。通信结束后主动断开连接，释放资源。

SenderView:

1. 展示界面
2. 当配置好传输token而且点击start之后开启TCPServer
3. 等待客户端TCPClient的连接
4. 开启sender
5. sender持续发送start信号
6. sender并且等待有一个基于UDP的Receiver去响应start以进行信息的持续发送
7. TCPServer接受到来自TCPClient的token消息之后和设置好的token进行比对，如果相同返回b"success" 如果不同 返回 b"false"
8. 接受到基于UDP 的Reciever的start响应信号之后开始进行不断的截图操作
9. 先发送总长度
10. 将截图信息分成1464大小的数据段
11. 将1464大小的数据段加上4大小的代表包序号的数字加上4大小的代表包内有效位数的数字不断的向广播地址进行发送 (<broadcast>:port)
12. 发送b"end"信息代表一帧图片数据的发送完毕
13. 当按下stop按键的时候TCPServer, Sender发送b"shutdown"信号之后, 关闭Sender.
14. 如果再次按下start按键之后回到第一步，继续开始执行

ReceiveViewr行为:

1. 输入目标地址:端口
2. 输入连接口令
3. 点击连接按键
4. 开清TCPClient，并随后向目标地址发送"token:${token}"
5. 接受TCPServer返回的答复
6. 如果是b"false"那么修改view中的tkinter的窗体内容
7. 如果是b"success"那么修改view汇总的tkinter的窗体内容，并开启Receiver的UDP接受线程
8. UDP等待接受start信号
9. UDP发送start信号给目标
10. UDP开始接受屏幕截图数据
11. 接受到长度信息
12. 建立数组并初始化为全b”\x00”的数据，作为认为全丢失的时候数据的样子,初始化其他信息
13. 一个数据包一个的接受
14. 将数据包拆解为数据，有效长度，编号
15. 根据编号和有效长度插入数据
16. 当接受到b"end"信息的时候认为一帧数据发送完毕了, 进行图片的bytes2Img转化以及图片格式的转化，然后展示到tkinter窗体中
17. 等待接受下一个长度信息(其他非特殊信号会被跳过)
18. 接收到b"shutdown"信号之后会进行Receiver的关闭，但是TCPClient并不会关闭，因为TCPClient的连接每次发送完得到响应之后就会关闭，下一次使用都需要重新创建.
19. 实验模块说明

ehe

1. 源代码展示

ehe

1. 实验结果

ehe

1. 实验总结

ehe