

廈門大學



信息学院软件工程系

《计算机网络》实验报告

题 目 实验五 利用 Socket API 实现许可认证软件

班 级 软件工程 2020 级数媒班

姓 名 陈鑫蕾

学 号 22920202202877

实验时间 2022 年 11 月 30 日

2022 年 11 月 30 日

填写说明

- 1、本文件为 Word 模板文件，建议使用 Microsoft Word 2019 打开，在可填写的区域中如实填写；
- 2、填表时勿破坏排版，勿修改字体字号，打印成 PDF 文件提交；
- 3、文件总大小尽量控制在 1MB 以下，最大勿超过 5MB；
- 4、应将材料清单上传在代码托管平台上；
- 5、在实验课结束 14 天内，按原文件发送至课程 FTP 指定位置。

1 实验目的

通过完成实验，掌握应用层文件传输的原理；了解传输过程中传输层协议选用、应用层协议设计和协议开发等概念

2 实验环境

操作系统：Win10

编程语言：C++, Python

3 实验结果

首先进行 client, serve 的编写，当遇到套接字未正确创建或连接失败等情况时，程序会终止。

Client.c

创建套接字，将套接字连接到指定的服务器，重复从套接字读取数据并写入用户的屏幕

```
}  
/* Create a socket. */  
sd = socket(PF_INET, SOCK_STREAM, ptrp->p_proto);  
if (sd < 0) {  
    fprintf(stderr, "socket creation failed\n");  
    exit(1);  
}  
  
/* Connect the socket to the specified server. */  
if (connect(sd, (struct sockaddr*)&sad, sizeof(sad)) < 0) {  
    fprintf(stderr, "connect failed\n");  
    exit(1);  
}
```

Serve.c

创建套接字，将本地地址绑定到套接字。结婚则指定请求队列的大小，最后主服务器循环-接受和处理请求。

```

/* Create a socket */
sd = socket(PF_INET, SOCK_STREAM, ptrp->p_proto);
if (sd < 0) {
    fprintf(stderr, "socket creation failed\n");
    exit(1);
}

/* Bind a local address to the socket */
if (bind(sd, (struct sockaddr*)&sad, sizeof(sad)) < 0) {
    fprintf(stderr, "bind failed\n");
    exit(1);
}

/* Specify size of request queue */
if (listen(sd, QLEN) < 0) {
    fprintf(stderr, "listen failed\n");
    exit(1);
}

```

接着进行 socket_api.py 的编写。

使用 g++ 命令，对之前编写的两段代码进行动态编译，生成 server，client 可执行文件。

```

# #编译服务器和客户端
cserver="g++ .\server.c -o .\EXE\server -lwsock32"
cclient="g++ .\client.c -o .\EXE\client -lwsock32"
os.system(cserver)
os.system(cclient)

```

```

PS E:\大三上\计算机网络\实验\5> python CS_project.py
.\server.c:3: warning: "WIN32" redefined
  3 | #define WIN32
    |
<built-in>: note: this is the location of the previous definition
.\server.c:37:5: warning: second argument of 'int main(int, char*)' should
  37 | int main(int argc, char* argv)
    |           ^~~~~~
.\client.c:3: warning: "WIN32" redefined
  3 | #define WIN32
    |
<built-in>: note: this is the location of the previous definition
.\client.c:39:5: warning: second argument of 'int main(int, char*)' should
  39 | int main(int argc, char* argv)
    |           ^~~~~~

```

发放许可证：依次要求用户输入用户名，口令许可证类型等信息。接着打开生成的服务器，随机生成十位序列号，颁发给用户。

```

#发放许可证
while(True):
    username=input("请输入用户名: ")
    password=input("请输入口令: ")
    type=input("请输入许可证类型: ")
    licensenum=licensenum+1
    flag=0 #记录访问次数
    usednum=[] #记录使用过的序列号

    #打开服务器
    open_server=".\\EXE\\server"
    os.system(open_server)

    num=random.randint(1000000000,9999999999) #随机生成十位数序列号
    for i in range(len(usednum)): #判断该序列号是否使用过
        if(usednum[i]==num):
            num=random.randint(1000000000,9999999999)
            i=0

    usednum.append(num)
    print("您的序列号是: "+str(num)+'\\n') #输出序列号

```

```

请输入用户名: CXL
请输入口令: 12345
请输入许可证类型: TCP
您的序列号是: 8063569038

```

显示许可证发放数量和连接人数：要求用户输入序列号，判断此序列号是否使用过。接着打开生成的服务器，向许可证发送验证。并记录许可证发放数量和连接人数

```

else:
    if(flag==0):
        input_num=input("请输入您的序列号: ")
        if(connectnum<maxnum):
            find=0
            for i in range(len(usednum)): #判断该序列号是否使用过
                if(str(usednum[i])==str(input_num)):
                    #print(str(usednum[i]))
                    find=1
                    break
            if(find==0):
                print("输入的序列号有误, 无法连接软件!")
            else:
                create_client=".\\EXE\\server" #用户运行软件, 向许可证服务器发送验证
                os.system(create_client)

```

设置计时器，每一段时间（此处为了方便看到结果，设置为 10s）向服务器报告其状态，显示许可证发放数量和连接人数

```
end_time=time.time()
if(end_time-start_time>10):
    start_time=time.time()
    print("当前许可证发放"+str(licensenum)+"张")
    print("目前有"+str(len(usednum))+ "人在使用该软件")
```

若使用人数超过最大人数，即设置无法接入

```
else:
    #达到最大连接人数时退出程序
    print("目前许可证使用已达到最大人数，无法连入")
```

使用了 while 循环，可以实现非用户主动退出，无法退出程序。既可以在许可证服务器崩溃后，重新启动并恢复。

4 实验代码

本次实验的代码已上传于以下代码仓库：
https://gitee.com/Cutie_Cheng/computer-network

5 实验总结

本次实验使用 C++编写服务器，客户端，使用 python 编写主程序，调用动态编译对服务器，客户端代码进行编译。并在主程序中调用。进一步学习了应用层文件传输的原理，了解了传输过程中传输层协议选用、应用层协议设计和协议开发等概念。