厦門大學

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 2019 级 3 班

姓　　名 王栎

学　　号 22920192204283

实验时间 2021 年 5 月 20 日

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

Windows10

3 实验结果

输入用户名、口令和许可证类型，许可证程序返回一个由 10 个数字组成的序列号

```
//先判断输入的是几项
if (info.size() < 2 || info.size() > 3)
{
    cout << "输入至少一个用户名和密码" << endl;
    cout << "并且按照序列，用户名，密码或者用户名，密码的顺序输入，用空格隔开" << endl;
    getline(cin, input);
    info.clear();
    info = split((char *)input.c_str());
    continue;
}
//如果输入的为三项，则表明有输入序列号，则要对序列号进行检验
else if (info.size() == 3)
{
    SerialNumber = info[0];
    //判断输入序列号是否长为10
    if (SerialNumber.size() != 10)
    {
        cout << "序列号的长度应该为10，请重新输入所有信息：" << endl;
        getline(cin, input);
        info.clear();
        info = split((char *)input.c_str());
        continue;
    }
```

组织用户运行软件时，向许可证服务器发送验证

```
//将客户端输入的序列号发送给服务端以求验证
string str;
if (info.size() == 3)
    str = info[0] + " " + info[1] + " " + info[2];
else
    str = info[0] + " " + info[1];

send(user, str.c_str(), sizeof(str), 0); //发送消息

writing = 0; //发送完之后，写状态结束
```

服务端建立连接，并根据客户端发送的消息进行验证，返回发送客户端

```
//如果第一个有套接字可读的消息，就建立建立连接
if (i == 0 && FD_ISSET(socnum[i], &fds))
{
    struct sockaddr_in client_address;
    socklen_t client_addrLength = sizeof(struct sockaddr_in);
    //返回一个用户的套接字
    int clientfd = accept(listener, (struct sockaddr *)&client_address, &client_addrLengt
    //添加用户，服务器上显示消息，并通知用户连接成功
    socnum.push_back(clientfd);
    cout << "客户端 " << clientfd << "成功连接本服务器" << endl;
    char ID[1024];
    sprintf(ID, "你好，你的ID为：%d", clientfd);

    //服务器产生ID并发送给客户端让客户端知道自己的ID
    send(clientfd, ID, sizeof(ID) - 1, 0); //减去最后一个'/0'
}
if (i != 0 && FD_ISSET(socnum[i], &fds))
{
    char buf[1024];
    memset(buf, '\0', sizeof(buf));
    int size = recv(socnum[i], buf, sizeof(buf) - 1, 0);
    //检测是否断线
    if (size == 0 || size == -1)
    {
        cout << "客户端 " << socnum[i] << "掉线" << endl;
```

```
    vector<string> info = split(buf);
    string str;
    if (info.size() == 2)
    {
        info[0] = "用户名：" + info[0];
        info[1] = "密码：" + info[1];
        str = info[0] + " " + info[1];
    }
    else
    {
        info[0] = "序列号：" + info[0];
        info[1] = "用户名：" + info[1];
        info[2] = "密码：" + info[2];
        str = info[0] + " " + info[1] + " " + info[2];
    }

    /*
    提取完三个（二个）信息后，进行校验，校验的方法应该加在这里
    */

    //将消息发送给发来消息的客户端
    char client[1024];
    sprintf(client, "客户端 %d，", socnum[i]);
    strcat(client, str.c_str());

    send(socnum[i], client, sizeof(client) - 1, 0);
}
```

软件 A 非正常退出（崩溃被其它程序中止）时，许可证服务器应剔除它

```
if (i != 0 && FD_ISSET(socnum[i], &fds))
{
    char buf[1024];
    memset(buf, '\0', sizeof(buf));
    int size = recv(socnum[i], buf, sizeof(buf) - 1, 0);
    //检测是否断线
    if (size == 0 || size == -1)
    {
        cout << "客户端 " << socnum[i] << "掉线" << endl;

        closesocket(socnum[i]);  //关闭这个套接字
        FD_CLR(socnum[i], &fds); //在列表列表中删除
        socnum.erase(socnum.begin() + i); //在vector数组中删除
        ind--;
    }
```

连接服务器失败时重连服务器

```
int flag = init();
if (flag == 1)
{
    Sleep(10000);
    cout << endl; //10秒后尝试重新连接服务器
    return;
} //连接服务器失败，需要尝试再次重连
```

接收消息

```
//服务端有信息发送回来时，接收消息
if (FD_ISSET(user, &fdread))
{
    int size = recv(user, recvbuf, sizeof(recvbuf) - 1, 0);
    if (size > 0)
    {
        cout << "服务端回复:" << recvbuf << endl;
        memset(recvbuf, '\0', sizeof(recvbuf));
    }
    else
    {
        cout << "服务端已关闭，等待服务端开启" << endl;
        Sleep(10000); //等待10秒后尝试重连
        cout << endl;
        cout << "开始尝试重新连接服务器" << endl
             << endl;
        stats = true; //表明服务器崩溃了
        return;
    }
}
if (FD_ISSET(user, &fedwrite))
{
    FD_ZERO(&fedwrite); //将fedwrite清零
    writing = 1;        //设置为写状态

    sendata();
}
```

4 实验代码

本次实验的代码已上传于以下代码仓库：https://github.com/aLily11/cnii

5 实验总结

通过这次实验学习用 socket 流进行网络传输数据，建立连接，绑定，监听，接收，传输数据，并且服务器端对客户端传输的序列号进行验证，并在客户端非正常退出时将它在列表中去除