

# 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 应用层协议服务配置

班 级 软件工程 2021 级卓越班

姓 名 刘陈清

学 号 37220222203693

实验时间 2023 年 5 月 23 日

2023 年 5 月 23 日

# 填写说明

- 1、本文件为 Word 模板文件，建议使用 Microsoft Word 2021 打开，在可填写的区域中如实填写；
- 2、填表时勿改变字体字号，保持排版工整，打印为 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，掌握基于 RFC 应用层协议规约文档传输的原理，实现符合接口且能和已有知名软件协同运作的软件。

## 2 实验环境

操作系统：Windows11

编程语言：C++

## 3 实验结果

按照 socket api 的方法首先配置好两个可以通信的进程

服务器的配置

```
// 创建服务器 socket
serverSocket = socket(AF_INET, SOCK_STREAM, 0);
if (serverSocket == INVALID_SOCKET) {
    cerr << "Socket creation failed" << endl;
    WSACleanup();
    return 1;
}

// 设置服务器地址结构
serverAddr.sin_family = AF_INET;
serverAddr.sin_addr.s_addr = INADDR_ANY;
serverAddr.sin_port = htons(PORT);

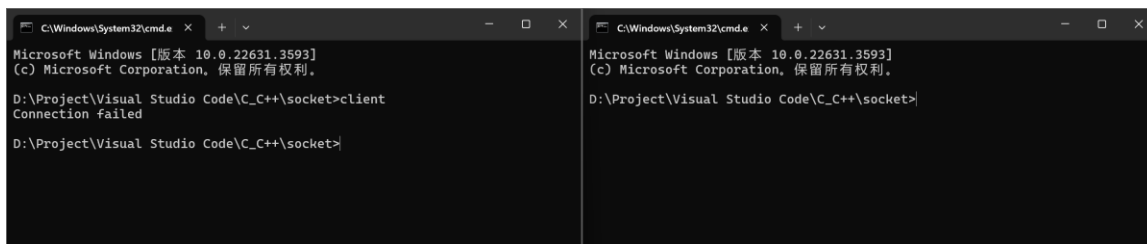
// 绑定服务器地址到 socket
if (bind(serverSocket, (struct sockaddr *)&serverAddr, sizeof(serverAddr)) == SOCKET_ERROR) {
    cerr << "Bind failed" << endl;
    closesocket(serverSocket);
    WSACleanup();
    return 1;
}

// 监听连接
if (listen(serverSocket, SOMAXCONN) == SOCKET_ERROR) {
    cerr << "Listen failed" << endl;
    closesocket(serverSocket);
    WSACleanup();
    return 1;
}

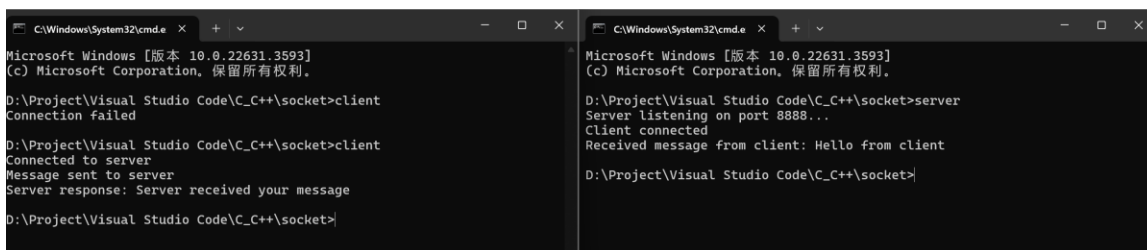
cout << "Server listening on port " << PORT << "..." << endl;
```

## 客户端的配置

服务器未启动时，客户端连接失败。



启动服务器后，客户端向服务器发送“Hello”，且服务器接收成功



```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.22631.3593]
(c) Microsoft Corporation. 保留所有权利。

D:\Project\Visual Studio Code\C_C++\socket>client
Connection failed

D:\Project\Visual Studio Code\C_C++\socket>client
Connected to server
Message sent to server
Server response: Server received your message

D:\Project\Visual Studio Code\C_C++\socket>

C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.22631.3593]
(c) Microsoft Corporation. 保留所有权利。

D:\Project\Visual Studio Code\C_C++\socket>server
Server listening on port 8888...
Client connected
Received message from client: Hello from client

D:\Project\Visual Studio Code\C_C++\socket>
```

在使用 g++ 进行编译时发现如下问题，在编译时使用 -lws2\_32 链接 ws2\_32.lib 即可正常生成可执行程序

```
undefined reference to `__imp_WSASStartup'
undefined reference to `__imp_socket'
undefined reference to `__imp_WSACleanup'
undefined reference to `__imp_htons'
: undefined reference to `__imp_bind'
: undefined reference to `__imp_closesocket'
: undefined reference to `__imp_WSACleanup'
: undefined reference to `__imp_listen'
: undefined reference to `__imp_closesocket'
: undefined reference to `__imp_WSACleanup'
: undefined reference to `__imp_accept'
: undefined reference to `__imp_closesocket'
: undefined reference to `__imp_WSACleanup'
: undefined reference to `__imp_recv'
: undefined reference to `__imp_send'
: undefined reference to `__imp_closesocket'
: undefined reference to `__imp_closesocket'
: undefined reference to `__imp_WSACleanup'
```

接下来该代码基础上编写登陆购买连接等逻辑

编写头文件，使得服务器与客户端两个代码间使用统一的确认信息

```

socket > C basic.h > LICENSE_PATH
1  #include <iostream>
2  #include <fstream>
3  #include <sstream>
4  #include <string>
5  #include <vector>
6  #include <map>
7  #include <winsock2.h>
8  #include <ctime>
9
10 using namespace std;
11
12 #define IS_A_USER "user"
13 #define IS_A_ADMIN "admin"
14 #define ACCOUNT_NOT_EXIST "null"
15 #define ACCOUNT_ALREADY_EXIST "Account already exist"
16 #define REGISTER_SUCCESS "Registration successful"
17 #define PASSWORD_INCORRECT "Password incorrect"
18 #define LOGIN_SUCCESS "Log in successfully"
19 #define NO_LICENSE "You have no license now"
20 #define HAVE_LICENSE "You are using a license"
21 #define USE_SUCCESS "Use the license successfully"
22 #define USE_FAILED "License is not existed or full"
23
24 #define USER_PATH "resource/server_user.txt"
25 #define ADMIN_PATH "resource/server_admin.txt"
26 #define LICENSE_PATH "resource/server_license.txt"
27
28 // 统一客户端与进程间的指令代码
29 #define I_ACCOUNT_EXIST "000"
30 #define I_LOGIN_USER "001"
31 #define I_LOGIN_ADMIN "002"
32 #define I_REGISTER_ACCOUNT "003"
33 #define I_BUY_LICENSE "004"
34 #define I_CHECK_LICENSE "005"
35 #define I_SEND_LICENSE "006"
36
37

```

服务器端:

```

76
77 // 加载用户信息
78 > void loadUserInfoFromFile() { ...
111 // 加载管理员信息
112 > void loadAdminInfoFromFile() { ...
150 // 加载许可证的使用情况
151 > void loadLicenseInfo() { ...
188 // 保存用户信息到文件
189 > void saveUserInfoToFile() { ...
202 // 保存管理员信息到文件
203 > void saveAdminInfoToFile() { ...
220 // 保存许可证使用情况
221 > void saveLicenseInfoToFile() { ...
238
239 // 账号是否存在

```

```

426 void handleClientConnection() {
427     struct sockaddr_in clientAddr;
428     int clientAddrLen = sizeof(clientAddr);
429     clientSocket = accept(serverSocket, (struct sockaddr *)&clientAddr, &clientAddrLen);
430 > if (clientSocket == INVALID_SOCKET) { ...
431
432     cout << "Client connected" << endl;
433
434     // 持续接收客户端数据
435     char buffer[1024];
436     int valread;
437     while (true) {
438         cout<<"Infos(handleClientConnection)"<<endl;
439         cout<<"users count:"<<users.size()<<endl;
440         cout<<"name password license"<<endl;
441 >         for(auto user: users) ...
442         cout<<"admins count:"<<admins.size()<<endl;
443         cout<<"name password licenseCount license(s)"<<endl;
444 >         for(auto admin: admins) ...
445         cout<<"license count:"<<usel.size()<<endl;
446         cout<<"licenseID capacity usedCount username(s)"<<endl;
447 >         for(auto item: usel) ...
448         cout<<endl;
449         memset(buffer, 0, sizeof(buffer)); // 清空缓冲区
450         valread = recv(clientSocket, buffer, sizeof(buffer), 0);
451 >         if (valread == SOCKET_ERROR) { ...
452         } else if (valread == 0) {
453             // 客户端关闭连接
454             cout << "Client disconnected" << endl;
455             closesocket(clientSocket);
456             return; // 返回等待下一个连接
457         } else {
458             cout << "Received message from client: " << buffer << endl;
459             stringstream ss = stringstream(buffer);
460             string infoType;
461             getline(ss,infoType,' ');
462
463             if(infoType==I_ACCOUNT_EXIST) accountExist(ss);
464             else if(infoType==I_LOGIN_USER) loginUser(ss);
465             else if(infoType==I_LOGIN_ADMIN) loginAdmin(ss);
466             else if(infoType==I_REGISTER_ACCOUNT) registerAccount(ss);
467             else if(infoType==I_BUY_LICENSE) buyLicense(ss);
468             else if(infoType==I_CHECK_LICENSE) checkLicense(ss);
469             else if(infoType==I_SEND_LICENSE) sendLicense(ss);
470
471         }
472     }
473 }

```

问题 输出 调试控制台 终端 端口

未在工作区检测到问题。

客户端

```

> // 登陆的多个流程...
> string accountExist(string &name) ...
> bool loginUser(string username) ...
> bool loginAdmin(string adminName) ...
> bool registerAccount(string name) ...
> bool login(string name) ...

// 购买与使用许可证
> bool buyLicense() ...
> bool toUse() { ...
> int main() { ...

```

```

64 int main() {
65     if (!connectToServer()) { ...
66
67     while (true) {
68         reset();
69         // 读取用户输入
70         std::cout << "Enter accountname (or 'exit' to quit): ";
71         string accountname;
72         std::cin >> accountname;
73
74         // 检查用户是否想退出
75         if (accountname == "exit") {
76             break;
77         }
78
79         // 检查消息是否为空
80         if (accountname.empty()) {
81             cout << "Cannot send empty message" << endl;
82             continue;
83         }
84
85         if (login(accountname))
86         {
87             if(isAdmin)
88             {
89                 string confirm;
90                 cout<<"Buy license?(yes/not):";
91                 cin>>confirm;
92                 if(confirm == "yes") {
93                     if(buyLicense()) cout<<"Buy license successfully!"<<endl;
94                     else cout<<"Buy license failed"<<endl;
95                 }
96             }
97             else
98             {
99                 toUse();
100             }
101         }
102     }
103 }
104
105
106

```

当输入用户名不存在时，实现注册

客户端 ↓



```
Connected to server
Enter accountname (or 'exit' to quit): 10005
Account not found. Would you like to register? (yes/no): yes
Enter password: 123
Are you an administrator? (yes/no): no
Enter accountname (or 'exit' to quit): |
```

服务器↓

```
Received message from client: 000 10005
Infos(handleClientConnection)
users count:4
name password license
10001 123 null
10002 123 null
10003 123 null
10004 123 null
admins count:4
name password licenseCount license(s)
111 123 0
222 123 0
333 123 0
444 123 0
license count:0
licenseID capacity usedCount username(s)

Received message from client: 003 10005 123 user
Infos(handleClientConnection)
users count:5
name password license
10001 123 null
10002 123 null
10003 123 null
10004 123 null
10005 123 null
admins count:4
name password licenseCount license(s)
111 123 0
222 123 0
333 123 0
444 123 0
license count:0
licenseID capacity usedCount username(s)
```

购买许可证

客户端，购买成功接收到服务器随机生成的 license↓

```
Enter accountname (or 'exit' to quit): 111
Is a admin
Enter password(or 'exit' to quit):123
Buy license?(yes/not):yes
Input count of license could capacity?(50/10/2):2
get license:2439726918
Buy license successfully!
Enter accountname (or 'exit' to quit): |
```

服务器随机生成许可证

```
string id;
srand(time(NULL));
for (int i = 0; i < 10;i++)
{
    id.push_back(rand()%10+'0');
}
cout<<"The license is:"<<id<<endl;
```

在本地文件中成功更新该记录

名称	修改日期	2439726918 2 0
server_admin.txt	2024-05-23 20:25	
server_license.txt	2024-05-23 20:25	
server_user.txt	2024-05-23 20:23	

用两个账户使用该许可证（便于演示，该许可证容量只有2）

```

Enter accountname (or 'exit' to quit): 10001
This account is a user
Enter password(or 'exit' to quit):123
Log in successfully
You have no license now
Enter a license to use the software:2439726918
Use the license successfully
Enter accountname (or 'exit' to quit): 10002
This account is a user
Enter password(or 'exit' to quit):123
Log in successfully
You have no license now
Enter a license to use the software:2439726918
Use the license successfully
Enter accountname (or 'exit' to quit): |

```

服务器输出的信息可以看到该许可证的两个用户

```

licenseID capacity usedCount username(s)
2439726918 2 2 10001 10002

```

此时新用户使用该许可证时无法通过

```

Use the license successfully
Enter accountname (or 'exit' to quit): 10003
This account is a user
Enter password(or 'exit' to quit):123
Log in successfully
You have no license now
Enter a license to use the software:2439726918
License is not existed or full

```

基本实现实验要求功能

## 4 实验代码

本次实验的代码已上传于以下代码仓库: [Codes/exp5 at main · MidiAug/Codes \(github.com\)](https://github.com/MidiAug/Codes/tree/main/exp5)

## 5 实验总结

通过完成实验，我掌握了应用层文件传输的原理，理解了传输过程中应用层协议的重要性，并学会了设计和实现简单的许可证验证机制。