

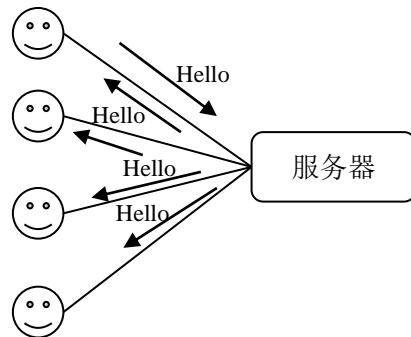


院 系 数据科学与计算机学院 班 级 19 学号 19335174 姓名 施天子

【实验题目】多人聊天编程实验

【实验目的】掌握套接字的多线程编程方法。

【实验介绍】利用客户/服务器(Client/Server 或 CS)模式实现一个多人聊天(群聊)程序。其功能是每个客户发送给服务器的消息都会传送给所有的客户端。



【实验环境】

(1) 采用 Windows VC++ 控制台编程，建立新项目的方法：选择菜单“文件/新建项目/Win32 控制台应用程序/空项目”，然后增加 CPP 空文件，拷贝源程序。集成环境使用 VS2012 或更高版本。

(2) 采用 linux 编程

【参考资料】

- <https://docs.microsoft.com/en-us/windows/desktop/WinSock/getting-started-with-winsock> (套接字)
- <https://www.cnblogs.com/hgwang/p/6074038.html> (套接字)
- <https://www.jb51.net/article/37410.htm> (字符串)
- <https://docs.microsoft.com/en-us/cpp/c-runtime-library/stream-i-o?view=vs-2017> (字符串)
- <https://docs.microsoft.com/en-us/cpp/c-runtime-library/reference/crt-alphabetical-function-reference?view=vs-2017#s> (字符串)
- <http://www.runoob.com/cprogramming/> (字符串)

【注意事项】

依步骤完成以下实验内容，**尽量多完成一些步骤**。把出现的问题、解决方法和一些思考和体会写在**实验体会**部分；对典型的运行情况的客户和服务器控制台进行截屏并放在**实验结果**部分；截屏用按键(Ctrl+Alt+PrintScreen)单独截取控制台窗口。截屏应尽量 windows 绘图程序缩小尺寸(能看清就行)后粘入。

端口号采用 50500

字符串可以采用函数 scanf 或 gets_s 输入。

【参考资料】

- 1、 例程 “_beginthreadex” (创建线程)
- 2、 例程 “TCPServer 和 TCPClient” (传送服务器时间)
- 3、 课件 “套接字并发编程.PDF”
- 4、 Chat 实验的课件

【实验内容】

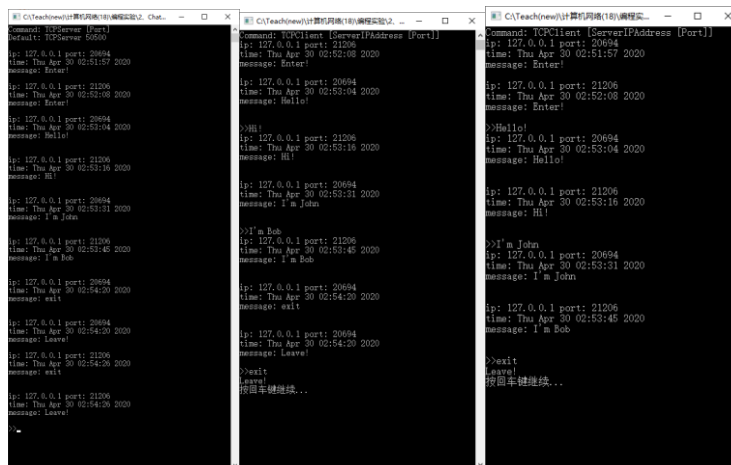
先阅读课件“套接字并发编程.PDF”。重点是读懂课件中“chat 并发编程(服务器)”和“chat 并



发编程(客户端)”的流程图。然后,完成下面步骤(截屏要同时显示服务器和至少两个客户端):

注意保存每一步完成后的源码。

第4步完成后的参考截屏:



服务器

客户端 1

客户端 2

(1)编写多人聊天程序,要求客户端和服务器都采用多线程方式进行编程。每个客户端都采用 TCP 协议连接服务器并保持连接。服务器同时与所有客户端建立和保持连接。每个客户端输入的消息都会通过服务器转发给所有客户。

客户端程序:

```
#include <stdlib.h>
#include <stdio.h>
#include <winsock2.h>
#include <string.h>
#include <process.h>
#include <windows.h>

#define BUFLLEN 2000 // 缓冲区大小
#define WSVERS MAKEWORD(2, 0) // 指明版本2.0
#pragma comment(lib, "ws2_32.lib") // 使用winsock 2.0 Library

SOCKET sock;
HANDLE thread;
unsigned thread_id;
unsigned int __stdcall client(void* p);

unsigned int __stdcall client(void* p) {
    char buf[BUFLLEN];
    while (true) {
        int cc = recv(sock, buf, BUFLLEN, 0);
        if (cc == SOCKET_ERROR || cc == 0) {
            printf("Error:%d\n与服务器断开连接! \n", GetLastError());
            CloseHandle(thread);
        }
    }
}
```



```
(void) closesocket(sock);
break;
}
else {
    buf[cc] = '\0';
    printf("%s\n\n", buf);
}
}
return 0;
}

void main(int argc, char *argv[])
{
    char *host = "172.19.46.225"; // server IP to connect
    char *service = "50500"; // server port to connect
    struct sockaddr_in sin; // an Internet endpoint address
    char buf[BUFLen+1]; // buffer for one line of text

    WSADATA wsadata;
    WSAStartup(WSVers, &wsadata); //加载winsock library。WSVERS
    为请求的版本，wsadata返回系统实际支持的最高版本

    sock = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP); //创建套接字，参数：因特网协
    议簇(family)，流套接字，TCP协议

    //返回：要监听套接字的描
    述符或INVALID_SOCKET
    printf("Command: TCPClient [ServerIPAddress [Port]]\n");
    memset(&sin, 0, sizeof(sin)); // 从&sin开始的长度为
    sizeof(sin)的内存清0
    sin.sin_family = AF_INET; // 因特网地址簇
    sin.sin_addr.s_addr = inet_addr(host); // 服务器IP地址(32位)
    sin.sin_port = htons((u_short)atoi(service)); // 服务器端口号
    connect(sock, (struct sockaddr *)&sin, sizeof(sin)); // 连接到服务器
    thread = (HANDLE)_beginthreadex(NULL, 0, client, NULL, 0, &thread_id);
    while (true) {
        scanf("%s", buf);
        if (!strcmp(buf, "exit")) break;
        (void)send(sock, buf, strlen(buf), 0);
    }
    CloseHandle(thread);
    closesocket(sock);
    WSACleanup();

    puts("按回车键继续...");
    getchar();
}
```



服务器端程序:

```
#include <stdlib.h>
#include <stdio.h>
#include <winsock2.h>
#include <time.h>
#include "conio.h"
#include <Windows.h>
#include <process.h>

#define BUFLen 2000
#define WSVERS MAKEWORD(2, 0)

#pragma comment(lib, "ws2_32.lib") //使用winsock 2.2 library
struct sockaddr_in fsin; //the from address of a client
SOCKET msock, ssock; //master & slave sockets
WSADATA wsadata;
char* service = "50500";

int alen; //from-address length
char* pts; //pointer to time string
time_t now; //current time
SOCKET sockets[BUFLen];
HANDLE thread[BUFLen];
unsigned int thread_id, Thread_id[BUFLen], cnt;

unsigned int __stdcall server(void* p) {
    char buf[BUFLen], buf2[BUFLen];
    time_t time_now;
    char* pts;
    SOCKET sock = ssock;
    int now = cnt;
    sockets[now] = sock;
    Thread_id[now] = thread_id;

    (void)time(&time_now);
    pts = ctime(&time_now);

    sprintf(buf2, "ip: %s port: %d\ntime: %smesssage: Enter!", inet_ntoa(fsin.sin_addr),
    fsin.sin_port, pts);
    printf("%s\n", buf2);
    for (int i = 0; i <= cnt; i++) {
        if (sockets[i] != NULL) {
            (void)send(sockets[i], buf2, strlen(buf2), 0);
        }
    }
    printf("\n");
}
```



```
while (true) {
    int cc = recv(sock, buf, BUFLen, 0);
    if (cc == SOCKET_ERROR) {
        sprintf(buf2, "ip: %s port: %d\ntime: %smesssage: Leave!",
inet_ntoa(fsin.sin_addr), fsin.sin_port, pts);
        printf("%s\n", buf2);
        (void) closesocket(sock);
        sock = NULL;
        sockets[now] = NULL;
        CloseHandle(thread[now]);
        for (int i = 0; i <= cnt; i++) {
            if (sockets[i] != NULL) {
                (void) send(sockets[i], buf2, strlen(buf2), 0);
            }
        }
        printf("\n");
        break;
    }
    else if (cc > 0) {
        buf[cc] = '\0';
        sprintf(buf2, "ip: %s port: %d\ntime: %smesssage: %s",
inet_ntoa(fsin.sin_addr), fsin.sin_port, pts, buf);
        printf("%s\n", buf2);
        for (int i = 0; i <= cnt; i++) {
            if (sockets[i] != NULL) {
                (void) send(sockets[i], buf2, strlen(buf2), 0);
            }
        }
        printf("\n");
    }
}

return 0;
}

void
main(int argc, char *argv[])
{
    struct sockaddr_in sin;          //an Internet endpoint address
    WSStartup(WSVERS, &wsadata);    // 加载winsock library。WSVERS
指明请求使用的版本。wsadata返回系统实际支持的最高版本
    msock = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP); // 创建套接字，参数：因特网协议
簇(family)，流套接字，TCP协议
                                                // 返回：要监听套接字的描
述符或INVALID_SOCKET
    cnt = -1;
    memset(&sin, 0, sizeof(sin));    // 从&sin开始的长度为
```



sizeof(sin)的内存清0

```
sin.sin_family = AF_INET; // 因特网地址簇(INET-Internet)
sin.sin_addr.s_addr = INADDR_ANY; // 监听所有(接口的)IP地址。
sin.sin_port = htons((u_short)atoi(service)); // 监听的端口号。atoi一把ascii
转化为int, htons--主机序到网络序(16位)
bind(msock, (struct sockaddr *)&sin, sizeof(sin)); // 绑定监听的IP地址和端口号

listen(msock, 5); // 等待建立连接的队列长度为5
printf("Command: TCPServer [Port]\n");
printf("Default: TCPServer 50500\n\n");
while(!_kbhit()) { // 检测是否有按键
    alen = sizeof(struct sockaddr); // 取到地址结构的长度
    ssock = accept(msock, (struct sockaddr *)&fsin, &alen); // 如果有新的连接请求, 返回连接套接字, 否则, 被阻塞。fsin包含客户端IP地址和端口号
    thread[++cnt] = (HANDLE)_beginthreadex(NULL, 0, server, NULL, 0, &thread_id);
}
(void) closesocket(msock); // 关闭监听套接字
WSACleanup(); // 卸载winsock library
for (int i = 0; i <= cnt; i++)
    if (thread[i] != NULL)
        WaitForSingleObject(thread[i], INFINITE);
printf("按回车键继续...");
getchar();
}
```

运行截屏:

服务器

```
C:\Users\DELL\Desktop\19335174_施天予_chat实验源码\TCPServer\Debug\TCPServer.exe
Command: TCPServer [Port]
Default: TCPServer 50500

ip: 172.19.46.225 port: 15081
time: Thu Mar 18 20:42:53 2021
message: Enter!

ip: 172.19.46.225 port: 15593
time: Thu Mar 18 20:42:55 2021
message: Enter!

ip: 172.19.46.225 port: 15593
time: Thu Mar 18 20:42:53 2021
message: hello

ip: 172.19.46.225 port: 15593
time: Thu Mar 18 20:42:55 2021
message: hi
```

第一个客户端

```
C:\Users\DELL\Desktop\19335174_施天予_chat实验源码\TCPClient\Debug\TCPClient.exe
Command: TCPClient [ServerIPAddress [Port]]
ip: 172.19.46.225 port: 15081
time: Thu Mar 18 20:42:53 2021
message: Enter!

ip: 172.19.46.225 port: 15593
time: Thu Mar 18 20:42:55 2021
message: Enter!

hello
ip: 172.19.46.225 port: 15593
time: Thu Mar 18 20:42:53 2021
message: hello

ip: 172.19.46.225 port: 15593
time: Thu Mar 18 20:42:55 2021
message: hi
```



第二个客户端

```
C:\Users\DELL\Desktop\19335174_施天予_chat实验源码\TCPClient\Debug\TCPClient.exe
Command: TCPClient [ServerIPAddress [Port]]
ip: 172.19.46.225 port: 15593
time: Thu Mar 18 20:42:55 2021
message: Enter!

ip: 172.19.46.225 port: 15593
time: Thu Mar 18 20:42:53 2021
message: hello

hi
ip: 172.19.46.225 port: 15593
time: Thu Mar 18 20:42:55 2021
message: hi
```

- (2) 服务器程序转发某个客户端发来的消息时都在消息前面加上该客户端的 IP 地址和端口号以及服务器的当前时间。要求服务器程序把转发的消息也显示出来。

服务器程序(修改部分):

```
else if (cc > 0) {
    buf[cc] = '\0';
    sprintf(buf2, "ip: %s port: %d\ntime: %smessage: %s", inet_ntoa(fsin.sin_addr), fsin.sin_port, pts, buf);
    printf("%s\n", buf2);
    for (int i = 0; i <= cnt; i++) {
        if (sockets[i] != NULL) {
            (void)send(sockets[i], buf2, strlen(buf2), 0);
        }
    }
    printf("\n");
}
```

运行截屏:

服务器

```
C:\Users\DELL\Desktop\19335174_施天予_chat实验源码\TCPServer\Debug\TCPServer.exe
Command: TCPServer [Port]
Default: TCPServer 50500

ip: 172.19.46.225 port: 15081
time: Thu Mar 18 20:42:53 2021
message: Enter!

ip: 172.19.46.225 port: 15593
time: Thu Mar 18 20:42:55 2021
message: Enter!

ip: 172.19.46.225 port: 15593
time: Thu Mar 18 20:42:53 2021
message: hello

ip: 172.19.46.225 port: 15593
time: Thu Mar 18 20:42:55 2021
message: hi
```

第一个客户端

```
C:\Users\DELL\Desktop\19335174_施天予_chat实验源码\TCPClient\Debug\TCPClient.exe
Command: TCPClient [ServerIPAddress [Port]]
ip: 172.19.46.225 port: 15081
time: Thu Mar 18 20:42:53 2021
message: Enter!

ip: 172.19.46.225 port: 15593
time: Thu Mar 18 20:42:55 2021
message: Enter!

hello
ip: 172.19.46.225 port: 15593
time: Thu Mar 18 20:42:53 2021
message: hello

ip: 172.19.46.225 port: 15593
time: Thu Mar 18 20:42:55 2021
message: hi
```

第二个客户端



C:\Users\DELL\Desktop\19335174_施天予_chat实验源码\TCPClient\Debug\TCPClient.exe

Command: TCPClient [ServerIPAddress [Port]]

ip: 172.19.46.225 port: 15593

time: Thu Mar 18 20:42:55 2021

message: Enter!

ip: 172.19.46.225 port: 15593

time: Thu Mar 18 20:42:53 2021

message: hello

hi

ip: 172.19.46.225 port: 15593

time: Thu Mar 18 20:42:55 2021

message: hi

- (3) 新客户刚连接时服务器端把“enter”消息（包含客户端 IP 地址和端口号）发送给所有客户端。

服务器程序(修改部分):

```
printf(buf2, "ip: %s port: %d\ntime: %smessage: Enter!", inet_ntoa(fsin.sin_addr), fsin.sin_port, pts);  
printf("%s\n", buf2);  
for (int i = 0; i <= cnt; i++) {  
    if (sockets[i] != NULL) {  
        (void)send(sockets[i], buf2, strlen(buf2), 0);  
    }  
}
```

运行截屏:

服务器

C:\Users\DELL\Desktop\19335174_施天予_chat实验源码\TCPClient\Debug\TCPClient.exe

Command: TCPClient [ServerIPAddress [Port]]

ip: 172.19.46.225 port: 15081

time: Thu Mar 18 20:42:53 2021

message: Enter!

ip: 172.19.46.225 port: 15593

time: Thu Mar 18 20:42:55 2021

message: Enter!

hello

ip: 172.19.46.225 port: 15593

time: Thu Mar 18 20:42:53 2021

message: hello

ip: 172.19.46.225 port: 15593

time: Thu Mar 18 20:42:55 2021

message: hi

第一个客户端

C:\Users\DELL\Desktop\19335174_施天予_chat实验源码\TCPClient\Debug\TCPClient.exe

Command: TCPClient [ServerIPAddress [Port]]

ip: 172.19.46.225 port: 15081

time: Thu Mar 18 20:42:53 2021

message: Enter!

ip: 172.19.46.225 port: 15593

time: Thu Mar 18 20:42:55 2021

message: Enter!

hello

ip: 172.19.46.225 port: 15593

time: Thu Mar 18 20:42:53 2021

message: hello

ip: 172.19.46.225 port: 15593

time: Thu Mar 18 20:42:55 2021

message: hi



第二个客户端

```
C:\Users\DELL\Desktop\19335174_施天予_chat实验源码\TCPClient\Debug\TCPClient.exe
Command: TCPClient [ServerIPAddress [Port]]
ip: 172.19.46.225 port: 15593
time: Thu Mar 18 20:42:55 2021
message: Enter!

ip: 172.19.46.225 port: 15593
time: Thu Mar 18 20:42:53 2021
message: hello

hi
ip: 172.19.46.225 port: 15593
time: Thu Mar 18 20:42:55 2021
message: hi
```

- (4) 客户端输入 exit 时退出客户端程序（正常退出），或者客户端直接关闭窗口退出（异常退出），服务器都会把该客户 leave 的消息广播给所有客户。

服务器程序(修改部分):

```
while (true) {
    int cc = recv(sock, buf, BUFLen, 0);
    if (cc == SOCKET_ERROR) {
        sprintf(buf2, "ip: %s port: %d\ntime: %smessage: Leave!", inet_ntoa(fsin.sin_addr), fsin.sin_port, pts);
        printf("%s\n", buf2);
        (void) closesocket(sock);
        sock = NULL;
        sockets[now] = NULL;
        CloseHandle(thread[now]);
        for (int i = 0; i <= cnt; i++) {
            if (sockets[i] != NULL) {
                (void) send(sockets[i], buf2, strlen(buf2), 0);
            }
        }
        printf("\n");
        break;
    }
}
```

运行截屏:

服务器

```
C:\Users\DELL\Desktop\19335174_施天予_chat实验源码\TCPServer\Debug\TCPServer.exe
Command: TCPServer [Port]
Default: TCPServer 50500

ip: 172.19.46.225 port: 63211
time: Thu Mar 18 20:56:33 2021
message: Enter!

ip: 172.19.46.225 port: 64235
time: Thu Mar 18 20:56:36 2021
message: Enter!

ip: 172.19.46.225 port: 64235
time: Thu Mar 18 20:56:36 2021
message: Leave!
```

第一个客户端在输入“exit”后已退出（直接退出了无法截屏）

第二个客户端

```
C:\Users\DELL\Desktop\19335174_施天予_chat实验源码\TCPClient\Debug\TCPClient.exe
Command: TCPClient [ServerIPAddress [Port]]
ip: 172.19.46.225 port: 63211
time: Thu Mar 18 20:56:33 2021
message: Enter!

ip: 172.19.46.225 port: 64235
time: Thu Mar 18 20:56:36 2021
message: Enter!

ip: 172.19.46.225 port: 64235
time: Thu Mar 18 20:56:36 2021
message: Leave!
```



(5) 运行客户端程序测试与老师的服务器程序的连接 (172.18.187.251:50500)。

运行截屏 (客户端):

```
C:\Users\DELL\Desktop\19335174_施天予_chat实验源码\TCPClient\Debug\TCPClient.exe
Command: TCPClient [ServerIPAddress [Port]]
ip: 172.19.46.225 port: 55276
time: Thu Mar 18 21:03:25 2021
message: Enter!

>>

张老师您好
ip: 172.19.46.225 port: 55276
time: Thu Mar 18 21:03:48 2021
message: 张老师您好

>>

exit
Error:10053
与服务器断开连接!
```

(6) 与同学的程序进行相互测试, 一个人可以与多人测试, 截屏选择其中一个。

同学的学号姓名 (可以多人): 覃浩南 19335179 孙奥远 19335177

IP 地址: 我 172.26.7.147 覃浩南 172.26.105.139 孙奥远 172.26.96.7

作为服务器运行截屏:

```
C:\Users\DELL\Desktop\19335174_施天予_chat实验源码\TCPServer\Debug\TCPServer.exe
Command: TCPServer [Port]
Default: TCPServer 50500

ip: 172.26.7.147 port: 35793
time: Fri Mar 19 23:15:06 2021
message: Enter!

ip: 172.26.105.139 port: 20228
time: Fri Mar 19 23:15:07 2021
message: Enter!

ip: 172.26.96.7 port: 6695
time: Fri Mar 19 23:15:11 2021
message: Enter!

ip: 172.26.96.7 port: 6695
time: Fri Mar 19 23:15:06 2021
message: 曼城巴黎皇马利物浦多特切尔西波尔图拜仁

ip: 172.26.96.7 port: 6695
time: Fri Mar 19 23:15:07 2021
message: 多特和切尔西
```

作为客户端运行截屏:

```
C:\Users\DELL\Desktop\19335174_施天予_chat实验源码\TCPClient\Debug\TCPClient.exe
Welcome!
The man's IP is 172.26.7.147 and his port is 46298.
Now is Fri Mar 19 22:52:35 2021
message: Enter!

快赌球
IP: 172.26.7.147 port: 46298
Current Time: Fri Mar 19 22:52:52 2021
message: 快赌球

曼城拜仁波尔图切尔西皇马利物浦巴黎多特
IP: 172.26.7.147 port: 46298
Current Time: Fri Mar 19 22:54:07 2021
message: 曼城拜仁波尔图切尔西皇马利物浦巴黎多特

Welcome!
The man's IP is 127.0.0.1 and his port is 12.
Now is Fri Mar 19 22:55:07 2021
message: Enter!

IP: 127.0.0.1 port: 12
Current Time: Fri Mar 19 22:55:12 2021
message: 我来了
```



【完成情况】

是否完成了这些步骤? (√完成 ×未做或未完成)

(1) [√] (2) [√] (3) [√] (4) [√] (5) [√] (6) [√]

【实验体会】

写出实验过程中遇到的问题，解决方法和自己的思考;并简述实验体会（如果有的话）。

这次实验主要是对套接字的延伸，也用到了多线程编程的知识。实验中各种复杂的函数一开始把我搞晕了，debug 也搞了好久，但好在通过不断地不断地看示例和学习我终于掌握了一些。

在实验时我发现了一个问题，就是在客户端的程序中我一开始把获取时间的变量作为全局变量，在线程函数内直接调用已取得的时间，但这样就不能把“leave”消息转发到其他客户端。在线程函数内再设一个时间变量才能正确转发“leave”。第二个问题是，在和同学测试时没有关防火墙，发现收不到同学的消息，而且如果在输入消息的同时另一个同学已经发送了消息，那么我正在输入的消息会被“吞掉”，无法继续输入。还有如果同学间设置的缓冲区长度大小是不同的，那么也有可能出现乱码等情况。

总而言之，虽然这次实验消耗了我大量的时间和精力，但我也学到了很多，也是非常值得的。希望以后能再接再厉！

【交实验报告】

(1) 每位同学单独完成本实验内容并填写实验报告。

(2) 上交网址: <http://172.18.187.251/netdisk/default.aspx?vm=19net>

实验上交/编程实验/3、Chat 实验

(3) 截止日期（不迟于）：3 月 23 日（周二） 23:00。

上传文件：(1) 学号_姓名_chat 实验报告.doc （最好用 doc 文件）

(2) 学号_姓名_chat 实验要求.rar （打包源程序文件和可执行文件）