

20220414-socket

1.过程描述

1.1 简单的socket编程实现

2.结果输出

1.过程描述

1.1 简单的socket编程实现

```

1  #include <stdio.h>
2  #include <WinSock2.h>
3  #pragma comment (lib, "ws2_32.lib") //在编译时加载ws_32.dll
4
5  int main()
6  {
7      //初始化 DLL
8      WSADATA wsaData;
9      WSAStartup(MAKEWORD(2, 2), &wsaData); //指明WinSock规范的版本;wsadata为指向WSADATA结构体的指针
10
11     //创建套接字
12     SOCKET servSock = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);
13     /*
14     原型为SOCKET socket(int af,int type,int protocol)
15     1.af为地址族, 常用的有AF_INET, AF_INET6
16     PF_INET为IPv4协议族, 与AF_INET等价;
17
18     2.type为数据传输方式, 主要有两种:
19     SOCK_STREAM表示面向连接的数据传输方式, 传输较为准确, 损坏或丢失可以重发, 但效率较慢, HTTP协议使用的为该方式
20     SOCK_DGRAM表示无连接的数据传输方式, 计算机只管传输不作校验, 效率较高
21
22     3.protocol为传输协议, 常用的有IPPROTO_TCP和IPPROTO_UDP
23     */
24
25     //绑定套接字
26     sockaddr_in sockAddr;
27     memset(&sockAddr, 0, sizeof(sockAddr)); //每个字节用0填充
28     sockAddr.sin_family = PF_INET; //使用IPV4地址
29     sockAddr.sin_addr.s_addr = inet_addr("127.0.0.1"); //具体的IP地址, 使用inet_addr函数将字符串转化为合适的类型
30     sockAddr.sin_port = htons(1234); //端口
31     bind(servSock, (SOCKADDR*)&sockAddr, sizeof(SOCKADDR)); //将套接字与IP、端口绑定
32     /*
33     原型为int bind(SOCKET sock,const struct sockaddr *addr,int addrlen)
34     1.sock即为套接字句柄
35     2.这里使用sockaddr_in结构体, 然后再强制转换为sockaddr类型
36     sockaddr是一种通用的结构体, 可以用来保存多种类型的IP地址和端口号, 而sockaddr_in是专门用来保存IPV4地址的结构体, 另外还有sockaddr_in6
37     */
38
39     //进入监听状态
40     listen(servSock, 20);

```

```

41     /*
42     listen使套接字进入被动监听状态
43     原型为int listen(SOCKET sock,int backlog)
44     backlog为请求队列的最大长度
45     1.当套接字正在处理客户端请求时，如果有新的请求进来，套接字是没法处理的，只能放进缓
    冲区，这个缓冲区中的请求并称为请求队列
46     2.backlog设置SOMAXCONN，则由系统决定队列长度，值一般比较大
47     */
48
49     //接收客户端请求
50     SOCKADDR clntAddr;
51     int nSize = sizeof(SOCKADDR);
52     SOCKET clntSock = accept(servSock, (SOCKADDR*)&clntAddr, &nSize);
53     /*
54     accept用于响应客户端请求
55     原型为SOCKET accept(SOCKET sock,struct sockaddr *addr,int *addrlen)
56     accpet返回一个新的套接字来和客户端通信，addr保存了客户端的IP地址和端口号，而sock
    是服务器端的套接字
57     listen只是让套接字进入监听状态，并没有真正接收客户端请求，listen后面的代码会继续
    执行，知道遇到accept。accept会阻塞程序执行（后面代码不能被执行），直到有新的请求到来
58     */
59
60     //向客户端发送数据
61     const char* str = "Hello World";
62     send(clntSock, str, strlen(str) + sizeof(char), NULL);
63     /*
64     原型为int send(SOCKET sock,const char* buf,int len,int flags)
65     buf为要发送的数据的缓冲区地址，len为要发送的数据的字节数，flags为发送数据时的选
    项，一般为0或NULL
66     */
67
68     //关闭套接字
69     closesocket(clntSock);
70     closesocket(servSock);
71
72     //终止DLL的使用
73     WSACleanup();
74     return 0;
75
76 }

```

```
1  ▾ #include <stdio.h>
2  #include <stdlib.h>
3  #include <WinSock2.h>
4  #pragma comment(lib,"ws2_32.lib")
5
6  int main()
7  ▾ {
8      //初始化DLL
9      WSADATA wsaData;
10     WSStartup(MAKEWORD(2, 2), &wsaData);
11
12     //创建套接字
13     SOCKET sock = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);
14
15     //向服务器发起请求
16     sockaddr_in sockAddr;
17     memset(&sockAddr, 0, sizeof(sockAddr));
18     sockAddr.sin_family = PF_INET;
19     sockAddr.sin_addr.s_addr = inet_addr("127.0.0.1");
20     sockAddr.sin_port = htons(1234);
21     connect(sock, (SOCKADDR*)&sockAddr, sizeof(SOCKADDR)); //与bind用法相似
22
23     //接收服务器传回的数据
24     char szBuffer[MAXBYTE] = { 0 };
25     recv(sock, szBuffer, MAXBYTE, NULL);
26     /*
27     原型为int recv(SOCKET sock,char* buf,int len,int flags)
28
29     */
30
31     //输出接收到的数据
32     printf("Message from server: %s\n", szBuffer);
33
34     //关闭套接字
35     closesocket(sock);
36
37     //终止使用DLL
38     WSACleanup();
39
40     system("pause");
41     return 0;
42
43 }
```

2.结果输出

今天上午看了《从微积分到5G》一书，前面主要都是一些数学知识，感觉看起来还OK，后面逐步开始展开到信号系统的知识，其中卷积的部分公式是大概记住了，但总感觉没真正搞明白其中的原理，后面需要再加强一下。晚上跟着教程大概码了一个比较简单的socket套接字程序，一开始感觉很多代码有点不明所以，通过一行一行注释逐渐明白了整个过程大致是如何实现的。几个星期前其实已经照着教材码过一个server跟client的程序，比起今天来说要相对复杂一些，但当时问题更多。经过这几周对于C++基础知识的恶补，再回看之前的代码确实感觉在认知上有了一定的长进。

明天预计将继续套接字的编程实践，原先的盘算是在编程的过程中顺便复习（或学习）一下计算机网络的相关知识，但目前感觉下来，socket编程还是在一个比较高的层级上实现的，对于书上所说的很多原理的知识并没有太多涉及，有些可能很复杂的通信过程有时候可能一个函数调用就解决了，基本不会需要用到特别底层的知识。后续如果有时间的话想进一步探究一下socket源码，可能这样更接近于实现最初的目标。

上海的疫情形势依然严峻，今天看到的很多讯息都让我对社会陷入了深深的怀疑。对很多正在外面经受磨难的人我感到十分难过，而自己只是一个微不足道的个体无能为力。对于一些过激的言论我曾想过反驳，但未经他人苦莫劝他人善，作为一个在温室里的学生确实没资格对别人的抱怨或呼喊指指点点。我也不觉得自己处在他们的境况下能表现得更理智。目前感觉疫情已经发展到了一个重要关口，接下来无论怎么选都需要付出极大得代价。原先大家可能都想着以拖待变，但似乎留给我们拖延的空间已经越来越小了。而我如今能做的，也只能是无力地祈祷一声人人健康。