

# 计算机网络原理实验报告

学院 计算机学院 专业 计算机科学与技术 班级 10012006

学号 2020303245 姓名 夏卓 实验时间: 2022/11/12

## 一、实验名称:

基于 TCP 协议通信编程

## 二、实验目的:

掌握 TCP 协议通信流程,以及 TCP 协议可靠性通信应用系统设计与实现原理;掌握 TCP 协议通信时序关系,并分析采用这种时序关系的原因;掌握利用 TCP 协议实现两台计算机之间的单向通信、双向通信和多媒体文件传输。

## 三、实验环境:

Win10, Intelx86

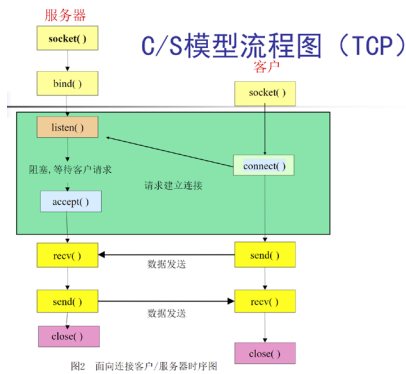
## 四、实验内容及步骤:

### 实验内容:

1. 单向通信: 客户端将从键盘输入任意字符串发送给服务器, 服务器接收到后将该字符串在屏幕上打印, 并将客户端 IP 地址和 PORT 打印;
2. 双向通信: (1) 客户端将从键盘输入任意字符串发送给服务器, 服务器接收到后将该字符串在屏幕上打印, 并将客户端 IP 地址和 PORT 打印; (2) 服务器将客户端发送来的字符串发送给客户端, 客户端打印字符串和服务 IP 地址和 PORT;
3. 传输多媒体文件: 客户端将一个视频文件发送给服务器, 在服务器上能播放; 在发送过程中将“网络接口-停止”(3 到 6 秒), 检测可否正常传输文件。

### 实验步骤:

(1) 理解 TCP 协议的通信流程; 掌握 Socket 编程接口的使用方法



SOCKET socket( int af, int type, int protocol );	TCP	S/C
int bind( SOCKET s, const struct sockaddr FAR * seradd, int seraddlen )	TCP	S
int listen( SOCKET s, int quelen )	TCP	S
SOCKET accept( SOCKET s, struct sockaddr FAR * cliaddr, int FAR * cliaddrlen )	TCP	S
int connect( SOCKET s, const struct sockaddr FAR * seradd, int seraddlen )	TCP	C
int send( SOCKET s, const char FAR * buf, int len, int flags )	TCP	S/C
int recv( SOCKET s, char FAR * buf, int len, int flags )	TCP	S/C
int closesocket( SOCKET s )	TCP	S/C

- (2) 编写程序，利用 TCP 协议实现客户端到服务器单向和双向数据的传输和接收。

与 UDP 一样，程序参考书上 P180-196，需要注意根据实际情况，更改客户端的目的 IP 地址及端口号与服务器的实际 IP 地址与端口号一致，可以在命令行使用 ipconfig 命令进行查看。另外最好打开防火墙，以免通信信息被防火墙屏蔽，使用无线热点进行连接，效果更好。

如下图所示，需要更改的 IP 地址即是局域网 IPv4 的地址：

```
无线局域网适配器 WLAN:

    连接特定的 DNS 后缀 . . . . . : 
    IPv6 地址 . . . . . : fd31:1927:f25a:5:959e:64ee:171d:c633
    临时 IPv6 地址. . . . . : fd31:1927:f25a:5:94:9734:e92:14b5
    本地链接 IPv6 地址. . . . . : fe80::105a:cd46:3b91:4c50%15
    IPv4 地址 . . . . . : 10.30.217.218
    子网掩码 . . . . . : 255.255.0.0
    默认网关. . . . . : 10.30.0.1
```

- (3) 编写程序，利用 TCP 协议实现客户端到服务器单向文件 (\*.avi) 的传输；在文件传输过程中，将网络分别中断 1s、5s、10s 等时间，检测是否可以正常传输文件。

与 UDP 类似，由于待传输文件较小，通信时间很短，为了能够实现中途断网的目的，我们在文件传输过程中使用了 sleep 函数让客户端休眠了 2s，以便让服务器有足够的时间断网后重连：

```
if (flag)
{
    printf("start to sleep\n");
    Sleep(2000);
    printf("sleep end\n");
    flag = false;
}
```

## 五、实验结果：

### 1. 单向通信

客户端：

```
(base) PS D:\大学\大三上\计网\实验1\客户端> .\TCP_single_client
start send data to server:
这里是客户端
TCP连接是安全的
```

服务器：

```
D:\project1\Project1\Debug\Project1.exe
server start to receive data:
```

### 2. 双向通信

客户端：

```
(base) PS D:\大学\大三上\计网\实验1\客户端> .\TCP_double_client
start send data to server:
Hello!
client receive data from server:Hello!
TCP yyds!
client receive data from server:TCP
client receive data from server:yyds!
```

服务器：

```
D:\project1\Project1\Debug\Project1.exe
server start to receive data:
server recieve data from client:Hello!
server recieve data from client:TCP
```

### 3. 传输多媒体文件

客户端：

```
(base) PS D:\大学\大三上\计网\实验1\客户端> .\TCP_fileTrans_client
client start to send file!
file send end successfully!
```

服务器：

```
Microsoft Visual Studio 调试控制台
server start to receive data
server start to recieve file
server recieve data end successfully
D:\project1\Project1\Debug\Project1.exe (进程 15972) 已退出，代码为 0。
要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
```

中途断网：

```
(base) PS D:\大学\大三上\计网\实验1\客户端> .\TCP_fileTrans_client
client start to send file!
start to sleep
sleep end
file send end successfully!
```

实验结果为当中途断网时间较短时，如 1s、5s，客户端会在断网期间暂停发送数据一段时间，当网络恢复后又继续发送，服务器端能够接收到完整数据；但当中途断网时间过长，如 10s 时，客户端在停止发送数据一段时间后又会继续发送数据，导致服务器无法接收到完整数据。

## 六、 实验总结

### TCP C/S 模型通信问题

#### 1. TCP 建立连接的相关函数有哪些？

TCP 请求建立连接过程中，客户端需要使用 `connect()` 函数主动请求与服务器建立连接，而服务器需要使用 `listen()` 函数阻塞侦听连接请求，并将请求放入到等待队列中排队等待处理，然后使用 `accept()` 函数处理客户进程的连接请求。

#### 2. TCP 客户端什么时间知道通信五元组？

TCP 客户端需要在一开始就知道通信五元组，主动向服务器发送连接请求，连接建立后，之后双方才能进行通信。

#### 3. TCP 服务端什么时间知道通信五元组？

TCP 服务端只有在调用了 `accept()` 函数开始处理客户进程的连接请求后，才知道通信五元组。

#### 4. TCP 连接一旦建立，第一次通信哪一方发送数据？为什么没有限制？

TCP 连接建立好以后，客户端与服务器谁先发送 TCP 数据段都可以。这是因为 TCP 时基于连接的通信模式，客户进程标识信息在其调用 `accept()` 后，服务器就已经能确定通信五元组的信息了，此后双方向对方发送数据时不必再指定对方地址，因此没有限制。

5. TCP 建立连接时，为什么是客户端必须向服务器发送连接请求，而不是相反方向？

TCP 服务器进程无法在一开始就得知客户端的 IP 地址及端口号，无法主动建立连接，而是一般在某个周知的端口上等待客户进程的连接请求，只有当有连接请求来到时，服务器进程才被唤醒并处理客户进程的连接请求。因此必须是客户端首先通过调用 `connect()` 函数主动请求与服务器建立连接之后，双方才能开始通信。

### TCP 与 UDP 编程区别

基于连接与无连接：

TCP 存在建立连接和释放连接三次握手，而 UDP 直接发送数据，无连接过程；

TCP 保证数据传输可靠性，UDP 不保证。

TCP 采用字节流模式，而 UDP 采用数据报模式。

对系统资源的要求：

TCP 较多，UDP 少；

TCP 程序结构较复杂，UDP 程序结构较简单。

系统函数调用不同：

`socket()` 的参数不同；

UDP Server 不需要调用 `listen` 和 `accept`；

TCP 收发数据不必指定对方地址，使用 `send/recv` 函数；

UDP 每次收发数据均需要指定对方地址，使用 `sendto/recvfrom` 函数；

TCP 客户进程标识信息在 `accept` 时服务器才能确定；

UDP 客户进程表示信息在服务器接收到数据的同时就能确定。

在传输多媒体文件的过程中，若中途断网，则基于 UDP 的通信过程会丢失数据，而当断网时间不长时，基于 TCP 的通信过程不会丢失数据。

**教师评语：**

**成绩：** \_\_\_\_\_ **教师签名：** \_\_\_\_\_ **批阅日期：** \_\_\_\_\_