

# 深圳大学实验报告

课程名称	计算机网络		
实验名称	实验 1：流式视频传输		
学 院	计算机与软件学院		
专 业	软件工程（腾班）		
指导教师	张磊		
报 告 人	黄亮铭	学号	2022155028
实验时间	2024 年 3 月 6、13 日		
提交时间	2024 年 3 月 14 日		

教务处制

## 一、实验目的与要求

1. 实验目的
  - a) 理解 TCP 套接字的定义；
  - b) 掌握基本的 TCP 套接字编程方法；
  - c) 了解简单网络应用的编程思路；
  - d) 了解网络编程相关的一些库。
2. 实验要求
  - a) 了解网络编程的相关库；
  - b) 掌握编写简单网络应用的技能；
  - c) 依照步骤完成实验内容；
  - d) 对实验结果截图；
  - e) 撰写实验报告。
3. 任务一目标：建立 server 和 client 之间的 socket 连接；实现 server 端向 client 端发送视频文件。
4. 任务一要求：按照给定的代码文件及要求，完成代码补充，并编译运行。
5. 任务二目标：实现连续文件传输；调用提供的代码库，实现视频在线播放。
6. 任务二要求：利用提供的播放器代码库，是实现视频段文件的连续播放。

## 二、实验过程

### 任务一：单文件视频传输

#### 1. 下载源代码框架

下载源代码框架并解压，在 server.c 所在的目录下建立/video 目录，下载并解压测试视频在此；在 client.c 目录下创建一个空的/download 目录

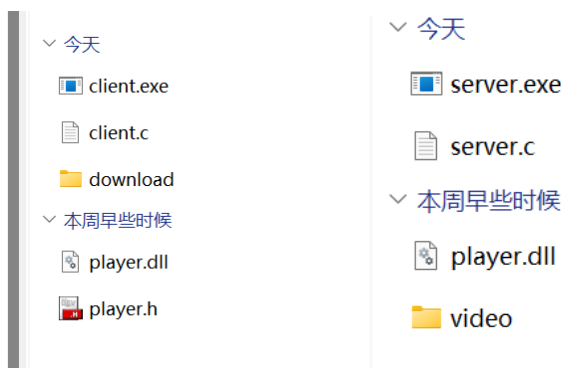


图 1

2. 通过套接字在 Client 和 Server 之间建立 TCP Socket 连接  
已阅读代码并理解。
3. 在服务端使用 **send()** 函数向客户端发送视频文件（编程实现）  
这里只截取需要我实现的内容：将发送缓冲区的内容发送至客户端
  - a) 首先定义 buf\_len，用于获得读取的字节数，同时判断是否读取成功，然后处理读取失败的情况；

- b) 使用 `send` 函数发送视频文件到客户端，定义临时变量 `tmp_send_count` 计算每次发送的字节数，也用于判断缓冲区内容是否发送成功，如果失败则由相应的代码处理该情况；
- c) 最后将 `tmp_send_count` 累加到 `send_count` 上，用于判断视频文件是否发送完成，额外使用 `buf_len` 和 `BUFFER_SIZE` 来判断文件是否全部读取。
- d) 代码截图如下

```

127 | //读取视频文件
128 | int buf_len;
129 | while ((buf_len = fread(buffer, sizeof(char), BUFFER_SIZE, fp)) > 0) {
130 |     //发送视频文件
131 |     int tmp_send_count = send(new_socket, buffer, buf_len, 0);
132 |     if (tmp_send_count < 0) {
133 |         perror("Error in sending video\n");
134 |         exit(EXIT_FAILURE);
135 |     }
136 |     //累计总字节数，用于判断是否发送完成
137 |     send_count += tmp_send_count;
138 | }

```

图 2

4. 在客户端使用 `recv()` 函数接收服务端发送的视频文件（编程实现）  
这里只截取我需要实现的内容：接收服务端发送的数据并写入接收缓冲区
  - a) 使用临时变量 `tmp_recv_count` 获得 `recv` 函数的接收到来自客户端的字节数，同时使用该变量判断数据是否接收成功，若不成功则进入相应的代码处理；
  - b) 因为以 `char` 类型读取数据，影刺使用 `strcat` 函数拼接视频文件；
  - c) 最后使用 `recv_count` 累加 `tmp_recv_count`，判断是否接收完成；
  - d) 代码截图如下

```

100 | int tmp_recv_count = 0;
101 | if ((tmp_recv_count = recv(sock, buffer, BUFFER_SIZE, 0)) < 0) {
102 |     perror("Error in receiving\n");
103 |     return -1;
104 | }
105 | //拼接视频文件
106 | memcpy(video_segement + recv_count, buffer, tmp_recv_count);
107 | //累计总字节数，用于判断是否接收完成
108 | recv_count += tmp_recv_count;
109 | /*****

```

图 3

## 任务二：实现流式视频传输

5. 下载实验 1 所需的文件库，解压到 `client.c` 和 `server.c` 所在的文件夹

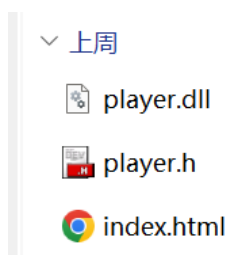


图 4

6. 在文件头引入#include “player.h”（客户端）

```
* client.c      server.c
1  #include <winsock2.h>
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <string.h>
5  #include <unistd.h>
6  #include <malloc.h>
7  #include "player.h"
8
```

图 5

7. 在初始化阶段调用 StartStreamingServer()函数（客户端）

```
24 // 初始化播放器库
25 StartStreamingServer();
26
```

图 6

8. 每收到一个视频分段后调用 ReceiveSegment(char \*video\_segement, const char \*file\_path, int file\_size)函数（客户端）

```
128 // 播放器缓存并在未来播放该分段
129 ReceiveSegment(video_segement, file_path, file_size);
```

图 7

9. 在 main 程序的最后返回前，调用 WaitEnd()等待播放器播放完视频并释放资源（客户端）

```
156 // 等待播放器播放完视频并释放资源
157 WaitEnd();
158 return 0;
```

图 8

10. 基于给定的视频文件列表，按照视频的播放顺序，向 server 发送请求（客户端）

- a) 使用 for 循环视频段序号，同时使用选择语句选择传输不同分辨率和码率的视频

```
53 int p = 360, k = 1000;
54 for (int index = 0; index < VIDEO_LEN; index++) {
136 // 传输不同清晰度的视频
137 if (index == 59) {
138 switch (p) {
139 case 360: index = 0, p = 480, k = 2500; break;
140 case 480: index = 0, p = 720, k = 5000; break;
141 case 720: index = 0, p = 1080, k = 8000; break;
142 case 1080: break;
143 }
144 }
145 }
```

图 9

- b) 利用 sprintf 函数拼接视频文件命名

```
***** 任务2：如何按顺序选择视频文件？ *****
if (sprintf(req, "ocean-%dp-%dk-%d.ts", p, k, index) < 0) {
    perror("Error");
    return -1;
}
*****
```

图 10

11. 接收 client 端地发送请求，并发送指定视频文件（服务端）

```
// 接收客户端下载请求
int bytes_recv = 0;
char req[REQUEST_SIZE] = "";
bytes_recv = recv(new_socket, req, REQUEST_SIZE, 0);
if (bytes_recv <= 0)
    printf("ERROR in recv\n");

// 读取视频文件
int buf_len;
while ((buf_len = fread(buffer, sizeof(char), BUFFER_SIZE, fp)) > 0) {
    // 发送视频文件
    int tmp_send_count = send(new_socket, buffer, buf_len, 0);
    if (tmp_send_count < 0) {
        perror("Error in sending video\n");
        exit(EXIT_FAILURE);
    }
    // 累计总字节数，用于判断是否发送完成
    send_count += tmp_send_count;
}
```

图 11

三、实验结果

任务一实验截图如下

1. 启动服务端

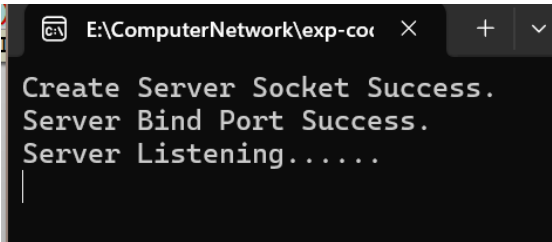


图 11

2. 启动客户端

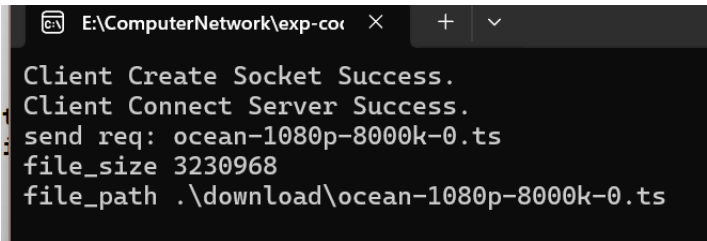


图 12

3. 传输结果

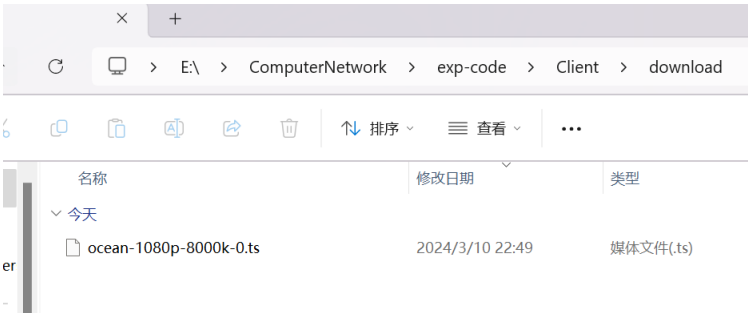


图 13

任务二实验截图如下

1.启动服务端

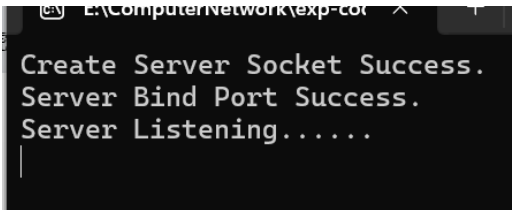


图 14

2.启动客户端

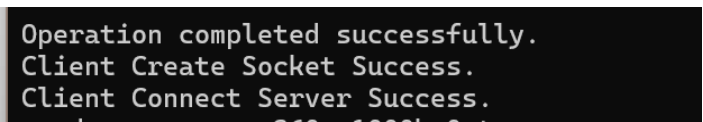


图 15

3.传输结果

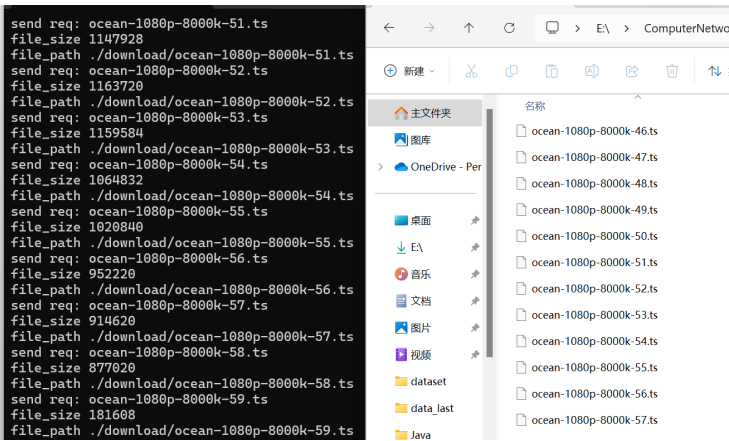


图 16

4.播放结果

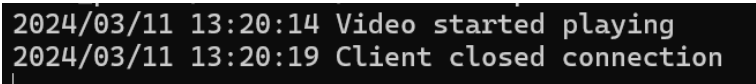


图 17

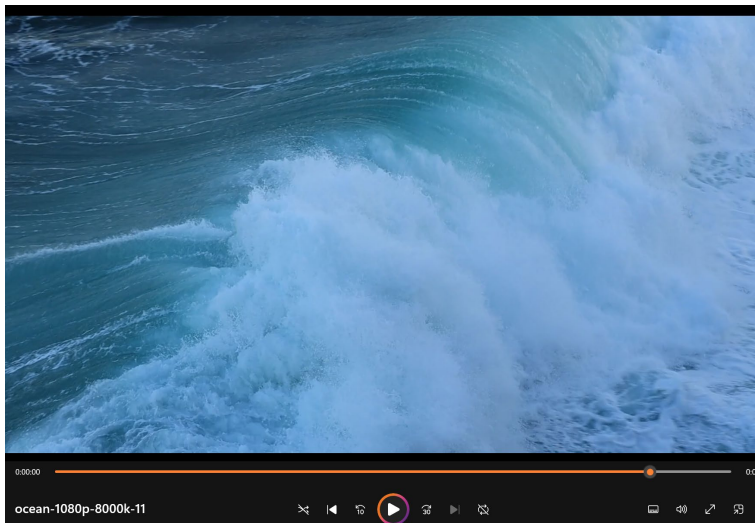


图 18

## 四、实验分析

1. 任务一的实验结果截图表明视频文件成功传输；
2. 任务二的实验结果截图表明多项视频文件成功传输，并且可以使用 m3u8 播放器播放 60s 的完整视频；
3. 观察到视频播放的时候有所卡顿，与预期流畅播放不符合。经查询资料发现，需要将.ts 格式的文件转化为.mp4 格式的文件才能流畅播放；
4. 第一版代码编写完成后，发现视频文件并不能被播放，仔细阅读代码发现，在客户端拼接视频时，使用了 strcat 进行拼接，该函数会自动在最末端加上 '\0' 导致视频文件损坏，更改为 memcpy 后，视频文件成功播放。

## 五、实验总结

1. 通过本次实验，掌握了基本 TCP 套接字的编程方法；
2. 了解了网络编程相关的一些库，并将其运用于代码中；
3. 学习了简单网络应用的编程思路；
4. 学会了如何针对性地查询相关资料。

## 六、思考题

1. 仔细阅读并理解所实现的代码，完成以下问题：

- 请根据任务2中进一步完善代码：server和client如何确定所有视频的传输已经完成，并分别结束视频的接收和发送，然后关闭连接和播放器？请在代码中实现，并解释和说明；
- 请分别结合任务1和任务2所实现的代码，解释并说明在server向client传输单个视频时，使用视频文件长度和文件结束终止符号的作用是什么？如果去掉相应的判断会出现什么现象？还有什么方法能够解决出现的问题？
- 在任务2中，我们将一个长的视频段切分为小的视频段进行传输，那么在TCP协议和IP协议中，又是如何将一个完整的文件切分为独立的数据包进行传输的？其数据包中的数据单元大小限制是多少？如果超出了限制，协议中是如何处理的？那么，在TCP/IP协议中，如何传输超大型文件（大于2G），其中存在的问题是什么，如何解决？
- 请自行调查和学习TCP/IP协议中，数据分片的相关资料，同时结合上述几个问题的内容，谈谈你的理解；

1) server 端：在接收到 END 信号后就关闭该端口号的读写操作，同时退出循环

```
/****** 任务2（扩展）：如何根据client信号，终止传输并退出循环？
if (!strcmp(req, "END")) {
    shutdown(new_socket, SD_BOTH);
    break;
}
```

图 19

client 端：发送 END 信号，如发送失败则进行相应的处理，否则关闭该端口号的读写操作。

```
/****** 任务2（扩展）：如何在视频流传输完成后，通知server结束视频传输？
strcpy(buffer, "END");
if (send(sock, buffer, strlen(buffer), 0) <= 0) {
    perror("Error in end");
    return -1;
}
shutdown(sock, SD_BOTH);
```

图 20

2) 使用视频文件长度的作用是准确确定数据量, 以便将视频文件分成多段发送, 同时保存完整文件。

文件结束终止符号的作用是正确解析接收的数据, 并知道在何时停止接收数据, 并将当前数据写入文件。

如果去掉则不能确定文件的大小, 也无法确定文件是否传输结束;

其他解决办法: 使用分块传输编码; 采用流式传输协议。

3) 当完整文件需要传输时, TCP 协议会将文件分割成 TCP 段, 每一段包含文件的一部分;

数据包中的数据单元大小收到最大传输单元的限制, 如果超出限制, TCP 协议会将它们分段, 并通过序列号和确认号确保它们正确到达目的地;

传输超大型文件: 将文件分割成多个块, 每一个块都由 TCP/IP 协议封装, 通过序列号和确认号确保它们正确到达目的地。

存在的问题: 如果为 32 位的操作系统, 受文件大小控制, 传输过程中可能出现未知错误; 导致系统资源被耗尽; 在网络带宽有限的情况下, 传输大型文件会导致网络拥塞和传输效率降低;

解决办法: 在传输前将文件压缩; 同时传输文件的多个不重复的部分。



## 指导教师批阅意见

## 成绩评定

指导教师签字：

年 月 日

- 注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。  
2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。