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

# 实验目的与要求

* + - 1. 实验目的
         1. 理解TCP套接字的定义；
         2. 掌握基本的TCP套接字编程方法；
         3. 了解简单网络应用的编程思路；
         4. 了解网络编程相关的一些库。
      2. 实验要求
         1. 了解网络编程的相关库；
         2. 掌握编写简单网络应用的技能；
         3. 依照步骤完成实验内容；
         4. 对实验结果截图；
         5. 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。

# 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

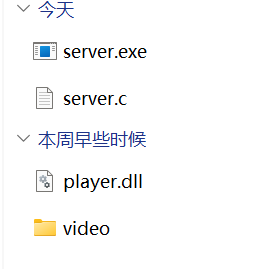
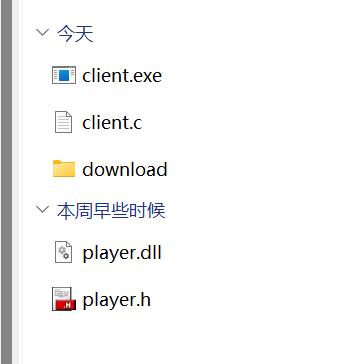


图1

* + - 1. 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。

* + - 1. 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

1. 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；
2. 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；
3. 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。
4. 代码截图如下

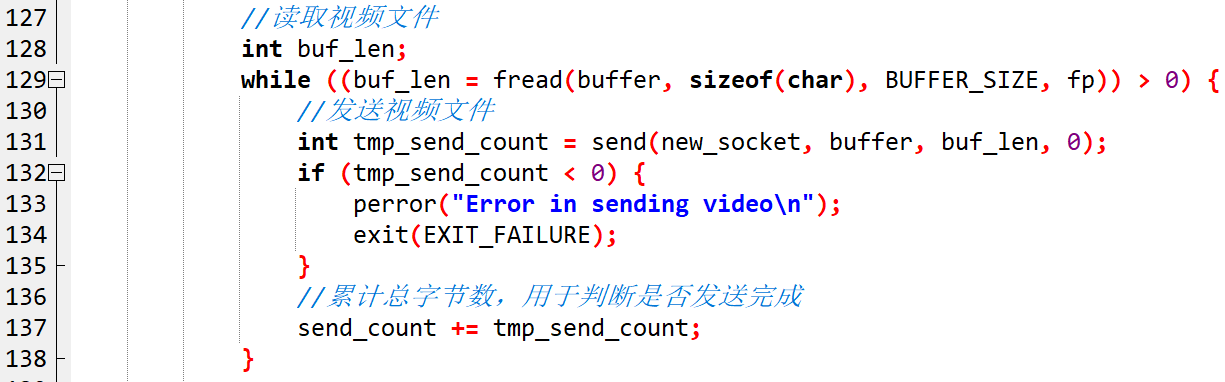


图2

* + - 1. 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

1. 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；
2. 因为以char类型读取数据，影刺使用strcat函数拼接视频文件；
3. 最后使用recv\_count累加tmp\_recv\_count，判断是否接收完成；
4. 代码截图如下

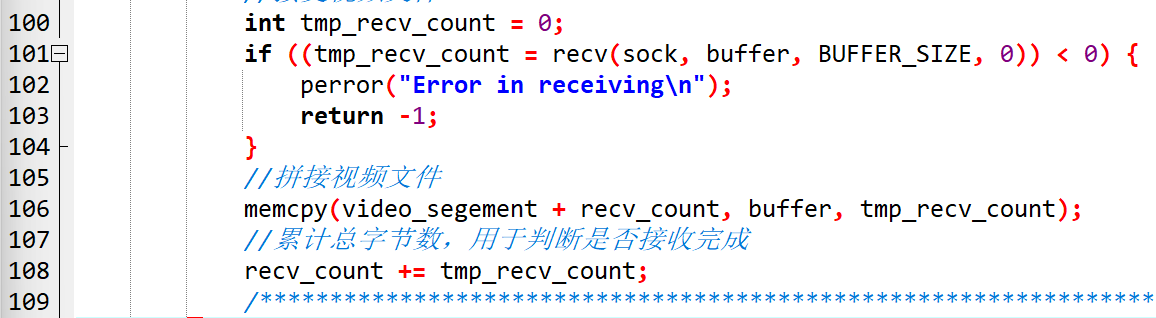


图3

**任务二：实现流式视频传输**

* + - 1. 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

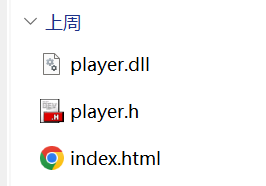


图4

* + - 1. 在文件头引入#include “player..h”（客户端）

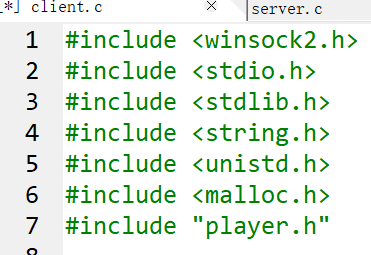


图5

* + - 1. 在初始化阶段调用StartStreamingServer()函数（客户端）

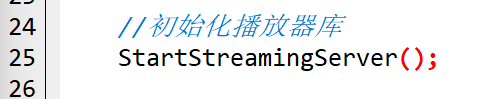


图6

* + - 1. 每收到一个视频分段后调用ReceiveSegement(char \*video\_segement, const char \*file\_path, int file\_size)函数（客户端）

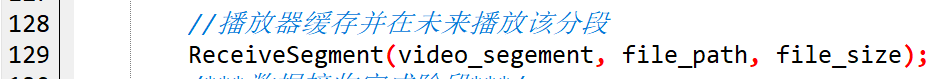


图7

* + - 1. 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）

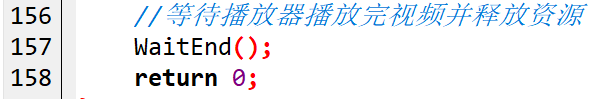
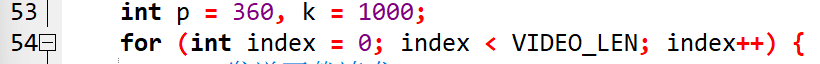


图8

* + - 1. 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）
         1. 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



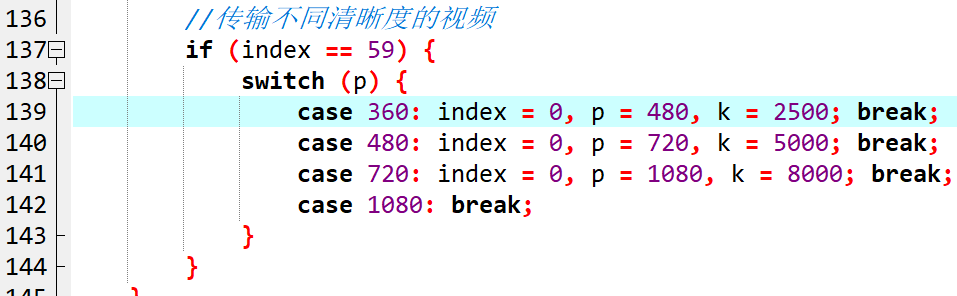


图9

* + - * 1. 利用sprintf函数拼接视频文件命名

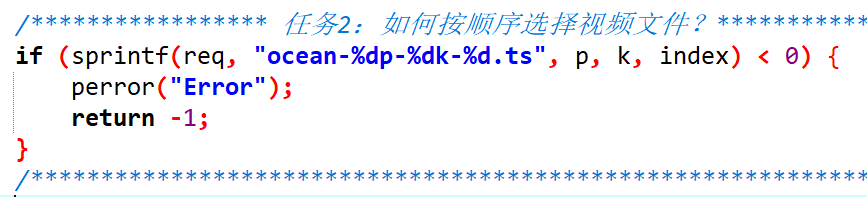
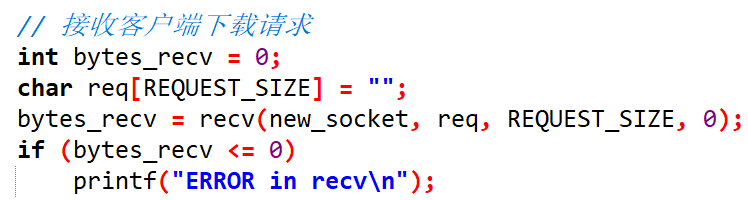


图10

* + - 1. 接收client端地发送请求，并发送指定视频文件（服务端）



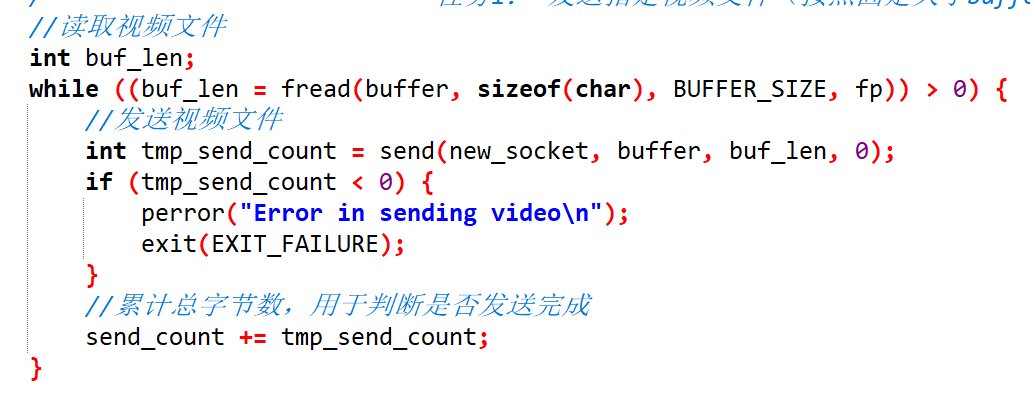


图11

# 实验结果

**任务一实验截图如下**

* + - 1. 启动服务端



图11

* + - 1. 启动客户端

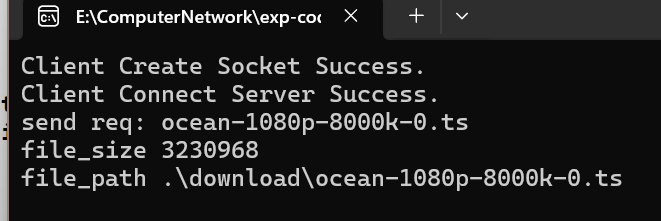


图12

* + - 1. 传输结果

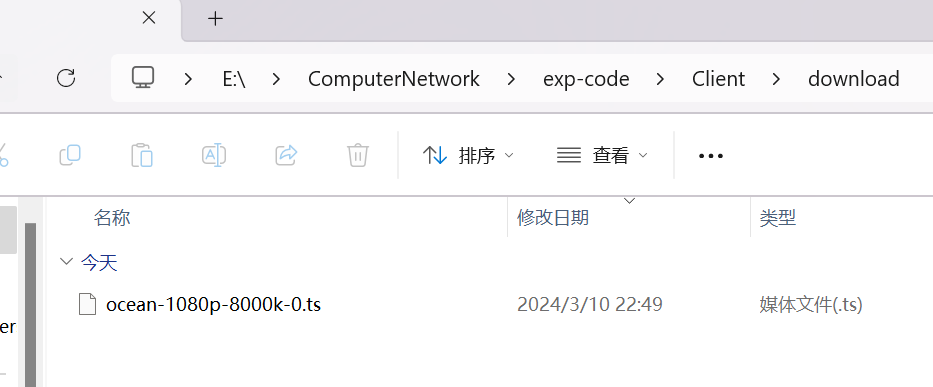


图13

**任务二实验截图如下**

1.启动服务端

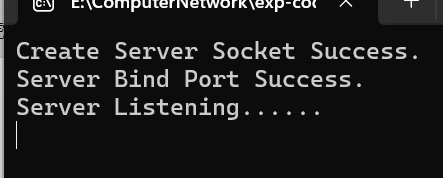


图14

2.启动客户端

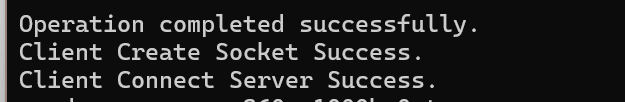


图15

3.传输结果

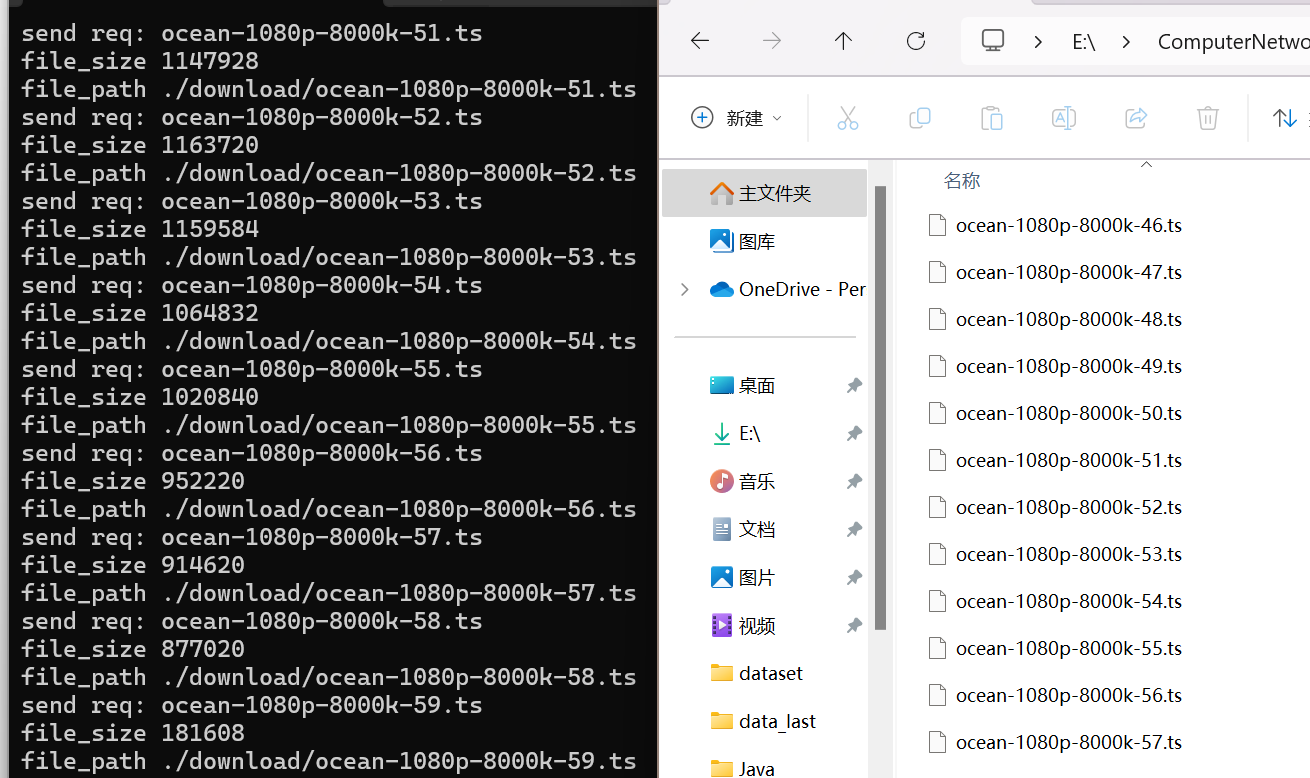


图16

4.播放结果

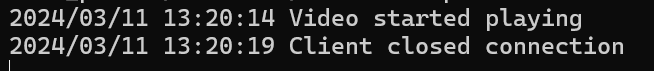


图17

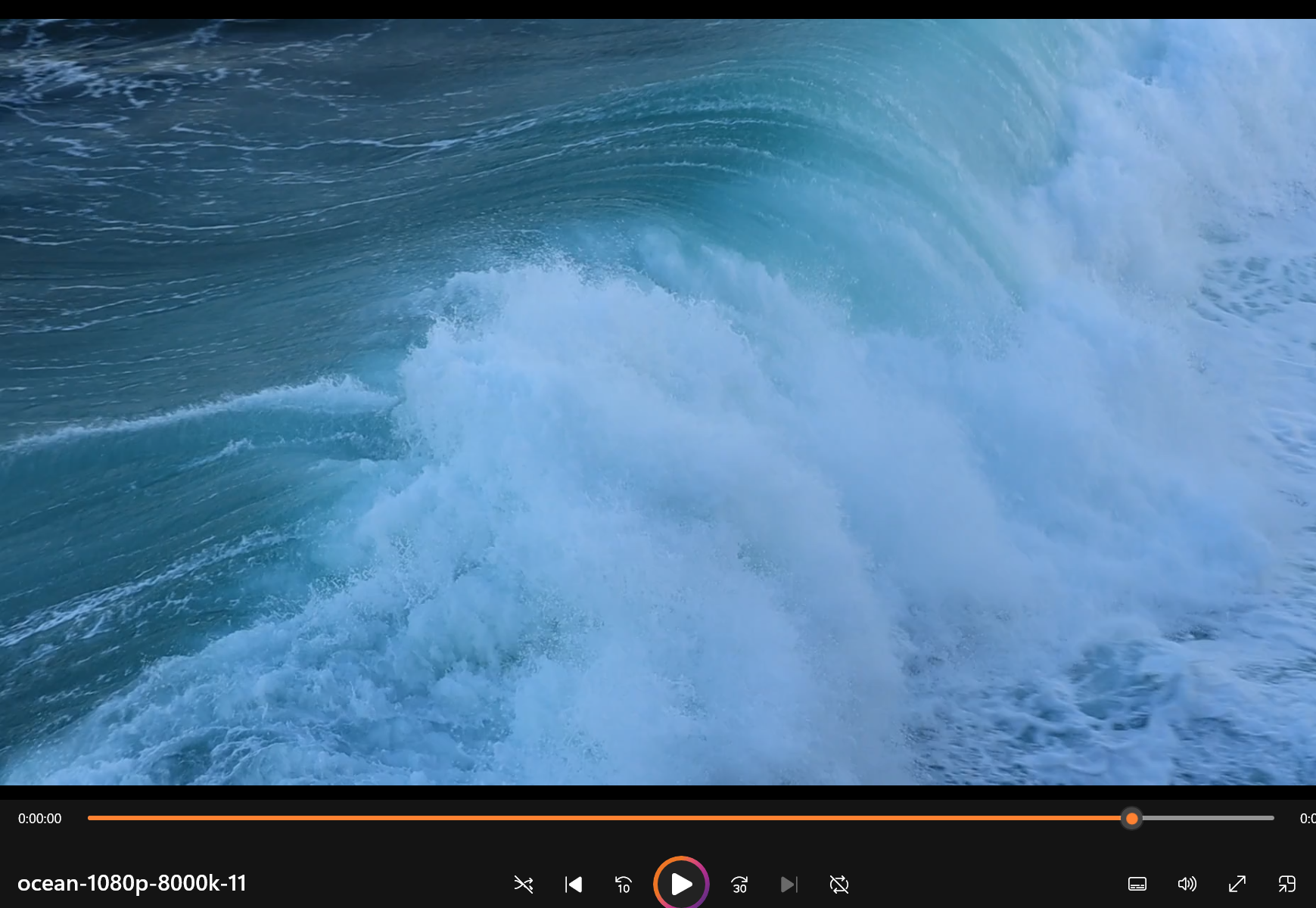


图18

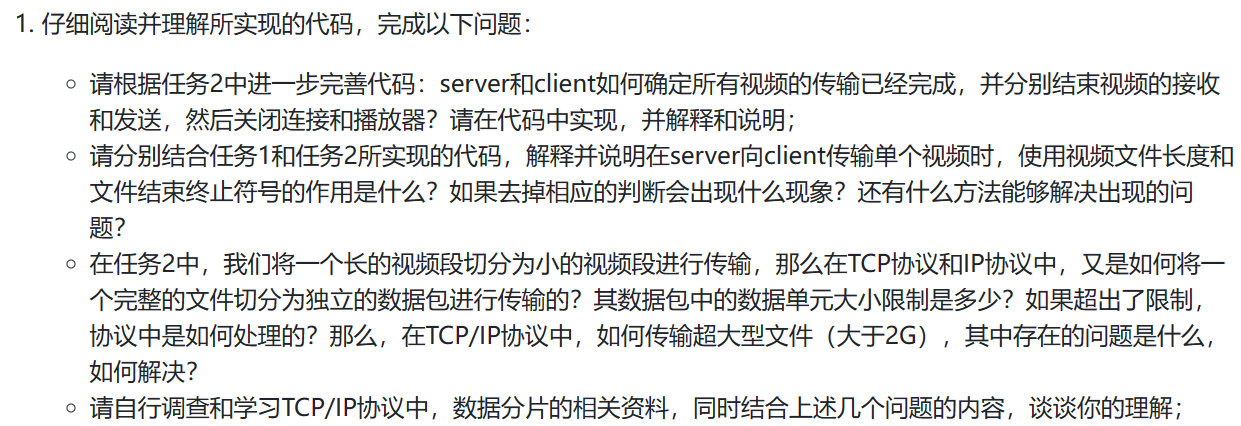
# 实验分析

* + - 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。

# 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。

# 思考题



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

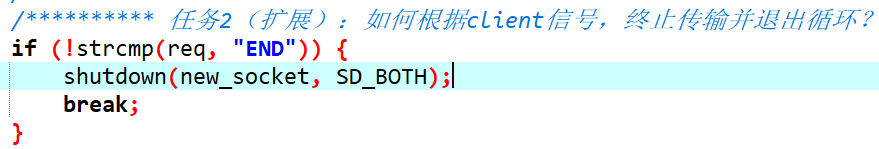


图19

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。

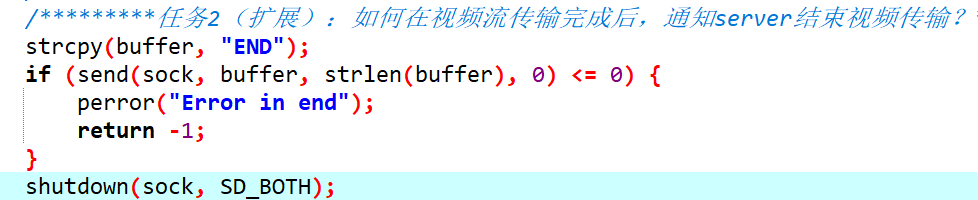


图20

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。