

Project 1: 网络文件传输

1 Introduction

在本次实验中需要实现一个简单的网络文件传输系统。本次实验的目的为熟悉 socket 编程，以及熟悉网络协议的工作原理。如果你对 socket 编程不是很熟悉，可以参考我们提供的资料(<http://netlab.csu.edu.cn/index8>)

2 Protocol

文件传输系统分为两部分，服务端和客户端。要求实现的功能包括：

1. 上传文件
2. 下载文件
3. 获得服务器上的文件列表

本次实验中数据的传输采用 TCP 协议。我们会给出服务端，ip 地址为 **202.197.61.231** 端口为 **10086**

(注：请在校园网内访问服务端，宿舍网络即为校园网，外网访问会被防火墙拒绝，**请不要上传超过 1MB 的文件**)

2.1 客户端请求服务端的消息格式

客户端发送给服务端的消息为以下 3 种格式：

注：变量有先后次序

Get:// 下载文件

```
uint8_t type = 0;  
uint8_t sizeofFilename;  
char filename[sizeofFilename];
```

ListAll://获得服务器上的文件列表

```
uint8_t type = 1;
```

不能上传超过 **1MB** 的文件

Put://上传文件

```
uint8_t type = 2;  
uint8_t sizeofFilename;  
uint32_t filesize;  
char filename[sizeofFilename];  
byte file[filesize]
```

注 1: `uint8_t` 为无符号 8 位整型, `uint32_t` 为无符号 32 位整型 (在 C 中通过引入 `#include <inttypes.h>` 来使用, 也可使用其他语言, 但请严格遵守消息格式并使用字节流传输)。Get 消息通知服务器需要下载的文件的文件名, ListAll 消息用来获得服务器上的文件列表, Put 消息通知服务器上上传文件的文件名以及文件大小。

注 2: 每次 get 和 put 成功之后, 服务器都会在完成上传或者接收字节流之后, 返回一个信息 (下面的 error 或者 success), 需要在上传或者下载的最后, 接受这个信息 (不然可能会导致下一次接收服务器发送的信息时出错), 并且根据第一个 reply type 来确定是错误信息还是成功信息。ListAll 信息则不会返回额外的信息。

2.2 服务端响应客户端的消息格式

服务端响应客户端的消息格式

注: 变量有先后次序

Error://服务器处理错误时返回

```
uint8_t replyType = 0;  
uint8_t sizeOfInfo;  
char info[sizeOfInfo];
```

Success:// 服务器处理成功时返回

```
uint8_t replyType = 1;  
uint16_t sizeOfInfo;  
char info[sizeOfInfo];
```

Stream://服务器返回的文件流

```
uint8_t replyType = 2;  
uint32_t sizeOfData;  
byte info[sizeOfData];
```

不能得到超过 4G 的文件

Error 为服务器处理错误时返回，包括客户端发送的消息不合法，下载文件名不合法，以及上传文件名不合法等。

Success 为服务器处理成功时返回，包括请求服务器文件列表成功，上传文件成功等

Stream 为下载文件时服务器返回的文件流。