TCP编程报告

程序使用与架构

客户端服务器使用方法见`README.md`

程序使用了`spdlog`第三方库来进行日志打印

`client/`与`server/`分别为客户端，服务器所在源码目录

协议描述

实现了断点重传

客户端与服务器通信主要依赖如下的结构

```
struct msg_t {
    int len;  //为整个结构的长度，注意不一定为sizeof(msg_t)，因为data有效长度一般小于4096
    char type;
    char data[4096];
};
```

针对每种`type`，其`data`格式也不一样，如下表所示

type	data格式	描述
Upload	filename file_len is_continue	file_len为文件总长度，is_continue指明是否断点续传
Download	filename is_continue current_file_len	is_continue指明是否断点续传，current_file_len为当前客户端已有文件的长度
UploadRespond	current_file_len	current_file_len指明服务器端已有文件的长度
DownloadRespond	file_len	file_len为服务器端文件总长度
UploadRespondWarn	current_file_len message	current_file_len指明服务器端已有文件长度，message为警告信息
DownloadRespondWarn	file_len message	file_len为服务器端文件总长度，message为警告信息
UploadRespondErr	message	message为出错信息
DownloadRespondErr	message	message为出错信息
DATA	文件内容	文件内容

服务器端

服务器端使用多进程的模式进行工作，当有一个连接连入时，服务器`fork`一个新的工作进程对这个连接进行处理，原来进程继续等待新的连接

服务器端工作进程流程如下

1. 接受客户端的msg
    1. 若请求为Upload
        1. 尝试打开文件，若打开成功且客户端要求断点续传，则将current_file_len置为文件长度，否则置为0
        2. 若文件存在但客户端不要求断点续传，则文件将会被覆盖，向客户端发送UploadRespondWarn消息，否则发送UploadRespond消息
        3. 不断接受DATA类型消息，写入文件直到写完
        4. 结束并退出
    2. 若请求为Download
        1. 检查文件是否存在，若不存在，则向客户端发送DownloadRespondErr消息，并退出
        2. 若客户端要求断点重传，则尝试进行文件指针偏移，若此时失败，则向客户端发送DownloadRespondErr消息，并退出
        3. 向客户端发送DownloadRespond消息，告知其文件长度
        4. 不断地读文件，并想客户端发送DATA消息，直到文件发送完
        5. 结束，退出
    3. 若为其他请求，报错，退出

客户端

客户端为单进程，由命令行参数指定连接服务器的地址，端口，上传/下载，文件名，是否断点重传

客户端流程如下

1. 若为上传
    1. 尝试打开文件，若文件不存在，报错并退出
    2. 向服务器发送Upload消息，告知其文件名，文件长度，是否断点续传
    3. 接受服务器的UploadRespond/UploadRespondWarn/UploadRespondErr消息，收到其他类型消息则报错，退出
        1. 若为UploadRespondErr消息，则报错，退出
        2. 若为UploadRespondWarn消息，则打印警告信息，获得当前服务器文件长度
        3. 若为UploadRespond消息，则获得当前服务器文件长度
    4. 打开文件，不断发送DATA消息传递文件内容给服务器，直到文件传完
    5. 结束，退出
2. 若为下载
    1. 若为断点续传，则尝试打开文件并得到当前文件长度，设置current_file_len，否则将其置为0
    2. 向服务器发送Download消息，告知其文件名，是否断点续传，当前文件长度
    3. 接受服务器的Download/DownloadWarn/DownloadErr消息，其他消息将报错，退出
        1. 若为DownloadErr消息，则打印出错信息，退出
        2. 若为DownloadWarn消息，则打印警告信息，得到文件总长度
        3. 若为Download消息，则得到文件总长度
    4. 不断地接受服务器的DATA消息，写入文件中，直到写完
    5. 结束，退出

异常处理

地址重用

服务器端设置了`SO_REUSEADDR`选项，使得服务器在崩溃之后，能在2MSL之内就能重启服务

僵尸进程

当服务器端工作进程结束后，它不会自动被杀死，而是称为僵尸进程，这里重写了服务器端的`SIGCHLD`信号处理，使用`waitpid`来杀死所有结束的工作进程

accept处理

服务器端accept考虑在出现`EINTR`，`ECONNABORTED`和`EHOSTUNREACH`错误时重新accept，这样使得其能正常给多个用户提供服务

发呆连接

由于服务器端是独占进程的读写，这里设置socket选项`SO_RCVTIMEO`和`SO_SNDTIMEO`来进行有时限的读写，时限设置为了2min

SIGPIPE信号忽略

服务器将忽略`SIGPIPE`，这样使得其不会异常退出

SO_LINGER选项

服务器设置SO_LINGER选项，从而避免过多的FIN_WAIT_2类型的连接