

FPT 实验报告

黄予 2013011363 计 34

一、实验要求

在 Linux 系统上使用 Socket 实现 FTP 客户端和服务器程序，使客户端可以连接至服务器，可以进行 FTP 的基本操作。

二、实现

1. 程序实现了 get、put、pwd、dir、cd、?、quit 七种命令，其中 get、put 命令均通过了 800M 的大文件传输测试；cd 命令支持..回退。程序还支持多客户端连接，并支持客户端同时下载或上传不同文件，客户端数目上限设置为 20。
2. 采用被动模式。为避免与系统 ftp 冲突，本程序命令连接使用 8001 端口，而不使用 21 端口；数据端口由服务器随机选择未占用端口告知客户端，由客户端向服务器发起建立数据连接的请求。
3. 每次开始文件传输(get、put 命令)都需要重新建立数据连接，通过一方断开连接标志文件传输结束。文件传输采用流模式，传输 buffer 大小设置为 4096 字节。
4. 服务器端的主要流程为，主线程监听 8001 端口，当一个客户端请求连接时，服务器都会新建一个线程负责为该用户服务，主线程继续监听。

三、源代码说明

1. Client.cpp Client.h: 客户端程序
2. Server.cpp Server.h Service.cpp Service.h: 服务器程序，其中 Server 为主线程，负责监听 8001 端口，管理所有客户端的连接；Service 负责管理与一个客户端的连接，为客户端提供七种命令的服务。
3. Common.cpp Common.h: 一些客户端与服务器通用的函数和常量放置于此，例如文件传输、字符串分割、地址设置等等。
4. HandleError.cpp HandleError.h: 负责处理错误的类。
5. Server_main.cpp Client_main.cpp: 分别为运行服务器和客户端的入口，即 main 函数。
6. 运行 makefile 会生成 Server_main.out 与 Client_main.out 可执行文件，在 Linux 下先运行服务器（./Server_main.out），再运行客户端（./Client_main.out）即可。
7. 源代码见附件

四、遇到的问题

1. 本实验开始采用块传输模式，采用传输头标记文件传输是否结束，但是由于未知的原因，当文件为几十 M 或者更小时，可以正确传输；但是当文件为几百 M 时，传输到一半文件头会出错，导致传输终止。调试一天无果，最后采用主流的断开连接标记文件是否传输结束。