

FTP大作业实验报告

软61_2016080039_金性佰

一. 实验环境

CPU: Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz

RAM: 16.00GB

Language: C

二. 实验目的

1. Support integral large file transmission;
2. Support connections from multiple clients;
3. Resume transmission after connection terminated;
4. File transmission without blocking the server, GUI;
5. Able to log in a provided commercial FTP server and download/upload files;
6. Friendly GUI

三. FTP

1. FTP运行方法:

1)server

在terminal实行makefile, 就生成server文件。打/server或/server -port n -root path可以运行server文件。

2) client

client是双击exe文件可以运行。运行后插入正确的信息, 可以连接到服务器

2. FTP工作流程:

FTP客户端: 创建 Socket -> 连接至服务器 -> 输入信息 -> 等待响应 -> 关闭Socket

FTP服务端: 创建 Socket-> 监听 -> 接收连接的请求 -> 接受并处理命令 -> 关闭Socket

3.FTP实现功能:

本次实验实现了USER, PASS, RETR, STOR, QUIT, SYST, TYPE, PORT, PASV, MKD, CWD, PWD, LIST, RMD, RNFR, RNTD等指导文件中提到的所有功能

USER: 输入用户名 (默认为anonymous)

PASS: 输入密码（默认为anonymous@）

PORT: 模式配置active模式。（服务端连客户端）

PASV: 配置passive模式。（客户端连服务端）

RETR: 从服务器下载文件

STOR: 上传文件到服务器

QUIT: 结束FTP

ABOR: 结束FTP

MKD: 创建一个文件夹

CWD: 进入文件夹

RMD: 删除空文件夹

LIST: 返回当前目录的文件和文件夹。

RNFR: 选择要改变名字的文件（与RNTO一起使用）

RNTO: 输入想要的新的文件名（与RNFR一起使用）

PWD: 显示当前路径

4.多个用户连接

由fork函数利用multi-process方式实现了多个用户的连接。我做的服务器可以接受最多30位用户。

四. 功能实现

1. Support integral large file transmission;

在检测作业中可发现好几GB的文件也能传送。利用PORT和PASV方式另外port上连接，传送文件binary值，

2. Support connections from multiple clients;

多个客户端的连接使用了fork函数，每一个客户端账号密码正确输入之后，使用fork来创建线程。

3. Able to log in a provided commercial FTP server and download/upload files;

我做的服务器可以连接到助教提供的服务器。

4. Friendly GUI

客户端的GUI使用了Windows的QT。很容易利用ftp上传文件和下载文件。

五.实验终结

本次实验的资料和文档大多数是英文的，理解花了几天。所以不仅理解UDP和FTP的能力，英语实力也好像有所提高。做UDP作业的时候很简单，但TCP作业部分开始后，我感觉真的要死了。需要符合一些

规格，感觉明明写对了但是真实操作中发现不能正常工作，每次遇到莫名其妙的bug，而且也不是每一次发生。。虽然本次作业很痛苦，但是收获也很多，对FTP的理解更深了一步，服务端和客户端是怎么互相交流的等等。但是只是因为其他作业，时间不够，没能好好完成，所以很遗憾。。