***Project report***

Final Project-FTP Proxy

**Course Title:**  *Internet Applications*

**Name:** *LIU Mengqi (2013213062*

**Date:** *June 17th, 2016*

# Overview

* 1. Deeply understand the related knowledge of FTP (File Transfer Protocol).
  2. Complete a FTP proxy program based on Linux command line terminal.

The requirements for this project are:

1. The FTP server can be set up using the existing software, for example FileZilla server. For the FTP client, students can use one general FTP client tool software, for example FileZilla client.
2. The FTP proxy performs as both FTP client and FTP server. FTP client will connect FTP proxy first and send the FTP requests. FTP proxy is able to receive the requests, and then forward the requests to FTP server. After that, it can receive the replies from the FTP server, and then forward the replies to the FTP client.
3. FTP proxy has to listen at port 21.
4. FTP proxy is able to set up separate control connections with FTP client and FTP server separately.
   1. FTP proxy is able to receive the commands from FTP client using control connection. The commands include: PWD, CWD, LIST/MLSD, MDIR, DELE, RNFR/RNTO, RETR, and STOR. And it can resolve the commands and modify if necessary (i.e. modify the parameters for PORT), and then forward the commands to the FTP server using control connection.
   2. FTP proxy is able to receive the FTP replies from FTP server using control connection. And it can resolve the replies and modify if necessary (i.e. modify the parameters in the replies for PASV), and then forward the replies to the FTP client.
5. FTP proxy is able to set up data connections if required with FTP client and FTP server separately. In this project, cache mechanism will be used in FTP proxy (Only pdf file or image is required in this case). If the file that FTP client wants to download already exists in the cache, FTP proxy will not download the file from FTP server but send the file in the cache to FTP client. And no data connection will be set up between FTP proxy and FTP server. Otherwise, data connection will be set up between FTP proxy and FTP server.
   1. If FTP client wants to upload a file, FTP proxy can receive the file from FTP client and then upload the file to FTP server.
   2. If FTP client wants to download a file, FTP proxy can receive the file from FTP server and forward the file to FTP client. Meanwhile, the file will store in its cache.
6. For data connection, it is required to implement ***both passive mode and active mode***.
7. Detailed design document and user manual.
8. Detailed annotation of code and nice programming style.
9. Stable and friendly to users, and be able to handle error commands.

# Requirements Analysis

* 1. Environment

### 2.1.1 Ubuntu

本次实验的环境为Ubuntu System，需要安装VM workstation将Ubuntu虚拟机导入

使用Ubuntu虚拟机时，需要对Linux系统的命令熟练掌握

### 2.1.2 FTP Client and Server

（这一部分的配置工作在你们之前没做的FTP里面都有详细，可以参考那个updateFTP进行填充）

本次实验要求FTP传输的客户端和服务器端，客户端和服务器端都采用现有的软件工具进行配置完成，只有Proxy服务器需要自己进行编写

服务器 SERVER：

sudo -s launchctl load -w /System/Library/LaunchDaemons/ftp.plist

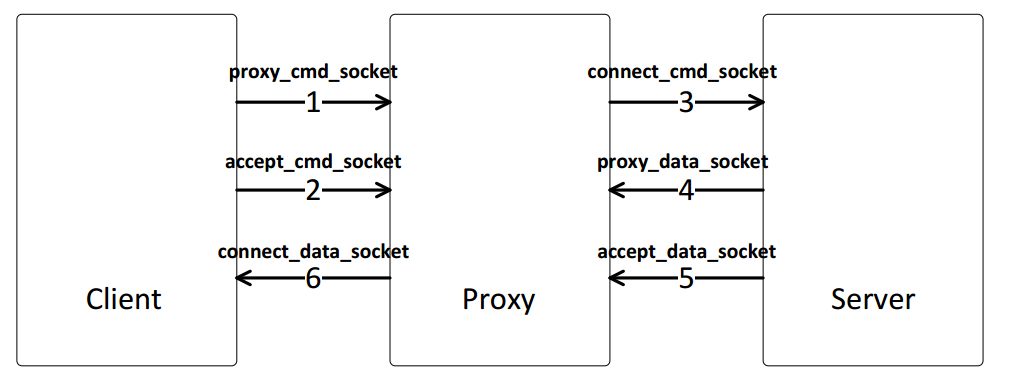
客户端 CLIENT：FileZilla Client

* 1. Basic Function

功能需求：

1. FTP服务器先连接Proxy，发送FTP请求。代理接收到请求，之后将请求转给服务器端。之后，能够收到服务器的回复，将回复转发给FTP客户端。
2. FTP代理能够通过control connection接收到来自客户端的command请求，其中包含：PWD, CWD, LIST/MLSD, MDIR, DELE, RNFR/RNTO, RETR, and STOR.
3. FTP代理能接收到FTP服务器的回复，可以解析，如果有必要进行修改（修改PASV参数），再转发给客户端
4. FTP代理需要有缓存机制（PDF和图片）。如果客户端想下载的文件已经存在在缓存中，代理就不需要再从服务器下载。
   * 1. 客户端上传文件，代理能收到文件，并上传至服务器
     2. 客户端向下载文件，代理能从服务器下载，转发给客户端
5. 数据传输data connection，需要实现passive mode和active mode（被动模式和主动模式）
6. **Preliminary Design**

## Modular Decomposition



**MODULE 1:**

登录信息，包括登录服务器和客户端

**MODULE 2:**

连接信息，客户端与代理连接，代理与服务器连接

First we are going to establish a communication path among client, proxy and server for the exchange of commands and replies.

The connection process use socket structure. Among their communication, all the types’ data transmission use socket type with required information, like IP address or port number.

**MODULE 3:**

socket传输设置

**MODULE 4:**

控制链路的建立control connection

We operate under command connection and for each commands, it will generate at least one reply 3-digit code followed by delimiter and text message and we can distinguish it to enter different operations.

**MODULE 5:**

数据传输链路的建立 data connection

We set a full duplex connection over which data is transferred, in a specified mode and type. The data transferred among server, proxy and client. And the data connection can be passive or active.

Client can upload or download files from server through proxy.

**MODULE 6:**

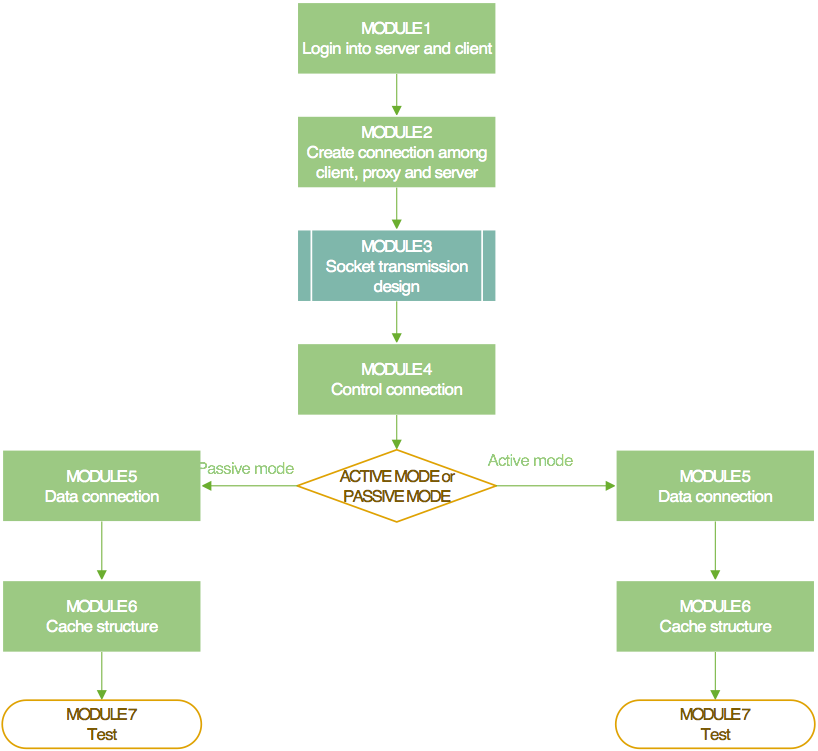
缓存机制的建立 cache structure

We set a structure which can judge whether proxy has this file or not. For proxy, it can check whether I have saved this file or not. If yes, with client’s request, it will transmit file to client. If not, it will download the file from server, and transmit it to client. At the same time, it will save it in its own content.

**MODULE 7:**

测试程序，测试运行结果

## Relationship between Modules



1. **Detailed Design**

## Communication

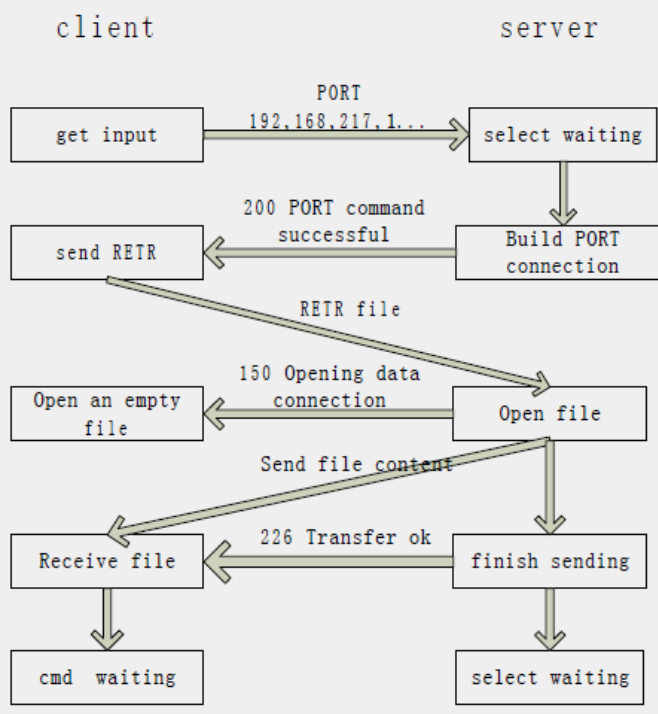
客户端和服务器连接代理的步骤：

1. 服务端通过命令行开启
2. 客户端通过FileZilla这个已有的软件
3. Proxy搭建在Ubuntu系统之上
4. 开启proxy\_cmd\_socket、bind（）、listen操作
5. 开启proxy\_data\_socket、bind（）、listen操作
6. 将proxy\_cmd\_socket加入master\_set集合
7. 将proxy\_data\_socket加入master\_set集合
8. select循环监听，这里只对读操作的变化进行监听（working\_set为监视读操作描述符所建立的集合）
9. 判断selectResult是否大于零。selectResult > 0 时 开启循环判断有变化的文件描述符为哪个socket
10. 判断变化的文件描述符是否存在于working\_set集合，如果在，执行accept操作,建立proxy和客户端之间的控制连接
11. 执行connect操作,建立proxy和服务器端之间的控制连接

## Upload and Download Files

Upload files:

Before data transmission, the system will establish data connection according to passive or active mode, client sends control word “RETR” attached with the file name. After successfully creating the file, server will return 150 feedback message. Then use function ftp\_get( ) to create the file at local position, receive data from the server and write it into the new created file.



Download files:

Before data transmission, the system will establish data connection according to passive or active mode, client sends control word “STOR” attached with the file name. After successfully creating the file, server will return 150 feedback message. Then use function ftp\_put( ) to read data out from the local file and send it over the data connection.

## Cache Structure

缓存设计的原理分析：

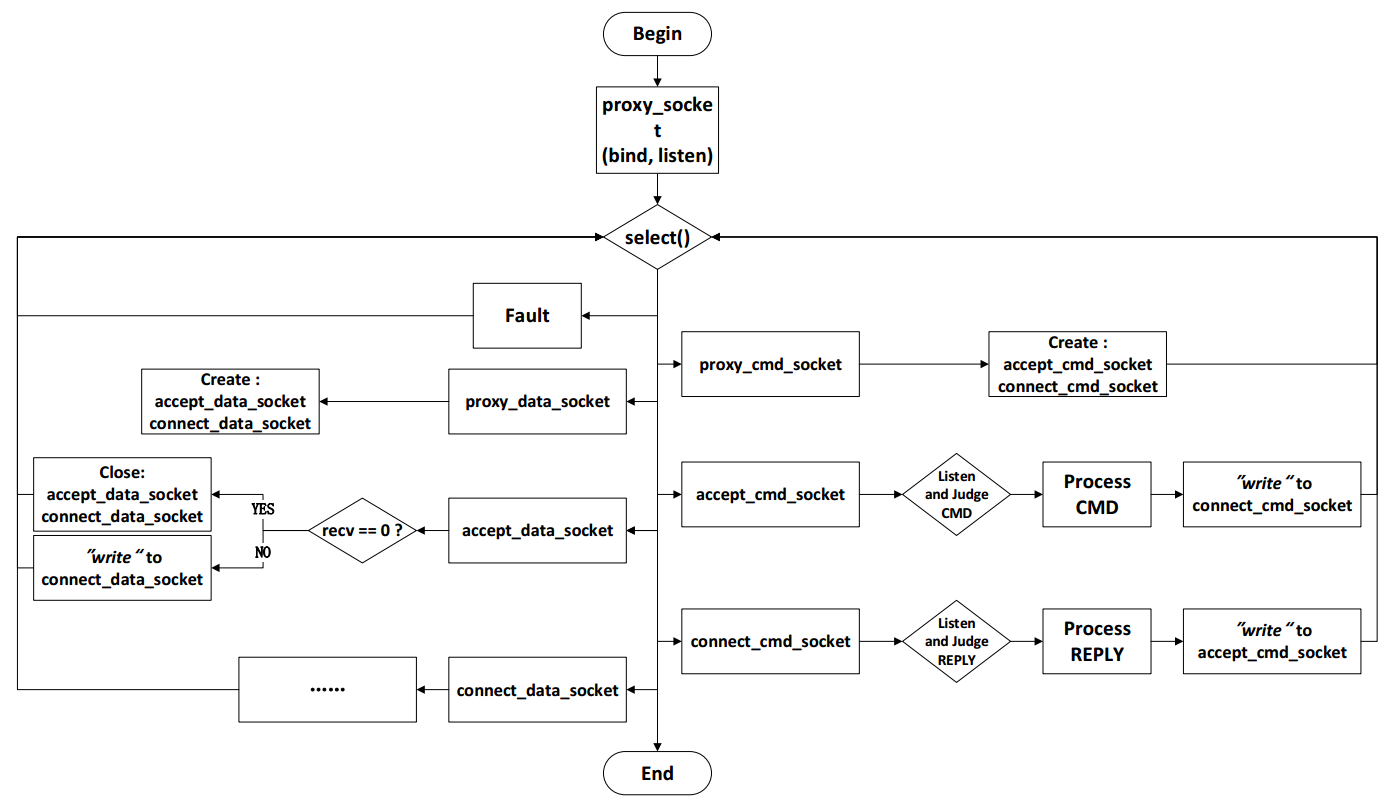
1. 客户端向服务器请求下载某个文件
2. 消息被proxy接收，判断当前缓存目录之下是否有该文件名
3. 如果判断有该文件名，proxy直接将文件传递给客户端，客户端从proxy成功下载文件
4. 如果判断没有该文件名，proxy将客户端的请求转发给服务器，服务器将文件传输至proxy，proxy判断自己没有此文件，缓存的同时，将文件再转发给客户端，此情况下，客户端从服务器成功下载文件

## Active and Passive Mode

Active模式：

1. 客户端发送请求给服务器，携带有自身的IP地址和端口号，以192，100，125，83， 0，1024（只是个例子）的形式，前四位为IP地址，后两位为和端口号相关，第一位\*256+第二位为客户端主动开放的端口号
2. proxy接收到客户端的请求，进行解析，此时发送给服务器的请求需要更换为自己的IP以及自己的port number，将自己作为客户端进行发送
3. 服务器接收到请求之后进行应答，proxy收到
4. Proxy将消息再转发给客户端
5. 经过这一系列的control connection之后，将主动模式的数据通道data transmission建立好，之后的数据传输就在该建立好的数据通道上进行

流程图：



Passive模式：

1. 服务器将IP地址和端口号信息发送给客户端，proxy收到消息
2. Proxy对消息进行解析，保存服务器的IP地址和端口号
3. Proxy向客户端主动暴露自己的IP地址和端口号，发送消息给客户端
4. 客户端收到消息，向proxy的指定IP地址和端口号发送消息
5. Proxy接收到消息之后，根据记录的服务器的IP地址和端口号，再讲消息转发给客户端
6. 至此，被动模式的数据传输通道被建立，之后的被动模式下的数据传输都根据该数据通道进行

（这块看要求需要补充一个流程图）

1. **Results**

## Connection among Proxy, Client and Server