项目文档

——计网第一次大作业

一．基本信息

姓名：邵钰杉

班级：软件 51

学号：2014011003

二．实验环境

虚拟机 VMware

操作系统：Ubuntu 16.04

内存：4G

处理器：Intel Core i7-7700HQ

三．实现的指令

1. **USER** & **PASS:** Use USER [username] and PASS [password] to log in the FTP server. USER anonymous allows guests loging in. User verification is implemented and user-table is stored in server/src/user.config.

2.**PORT** & **PASV**: PORT [IP and PORT] or PASV are used to set the mode of FTP server. If PORT is set, the client opens a socket and waits the server to connect. If PASV is set, the server sends IP and PORT to the client and waits for the client. PORT and PASV is one-time used.

3.**RETR**: RETR [file] retrieval [file] from FTP server.

4.**STOR**: STOR [file] upload [file] to FTP server.

5.**SYST**: on receiving a SYST command, return the string “215 UNIX Type: L8” to the client.

6.**TYPE**: on receiving a “TYPE I” command, return “200 Type set to I.” to the client. On receiving some other TPYE command, return an appropriate error code.

7.**LIST**: LIST [path/file] asks server to list the file or directories in target path or show

information of file. If empty paremeters are passed, the server show the list of the current name prefix. Usually, there could be a number of files or directories in a directory. So, the server use a file-transfer style to response to the LIST verb. And before List, PORT or PASV should be set.MKD: MDK [dir] create a new directory of server.

8.**CWD**: CWD [path] asks the server to set the name prefix to this pathname, or to another pathname that will have the same effect as this pathname if the filesystem does not change.

9.**RMD**: RMD [dir] removes an empty directory of server. Notice that [dir] must be an empty directory.

10.**MKD**：CWD [path] asks the server to make a new file directory in path.

11.**QUIT** & **ABOR**: On receiving a QUIT command, you will need to return the appropriate acknowledgement code to the client and then log the client out. The ABOR command is the same as QUIT command.

四．本次实验的亮点

1.实现了一个用户表，这样就可以灵活地加入新的用户。

2.在 server 端实现了多线程，能够处理多个 client 端的访问。

五．总结与经验

因为这次我的 client 端和 server 端都是用 C 语言实现的，所以我真的是深入到了 FTP 的底层细节。PASV 和 PORT 模式，RETR 和 STOR 指令，每个都是要一步一步地去对齐然后才能连接，每次读取的信息都要对"\n\r"做处理，很多细节都是做之前根本想不到的。可以说，这次大作业让我对 FTP 协议的底层实现有了很深的理解

与此同时呢，我也给自己挖了一个大坑。为什么这么说呢？因为我选择了用 C 语言去写 client 端！client 端的 C 代码我大概写了一千多行，当我看到舍友用 80 多行 python 代码实现的 client 端时我是崩溃的。

虽然这么说，但是这次实验中我也遇到了很多 C 语言或者 FTP 的小细节，我觉得很有意思，就写在下面和大家分享一下。

（1）所有的 port 必须要用 htons 处理啊，要不然 server 和 client 端对不上的。

（2）malloc 分配的地址是在堆上的，在开多线程时，多个线程里的同名指针其实指向的是

同一个地址，是对同一片数据区域做的操作。

（3）不要没事去关 client 端的端口，要不然 server 端发回来的数据有时会溢出到 client 端的 stdin 上。

（4）“.h 文件中声明，.c 文件中实现“ 是一个很好的习惯，要不然会有很多 warning 报错提示你有很多变量定义了未使用。

（5）C 语言中对字符串的处理真的很不合理，用'\0'合理简直毫无依据，会造成很多的 bug。如果我要去设计一门语言的话，我会选择对字符串加一个 length 的参数，就像 python 语言那样。