计算机网络实验一报告

1、实验环境

安装 **virtualbox** 搭建虚拟机，于虚拟环境中进行代码书写以及调试。

2、UDP 实验

实现 **UDPServer,UDPClient** 部分的代码，达成两端通信的需求。实现 **client** 发送 **0~50**。实现 **sever** 统计消息序号并返回“序号 内容”。

3、FTP Server 实验：

实验中使用 **Berkerley API** 实现了简单的 **Ftp** 服务器，功能、代码相关如下。

3.1、重要全局变量以及宏定义

serverSocket,clientSocket,dataSocket

分别为服务端监听 socket,接收客户端 socket，数据传输口 socket。

serverAddr,clientAddr,dataAddr;

分别为服务端地址,接收客户端地址，数据传输对应的地址。

serverAdrLen,clientAdrLen,dataAdrlen;

对应以上地址的大小。

TYPE 传输类型 buffer 读取输入的缓存区,Res 统计读取字符个数;

FILEBUF_SIZE 1024 最大传输量

3.2、重要函数

void WriteToClient(char *reply,int clientSocket);

输出到对应的客户端，打印输出内容。

void ReadFromCLient();

处理客户端指令，使得 buffer 去掉\r\n，并且打印指令内容。

void ntorn_tail(char *info);

处理系统 ls 指令得到的信息将每条信息尾部\n 转化为\r\n。

3.3、功能实现

3.3.1、Server 建立

首先，读取参数得到特定端口以及工作目录，而后创建 **Socket** 绑定并且监听，进入服务器主循环接收用户端的连接。

3.3.2、USER、PASS

为了保证只有登陆后才能使用主要功能，实现了一个循环，只有当用户输入正确的用户名(要求所说的 **anonymous** 才能登陆后使用 **FTP** 服务器的功能。

而 **PASS** 指令因为没有实际效益所以把他当做普通指令处理。

3.3.3、PORT、PASV

a、PORT 实现：

通过 **sscanf** 读取 **buffer** 得到客户端发送的 **IP** 地址，而后向客户端端口连接。

b、PASV 实现：

得到客户端的 **PASV** 请求后发送自身 **IP** 地址。（注：在取得合适端口号上，采用

初始化时令地址端口为 0，系统会自动选取合适的端口，然后使用系统函数 getsockname()来获取到这个端口号后发送，这个方法比随机的要健壮）

3.3.4、RETR、STOR

二者实现方式相似，都是一方向另一方传递数据，所以挑 RETR 进行简要说明。处理 RETR 首先要保证数据连接存在，否则就返回错误信息。另外要处理文件不存在的情况。这二者异常处理已经实现。而后向客户端发送确认信息。

后进入读取本地文件的循环，用 fileno 函数通过 filenname 句柄实现对本地文件的读取，循环读取，每次循环确认连接存在后向对方写数据，每次循环对 databuf 读取的缓存区进行清除。

3.3.5、SYST,TYPE,QUIT,ABOR

QUIT：退出最外层读取循环。

3.3.6、MKD,RMD,CWD

以上指令接收到服务器即调用系统函数（mkdir,rmdir,chdir）

3.3.7、LIST

利用 popen() 函数通过创建一个管道，调用 fork 产生一个子进程，执行一个 shell 以运行“ls -l”来开启系统 ls 进程，再使用 read 进行读取以获得得到的目录信息，注意的是得到每一列都是以'\n'结尾的，但是标准内要求以'\r\n'结尾，故编写了 ntorn_tail()函数实现了转换，然后标准传向客户端。

3.3.8、Multiple User

采用 fork 函数创建多进程。

```
pid_t p = fork();
if(p < 0){printf("fork failed");}
else if(p != 0){continue;}
```

以上代码使得主进程永远处于监听接收状态，如果接收到连接则新建一个子进程处理该连接，而主进程再回到循环首继续监听。

4、FTP Client 实验结果：

功能、代码相关如下。

4.1、重要全局变量以及宏定义

ServerSocket 服务端 socket，data_sock 用于数据传输的 socket
TYPE 传输类型 FILEBUF_SIZE 1024 最大传输量
MAX_INPUT_SIZE 254 用户端输入信息最大量

4.2、重要函数

void ReadFromServer();读取 server 端数据。
void WriteToServer(char *command,char *params);向 server 端输入信息
int GetBufCode();得到读入的 buf 里边的有效指令编号。
int getlocalip(char *outip);得到有效的本地 IP，用于 PORT。

4.3、功能实现

实现所有与服务端对应功能，并且功能的实现代码与服务端高度对应（包括 PORT 对应 PASV，STOR 对应 RETR 等），就不一一列举。

4.4、用户手册

4.4.1、连接服务器

输入服务器地址以及端口号，连接进入服务器，如果连接失败会要求重新输入。（注：NAT 模式接外网，桥接模式接局域网，外网无法测试 PORT 只能测试 PASV，局域网测试全部功能正常使用）

4.4.2、直接输入用户名和密码，登陆成功后才能继续否则重新输入账户。

4.4.3、命令使用表格

命令名称	用户操作名	使用样例	使用用途
USER	直接输入用户名	anonymous	用户名
PASS	直接输入密码	1	密码
QUIT	quit	quit	退出
ABOR	abor	abor	退出
TYPE	type	type A/type I	改变传输模式
SYST	syst	syst	获取系统
PORT	port	port	建立 port 连接
PASV	pasv	pasv	建立 pasv 连接
LIST	ls	ls	默认 port 获取列表
MKD	mkd “dirname”	mkd 123	建立文件夹
RMD	rmd “dirname”	rmd 123	删除空文件夹
CWD	cd “dirname”	cd 123	改变工作目录
RETR	getp “dirname”	getp 123	port 下载文件
RETR	geta “dirname”	geta 123	pasv 下载文件
STOR	putp “dirname”	putp 123	port 下载文件
STOR	puta “dirname”	puta 123	pasv 下载文件
PWD	pwd	pwd	获取当前目录
RNFR	rnfr “dir1” “dir2”	rnfr 1 2	重命名
DELE	dele “dirname”	dele 123	删除文件

5、Optional Codes

实现了 DELE 指令删除文件（实际上可直接删除整个文件夹），实现了 RNFR 指令重命名，实现了 PWD 指令获取当前工作目录，同时客户端操作时会即时显示当前工作路径，给命令行润色一下。

6、不足：

在虚拟机运行下，无法获得本机的外网 IP，所以只能程序去连接别人。