***/\*FTP server\*/***

#include <sys/socket.h>

#include <netinet/in.h>

#include <string.h>

#include <stdio.h>

#include <stdlib.h>

*/\*for getting file size using stat()\*/*

#include<sys/stat.h>

*/\*for sendfile()\*/*

#include<sys/sendfile.h>

*/\*for O\_RDONLY\*/*

#include<fcntl.h>

int main(int argc,char \*argv[])

{

struct sockaddr\_in server, client;

struct stat obj;

int sock1, sock2;

char buf[100], command[5], filename[20];

int k, i, size, len, c;

int filehandle;

sock1 = socket(AF\_INET, SOCK\_STREAM, 0);

if(sock1 == -1)

{

printf("Socket creation failed");

exit(1);

}

server.sin\_port = atoi(argv[1]);

server.sin\_addr.s\_addr = 0;

k = bind(sock1,(struct sockaddr\*)&server,sizeof(server));

if(k == -1)

{

printf("Binding error");

exit(1);

}

k = listen(sock1,1);

if(k == -1)

{

printf("Listen failed");

exit(1);

}

len = sizeof(client);

sock2 = accept(sock1,(struct sockaddr\*)&client, &len);

i = 1;

while(1)

{

recv(sock2, buf, 100, 0);

sscanf(buf, "%s", command);

if(!strcmp(command, "ls"))

{

system("ls >temps.txt");

i = 0;

stat("temps.txt",&obj);

size = obj.st\_size;

send(sock2, &size, sizeof(int),0);

filehandle = open("temps.txt", O\_RDONLY);

sendfile(sock2,filehandle,NULL,size);

}

else if(!strcmp(command,"get"))

{

sscanf(buf, "%s%s", filename, filename);

stat(filename, &obj);

filehandle = open(filename, O\_RDONLY);

size = obj.st\_size;

if(filehandle == -1)

size = 0;

send(sock2, &size, sizeof(int), 0);

if(size)

sendfile(sock2, filehandle, NULL, size);

}

else if(!strcmp(command, "put"))

{

int c = 0, len;

char \*f;

sscanf(buf+strlen(command), "%s", filename);

recv(sock2, &size, sizeof(int), 0);

i = 1;

while(1)

{

filehandle = open(filename, O\_CREAT | O\_EXCL | O\_WRONLY, 0666);

if(filehandle == -1)

{

sprintf(filename + strlen(filename), "%d", i);

}

else

**break**;

}

f = malloc(size);

recv(sock2, f, size, 0);

c = write(filehandle, f, size);

close(filehandle);

send(sock2, &c, sizeof(int), 0);

}

else if(!strcmp(command, "pwd"))

{

system("pwd>temp.txt");

i = 0;

FILE\*f = fopen("temp.txt","r");

while(!feof(f))

buf[i++] = fgetc(f);

buf[i-1] = '**\0**';

fclose(f);

send(sock2, buf, 100, 0);

}

else if(!strcmp(command, "cd"))

{

if(chdir(buf+3) == 0)

c = 1;

else

c = 0;

send(sock2, &c, sizeof(int), 0);

}

else if(!strcmp(command, "bye") || !strcmp(command, "quit"))

{

printf("FTP server quitting..**\n**");

i = 1;

send(sock2, &i, sizeof(int), 0);

exit(0);

}

}

return 0;

}

***/\*FTP Client\*/***

#include <sys/socket.h>

#include <netinet/in.h>

#include <string.h>

#include <stdio.h>

#include <stdlib.h>

*/\*for getting file size using stat()\*/*

#include<sys/stat.h>

*/\*for sendfile()\*/*

#include<sys/sendfile.h>

*/\*for O\_RDONLY\*/*

#include<fcntl.h>

int main(int argc,char \*argv[])

{

struct sockaddr\_in server;

struct stat obj;

int sock;

int choice;

char buf[100], command[5], filename[20], \*f;

int k, size, status;

int filehandle;

sock = socket(AF\_INET, SOCK\_STREAM, 0);

if(sock == -1)

{

printf("socket creation failed");

exit(1);

}

server.sin\_family = AF\_INET;

server.sin\_port = atoi(argv[1]);

server.sin\_addr.s\_addr = 0;

k = connect(sock,(struct sockaddr\*)&server, sizeof(server));

if(k == -1)

{

printf("Connect Error");

exit(1);

}

int i = 1;

while(1)

{

printf("Enter a choice:**\n**1- get**\n**2- put**\n**3- pwd**\n**4- ls**\n**5- cd**\n**6- quit**\n**");

scanf("%d", &choice);

switch(choice)

{

case 1:

printf("Enter filename to get: ");

scanf("%s", filename);

strcpy(buf, "get ");

strcat(buf, filename);

send(sock, buf, 100, 0);

recv(sock, &size, sizeof(int), 0);

if(!size)

{

printf("No such file on the remote directory**\n\n**");

**break**;

}

f = malloc(size);

recv(sock, f, size, 0);

while(1)

{

filehandle = open(filename, O\_CREAT | O\_EXCL | O\_WRONLY, 0666);

if(filehandle == -1)

{

sprintf(filename + strlen(filename), "%d", i);*//needed only if same directory is used for both server and client*

}

else **break**;

}

write(filehandle, f, size, 0);

close(filehandle);

strcpy(buf, "cat ");

strcat(buf, filename);

system(buf);

**break**;

case 2:

printf("Enter filename to put to server: ");

scanf("%s", filename);

filehandle = open(filename, O\_RDONLY);

if(filehandle == -1)

{

printf("No such file on the local directory**\n\n**");

**break**;

}

strcpy(buf, "put ");

strcat(buf, filename);

send(sock, buf, 100, 0);

stat(filename, &obj);

size = obj.st\_size;

send(sock, &size, sizeof(int), 0);

sendfile(sock, filehandle, NULL, size);

recv(sock, &status, sizeof(int), 0);

if(status)

printf("File stored successfully**\n**");

else

printf("File failed to be stored to remote machine**\n**");

**break**;

case 3:

strcpy(buf, "pwd");

send(sock, buf, 100, 0);

recv(sock, buf, 100, 0);

printf("The path of the remote directory is: %s**\n**", buf);

**break**;

case 4:

strcpy(buf, "ls");

send(sock, buf, 100, 0);

recv(sock, &size, sizeof(int), 0);

f = malloc(size);

recv(sock, f, size, 0);

filehandle = creat("temp.txt", O\_WRONLY);

write(filehandle, f, size, 0);

close(filehandle);

printf("The remote directory listing is as follows:**\n**");

system("cat temp.txt");

**break**;

case 5:

strcpy(buf, "cd ");

printf("Enter the path to change the remote directory: ");

scanf("%s", buf + 3);

send(sock, buf, 100, 0);

recv(sock, &status, sizeof(int), 0);

if(status)

printf("Remote directory successfully changed**\n**");

else

printf("Remote directory failed to change**\n**");

**break**;

case 6:

strcpy(buf, "quit");

send(sock, buf, 100, 0);

recv(sock, &status, 100, 0);

if(status)

{

printf("Server closed**\n**Quitting..**\n**");

exit(0);

}

printf("Server failed to close connection**\n**");

}

}

}