/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

NETSURF ORGANIZATION

FTP server

ROLL NO. SYA03 , SYA05 , SYA10 , SYA15 , SYA20

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<sys/socket.h>

#include<netinet/in.h>

#include<string.h>

#include<stdio.h>

#include<stdlib.h>

#include<sys/types.h>

#include<netinet/ip.h>

#include<arpa/inet.h>

int main()

{

system("clear");

int sockfd, newfd;

sockfd = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

struct sockaddr\_in server, client;

if(sockfd==-1)

{

perror("\*\*\*\*\*socket creation is failed\*\*\*\*\*\n");

exit(0);

}

printf("\n\*\*\*\*\*Server program waiting for Client .... \n\n");

server.sin\_family = AF\_INET;

/\* Set port number, using htons function to use proper byte order \*/

server.sin\_port = htons(7891);

/\* Set IP address to localhost \*/

server.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

/\* Set all bits of the padding field to 0 \*/

//memset(serverAddr.sin\_zero, '\0', sizeof serverAddr.sin\_zero);

int b;

b=bind(sockfd,(struct sockaddr\*)&server,sizeof (struct sockaddr));

if(b==-1)

{

perror("Bind failuer\n");

exit(0);

}

listen(sockfd,5);

int size=sizeof (struct sockaddr);

newfd=accept(sockfd, (struct sockaddr\*) &client, &size);

if(newfd==-1)

{

perror("\naccept failuer\n");

exit(0);

}

printf("\nclient request accepted\n");

system("ls>a.txt");

char str [50];

char \*msg=" ", \*\*msg1=" ";

FILE \*fp=fopen("a.txt","r");

int k, p, l1, l2;

msg=(char\*) malloc (1);

while (fgets(str, 50, fp)!=NULL)

{

l1=strlen(str);

l2=strlen(msg);

msg=(char\*) realloc (msg, l1+l2);

strcat(msg, str);

}

fclose(fp);

int s=strlen(msg);

send(newfd, &s, 4, 0);

send(newfd, msg, strlen(msg), 0);

char fname[20];

int f1=recv (newfd, fname, sizeof (fname),0);

fname[f1]='\0';

printf("\nfile requist recieived from client is %s \n",fname);

fp=fopen(fname,"r");

msg1=(char\*) malloc (1);

while (fgets(str, 50, fp)!= NULL)

{

l1=strlen(str);

l2=strlen(msg1);

msg1=(char\*) realloc (msg1, l1+l2);

strcat(msg1, str);

}

int p3 = strlen(msg1);

send(newfd, &p3, 4, 0);

send(newfd, msg1, strlen(msg1), 0);

printf("\n\nfile successful transferer from server to client\n\n");

close(sockfd);

close(newfd);

}