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

NETSURF ORGANIZATION

FTP client

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

#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("\nReady for sending...");

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");

int c=connect(sockfd,(struct sockaddr\*) &server, sizeof (server));

if (c==-1)

{

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

exit(0);

}

printf("\n\*\*\*\*\*connected to server\*\*\*\*\*\n");

int l,k;

recv(sockfd, &l, 4, 0);

char msg[l];

char fname[20];

k=recv(sockfd, msg, sizeof(msg), 0);

msg[k]='\0';

printf("%s\n",msg);

printf("\nEnter file name to request from server:");

scanf("%s",fname);

send(sockfd, fname, strlen(fname), 0);

printf("\nfile name successful send\n");

int p4, k1;

recv(sockfd, &p4, 4, 0);

char msg2[p4];

k1=recv(sockfd, msg2, sizeof(msg2), 0);

msg2[k1]='\0';

printf("\nfile receive from server is:\n%s",msg2);

printf("\nsize of file is:%d",p4);

char fname1[20];

printf("\nEnter new file name :\n");

scanf("%s",fname1);

FILE \*fp=fopen(fname1,"w");

fwrite(msg2, 1, &p4, fp);

fclose(fp);

printf("\n\nfile successfuly transfer & create in client\n\n");

close(sockfd);

}