**CN EXP – 8**

Name Madhurya Mozumder

Reg no RA1911028010036

Batch I2

IMPLEMENTATION OF FILE TRANSFER PROTOCOL

GIVEN REQUIREMENTS:

There are two hosts, Client and Server. The Client sends the name of the file it needs from the Server and the Server sends the contents of the file to the Client, where it is stored in a file.

TECHNICAL OBJECTIVE:

To implement FTP application, where the Client on establishing a connection with the Server sends the name of the file it wishes to access remotely. The Server then sends the contents of the file to the Client, where it is stored.

Server.c

#include <sys/types.h>

#include <sys/socket.h>

#include <sys/stat.h>

#include <arpa/inet.h>

#include <netinet/in.h>

#include <netdb.h>

#include <unistd.h>

#include <stdio.h>

#include <string.h>

int main(int argc, char \*argv[])

{

        int sd, ad, size;

        struct sockaddr\_in servaddr, cliaddr;

            socklen\_t clilen;

            clilen = sizeof(cliaddr);

            struct stat x;

             char buff[100], file[10000];

             FILE \*fp;

         bzero(&servaddr, sizeof(servaddr));

         servaddr.sin\_family = AF\_INET;

         servaddr.sin\_addr.s\_addr = htonl(INADDR\_ANY);

          servaddr.sin\_port = htons(1111);

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

         bind(sd, (struct sockaddr \*)&servaddr, sizeof(servaddr));

        listen(sd, 5);

         printf("%s\n", "Server Is Running....");

        ad = accept(sd, (struct sockaddr \*)&cliaddr, &clilen);

        while (1)

         {

             bzero(buff, sizeof(buff));

              bzero(file, sizeof(file));

              recv(ad, buff, sizeof(buff), 0);

              fp = fopen(buff, "r");

              stat(buff, &x);

              size = x.st\_size;

              fread(file, sizeof(file), 1, fp);

             printf("Contents of File:\n%s", file);

            printf("File Sent\n");

            send(ad, file, sizeof(file), 0);

        }

        return 0;

}

Client.c

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include <netdb.h>

#include <stdio.h>

#include <unistd.h>

#include <string.h>

int main(int argc, char \*argv[])

{

        int sd, cd;

        struct sockaddr\_in servaddr, cliaddr;

        socklen\_t clilen;

        char buff[100], file[10000];

        struct hostent \*h;

        h = gethostbyname(argv[1]);

        bzero(&servaddr, sizeof(servaddr));

        servaddr.sin\_family = h->h\_addrtype;

        memcpy((char \*)&servaddr.sin\_addr.s\_addr, h->h\_addr\_list[0], h->h\_length);

        servaddr.sin\_port = htons(1111);

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

        cd = connect(sd, (struct sockaddr \*)&servaddr, sizeof(servaddr));

        while (1)

        {

                printf("%s\n", "Enter the File Name :");

                scanf("%s", buff);

                send(sd, buff, strlen(buff) + 1, 0);

                printf("%s\n", "File Output :");

                recv(sd, file, sizeof(file), 0);

                printf("%s", file);

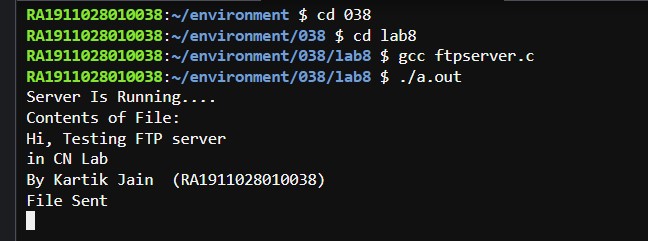
        }

    return 0;

}

OUTPUT SCRRENSHOT: -

Server:



Client:

