

Exp No: 8
18/09/21

Raheem Bharat Mishra
RA1911028010107
J1 Section

IMPLEMENTATION OF FILE TRANSFER PROTOCOL

Given Requirement

AIM: To implement the 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.

METHODOLOGY:

Server:-

- Include necessary header files.
- Create a socket using `family AF_INET`
Type as `SOCK_STREAM`.
- Initialize server address to `0.0.0.0` During the `listen` function.
- Assign the `sin_family` to `AF_INET`, `sin_addr` to `INADDR_ANY`, `sin_port` to dynamically assigned port number.
- Bind the local host address to the socket for connection request from the client.
- Accept connection request from the client using `accept` function.
- Within an infinite loop, receive filename from the client.

- Open the file, read the file contents to a buffer and send the buffer to the client.

Client.

- Include the necessary header files.
- Create a socket using socket function with family AF_INET type as SOCK_STREAM.
- Initialize the server address to 0 using the bzero function.
- Assign the sin_family to AF_INET.
- Get the server IP Address and port number from the console.
- Using get host by name function assign it to a hostent structure and assign it to sin_addr of the server address structure.
- Within an infinite loop, send the name of the file to be viewed to the server.
- Receive the file contents, store it in the file and print it in the console.

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, nize;
    struct sockaddr_in servaddr, cliaddr;
    socklen_t clien;
    clien = sizeof(cliaddr);
    struct stat s;
    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(9053);
    sd = socket(AF_INET, SOCK_STREAM, 0);
    bind(sd, (struct sockaddr*)&servaddr, sizeof(servaddr));
    listen(sd, 5);
    printf("%s\n", "Server is running..");
    add = accept(sd, (struct sockaddr*)&cliaddr, &clien);
    while (1) {
        bzero(buff, sizeof(buff));
        bzero(file, sizeof(file));
        read(sd, buff, sizeof(buff), 0);
        fp = fopen(buff, "r");
    }
}

```



```

stat (buff, &n);
size = n.st_size;
fread (file, sizeof(file), 1, fp);
send (ad, file, sizeof(file), 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>

int main (int argc, char *argv[])
{
    int sd, cd;

    struct sockaddr_in servaddr, cliaddr;
    socklen_t clen;
    char buff[100], file[20000];
    struct hostent *h;
    h = gethostbyname (argv[1]);
    bzero (&servaddr, sizeof(servaddr));
    servaddr.sin_family = h->h_addrtype;
    memcpy (&servaddr.sin_addr.s_addr, h->h_length);
}

```

```
servaddr.sin_port = htons(9053);
```

```
sd = socket(AF_INET, SOCK_STREAM, 0);
```

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

```
while(1)
```

```
{
```

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

```
recv(sd, buff, 1024, 0);
```

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

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

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

```
printf("%s", file);
```

```
}
```

```
return 0;
```

```
}
```

Result: The FTP protocol is implement with socket programming



