26/10/2021 **CN LAB 8** 2019103573

## FILE TRANSFER PROTOCOL (FTP)

## SERVER.C

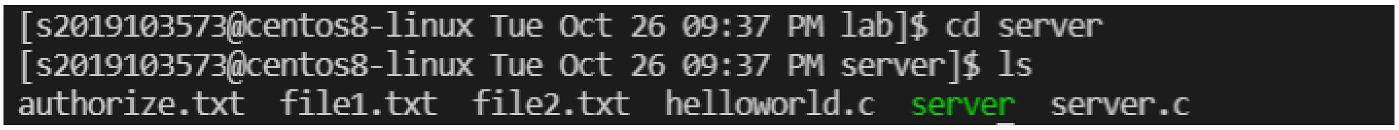
```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <arpa/inet.h>

#define CONTROLPORT 2578
#define SERVERPORT 2488

int listeningSocketConnection(long port)
{
    int socketfd;
    struct sockaddr_in server_addr;
    socketfd = socket(AF_INET, SOCK_STREAM, 0);
    if (socketfd < 0)
    {
        fprintf(stderr, "ERROR IN SOCKET CREATION");
        return -1;
    }
    server_addr.sin_family = AF_INET;
    server_addr.sin_port = htons(port);
    inet_pton(AF_INET, "127.0.0.1", &server_addr.sin_addr);
    if (bind(socketfd, (struct sockaddr *)&server_addr, sizeof(server_addr)) < 0)
    {
        fprintf(stderr, "ERROR IN BIND CREATION");
        return -1;
    }
    if (listen(socketfd, 5) < 0)
    {
        fprintf(stdout, "ERROR IN LISTEN CREATION");
        return -1;
    }
    fprintf(stdout, "LISTENING AT %s : %d \n", inet_ntoa(server_addr.sin_addr),
ntohs(server_addr.sin_port));
    return socketfd;
}

void loading()
{
    for (int k = 0; k < 10000000; k++)
    {
        int j = k;
    }
}

void file_transfer(int client)
{
    int socketfd, clientfd, length = 0;
    struct sockaddr_in server_addr, client_addr;
    char buffer[1024], filename[1024];
    const char *port = "2488";
    socketfd = listeningSocketConnection(SERVERPORT);
    if (socketfd < 0)
        return;
    send(client, port, strlen(port) + 1, 0);
    clientfd = accept(socketfd, (struct sockaddr *)&client_addr, &length);
    if (clientfd < 0)
    {
        fprintf(stderr, "Error in accepting connection.\n");
        close(socketfd);
        return;
    }
    fprintf(stdout, "New file trasfer connection established.\n");
    while (1)
    {
        recv(clientfd, filename, sizeof(filename), 0);
        if (strncmp(filename, "exit", strlen("exit")) == 0)
        {
            fprintf(stdout, "Client exiting file_transfer.\n");
            break;
        }

        fprintf(stdout, "\n[-] Requested file : %s\n", filename);
        int fd = open(filename, O_RDONLY);
        if (fd < 0)
        {
            send(clientfd, "404", strlen("404") + 1, 0);
            fprintf(stdout, "Requested file not found.\n");
            continue;
        }

        send(clientfd, "FOUND", strlen("FOUND") + 1, 0);
        fprintf(stdout, "Requested file found.\n");
        continue;
    }

    close(clientfd);
    close(socketfd);
    fprintf(stdout, "Closing the file-transfer connection.\n\n");
    return;
}

void authorize_and_handle(int clientfd)
{
    char buffer[1024], linebuffer[1024];
    char username[1024], password[1024];
    char *user, *pass;
    int n, i, j, flag = 0;

AUTHORIZE:
    n = recv(clientfd, buffer, sizeof(buffer), 0);
    if (strncmp(buffer, "auth$", sizeof("auth$")) == 0)
    {
        send(clientfd, "INVALID REQUEST", sizeof("INVALID REQUEST") + 1, 0);
        return;
    }

    i = 5;
    j = 0;
    while (buffer[i] != '$' && buffer[i] != '\n' && buffer[i] != '\0')
    {
        username[j] = buffer[i];
        j++;
        i++;
    }

    username[j] = '\0';
    i++;
    j = 0;
    while (buffer[i] != '$' && buffer[i] != '\n' && buffer[i] != '\0')
    {
        password[j] = buffer[i];
        j++;
        i++;
    }

    password[j] = '\0';
    printf("\n---------------------------------------------------\n");
```

```c
fprintf(stdout," [-] REQUEST FOR CLIENT's USERNAME : %s\n\tCLIENT's PASSWORD %s\n",username,password);
printf("\nCHECKING");
for(int k = 0 ; k < 8 ; k++){
        loading();
        printf(".");
}

printf("\n");
FILE *fd = fopen("authorize.txt","r");

if(fd < 0) {
        fprintf(stderr, "Error in opening auth file.\n");
        send(clientfd, "ERROR", strlen("ERROR") + 1, 0);
        return;
}

while(fgets(linebuffer,sizeof(linebuffer),fd)){
        user = strtok(linebuffer, "$");
        pass = strtok(NULL, "\n");
        if (strcmp(username, user) == 0 && strcmp(password, pass) == 0)
        {
            //send(clientfd, "LOGIN SUCCESSFULLY", strlen("LOGIN SUCCESSFULLY") + 1, 0);
            file_transfer(clientfd);
            flag = 1;
            break;
        }
}
if(flag == 0) {
        fprintf(stdout, "Invalid Access Denied.\n");
        send(clientfd, "INVALID_CRED", strlen("INVALID_CRED") + 1, 0);
        goto AUTHORIZE;
}
return;
}

int main()
{
    int socketfd, clientfd, length = 0;
    struct sockaddr_in client_addr;
    socketfd = listeningSocketConnection(CONTROLPORT);

    if (socketfd < 0)
    {
        return -1;
    }
    while (1)
    {
        clientfd = accept(socketfd, (struct sockaddr *)&client_addr, &length);
        if (clientfd < 0)
        {
            fprintf(stderr, "ERROR IN ACCEPTING CONNECTION\n");
            continue;
        }

        fprintf(stdout, "NEW CONTROL CONNECTION ESTABLISHED\n");
        authorize_and_handle(clientfd);
        close(clientfd);
    }
    close(socketfd);
    return 0;
}
```

## CLIENT.C

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/socket.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#define SERVERPORT 2578
int getConnectedSocket(long port)
{
    int socketfd = 0, n = 0;
    struct sockaddr_in server_addr;
    socketfd = socket(AF_INET, SOCK_STREAM, 0);
    if (socketfd < 0)
    {
        fprintf(stderr, "Error in socket creation.\n");
        return -1;
    }
    server_addr.sin_family = AF_INET;
    server_addr.sin_port = htons(port);
    inet_pton(AF_INET, "127.0.0.1", &server_addr.sin_addr);
    if (connect(socketfd, (struct sockaddr *)&server_addr, sizeof(server_addr)) < 0)
    {
        fprintf(stderr, "Error in connection.\n");
        return -1;
    }
fprintf(stdout,"Connection established with %s :
%d\n",inet_ntoa(server_addr.sin_addr),ntohs(server_addr.sin_port));
return socketfd;
}
void file_transfer(int file_port)
{
    int socketfd = 0;
    char filename[1024], buffer[1024];
    filename[0] = '\0';
    char cnfm;
    socketfd = getConnectedSocket(file_port);
    if (socketfd < 0)
        return;
    fprintf(stdout, "Enter exit to close the connection.\n");
    while (1)
    {
        fprintf(stdout, "\n---------------------------------------------\n");
        fprintf(stdout, "[-] Enter filename : ");
        scanf("%s", filename);
        if (strncmp(filename, "exit", strlen("exit")) == 0)
        {
            send(socketfd, "exit", strlen("exit") + 1, 0);
            break;
        }
        send(socketfd, filename, strlen(filename) + 1, 0);
        recv(socketfd, buffer, sizeof(buffer), 0);
        if (strncmp(buffer, "404", strlen("404")) == 0)
        {
            fprintf(stdout, "File not found. Try again.\n");
            continue;
        }
        else if (strncmp(buffer, "FOUND", strlen("FOUND")) == 0)
        {
            fprintf(stdout, "File found.\n");
        }
        else
        {
            fprintf(stdout, "Unexpected Error.\n");
            break;
        }
    }
    close(socketfd);
    return;
}

void loading()
{
    for (int k = 0; k < 10000000; k++)
    {
        int j = k;
    }
}

int main()
{
    int socketfd;
    struct sockaddr_in client_addr;
    char buffer[1024], username[1024], password[1024];
    socketfd = getConnectedSocket(SERVERPORT);
    if (socketfd < 0)
    {
        return -1;
    }

REQ:
    buffer[0] = '\0';
    fprintf(stdout, "\n----------------------------------------\n");
    fprintf(stdout, "\n [-] Enter username : ");
    fscanf(stdin, "%s", username);
    fprintf(stdout, "\n [-] Enter password : ");
    fscanf(stdin, "%s", password);
    strcat(buffer, "auth$");
    strcat(buffer, username);
    strcat(buffer, "$");
    strcat(buffer, password);
    strcat(buffer, "$");
    printf("\nLOADING");

    for (int k = 0; k < 8; k++)
    {
        loading();
        printf(".");
    }

    printf("\n");
    send(socketfd, buffer, strlen(buffer) + 1, 0);
    recv(socketfd, buffer, sizeof(buffer), 0);

    if (strncmp(buffer, "INVALID_CRED", strlen("INVALID_CRED")) == 0)
    {
        fprintf(stdout, "INVALID..TRY AGAIN\n");
        goto REQ;
    }
    if (strncmp(buffer, "ERROR", strlen("ERROR")) == 0)
    {
        fprintf(stdout, "Couldn't authorize credentials now. Try Again.\n");
        return -1;
    }

    // recv(socketfd,buffer,sizeof(buffer),0);
    //fprintf(stdout,"\nCONNECTED TO SERVER PORT : %s\n",buffer);

    int file_port = atoi(buffer);
    file_transfer(file_port);
    close(socketfd);
    return 0;
}
```

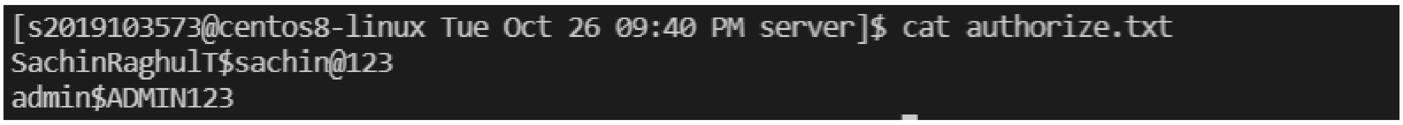
## SERVER DIRECTORY



```
[s2019103573@centos8-linux Tue Oct 26 09:37 PM lab]$ cd server
[s2019103573@centos8-linux Tue Oct 26 09:37 PM server]$ ls
authorize.txt  file1.txt  file2.txt  helloworld.c  server  server.c
```

## TEXT FILE :-

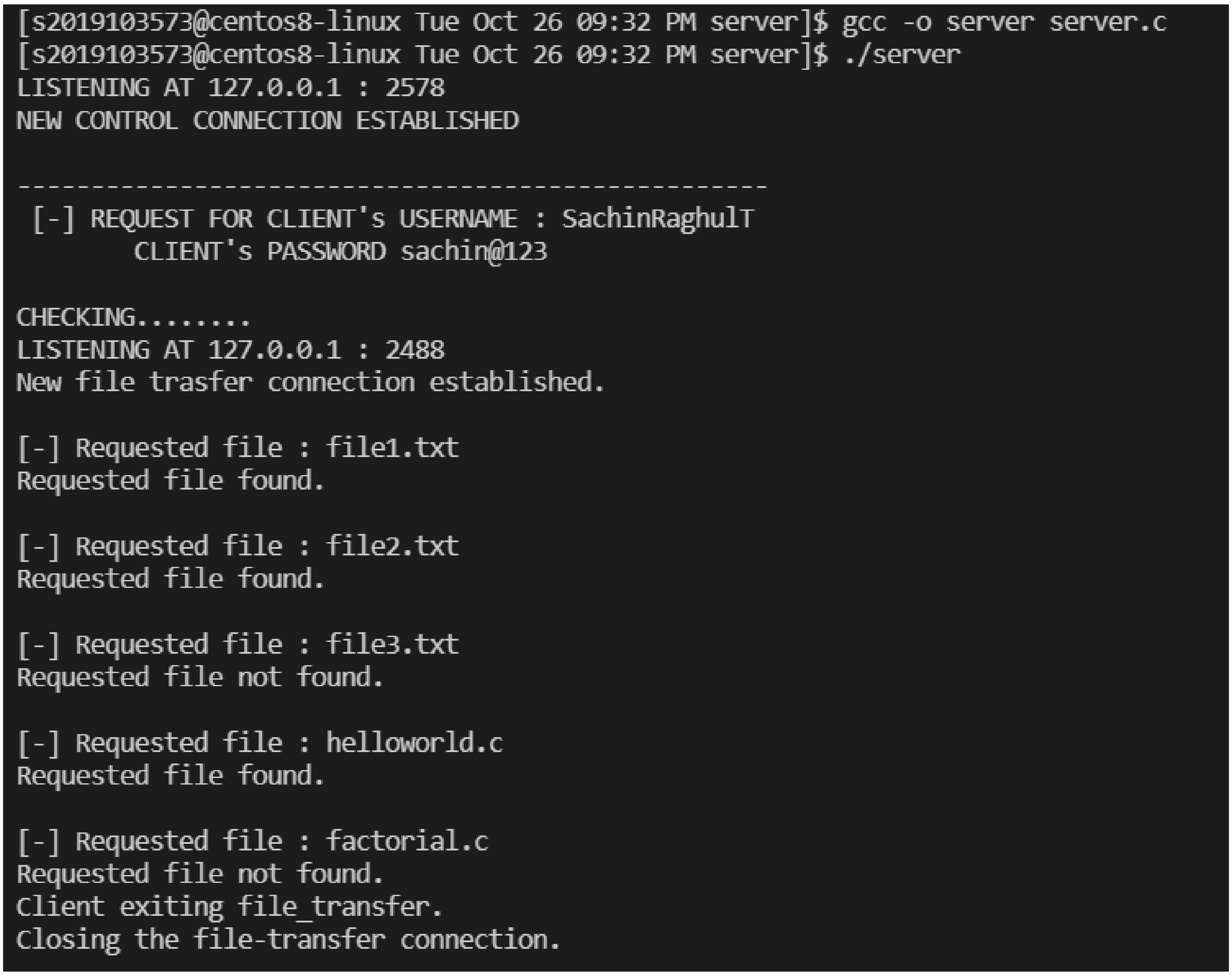
**authorize.txt**



```
[s2019103573@centos8-linux Tue Oct 26 09:40 PM server]$ cat authorize.txt
SachinRaghulT$sachin@123
admin$ADMIN123
```

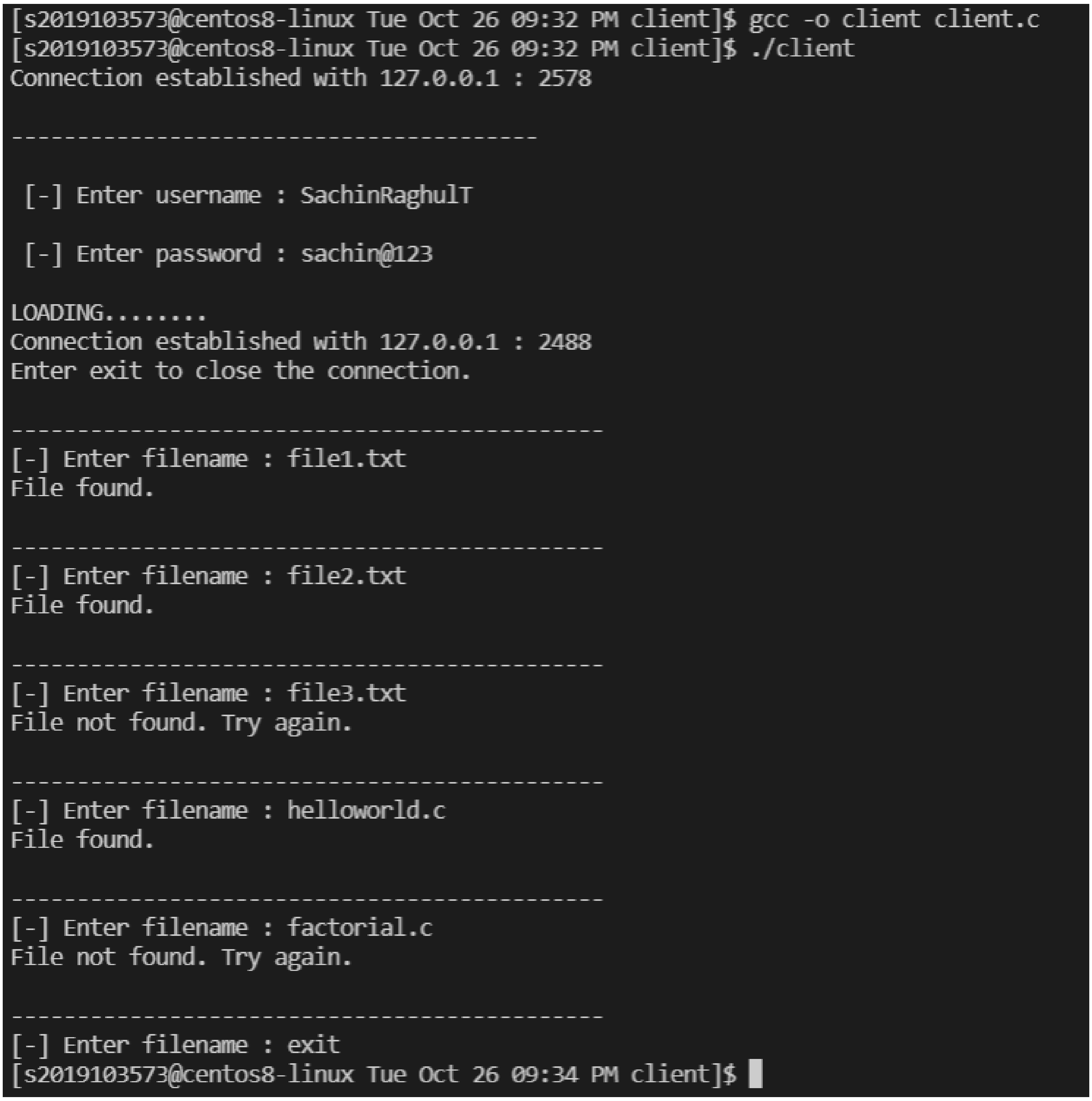
## OUTPUT :-

**SERVER OUTPUT**



```
[s2019103573@centos8-linux Tue Oct 26 09:32 PM server]$ gcc -o server server.c
[s2019103573@centos8-linux Tue Oct 26 09:32 PM server]$ ./server
LISTENING AT 127.0.0.1 : 2578
NEW CONTROL CONNECTION ESTABLISHED

---------------------------------------------------
 [-] REQUEST FOR CLIENT's USERNAME : SachinRaghulT
        CLIENT's PASSWORD sachin@123

CHECKING........
LISTENING AT 127.0.0.1 : 2488
New file trasfer connection established.

[-] Requested file : file1.txt
Requested file found.

[-] Requested file : file2.txt
Requested file found.

[-] Requested file : file3.txt
Requested file not found.

[-] Requested file : helloworld.c
Requested file found.

[-] Requested file : factorial.c
Requested file not found.
Client exiting file_transfer.
Closing the file-transfer connection.
```

**CLIENT OUTPUT**



```
[s2019103573@centos8-linux Tue Oct 26 09:32 PM client]$ gcc -o client client.c
[s2019103573@centos8-linux Tue Oct 26 09:32 PM client]$ ./client
Connection established with 127.0.0.1 : 2578

----------------------------------------


 [-] Enter username : SachinRaghulT

 [-] Enter password : sachin@123

LOADING........
Connection established with 127.0.0.1 : 2488
Enter exit to close the connection.

---------------------------------------------
[-] Enter filename : file1.txt
File found.

---------------------------------------------
[-] Enter filename : file2.txt
File found.

---------------------------------------------
[-] Enter filename : file3.txt
File not found. Try again.

---------------------------------------------
[-] Enter filename : helloworld.c
File found.

---------------------------------------------
[-] Enter filename : factorial.c
File not found. Try again.

---------------------------------------------
[-] Enter filename : exit
[s2019103573@centos8-linux Tue Oct 26 09:34 PM client]$ 
```