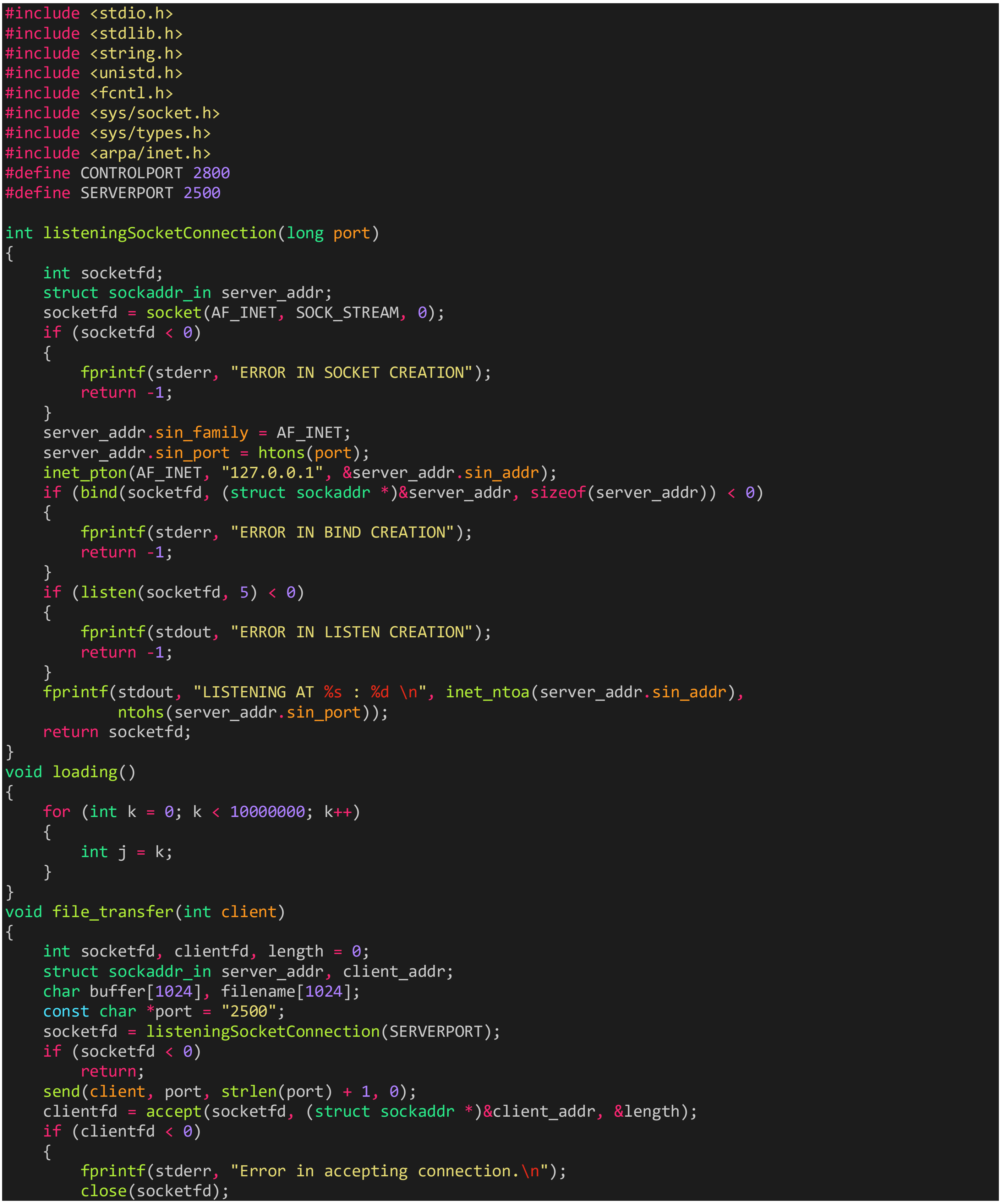
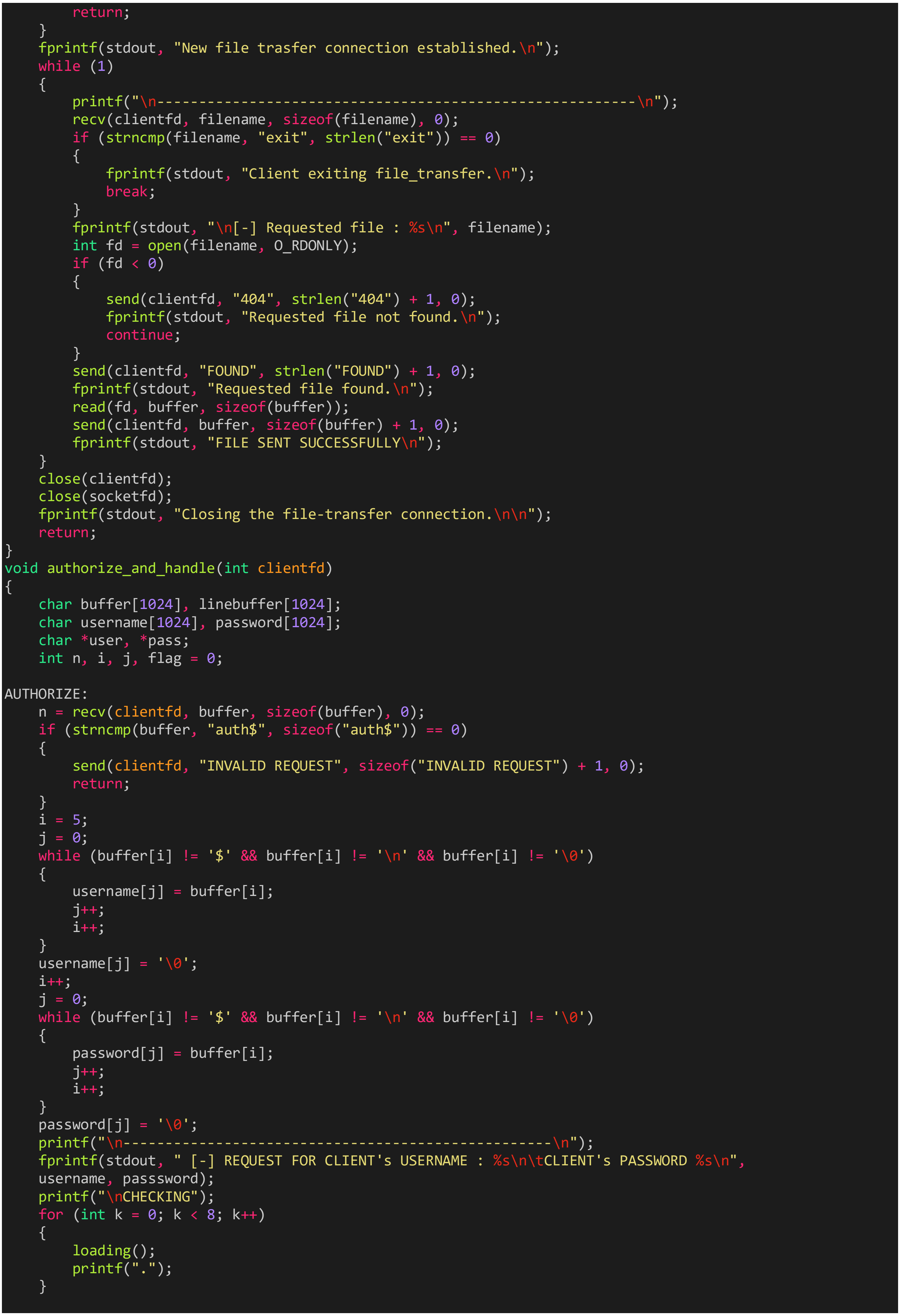
26/10/2021 **CN SPOT 8** 2019103573

FILE TRANSFER PROTOCOL (FTP)

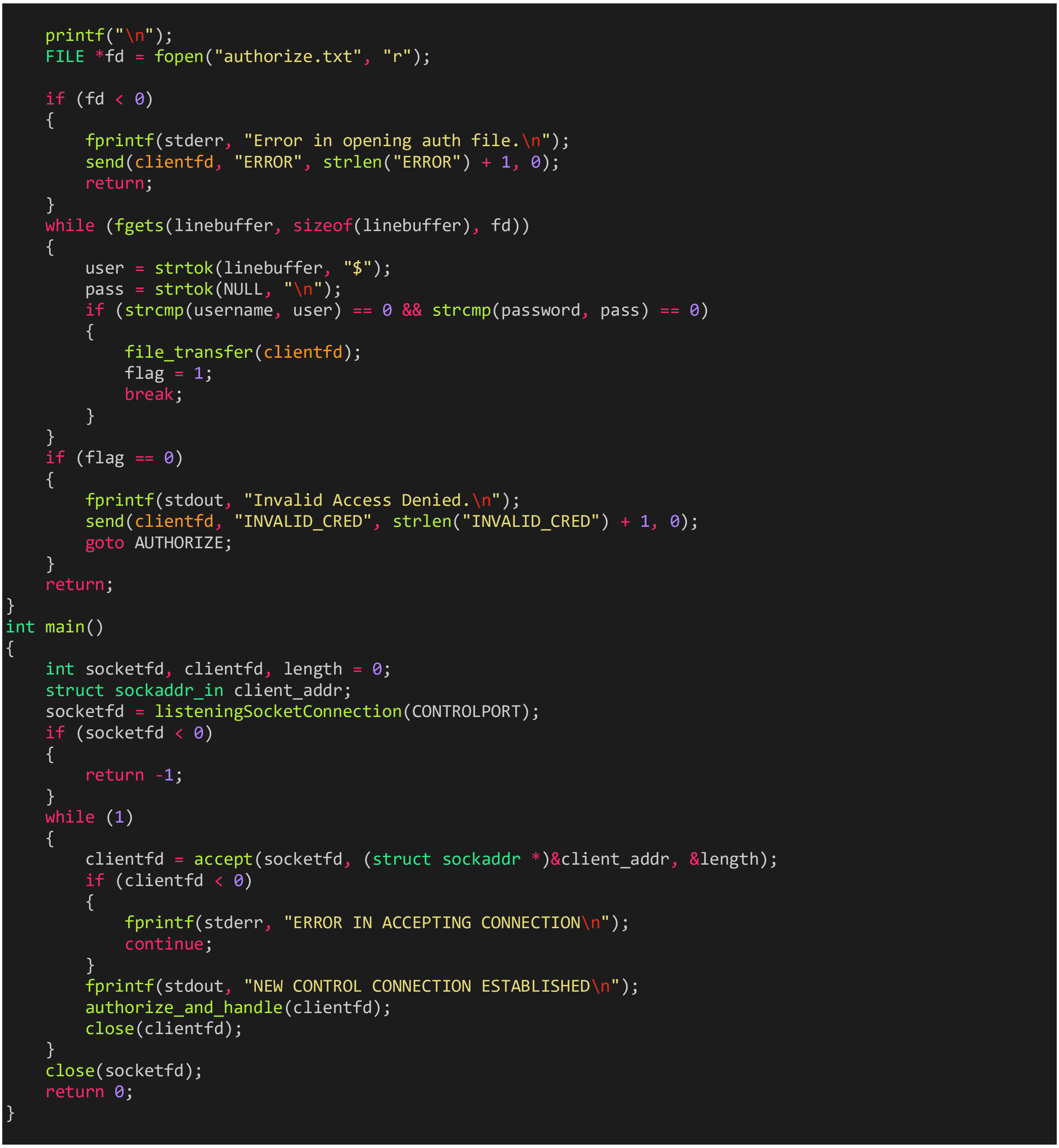
SERVER.C



```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 2800
#define SERVERPORT 2500

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 = "2500";
    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);
```



```c
        return;
    }
    fprintf(stdout, "New file trasfer connection established.\n");
    while (1)
    {
        printf("\n------------------------------------------------------------\n");
        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");
        read(fd, buffer, sizeof(buffer));
        send(clientfd, buffer, sizeof(buffer) + 1, 0);
        fprintf(stdout, "FILE SENT SUCCESSFULLY\n");
    }
    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");
    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(".");
    }
```



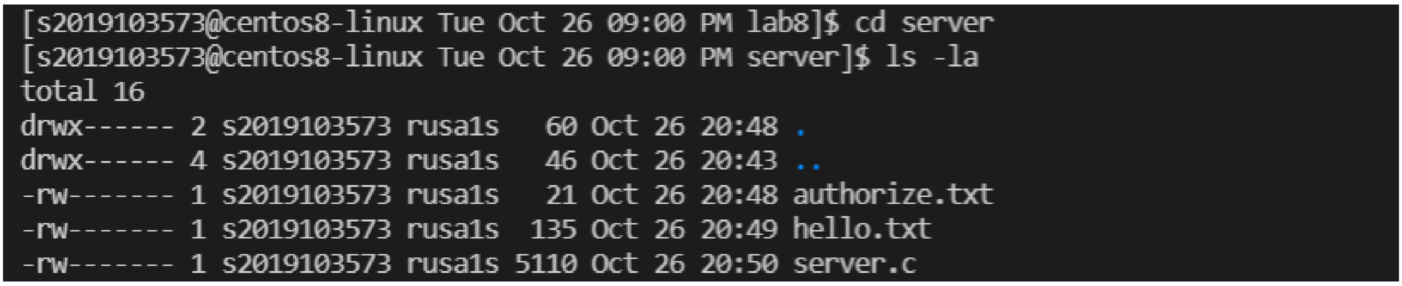
```c
    printf("\n");
    FILE *fd = fopen("authorize.txt", "r");

    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)
        {
            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 2800

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;
        }
        recv(socketfd, buffer, sizeof(buffer), 0);
        int fd = open(filename, O_CREAT | O_WRONLY);
        write(fd, buffer, strlen(buffer) + 1);
        close(fd);
        fprintf(stdout, "FILE RECEIVED AND STORED LOCALLY.\n");
        printf("\n________________\n");
        printf("\n%s\n", filename);
        printf("\n________________\n");
        printf("\n%s\n", buffer);
    }
    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;
    }

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

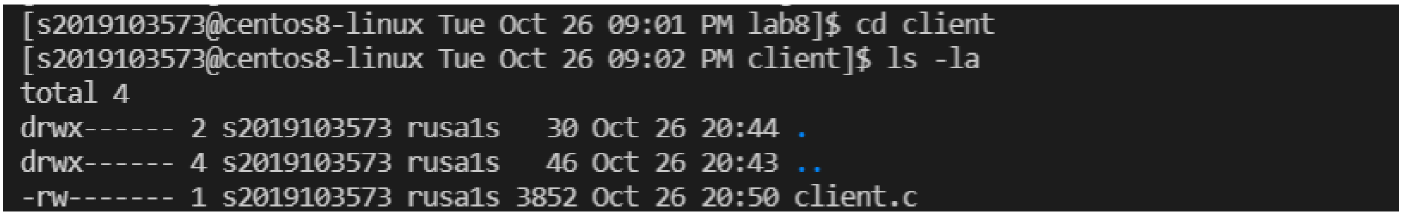
BEFORE EXECUTION :-

SERVER DIRECTORY



```
[s2019103573@centos8-linux Tue Oct 26 09:00 PM lab8]$ cd server
[s2019103573@centos8-linux Tue Oct 26 09:00 PM server]$ ls -la
total 16
drwx------ 2 s2019103573 rusa1s   60 Oct 26 20:48 .
drwx------ 4 s2019103573 rusa1s   46 Oct 26 20:43 ..
-rw------- 1 s2019103573 rusa1s   21 Oct 26 20:48 authorize.txt
-rw------- 1 s2019103573 rusa1s  135 Oct 26 20:49 hello.txt
-rw------- 1 s2019103573 rusa1s 5110 Oct 26 20:50 server.c
```

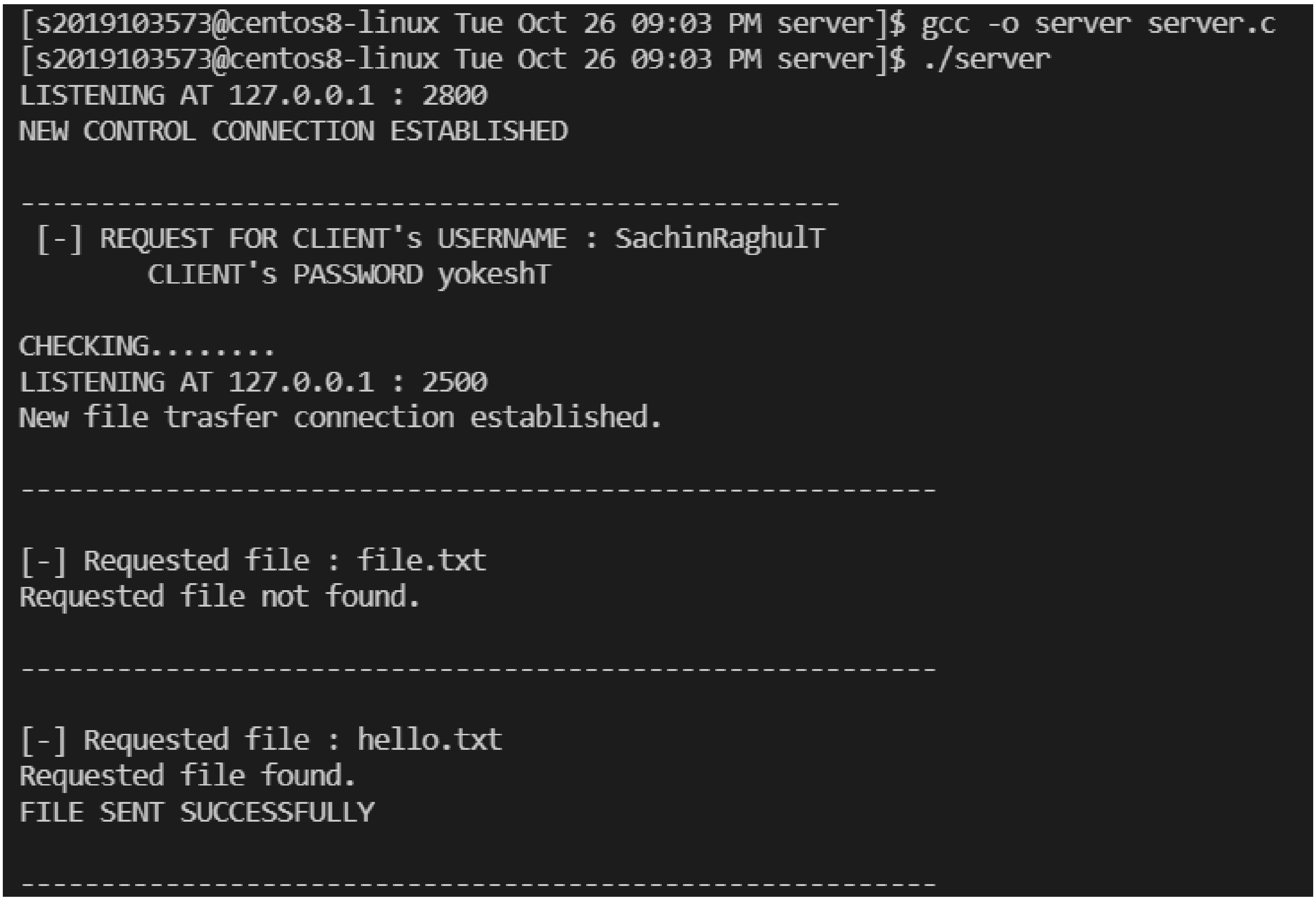
CLIENT DIRECTORY



```
[s2019103573@centos8-linux Tue Oct 26 09:01 PM lab8]$ cd client
[s2019103573@centos8-linux Tue Oct 26 09:02 PM client]$ ls -la
total 4
drwx------ 2 s2019103573 rusa1s   30 Oct 26 20:44 .
drwx------ 4 s2019103573 rusa1s   46 Oct 26 20:43 ..
-rw------- 1 s2019103573 rusa1s 3852 Oct 26 20:50 client.c
```

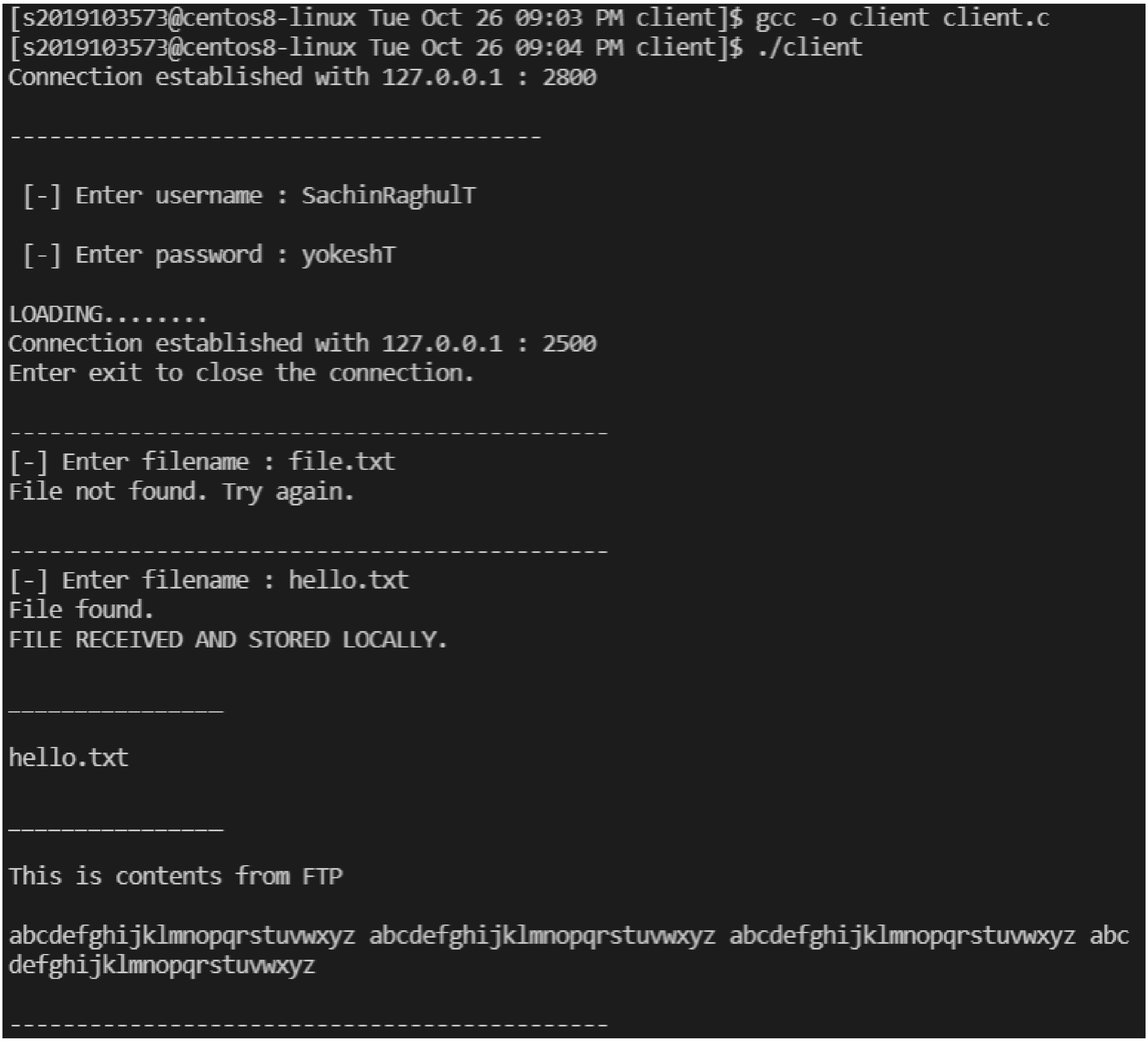
OUPUT :-

SERVER OUPTUT



```
[s2019103573@centos8-linux Tue Oct 26 09:03 PM server]$ gcc -o server server.c
[s2019103573@centos8-linux Tue Oct 26 09:03 PM server]$ ./server
LISTENING AT 127.0.0.1 : 2800
NEW CONTROL CONNECTION ESTABLISHED

----------------------------------------------------
 [-] REQUEST FOR CLIENT's USERNAME : SachinRaghulT
        CLIENT's PASSWORD yokeshT

CHECKING........
LISTENING AT 127.0.0.1 : 2500
New file trasfer connection established.

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

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

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

[-] Requested file : hello.txt
Requested file found.
FILE SENT SUCCESSFULLY

----------------------------------------------------------
```

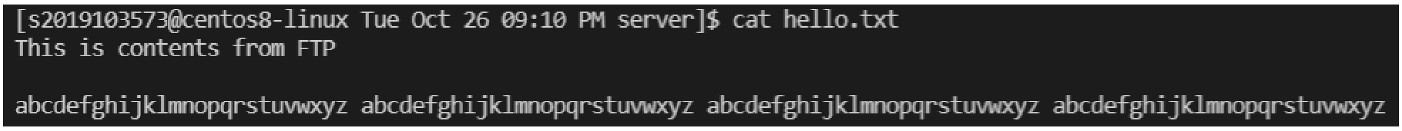
CLIENT OUTPUT



```
[s2019103573@centos8-linux Tue Oct 26 09:03 PM client]$ gcc -o client client.c
[s2019103573@centos8-linux Tue Oct 26 09:04 PM client]$ ./client
Connection established with 127.0.0.1 : 2800

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


 [-] Enter username : SachinRaghulT

 [-] Enter password : yokeshT

LOADING........
Connection established with 127.0.0.1 : 2500
Enter exit to close the connection.

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

---------------------------------------------
[-] Enter filename : hello.txt
File found.
FILE RECEIVED AND STORED LOCALLY.


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

hello.txt


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

This is contents from FTP

abcdefghijklmnopqrstuvwxyz abcdefghijklmnopqrstuvwxyz abcdefghijklmnopqrstuvwxyz abc
defghijklmnopqrstuvwxyz

---------------------------------------------
```

FTP FILE :-

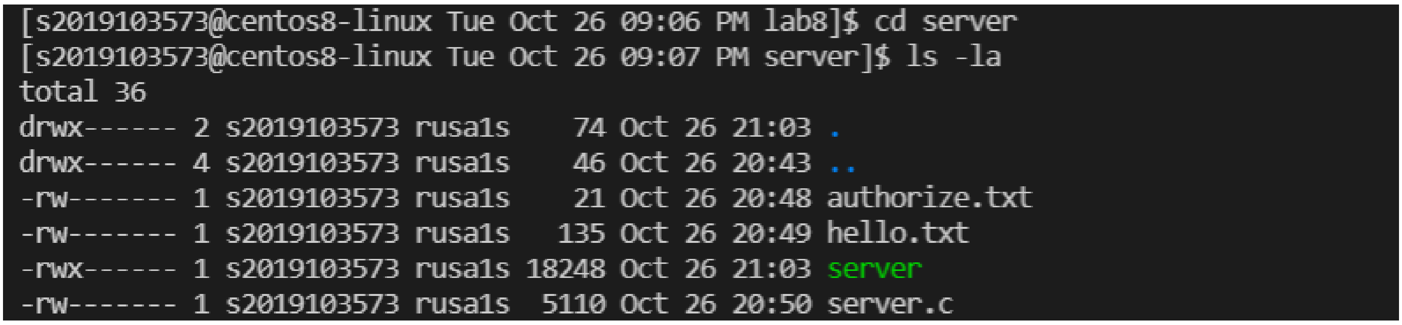
hello.txt



```
[s2019103573@centos8-linux Tue Oct 26 09:10 PM server]$ cat hello.txt
This is contents from FTP

abcdefghijklmnopqrstuvwxyz abcdefghijklmnopqrstuvwxyz abcdefghijklmnopqrstuvwxyz abcdefghijklmnopqrstuvwxyz
```

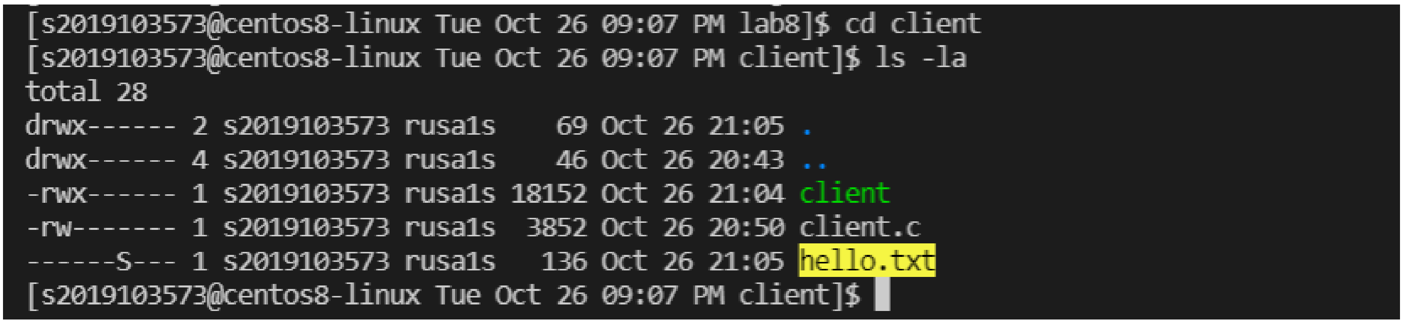
AFTER EXECUTION :-

SER VER DIRECTORY



```
[s2019103573@centos8-linux Tue Oct 26 09:06 PM lab8]$ cd server
[s2019103573@centos8-linux Tue Oct 26 09:07 PM server]$ ls -la
total 36
drwx------ 2 s2019103573 rusa1s    74 Oct 26 21:03 .
drwx------ 4 s2019103573 rusa1s    46 Oct 26 20:43 ..
-rw------- 1 s2019103573 rusa1s    21 Oct 26 20:48 authorize.txt
-rw------- 1 s2019103573 rusa1s   135 Oct 26 20:49 hello.txt
-rwx------ 1 s2019103573 rusa1s 18248 Oct 26 21:03 server
-rw------- 1 s2019103573 rusa1s  5110 Oct 26 20:50 server.c
```

CLIENT DIRECTORY



```
[s2019103573@centos8-linux Tue Oct 26 09:07 PM lab8]$ cd client
[s2019103573@centos8-linux Tue Oct 26 09:07 PM client]$ ls -la
total 28
drwx------ 2 s2019103573 rusa1s    69 Oct 26 21:05 .
drwx------ 4 s2019103573 rusa1s    46 Oct 26 20:43 ..
-rwx------ 1 s2019103573 rusa1s 18152 Oct 26 21:04 client
-rw------- 1 s2019103573 rusa1s  3852 Oct 26 20:50 client.c
------S--- 1 s2019103573 rusa1s   136 Oct 26 21:05 hello.txt
[s2019103573@centos8-linux Tue Oct 26 09:07 PM client]$ 
```