ARUN M

76
R6B

Networking Lab

V. Implement File Transfer Protocol.

Client

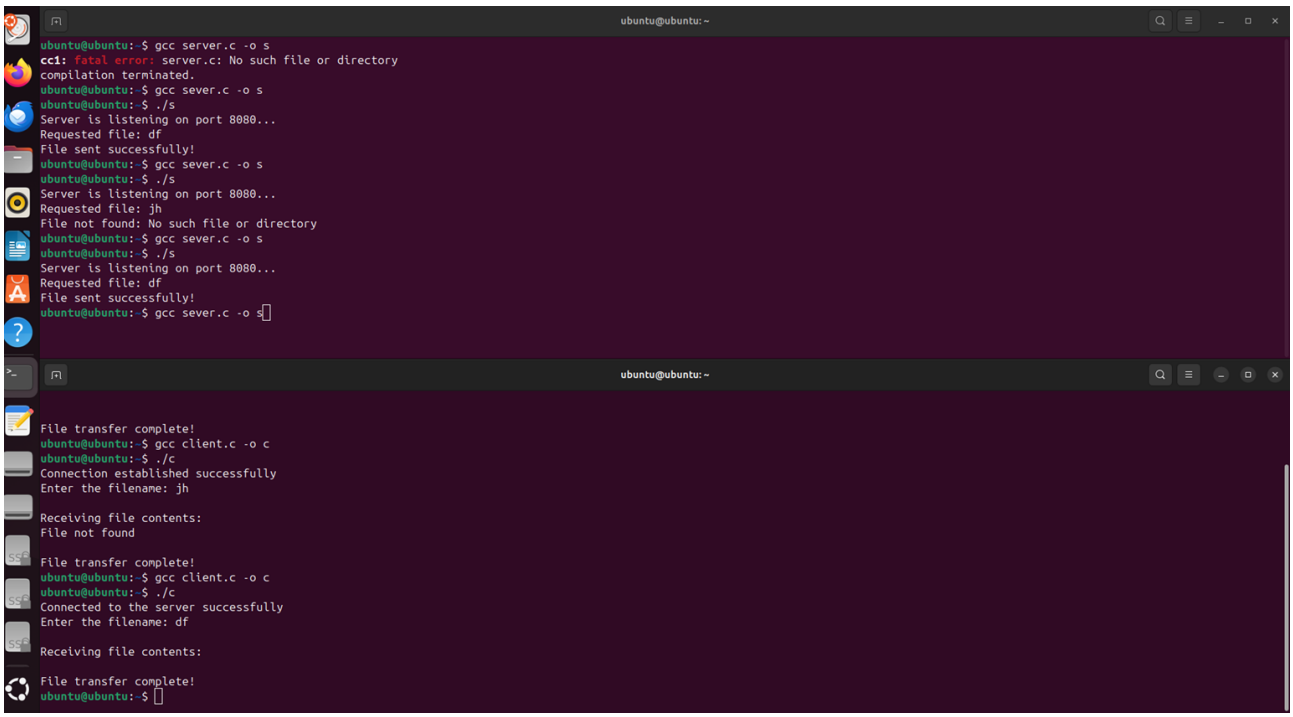
```
#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <stdlib.h>
#include <arpa/inet.h>
#include <string.h>
#include <unistd.h>

int main() {
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0) {
        perror("Socket creation failed");
        return 1;
    }

    struct sockaddr_in server_addr;
    server_addr.sin_family = AF_INET;
    server_addr.sin_port = htons(8080);
    server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");

    if (connect(sockfd, (struct sockaddr*)&server_addr, sizeof(server_addr)) < 0) {
        perror("Connection failed");
        close(sockfd);
        return 1;
    }

    printf("Connected to the server successfully\n");

    char filename[100];

    printf("Enter the filename: ");
    scanf("%99s", filename);
    if (send(sockfd, filename, strlen(filename), 0) < 0) {
        perror("Failed to send filename");
        close(sockfd);
        return 1;
    }
```

```
    char buffer[1000];
    int bytes_received;

    printf("\nReceiving file contents:\n");
    while ((bytes_received = recv(sockfd, buffer, sizeof(buffer) - 1, 0)) > 0) {
        buffer[bytes_received] = '\0';

        printf("%s", buffer);
    }

    if (bytes_received < 0) {
        perror("Error receiving file data");
    }

    printf("\nFile transfer complete!\n");


    return 0;
}
```

server

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

int main() {
    int server_fd, new_socket;
    struct sockaddr_in server_address;
    socklen_t address_length = sizeof(server_address);

    server_fd = socket(AF_INET, SOCK_STREAM, 0);
    if (server_fd < 0) {
        perror("Failed to create socket");
        return 1;
    }

    server_address.sin_family = AF_INET;
    server_address.sin_addr.s_addr = inet_addr("127.0.0.1");
    server_address.sin_port = htons(8080);

    if (bind(server_fd, (struct sockaddr *)&server_address, sizeof(server_address)) < 0) {
        perror("Binding failed");
```

```
    close(server_fd);
    return 1;
}
if (listen(server_fd, 3) < 0) {
    perror("Failed to listen");
    close(server_fd);
    return 1;
}

printf("Server is listening on port 8080...\n");

new_socket = accept(server_fd, (struct sockaddr *)&server_address, &address_length);
if (new_socket < 0) {
    perror("Failed to accept connection");
    close(server_fd);
    return 1;
}

char filename[100], buffer[1000];

int bytes_received = recv(new_socket, filename, sizeof(filename) - 1, 0);
if (bytes_received < 0) {
    perror("Failed to receive filename");
    close(new_socket);
    close(server_fd);
    return 1;
}

filename[bytes_received] = '\0';
printf("Requested file: %s\n", filename);

FILE *file = fopen(filename, "r");
if (file == NULL) {
    perror("File not found");
    send(new_socket, "File not found", strlen("File not found"), 0);
    close(new_socket);
    close(server_fd);
    return 1;
}

while (fgets(buffer, sizeof(buffer), file) != NULL) {
    send(new_socket, buffer, strlen(buffer), 0);
}

printf("File sent successfully!\n");

fclose(file);
close(new_socket);
close(server_fd);

return 0;
```


ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ gcc server.c -o s
cc1: fatal error: server.c: No such file or directory
compilation terminated.
ubuntu@ubuntu:~$ gcc sever.c -o s
ubuntu@ubuntu:~$ ./s
Server is listening on port 8080...
Requested file: df
File sent successfully!
ubuntu@ubuntu:~$ gcc sever.c -o s
ubuntu@ubuntu:~$ ./s
Server is listening on port 8080...
Requested file: jh
File not found: No such file or directory
ubuntu@ubuntu:~$ gcc sever.c -o s
ubuntu@ubuntu:~$ ./s
Server is listening on port 8080...
Requested file: df
File sent successfully!
ubuntu@ubuntu:~$ gcc sever.c -o s
ubuntu@ubuntu: ~
File transfer complete!
ubuntu@ubuntu:~$ gcc client.c -o c
ubuntu@ubuntu:~$ ./c
Connection established successfully
Enter the filename: jh
Receiving file contents:
File not found
File transfer complete!
ubuntu@ubuntu:~$ gcc client.c -o c
ubuntu@ubuntu:~$ ./c
Connected to the server successfully
Enter the filename: df
Receiving file contents:
File transfer complete!
ubuntu@ubuntu:~$

}