

Code

Server code

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <sys/socket.h>

#include <netinet/in.h>


#define PORT 8080


void handle_client(int client_socket) {

    char buffer[1024];

    FILE *file;


    // Authentication

    memset(buffer, 0, sizeof(buffer));

    recv(client_socket, buffer, sizeof(buffer), 0);

    char username[50], password[50];

    sscanf(buffer, "%s %s", username, password);


    if (strcmp(username, "user1") == 0 && strcmp(password, "pass123") == 0) {

        send(client_socket, "LOGIN SUCCESS", 13, 0);

    } else {

        send(client_socket, "LOGIN FAILED", 12, 0);

        close(client_socket);

        return;

    }

}
```

```

while (1) {

    memset(buffer, 0, sizeof(buffer));

    int bytes_received = recv(client_socket, buffer, sizeof(buffer), 0);

    if (bytes_received <= 0) break;


    if (strncmp(buffer, "UPLOAD ", 7) == 0) {

        char filename[100];

        sscanf(buffer + 7, "%s", filename);


        file = fopen(filename, "w");

        send(client_socket, "READY", 5, 0);


        memset(buffer, 0, sizeof(buffer));

        recv(client_socket, buffer, sizeof(buffer), 0);

        fprintf(file, "%s", buffer);

        fclose(file);


        send(client_socket, "UPLOAD OK", 9, 0);
    }

    else if (strcmp(buffer, "LIST") == 0) {

        system("ls > temp.txt");

        file = fopen("temp.txt", "r");

        memset(buffer, 0, sizeof(buffer));

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

        fclose(file);


        send(client_socket, buffer, strlen(buffer), 0);
    }
}

```

```

    }

    else if (strncmp(buffer, "DOWNLOAD ", 9) == 0) {

        char filename[100];

        sscanf(buffer + 9, "%s", filename);


        file = fopen(filename, "r");

        if (!file) {

            send(client_socket, "NO FILE", 7, 0);

        } else {

            memset(buffer, 0, sizeof(buffer));

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

            fclose(file);

            send(client_socket, buffer, strlen(buffer), 0);

        }

    }

    else if (strcmp(buffer, "EXIT") == 0) {

        break;

    }

}

close(client_socket);

}

```

```

int main() {

    int server_socket, client_socket;

    struct sockaddr_in server_addr, client_addr;

    socklen_t len = sizeof(client_addr);


    server_socket = socket(AF_INET, SOCK_STREAM, 0);

```

```

server_addr.sin_family = AF_INET;
server_addr.sin_port = htons(PORT);
server_addr.sin_addr.s_addr = INADDR_ANY;

if (bind(server_socket, (struct sockaddr*)&server_addr, sizeof(server_addr)) < 0) {
    perror("Bind failed");
    exit(1);
}

listen(server_socket, 5);
printf("Server Running on port %d...\n", PORT);

while (1) {
    client_socket = accept(server_socket, (struct sockaddr*)&client_addr, &len);
    printf("Client Connected.\n");
    handle_client(client_socket);
}

close(server_socket);
return 0;
}

```

Client code

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

```

```
#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>


#define PORT 8080


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

    int sock;

    struct sockaddr_in server_addr;

    char buffer[1024];


    if (argc != 2) {

        printf("Usage: %s <server_ip>\n", argv[0]);

        return 1;

    }


    sock = socket(AF_INET, SOCK_STREAM, 0);


    server_addr.sin_family = AF_INET;

    server_addr.sin_port = htons(PORT);

    inet_pton(AF_INET, argv[1], &server_addr.sin_addr);


    if (connect(sock, (struct sockaddr*)&server_addr, sizeof(server_addr)) < 0) {

        perror("Connect failed");

        return 1;

    }


    // Authentication
```

```
char username[50], password[50];

printf("Enter username: ");
scanf("%s", username);

printf("Enter password: ");
scanf("%s", password);

sprintf(buffer, "%s %s", username, password);
send(sock, buffer, strlen(buffer), 0);
```

```
memset(buffer, 0, sizeof(buffer));
recv(sock, buffer, sizeof(buffer), 0);
printf("%s\n", buffer);
```

```
if (strcmp(buffer, "LOGIN FAILED") == 0) {
    close(sock);
    return 0;
}
```

```
while (1) {
    printf("Enter command (UPLOAD <file>, DOWNLOAD <file>, LIST, EXIT):\n> ");
    scanf(" %[^\n]", buffer);

    if (strncmp(buffer, "UPLOAD ", 7) == 0) {
        char filename[100];
        sscanf(buffer + 7, "%s", filename);

        send(sock, buffer, strlen(buffer), 0);
        memset(buffer, 0, sizeof(buffer));
    }
}
```

```

recv(sock, buffer, sizeof(buffer), 0);
printf("Server: %s\n", buffer);

FILE *file = fopen(filename, "r");
if (file) {
    memset(buffer, 0, sizeof(buffer));
    fread(buffer, 1, sizeof(buffer), file);
    fclose(file);
    send(sock, buffer, strlen(buffer), 0);

    memset(buffer, 0, sizeof(buffer));
    recv(sock, buffer, sizeof(buffer), 0);
    printf("Server: %s\n", buffer);
} else {
    printf("File not found.\n");
}
}
else {
    send(sock, buffer, strlen(buffer), 0);
    memset(buffer, 0, sizeof(buffer));
    recv(sock, buffer, sizeof(buffer), 0);
    printf("Server: %s\n", buffer);
}

if (strcmp(buffer, "EXIT") == 0) break;
}

close(sock);

```

```
return 0;
```

```
}
```