

Hướng dẫn thực hành đồ án mạng máy tính

Lập trình socket

I. Nội dung và yêu cầu

1. **Nội dung:** Chương trình mô phỏng ứng dụng File Server gồm một server và nhiều client. Server sẽ lưu trữ file để các client lên tải về. Mọi quá trình truyền file cũng như mọi thông báo (client đăng nhập, client đăng xuất) đều được thể hiện trên màn hình của server.
2. **Yêu cầu:**
 - Server phải cho phép nhiều client đăng nhập và tải file cùng lúc. Trên lý thuyết, số lượng client là không giới hạn.
 - Client cần đăng ký một username và password để đăng nhập. Nếu username đã được sử dụng, server sẽ yêu cầu client chọn username khác cho đến khi không bị trùng. Client sẽ sử dụng tên này trong toàn bộ quá trình trao đổi với server. Tại giao diện server phải có mục thể hiện danh sách tất cả client đang kết nối đến.
 - Khi một client đăng nhập hoặc đăng xuất (hoặc thoát chương trình), phải có dòng thông báo trên màn hình server và mọi client. Ví dụ:
 - o NguyenVanA log off
 - o LeVanC log in
 - Nếu server thoát, các client đều phải nhận biết là ngắt kết nối.
 - Tại mỗi client cần hiển thị danh sách các file mà server chia sẻ để client có thể chọn tải về. Lưu ý: Cùng một thời điểm server cho phép nhiều client tải về cùng 1 file và kích thước file có thể lên đến **200 MB**.

II. Tổng quan về socket programming

- Socket for windows : Winsock programming, client – server infrastructure
- Winsock version: first you should specify the version of winsock you wish to use. Version 2 is the latest version of winsock at this time.
- SOCKADDR_IN : a structure to contain the internet address known as: SOCKADDR_IN. The definition of that structure is:

```
struct SOCKADDR_IN
```

```
{  
    short    sin_family           //specifies the address family, usually the constant "AF_INET"  
    u_short  sin_port;           // specifies the port number and must be used with the htons()  
    struct    in_addr sin_addr;   // hold the IP address returned by inet_addr()  
    char      sin_zero[8];       // usually set to 0  
}
```

- **Socket functions:**

- `htons()` = function which converts the “host byte order” to “network byte order”
- `inet_addr()` = this function takes the string you pass into it as an argument and converts it to the required form for use in the socket connection, negating the need to use `htonl()`.
- `WSAStartup()` : return a integer or long value = starts the Winsock API.
- `bind()` = binds a socket to the `SOCKADDR_IN` structure containing the IP address and port used to build the connection.
- `listen()` = tells a socket to listen for an incoming connection. It takes 2 arguments: the socket to listen on and MAXIMUM number of connections (`SOMAXCONN` depends on system)
- `accept()` = waits for a connection and wakes when a connection is established, usually when coding the SERVER part of a client=server application.
- `connect()` = connects a client to server that is in the “listening” state and set to “accept” a connection.
- `WSACleanup()` = stops the Winsock API
- `closesocket()` = closes an open Socket.

- **Categories :** TCP Socket and UDP Sockets

The process for **initiating** a **TCP** socket connection is:
`SOCKET sock_X = socket(AF_INET, SOCK_STREAM, NULL);`

The method for **sending** data with **TCP** sockets is:
`answer = send(sock_X, "You have connected to CG SERVER!", 46, NULL);`

The method for **receiving** data with **TCP** sockets is:
`answer = recv(sock_X, MESSAGE, sizeof(MESSAGE), NULL);`

- **Khai báo chung cho winsock**

```
#include <winsock2.h>
```

```
#include <ws2tcpip.h>
```

```
#include <stdio.h>
```

```
#pragma comment(lib, "Ws2_32.lib")
```

```
int main() {
```

```
    return 0;
```

```
}
```

- **Nguồn tham khảo**

<https://docs.microsoft.com/en-us/windows/desktop/winsock/creating-a-basic-winsock-application>

<https://docs.microsoft.com/en-us/windows/desktop/winsock/getting-started-with-winsock>

<https://www.youtube.com/watch?v=NIqYwXcUdn0&list=PLD9o6GiGMnJm1HKXgRnA1zC56LsFE5CMG>

III. Danh sách các chức năng trên file transfer protocol

- + **gethostbyname()**; // Truy xuất thông tin máy chủ thông qua tên máy chủ, nếu không có lỗi sẽ trả về con trỏ hostent.
- + **chuanHoa(string &s)**; // Chuẩn hóa lệnh nhập vào (Xóa các dấu khoảng trắng, dấu tab).
- + **getFileName(string &cmd)**; // Trả về tên file.
- + **errexist(const char *, ...)**; // Thông báo khi có lỗi kết nối socket tới server.
- + **pause(void)**; // Tạm dừng màn hình, nhấn phím bất kì để tiếp tục.
- + **replylogcode(int code)**; // Xuất lỗi.
- + **LoginFTP(SOCKET socket_descriptor, HOSTENT * pHostEnt)**; // Đăng nhập FTP (gửi user, password).
- + **dir(SOCKET soc, string cmd, bool modePasv)**; // Liệt kê thư mục, tập tin trên Server (chi tiết hơn ls).
- + **ls(SOCKET soc, string cmd, bool modePasv)**; // Liệt kê thư mục, tập tin trên Server.
- + **put(SOCKET soc, string cmd, bool modePasv)**; // Upload một file lên Server.
- + **mput(SOCKET soc, string cmd, bool modePasv)**; // Upload nhiều file lên Server.
- + **get(SOCKET soc, string cmd, bool modePasv)**; // Download một file từ Server.
- + **mget(SOCKET soc, string cmd, bool modePasv)**; // Download nhiều file từ Server.
- + **cd(SOCKET soc, string cmd)**; // Thay đổi đường dẫn trên Server.
- + **lcd(char cmd[])**; // Thay đổi đường dẫn ở Client.
- + **Delete(SOCKET soc, string cmd)**; // Xóa một file.
- + **mDelete(SOCKET soc, string cmd, bool modePasv)**; // Xóa nhiều file
- + **mkdir(SOCKET soc, string cmd)**; // Tạo thư mục trên Server
- + **rmdir(SOCKET soc, string cmd)**; // Xóa thư mục rỗng trên Server
- + **pwd(SOCKET soc)**; // Hiển thị đường dẫn hiện tại của Server
- + **passiveMode(SOCKET soc, SOCKET &dsoc)**; // Chuyển sang passive, trả về bool
- + **activeMode(SOCKET soc, SOCKET &dsoc, int stat)**; // Chuyển sang active trả về bool.
- + **quit(SOCKET soc)**; // Thoát khỏi server.

Nguồn tham khảo: trên file đính kèm

IV. Các kênh tham khảo chính:

1. <https://github.com/gudipati/FTP-ClientServer>
2. <https://github.com/hmhuan/DemoFTPClient>

V. Workplace:

Mọi người đều có thể truy cập và làm việc trên repository này.

https://github.com/VoQuocThang/FTP_SOCKET_PROGRAMMING