وزارت علوم، تحقیقات و فناوری
دانشگاه تحصیلات تکمیلی علوم پایه
گاوازنگ – زنجان

نام : رادین صاحبدل

شماره دانشجویی : 14024142

نام استاد : دکترپهلوانی

نام درس : ACN

تمرین شماره 2

فصل پاییز

FTP Server Programming

The File Transfer Protocol (FTP) is a standard communication protocol used for the transfer of computer files from a server to a client on a computer network. FTP is built on a client–server model architecture. You have to write a C++ program for a FTP Server to do the following:

- Upload a file
- Download a file
- Delete a file
- Search for file
- List the files
- Data Encryption (Optional)

Use CMAKE to build your program.

Your program should compile in a Linux environment.

Push your codes in a GitHub repository.

You can use a FTP Client app like **FileZilla** or **WinSCP** to test your server.

You must send a PDF as your program's documentation containing a link to your GitHub repository and screenshots from a FTP Client application showing the testing process and results.

در این تمرین باید برنامه ای با C++ برای یک سرور FTP بنویسیم تا قابلیت‌های زیر را داشته باشد:

آپلود فایل :امکان ارسال فایل از سمت کلاینت به سرور.

دانلود فایل :امکان دریافت فایل از سرور به کلاینت.

حذف فایل :قابلیت حذف یک فایل از روی سرور.

جستجوی فایل :امکان جستجو برای فایل‌های موجود در سرور.

فهرست فایل‌ها :نمایش لیستی از فایل‌های موجود روی سرور.

رمزگذاری داده‌ها (اختیاری) :امکان رمزگذاری اطلاعات برای امنیت بیشتر.

این برنامه باید با استفاده از CMakeساخته شود و در محیط لینوکس کامپایل شود. همچنین باید کدها را در یک مخزن GitHubآپلود کنیم و مستندات برنامه شامل لینک به مخزن و تصاویر تست با استفاده از کلاینت FTP مانند FileZillaیا WinSCPرا ارسال کنید.

با سلام

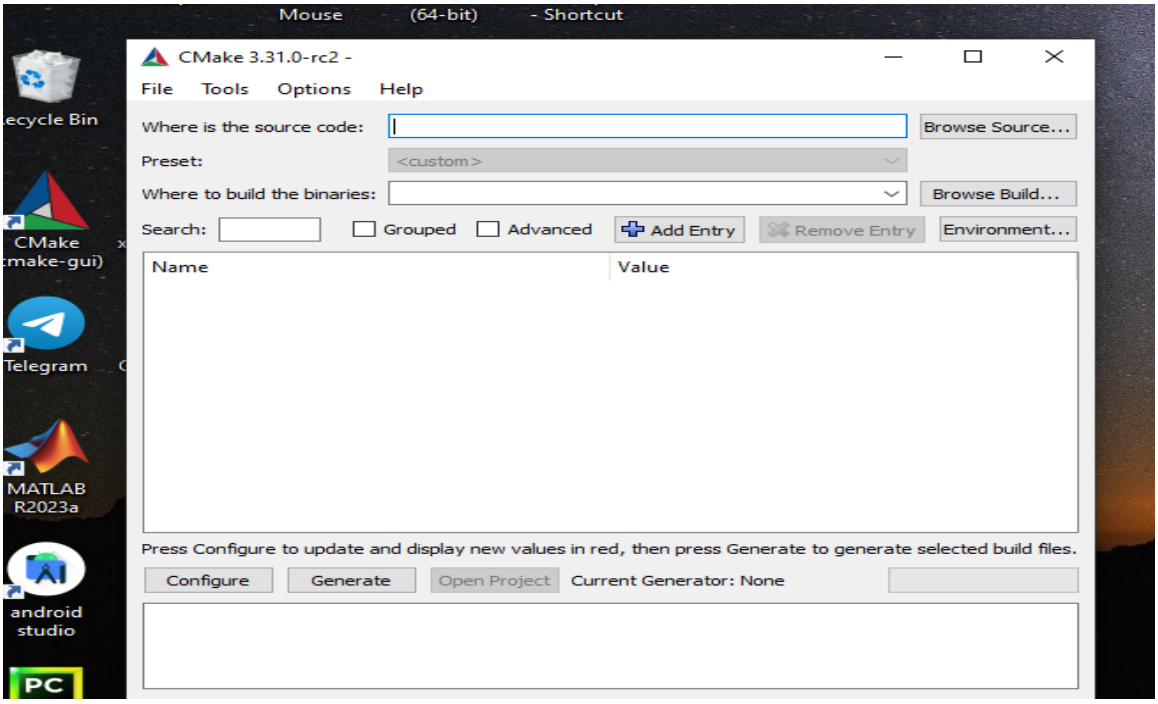
لازم به ذکرهست که من رولپتاپم لینوکس ندارم واقعیتش فرصت نکردم بزنم حتما برا تمرین بعد میزنم .

مراحل انجام

اول باید با دستورات زیر بیایم ببینیم که لینوکس آپدیت هست یا نه

```
sudo apt update
sudo apt upgrade
```

برای کامپایل کردن برنامه های مثل cmake و gitub نیازمندیم من cmake رو هم نصب کردم .



sudo apt install build-essential cmake git

کد برای سرورFTP :

```
#include <iostream>
 #include <fstream>
 #include <filesystem>
#include <thread>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <unistd.h>
namespace fs = std::filesystem;
void handle_client(int client_socket) {
char buffer[1024];
int bytes_received;
while ((bytes_received = recv(client_socket, buffer, 1024, 0)) > 0) {
buffer[bytes_received] = '\0';
 std::string command(buffer);
if (command.find("UPLOAD") == 0) {
 // مدیری ت آپلود فایل
 std::string filename = command.substr(7);// "UPLOAD filename"
std::ofstream file(filename, std::ios::binary);
while ((bytes_received = recv(client_socket, buffer, 1024, 0)) > 0) {
```

```
file.write(buffer, bytes_received);
        }
        file.close();
}
else if (command.find("DOWNLOAD") == 0) { مدیری ت دانلود فایل
// std::string filename = command.substr(9); // "DOWNLOAD filename"
std::ifstream file(filename, std::ios::binary);
while (!file.eof()) {
file.read(buffer, 1024);
send(client_socket, buffer, file.gcount(), 0);
} file.close();
} else if (command.find("DELETE") == 0) {
// مدیریت حذف فایل
std::string filename = command.substr(7);
// "DELETE filename"
fs::remove(filename);
} else if (command.find("LIST") == 0) {
// مدیری ت لیست کردن فایل ها
for (const auto &entry : fs::directory_iterator(".")) {
std::string filename = entry.path().filename();
send(client_socket, filename.c_str(), filename.size(), 0); send(client_socket, "\n",
1, 0);
}
}
else if (command.find("SEARCH") == 0) {
```

```
// مدیری ت جستجوی فایل
std::string filename = command.substr(7); // "SEARCH filename"
 bool found = fs::exists(filename);
 std::string response = found ? "FOUND\n" : "NOT FOUND\n";
send(client_socket, response.c_str(), response.size(), 0);
}
else {
std::string msg = "است اشتباه دستور\n";
 send(client_socket, msg.c_str(), msg.size(), 0);
}
 } close(client_socket);
}
 int main() {
        int server_socket = socket(AF_INET, SOCK_STREAM, 0);
        struct sockaddr_in server_address;
        server_address.sin_family = AF_INET;
        server_address.sin_port = htons(8080); server_address.sin_addr.s_addr =
        INADDR_ANY;
        bind(server_socket, (struct sockaddr*)&server_address,
        sizeof(server_address));
         listen(server_socket, 5);
        srd::count<< 8080 راه اندازی شد.در حال گوش دادن به پورت“<<std::endl;

        while (true) {
```

```
 int client_socket = accept(server_socket, nullptr, nullptr);
std::thread(client_thread, handle_client, client_socket).detach();
}
 close(server_socket);
 return 0;
 }
```

درکد بالا کد یک سرور FTP ساده را با استفاده از زبان C++ پیاده‌سازی کرده ایم که می‌توانیم دستورات مختلفی مانند آپلود، دانلود، حذف، فهرست‌کردن و جستجوی فایل‌ها را اجرا کند. بیایید به جزئیات کد بپردازیم:

کتابخانه‌های مورد استفاده

Iostream: برای ورودی و خروجی استاندارد مانند چاپ پیام‌ها.

fstream: برای خواندن و نوشتن فایل‌ها.

filesystem: برای مدیریت فایل‌ها و دایرکتوری‌ها.

thread: برای ایجاد چندین رشته (thread) که به طور همزمان به درخواست‌های کلاینت‌ها پاسخ دهد.

sys/socket.h, arpa/inet.h, unistd.h :برای ایجاد و مدیریت اتصالات شبکه‌ای.

تابع handle_client

این تابع درخواست‌های کلاینت را مدیریت می‌کند و شامل چندین دستور مختلف است:

1. **دریافت داده از کلاینت**:

یک بافر به طول 1024 بایت برای دریافت داده‌ها از کلاینت تعریف شده است.

داده‌ها با استفاده از recvدریافت می‌شوند و در متغیر commandذخیره می‌شوند.

2. **دستورات**:**FTP**

آپلود فایل :زمانی که کلاینت درخواست "UPLOAD filename" ارسال می‌کند:

نام فایل استخراج شده و یک فایل خروجی برای نوشتن آن ایجاد می‌شود.

داده‌های فایل تا زمانی که از کلاینت دریافت می‌شوند، در این فایل نوشته می‌شوند و سپس فایل بسته می‌شود.

دانلود فایل :زمانی که کلاینت درخواست "DOWNLOAD filename" ارسال می‌کند:

فایل مشخص‌شده باز می‌شود و داده‌ها به کلاینت ارسال می‌شوند تا زمانی که به انتهای فایل برسیم.

حذف فایل :زمانی که کلاینت درخواست "DELETE filename" ارسال می‌کند:

فایل با نام مشخص‌شده حذف می‌شود.

لیست فایل‌ها :زمانی که کلاینت درخواست "LIST" ارسال می‌کند:

از طریق filesystem، تمام فایل‌ها در دایرکتوری فعلی خوانده شده و نام آن‌ها به کلاینت ارسال می‌شود.

جستجوی فایل :زمانی که کلاینت درخواست "SEARCH filename" ارسال می‌کند:

بررسی می‌شود که فایل وجود دارد یا خیر و سپس پیامی به کلاینت ارسال می‌شود که فایل یافت شده است یا خیر.

3. **پاسخ به دستورات نامعتبر** :اگر دستور نامعتبر باشد، پیامی به کلاینت ارسال می‌شود.

تابع main

تابع اصلی برنامه که سرور را راه‌اندازی می‌کند:

1. **ایجاد سوکت سرور**:

با استفاده از socketیک سوکت TCP ایجاد می‌شود.

2. **پیکربندی سرور**:

با استفاده از ساختار sockaddr_in، تنظیمات سرور (مانند نوع آدرس و شماره پورت) تعیین می‌شود.

bind:برای اختصاص آدرس و پورت به سوکت سرور استفاده می‌شود.

listen:برای شروع گوش دادن به درخواست‌های ورودی از کلاینت‌ها استفاده می‌شود.

3. **پذیرش کلاینت‌ها و ایجاد رشته جدید**:

در یک حلقه بی‌پایان، درخواست‌های ورودی پذیرفته می‌شوند و برای هر کلاینت یک رشته جدید برای اجرای تابع handle_clientایجاد می‌شود.

4. **پایان کار سرور**:

در پایان، closeبرای بستن سوکت سرور استفاده می‌شود.

برای ساخت برنامه از Cmake استفاده میکنیم .فایل Cmakelistes.txt رانیز ایجاد میکنیم .

```
cmake_minimum_required(VERSION 3.10)
 project(FTPServer)
 set(CMAKE_CXX_STANDARD 17)
 add_executable(FTPServer main.cpp)
```

برای ایجاد دایرکتوری ساخت:

```
mkdir build
cd build
cmake..
```

برای کامپایل بنامه هم :

```
Make
```

بعد این مرحله ها ما یک فایلی اجرایی ایجاد کردیم به نام FTPServer

تست با استفاده از کلاینت FTP

برنامه هایی مانندFileZilla یاWinSCP را نصب میکنیم سپس به سرور خودمون با استفاده ازlocalhost و پورت 8080متصل میشویم هر کدام از عملکردها آپلود، دانلود و غیره را تست میکنیم .

ایجاد Repositoryدر گیت هاب :

به حساب کاربری گیت هاب وارد میشویم .

مخزن جدید ایجاد میکنیم و نامی مناسب برای آن انتخاب میکنیم، مثال-Project-Server.
FTP دستورالعمل های گیتهاب را برای آپلود پروژه به آن دنبال میکنیم بعد ایجاد مخزن ، دستورات زیر را در ترمینال لینوکس اجرا میکنیم تا پروژه را به گیت هاب ارسال کنیم :

```
git init
git add .
git commit -m "Initial commit"
git branch -M main
```

git remote add origin https://github.com/RAdinin/14024142_FTPserver_project?tab=readme-ov-file

git push -u origin main

در کل ما الان میتونیم با اجرای این کد ، یک سرور FTP ساده راه‌اندازی کنیم که می‌تواند به درخواست‌های مختلف از جمله آپلود، دانلود، حذف، جستجو و نمایش لیست فایل‌ها پاسخ دهد.

سپاس