Network Programming Project 3 (Part 2)
HTTP Server and CGI Programs

NP TA

Deadline: Friday, 2022/12/02 23:59

1 Introduction

The project is divided into two parts. This is the second part of the project.
For this part, you are asked to provides the same functionality as part 1, but with some rules slightly differs:

1. Implement one program, **cgi_server.exe**, which is a combination of **http_server**, **panel.cgi**, and **console.cgi**.
2. Your program should run on **Windows 10**.

2 Specification

2.1 cgi_server.exe

1. The **cgi_server.exe** accepts TCP connections and parse the HTTP requests (as **http_server** does), and we will only test for the HTTP GET method.
2. You don't need to fork() and exec() since it's relatively hard to do it on Windows. Simply parse the request and do the specific job **within** the same process. We guarantee that in this part the URI of HTTP requests will be **"/panel.cgi"** or **"/console.cgi" plus a query string**:
   (a) If it is **/panel.cgi**,
   Display the panel form just like **panel.cgi** in part 1. This time, you can hard code the input file menu (t1.txt ~ t5.txt).
   (b) If it is **/console.cgi?h0=...**,
   Connect to remote servers specified by the query string. Note that the behaviors **MUST** be the same as part 1 **in the user's point of view** (though the procedure is different in this part).

2.2 test_case/ (Provided by TA)

1. This directory contains test cases, and each of which lists the commands to run remotely. You can put new test cases into this directory, and select it in the form generated by **panel.cgi**.

2.3 np_single_golden (Provided by TA)

1. This executable file is a Remote Working Ground Server in project2. We will use it for demo. You do **NOT** need to use your code for this server.

2.4 Execution Flow

2.4.1 Initial Setup

The structure of your working directory:

```
working_dir
    |-- cgi_server.exe
    |-- test_case/
```

2.4.2 Execution

1. Run your **cgi_server.exe** by **cgi_server.exe [port]**
2. Open a browser and visit `http://[NP_server_host]:[port]/panel.cgi`
3. Fill the form with the servers to connect to and select the input file, then click **Run**.
4. The web page will be automatically redirected to `http://[NP_server_host]:[port]/console.cgi` and your **console.cgi** should start now.

3 Requirements

1. You need to implement one program in this part: **cgi_server.exe**.
   Every function that touches networking operations (e.g., DNS query, connect, accept, send, receive) **MUST** be implemented using the library **Boost.Asio**. Directly using low-level system calls such as 'read', 'write', 'listen', are thereby **NOT** allowed.
2. All of the network operations should implement in **non-blocking (asynchronous)** approaches.
3. We will use **a MinGW distribution** (`https://nuwen.net/mingw.html`) with **18.0** distro to compile and execute your **cgi_server.exe**

   Below is an example command to compile your code in MinGW:
   `g++ cgi_server.cpp -o cgi_server -lws2_32 -lwsock32 -std=c++14`
4. You must provide **Makefile**. After typing command **"make part2"**, the executable **cgi_server.exe** should be generated. The executable should be placed **in the top layer of the directory**.
5. You can only use **C/C++** to implement this project. Except for **Boost**, other third-party libraries are **NOT allowed**.

4 Submission

1. New E3

   (a) Create a directory named as your student ID, and put your **Makefile** and **source codes in both part 1 and part 2** into the directory. Do **NOT** put anything else in it (e.g., **.git**, **__MACOSX**, **panel.cgi**, **test_case/**).

   (b) **zip** the directory and upload the .zip file to the New E3 platform
   Attention!! we only accept .zip format
   e.g. Create a directory 310550000, **zip the folder 310550000 into 310550000.zip**, and upload 310550000.zip to New E3. The zip file structure may be like:

```
310550000.zip
    |-- 310550000 (dir)
        |-- Makefile
        |-- http_server.cpp    # created in part 1
        |-- console.cpp        # created in part 1
        |-- cgi_server.cpp     # created in part 2
        |(other source codes)
```

2. Bitbucket:

   (a) You are **NOT** required to use git and Bitbucket for part 2.

3. **We take plagiarism seriously.**

```
All projects will be checked by a cutting-edge plagiarism detector.
You will get zero points on this project for plagiarism.
Please don't copy-paste any code from the internet, this may be
considered plagiarism as well.
Protect your code from being stolen.
```

5 Notes

1. You will lose points for violating any of the rules mentioned in this spec.
2. Enjoy the project!