Network Programming Project 3 (Part 2)
Remote Batch System

NP TA

Deadline: Monday, 2019/12/16 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. You only have to 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 OS**.

2 Specification

2.1 cgi_server.exe

1. The **cgi_server.exe** accepts TCP connections and parse the HTTP requests (as **http_server** does). The URI of HTTP requests will always be in the form of /${cgi_name}.cgi, 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 must be **"/panel.cgi"** or **"/console.cgi" plus a query string**:
   (a) If it is **/panel.cgi**,
   Display the panel form just like **panel.cgi**. This time, you can hardcode the input file menu (t1.txt ∼ t10.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 **in the user's point of view** (although the procedure is different in this part).

2.2 test_case/

It contains test cases that are literally the commands going to run remotely. You can put new test cases (with file name t1.txt ∼ t10.txt, since it's hard-coded in this part) into this directory, and select it by panel.cgi.

2.3 Execution Flow

2.3.1 Initial Setup

The structure of your working directory:

```
working_dir
    |----- cgi_server.exe    # test cases to run remotely
    |----- test_case/        # test cases to run remotely
```

2.3.2 Execution

1. Run your **cgi_server.exe** by **./cgi_server.exe [port]**
2. Open a browser and visit `http://localhost:[port]/panel.cgi`
3. Fill the form with the servers your want to connect to and select the input file then click **Run**.
4. The web page will automatically redirects to `http://localhost:[port]/console.cgi` and runs the **console.cgi** procedure just like Part 1 does.

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**.
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`) and the command:

   ```
   g++ main.cpp -o cgi_server -lws2_32 -lwsock32 -std=c++14
   ```

   to compile and generate your **cgi_server.exe**

   For the convenience of demoing, **do NOT** write your code in several sources and headers. Instead, write **EVERYTHING** in one "`main.cpp`".

4 About Submission

1. E3
   (a) Put your "`main.cpp`" and source codes of part 1 in **the same directory layer**. 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 E3 platform
   Attention!! we only accept .zip format
   e.g. Create a directory 0856000, the directory structure may be like:

```
0856000
    |-- Makefile          # created in part 1
    |-- http_server.cpp   # created in part 1
    |-- console.cpp       # created in part 1
    |-- main.cpp          # created in part 2
    |(other source codes)
```

Zip the folder 0856000 into 0856000.zip, and upload 0856000.zip to E3.

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.
   ```