

Network Programming Project 3 (Part 2)

Remote Batch System

NP TA

Deadline: Sunday, 2018/12/16 23:59

1 Introduction

In this project, you are asked to write a remote batch system that runs over HTTP.

The program runs on Windows operating system.

The difference between **part 1** and **part 2** is that:

1. You only have to implement one program (`cgi_server.exe`) in **part 2**.
2. `cgi_server.exe` is a combination of `http_server`, `panel.cgi` and `console.cgi` from **part 1**

2 Requirements

1. All programs in this project **MUST** be implemented using **Boost.Asio**. Directly using low-level network related system calls (e.g. `read`, `write`, `listen`, `accept`, `select` ...) is **NOT** allowed.
2. You are asked to implement **only one** program in this part: `cgi_server.exe`
3. All network operations (e.g. DNS query, `connect`, `accept`, `send`, `receive` ...) **MUST** be implemented with non-blocking (asynchronous) approaches.

3 Specification

3.1 `cgi_server.exe`

1. The `cgi_server` accepts TCP connections and parse the HTTP requests.
2. In this project, the URI of HTTP requests will always only be in the form of `/XXXXXX.cgi` (e.g. `/panel.cgi`), and we will only test the HTTP GET method.
3. The `cgi_server` should parse the HTTP headers (just like `http_server` in **part 1**). And do the specified task within the same process, since it is relatively hard to `fork()` and `exec()` on Windows.
 - (a) `panel.cgi`
 - i. Display the panel form as `panel.cgi` in part 1. But in this part, you can hardcode the input file menu (`t1.txt ~ t10.txt`).
 - (b) `console.cgi`
 - i. connect to remote servers specified in the GET query parameters
 - ii. The behaviors **MUST** be the same **in the user's point of view** (although the procedure is different in this part), please refer to `console.cgi` in part 1 for more details.

3.2 test_case/

1. Put all the test cases into this directory, and your program should send the files in this directory. Notice that the test cases are not in the same layer as **cgi_server.exe**

3.3 Execution Flow

3.3.1 Initial Setup

The structure of your working directory:

```
working_dir
|-----cgi_server.exe    # CGI server.
|-----test_case/        # Put all test cases into this folder.
```

3.3.2 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 you want 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 Remote Batch System (console.cgi) should start now.

3.4 About Submission

1. E3
 - (a) For the convenience during demo, please do **NOT** write your code in several sources and headers. Write **EVERYTHING** in one "main.cpp". I know this is not a good habit, sorry for that :(
 - (b) Do **NOT** submit the Visual Studio project. We only need your source code.
 - (c) Put your **main.cpp** in the **same directory in part 1**.
 - (d) Upload **ONLY** your code and Makefile.
Do **NOT** upload anything else (e.g. **.git**, **_MACOSX**, **panel.cgi**, **test_case/...**)
 - (e) **zip** the directory and upload the .zip file to the E3 platform
Attention!! we only accept .zip format
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.