Project1: Web Server Programming

CSE351 Introduction to Networks (Spring 2019)

Objective

The goal of this project is to implement a web server in three different ways: single-threaded, multi-threaded, and thread-pooled. These methods are known to provide substantial performance difference in serving multiple requests or clients. Understanding this performance difference by observing and monitoring the web server behaviors is a part of this project.

Program Specification

Skeleton code is provided in assignment tab of BB. Copy the code to your Linux System (VM, MacOS, Ubuntu, etc.), and modify it to implement your own web server. Your web server should send requested files to the client through Hyper Text Transfer Protocol (HTTP).
Check the message format of HTTP (https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol~~).~~

Basically, you should fill up empty parts in the skeleton code to run this code successfully. This code is consisted of two functions: main() and respond()

main() function lets a socket bound to address and port (using bind() function) and listen on the socket (using listen() function). Then, it accepts a connection from a client (using accept() function) and–sends the requested file to the client through HTTP message format. The server should continuously accept other clients after handling a request.

respond() function reads **a** request which contains **a** file path, takes the file path (e.g, index.html, public/image1.jpg, bootstreap.js), read the file, attach the HTTP header to make a response message, and send the response message back to the client.

Implementation Guides

There are three ways to implement a web server.

1. Single-threaded Server

   1) Wait for client request
   2) Process the request
   3) Repeat this cycle.

   All of these steps are executed in a single thread.

2. Multi-threaded Server

   Different from single-threaded server, multi-threaded server
   1) wait for client request
   2) when it accepts a new client connection, creates a new thread, and waits for another clients
   3) the new thread processes the new request

   ⇨ Requests of clients can be processed concurrently

3. Thread-pooled Server

   Web server creates threads very frequently, so it can consume lots of computing resources to create new threads. To solve this problem, thread-pooled server creates threads in advance, wake the thread and give task to the thread whenever it is needed.
   https://en.wikipedia.org/wiki/Thread_pool

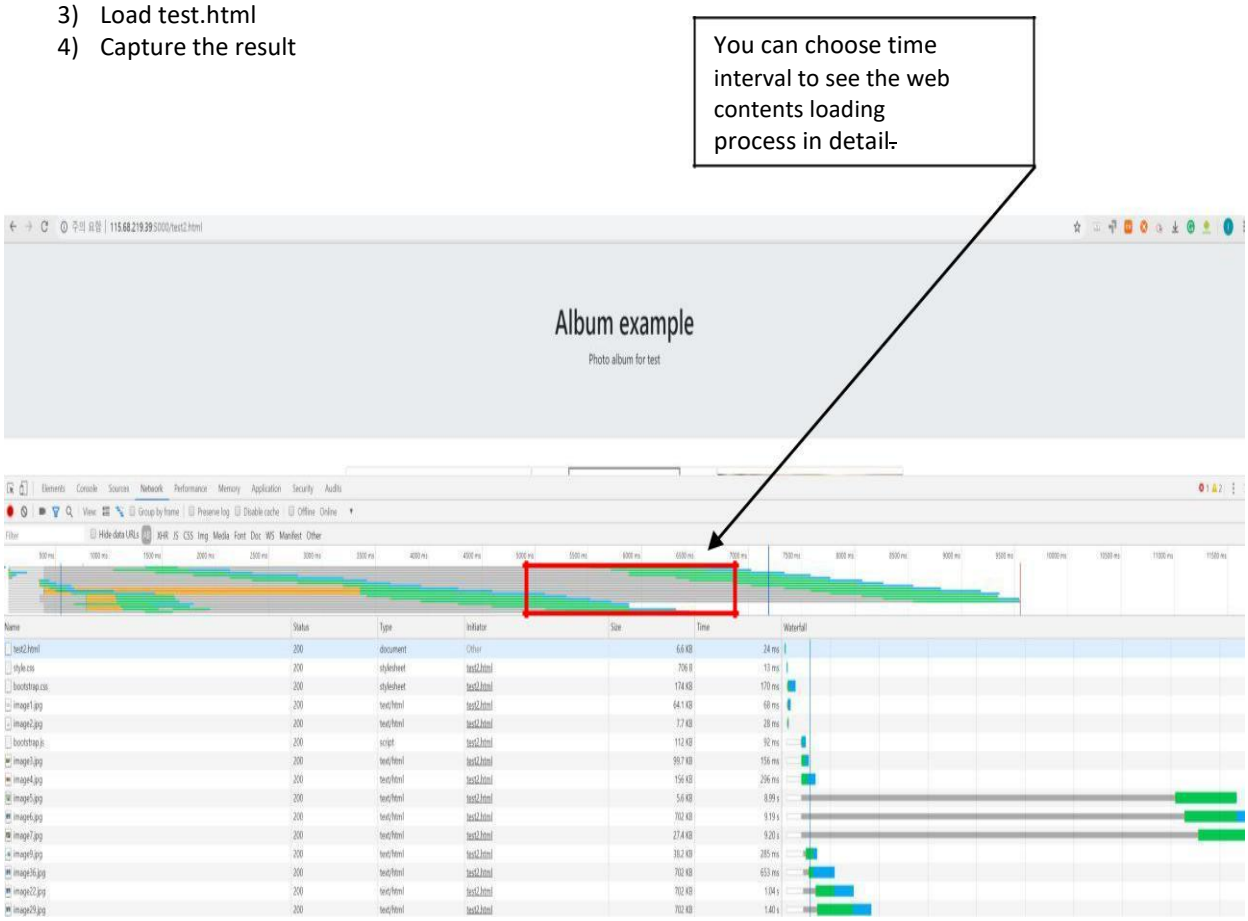
   The <threadpool/thpool.h> is provided in the skeleton code, so you can use this to implement thread-pooled server

http://tutorials.jenkov.com/java-multithreaded-servers/index.html

This link can be helpful to understand the implementation techniques.

Writing a Report

1) Open the Chrome developer tool (Ctrl + Shift + I)
2) Select ‘network’ tab
3) Load test.html
4) Capture the result



5) Describe what you find and understand from this monitoring tool in the Report.doc file

Describe the characteristics of single-threaded, multi-threaded, and thread-pooled methods.

Analyze network downloading time data and compare those three methods based on the graphs you obtain from monitoring tool.

Grading Policy

Single-threaded Server : 40%

Multi-threaded Server : 30%

Thread-pooled Server : 10%

Report : 20%

Compile and Usage

File name:

server.c

Compile:

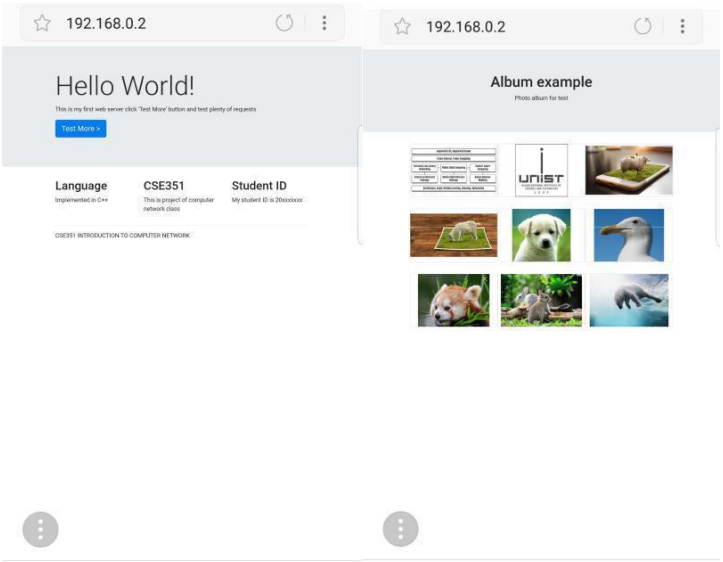
make

Usage:

1) Check ip address of your VM (type in “ifconfig” in terminal).
2) Compile your code (make).
3) Run server (./server).
4) Open web browser in your smart phone.
5) Open link “http://IP:5000”. (e.g., http://192.168.0.1:5000)

Output Specification

Before you submit code, change index.html file to include your student ID
Click “Test More” button and check bunch of image loading works well.



Submission

- Submit your codes to jongyunlee@unist.ac.kr.
- Title of the email should be in the format of “[CSE351] webserver_<student id>”.
- Include your report and code to a folder webserver_<student id> and compress the folder.
  - Make the compressed folder name as “webserver_<student id>”.
  - You should submit only one compressed file.