UNIVERSITY AT ALBANY
State University of New York

COLLEGE OF ENGINEERING AND APPLIED SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

ICSI 333 – System Fundamentals, Spring 2022

Project 4

The total grade for the assignment is 100 points.

You must follow the programming and documentation guidelines (see file *Programming Assignments Requirements and Recommendations.docx*).

Due date: 8:00 am on Monday, May 2, 2022.

This is a team project for two students, and you will need to find a partner. If the number of students in the class is odd, there will be one team of three students. Please contact your TA during the first three days since the project is open if you plan a team of three. Likely the first request will be accepted.

Your project will not be accepted and graded if you have no teammate. Nevertheless, each student must submit the project individually, and each student will be graded separately.

Each team member must participate in developing, documenting, and testing the program. Each team should include additional documentation at the beginning of the source file indicating how the work for the assignment was divided between the team members. Please specify who developed each part and how the testing work was divided between the team members.

DESCRIPTION

In this project, you must use C language to write a simple web server, and you will need knowledge about networking, threading, and file system calls.

A web server is a program that connects to a port, listens for requests, and sends the appropriate file to the requestor. Web servers must respond to requests and create a response using HTTP (Hypertext Transfer Protocol). HTTP is an application-level protocol that provides a way to interact with web resources such as HTML files by transmitting hypertext messages between clients and servers. You can find more information about HTTP online from the popular Wikipedia page to the most updated standards.

HTTP utilizes specific request methods to perform various tasks. Web servers may not use all of them, but every web server support `GET` request. For your web server, you will implement only this request.

Your web server must use TCP connection to receive a request that consists of the word "`GET`", a space, a path to the file, a space, and the version of HTTP – all on the first line. There will be other lines in a real request, but you can ignore them in this project.

Your web server program must generate a response that will consist of

(i) the string "`HTTP/1.0`"; followed by
(ii) a message with the code `200` - for success or `404` - for not existing file;
(iii) a new line; then
(iv) the string "`Content-Length:` " + the number of bytes in the message (ii); followed by
(v) **two** new lines; and
(vi) (vi) the content of the file.

If the file is open, the good alternative in (iv) is the file size, not the message size, though either solution is acceptable.

To send the request, you may create another program – the client or use a utility like `telnet` or `nc`. You can also use two computers or one computer with two terminal windows and `localhost` as a hostname. Your program should run on port 8000 (port 80, which is usual for web applications, will be blocked by your firewall).

Your web server program should take a path as a command-line argument (and fail with a usage message if there is no parameter). That path will be the path that your web server will use to serve files.

For example, the path given as an argument is `/home`. If the request is aimed to `localhost:8000/data.html`, your web server will look at: `/home` and try to serve `data.html`. If the file is there, it should return it with a "`200 OK`" message, as above. If the file does not exist, you should return a "`404 File Not Found`" message with a pre-designed response (it could be a file or could be a hard-coded message).

An example session follows (red is the request, gray is the request part that you may ignore; blue is the response).

The server window:

```
% ./my_server /home
```

The client window with utility `nc`:

```
% nc localhost 8000

GET /data.html HTTP/1.0
Host: localhost:8000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:72.0) Gecko/20100101
Firefox/72.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
```

```
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1

HTTP/1.0 200 OK
Content-Length:13

line 1
line 2
```

The size and content of the file `/home/data.html` are shown above.

Your web server should perform an infinite loop of waiting for and processing requests.

Your web server should launch a **new thread** for each request. You should use POSIX **system calls for files** (not `fopen()`, `fclose()`, etc.) and `pthreads` for threading.

Your code (without comments) is expected to be 150 lines.

SUBMISSION, GRADING, AND ACADEMIC INTEGRITY

The project must be submitted on Blackboard. You have three attempts; please read *Programming Assignments Requirements and Recommendations* on Blackboard for suggested use of the attempts and **submission** package.

Please read *Programming Assignments Requirements and Recommendations* on Blackboard for the **grading** rubric.

Please read *Programming Assignments Requirements and Recommendations* on Blackboard for a strong warning on **cheating**.

GRADING NOTES

The program will be graded by TA. The total grade for this assignment is 100 points, with

- 85 points as described in (1) - (4) of Grading in *Programming Assignments Requirements and Recommendations*. Please note that in (1) the following will be considered:
    - correct program without threads – minus 10 points,
    - correct program with C standard library functions instead of system calls for file handling – minus 10 points
- 15 points for teamwork – reasonable and documented load distribution.