CME 3205 Operating Systems

Assignment #2

Multi-threaded HTTP Server

Due date : 9 December 2018, 11:59 pm (No extension, please do not request!!)

Task

In this assignment, you are expected to write a small web server in c that supports a subset of the HTTP 1.0 specifications. The server;

- should be able to handle simultaneous requests
- implement the HTTP methods GET and HEAD
- handle and respond to invalid requests.

You should be able to demonstrate that your Web server is capable of delivering your home page to a Web browser. You should implement version 1.0 of HTTP, as defined in RFC 1945, where separate HTTP requests are sent for each component of the Web page. The server will be able to handle multiple simultaneous service requests in parallel. This means that the Web server is multi-threaded. In the main thread, the server listens to a fixed port. When it receives a TCP connection request, it sets up a TCP connection through another port and services the request in a separate thread.

To simplify this programming task, we will develop the code in two stages. In the first stage, you can write a multi-threaded server that simply displays the contents of the HTTP request message that it receives. After this program is running properly, you can add the code required to generate an appropriate response.

As you are developing the code, you can test your server from a Web browser, such as Chrome, Safari, Firefox web browser as http client application. But remember that you are not serving through the standard port 80, so you need to specify the port number within the URL that you give to your browser. For example, if your machine's name is **testhost.mydomain.com** your server is listening to port **6789**, and you want to retrieve the file **index.html**, then you would specify the following URL within the browser:

http:// testhost.mydomain.com:6789/index.html

If you omit ":6789", the browser will assume port 80 which most likely will not have a server listening on it.

When the server encounters an error, it sends a response message with the appropriate HTML source so that the error information is displayed in the browser window.

The Figure-1 shows the basic communications between server and client.

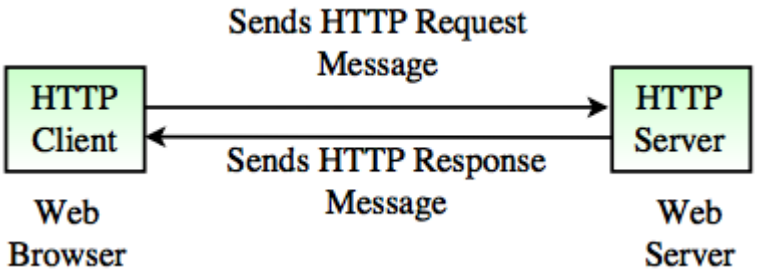
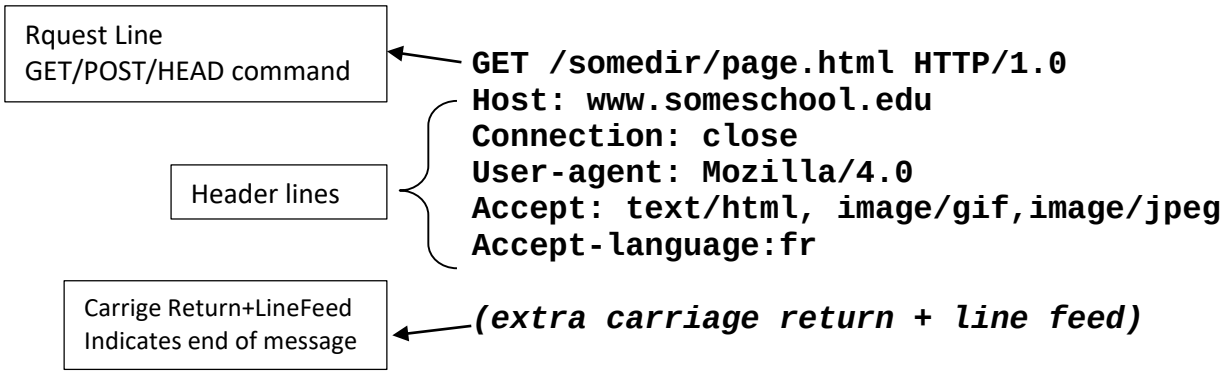


Figure-1: Http server and client communication

A typical Http message sent from server to client or vice versa, is as follows:



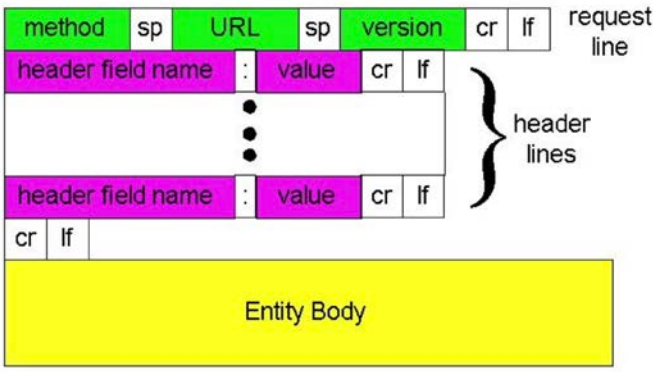


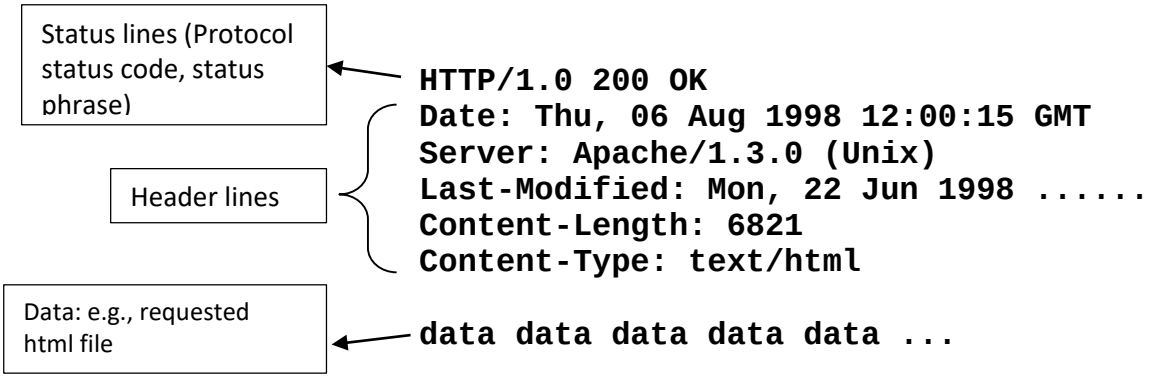
Figure-2: General format of Http request message.

Figure-3: An example of Http request message.

```
GET /2014/07/up-and-running-with-spring-framework-4.html HTTP/1.1
User-Agent: curl/7.34.0
Host: www.sanjaypatel.name
Accept: */*
```

Figure-4: An example of Http request message.

A typical example of a response message is as follows:



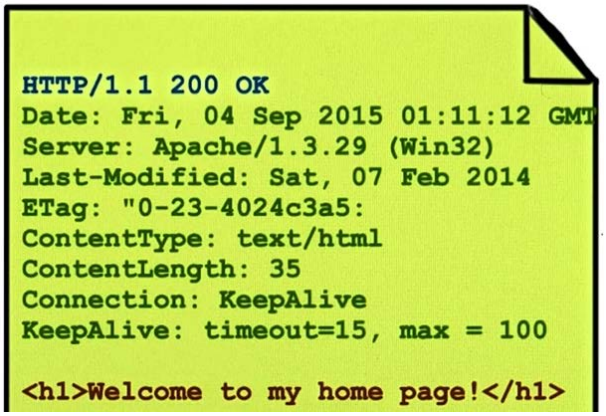


Figure-5: An example of Http response message.

```
< HTTP/1.1 200 OK
< Content-Type: text/html; charset=UTF-8
< Expires: Fri, 22 Aug 2014 10:37:28 GMT
< Date: Fri, 22 Aug 2014 10:37:28 GMT
< Cache-Control: private, max-age=0
< Last-Modified: Tue, 12 Aug 2014 10:52:53 GMT
```

Figure-6: An example of Http response message.

First line in response message from server to client includes server status code. Some of status codes and their explanations are as follows:

200 OK
request succeeded, requested object later in this message

301 Moved Permanently
requested object moved, new location specified later in this message (Location:)

400 Bad Request

request message not understood by server

404 Not Found
requested document not found on this server

General Requirements

- For this assignment you will work individual.
- The POSIX library (pthread) will be used.
- We compile your code on Debian 8 OS with this code:
- Server takes too many requests but it can return response only 10 requests. More than 10 requests are refused. In this state, server returns a "Server is busy" message
- Server accept the requests that will be html and jpeg files. The other types will be not accepting.
- You only upload a single c file.
- We compile your solution using the following code.
- gcc StudentNumber.c –o StudentNumber –lpthread

Submission

Submission will be via Classroom.

- Name your **code file** as: **StudentNumber.c** (do not use another naming convention.) If you don't follow the naming rules, a penalty applies **(15 pts)**
- Late submission is not allowed.

Honesty

Your submissions will be scanned among each other as well as the Internet repository. Any assignments that are over the similarity threshold of a system for Detecting Software Similarity will get 0. We strongly encourage you not to submit your assignment rather than a dishonest submission.

For Questions

For any questions about the assignment please use Classroom systems comments under Assignment announcement. Before asking your question, please check carefully previous questions and answers, where similar questions were already asked by someone else already answered.
No private questions via email will be answered!!! We will try to answer any of your questions as soon as possible, except the ones "Hocam my code does not work, can you fix it" or "I have implemented it but it does not work, can you look at it". Debuggers are far more suitable options.

Good luck !!!