

CS 4453 Assignment 2

Due on Thursday, Feb. 12, 2015

Each student is required to do this assignment **individually** and to hand in the following:

- Provide a hard copy of your complete solutions of Problems, showing all your work and including computer input and outputs, if any (but not the source codes). If you only can finish partially, then you can tell what progress you have made and hand in the incomplete work.

- Write sufficient comments on your source codes and send your source code files (if there are) to

`cs4453@peace.lakeheadu.ca`.

Name, Student Number, Course Number (CS 4453) and your email address should be included.

Assignments which are not met the above requirements will not be marked. The score of the assignment will depend on:

Specification and documentation: 20 %

Correctness: 80 %

Late assignments will be penalized and will not be accepted after 3 days.

Problem 1.

In this problem, you are asked to use `telnet` (an remote access Unix program) to perform some HTTP requests. If you use Linux or Mac OS, then you can use a terminal to run it. If you use windows, then you can install Cygwin and `telnet`. Use `telnet` to perform the following HTTP requests. Note that after the request, you need to press the carriage return twice. Record the message you get and see what status codes you received. Here we just use `telnet` to establish a TCP connection, but not use the whole `telnet` protocol.

1. `telnet peace.lakeheadu.ca 80`

```
GET /~wei/ HTTP/1.1
Host: peace.lakeheadu.ca
Connection: close
```

2. telnet peace.lakeheadu.ca 80

```
HEAD /~wei/cs4476.html HTTP/1.1
Host: peace.lakeheadu.ca
```

3. telnet ccc.cs.lakeheadu.ca 80

```
GET /notexist.html HTTP/1.1
Host: ccc.cs.lakeheadu.ca
```

Problem 2.

Build a one-time simple HTTP server in Java (just use one java class). Here “one-time” means the server only can accept one request, then close.

1. The outline of the server is as follows:

- Create a `ServerSocket` which is listening at port 4453.
- When a client try to establish a connection, the method `accept()` is used to create a socket.
- When a client sends an HTTP request, the server uses `StringTokenizer` to check if the request start with “GET”.
- Then retrieve the file name requested (need to remove “/”, if there is one before the file name).
- Obtain the file and prepare to send it back.
- Send back

```
HTTP/1.0 200 Document Follows\r\n
```

- If the requested file is *.jpg, then send back

```
Content-Type: image/jpeg\r\n}
```

You can also check some of other types if you want.

- Send back the requested file.
- Close the socket.

2. Write a simple `html` file `try.html` such as “Welcome World” page. Put the file at the location where you run your server.

3. Use the web browser to visit: `localhost:4453/try.html`. If everything is OK, the browser will display your page.

Problem 3.

Suppose Bob joins a BitTorrent torrent, but he does not want to upload any data to any other peers (so called free-riding).

- Bob claims that he can receive a complete copy of the file that is shared by the swarm. Is Bob's claim possible? Why or why not?
- Bob further claims that he can further make his "free-riding" more efficient by using a collection of multiple computers (with distinct IP addresses) in the computer lab in his department. How can he do that?

Problem 4.

Still use Wireshark to record HTTP packets when visit `http://ccc.cs.lakeheadu.ca`. Then do some analysis of these data:

- Look at the packets' source and destination and indicate how the web page was downloaded.
- Refresh the web, and look at the packets to see what are the difference between the refresh and the original loading of the web.
- You may check some information from the content of the packets to further understand the packets. Report your analysis.