ELEC 4120: Computer Communication Networks
Prof. Tsang, Danny H. K.
(Fall 2018)

Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology

Programming Assignment 2

Notes:
1) In your source code, please write down proper comments. You may discuss ideas with others in the class, but you should implement your own program. Do not copy from others.

2) Although Python 3 is preferred, you are welcome to use any programming language.

3) The submission contains two steps: source code submission and on-site demonstration. You will get zero point if you fail to complete either of the following two steps.

   a. **Source code submission:** Please compress the source code and executable files into a ZIP file, with your "`student ID-operating system-programming language`", for example, "`20010000-Windows-Python3`", as the file name. Upload the ZIP file to the "programming assignment 2" field, under the "assignments" section of the ELEC4120 Canvas website. The deadline for source code submission is hard at **2 p.m. on December 4, 2018**, with no late submission acceptable.

   b. **On-site demonstration:** You need to demonstrate your program to our TAs face to face, after which you may be asked questions regarding your program. The demonstration sessions will be held during December 4 to 6. Please book a time slot with our TAs in advance through Canvas. See https://community.canvaslms.com/docs/DOC-10580-4212716665 if you do not know how to do this. Please email our TAs if you need special arrangements.

      Meanwhile, if you cannot bring your laptop to the demo venue, please kindly email our TAs in advance, and our TAs will be happy to arrange a machine for you accordingly.

Simple HTTP Client

When your simple HTTP client starts, it should ask the user to input an URL from the keyboard, for example, "`http://www.ece.ust.hk/~eetsang/index.html`" (quotation marks excluded). Then the client extracts the server hostname part from the URL, establishes a socket connection to the HTTP server, uses the `GET` method to fetch the corresponding HTML base file from the server, prints the server-returned HTTP response on the screen, and saves the HTML file locally if and only if the response status code is `200`.

Detailed requirements:
1. You should manually implement the whole procedure, including socket establishment, sending `GET` requests to the server, and receiving the raw data from the server. Every step above counts for some points.
   a) You are not allowed to use or import any pre-implemented modules/packages for HTTP, such as the `http.client` module in Python.
   b) Only the external modules/packages that have been used in the first programming assignment can be imported for this programming assignment (e.g., the `socket` module in Python).
   c) You can still import external modules/packages not for socket programming purposes (e.g., to save files to the operating system).

2. You only need to fetch the user-input HTML base file (not necessarily ending with ".html", such as "`http://www.ece.ust.hk/ece.php`"). Other objects embedded in the base file, such as images, are not required to be downloaded. If the file name is not specified (e.g., "`http://www.ece.ust.hk/`"), the client just reports this to the HTTP server (i.e., "`GET /`" for "`http://www.ece.ust.hk/`").

3. The client should print the entire server-returned HTTP response on the screen, but not the content of the HTML file. Here are two examples:

   [Example 1] The client was requesting "`http://www.ece.ust.hk/~eetsang/index.html`" (quotation marks excluded).
   ```
   HTTP/1.1 200 OK
   Date: Fri, 02 Nov 2018 09:36:21 GMT
   Server: Apache/2.2.25 (Unix)
   Last-Modified: Wed, 28 Mar 2018 07:38:52 GMT
   ETag: "720e48-10c4-568741935bb00"
   Accept-Ranges: bytes
   Content-Length: 4292
   Connection: close
   Content-Type: text/html; charset=UTF-8
   ```

   [Example 2] The client was requesting "`http://www.ece.ust.hk/`" (quotation marks excluded).
   ```
   HTTP/1.1 302 Found
   Date: Fri, 02 Nov 2018 09:30:45 GMT
   Server: Apache/2.2.25 (Unix)
   Location: http://www.ece.ust.hk/
   Content-Length: 287
   Connection: close
   Content-Type: text/html; charset=iso-8859-1
   ```

4. The stored HTML file should be a complete HTML file, where the server-returned HTTP response has been removed.

5. After fetching an HTML file, your client should remain active, waiting for the user to input the next URL. The client exits when the user types "`bye`" (quotation marks excluded).

6. A user may request two HTML files with the same name but from different URLs. For example, a user may sequentially request `http://www.ece.ust.hk/~eetsang/index.html` and `http://www.ece.ust.hk/~eeknlau/index.html`. In this case, the client should not overwrite the first HTML file when saving the second HTML file. (A simple solution is to attach a timestamp to or include the entire URL in the name of your saved file.)

7. A user may omit the "`http://`" part when inputting the URL. In this case, the URL should still be considered as legal.