CSCD 330-040 Computer Networks

Lab 4, HTTP and Sockets - Due, October 17, 2023

October 11, 2023

Overview:

Now that we have gone over sockets and the HTTP protocol let's put them together and create a program similar to `curl` and `wget`. Your client must retrieve a webpage via HTTP. Additionally your program must be able to handle the retrieval of pages that are not the root directory. For example, you must be able to call the API you wrote for Lab 3.

Client specifications:

Your client must take 3 command line inputs.

1. A flag indicating if it should print to standard out (the terminal) or save to file. You may use a default file name.
2. The target port.
3. The URL to retrieve.

Your client MUST be able to access pages that have specified paths. For example, You must be able to call the APIs from the Lab 3 flask example from your program.

You ***MUST*** use the socket library and you may not use any other library to make the connection! You **MAY** use the `urllib.parser` library to parse your URL.

YOUR PROGRAM DOESN'T HAVE TO HANDEL HTTPS!

Most websites are HTTPS so I recommend you test your website on http://httpforever.com . Make sure you retrieve the whole page. You might have to make more than 1 call to `recvfrom()` to get the whole page.

DO NOTs:

- Do not share code with your classmates.
  - This is **NOT** a group assignment.
  - If you need a reminder of the rules, please read the syllabus.

- Do not discuss your implementation at a low level.
  - Only discuss things at a very high level.
  - e.g., I used the function `urlparse` form the urllib library.

Turn in:

Submit a tarball of everything (source code, test files, etc.). `tar` is an archiving and compression tool that is very helpful. To "tar up," all the python files, all the bash files and the README in the directory you are in, you would use the command:

```
tar -czvf lab4_turnin_MYNAME.tar.gz *.sh *.py README
```

I expect a test file/program that runs the program on sample input (3-5). You must submit a script (preferably bash) that uses your program to retrieve pages. If you're testing on the Lab 3 flask example server, you must include it in your tar file and specify that it's used in your README. You must also have a README file (plain text is fine) that describes your program. This would include any packages you had to install to make your program work, the default file name your program saves to etc.

File formats:

Make your final program `lab4.py`. Your bash files should be named `test.sh`. Use the naming convention given in the tar example above in the `Turn in` section (`lab4_turnin_MYNAME.tar.gz`).