CSc361 Fall 2021 Programming Assignment (P1): Simple Web Server (SWS) Specification

Spec out: by Tuesday, September 14, 2021
Code due: by Friday, October 1, 2021, 5pm through bright.uvic.ca (also known as brightspace or bright)

Assignment objective:

In this programming assignment, you will use the STREAM socket (i.e., supported by TCP) in Python to create a Simple Web Server (SWS) with select() supporting both persistent and non-persistent HTTP connections. Please note that you need to use select() non-blocking socket() to complete P1 as a preparation for P2.

Assignment schedule with regard to tutorials and labs:

There are three tutorials (T2, T3 and T4) and two lab sessions (L2 and L3) associated with this assignment.

In T2 on September 17, the tutorial instructor will go through the P1 specification and answer your questions, and will also provide a simple design for your reference. In L2 on September 20, 21 or 22, the lab instructor will go through HTTP (both persistent and non-persistent, using nginx-light and wget) packet capture and analysis, and help students form their design for P1.

In T3 on September 24, the tutorial instructor will check students P1 design, provide feedback and instruction on submission through brightspace, and give a quick demo. In L3 on September 27, 28 or 29, the lab instructor will check students P1 implementation, capture and analyze SWS packets, and provide help if needed.

In T4 on October 1, the tutorial instructor will provide some last-minute reminders and help.

Please follow our tutorial and lab schedule closely for this assignment, which ensures its success and smoothness.

SWS requirements:

SWS only supports the “GET /filename HTTP/1.0” command, and “Connection: keep-alive” and “Connection: close” request and response header when supporting persistent HTTP connection. The request header is terminated by an empty line known as “\r\n”, where “\r” indicates a carriage return and “\n” a (new) line feed.

If unsupported commands are received or in unrecognized format, SWS will respond “HTTP/1.0 400 Bad Request”. If the file indicated by filename is inaccessible, SWS will return “HTTP/1.0 404 Not Found”. Such responses will be followed by an empty line, indicating the end of the response.

For successful requests, SWS will respond “HTTP/1.0 200 OK”, followed by the response header if any, an empty line indicating the end of the response header, and the content of the file.

How to run SWS:

On H2 in PicoNet, “python3 sws.py ip_address port_number”, where ip_address and port_number indicate where SWS binds its socket for incoming requests.

On H1 in PicoNet, “nc sws_ip_address sws_port_number” to connect to SWS, and type “GET /sws.py HTTP/1.0” followed by “Connection: keep-alive” and an empty line to request the file sws.py from SWS (in this case, SWS shall keep the connection alive after sending back sws.py following an empty line after “Connection: keep-alive”, and wait for the next request from the same client through the same TCP connection, until the connection times out, i.e., “Connection: close”). If the client does not include “Connection: keep-alive” or does include “Connection: close” in its request, SWS will close the connection after serving the request.

For each served request, even if unsuccessfully, SWS will output a log line “time: client_ip:client_port request; response”, e.g., “Wed Sep 15 21:44:35 PDT 2021: 192.168.1.100:54321 GET /sws.py HTTP/1.0; HTTP/1.0 200 OK”. Please follow the log format closely for marking purposes. Please note that SWS will keep waiting to serve more clients and can serve multiple concurrent clients by using select(), until interrupted by Ctrl-C.

How to test SWS:

Capture and analyze the interaction between SWS and its clients (nc, wget or even a regular web browser) with tcpdump and Wireshark. Your code will be evaluated in PicoLab as you have in ECS360.

What to submit:

sws.py source file, and the tcpdump files on R, showing the interaction between SWS and its clients, in both persistent (sws-persistent.cap) and non-persistent (sws-non-persistent.cap) connections. Please also copy and paste the content of sws.py and tcpdump -r the capture file to the text input box on brightspace.

When:

By Friday, October 1, 2021, 5pm through brightspace -> assignments -> p1

Questions and answers:

In addition to associated tutorials and labs, brightspace -> discussion forums -> p1

Academic integrity:

This is an individual assignment and your submitted work shall be done by yourself alone.

Append: A simple design for your reference