Project 1: HTTP-GET using BSD Sockets Due date: February 8, 2021 at 11:59 p.m. PST

This project MUST be completed individually.

REVISION HISTORY

- January 26, 2021 - Added requirement for Host header field to be included.
- January 26, 2021 - Clarified that 1 should be returned when there is a socket error.

1. PROJECT GOALS

In this project, you will develop a simple C++ application that connects to a web server over HTTP, performs a simple request, and outputs the HTTP response received from the host. As part of this, we will take the chance to learn a bit more about how web browsers and servers work behind the scenes. Please note that your program will not work with HTTPS URLs, since these require the use TLS, which we will not require you to implement.

2. PROGRAMMING ENVIRONMENT

You should test and develop your code on the SEASnet system, as your submission will be compiled and graded on this environment. However, since C++ is a generally cross-platform language across Unix-like systems, you should be able to compile and run your code on any modern Unix-like machine (e.g., Linux, macOS, or *BSD) that has a C++ compiler.

On SEASnet, you can use Vim, Emacs, or Nano to edit your source code and "g++" to build it. However, you must submit a Makefile that can be used to compile your code into an executable named "http-get".

3. INSTRUCTIONS

(1) Read Chapter 2 of the textbook carefully. This will help familiarize you with the HTTP protocol. You should also consider reviewing the lecture and discussion slides pertaining to HTTP and socket programming.

(2) If you haven't already done so, familiarize yourself with the basics of BSD sockets and programming in C/C++. An overview of BSD sockets can be found here: https://en.wikipedia.org/wiki/Berkeley_sockets

Please use `getaddrinfo()` to resolve domain names into IP address. Instructions on how to use this function can be found at: https://en.wikipedia.org/wiki/Getaddrinfo

(3) Create a program that named `http-get` that accepts an arbitrary URL (formatted like the following without surrounding quotes: "http://catalog.registrar.ucla.edu/ucla-catalog20-21-2.html") as an argument, establishes a connection to the server hosting this URL, sends a properly formatted HTTP GET request for this URL to the server, receives an HTTP response, closes the connection to the server, and then outputs this raw response to stdout.

Requirements:

- The program must accept a URL as argument provided in the format `-a URL`.
- If the URL is incorrectly formatted, the server name cannot be resolved via `getaddrinfo()`, or there is some socket error, your program must exit with return code 1 and print nothing to stdout.
- If the `-a URL` argument is not specified, your program must exit with return code 2 and print nothing to stdout.

- The HTTP request you send should specify HTTP version 1.1 and only include two header fields: "User-Agent", which should contain the value "http-get 1.0" and "Host", which should contain the host name of the website (e.g., "cs.ucla.edu").
- If the request succeeds, your program must exit with return code 0.
- If the `-c` argument is specified to `http-get`, your program must output the generated HTTP GET request to stdout and then exit with return code 0 (terminating before it would establish a connection to the server and send the HTTP request).

- You are **NOT** allowed to use any abstractions or libraries above BSD sockets, such as Boost.Asio.

We strongly recommend you use `getopt()` to parse command line options.

4. Grading Criteria

This is an **individual project**, meaning that no collaboration is allowed. You are allowed to use online resources to understand how to use the BSD sockets library; however, you must not copy code from the Internet and must credit any resources used in comments contained in your source code.

Your code will be graded based upon the following criteria:

- Whether your Makefile compiles your code properly on the SEASnet systems.
- Whether you included a file named "README" that contains your name on the first line and your UID on the second line.
- Whether your program runs on SEASnet without any errors or segfaults.
- Whether your program outputs a correctly formatted HTTP GET *request* and exits with return code 0 when run with the `-c` argument.
- Whether your program exits with return code 1 when provided with an incorrectly formatted URL.
- Whether your program exits with return code 2 when run without the `-a URL` argument.
- Whether your program correctly uses the BSD sockets API to retrieve the specified URL (when run without `-c`), prints the response to the command line, and then exits with return code 0.

5. Project Submission

Put all your files into a directory and compress the contents of this directory into a file named "UID.tar.gz" (replacing UID with your UCLA ID). You **MUST** put all your files directly at the root of this archive (and not inside a directory) to ensure your code is graded properly.

Please submit your project via CCLE – submissions via any other method will not be accepted.

Your submission should include the following:

- Your source code.
- A Makefile that builds your code into an executable named `http-get` in the same directory when when one types "make".
- A file named "README" containing your name on the first line and your UID on the second line.

Acknowledgements

This project was adapted and modified from an earlier instance of CS 118 at UCLA taught by Prof. Songwu Lu.