Introduction to Computer Networks
Lab 2: Linux Socket Programming

1. Description

Write a program that can display all the hyperlinks on a given web page. The primary objective is to learn socket programming and familiarize yourself with the HTTP protocol.

2. Requirements

(a) You are required to implement this project using C (not C++). Other languages (such as Python) are not allowed in this lab because they offer numerous off-the-shelf libraries, which would undermine the purpose of practicing socket programming.

(b) **Input**: The URL of a webpage without “http://”. Take http://can.cs.nthu.edu.tw/index.php as an example; the input is can.cs.nthu.edu.tw/index.php.
Note that both can.cs.nthu.edu.tw/ and can.cs.nthu.edu.tw are valid URLs. The first step to do is parsing the input URL into hostname and path. What follows are some examples:

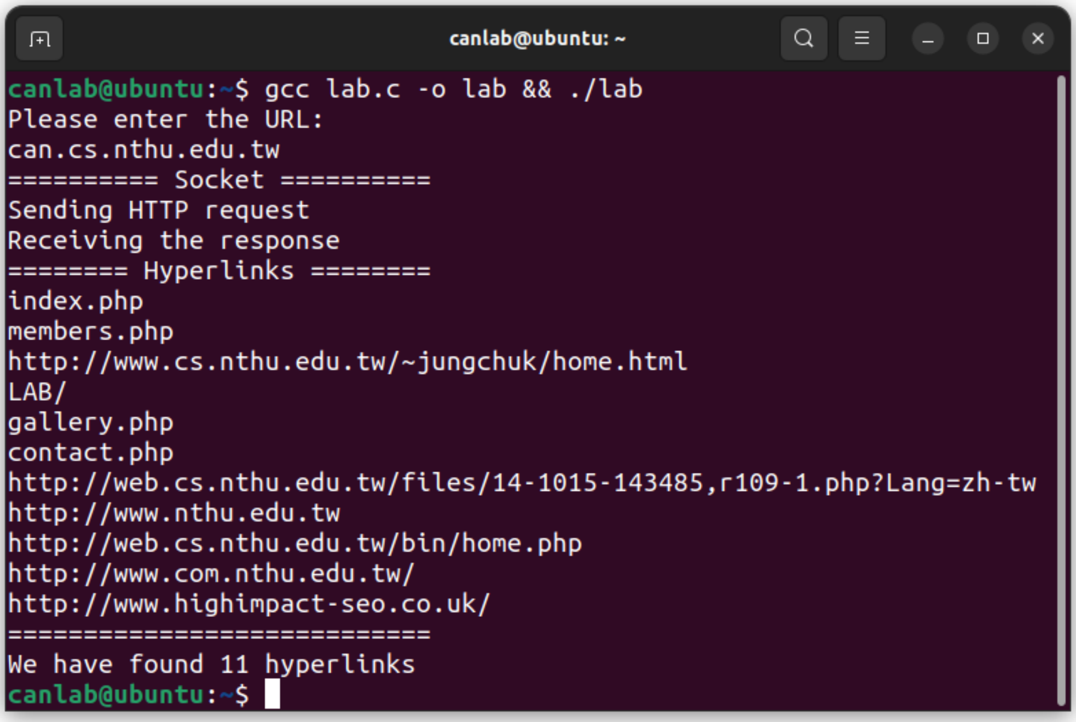
Input	Hostname	Path
`can.cs.nthu.edu.tw`	`can.cs.nthu.edu.tw`	`/`
`can.cs.nthu.edu.tw/`	`can.cs.nthu.edu.tw`	`/`
`can.cs.nthu.edu.tw/index.php`	`can.cs.nthu.edu.tw`	`/index.php`
`can.cs.nthu.edu.tw/contact.php`	`can.cs.nthu.edu.tw`	`/contact.php`
`www.cs.nthu.edu.tw/~jungchuk/home.html`	`www.cs.nthu.edu.tw`	`/~jungchuk/home.html`

Since most websites are now using the HTTPS (https://) protocol, you need to be aware of whether or not the website you test your program supports HTTP (http://).

(c) **Output**: Print all hyperlinks and the total number of hyperlinks in the given webpage. Note that only the hyperlinks in the format of `<a … href="abc" …></a>` should be printed out and counted. Do not print/count other formats such as `<link … href="abc" …>`.

(d) Due to differences in system environments, compiler versions, and settings, the code that compiles and runs on your computer may not work on the TAs’ systems. To ensure fairness in grading, we provide a virtual machine (refer to “Linux Socket Tutorial”). The code you write will be compiled and graded within the provided virtual machine. Please test your code using the virtual machine before your final submission.

3. Example



```
canlab@ubuntu:~$ gcc lab.c -o lab && ./lab
Please enter the URL:
can.cs.nthu.edu.tw
========== Socket ==========
Sending HTTP request
Receiving the response
======== Hyperlinks ========
index.php
members.php
http://www.cs.nthu.edu.tw/~jungchuk/home.html
LAB/
gallery.php
contact.php
http://web.cs.nthu.edu.tw/files/14-1015-143485,r109-1.php?Lang=zh-tw
http://www.nthu.edu.tw
http://web.cs.nthu.edu.tw/bin/home.php
http://www.com.nthu.edu.tw/
http://www.highimpact-seo.co.uk/
============================
We have found 11 hyperlinks
canlab@ubuntu:~$
```

4. Hint:

(a) Refer to “Linux Socket Tutorial” for information about socket programming on Linux.

(b) Establish a connection with the web server using the given hostname.

- Convert the hostname (e.g., can.cs.nthu.edu.tw) into its corresponding IP address.
- Set the appropriate protocol. (e.g., use TCP for HTTP)

(c) Get the HTML source code of the web page.

- Send an HTTP GET request message to the server.
  `GET /{path} HTTP/1.1\r\nHost: {hostname}\r\nConnection: close\r\n\r\n`
- Receive the response from the server.
- Note that calling “`recv()`” once normally returns the data available at that time moment (up to some amount), rather than returning the entire set of requested data. You can either use a loop or set a certain “flag” argument for the `recv()` to handle this problem. (reference: `recv()`)

(d) Show all hyperlinks in the web page.

- `<a … href="this-is-a-hyperlink" …></a>`
- Attributes (`class`, `style`, `href`, etc.) of HTML elements can occur in an arbitrary order. As long as an anchor “`<a>`” element contains “`href`”, it signifies a hyperlink. So, a string that starts with “`<a`”, ends with “`>`”, and contains the attribute “`href`” is a hyperlink you should print.

5. Submission

(a) Please provide a readme.pdf file to show what functionalities your program has.

- For example, is it able to be compiled by gcc? Does it meet all requirements? How do you handle the response and extract all the hyperlinks? What have you learned from this lab?
- If you can run your C program, please provide a screenshot to show how it works (similarly to our example mentioned above).

(b) Compress the C source file(s) and related files (including readme.pdf) into studentID_lab2.zip (e.g., 111012345_lab2.zip).

```
├── lab.c
├── (my_module.h / my_module.c, if applicable)
└── readme.pdf
```

(c) Upload your zip file to eeclass.

(d) Discussion is encouraged; however, plagiarism is not allowed. We will use tools like Moss for similarity comparison. If plagiarism is detected, 0 points will be given.

(e) If you have referred to any books or online materials, please indicate the source in the readme.pdf to avoid from being mistaken for plagiarism. For example, you can add a “Reference” section:

Reference
[1] *How to do socket programming in C*, https://example.com/
[2] …

(f) Submit your assignment by the deadline. Late submissions will not be accepted, and a score of 0 will be assigned.