Communication and Networking Lab

Hw_2 Report

Name: 林奕為

Student's Id: 110062271

Department: CS

Contents

Abstract:

I implemented all the requirements mentioned in the specification.

The program can be successfully compiled and run on a Linux OS.

It first converts the input host name to Ipv4 address using gethostbyname()

```
struct hostent *hostent = gethostbyname(host);
```

Also, I also check whether the input path is empty, if it's empty, the default value
Is " / "

```
if(t_path) sprintf(path, "/%s", t_path);
    else sprintf(path, "/");  // path is empty
```

Next it sends the request to the server

```
const char *getRequest = "GET %s HTTP/1.1\r\nHost: %s\r\nConnection:
close\r\n\r\n";
    char request[MAX_BUFFER_SIZE];
    sprintf(request, getRequest, path, host);
    send(sockfd, request, strlen(request), 0);
```

Then it receives the responses from the HTTP server, and call printLink() function to
parse the message(which is stored in buffer) pass in while the number of bytes read
in > 0.

Below is the source code about the part dealing with continuously reading in
message:

```
char buffer[MAX_BUFFER_SIZE];
    ssize_t bytesRead;

    while ((bytesRead = recv(sockfd, buffer, sizeof(buffer) - 1, 0)) >
0) {
        buffer[bytesRead] = '\0';
        print_Link(buffer);
    }
```

The printLink function utilizes C library functions such as strstr(), strchr() to determine the value of the hyperlink.

It calls the while loop, first examine if there's still exist <a> tag in the remaining message, next the position of *pos* and *end* will be determined using strchr(), if there's exist href which denotes an instance of hyperlink using strstr(), the *pos* and *end* will represent the starting and ending position of the hyperlink's value respectively.

Below is the source code of print_Link() function:

```c
void printLink(char *buffer) {
    char *pos = buffer;
    while ((pos = strstr(pos, "<a")) != NULL) {
        pos = strstr(pos, "href");
        if (pos == NULL) {
            continue;
        }

        pos = strchr(pos, '"');
        if (pos == NULL) {
            continue;
        }

        pos++;
        char *end = strchr(pos, '"');
        if (end == NULL) {
            continue;
        }

        int len = end - pos;
        char link[MAX_BUFFER_SIZE];
        strncpy(link, pos, len);
        link[len] = '\0';
        printf("Link_num: %d\n%s\n", link_num+1, link);

        link_num++;
        pos = end + 1;
    }
}
```
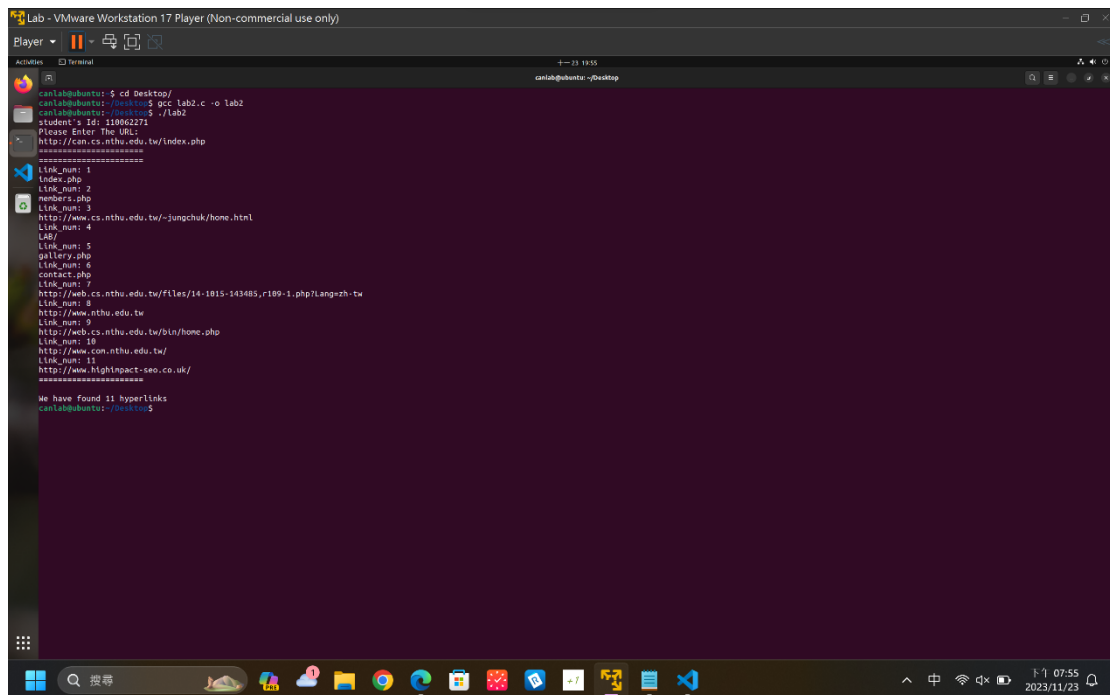
What I have learned in this lab:

I learn about the basic knowledge of socket programming; I also implement a simple program to print all the hyperlinks and the number of that in the HTTP code of a website.

It is a different experience honestly, that my code can deal with real-time data which come from Internet, most of the time my code only runs on my local machine.

Execution Result:

Referenced

[C Program to display hostname and IP address - GeeksforGeeks](#)

Which I used to search for the method of converting the host name to Ip address.

[strstr() in C/C++ - GeeksforGeeks](#)
[strchr() function in C++ and its applications - GeeksforGeeks](#)

used for parsing the received message.

[strtok() and strtok_r() functions in C with examples - GeeksforGeeks](#)

parsing input web URL.

Other references:
Linux Socket Tutorial by NTHU for some basic knowledge i.e. how to create a socket and how to connect it to the server, send(not used) and receive message.