Introduction to Computer Networks - Lab 1

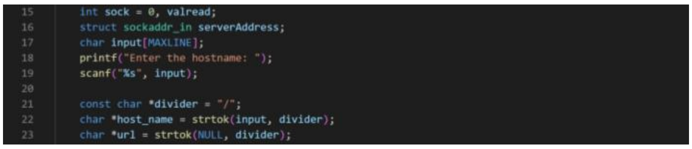
Name: Pei Chi, Huang

Student ID: 108020017

Department: Interdisciplinary Program of Science

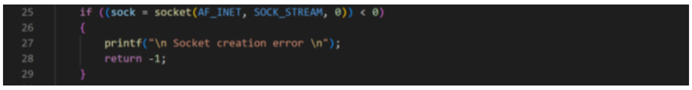
---

1. First, require the user to input the URL of the desired webpage. Then separate the input string by a "/", where the former one is the host name.



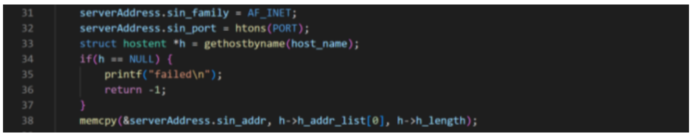
```c
int sock = 0, valread;
struct sockaddr_in serverAddress;
char input[MAXLINE];
printf("Enter the hostname: ");
scanf("%s", input);

const char *divider = "/";
char *host_name = strtok(input, divider);
char *url = strtok(NULL, divider);
```

2. Create a TCP socket. If the socket creation failed, then return -1.



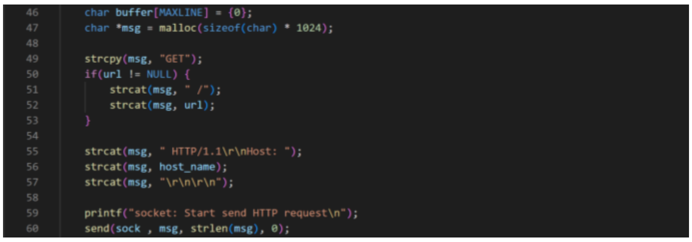
```c
if ((sock = socket(AF_INET, SOCK_STREAM, 0)) < 0)
{
    printf("\n Socket creation error \n");
    return -1;
}
```

3. Set the server address with domain name instead of IP. The port number is set to 80 which is the default one for HTTP server.



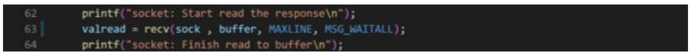
```c
serverAddress.sin_family = AF_INET;
serverAddress.sin_port = htons(PORT);
struct hostent *h = gethostbyname(host_name);
if(h == NULL) {
    printf("failed\n");
    return -1;
}
memcpy(&serverAddress.sin_addr, h->h_addr_list[0], h->h_length);
```

4. Process the request message and send it to the socket.



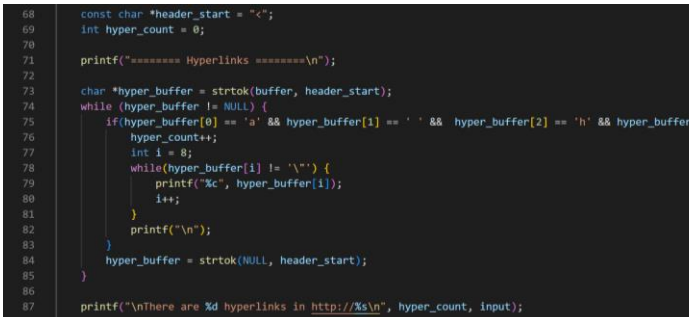
```c
char buffer[MAXLINE] = {0};
char *msg = malloc(sizeof(char) * 1024);

strcpy(msg, "GET");
if(url != NULL) {
    strcat(msg, " /");
    strcat(msg, url);
}

strcat(msg, " HTTP/1.1\r\nHost: ");
strcat(msg, host_name);
strcat(msg, "\r\n\r\n");

printf("socket: Start send HTTP request\n");
send(sock , msg, strlen(msg), 0);
```

5. Read message sent from the socket, and save it to a buffer.



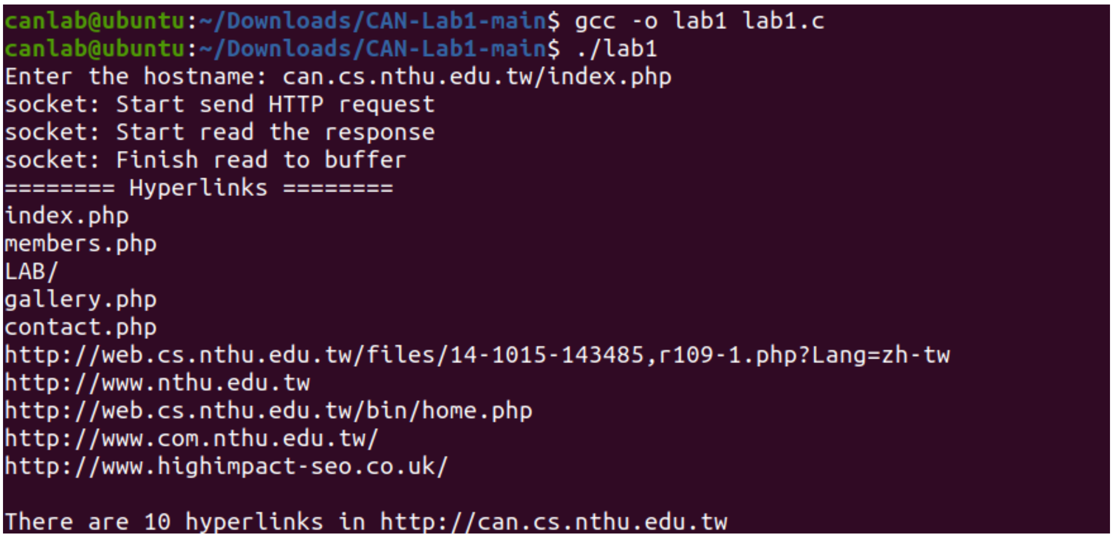
```c
printf("socket: Start read the response\n");
valread = recv(sock , buffer, MAXLINE, MSG_WAITALL);
printf("socket: Finish read to buffer\n");
```

6. Parse the message saved in the buffer. Since each hyperlink in the response message is in the form `<a href="XXX">`. I manually classified every tag starts with `a href`. If any hyperlink is identified, print out the URL inside the quotation marks and increase the count of hyperlinks by 1. In the end, printed out the total number of hyperlinks in the given URL.



```c
const char *header_start = "<";
int hyper_count = 0;

printf("======== Hyperlinks ========\n");

char *hyper_buffer = strtok(buffer, header_start);
while (hyper_buffer != NULL) {
    if(hyper_buffer[0] == 'a' && hyper_buffer[1] == ' ' &&  hyper_buffer[2] == 'h' && hyper_buffer
        hyper_count++;
        int i = 8;
        while(hyper_buffer[i] != '\"') {
            printf("%c", hyper_buffer[i]);
            i++;
        }
        printf("\n");
    }
    hyper_buffer = strtok(NULL, header_start);
}

printf("\nThere are %d hyperlinks in http://%s\n", hyper_count, input);
```

Compile with gcc under vmware environment. The result was shown below:



```
canlab@ubuntu:~/Downloads/CAN-Lab1-main$ gcc -o lab1 lab1.c
canlab@ubuntu:~/Downloads/CAN-Lab1-main$ ./lab1
Enter the hostname: can.cs.nthu.edu.tw/index.php
socket: Start send HTTP request
socket: Start read the response
socket: Finish read to buffer
======== Hyperlinks ========
index.php
members.php
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/

There are 10 hyperlinks in http://can.cs.nthu.edu.tw
```