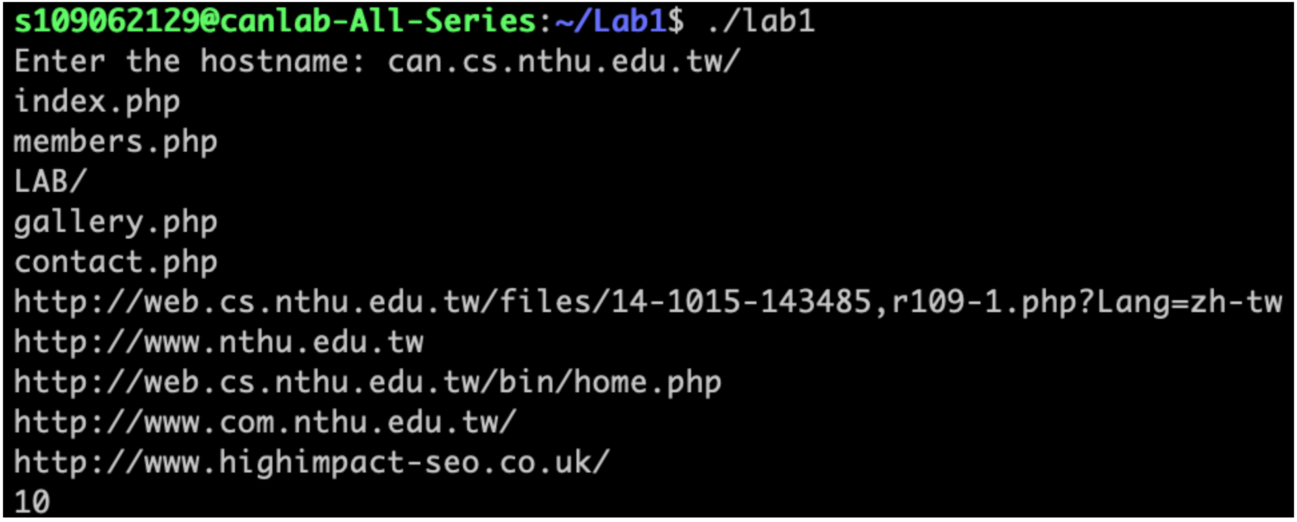
Lab 1

學號: 109062129　　　　　　　　姓名: 林奕廷

1. Description

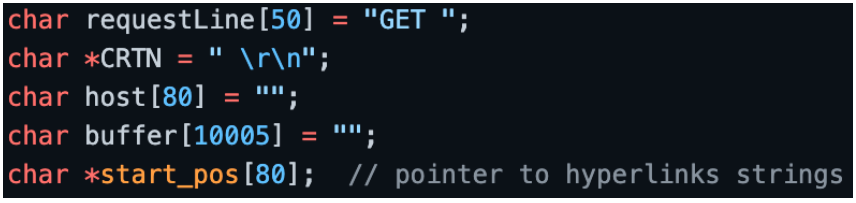
   This socket programming C program can be compiled by gcc, run under Linux machine, and it meets all the requirements. To test the program, simply input the URL of a webpage, and the program will print out all hyperlinks on the given webpage, and count the number of links.

2. Example



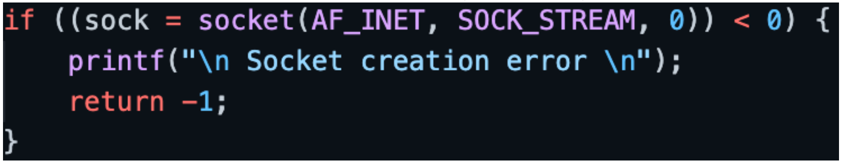
```
s109062129@canlab-All-Series:~/Lab1$ ./lab1
Enter the hostname: can.cs.nthu.edu.tw/
index.php
members.php
LAB/
gallery.php
contact.php
http://web.cs.nthu.edu.tw/files/14-1015-143485,r109-1.php?Lang=zh-tw
http://www.nthu.edu.tw
http://web.cs.nthu.edu.tw/bin/home.php
http://www.com.nthu.edu.tw/
http://www.highimpact-seo.co.uk/
10
```

3. Code snippets



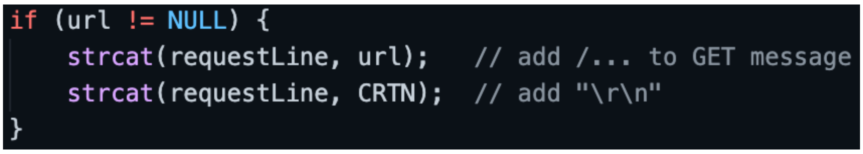
```c
char requestLine[50] = "GET ";
char *CRTN = " \r\n";
char host[80] = "";
char buffer[10005] = "";
char *start_pos[80];  // pointer to hyperlinks strings
```

Char initialization, and I use an array of char pointers to store locations of hyperlinks in buffer string.



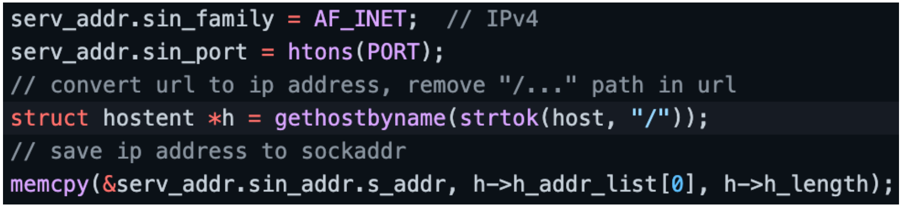
```c
if ((sock = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
    printf("\n Socket creation error \n");
    return -1;
}
```

Socket creation.



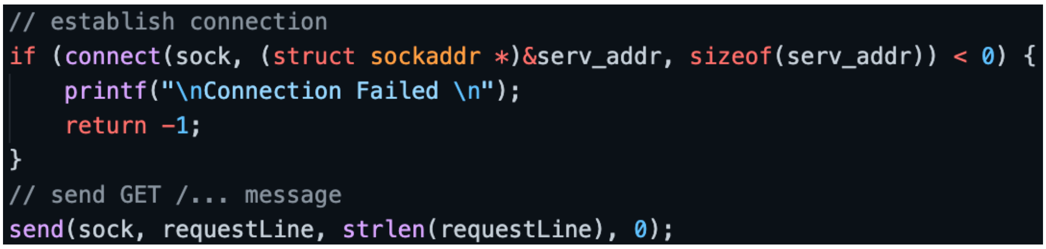
```c
if (url != NULL) {
    strcat(requestLine, url);   // add /... to GET message
    strcat(requestLine, CRTN);  // add "\r\n"
}
```

Complete request message to be sent.



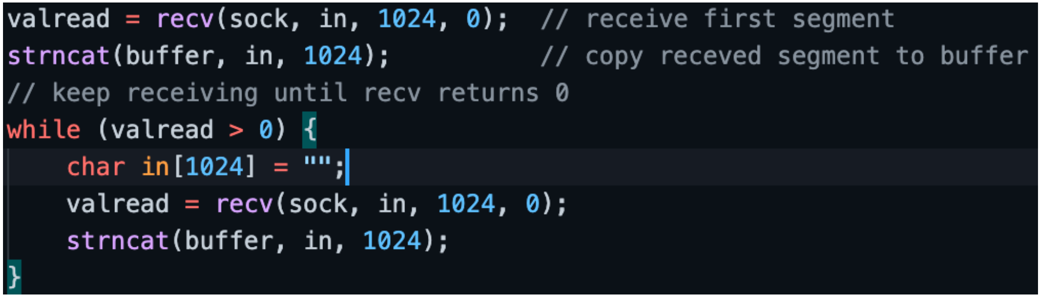
```c
serv_addr.sin_family = AF_INET;  // IPv4
serv_addr.sin_port = htons(PORT);
// convert url to ip address, remove "/..." path in url
struct hostent *h = gethostbyname(strtok(host, "/"));
// save ip address to sockaddr
memcpy(&serv_addr.sin_addr.s_addr, h->h_addr_list[0], h->h_length);
```

Convert url to ip address via gethostbyname function, and save it to struct serv_addr.



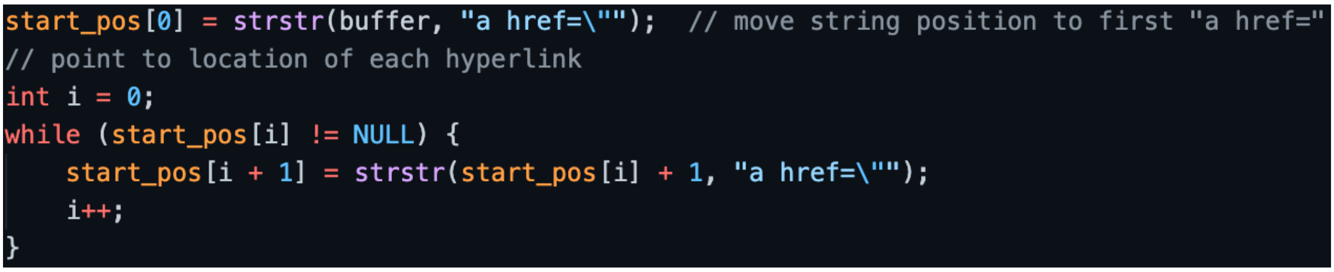
```
// establish connection
if (connect(sock, (struct sockaddr *)&serv_addr, sizeof(serv_addr)) < 0) {
    printf("\nConnection Failed \n");
    return -1;
}
// send GET /... message
send(sock, requestLine, strlen(requestLine), 0);
```

Establish connection, and send request message socket.



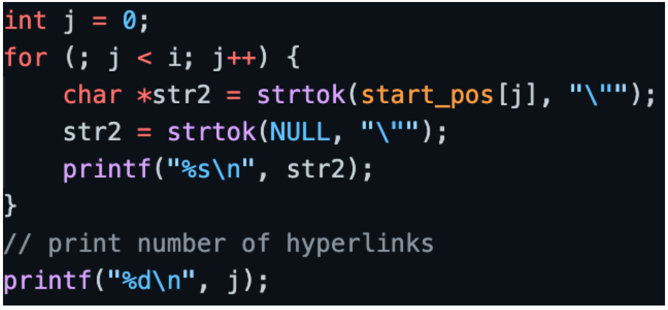
```
valread = recv(sock, in, 1024, 0);  // receive first segment
strncat(buffer, in, 1024);          // copy receved segment to buffer
// keep receiving until recv returns 0
while (valread > 0) {
    char in[1024] = "";
    valread = recv(sock, in, 1024, 0);
    strncat(buffer, in, 1024);
}
```

Read in received segments and put them together into buffer string. Here I use valread > 0 as the condition for this while loop, as recv returns 0 when there are nothing else to be received, and returns -1 if there are errors.



```
start_pos[0] = strstr(buffer, "a href=\"");  // move string position to first "a href="
// point to location of each hyperlink
int i = 0;
while (start_pos[i] != NULL) {
    start_pos[i + 1] = strstr(start_pos[i] + 1, "a href=\"");
    i++;
}
```

Use strstr function in C library to locate every position in buffer that matches “a href=”, and store the position in char pointer of array afterwards.



```
int j = 0;
for (; j < i; j++) {
    char *str2 = strtok(start_pos[j], "\"");
    str2 = strtok(NULL, "\"");
    printf("%s\n", str2);
}
// print number of hyperlinks
printf("%d\n", j);
```

Use strtok function to get start and end positions of each hyperlink. For example, if the current line is “a href="index.php">Home</a></div>”, after calling this function for the first time in loop, str2 points to “a href=”. After calling the function one more time, char 2 points to “index.php”, which is exactly the result we wanted. Finally, output j, the total number of hyperlinks.