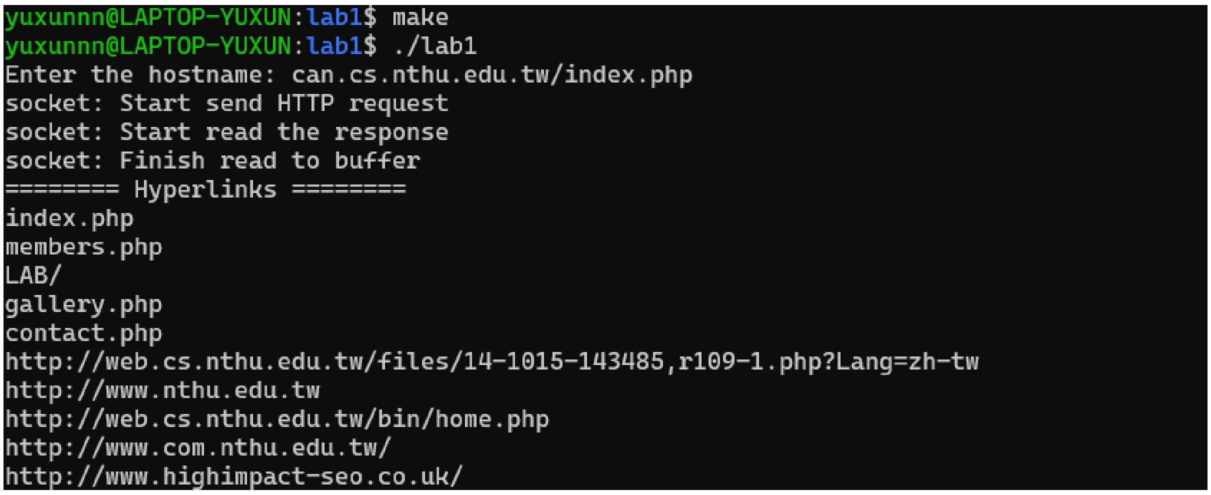
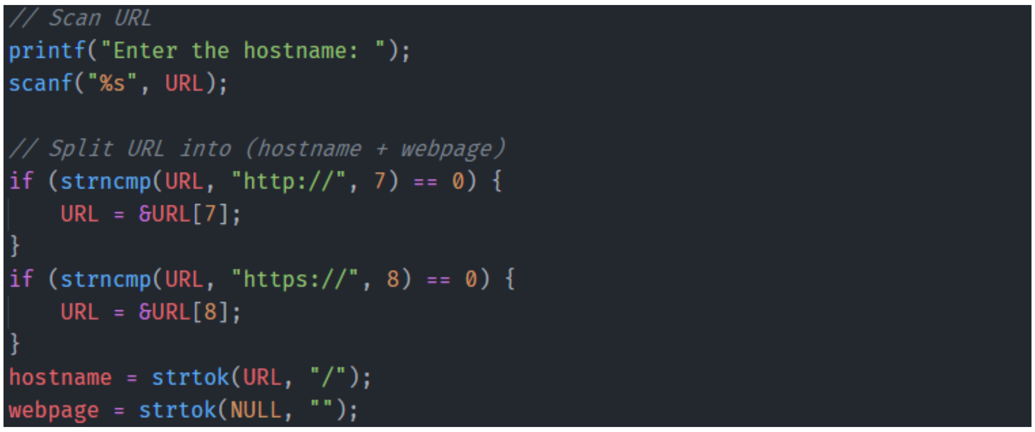


```
yuxunnn@LAPTOP-YUXUN:lab1$ make
yuxunnn@LAPTOP-YUXUN:lab1$ ./lab1
Enter the hostname: can.cs.nthu.edu.tw/index.php
socket: Start send HTTP request
socket: Start read the response
socket: Finish read to buffer
======== Hyperlinks ========
index.php
members.php
LAB/
gallery.php
contact.php
http://web.cs.nthu.edu.tw/files/14-1015-143485,r109-1.php?Lang=zh-tw
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/
```

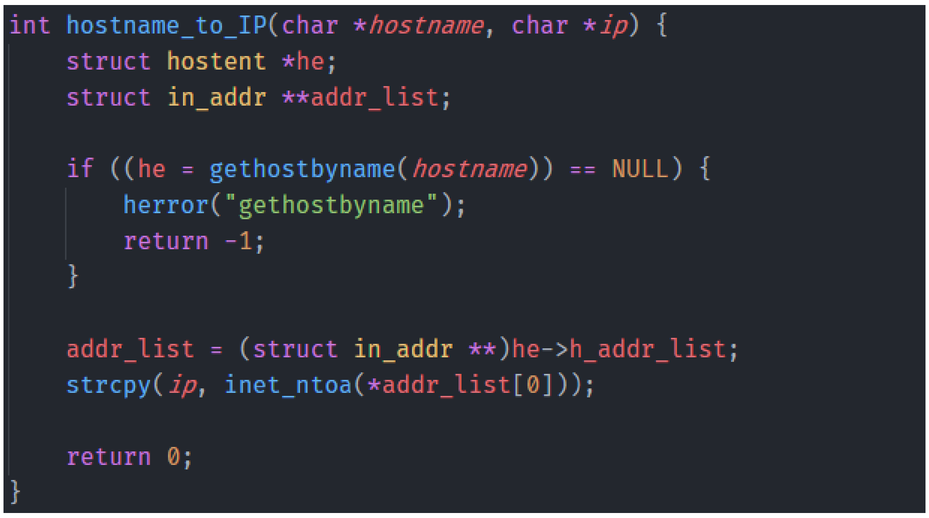
1. 分割讀取的 URL 為 hostname 和 webpage



```c
// Scan URL
printf("Enter the hostname: ");
scanf("%s", URL);

// Split URL into (hostname + webpage)
if (strncmp(URL, "http://", 7) == 0) {
    URL = &URL[7];
}
if (strncmp(URL, "https://", 8) == 0) {
    URL = &URL[8];
}
hostname = strtok(URL, "/");
webpage = strtok(NULL, "");
```

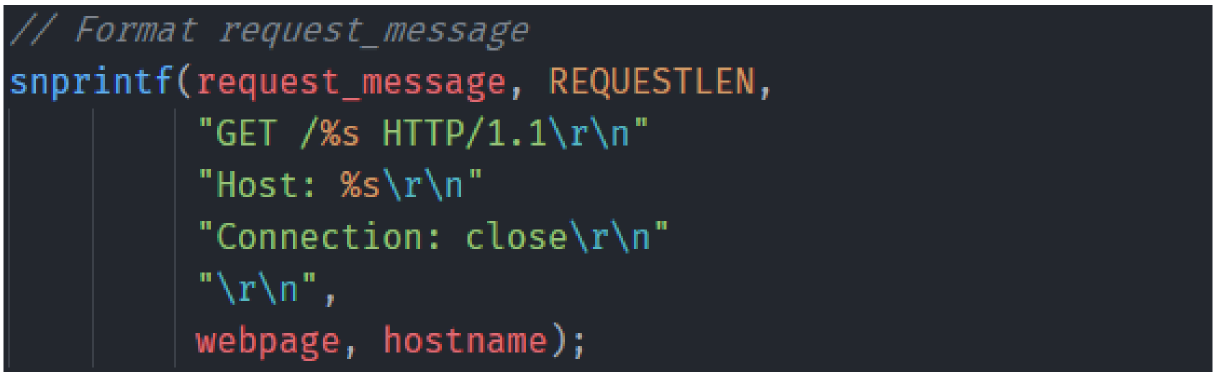
先以 strncmp 判斷 URL 前方是否包含 http://或 https://，再用 strtok 分割出 hostname 以及 webpage，舉例來說，如果 URL = can.cs.nthu.edu.tw/index.php，hostname = can.cs.nthu.edu.tw、webpage = index.php。

2. 獲取 hostname 的 ip



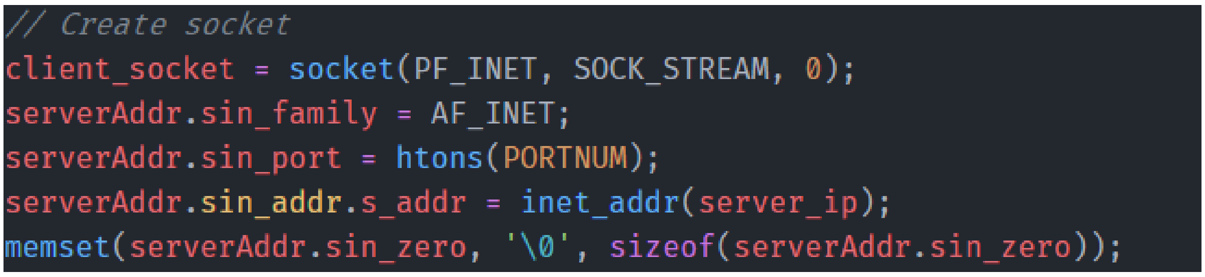
```c
int hostname_to_IP(char *hostname, char *ip) {
    struct hostent *he;
    struct in_addr **addr_list;

    if ((he = gethostbyname(hostname)) == NULL) {
        herror("gethostbyname");
        return -1;
    }

    addr_list = (struct in_addr **)he->h_addr_list;
    strcpy(ip, inet_ntoa(*addr_list[0]));

    return 0;
}
```

3. 設定 request message 的格式



```c
// Format request_message
snprintf(request_message, REQUESTLEN,
            "GET /%s HTTP/1.1\r\n"
            "Host: %s\r\n"
            "Connection: close\r\n"
            "\r\n",
            webpage, hostname);
```

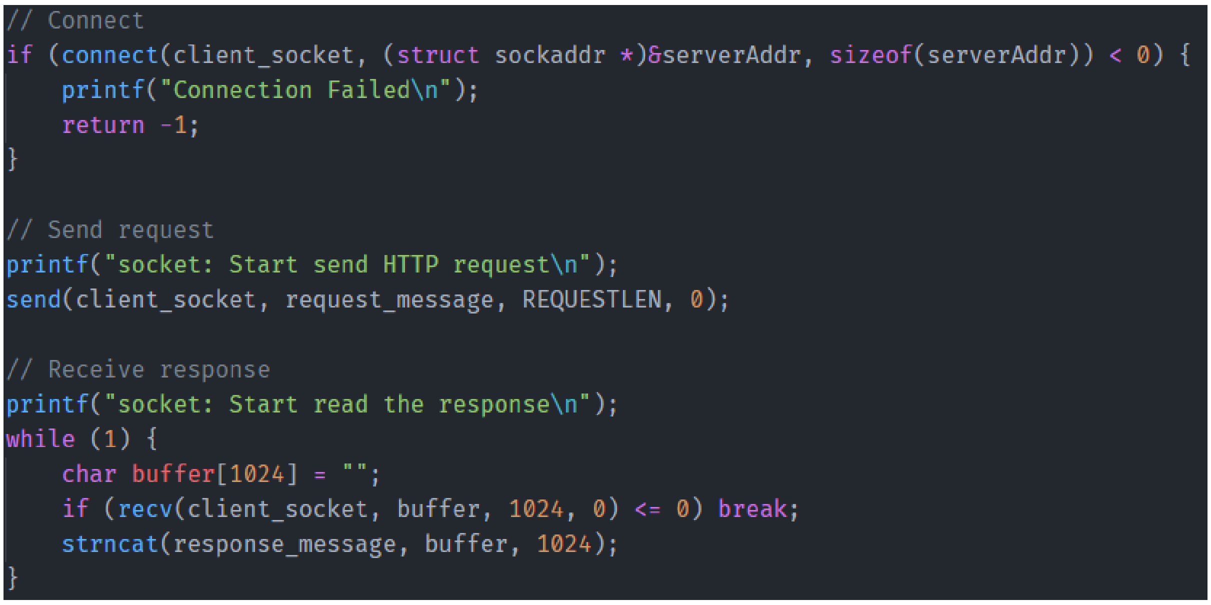
4. 創建 socket



```c
// Create socket
client_socket = socket(PF_INET, SOCK_STREAM, 0);
serverAddr.sin_family = AF_INET;
serverAddr.sin_port = htons(PORTNUM);
serverAddr.sin_addr.s_addr = inet_addr(server_ip);
memset(serverAddr.sin_zero, '\0', sizeof(serverAddr.sin_zero));
```

建立我們的 client socket，PF_INET 代表我們使用 IPV4 網路協定，SOCK_STREAM 表示我們使用 TCP，0 表示我們根據上面兩個協定選擇預設協定，最後還要指定我們要連到 server 的設定，包含協定、port、server 的 ip。

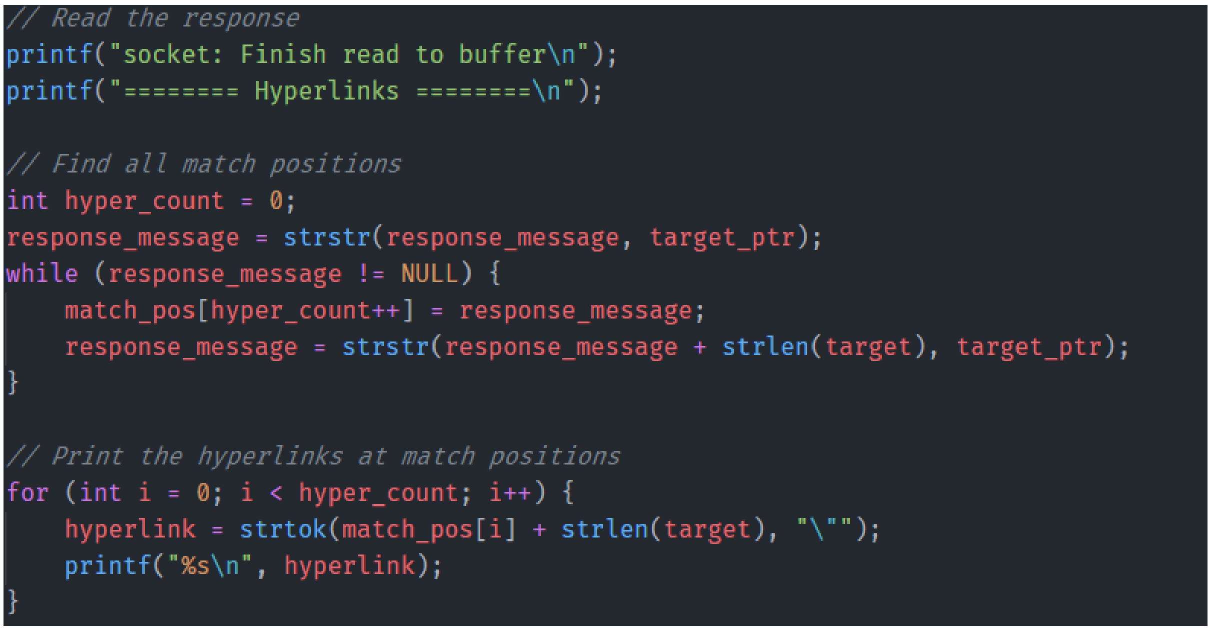
5. 連接 -> 傳送 -> 接收



```c
// Connect
if (connect(client_socket, (struct sockaddr *)&serverAddr, sizeof(serverAddr)) < 0) {
    printf("Connection Failed\n");
    return -1;
}

// Send request
printf("socket: Start send HTTP request\n");
send(client_socket, request_message, REQUESTLEN, 0);

// Receive response
printf("socket: Start read the response\n");
while (1) {
    char buffer[1024] = "";
    if (recv(client_socket, buffer, 1024, 0) <= 0) break;
    strncat(response_message, buffer, 1024);
}
```

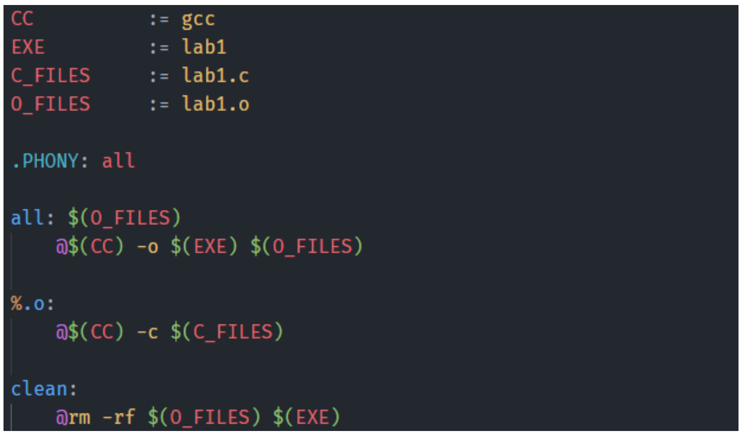
Connect 成功後，傳送 request message，並且接收到 buffer，然後接到 response message 上。

6. 讀取 response message 找出符合的資料

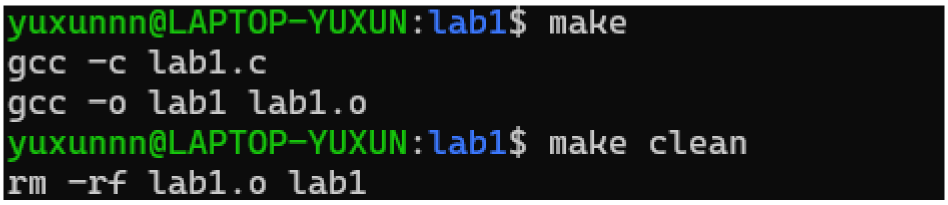


```c
// Read the response
printf("socket: Finish read to buffer\n");
printf("======== Hyperlinks ========\n");

// Find all match positions
int hyper_count = 0;
response_message = strstr(response_message, target_ptr);
while (response_message != NULL) {
    match_pos[hyper_count++] = response_message;
    response_message = strstr(response_message + strlen(target), target_ptr);
}

// Print the hyperlinks at match positions
for (int i = 0; i < hyper_count; i++) {
    hyperlink = strtok(match_pos[i] + strlen(target), "\"");
    printf("%s\n", hyperlink);
}
```

7. 使用 makefile 產生執行檔



```makefile
CC          := gcc
EXE         := lab1
C_FILES     := lab1.c
O_FILES     := lab1.o

.PHONY: all

all: $(O_FILES)
    @$(CC) -o $(EXE) $(O_FILES)

%.o:
    @$(CC) -c $(C_FILES)

clean:
    @rm -rf $(O_FILES) $(EXE)
```



```
yuxunnn@LAPTOP-YUXUN:lab1$ make
gcc -c lab1.c
gcc -o lab1 lab1.o
yuxunnn@LAPTOP-YUXUN:lab1$ make clean
rm -rf lab1.o lab1
```