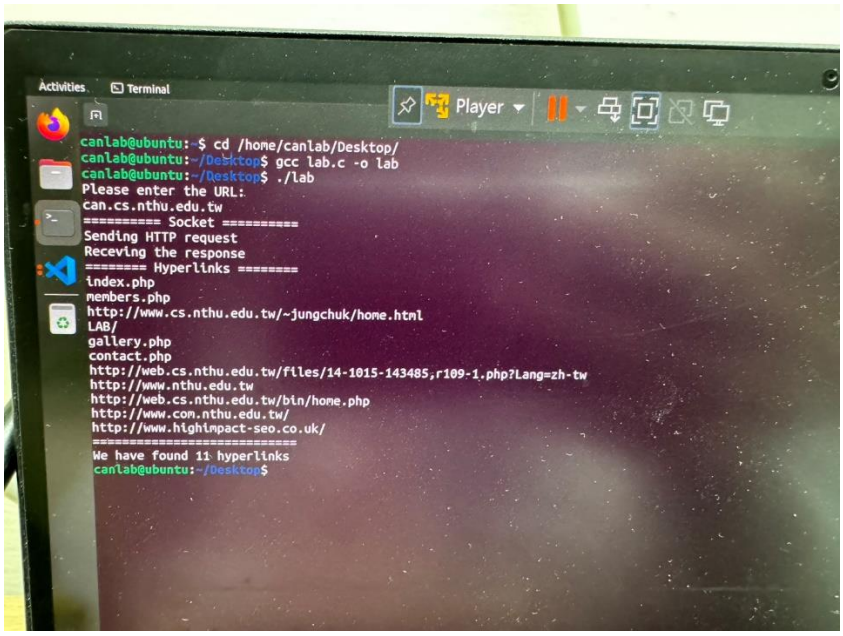
111062108 許凱棋

(1)編譯結果(由於電腦無法於 ubuntu 截圖，請原諒我用手機拍照)



程式碼解釋:

先利用 for 迴圈將 hostname，pathname 區分開來，分別存在 host，pathname 內，hostnum，pathnum 則是各自的字元數，而且 pathname 內至少含有一個 “/”

```
char input[BUFFER_SIZE];
printf("Please enter the URL:\n");
scanf("%s",input);
printf("========== Socket ==========\n");
char host[BUFFER_SIZE];
int hostnum;
char pathname[BUFFER_SIZE]="/";
int pathnum=0;
int horp=0;
char ip[BUFFER_SIZE];
for(int i=0;i<strlen(input);i++){
    if(horp==0&&input[i]=='/'){
        hostnum=i;
        horp=1;
        pathnum++;
    }
    else if(horp==1){
        pathname[pathnum]=input[i];
        pathnum++;
    }
    else if(horp==0) host[i]=input[i];
}
```

此段程式碼是利用 getaddrinfo 將 host 轉為 ip address，並做適當的除錯，且可以做到 IPV4 及 IPV6 兩種形式的控制，參考網站:

https://stackoverflow.com/questions/75843180/get-ip-address-as-string-from-struct-addrinfo

```
int sockfd;
struct addrinfo *addip = NULL;
int err = getaddrinfo(host,NULL,NULL, &addip);
if (err != 0) {
    fprintf(stderr, "error in getaddrinfo: %s\n", gai_strerror(err));
    return -1;
}
if (addip->ai_family == AF_INET) {
    struct sockaddr_in *psai = (struct sockaddr_in*)addip->ai_addr;
    if (inet_ntop(addip->ai_family, &(psai->sin_addr), ip, INET_ADDRSTRLEN) != NULL) {
    }
} else if (addip->ai_family == AF_INET6) {
struct sockaddr_in6 *psai = (struct sockaddr_in6*)addip->ai_addr;
if (inet_ntop(addip->ai_family, &(psai->sin6_addr), ip, INET6_ADDRSTRLEN) != NULL) {
}
}
```

這邊則是控制傳出 request 的 message，並加上適當的除錯

```
struct sockaddr_in server_addr;
socklen_t addrlen = sizeof(server_addr);
char message[BUFFER_SIZE] = "GET ";
unsigned char buffer[BUFFER_SIZE] = {'\0'};
server_addr.sin_family = AF_INET;
server_addr.sin_addr.s_addr = inet_addr(ip);
server_addr.sin_port = htons(PORT);
strcat(message,pathname);
strcat(message," HTTP/1.1\r\n");
strcat(message,"Host: ");
strcat(message,host);
strcat(message,"\r\nConnection: close\r\n\r\n");
if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
perror("socket()");
exit(EXIT_FAILURE);
}
if (connect(sockfd, (struct sockaddr *)&server_addr, addrlen) == -1) {
perror("connect()");
fprintf(stderr, "Please start the server first\n");
exit(EXIT_FAILURE);
}
send(sockfd, message, strlen(message), 0);
printf("Sending HTTP request\n");
```

這邊是確保收到的 response 並沒有出錯

```
if(recv(sockfd, buffer, BUFFER_SIZE, MSG_WAITALL)!=0){
    printf("Receving the response\n");
}
else return 0;
```

最後一段，是從收到的 response 中，將 hyperlinks 切割出來，先判斷是否有<a，在判斷是否有 href=”
如果都有就將資訊記錄下來，並輸出，t 則是控制是否兩個 case 都有達成。

```
printf("======== Hyperlinks ========\n");
int num=0;
for(int i=0;i<strlen(buffer);i++){
    int t=0;
    if(buffer[i]=='<'&&buffer[i+1]=='a'){
        int j=0;
        char tmp[BUFFER_SIZE]="\0";
        while(buffer[i]!='>'){
            i++;
            if(buffer[i]=='h'&&buffer[i+1]=='r'&&buffer[i+2]=='e'&&buffer[i+3]=='f'&&buffer[i+4]=='='&&buffer[i+5]=='"'){
                i+=6;
                num++;
                if(buffer[i]!='"')t=1;
                while(buffer[i]!='"'){
                    tmp[j]=buffer[i];
                    j++;
                    i++;
                }
            }
        }
        if(t==1){
            printf("%s\n",tmp);
        }
    }
}
//printf("%s\n", buffer);
printf("============================\n");
printf("We have found %d hyperlinks\n",num);
close(sockfd);

return 0;
```

學到的東西:

1. response 中開頭為<a 中間有 href=，接著就是 hyperlinks
2. recv 最後一個 flag 若為 MSG_WAITALL 就會接收所有的 response
3. getaddrinfo 函式可以做到 DNS 做的事情(將 host 改成 IP address)