

電腦網路實驗實驗報告 <Socket Programming>

姓名：蕭恒智 學號：407415038

1. 實驗名稱

Socket programming

2. 實驗目的

瞭解 Network Programming，實作透過 UDP 溝通的 Server 與 Client
以及瞭解 IPv6

3. 實驗設備

電腦、記事本、Terminal

4. 實驗步驟

(一) UDP

1. 架設 server

2. Client 端連接上

3. 如下圖成功傳送訊息

```
[(base) xiaohengzhide-MacBook-Pro:實驗四 johnson$ ./udp_server
Server is ready to receive !!
Can strike Cntrl-c to stop Server >>
Received data form 127.0.0.1 : 58253
1234
Can Strike Crttl-c to stop Server >>
Received data form 127.0.0.1 : 57916
johnson
Can Strike Crttl-c to stop Server >>
```

圖一、UDP Server

```
[(base) xiaohengzhide-MacBook-Pro:實驗四 johnson$ ./udp_client 127.0.0.1
1234
[(base) xiaohengzhide-MacBook-Pro:實驗四 johnson$ ./udp_client 127.0.0.1
johnson
```

圖二、UDP Client

(二) IPv6

- 1.架設 Server 端，並設定 port number 跟 IP 位址
- 2.架設 Client 端，並輸入 port number 跟 IP 位址
- 3.連線成功，Client 端傳送訊息
- 4.Client 接收 Server 端收到訊息的回應

```
[(base) xiaohengzhide-MacBook-Pro:實驗四 johnson$ ./server fe80::4b4:2655:bf3f:7c84%en0 8001
[INFO] server setup fe80::4b4:2655:bf3f:7c84%en0[8001]
[INFO] Connection from fe80::4b4:2655:bf3f:7c84[50912]
[CLIENT] abc
[CLIENT] 123
[CLIENT] johnson
[CLIENT] Lab4
```

圖三、Server 端

```
[(base) xiaohengzhide-MacBook-Pro:實驗四 johnson$ ./client fe80::4b4:2655:bf3f:7c84%en0 8001
scope ID = 4
Please enter message : abc
[SERVER] abc[server readed]
Please enter message : 123
[SERVER] 123[server readed]
Please enter message : johnson
[SERVER] johnson[server readed]
Please enter message : Lab4
[SERVER] Lab4[server readed]
Please enter message : 
```

圖四、Client 端

```

1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4 #include <errno.h>
5 #include <unistd.h>
6
7 #include <sys/socket.h>
8 #include <netinet/in.h>
9 #include <arpa/inet.h>
10 #include <netdb.h>
11
12 //error message
13 void error_msg(char*msg);
14
15 int setup_socket(int IP_ver);
16
17 //setup server address
18 void setup_address(char*SERVER_IP, int SERVER_PORT, struct sockaddr_in*storeAddr);
19
20 //interact with server
21 void interact_with_server(int serverSock);
22
23 //get scope id from link local address
24 int getLink_local_addrSCOPEID(char*LINK_LOCAL_ADDR);
25
26 int main(int argc, char *argv[]){
27     //arguments
28     char SERVER_IP[50] = {0};
29     int SERVER_PORT = 0;
30
31     //socket
32     int serverSock;
33     struct sockaddr_in6 serverAddr;
34
35     //process arguments
36     if(argc<3){
37         error_msg("[usage] TCP_client SERVER_IP6 SERVER_PORT (SERVER_IP : IP6%F_NAME)");
38     }
39     strcpy(SERVER_IP, argv[1]);
40     SERVER_PORT = atoi(argv[2]);
41
42     //setup
43     serverSock = setup_socket(6);
44     setup_address(SERVER_IP, SERVER_PORT, &serverAddr);
45     if( connect(serverSock, (struct sockaddr*)&serverAddr, sizeof(serverAddr))<0){
46         error_msg("[ERROR] Failed to connect to server.");
47     }
48     //interact with server
49     interact_with_server(serverSock);
50
51     //close socket
52     close(serverSock);
53
54     return 0;
55 }
56
57 //error message
58 void error_msg(char*msg) {
59     fprintf(stderr, "%s %s\n", msg, strerror(errno));
60     exit(1);
61 }
62
63 //setup socket with specified SERVER_IP, port
64 int setup_socket(int IP_ver){
65     int serverSock;
66     serverSock = socket(IP_ver==4 ? PF_INET : PF_INET6, SOCK_STREAM, 0); //x
67     return serverSock;
68 }
69
70 //setup server address
71 void setup_address(char*SERVER_IP, int SERVER_PORT, struct sockaddr_in*storeAddr){
72     char SERVER_IP_CHR[50] = {0};
73     strncpy(SERVER_IP_CHR, SERVER_IP, (size_t)(strlen(SERVER_IP, '\n')-SERVER_IP));
74
75     storeAddr->sint_family = AF_INET6; //x
76     inet_pton(AF_INET6, SERVER_IP_CHR, &storeAddr->sint_addr); //x
77     storeAddr->sint_port = htons(SERVER_PORT); //x
78     storeAddr->sint_scope_id = getLink_local_addrSCOPEID(SERVER_IP); //x
79     return;
80 }
81
82 //get scope id from link local address
83 int getLink_local_addrSCOPEID(char*LINK_LOCAL_ADDR){
84     int scope_ID=-1;
85     //getaddrinfo
86     struct addrinfo hints, *info;
87     memset(&hints, 0, sizeof(hints));
88     hints.ai_flags = AI_NUMERICHOST;
89
90     //int getaddrinfo(const char*node, const char*service, const struct addrinfo*hints, st
91     if(getaddrinfo(LINK_LOCAL_ADDR, NULL, &hints, &info)==0){
92         struct sockaddr_in6 *sint_info = (struct sockaddr_in6*)info->ai_addr;
93         scope_ID = sint_info->sint_scope_id;
94         freeaddrinfo(info);
95     }
96     printf("scope ID = %d\n", scope_ID);
97     return scope_ID;
98 }
99
100 //interact with server
101 void interact_with_server(int serverSock) {
102     char data[100] = {0};
103
104     while(1){
105         memset(data, 0, sizeof(data));
106
107         //read message from user
108         printf("Please enter message : ");
109         fgets(data, sizeof(data)-1, stdin);
110         data[strlen(data)-1] = '\0';
111
112         //send message to server
113         send(serverSock, data, strlen(data), 0);
114
115         //receive message from server
116         if(recv(serverSock, data, sizeof(data), 0) <= 0){
117             break;
118         }
119         printf("[SERVER] %s\n", data);
120     }
121
122     return;
123 }

```

圖三、Client 端 code

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <errno.h>
5 #include <unistd.h>
6
7 //socket
8 #include <sys/socket.h>
9 #include <netinet/in.h>
10 #include <arpa/inet.h>
11 #include <netdb.h>
12
13 //display error msg
14 void error_msg(char*msg);
15
16 //setup socket
17 int setup_socket(int IP_ver);
18
19 //setup server address
20 void setup_address(char*SERVER_IP, int SERVER_PORT, struct sockaddr_in*storeAddr);
21 //get scope id from link local address
22 int getLink_local_addrSCOPEID(char *LINK_LOCAL_ADDR);
23
24 //interact with client
25 void interact_with_client(int serverSock);
26
27 int main(int argc, char *argv[]){
28     //arguments
29     char SERVER_IP[50]={0};
30     int SERVER_PORT=0;
31
32     //socket - server
33     int serverSock=0;
34     struct sockaddr_in6 serverAddr;
35
36     //socket - client
37     int clientSock=0;
38
39     //process arguments
40     if(argc<3){
41         error_msg("[usage] TCP_server SERVER_IP SERVER_PORT (SERVER_IP : IP%F_NAME)");
42     }
43     strcpy(SERVER_IP, argv[1]);
44     SERVER_PORT = atoi(argv[2]);
45
46     //setup server socket
47     serverSock = setup_socket(6);
48
49     memset(&serverAddr, 0, sizeof(serverAddr));
50     setup_address(SERVER_IP, SERVER_PORT, &serverAddr);
51
52     if(bind(serverSock, (struct sockaddr*)&serverAddr, sizeof(serverAddr))<0){
53         error_msg("[ERROR] Failed to bind.");
54     }
55     listen(serverSock, 2);
56     printf("[INFO] server setup %s\n", SERVER_IP, SERVER_PORT);
57
58     //interact with client
59     interact_with_client(serverSock);
60
61     //close socket
62     close(serverSock);
63
64     return 0;
65 }
66
67 //error message
68 void error_msg(char*msg){
69     fprintf(stderr, "%s %s\n", msg, strerror(errno));
70     exit(1);
71 }
72
73 //setup socket
74 int setup_socket(int IP_ver){
75     int sockFd;
76     int sockopt=1;
77
78     sockFd = socket(IP_ver==4 ? PF_INET : PF_INET6, SOCK_STREAM, 0);
79     setsockopt(sockFd, SOL_SOCKET, SO_REUSEADDR, &sockopt, sizeof(sockopt));
80     return sockFd;
81 }
82
83 //setup server address (only for IPv6)
84 void setup_address(char*SERVER_IP, int SERVER_PORT, struct sockaddr_in*storeAddr) {
85     char SERVER_IP_CHR[50]={0};
86     strncpy(SERVER_IP_CHR, SERVER_IP, (size_t)(strlen(SERVER_IP, '\n')-SERVER_IP));
87
88     storeAddr->sint_family = AF_INET6;
89     inet_pton(AF_INET6, SERVER_IP_CHR, &storeAddr->sint_addr);
90     storeAddr->sint_port = htons(SERVER_PORT);
91     storeAddr->sint_scope_id = getLink_local_addrSCOPEID(SERVER_IP);
92     return;
93 }
94
95 //get scope id from link lo
96 int getLink_local_addrSCOPEID(char *LINK_LOCAL_ADDR){
97     int scope_ID=-1;
98     //getaddrinfo
99     struct addrinfo hints, *info;
100     memset(&hints, 0, sizeof(hints));
101     hints.ai_flags = AI_NUMERICHOST;
102
103     //int getaddrinfo(const char*node, const c
104     if(getaddrinfo(LINK_LOCAL_ADDR, NULL, &hints, &info)==0){
105         struct sockaddr_in6 *sint_info = (struct sockaddr_in6*)info->ai_addr;
106         scope_ID = sint_info->sint_scope_id;
107         freeaddrinfo(info);
108     }
109     return scope_ID;
110 }
111
112 //socket - client
113 void interact_with_client(int serverSock){
114     int clientSock=0;
115     struct sockaddr_in6 clientAddr;
116
117     int clientAddrLength = sizeof(clientAddr);
118     char clientAddr_str[50]={0};
119
120     //Message
121     char msg[BUFFSIZE]={0};
122
123     while(1){
124         //wait for client
125         clientSock = accept(serverSock, (struct sockaddr*)&clientAddr, &clientAddrLength);
126         inet_ntop(AF_INET6, &clientAddr.sint_addr, clientAddr_str, sizeof(clientAddr_str));
127         printf("[INFO] Connection from %s\n", clientAddr_str, ntohs(clientAddr.sint_port));
128
129         while(1){
130             //receive message from server
131             memset(msg, 0, sizeof(msg));
132             if(recv(clientSock, msg, sizeof(msg), 0)<=0){
133                 printf("[INFO] Client disconnected. \n");
134                 break;
135             }
136             printf(" [CLIENT] %s\n", msg);
137
138             strcat(msg, "[server readed]");
139             //send message back to client
140             send(clientSock, msg, strlen(msg), 0);
141         }
142     }
143
144     return;
145 }

```

圖四、Server 端 code

5. 問題與討論

這次的實驗在抓 IPv6 位置時，我一直遇到問題，最後終於在網路上找到答案，抓到了正確的 IPv6 位置，程式終於可以正常運作了。

6. 心得與感想

這次實驗都是要練習 coding，在語法上與之前的相當類似，但是額外使用了許多新的函式，其實打起來挺複雜的，在打 code 的途中遇到蠻多問題，但最後做出來，覺得蠻有趣的，也學到了很多。

7. 參考文獻