

tags: 網路程式設計

網路程式作業 HW12

三種傳播類型:

Unicast 單播(1 to 1)

Broadcast 廣播(1 to all)

Multicast 群播(1 to part)

廣播位址：

用全1的取代IP位址中主機部分

練習一基本廣播示範 (client,server)

- 兩端都要建立UDP socket，並打開廣播功能
- 可以同時執行兩個server嗎？可以 client端會跑更快，依次接收到更多訊息。
- 可以同時執行兩個client嗎？不行 為何？ 一台電腦只有綁定一個port

練習2：server傳送多份資料(相同port)

- Server每秒傳送 : 1111111111111111,...
aaaaaaaaaaaaaaaaaaaaa,...
AAAAAAAAAAAAAAAAAA,...(不同資料)

```

C:\Users\ASUS\Desktop\網程> server start broadcasting on port:5678
server broadcast: 11111111111111111111
server broadcast: aaaaaaaaaaaaaaaaaaaaaa
server broadcast: AAAAAAAAAAAAAAAAAAAAAA
server broadcast: 22222222222222222222
server broadcast: bbbbbbbbbbbbbbbbbbbbbb
server broadcast: BBBBBBBBBBBBBBBBBBBBBBBB
server broadcast: 33333333333333333333
server broadcast: cccccccccccccccccccccc
server broadcast: CCCCCCCCCCCCCCCCCCCC
server broadcast: 44444444444444444444
server broadcast: dddddddddddddddddddd
server broadcast: DDDDDDDDDDDDDDDDDDDDD
server broadcast: 55555555555555555555
server broadcast: eeeeeeeeeeeeeeeeeeee
server broadcast: EEEEEEEEEEEEEEEEEEEE
server broadcast: 66666666666666666666
server broadcast: ffffffffffffffffffffff
server broadcast: FFFFFFFFFFFFFFFFFFFF
server broadcast: 77777777777777777777
server broadcast: gggggggggggggggggggggg
server broadcast: GGGGGGGGGGGGGGGGGGGGG
server broadcast: 88888888888888888888
server broadcast: hhhhhhhhhhhhhhhhhhhhhh
server broadcast: HHHHHHHHHHHHHHHHHHHH

C:\Users\ASUS\Desktop\網程> client(from:172.20.10.9):44444444444444444444
client(from:172.20.10.9):dddddddddddddddddddd
client(from:172.20.10.9):DDDDDDDDDDDDDDDDDDDD
client(from:172.20.10.9):55555555555555555555
client(from:172.20.10.9):eeeeeeeeeeeeeeeeeeee
client(from:172.20.10.9):EEEEEEEEEEEEEEEEEEEE
client(from:172.20.10.9):66666666666666666666
client(from:172.20.10.9):ffffffffffffffffffffff
client(from:172.20.10.9):FFFFFFFFFFFFFFFFFFFF
client(from:172.20.10.9):77777777777777777777
client(from:172.20.10.9):gggggggggggggggggggggg
client(from:172.20.10.9):GGGGGGGGGGGGGGGGGGGG
client(from:172.20.10.9):88888888888888888888
client(from:172.20.10.9):hhhhhhhhhhhhhhhhhhhhh
client(from:172.20.10.9):HHHHHHHHHHHHHHHHHHHH

```

:::spoiler server

```

1  #include <stdio.h>
2  #include <string.h>
3  #include <winsock.h>
4
5  #define MAXLINE 20      /* 字串緩衝區長度 */
6  #define BraodcastPort 5678
7
8  int main()
9  {
10     SOCKET serv_sd;      /* socket 描述子 */
11     int cli_len, n,i=1;
12     char str[MAXLINE];
13
14     struct sockaddr_in serv, cli;
15     WSADATA wsadata;
16
17     WSStartup(0x101, &wsadata); //呼叫 WSStartup() 註冊
winSock DLL 的使用
18

```

```
19     serv_sd=socket(AF_INET, SOCK_DGRAM, 0); // 開啟 UDP
    socket
20     BOOL broadcast = TRUE;
21     setsockopt(serv_sd, SOL_SOCKET, SO_BROADCAST, ( const
    char* )&broadcast, sizeof(broadcast));
22
23
24     cli_len = sizeof(cli);
25
26
27     printf("server start broadcasting on
    port:%d\n", BraodcastPort);
28
29     cli.sin_family      = AF_INET;
30     cli.sin_addr.s_addr = inet_addr("255.255.255.255");
31     cli.sin_port        = htons(BraodcastPort);
32
33     while(1){
34         memset(str, i%10 + '0', sizeof(str));
35         sendto(serv_sd, str, strlen(str), 0,
    (LPSOCKADDR)&cli, cli_len);
36         printf("server broadcast: %s\n", str); //顯示送去
    client的字串
37
38         memset(str, (i-1)%26 + 'a', sizeof(str));
39         sendto(serv_sd, str, strlen(str), 0,
    (LPSOCKADDR)&cli, cli_len);
40         printf("server broadcast: %s\n", str); //顯示送去
    client的字串
41
42         memset(str, (i-1)%26 + 'A', sizeof(str));
43         sendto(serv_sd, str, strlen(str), 0,
    (LPSOCKADDR)&cli, cli_len);
44         printf("server broadcast: %s\n", str); //顯示送去
    client的字串
45
46         sleep(1000);
```

```

47         i++;
48     }
49
50     //結束 winSock DLL 的使用
51     closesocket(serv_sd);
52     //closesocket(cli_sd);
53     WSACleanup();
54 }
55

```

...

:::spoiler client

```

1  #include <stdio.h>
2  #include <string.h>
3  #include <winsock.h>
4
5  #define MAXLINE 20    /* 字串緩衝區長度 */
6  #define BraodcastPort 5678
7
8  int main()
9  {
10     SOCKET serv_sd;        /* socket 描述子 */
11     int cli_len, n,i=1;
12     char str[MAXLINE];
13
14     struct sockaddr_in serv, cli;
15     WSADATA wsadata;
16
17     WSStartup(0x101, &wsadata); //呼叫 WSStartup() 註冊
    winSock DLL 的使用
18
19     serv_sd=socket(AF_INET, SOCK_DGRAM, 0); // 開啟 UDP
    socket
20     BOOL broadcast = TRUE;
21     setsockopt(serv_sd, SOL_SOCKET, SO_BROADCAST, ( const
    char* )&broadcast, sizeof(broadcast));

```

```
22
23
24     cli_len = sizeof(cli);
25
26
27     printf("server start broadcasting on
port:%d\n", BraodcastPort);
28
29     cli.sin_family      = AF_INET;
30     cli.sin_addr.s_addr = inet_addr("255.255.255.255");
31     cli.sin_port        = htons(BraodcastPort);
32
33     while(1){
34         memset(str, i%10 + '0', sizeof(str));
35         sendto(serv_sd, str, strlen(str), 0,
(LPSOCKADDR)&cli, cli_len);
36         printf("server broadcast: %s\n", str); //顯示送去
client的字串
37
38         memset(str, (i-1)%26 + 'a', sizeof(str));
39         sendto(serv_sd, str, strlen(str), 0,
(LPSOCKADDR)&cli, cli_len);
40         printf("server broadcast: %s\n", str); //顯示送去
client的字串
41
42         memset(str, (i-1)%26 + 'A', sizeof(str));
43         sendto(serv_sd, str, strlen(str), 0,
(LPSOCKADDR)&cli, cli_len);
44         printf("server broadcast: %s\n", str); //顯示送去
client的字串
45
46         sleep(1000);
47         i++;
48     }
49
50     //結束 winSock DLL 的使用
51     closesocket(serv_sd);
```

- Client依內容過濾，只顯示特定資料
- 每顯示10筆，重新輸入

:::spoiler server

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <winsock.h>
4
5 #define MAXLINE 20      /* 字串緩衝區長度 */
6 #define BraodcastPort 5678
7
```

```

8  int main()
9  {
10     SOCKET  serv_sd;          /* socket 描述子 */
11     int      cli_len, n,i=1;
12     char     str[MAXLINE];
13
14     struct  sockaddr_in      serv, cli;
15     WSADATA wsadata;
16
17     WSStartup(0x101, &wsadata); //呼叫 WSStartup() 註冊
winSock DLL 的使用
18
19     serv_sd=socket(AF_INET, SOCK_DGRAM, 0); // 開啟 UDP
socket
20     BOOL broadcast = TRUE;
21     setsockopt(serv_sd, SOL_SOCKET, SO_BROADCAST, ( const
char* )&broadcast, sizeof(broadcast));
22
23
24     cli_len = sizeof(cli);
25
26
27     printf("server start broadcasting on
port:%d\n",BraodcastPort);
28
29     cli.sin_family      = AF_INET;
30     cli.sin_addr.s_addr = inet_addr("255.255.255.255");
31     cli.sin_port        = htons(BraodcastPort);
32
33     while(1){
34         memset(str, i%10 +'0', sizeof(str));
35         sendto(serv_sd, str, strlen(str), 0,
(LPSOCKADDR)&cli,cli_len);
36         printf("server broadcast: %s\n",str); //顯示送去
client的字串
37
38         memset(str, (i-1)%26 +'a', sizeof(str));

```

```

39         sendto(serv_sd, str, strlen(str), 0,
(LPSOCKADDR)&cli,cli_len);
40         printf("server broadcast: %s\n",str);//顯示送去
client的字串
41
42         memset(str, (i-1)%26 +'A', sizeof(str));
43         sendto(serv_sd, str, strlen(str), 0,
(LPSOCKADDR)&cli,cli_len);
44         printf("server broadcast: %s\n",str);//顯示送去
client的字串
45
46         sleep(1000);
47         i++;
48     }
49
50     //結束 winsock DLL 的使用
51     closesocket(serv_sd);
52     //closesocket(cli_sd);
53     WSACleanup();
54 }
55

```

:::

:::spoiler client

```

1  #include <stdio.h>
2  #include <string.h>
3  #include <winsock.h>
4  #define BraodcastPort 5678
5  #define MAXLINE 1024
6
7  int main(int argc, char** argv) {
8
9      SOCKET          sd;
10     struct sockaddr_in serv,cli;
11     char             str[1024];
12     WSADATA          wsadata;

```



```

13     int n,serv_len,channel;
14
15     WSStartup(0x101,(LPWSADATA) &wsadata); // 呼叫
WSStartup() 註冊 WinSock DLL 的使用
16
17     sd=socket(AF_INET, SOCK_DGRAM, 0);
18     BOOL broadcast = TRUE;
19     if( setsockopt(sd, SOL_SOCKET, SO_BROADCAST, (const
char*)&broadcast, sizeof(broadcast))<0)
20         printf("setsockopt() error!\n");
21
22     cli.sin_family      = AF_INET;
23     cli.sin_addr.s_addr = 0;//自動找IP
24     cli.sin_port        = htons(BroadcastPort);
25     if( bind(sd, (LPSOCKADDR) &cli, sizeof(cli)) <0 ){
26         printf("bind error!\n");
27         system("pause");
28         return 0;
29     }
30     serv_len=sizeof(serv);
31     while(1){
32         printf("Channel(1-3):");
33         scanf("%d",&channel);
34         printf("\n");
35         for(int i=0;i<30;i++)
36             {
37                 n=recvfrom(sd, str, MAXLINE, 0,(LPSOCKADDR)
&serv,&serv_len ); //由echo server接收
38                 switch(channel)
39                 {
40                     case 1:
41                         if(str[0] >= '0' && str[0] <= '9')
42                             {
43
44                             printf("client(from:%s):%s\n",inet_ntoa(serv.sin_addr),s
tr);

```

```

45         break;
46     case 2:
47         if(str[0] >= 'a' && str[0] <= 'z')
48         {
49
50             printf("client(from:%s):%s\n",inet_ntoa(serv.sin_addr),s
tr);
51         }
52         break;
53     case 3:
54         if(str[0] >= 'A' && str[0] <= 'Z')
55         {
56
57             printf("client(from:%s):%s\n",inet_ntoa(serv.sin_addr),s
tr);
58         }
59         break;
60     }
61 }
62
63 closesocket(sd); //關閉 socket
64 WSACleanup(); // 結束 winSock DLL 的使用
65 return 0;
66 }
67
68

```

...

練習4：Client使用==thread==作出選台效果

- 隨時都可以重新輸入，顯示特定資料

```

C:\Users\ASUS\Desktop\網程\ x + v
server start broadcasting on port:5678
server broadcast: 11111111111111111111
server broadcast: aaaaaaaaaaaaaaaaaaaaaa
server broadcast: AAAAAAAAAAAAAAAAAAAAAA
server broadcast: 22222222222222222222
server broadcast: bbbbbbbbbbbbbbbbbbbbbb
server broadcast: BBBBBBBBBBBBBBBBBBBBBB
server broadcast: 33333333333333333333
server broadcast: cccccccccccccccccccc
server broadcast: CCCCCCCCCCCCCCCCCCCC
server broadcast: 44444444444444444444
server broadcast: dddddddddddddddddddd
server broadcast: DDDDDDDDDDDDDDDDDDDDD
server broadcast: 55555555555555555555
server broadcast: eeeeeeeeeeeeeeeeeeee
server broadcast: EEEEEEEEEEEEEEEEEEEE
server broadcast: 66666666666666666666
server broadcast: ffffffffffffffffffffff
server broadcast: FFFFFFFFFFFFFFFFFFFF
server broadcast: 77777777777777777777
server broadcast: gggggggggggggggggggg
server broadcast: GGGGGGGGGGGGGGGGGGGG
server broadcast: 88888888888888888888
server broadcast: hhhhhhhhhhhhhhhhhhhh
server broadcast: HHHHHHHHHHHHHHHHHHH
server broadcast: 99999999999999999999
server broadcast: iiiiiiiiiiiiiiiiii
server broadcast: IIIIIIIIIIIIIIIIIII
server broadcast: 00000000000000000000
server broadcast: jjjjjjjjjjjjjjjjjj

C:\Users\ASUS\Desktop\網程\ x + v
1
switch to channel 1
client(from:172.20.10.9):77777777777777777777
client(from:172.20.10.9):88888888888888888888
client(from:172.20.10.9):99999999999999999999
client(from:172.20.10.9):00000000000000000000
client(from:172.20.10.9):11111111111111111111
2client(from:172.20.10.9):22222222222222222222

switch to channel 2
client(from:172.20.10.9):mmmmmmmmmmmmmmmmmmmm
client(from:172.20.10.9):nnnnnnnnnnnnnnnnnnnn
client(from:172.20.10.9):oooooooooooooooooooo
client(from:172.20.10.9):pppppppppppppppppppp
client(from:172.20.10.9):qqqqqqqqqqqqqqqqqqqq
3
switch to channel 3
client(from:172.20.10.9):rrrrrrrrrrrrrrrrrrrrrr
client(from:172.20.10.9):ssssssssssssssssssss
client(from:172.20.10.9):tttttttttttttttttttt
1
switch to channel 1
client(from:172.20.10.9):11111111111111111111
client(from:172.20.10.9):22222222222222222222
client(from:172.20.10.9):33333333333333333333
client(from:172.20.10.9):44444444444444444444
client(from:172.20.10.9):55555555555555555555
client(from:172.20.10.9):66666666666666666666
client(from:172.20.10.9):77777777777777777777
client(from:172.20.10.9):88888888888888888888

```

:::spoiler server

```

1 #include <stdio.h>
2 #include <string.h>
3 #include <winsock.h>
4
5 #define MAXLINE 20      /* 字串緩衝區長度 */
6 #define BraodcastPort 5678
7
8 int main()
9 {
10     SOCKET serv_sd;      /* socket 描述子 */
11     int cli_len, n,i=1;
12     char str[MAXLINE];
13
14     struct sockaddr_in serv, cli;
15     WSADATA wsadata;
16
17     WSStartup(0x101, &wsadata); //呼叫 WSStartup() 註冊
winSock DLL 的使用

```

```
18
19     serv_sd=socket(AF_INET, SOCK_DGRAM, 0); // 開啟 UDP
socket
20     BOOL broadcast = TRUE;
21     setsockopt(serv_sd, SOL_SOCKET, SO_BROADCAST, ( const
char* )&broadcast, sizeof(broadcast));
22
23
24     cli_len = sizeof(cli);
25
26
27     printf("server start broadcasting on
port:%d\n", BraodcastPort);
28
29     cli.sin_family      = AF_INET;
30     cli.sin_addr.s_addr = inet_addr("255.255.255.255");
31     cli.sin_port        = htons(BraodcastPort);
32
33     while(1){
34         memset(str, i%10 + '0', sizeof(str));
35         sendto(serv_sd, str, strlen(str), 0,
(LPSOCKADDR)&cli, cli_len);
36         printf("server broadcast: %s\n", str); //顯示送去
client的字串
37
38         memset(str, (i-1)%26 + 'a', sizeof(str));
39         sendto(serv_sd, str, strlen(str), 0,
(LPSOCKADDR)&cli, cli_len);
40         printf("server broadcast: %s\n", str); //顯示送去
client的字串
41
42         memset(str, (i-1)%26 + 'A', sizeof(str));
43         sendto(serv_sd, str, strlen(str), 0,
(LPSOCKADDR)&cli, cli_len);
44         printf("server broadcast: %s\n", str); //顯示送去
client的字串
45
```

```

46         sleep(1000);
47         i++;
48     }
49
50     //結束 winSock DLL 的使用
51     closesocket(serv_sd);
52     //closesocket(cli_sd);
53     WSACleanup();
54 }
55

```

...

:::spoiler client

```

1  #include <stdio.h>
2  #include <string.h>
3  #include <winsock.h>
4  #define BraodcastPort 5678
5  #define MAXLINE 1024
6  SOCKET          sd;
7  struct sockaddr_in cli;
8  char            str[1024];
9  WSADATA          wsadata;
10 int n,cli_len,channel;
11
12 void *recvMess(void *argu)//每隔一秒鐘印出一次George的函數
13 {
14     while(1)
15     {
16         cli_len = sizeof(cli);
17         n=recvfrom(sd, str, MAXLINE, 0,(LPSOCKADDR)
18         &cli,&cli_len ); //由echo server接收
19         switch(channel)
20         {
21             case 1:
22                 if(str[0] >= '0' && str[0] <= '9')
23

```

```

23     printf("client(from:%s):%s\n",inet_ntoa(cli.sin_addr),st
24     r);
25         }
26         break;
27     case 2:
28         if(str[0] >= 'a' && str[0] <= 'z')
29         {
30             printf("client(from:%s):%s\n",inet_ntoa(cli.sin_addr),st
31             r);
32             }
33             break;
34     case 3:
35         if(str[0] >= 'A' && str[0] <= 'Z')
36         {
37             printf("client(from:%s):%s\n",inet_ntoa(cli.sin_addr),st
38             r);
39             }
40             break;
41         }
42     }
43 }
44 int main(int argc, char** argv) {
45     struct sockaddr_in serv;
46     WSADATA          wsadata;
47     int serv_len,threadID;
48
49     WSStartup(0x101,(LPWSADATA) &wsadata); // 呼叫
WSAStartup() 註冊 winSock DLL 的使用
50
51     sd=socket(AF_INET, SOCK_DGRAM, 0);
52     BOOL broadcast = TRUE;

```

```

53     if( setsockopt(sd, SOL_SOCKET, SO_BROADCAST, (const
char*)&broadcast, sizeof(broadcast))<0)
54         printf("setsockopt() error!\n");
55
56     serv.sin_family      = AF_INET;
57     serv.sin_addr.s_addr = 0;//自動找IP
58     serv.sin_port        = htons(BroadcastPort);
59     if( bind(sd, (LPSOCKADDR) &serv, sizeof(serv)) <0 ){
60         printf("bind error!\n");
61         system("pause");
62         return 0;
63     }
64     serv_len=sizeof(serv);
65
66     CreateThread(NULL,0, (LPTHREAD_START_ROUTINE)
recvMess,NULL,0, (LPDWORD)&threadID);
67     while(1)
68     {
69         scanf("%d",&channel);
70         printf("switch to channel %d\n",channel);
71     }
72     closesocket(sd); //關閉 socket
73     WSACleanup(); // 結束 winSock DLL 的使用
74     return 0;
75 }
76

```

...

練習5：用port實現多頻道

- Server在不同port傳送多種資料

- Client 切換不同的 port 聽(先關閉 socket，再重新建立並 bind)

```

C:\Users\ASUS\Desktop\網程\ x + v
server start broadcasting on port:5678
server broadcast: 11111111111111111111
server broadcast: aaaaaaaaaaaaaaaaaaaa
server broadcast: AAAAAAAAAAAAAAAAAAAA
server broadcast: 22222222222222222222
server broadcast: bbbbbbbbbbbbbbbbbbbb
server broadcast: BBBBBBBBBBBBBBBBBBBB
server broadcast: 33333333333333333333
server broadcast: cccccccccccccccccccc
server broadcast: CCCCCCCCCCCCCCCCCCCC
server broadcast: 44444444444444444444
server broadcast: dddddddddddddddddddd
server broadcast: DDDDDDDDDDDDDDDDDDDDD
server broadcast: 55555555555555555555
server broadcast: eeeeeeeeeeeeeeeeeeee
server broadcast: EEEEEEEEEEEEEEEEEEEE
server broadcast: 66666666666666666666
server broadcast: ffffffffffffffffffffffff
server broadcast: FFFFFFFFFFFFFFFFFFFFFF
server broadcast: 77777777777777777777
server broadcast: gggggggggggggggggggggg
server broadcast: GGGGGGGGGGGGGGGGGGGGG
server broadcast: 88888888888888888888
server broadcast: hhhhhhhhhhhhhhhhhhhhhh
server broadcast: HHHHHHHHHHHHHHHHHHHH
server broadcast: 99999999999999999999
server broadcast: iiiiiiiiiiiiiiiiiiiiii
server broadcast: IIIIIIIIIIIIIIIIIIII
server broadcast: 00000000000000000000
server broadcast: jjjjjjjjjjjjjjjjjjjj

C:\Users\ASUS\Desktop\網程\ x + v
client(from:172.20.10.9):55555555555555555555,20
client(from:172.20.10.9):66666666666666666666,20
client(from:172.20.10.9):77777777777777777777,20
client(from:172.20.10.9):88888888888888888888,20
client(from:172.20.10.9):99999999999999999999,20
client(from:172.20.10.9):00000000000000000000,20
client(from:172.20.10.9):11111111111111111111,20

switch to channel 1
client(from:172.20.10.9):22222222222222222222,20
client(from:172.20.10.9):33333333333333333333,20
client(from:172.20.10.9):44444444444444444444,20
client(from:172.20.10.9):55555555555555555555,20
client(from:172.20.10.9):66666666666666666666,20
2
switch to channel 2
client(from:172.20.10.9):qqqqqqqqqqqqqqqqqqqq,20
client(from:172.20.10.9):rrrrrrrrrrrrrrrrrrrr,20
client(from:172.20.10.9):ssssssssssssssssssss,20
client(from:172.20.10.9):tttttttttttttttttttt,20
client(from:172.20.10.9):uuuuuuuuuuuuuuuuuuuu,20
client(from:172.20.10.9):vvvvvvvvvvvvvvvvvvvv,20
3
switch to channel 3
client(from:172.20.10.9):wwwwwwwwwwwwwwwwwwww,20
client(from:172.20.10.9):xxxxxxxxxxxxxxxxxxxxxx,20
client(from:172.20.10.9):yyyyyyyyyyyyyyyyyyyy,20
client(from:172.20.10.9):zzzzzzzzzzzzzzzzzzzz,20
client(from:172.20.10.9):aaaaaaaaaaaaaaaaaaaaaa,20
client(from:172.20.10.9):bbbbbbbbbbbbbbbbbbbb,20

```

:::spoiler server

```

1
2 #include <stdio.h>
3 #include <string.h>
4 #include <winsock.h>
5
6 #define MAXLINE 20      /* 字串緩衝區長度 */
7 #define BraodcastPort 5678
8
9 int main()
10 {
11     SOCKET serv_sd;      /* socket 描述子 */
12     int cli_len, n,i=1;
13     char str[MAXLINE];
14
15     struct sockaddr_in serv, cli;
16     WSADATA wsadata;
17
18     WSStartup(0x101, &wsadata); //呼叫 WSStartup() 註冊
WinSock DLL 的使用

```



```
19
20     serv_sd=socket(AF_INET, SOCK_DGRAM, 0); // 開啟 UDP
socket
21     BOOL broadcast = TRUE;
22     setsockopt(serv_sd, SOL_SOCKET, SO_BROADCAST, ( const
char* )&broadcast, sizeof(broadcast));
23
24
25     cli_len = sizeof(cli);
26
27
28     printf("server start broadcasting on
port:%d\n", BraodcastPort);
29
30     cli.sin_family      = AF_INET;
31     cli.sin_addr.s_addr = inet_addr("255.255.255.255");
32     cli.sin_port        = htons(BraodcastPort);
33
34     while(1){
35         cli.sin_port      = htons(BraodcastPort);
36         memset(str, i%10 +'0', sizeof(str));
37         sendto(serv_sd, str, strlen(str), 0,
(LPSOCKADDR)&cli, cli_len);
38         printf("server broadcast: %s\n", str); //顯示送去
client的字串
39
40         cli.sin_port      = htons(BraodcastPort+1);
41         memset(str, (i-1)%26 +'a', sizeof(str));
42         sendto(serv_sd, str, strlen(str), 0,
(LPSOCKADDR)&cli, cli_len);
43         printf("server broadcast: %s\n", str); //顯示送去
client的字串
44
45         cli.sin_port      = htons(BraodcastPort+2);
46         memset(str, (i-1)%26 +'A', sizeof(str));
47         sendto(serv_sd, str, strlen(str), 0,
(LPSOCKADDR)&cli, cli_len);
```

```

48         printf("server broadcast: %s\n",str);//顯示送去
client的字串
49
50         sleep(1000);
51         i++;
52     }
53
54     //結束 winSock DLL 的使用
55     closesocket(serv_sd);
56     //closesocket(cli_sd);
57     WSACleanup();
58 }
59

```

:::

:::spoiler client

```

1  #include <stdio.h>
2  #include <string.h>
3  #include <winsock.h>
4  #define BraodcastPort 5678
5  #define MAXLINE 1024
6  SOCKET          sd;
7  struct sockaddr_in cli;
8  char            str[1024];
9  WSADATA          wsadata;
10 int n,cli_len,channel;
11
12 void *recvMess(void *argu)//每隔一秒鐘印出一次George的函數
13 {
14     while(1)
15     {
16         cli_len = sizeof(cli);
17         n=recvfrom(sd, str, MAXLINE, 0,(LPSOCKADDR)
&cli,&cli_len ); //由echo server接收
18
19         if(n>0)

```

```

20         {
21
22         printf("client(from:%s):%s,%d\n",inet_ntoa(cli.sin_addr)
, str,n);
22         }
23     }
24
25 }
26
27 int main(int argc, char** argv) {
28     struct sockaddr_in serv;
29     WSADATA          wsadata;
30     int serv_len,threadID;
31
32     WSStartup(0x101,(LPWSADATA) &wsadata); // 呼叫
WSStartup() 註冊 winsock DLL 的使用
33
34     sd=socket(AF_INET, SOCK_DGRAM, 0);
35     BOOL broadcast = TRUE;
36     if( setsockopt(sd, SOL_SOCKET, SO_BROADCAST, (const
char*)&broadcast, sizeof(broadcast))<0)
37     printf("setsockopt() error!\n");
38
39     serv.sin_family      = AF_INET;
40     serv.sin_addr.s_addr = 0;//自動找IP
41     serv.sin_port        = htons(BroadcastPort);
42     if( bind(sd, (LPSOCKADDR) &serv, sizeof(serv)) <0 ){
43         printf("bind error!\n");
44         system("pause");
45         return 0;
46     }
47     serv_len=sizeof(serv);
48
49     CreateThread(NULL,0,(LPTHREAD_START_ROUTINE)
recvMess,NULL,0,(LPDWORD)&threadID);
50     while(1)
51     {

```

```
52     scanf("%d",&channel);
53     printf("switch to channel %d\n",channel);
54     closesocket(sd); //關閉 socket
55     sd=socket(AF_INET, SOCK_DGRAM, 0); //重新連線
56     BOOL broadcast = TRUE;
57     if( setsockopt(sd, SOL_SOCKET, SO_BROADCAST,
(const char*)&broadcast, sizeof(broadcast))<0)
58         printf("setsockopt() error!\n");
59
60     serv.sin_family      = AF_INET;
61     serv.sin_addr.s_addr = 0; //自動找IP
62
63     switch(channel)
64     {
65         case 1:
66             serv.sin_port = htons(BraodcastPort);
67             if( bind(sd, (LPSOCKADDR) &serv,
sizeof(serv)) <0 ){
68                 printf("bind error!\n");
69                 system("pause");
70                 return 0;}
71             break;
72         case 2:
73             serv.sin_port = htons(BraodcastPort+1);
74             if( bind(sd, (LPSOCKADDR) &serv,
sizeof(serv)) <0 ){
75                 printf("bind error!\n");
76                 system("pause");
77                 return 0;}
78             break;
79         case 3:
80             serv.sin_port = htons(BraodcastPort+2);
81             if( bind(sd, (LPSOCKADDR) &serv,
sizeof(serv)) <0 ){
82                 printf("bind error!\n");
83                 system("pause");
84                 return 0;}
```

```
85         break;
86
87     }
88 }
89 closesocket(sd); //關閉 socket
90 WSACleanup(); // 結束 winsock DLL 的使用
91 return 0;
92 }
93
```

...

心得

這星期學到了如何廣播，但要小心使用，以免造成廣播風暴。