

## 網路程式設計 HW-5 種服務的示範程式

### 一、 程式說明

程式執行時，請輸入一個數字，表示要執行的服務，服務代號 1 到 5 分別為 1 回應服務 (Echo)、2 丟棄服務 (Discard)、3 日期時間服務 (Daytime)、4 每日一句 (QUOTE)、5 字元產生器 (CHARGEN)，輸入完後，等待程式執行，即可查看執行結果。

### 二、 執行畫面

執行 1 回應服務 (Echo)的執行結果如下。

```
C:\data\Code\Cpp\network_programming\cmake-build-debug\hw2.exe
+++++
| 請輸入服務的代號(1~5)以執行服務 |
| 1. 回應服務 (Echo) |
| 2. 丟棄服務 (Discard) |
| 3. 日期時間服務 (Daytime) |
| 4. 每日一句 (QUOTE) |
| 5. 字元產生器 (CHARGEN) |
+++++
1
+++++
| 執行結果 |
+++++
[傳送內容]
I love NCYU CSIE
[傳送內容長度] 17bytes
[回應內容]
I love NCYU CSIE
[回應內容長度] 17bytes

Process finished with exit code 0
```

執行 2 丟棄服務 (Discard) 的執行結果如下。

備註：執行此服務需手動中斷程式。

```
C:\data\Code\Cpp\network_programming\cmake-build-debug\hw2.exe
+++++
| 請輸入服務的代號(1~5)以執行服務 |
| 1. 回應服務 (Echo) |
| 2. 丟棄服務 (Discard) |
| 3. 日期時間服務 (Daytime) |
| 4. 每日一句 (QUOTE) |
| 5. 字元產生器 (CHARGEN) |
+++++
2
[訊息] 執行丟棄服務，若需結束，請手動關閉程式
```

執行 3 日期時間服務 (Daytime) 的執行結果如下。

```
C:\data\Code\Cpp\network_programming\cmake-build-debug\hw2.exe
+++++
| 請輸入服務的代號(1~5)以執行服務 |
| 1. 回應服務 (Echo) |
| 2. 丟棄服務 (Discard) |
| 3. 日期時間服務 (Daytime) |
| 4. 每日一句 (QUOTE) |
| 5. 字元產生器 (CHARGEN) |
+++++
3
+++++
| 執行結果 |
+++++
[傳送內容]

[傳送內容長度] 1bytes
[回應內容]
下午 12:34:15 2021/3/15

[回應內容長度] 24bytes

Process finished with exit code 0
```

執行 4 每日一句 (QUOTE)的執行結果如下。

```
C:\data\Code\Cpp\network_programming\cmake-build-debug\hw2.exe
+++++
| 請輸入服務的代號(1~5)以執行服務 |
| 1. 回應服務 (Echo) |
| 2. 丟棄服務 (Discard) |
| 3. 日期時間服務 (Daytime) |
| 4. 每日一句 (QUOTE) |
| 5. 字元產生器 (CHARGEN) |
+++++
4
+++++
| 執行結果 |
+++++
[傳送內容]

[傳送內容長度] 1bytes
[回應內容]
"The secret of being miserable is to have leisure to bother about whether
you are happy or not. The cure for it is occupation."
George Bernard Shaw (1856-1950)
[回應內容長度] 166bytes

Process finished with exit code 0
```

執行 5 字元產生器 (CHARGEN) 的執行結果如下。

```
C:\data\Code\Cpp\network_programming\cmake-build-debug\hw2.exe
+++++
| 請輸入服務的代號(1~5)以執行服務 |
| 1. 回應服務 (Echo) |
| 2. 丟棄服務 (Discard) |
| 3. 日期時間服務 (Daytime) |
| 4. 每日一句 (QUOTE) |
| 5. 字元產生器 (CHARGEN) |
+++++
5
+++++
| 執行結果 |
+++++
[傳送內容]

[傳送內容長度] 1bytes
[回應內容]
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefgh
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghi
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghij
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijk
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijkl
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklm
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmn
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmno
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnop
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopq
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqr
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrs
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijkl
[回應內容長度] 1024bytes

Process finished with exit code 0
```

執行時發生錯誤會提示錯誤訊息的結果如下。

```
C:\data\Code\Cpp\network_programming\cmake-build-debug\hw2.exe
+++++
| 請輸入服務的代號(1~5)以執行服務 |
| 1. 回應服務 (Echo) |
| 2. 丟棄服務 (Discard) |
| 3. 日期時間服務 (Daytime) |
| 4. 每日一句 (QUOTE) |
| 5. 字元產生器 (CHARGEN) |
+++++
1
+++++
| 執行結果 |
+++++
[錯誤] 無法連線，錯誤訊息：10061

Process finished with exit code 0
```

### 三、 程式碼

```
/*
 * NCYU 109 Network Programming HW2
 * Created by Linwebs on 2021/3/15.
 */

#include <iostream>
#include <winsock.h>

// 定義回傳字元最大數量
#define MAXLINE 1024

using namespace std;

/*
 * 執行完畢結果
 * int status          => 執行狀態
 * int send_len        => 傳送的内容長度
 * int recv_len        => 接收到的内容長度
 * string send         => 傳送的内容
 * string recv         => 接收到的内容
 */
```

```

struct content {
    int status;
    int send_len;
    int recv_len;
    string send;
    string recv;
};

void welcome_msg();

void choose_service();

void func_echo();

void func_discard();

void func_daytime();

void func_quote();

void func_chargen();

content *simpletcp_service(const char str[MAXLINE], const char
server_ip[16], u_short server_port);

void output(content *result);

int main() {
    welcome_msg();
    choose_service();
    return 0;
}

/*
 * 歡迎畫面
 */
void welcome_msg() {
    cout << "++++++" << endl;

```

```

cout << "| 請輸入服務的代號(1~5)以執行服務 |" << endl;
cout << "| 1. 回應服務 (Echo)          |" << endl;
cout << "| 2. 丟棄服務 (Discard)          |" << endl;
cout << "| 3. 日期時間服務 (Daytime)      |" << endl;
cout << "| 4. 每日一句 (QUOTE)           |" << endl;
cout << "| 5. 字元產生器 (CHARGEN)       |" << endl;
cout << "+++++" << endl;
}

/*
 * 選擇服務
 */
void choose_service() {
    char service;
    cin >> service;
    switch (service) {
        case '1':
            func_echo();
            break;
        case '2':
            func_discard();
            break;
        case '3':
            func_daytime();
            break;
        case '4':
            func_quote();
            break;
        case '5':
            func_chargen();
            break;
        default:
            cout << "[錯誤] 輸入的格式錯誤, 請重新輸入" << endl;
    }
}

/*
 * 執行 1 回應服務 (Echo)

```

```

*/
void func_echo() {
    content *result = simpletcp_service("I love NCYU CSIE",
    "127.0.0.1", 7);
    output(result);
}

/*
 * 執行 2 丟棄服務 (Discard)
 */
void func_discard() {
    cout << "[訊息] 執行丟棄服務，若需結束，請手動關閉程式" << endl;
    content *result = simpletcp_service("I love NCYU CSIE",
    "127.0.0.1", 9);
    output(result);
}

/*
 * 執行 3 日期時間服務 (Daytime)
 */
void func_daytime() {
    content *result = simpletcp_service("", "127.0.0.1", 13);
    output(result);
}

/*
 * 執行 4 每日一句 (QUOTE)
 */
void func_quote() {
    content *result = simpletcp_service("", "127.0.0.1", 17);
    output(result);
}

/*
 * 執行 5 字元產生器 (CHARGEN)
 */
void func_chargen() {
    content *result = simpletcp_service("", "127.0.0.1", 19);

```



```

    output(result);
}

/*
 * 執行完畢將結果輸出
 * content *result 執行結果指標
 */
void output(content *result) {
    cout << "+++++++" << endl;
    cout << "| 執行結果 |" << endl;
    cout << "+++++++" << endl;
    if (result->status == 0) {
        cout << "[傳送內容] " << endl;
        cout << result->send << endl;
        cout << "[傳送內容長度] " << result->send_len << "bytes" << endl;
        cout << "[回應內容] " << endl;
        cout << result->recv << endl;
        cout << "[回應內容長度] " << result->recv_len << "bytes" << endl;
    } else {
        cout << "[錯誤] 無法連線, 錯誤訊息: " << result->recv << endl;
    }
}

/*
 * 連線至 Simple TCP Service
 * const char str[MAXLINE]      => 傳送的内容
 * const char server_ip[16]     => 伺服器端 ip 位址 (ipv4 格式)
 * u_short server_port          => 伺服器端埠號
 *
 * 回傳 content 指標
 */
content *simpletcp_service(const char str[MAXLINE], const char
server_ip[16], u_short server_port) {
    SOCKET sd;
    struct sockaddr_in serv{};
    WSADATA wsadata;

    // send string

```

```

//const char str[MAXLINE] = "abc";

// receive string
char str_r[MAXLINE + 1] = "";

// echo server's ip address
//const char server_ip[16] = "127.0.0.1";

// echo server's port number
//u_short server_port = 7;

// receive bytes
int rec_len;

// connect status
int conn_status;

auto *result = new content;

// Call WSStartup() to Register "WinSock DLL"
WSStartup(0x101, (LPWSADATA) &wsadata);

// Open a TCP socket
sd = socket(AF_INET, SOCK_STREAM, 0);

// Prepare for connect.
// Include sockaddr_in struct (serv)
serv.sin_family = AF_INET;

// server's ip address
serv.sin_addr.s_addr = inet_addr(server_ip);

// server's port number
// htons: host to network
serv.sin_port = htons(server_port);

// connect to echo server
conn_status = connect(sd, (LPSOCKADDR) &serv, sizeof(serv));

```

```

if (conn_status == SOCKET_ERROR) {
    //cout << "connect function failed with error: " <<
WSAGetLastError() << endl;
    result->status = 1;
    result->recv = to_string(WSAGetLastError());
    closesocket(sd);
    WSACleanup();
    return result;
}

// send "how are you" to echo server
send(sd, str, int(strlen(str) + 1), 0);
// cout << "send: " << str << "(" << strlen(str) + 1 << " bytes)"
<< endl;

result->status = 0;
result->send = str;
result->send_len = int(strlen(str) + 1);

// receive echo server
rec_len = recv(sd, str_r, MAXLINE, 0);
if (rec_len == SOCKET_ERROR) {
    cout << "error";
}

// cout << "recv: " << str_r << "(" << rec_len << " bytes)" <<
endl;

result->recv = str_r;
result->recv_len = rec_len;
// close TCP socket
closesocket(sd);

// finish "WinSock DLL"
WSACleanup();

```

```
return result;  
}
```

#### 四、心得

這星期的課程，老師教我們使用 telnet 與 Windows 內建的 Simple TCP/IP Services 溝通，我在前一天晚上有稍微預習了一下課程的內容，在操作時發現 telnet 跳出的指令是 Ctrl+]，但由於 Windows 的注音輸入法會導致輸入 Ctrl+]時顯示全行的】符號，使其無法正確跳出，上網尋找許多解決方式，找到原因後，有人說使用嘸蝦米輸入法可解決問題，不過我後來想到的解決方式是切換鍵盤配置，將其改為英文(美國)的 US 鍵盤，就解決此問題了。