

## [2014 網路系統程式設計 Homework 8]

### ◆ 主旨：

Implement a multi-thread client-server project.

### ◆ 說明：

1. Here is the specification of a multi-thread client-server project.
2. General scenario:  
client connect, sends message. Message is put into a file with the name of the destination.
  - (1) Client-side. Client specifies a destination and a message. No code required – just use telnet. We'll have trust the clients to do it right, i.e. line1 is the destination, subsequent line the message.
  - (2) Server-side. The server screen has a menu
    - "1) Display number of current connections"
    - "2) Display statistics (average connect time etc.)"
    - "3) Re-start statistics gathering"
    - "4) Kill stale clients"
3. Implementation:  
Server consists of a number of separate threads:
  - (1) A port-listener: sits in a forever accept loop and fires up a servlet thread for each new client.
  - (2) A tidier and stats gatherer. This thread monitors the activity of all of the servlet threads. It also shuffles the array of data-structures representing these servlets. It commits completed messages to files, one per destination.
  - (3) Servlet threads (one per client).
4. Issues. Some mutex locking and semaphores will be needed to ensure communication between the various threads.
5. Pthreads: you may need to use the following:
  - (1) pthread\_attr\_init();
  - (2) pthread\_attr\_setdetachstate();

- (3) `pthread_cancel();`
- (4) `pthread_create();`
- (5) `pthread_mutex_lock();`
- (6) `pthread_mutex_unlock();`
- (7) `sem_wait();`
- (8) `sem_post();`

Consult the manual for details.

#### 6. Files provided :

- I. `collect_garb.c`
- II. `disconnect.c`
- III. `FuncSpec`
- IV. `get_stale.c`
- V. `list_conn.c`
- VI. `list_stats.c`
- VII. `listen_port.c`
- VIII. `Makefile`
- IX. `menu.c`
- X. `serve_client.c`
- XI. `sms.h`
- XII. `sms_server.c`
- XIII. `zap_servlet.c`
- XIV. `zap_stale.c`
- XV. `zero_stats.c`

#### 7. Data structures :

- (1) Servlet (door) is joined in a doubly linked list. It stores the current clients' information.
- (2) Stats (history) stores the offline client's information whether the client left correctly. It is a singly linked list and tend to be treated like stacks.
- (3) When a client left or was aborted, its servlet date will be removed from "Servlet list" and add its data into pending (pending\_stack) waiting for doing the rest things and getting into "history". Pending (pending\_stack) is also a singly linked list and tend to be treated like stacks.
- (4) Menu structure almost explains itself.

8. 注意：

sms.h 中，PORT number 請使用：

大一：51 + 學號後三碼 大二：

52 + 學號後三碼 大三：53 + 學

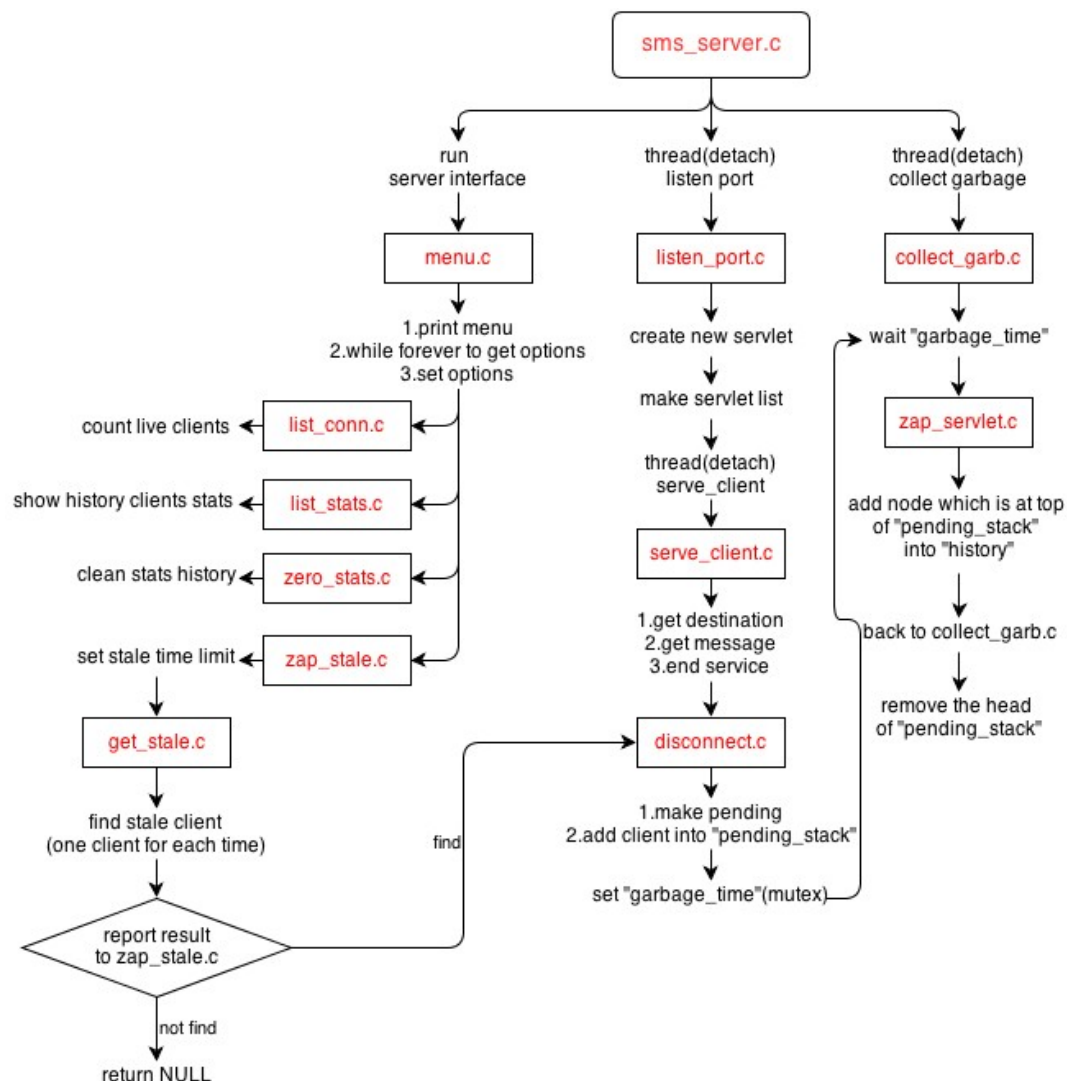
號後三碼 大四：54 + 學號後三

碼 碩一：61 + 學號後三碼 碩二：

62 + 學號後三碼 其他：63 + 學

號後三碼

9. 簡易流程圖，包含各個檔案的關係：



10. 執行結果如下：

The sample output for client is as follow

```
-bash-3.00$ telnet localhost 5678
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Destination : james
Now write your message: finish with ---
--> ---
Bye Bye!

Connection to localhost closed by foreign host.
-bash-3.00$ telnet localhost 5678
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Destination : bill
Now write your message: finish with ---
--> Sorry - time is up
Connection to localhost closed by foreign host.
-bash-3.00$ █
```

The sample output for client is as follows

```
-bash-3.00$ sms_server

    1) List number of current connections
    2) Summarise statistics
    3) Re-start statistics
    4) Zap stale clients and free memory

Please choose (1 - 4) : 2
No connections on record

    1) List number of current connections
    2) Summarise statistics
    3) Re-start statistics
    4) Zap stale clients and free memory

Please choose (1 - 4) : █
```

```
Please choose (1 - 4) : 1
There are 0 live connections
The oldest began 0 seconds ago
```

- 1) List number of current connections
- 2) Summarise statistics
- 3) Re-start statistics
- 4) Zap stale clients and free memory

```
Please choose (1 - 4) : 1
There are 1 live connections
The oldest began 16 seconds ago
```

- 1) List number of current connections
- 2) Summarise statistics
- 3) Re-start statistics
- 4) Zap stale clients and free memory

```
Please choose (1 - 4) : 2
1 connections 0.0% aborted, average connect = 1.0 seconds
average size = 0.0 bytes
```

- 1) List number of current connections
- 2) Summarise statistics
- 3) Re-start statistics
- 4) Zap stale clients and free memory

```
Please choose (1 - 4) : 1
There are 1 live connections
The oldest began 79 seconds ago
```

- 1) List number of current connections
- 2) Summarise statistics
- 3) Re-start statistics
- 4) Zap stale clients and free memory

```
Please choose (1 - 4) : 4
How many seconds counts as stale ? 60
Found a stale one
```

- 1) List number of current connections
- 2) Summarise statistics
- 3) Re-start statistics
- 4) Zap stale clients and free memory

```
Please choose (1 - 4) : 2
2 connections 50.0% aborted, average connect = 1.0 seconds
average size = 5.0 bytes

    1) List number of current connections
    2) Summarise statistics
    3) Re-start statistics
    4) Zap stale clients and free memory
```

◆ 限制：

1. 請在 Ubuntu 14.04 系統上使用 C 語言寫本次作業並進行測試，**Demo 時只接受助教指定作業伺服器上的執行結果。**
2. 嚴禁抄襲其他同學作業，參與者(**抄襲與被抄襲**)均以**零分**計算。
3. 請對你的程式碼有深入瞭解，demo 時助教會問。
4. 對題目有問題可以寄信問助教群(sp\_ta@net.nsysu.edu.tw)或是到實驗室(F5018)詢問，但不幫忙 debug。
5. **逾期以零分計算，不接受補交**，有問題請事先告知，Demo 時間會另外通知。

◆ 作業上傳：

1. 請壓縮成 zip 或 tar 的壓縮檔，並上傳至中山網路大學，作業命名規則為”**學號\_SP\_HW8**”，Example: M013040001\_SP\_HW8.zip。
2. 請於 **2014 年 12 月 02 日(週二) 23:59** 前上傳完畢，逾期以 零分計算，不接受補交，有問題請事先告知，再次強調，Demo 時間會另外通知。