[2016 Network System Programming Homework 8]

- **Motivation**

  Implement a multi-thread client-server project.

- **Description**

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();

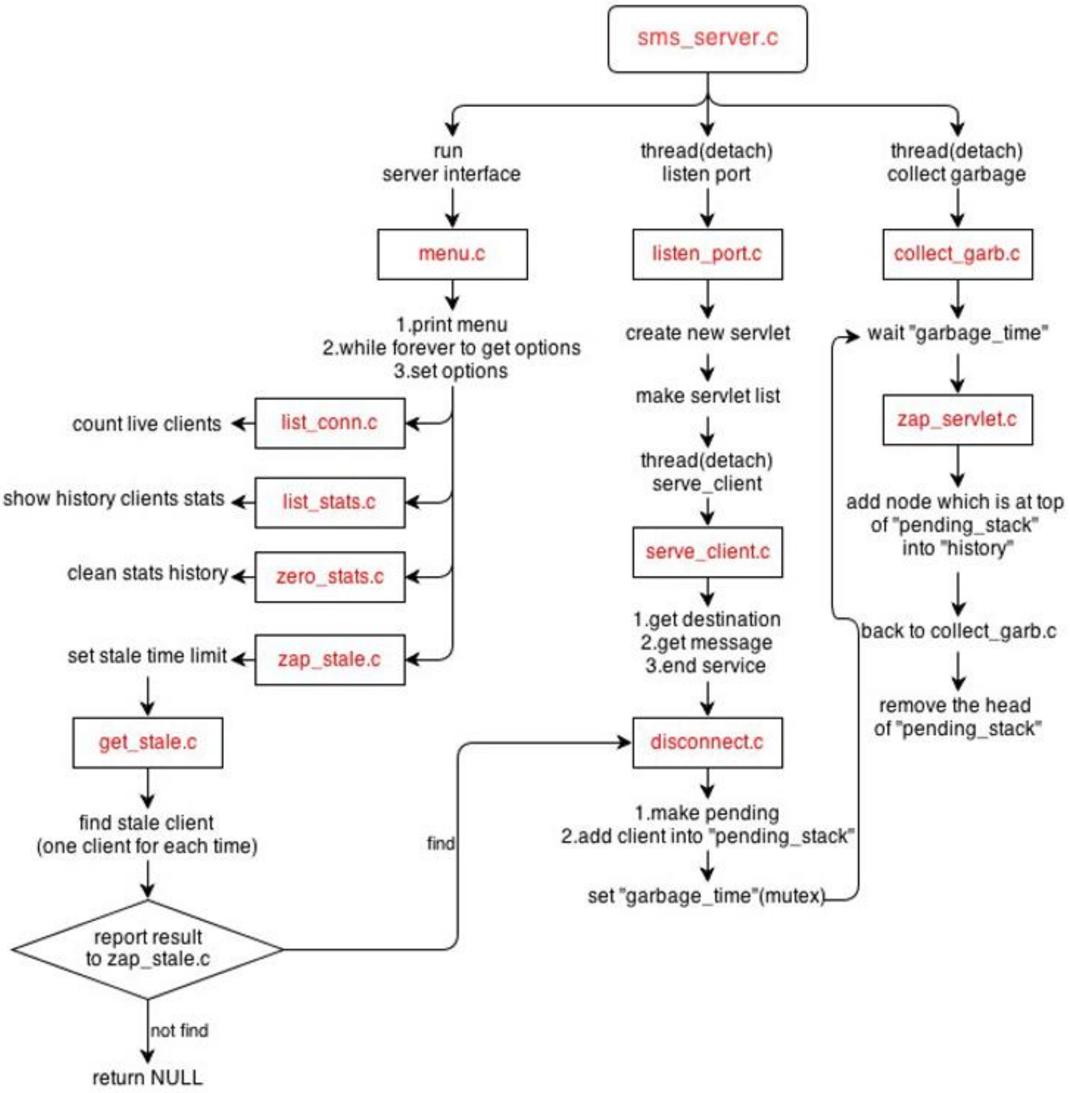
(2) pthread_attr_setdetachstate();
(3) pthread_cancel();
(4) pthread_create();
(5) pthread_mutex_lock();
(6) pthread_mutex_unlock();
(7) sem_wait();
(8) sem_post();
Consult the manual for details.

6. Files provided:
   collect_garb.c disconnect.c get_stale.c list_conn.c
   list_stats.c listen_port.c menu.c serve_client.c sms_server.c
   zap_servlet.c zap_stale.c zero_stats.c sms.h FuncSpec Makefile

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. Port number:
   In sms.h, the rules of port number are as follows.
   (1) Freshman: 51 + student ID last 3 numbers
   (2) Sophomore: 52 + student ID last 3 numbers
   (3) Junior: 53 + student ID last 3 numbers
   (4) Seniors: 54 + student ID last 3 numbers
   (5) Master Degree 1st year: 61 + student ID last 3 numbers
   (6) Master Degree 2nd year: 62 + student ID last 3 numbers

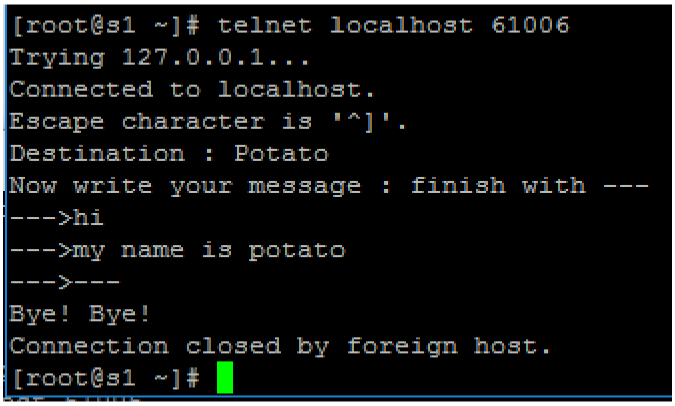
9. Flow chart:



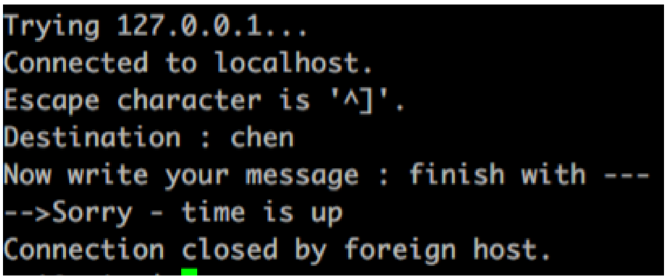
sms_server.c
run
server interface
thread(detach)
listen port
thread(detach)
collect garbage
menu.c
listen_port.c
collect_garb.c
1.print menu
2.while forever to get options
3.set options
create new servlet
wait "garbage_time"
make servlet list
list_conn.c
count live clients
list_stats.c
show history clients stats
zero_stats.c
clean stats history
zap_stale.c
set stale time limit
zap_servlet.c
thread(detach)
serve_client
serve_client.c
add node which is at top
of "pending_stack"
into "history"
back to collect_garb.c
1.get destination
2.get message
3.end service
remove the head
of "pending_stack"
get_stale.c
disconnect.c
find stale client
(one client for each time)
find
1.make pending
2.add client into "pending_stack"
set "garbage_time"(mutex)
report result
to zap_stale.c
not find
return NULL

10. Sample output:

I. Client:

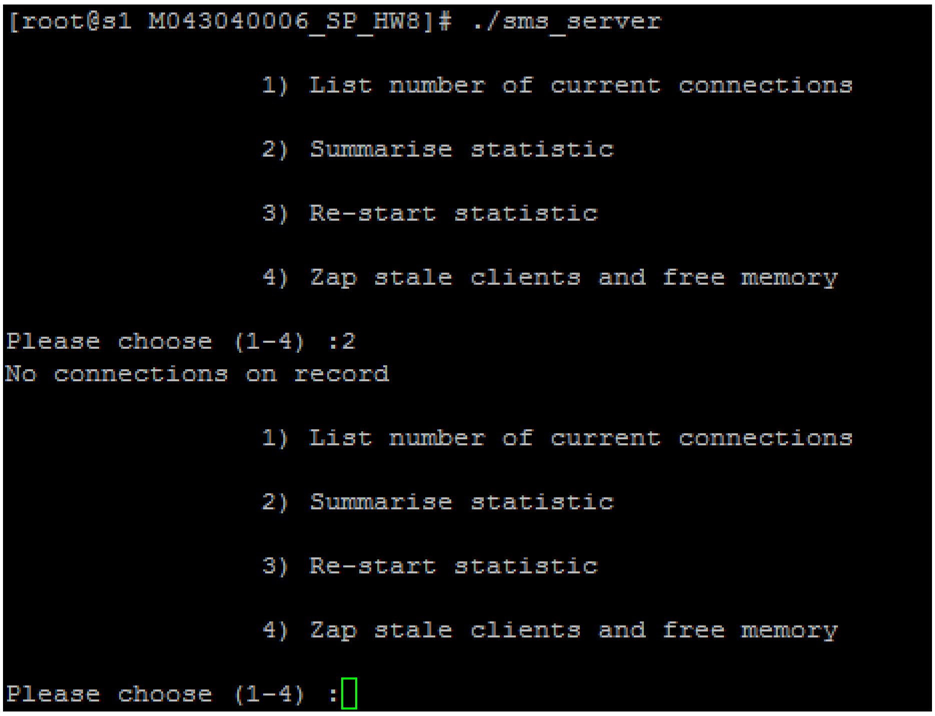


```
[root@s1 ~]# telnet localhost 61006
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Destination : Potato
Now write your message : finish with ---
--->hi
--->my name is potato
--->---
Bye! Bye!
Connection closed by foreign host.
[root@s1 ~]#
```

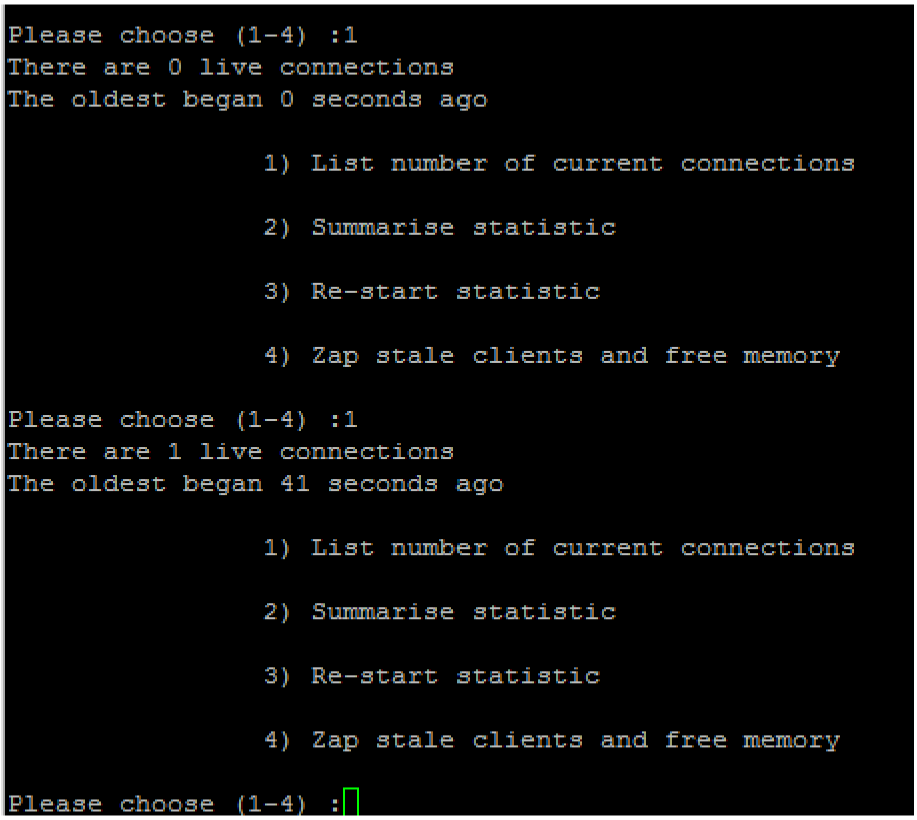


```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Destination : chen
Now write your message : finish with ---
-->Sorry - time is up
Connection closed by foreign host.
```

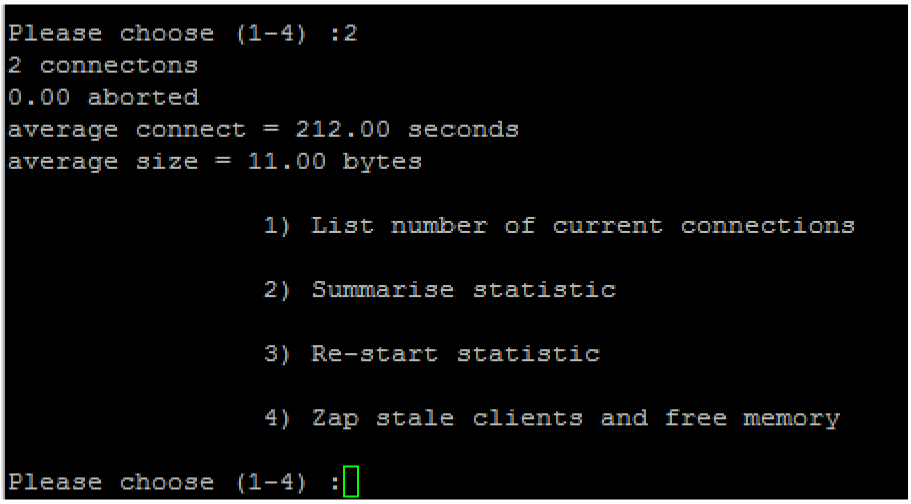
II. Server:



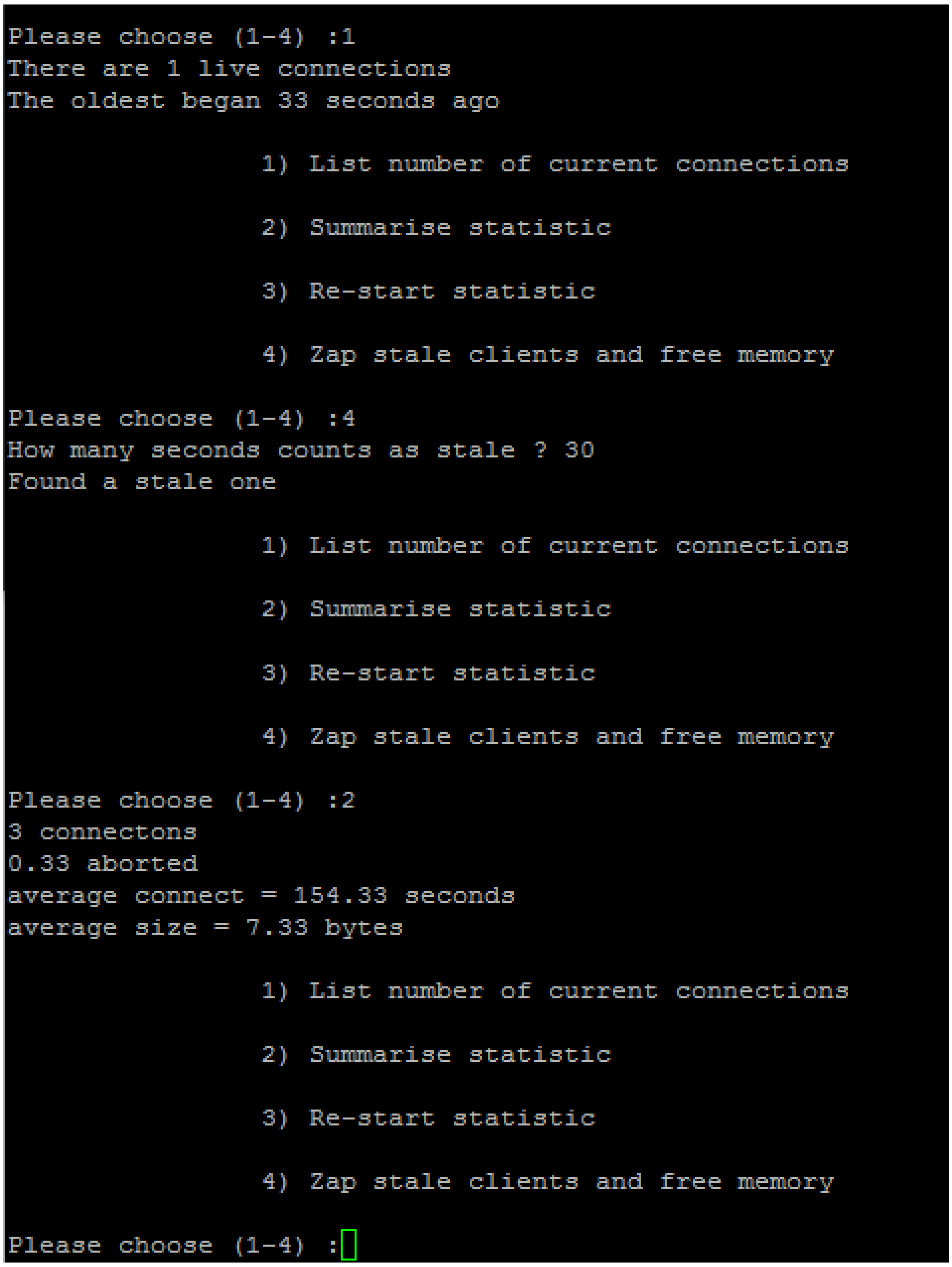
```
[root@s1 M043040006_SP_HW8]# ./sms_server

                1) List number of current connections

                2) Summarise statistic

                3) Re-start statistic

                4) Zap stale clients and free memory

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

                1) List number of current connections

                2) Summarise statistic

                3) Re-start statistic

                4) Zap stale clients and free memory

Please choose (1-4) :
```



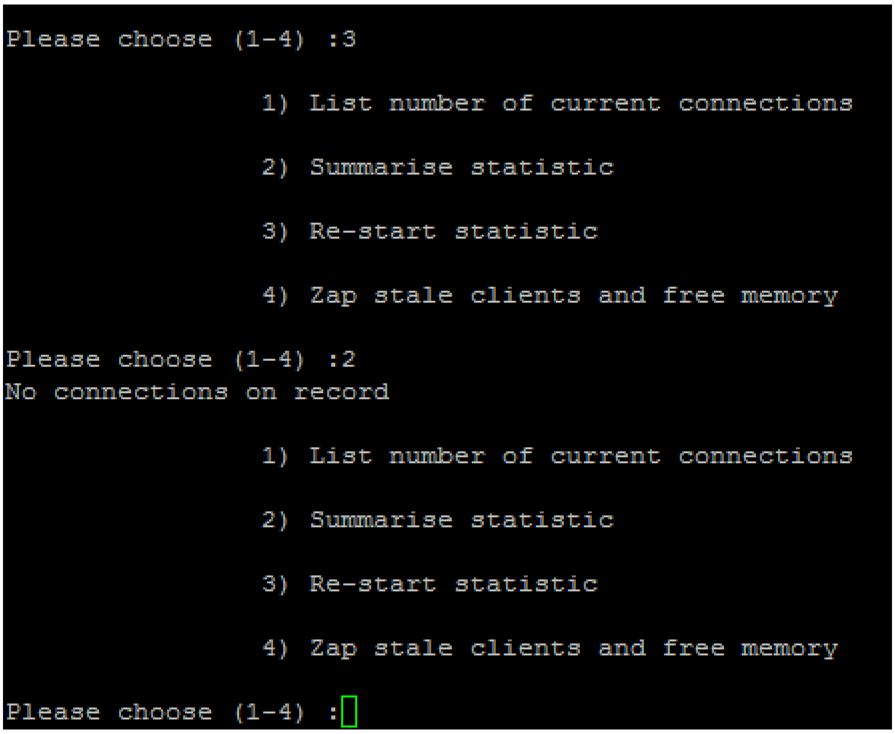
```
Please choose (1-4) :1
There are 0 live connections
The oldest began 0 seconds ago

                1) List number of current connections

                2) Summarise statistic

                3) Re-start statistic

                4) Zap stale clients and free memory

Please choose (1-4) :1
There are 1 live connections
The oldest began 41 seconds ago

                1) List number of current connections

                2) Summarise statistic

                3) Re-start statistic

                4) Zap stale clients and free memory

Please choose (1-4) :
```



```
Please choose (1-4) :2
2 connectons
0.00 aborted
average connect = 212.00 seconds
average size = 11.00 bytes

                1) List number of current connections

                2) Summarise statistic

                3) Re-start statistic

                4) Zap stale clients and free memory

Please choose (1-4) :
```



```
Please choose (1-4) :1
There are 1 live connections
The oldest began 33 seconds ago

                1) List number of current connections

                2) Summarise statistic

                3) Re-start statistic

                4) Zap stale clients and free memory

Please choose (1-4) :4
How many seconds counts as stale ? 30
Found a stale one

                1) List number of current connections

                2) Summarise statistic

                3) Re-start statistic

                4) Zap stale clients and free memory

Please choose (1-4) :2
3 connectons
0.33 aborted
average connect = 154.33 seconds
average size = 7.33 bytes

                1) List number of current connections

                2) Summarise statistic

                3) Re-start statistic

                4) Zap stale clients and free memory

Please choose (1-4) :
```



```
Please choose (1-4) :3

                1) List number of current connections

                2) Summarise statistic

                3) Re-start statistic

                4) Zap stale clients and free memory

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

                1) List number of current connections

                2) Summarise statistic

                3) Re-start statistic

                4) Zap stale clients and free memory

Please choose (1-4) :
```

額外上傳規定：

1. 本次作業期限為約一個月，但每個禮拜必須在禮拜二晚上(23:59)前上傳當週進度報告及程式碼至網路大學。
2. 進度報告須說明當週完成那些部分。
3. 進度報告格式不限。
4. 若有一週未繳交，則本作業扣總分 10 分。
5. 若提早完成，還是必須上傳。
6. 最後一週繳交完成的程式即可

規則：

1. 請使用 C 語言完成本次作業並在 Ubuntu 16.04 系統上進行執行、測試。
2. 必須上傳能編譯本次作業之 Makefile，內容不拘，未寫零分。
3. 請對你的程式碼有深入瞭解，demo 時助教會問。
4. 對作業有問題歡迎來信助教群（net_ta@net.nsysu.edu.tw）或是寄信約時間到實驗室（EC5018）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。
6. 嚴禁抄襲其他同學的作業，參與者(抄襲與被抄襲)均以零分計算。

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.zip”。

   Example: M043040006_SP_HW8.zip
2. 作業截止時間為 2017/01/09（Mon.）23:59，請在時間內上傳作業。