

[2024 Network System Programming Homework 12]

Rules:

1. Please use C language in this homework and run your program on Ubuntu 24.04.
2. Please provide **Makefile** to compile your homework.
3. Do not copy the others homework definitely.
4. If you have any question, please send email to sp_ta@net.nsysu.edu.tw or drop by Room EC5018. However, TA will not help you to debug program.

Turn in your homework:

1. Please compress your homework into **zip** archive.
2. Naming rules: "**SP_HW12.zip**".
3. Upload your homework (zip file) to NSYSU Cyber University (網路大學).
4. **Deadline: 2024/11/17 09:00. You cannot get any credit if you do not turn in your homework before the deadline.**

◆ 主旨：

本次作業著重於讓學生練習 shared memory 的 process synchronization。

◆ 題目：

1. The following assumes that you are using shared memory and IPC semaphores to communicate with a server. **But it is encouraged to use methods other than semaphores to achieve the same functionality.**
2. Use one of the semaphores for mutual exclusion, so that one client at a time talks to the server and the other is used for synchronization between client and server as follows:
 - I. It is the client's turn to talk when the synch semaphore value is 0. It is the server's turn when the value is non-zero.
 - II. The initial value is 0.
 - III. The server attempts to decrement and is blocked.
 - IV. The client increments the semaphore by 2 to alert the server and then waits until is 0 before looking for reply.
 - V. The server wakes up when the client increments the semaphore and does a look-up. It then attempts to decrement and wakes up the client.
3. The server should be named shm_server and is built from lookup5.c (the module which performs the dictionary look ups) and shm_server.c (which interfaces with the client through the shared memory and semaphore).
4. The client should be named shm_client and is built from main.c (the user interface module) and lookup6.c (which interfaces with the server through the shared memory and semaphores).
5. Edit the lookup6.c file to communicate using shared memory and synchronization by using two semaphores. Edit the shm_server.c file to wait on a semaphore, do look-up, use shared memory for communication, and notify the client using the second semaphore described above.

If you want to kill shared memory and semaphore, first using ipcs command to display IPC status, then using ipcrm -m shmid to kill shared memory and using ipcrm -s semid to kill semaphore.
6. Files provided:
 - I. dict.h
 - II. fixrec
 - III. lookup5.c
 - IV. lookup6.c
 - V. main.c
 - VI. shm_server.c
7. 執行結果如下：

```

f5018@XS35:~/HW6$ ./shm_server fixrec 0xbcde &
[1] 6839
f5018@XS35:~/HW6$ ipcs

----- 共用記憶體資料段 -----
鍵值      shmid      擁有者  perms      位元組  使用數  狀態
0x0000bcde 8912911      f5018   666         512      1
0x00000000 8945680      f5018   600        524288     2 目標
0x00000000 9207825      f5018   600        524288     2 目標
0x00000000 9240594      f5018   600        4194304    2 目標

----- 號誌陣列 -----
鍵值      semid      擁有者  perms      nsems
0x0000bcde 0          f5018   666          2

----- 訊息佇列 -----
鍵值      msqid      擁有者  perms      已用位元組  訊息
f5018@XS35:~/HW6$ ./shm_client 0xbcde
What word do you want : cynic
cynic : A blackguard who sees things as they are and not as they ought to be.
What word do you want : ambition
ambition : An overmastering desire to be vilified by enemies while living and made ridiculous by friends when dead.
What word do you want : ^C
f5018@XS35:~/HW6$

```