

## [2017 Network System Programming Homework5]

### Upload:

1. Please compress your homework into **zip** or **tar** archive.
2. Naming rules: "StudentID\_SP\_HW5.zip". For example:  
M053040086\_SP\_HW5.zip
3. Upload your homework to National Sun Yat-sen Cyber University.
4. **Deadline: 2017/11/24(Fri.) 23:59.**

### Note:

1. Please use C language in this homework and run your program on Ubuntu 16.04.
2. Please **provide Makefile** to compile your homework; otherwise, you will get ZERO.
3. **Do not copy homework of others. If it happened, you will get ZERO whether you are either the owner of the homework or the copycat.**
4. You have to deeply understand what your program do because TA will ask you something about your program during the demo.
5. If you have any question, please send email to sp\_ta@net.nsysu.edu.tw or come to EC5018, but TA does not help to debug.
6. If you do not submit your assignment on time, you will not hand in the delayed homework and get ZERO as well. If you have trouble, please advise in advance by email. Moreover, time and place for demo will be announced later.

Homework 5: Consider you are using a personal computer to create multiple virtual machines (VMs) to run tasks. When a VM has been done its work, it will be destroyed. This problem is to demonstrate the use of an IPC semaphore in coordinating VMs which require simultaneous multiple resource allocations, each requesting more than one of the resource. Allocate a semaphore to represent the resources of a personal computer with 8 CPU cores, 16GB RAM, and 500GB disk space to be spread across all concurrent VMs. VMs are created and destroyed on a first come, first served basis. A VM is created and goes only if there are an adequate number of cores, RAM and disk space to cover the VM, after all other VMs are accounted for. If the VM is a go, decrement the resource count, so that all resources in use are accounted for when the next VM is requested.

Define some VMs here. This following list specifies how many of each resource is requested per each VM. The values in this list cause contention: for example, the first VM uses 4 cores out of 8 available, making some of the others wait for a core. (cores, RAM, disk space): {(6, 5, 125), (2, 2, 500), (4, 5, 300), (2, 8, 125), (2, 8, 300)}. Please show the sequence of VMs to be created.