

Operating-system homework#3

Written exercises

• Chap.7

- 7.8: The Linux kernel has a policy that a process cannot hold a spinlock while attempting to acquire a semaphore. Explain why this policy is in place.
預防 deadlock，與效能問題。避免拿到 A 資源後又等待 B 資源，且 B 資源又必須等待 A 資源。

• Chap.8

- 8.20: In a real computer system, neither the resources available nor the demands of processes for resources are consistent over long periods (months). Resources break or are replaced, new processes come and go, and new resources are bought and added to the system.
 - If deadlock is controlled by the banker's algorithm, which of the following changes can be made safely (without introducing the possibility of deadlock), and under what circumstances?
 - (a) Increase Available (new resources added).
增加可用資源，不會造成 deadlock。
 - (b) Decrease Available (resource permanently removed from system).
如果減少的可用資源不會小於 process 的需求量，就不會造成 deadlock。
 - (c) Increase Max for one process (the process needs or wants more resources than allowed).
如果增加的 Max 沒有大於 Available，就不會造成 deadlock。
 - (d) Decrease Max for one process (the process decides that it does not need that many resources).
漸少 Max 不會造成大於 Available，不會造成 deadlock。
 - (e) Increase the number of processes.
只要增加後的 need 不會大於 Available，不會造成 deadlock。
 - (f) Decrease the number of processes
減少 process 不會大於 Available，不會造成 deadlock。
- 8.27: Consider the following snapshot of a system :

	Allocation	Max	Need
	A B C D	A B C D	A B C D
P0	1 2 0 2	4 3 1 6	3 1 1 4
P1	0 1 1 2	2 4 2 4	2 3 1 2
P2	1 2 4 0	3 6 5 1	2 4 1 1
P3	1 2 0 1	2 6 2 3	1 4 2 2
P4	1 0 0 1	3 1 1 2	2 1 1 1

– Use the banker's algorithm, determine whether or not each of the following states is unsafe. If the state is safe, illustrate the order in which the processes may complete. Otherwise, illustrate why the state is unsafe.

- (a) Available=(2,2,2,3)
P4: Available = (3,2,2,4)
P0: Available = (4,4,2,6)
- (b) Available=(4,4,1,1)
P2: Available = (5,6,5,1)
P4: Available = (6,6,5,2)

P1: Available = (4,5,3,8)	P1: Available = (6,7,6,4)
P2: Available = (5,7,7,8)	P0: Available = (7,9,6,6)
P3: Available = (6,8,7,9)	P3: Available = (8,9,6,7)

- 8.30: A single-lane bridge connects the two Vermont villages of North Tunbridge and South Tunbridge. Farmers in the two villages use this bridge to deliver their produce to the neighbor town.
 - The bridge can become deadlocked if a northbound and a southbound farmer get on the bridge at the same time. (Vermont farmers are stubborn and are unable to back up.)
 - Using semaphores and/or mutex locks, design an algorithm in pseudocode that prevents deadlock.
 - Initially, do not be concerned about starvation (the situation in which northbound farmers prevent southbound farmers from using the bridge, or vice versa)

```

semaphore mutex_biridge;
semaphore mutex_north_count;
semaphore mutex_south_count;

// north
while(true){
    wait(mutex_north_count);
    north_pass_count++;
    if (north_pass_count == 1){
        wait(mutex_biridge)
    }
    single(mutex_north_count);

    /* pass north bridge */

    wait(mutex_north_count);
    north_pass_count--;
    if (north_pass_count == 0){
        single(mutex_biridge)
    }
    single(mutex_north_count);
}

// south
while(true){
    wait(mutex_south_count);
    south_pass_count++;
    if (south_pass_count == 1){
        wait(mutex_biridge)
    }
    single(mutex_south_count);

    /* pass south bridge */

    wait(mutex_south_count);
    south_pass_count--;
    if (south_pass_count == 0){
        single(mutex_biridge)
    }
    single(mutex_south_count);
}

```

• Chap.9

- 9.15: Compare the memory organization schemes of contiguous memory allocation and paging with respect to the following issues:
 - ▶ (a) external fragmentation
 - paging: no external fragmentation
 - contiguous fix memory: no external fragmentation
 - contiguous variable partition memory: external fragmentation
 - ▶ (b) internal fragmentation
 - paging: internal fragmentation
 - contiguous fix memory: internal fragmentation
 - contiguous variable partition memory: no internal fragmentation
 - ▶ (c) ability to share code across processes
 - paging: can share code
 - contiguous: cannot share code

- 9.24: Consider a computer system with a 32-bit logical address and 8-KB page size. The system supports up to 1 GB of physical memory. How many entries are there in each of the following?
 - ▶ (a) A conventional, single-level page table
$$\frac{2^{32}}{2^{13}} = 2^{19}$$
 - ▶ (b) An inverted page table
$$\frac{2^{30}}{2^{13}} = 2^{17}$$