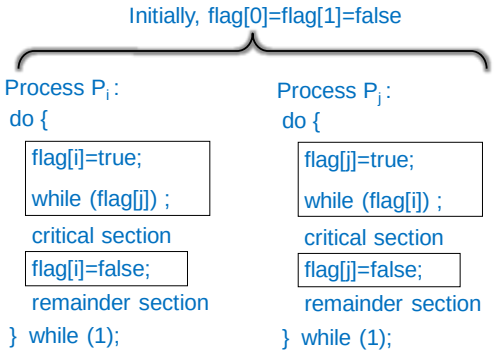
長庚大學108學年度第一學期作業系統期末測驗（滿分106）

系級: 姓名: 學號:

1. (8%) For the **second version** of **Peterson's Solution**, please explain the problem when we use the following algorithm to protect the critical sections of P_i and P_j:



Answer: When the two processes set flag[i] and flag[j] as true, no one can break the while loop of the enter section.

2. (8%) There are three processes:

   P_1: a * b → a
   P_2: a + c → a
   P_3: a + d → a

   P_1 should run before P_2 and P_3 do. The access to valuable "a" must be protected in a critical session. The order of P_2 and P_3 is arbitrary. We have only one semaphore S_1, and it is initialized as S_1=0. Now, the code of P_2 is provided as follows:

   wait(S_1)
   a = a + c;
   signal(S_1);

   Please provide the code of P_1 and P_3.

   Answer:

   P_1:

   a = a * b;
   signal(S_1);

   P_3:

   wait(S_1)
   a = a + d;
   signal(S_1);

3. (8%) The following code show the Readers and Writers Problem. In the Reader, please explain the meaning of (a) *if (readcount == 1){wait(wrt);}* and (b) *if (readcount== 0){signal(wrt);}*.

```
Initialization:
semaphore wrt=1;
semaphore mutex=1;
int readcount=0;

Writer:
    wait(wrt);
    ……
    writing is performed
    ……
    signal(wrt)
```

```
Reader:
    wait(mutex);
    readcount++;
    if (readcount == 1){wait(wrt);}
    signal(mutex);
    … reading…
    wait(mutex);
    readcount--;
    if (readcount== 0){signal(wrt);}
    signal(mutex);
```

Answer:

(a) When the first reader would like to use the shared resource, it has to check that no writer is using the shared resource.

(b) When the last reader finishes it access to the shared resource, it has to release the semaphore to let a writer (if there is any) use the shared resource.

4. (8%) Deadlock Prevention is to prevent the four necessary conditions of deadlock. Please explain the four necessary conditions: (a) mutual exclusion, (b) hold and wait, (c) no preemption, and (d) circular wait.

Answer:

(a) Only one process at a time can use a resource

(b) A process holding at least one resource is waiting to acquire additional resources held by other processes

(c) A resource can be released only voluntarily by the process holding it

(d) The waiting process chain forms a circle

5. (12%) Banker's Algorithm is a deadlock avoidance algorithm. Assume that there are 5 processes {P_0, P_1, P_2, P_3, P_4} and three types of shared resources {A, B, C} in the system, and the details are in the following table. (1) By Banker's Algorithm, is the system in a safe state? If your answer is yes, please provide a safe sequence. If your answer is no, please provide the reason. (2) Now, P_0 further has a request (1, 1 , 0) to use 1 more instance of A and 1 more instance of B. Should the request be granted? Again, provide the reason to support your answer.

	Allocation			Max			Need			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	0	1	0	7	5	3	7	4	3	3	3	2
P1	1	0	1	2	4	3	1	4	2			
P2	3	0	2	9	0	2	6	0	0			
P3	0	1	1	0	2	2	0	1	1			
P4	2	1	1	6	4	2	4	3	1			

Answer:

(1) Yes.

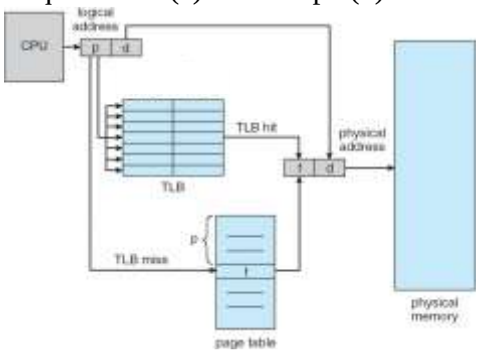
Run the Banker's Algorithm: Available(3, 3, 2) → P3 Need(0, 1, 1) → Available(3, 4, 3) → P1 Need(1, 4, 2) → Available(4, 4, 4) → P4 Need(4, 3, 1) → Available(6, 5, 5) → P2 Need(6, 0, 0) → Available(9, 5, 7) → P0 Need(7, 4, 3)

(2) No.

After the system grants the request: Available(3, 3, 2) → Available(2, 2, 2). P0 has Need(6, 3, 3) and Allocation(1, 2, 0).

We then run the Banker's Algorithm again: Available(2, 2, 2) → P3 Need(0, 1, 1) → Available(2, 3, 3) → No one can run now!

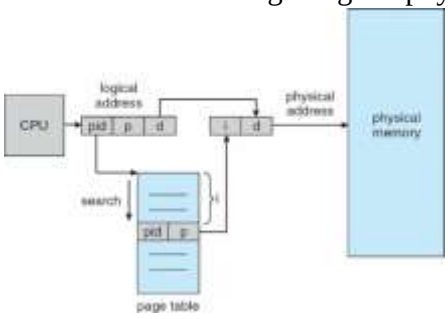
6. (8%) For memory management with page tables, considering the following figure, please answer the questions: (a) What is p? (b) What is f? (c) what is d? (d) What is the function of TLB?



Answer:
(a) Page number
(b) Frame number
(c) Offset
(d) TLB is some on-chip SRAM which can keep the frame numbers of recently referred pages so as to solve the "two memory accesses" problem of using page table.

7. (8%) There are 4 processes in a system, and the 4 processes would like to share 1 binary page. If each process has a page table, please provide the setting of the 4 page tables for the page sharing.
Answer:
Let each of the page tables have a node for mapping a page (of its virtual address) to the same physical memory frame (to be shared by the four processes).

8. (8%) For the inverted page table architecture, please briefly explain the mechanism of inverted page table architecture for getting the physical address.



Answer: It sequentially searches the pair (pid, p) in the page table. When the pair is found in the i-th entry of the page table, i is then used as the the frame number of the physical address, and d is the offset in the frame.

9. (10%) (a) If we use Contiguous Allocation with Dynamic Partitions to manage the main memory to serve the memory space requests of applications, is it possible to have Internal Fragmentation? You have to provide a reason to support your answer. (b) If we use Paging to manage the main memory to serve the memory space requests of applications, is it possible to have External Fragmentation? You have to provide a reason to support your answer.
Answer:
(a) No. Within this strategy, the allocated memory space to the application process is exactly equal to the required memory of the process.
(b) No. The paging strategy partitions the main memory into pages, and each page can be assigned to processes independently. Thus, there is no external fragmentation.

10. (12%) There is a system with only 3 memory frames. Given a reference string of pages {5→2→0→2→3→1→2→3→5→3}, please illustrate the page replacement of (a) the Least-Recently-Used (LRU) algorithm and (b) the First-In-First-Out (FIFO) algorithm. You should show the contents of memory frames and the LRU and FIFO queues.

Answer:

(a)

Memory Frame

5	5	5	5	3	3	3	3	3	3
	2	2	2	2	2	2	2	2	2
		0	0	0	1	1	1	5	5

LRU Queue

5	2	0	2	3	1	2	3	5	3
	5	2	0	2	3	1	2	3	5
		5	5	0	2	3	1	2	2

(b)

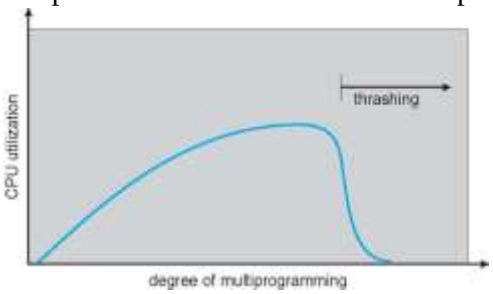
Memory Frame

5	5	5	5	3	3	3	3	5	5
	2	2	2	2	1	1	1	1	3
		0	0	0	0	2	2	2	2

FIFO Queue

5	2	0	0	3	1	2	2	5	3
	5	2	2	0	3	1	1	2	5
		5	5	2	0	3	3	1	2

11. (8%) In computer science, thrashing occurs when a computer's virtual memory subsystem is in a constant state of paging, rapidly exchanging data and binaries in memory for data and binaries on disk. (a) Please use the following figure to provide one possible reason for causing trashing. (b) Please provide a solution to address the problem is question (a).



Answer:

(a) When there are too many page faults, the CPU utilization could be low. However, when the CPU utilization is low, the scheduler might launch more processes for their execution, and it makes the CPU utilization even lower because there are page faults. That is thrashing.

(b) Operating systems should monitor the page-fault rate. If the page-fault rate is too high, scheduler should not load any new user processes despite the low CPU utilization.

12. (8%) For the Second-Chance Algorithm of page replacement, operating systems have to maintain a reference bit for each page. Please explain the Second-Chance Algorithm in detail.

Answer:

Whenever a page is accessed, the reference bit will be set as 1. When the algorithm has to select a victim page, it sequentially visits each memory frame and reset the reference bit to 0 if the reference bit is 1 before the visiting. If the reference bit of the currently visited frame is 0 before the visiting, the page in the frame is then selected as the victim page.