期末考評分規則_公告

1. (15%)

- 共 3 小題，每個小題各 5 分
  - 錯誤屬於觀念錯誤
    - 完全錯誤，扣 5 分
    - 如果是 a、b 小題，只計算出一個 process，扣 2 分
    - 如果是 b 小題，把 data、code section 跟 stack 當成連續空間計算，扣 2 分
    - 如果是 b 小題，沒有算出 data、code section 跟 stack 各自需要的「完整 page table」size，只算到 valid page 部分的 size，扣 2 分
  - 錯誤屬於計算錯誤，每有一個地方扣 1 分

參考答案

(a) 1-level

- page size=4 Kbytes=2^{12} bytes
  - 12 bits for page offset
  - **20** bits for page index
- each process needs 2^{20} pages → 2^{20} page entries are needed
  - 2^{20}×4 bytes (PTE size)=4MB for each process
- 10 processes → total 40MB are needed

(b) 2-level

- **10**-bit first-level page table field and a **10**-bit second-level page table field
  - first - level/outer page table size=2^{10}×4 bytes (PTE size)=4KB
  - in the same way, second - level/inner page table size=4KB
- 2-level paging
  - **1** first-level/outer page table is always needed
  - **3** second-level/inner page tables are needed: 2 for lowest 6MB, and 1 for the highest 12KB
  - total 16KB for each process
- 10 processes → total 160KB are needed

(c) inverted

- 1GB DRAM and page size is 4KB → total 1GB÷4KB=256KB pages

- page table size=256KB pages×4 bytes PTE size=1MB
- number of processes does not matter

2. (15%)

- a(5%)節省CPU time的浪費 (意思對即給分)
- b(10%)
    - 有問題參考講義 chaper 6 (p41-p42) 少一項扣一分

    - (5%) wait(X)
        - X.value–
        - if (X.value < 0)
        - add this process to the waiting queue
        - block()
    - (5%) signal(X)
        - X.value++
        - if (X.value <= 0)
        - remove a process P from the waiting queue
        - wakeup(P)

3. (10%)

Please describe why the copy-on-write (COW) method can reduce the time of forking a child process, compared to the traditional fork method?

- 有寫到這點就給分
    - (10%) No need to copy memory pages from parent to child (意思對即給分)
- 解釋COW也給分

4. (10%)

- (10%) 解釋、列式皆正確，答案正確(P=8*10^-6)
- (9%)解釋、列式皆正確，沒算完or計算錯誤(P=8*10^-6)
- (7%)1個條件的解釋(列式)有錯誤
- (4%)2個條件的解釋(列式)有錯誤
- (0%)3個條件的解釋(列式)都錯誤，或沒寫
- 沒說明假設變數，再扣1分(0分扣完為止)

- 時間單位沒有說明或是有誤，再扣1分(0分扣完為止)

5. (10%)

5. [10%] In a system with 3 page frames, what is the minimum number of page faults for a page replacement strategy for the following page reference string. Please explain your answer.

1, 2, 3, 4, 5, 3, 4, 1, 6, 7, 8, 7, 8, 9, 7, 8, 9, 5, 4, 2.

- (10%) 必須寫明過程，否則不給分
  - 非 optimal 策略 -> -5
    - 因為其他策略的結果不能保證結果為 minimum
  - 分段給分
    - 除前三個 page fault 以外，每個 page fault 各一分
      - 只計算到第一個錯誤的前一步（假如第八個 access 的 1 的結果出錯，之後的 678789789542 都視為錯誤）
    - 過程正確但算錯個數 -> -1
    - 少算前三個 -> -1

Optimal (Miss)																				
Access	1	2	3	4	5	3	4	1	6	7	8	7	8	9	7	8	9	5	4	2
Frame 1	1	1	1	1	5	5	5	5	5	5	5	5	5	9	9	9	9	5	4	2
Frame 2		2	2	4	4	4	4	4	4	4	8	8	8	8	8	8	8			
Frame 3			3	3	3	3	3	1	6	7	7	7	7	7	7	7	7			
Page Fault	v	v	v	v	v			v	v	v	v			v				v	v	v
FIFO (Miss)																				
Access	1	2	3	4	5	3	4	1	6	7	8	7	8	9	7	8	9	5	4	2
Frame 1	1	1	1	4	4	4	4	4	6	6	6	6	6	9	9	9	9	9	9	2
Frame 2		2	2	2	5	5	5	5	5	7	7	7	7	7	7	7	7	5	5	5
Frame 3			3	3	3	3	3	1	1	1	8	8	8	8	8	8	8	8	4	4
Page Fault	v	v	v	v	v			v	v	v	v			v				v	v	v
LRU (Hit, Miss)																				
Access	1	2	3	4	5	3	4	1	6	7	8	7	8	9	7	8	9	5	4	2
Frame 1	1	1	1	4	4	4	4	4	4	7	7	7	7	7	7	7	7	5	5	5
Frame 2		2	2	2	5	5	5	1	1	1	8	8	8	8	8	8	8	8	4	4
Frame 3			3	3	3	3	3	3	6	6	6	6	6	9	9	9	9	9	9	2
Page Fault	v	v	v	v	v			v	v	v	v			v				v	v	v

6. (15%)

說明

- 1.Mutual Exclusion 互斥性
  - If process P is executing in its critical section, then **no other** processes can be executing in their critical sections.
  - 同一時間，最多一個人在 critical section (<=1)
- 2.Progress 有進展
  - If no process is executing in its critical section and **some processes wish** to enter their critical sections, then only those processes that are not executing in their remainder section can participate in deciding which will enter its critical section next, and this selection **cannot be postponed indefinitely**.
  - 如果有些人想進，且目前 critical section 內沒人，OS 會從這些想進的人中選出一個人進去，不會一直不選。
- 3.Bounded Waiting 公平性
  - There exists **a bound**, or limit, on the **number of times(次數) "other" processes** are allowed to enter their critical sections after a process has made request to enter its critical section and before that request is granted.
  - 不能發生: "永遠只有某些人被選到，另一個人永遠不被選" 的情況。
    如果大家都不會被選，仍可能符合公平性，因為沒有 "只選某些人"。
- 同學們亦可參考 moodle 上課錄影: 20221125 0837-4 與 20221202 0613-1

答案

Does the codes satisfy all the **three requirements** of the critical section problem?
Explain your answer.

1. Mutual Exclusion (M.E.): Yes。這個寫法，當有一個人進入時，另一個人一定進不去，因為要離開 critical section 時，才會把自己的 flag 設成 false。
2. Progress (P.): No。這個寫法，可能會出現兩個人永遠都進不去的情況，例如 process Pi 執行完 flag[i] = true; 之後馬上換 process Pj 執行 flag[j] = true; 之後就兩個人都再也進不去。
3. Bounded Waiting (B.W.): Yes。這個寫法 Pi 如果能進去，且 Pj 正在等，下次一定換 Pj；如果出現兩個人都再也進不去的情形，那也一定是「兩個人都」進不去，所以還是算有符合公平性。

評分標準

題目有講明有三個 Requirements, 且要求 explain，符合也應解釋為何符合。
Mutual Exclusion / Progress / Bounded Waiting ; 各自獨立算 (各5%)

(1%) (1). 有寫對 Requirement 的名字 (e.g., Progress算對/Prograss算錯, 可寫中文名):
(1%) (2). (1)或(3)寫對的前提下，Yes/No 寫對
(3%) (3). 解釋為什麼考卷中的這段程式碼，符合或不符合

(3) 的回答內容，對於題目中程式碼的分析解釋，要符合該Requirement的定義要求。

– -對Requirement定義的理解錯誤: -2 (只講對部分 -1)

– -對程式碼本身理解錯誤or沒解釋程式碼: -1

– -如果有解釋程式碼，但跟要判斷的條件沒關聯 (例如，寫符合 Bounded Waiting，但 解釋符合的原因看起來是在解釋 Progress 的定義)，會 -3。

7. (10%)

[10%] Consider the following set of processes. For the priority number, a smaller number indicates a higher priority.

Process	Burst Time	Arrival Time	Priority
P1	10	0	1
P2	6	2	2
P3	3	4	3
P4	2	6	4

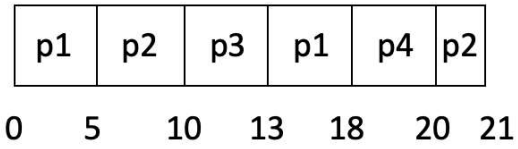
(a) [5%] Draw the Gantt chart illustrating the execution of these processes using RR (time quantum = 5), and list the **turnaround** time of each process.

(b) [5%] Draw the Gantt chart illustrating the execution of these processes using SRTF (Shortest Remaining Time First), and list the **waiting** time of each process.

- RR

進入時間	0	2	4	5	6	10
Process	p1	p2	p3	p1	p4	p2

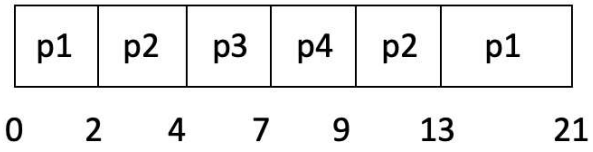
Gantt chart :



Process	Turnaround time
P1	18-0=18
P2	21-2=19
P3	13-4=9
P4	20-6=14

- SRTF

  Gantt chart :



Process	Waiting time
P1	(0-0) + (13-2) = 11
P2	(2-2) + (9-4) = 5
P3	(4-4) = 0
P4	(7-6) = 1

...

- 評分標準
  - Gantt chart : 2%
  - list turnaround time/waiting time for each process : 3%
- 使用到priority，則得0分。
- 第一個就畫錯，得0分
- turnaround time/waiting time for each process 正確答案，Gantt chart 正確，得 5 分
- turnaround time/waiting time for each process 未全對，Gantt chart 正確，得 2 分
- turnaround time/waiting time for each process 全錯，Gantt chart 錯一個地方扣一分，如果因為前面錯誤才因此導致後面的錯誤，那扣分只會扣在前面錯誤上，至多得1分。

8. (10%)

什麼是 processor affinity? 為甚麼load balancing通常抵銷了processor affinity帶來的好處?

- processor affinity: 維持一個process跑在相同的proceesor (4%)

- Load balancing 通常會進行process migration的動作, 但是 Processor affinity 會試著不要 migrate process(6%)

- 如果第一小題介紹affinity有提到在固定的processor執行相關的，第二小題load balancing有清楚寫到migration相關，第二小題也算對。

9. (5%)

分別解釋Page Fault 和 Segmentation Fault。
有寫到其中一個敘述即有分。
有錯誤的觀念或敘述再扣1~2分。

- Page Fault (3%)
  - valid access but not-in-memory ⇒ OS brings page to memory
- Segmentation Fault (2%)
  - invalid access ⇒ process abort/exception