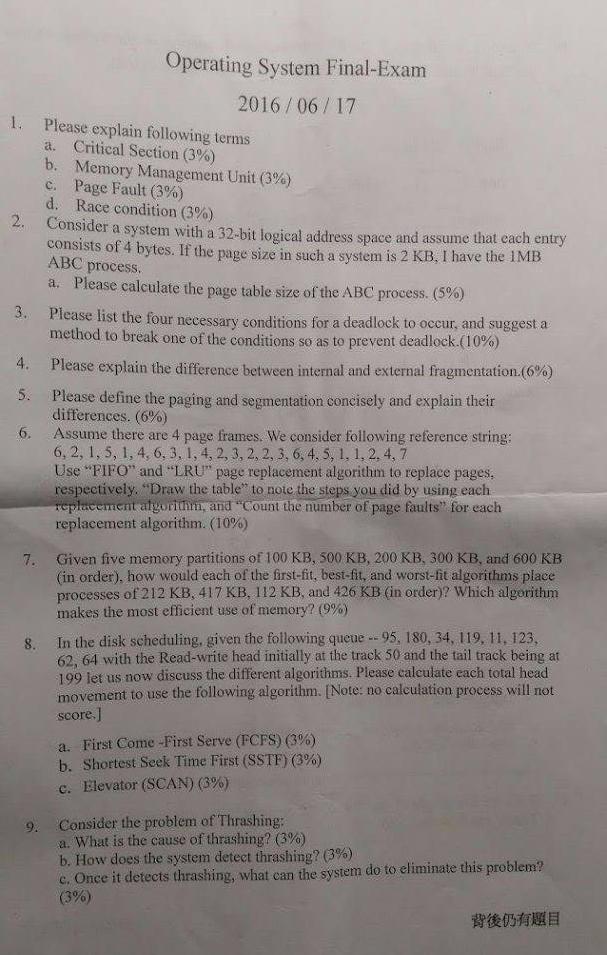
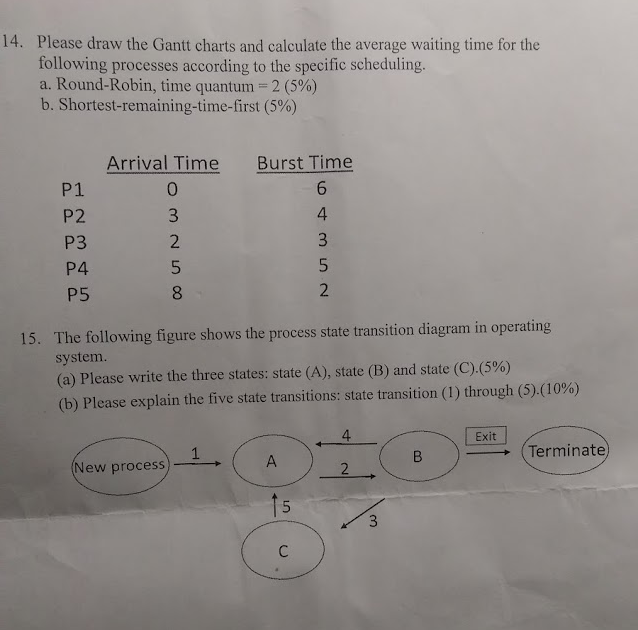
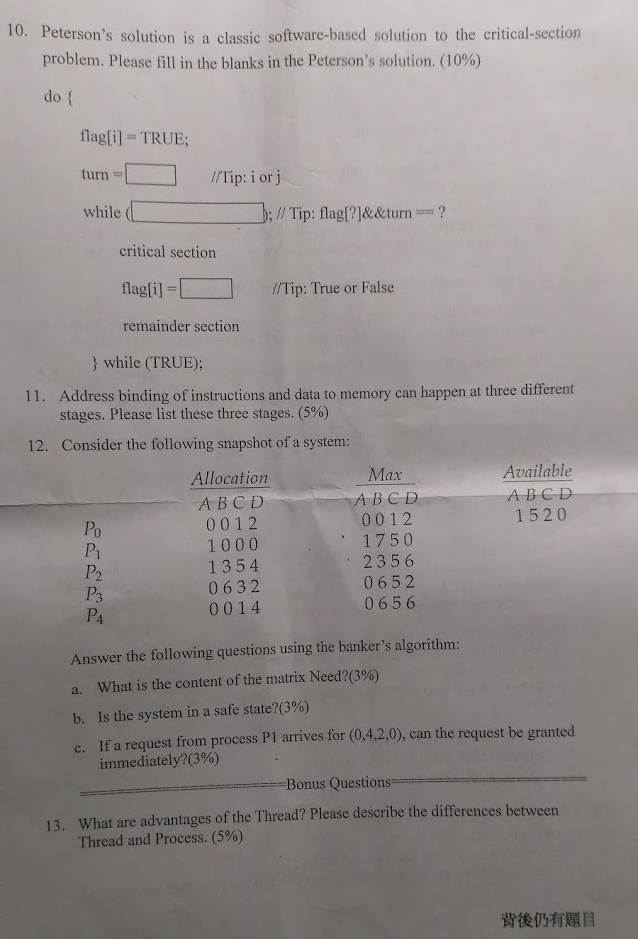
2016 Operating System Final (CLD)





# Answers

* 1. 當一個 process 在 critical section 時，其他 processes 不能也在 critical section，必須等待。
  2. 將虛擬記憶體位置對應到實體記憶體位置的硬體裝置。
  3. 在轉換 page 與對應的 frame 時，若 page table 中目標 page 為 invalid，則會對 OS 發出 trap，需要從 backing store 更新 page。
  4. 如果電腦中的兩個行程同時試圖修改一個共享記憶體的內容，在沒有並行控制的情況下，最後的結果依賴於兩個行程的執行順序與時機，有機會發生不正確的結果。

1. KB.

process page數量＝1MB/2KB=2^9個pages

所以這process的page table有2^9個entry -> page table size=2^9\*4(bytes)=2KB

* + 1. Mutual Exclusion
    2. Hold and Wait
    3. No Preemption
    4. Circular Wait
  1. Deadlocks Prevention
     1. For hold and wait, when a process request resources, OS 確保它手上沒有任何 resource
     2. For no preemption, 讓 process 之前可以互相 allocate resource。但有可能造成 starvation。
     3. For circular wait, 將每個 resource 都編上編號，規定每個 process 需要依照 resource 編號遞增的方式 request，若任一 process request 編號較小的資源，要先釋放手上編號大於目標 resource 的資源。

| External Fragmentation | Internal Fragmentation |
| --- | --- |
| * 記憶體中的 available space (hole) 總和雖然大於 process request，但因這些空間不連續，無法配置，造成使用率降低。 * 由連續記憶體配置引起。 * 可以透過 paging 解決。 | * 系統配置給某個 process 的空間稍大於該 process 所需空間，造成浪費。 * 由 page size 太大引起。 * 可以透過 segmentation 解決。 |

| Paging | Segmentation |
| --- | --- |
| * 將 process 配置到任何 available 的實體記憶體，使得記憶體不需要連續配置。 * 每個 page 與 frame size 相同。 * 有機會造成 internal fragmentation。 * 與使用者對記憶體配置管理的看法不一致。 | * Paging 的尺寸不同版。 * 使記憶體的虛擬配置看法與使用者一致。 * 每個 segment size 不同。 * Segmentation Table 需要儲存 segment limit (用於 check offset < limit)。 |



| FIFO (# of page fault = ) | | | | | | | | | | | | | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 6 | 2 | 1 | 5 | 1 | 4 | 6 | 3 | 1 | 4 | 2 | 3 | 2 | 2 | 3 | 6 | 4 | 5 | 1 | 1 | 2 | 4 | 7 |
| 6 | 6 | 6 | 6 |  | 4 | 4 | 4 | 4 |  | 2 |  | | | | | 2 | 2 |  | | | | 2 |
|  | 2 | 2 | 2 | 2 | 6 | 6 | 6 | 6 | 4 | 4 | 4 |
|  |  | 1 | 1 | 1 | 1 | 3 | 3 | 3 | 3 | 5 | 5 |
|  |  |  | 5 | 5 | 5 | 5 | 1 | 1 | 1 | 1 | 7 |
| LRU (# of page fault = ) | | | | | | | | | | | | | | | | | | | | | | |
| 6 | 2 | 1 | 5 | 1 | 4 | 6 | 3 | 1 | 4 | 2 | 3 | 2 | 2 | 3 | 6 | 4 | 5 | 1 | 1 | 2 | 4 | 7 |
| 6 | 6 | 6 | 6 |  | 4 | 4 | 4 |  | | 4 |  | | | | 4 |  | 4 | 4 |  | 4 |  | 4 |
|  | 2 | 2 | 2 | 2 | 6 | 6 | 2 | 2 | 5 | 5 | 5 | 7 |
|  |  | 1 | 1 | 1 | 1 | 1 | 1 | 6 | 6 | 6 | 2 | 2 |
|  |  |  | 5 | 5 | 5 | 3 | 3 | 3 | 3 | 1 | 1 | 1 |

* 1. First-Fit

| Action | Partitions Result | | | | |
| --- | --- | --- | --- | --- | --- |
| Allocate process KB to partition KB. |  |  |  |  |  |
| Allocate process KB to partition KB. |  |  |  |  |  |
| Allocate process KB to partition KB. |  |  |  |  |  |
| Process KB must wait for partition space. |  | | | | |

* 1. Best-Fit

| Action | Partitions Result | | | | |
| --- | --- | --- | --- | --- | --- |
| Allocate process KB to partition KB. |  |  |  |  |  |
| Allocate process KB to partition KB. No Kk |  |  |  |  |  |
| Allocate process KB to partition KB. |  |  |  |  |  |
| Allocate process KB to partition KB. |  |  |  |  |  |

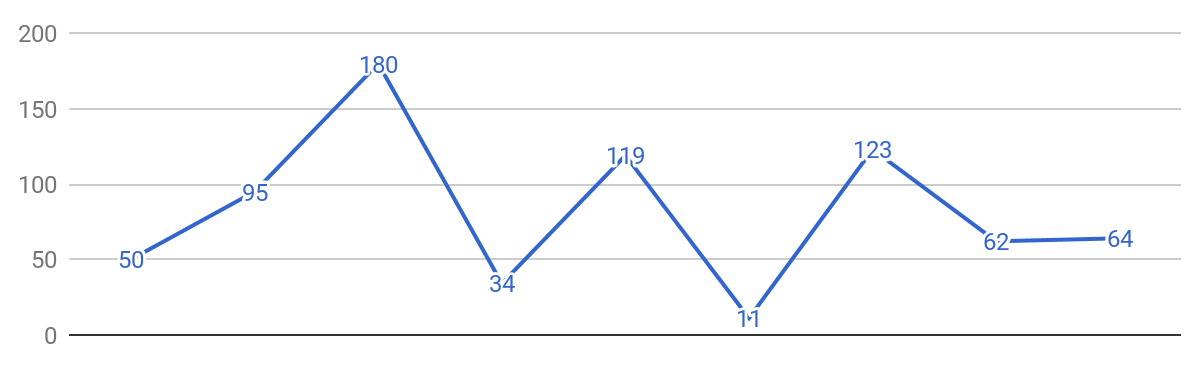
* 1. Worst-Fit

| Action | Partitions Result | | | | |
| --- | --- | --- | --- | --- | --- |
| Allocate process KB to partition KB. |  |  |  |  |  |
| Allocate process KB to partition KB. |  |  |  |  |  |
| Allocate process KB to partition KB. |  |  |  |  |  |
| Process KB must wait for partition space. |  | | | | |

Best-Fit 的記憶體使用率最高。

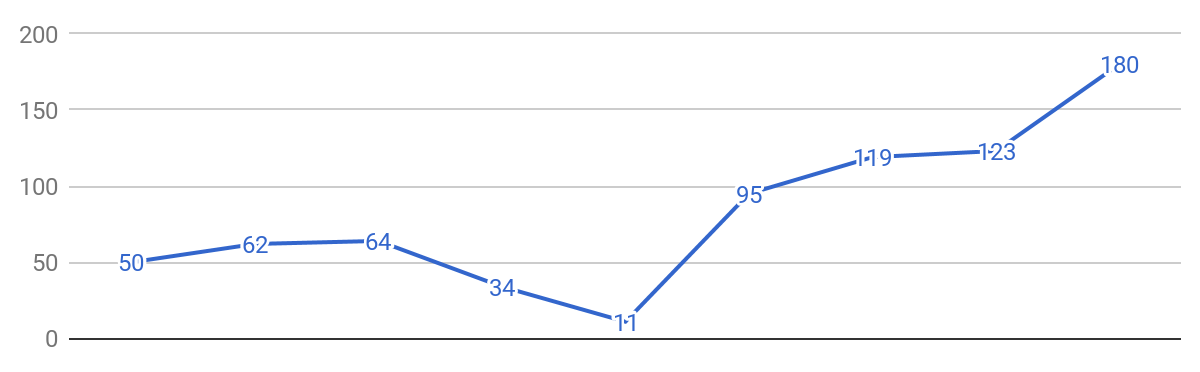
1. 1. FCFS

.



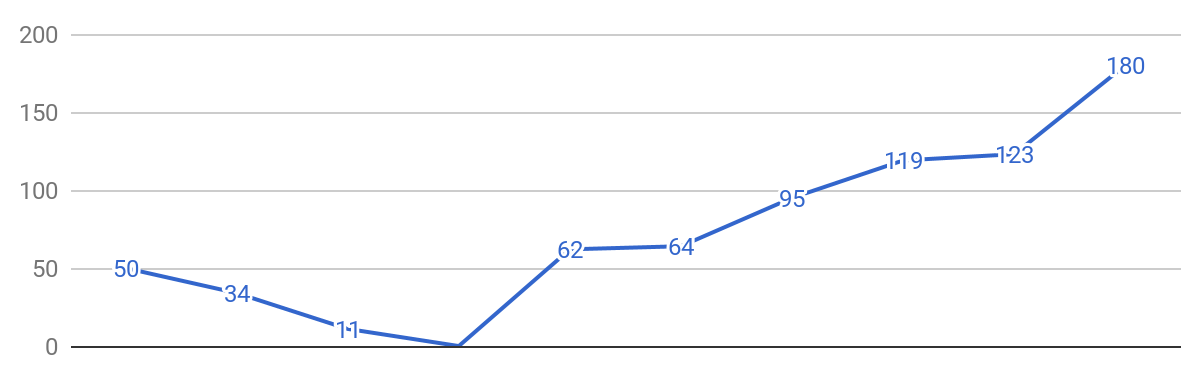
* 1. SSTF

.



* 1. SCAN

.



* 1. 因為 process 沒有分到足夠的 frame，因此會常常 page fault 而使得 CPU 使用率低，OS 試圖調高degree of multiprogramming 來提高CPU使用率，但反而因為 process 增加，每個 process 分到的frame 更少，CPU 使用率更低，惡性循環。

Total size of locality > total memory size.

* 1. OS 追蹤 CPU 使用率、page-fault rate，由 working-set model 或 page-fault frequency 來決定是否已經產生 thrashing。
  2. 延遲部分 processes 的 frame allocation、降低部分 processes 的 frame 的數量或減少 degree of multiprogramming。
  3. j;
  4. flag[j] && turn == j
  5. FALSE;

1. 1. Compile Time
   2. Load Time
   3. Execution Time

| 0 | 0 | 0 | 0 |
| --- | --- | --- | --- |
| 0 | 7 | 5 | 0 |
| 1 | 0 | 0 | 2 |
| 0 | 0 | 2 | 0 |
| 0 | 6 | 4 | 2 |

* 1. Yes.  
     可以找到 safe sequence 。
  2. Yes.  
     因假設接受此 request 後，仍可以找到 safe sequence ，不會進入unsafe state，故可接受此request。