12/10-12/17 回家作業

11.1

Answer:
在單用戶環境中，I/O 序列通常是空的。請求通常來自有單區塊或單一連續區塊的單一行程。在這種情況下，FCFS 是一種適合的硬碟調度方式。但 LOOK 幾乎同樣易於編程，並且在多行程執行、併行 I/O 時會提供更好的效能，例如當 OS 正在分頁時，網頁瀏覽器在後台回傳資料而且另一個應用程式在前景活動。

11.2

Answer:
磁碟中心到其他所有磁軌的距離最短。因此，磁碟磁頭傾向遠離磁碟邊緣。
另一種思考方式。磁頭所在的位置將磁盤圓形分割兩組。如果磁頭不在磁盤中心並且有新請求傳達，則新請求更有可能在包含磁盤中心的組中；因此，頭部更有可能朝那個方向移動。

11.3

Answer:
大多數磁碟都不會對主機輸出旋轉位置訊息，即使輸出了，到達排程器的時間也不准，另外排程器的消耗時間也會變化，所以旋轉位置訊息會是錯誤的。此外，磁碟請求通常以邏輯區塊來表示，又邏輯區塊與實體區塊間的對應非常複雜。

11.8

Answer:
可以。RAID 1 可以實現更高的讀取請求效能。當進行讀取時，RAID 1 系統可以決定應該存取兩區塊間的哪一個區塊來滿足需求。它可以根據磁盤磁頭的當前位置選擇，因此可以通過選擇更接近目標資料的磁盤磁頭來最佳化效能。

Operating System: Test 2 (12/17)

1. 請詳細說明並比較記憶體分頁與分段的差異? P424

Answer:

1. 分頁大小是固定，分段大小是可變的
2. 分頁會造成內部斷裂的問題 ，分段則會造成外部斷裂的問題
3. 分段系統的共享與存取控制較分頁系統容易

2. 造成輾轉的原因是什麼? 系統如何檢測出輾轉?一旦他檢測出輾轉，系統能做什麼以消除此問題?

Answer:

1. 輾轉現象(Thrashing)非常頻繁的分頁替換行為，如果某個行程花在分頁上的時間比花在執行上的時間還多，就是輾轉狀態。
2. 當提高多元程式規劃執行時，反而 CPU 使用率並沒有跟著提高就是發生輾轉現象(Thrashing)
3. 為了防止輾轉現象發生，必須要能提供行程所需要的一切的欄。可以使用工作組(Working-Set)的方法來觀察實際上一個行程再使用那些頁開始，並定義了行程執行中的局部區域模式(locality model)

3. 請說明何謂虛擬記憶體並列出其優點?

Answer:

虛擬記憶體是一種允許行程可以不必完全存在記憶體中的情況下被執行的技術。其優點為就是使用者程式可以比實體記憶體大。並且它將記憶體抽象成一個非常大且一致的儲存陣列，當使用者從實體記憶體觀察他是和記憶體分開的這項技術免除程式設計師對記憶體儲存限制的憂慮。

4. 解釋為什麼像是 IOS 和 Android 等行動作業系統不支援置換。

Answer:

行動裝置通常使用快取記憶體而非空間較大的硬碟當成持續性的儲存裝置。造成空間的限制是為什麼行動作業系統的設計者避免置換一項原因。其它的原因包含快取記憶體在超過容忍的寫入次數限制後會變成不可靠，和這些裝置的主記憶體和快取記憶體間不良的生產量。

5. 甚麼是"連續記憶體配置"? 甚麼是"記憶體外部斷裂(External fragmentation)"? 甚麼是"記憶體內部斷裂(Internal fragmentation)"? 有方法解決嗎?

Ans.

程式載入主記憶體內，它必須佔有連續位址的記憶體。

1. 因為行程持續地被載入與置換，使得可用的記憶體空間被分割成許多不連續的區塊。雖然記憶體所剩空間總和足夠讓此行程執行，卻因為空間不連續，此時外部斷裂的現象就發生了(External fragmentation)。
2. 一個分割區可以載入一個程式來執行，而 CPU 就在這些程式之間切換執行。由於分割區的大小固定，而程式的大小卻不一定剛好等於所分配到的分割區，導致可能有剩下一部份空間沒有用到，這個剩下的空間稱為內部斷裂(Internal fragmentation)。
3. 分頁法(paging)解決外部斷裂，另外使用聚集(compaction)來收集記憶體內零零散散的可用空間，使其成為一個大區間也是一個解決外部斷裂的方法。動態分割與分段能解決內部斷裂。

6. 假設對於每個記憶體參考，首先查詢 TLB。如果操作失敗，則查詢記憶體中的分頁表。 TLB 和記憶體的存取時間分別為 20ns 和 100ns。如果 TLB 的命中率為 80%，則計算有效存取時間。

Ans.

0.8x(20+100)+0.2x(20+200)=140ns

7. 考慮以下的分頁參考串列：1, 2, 3, 4, 2,1,5,6,2,1,2,3,7,6,3,2,1,2,3　假設使用 4 欄作需求分頁，對於以下的替換算法將產生多少次分頁錯誤?
   FIFO 替換演算法; LRU 替換演算法; 最佳替換演算法

Answer:

(1) FIFO replacement : 14

1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	
1	1	1	1			5	5	5	5		3	3	3		3	1		1	
	2	2	2			2	6	6	6		6	7	7		7	7		3	
		3	3			3	3	2	2		2	2	6		6	6		6	
			4			4	4	4	1		1	1	1		2	2		2	

(2) Optimal replacement: 8

1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	
1	1	1	1			1	1					7				7			
	2	2	2			2	2					2				2			
		3	3			3	3					3				3			
			4			5	6					6				1			

(3) LRU replacement : 10

1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	
1	1	1	1			1	1				1	1	6			6			
	2	2	2			2	2				2	2	2			2			
		3	3			5	5				3	3	3			3			
			4			4	6				6	7	7			1			

8. 假設一個磁碟機有 5000 個磁柱，其編號從 0 到 4999，他現正在服務磁柱 2150 的要求，而且先前要求是在磁柱 1805。如果這個未決的佇列之中的要求以 FIFO 的次序排列如下：2069, 1212, 2296, 2800, 544, 1618, 356, 1523, 4965, 3681
   由目前的讀寫頭位置開始，對於下列的每一種磁碟排班演算法，他的總距離(以磁柱數為單位)是多少？這個總距離是磁碟臂移動去滿足所有未決要求的總移動磁柱數。
   a. FCFS　b. SSTF　c. SCAN　d. LOOK　e. C-SCAN　f. C-LOOK

Answer:

a. FCFS 的排程為 2150, 2069, 1212, 2296, 2800, 544, 1618, 356, 1523, 4965, 3681. 總共移動距離為13,011.

b. SSTF的排程為2150, 2069, 2296, 2800, 3681, 4965, 1618, 1523, 1212, 544, 356. 總共移動距離為7586.

c. SCAN的排程為2150, 2296, 2800, 3681, 4965, 2069, 1618, 1523, 1212, 544, 356. 總共移動距離為7492.

d. LOOK的排程為2150, 2296, 2800, 3681, 4965, 2069, 1618, 1523, 1212, 544, 356 總共移動距離為7424.

e. C-SCAN的排程為2150, 2296, 2800, 3681, 4965, 356, 544, 1212, 1523, 1618, 2069. 總共移動距離為9917.

f. C-LOOK的排程為2150, 2296, 2800, 3681, 4965, 356, 544, 1212, 1523, 1618, 2069. 總共移動距離為

9137

9. 記憶體的區塊區分為 100k，500k，200k，200k 和 600k（按順序排序），計算每個 First-Fit，Best-Fit 和 Worst-Fit 算法如何放置 212K，417k，112k 和 426k？哪種算法能最有效率的利用這些記憶體？

Ans.

	100k	500k	200k,	200k	600k	
First-Fit		(1)配 212k 剩 288k (3)配 112k 剩 176k			(2)配 417k 剩 183k	(4)配 426k 無法配置
Best-Fit		(1)配 212k 剩 288k			(2)配 417k 剩 183k (3)配 112k 剩 71k	(4)配 426k 無法配置
Worst-Fit		(2)配 417k 剩 83k			(1)配 212k 剩 388k (3)配 112k 剩 276k	(4)配 426k 無法配置

皆無法順利配置