

國立臺灣科技大學答案卷

科目 OS 任課教師 古鴻英 系所班組別 ⑪ 資工三
 學號 B9415052 姓名 吳孟達 考試日期 97 年 6 月 19 日


評	分	教師簽名或蓋章
	91	

記分欄 從此處開始寫起。試卷用紙務須節用，非經主試認可不得續用其他紙張作答。

1a. inverted page table 以 physical memory 為對象，記錄該 frame 屬於那個程式的第幾個 page
 4 不單獨存放資訊：process id, page No.

1b. 存 limit & base, base 為該 segment 起點位址, limit 為 segment 的大小
 4

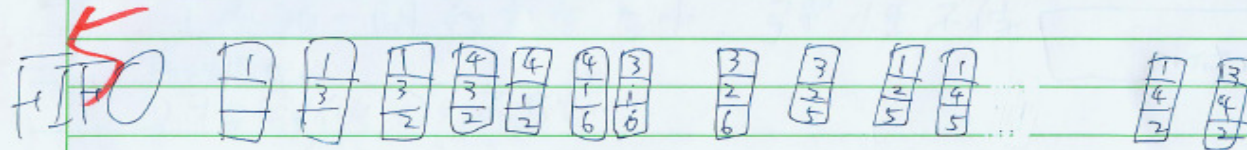
2a. 因為是有需要時才載入，
 4 故可減少一次的傳輸時間。
 並且也可讓程式大小大於記憶體容量。

2b. method 1 :  上限 大於上限，增加頁框數
 6 下限 小於下限，減少頁框數

method 2: 採 working set, working set 是一種藉由「觀察程式」過去使用到的 page 情形, 估計該程式所需頁框數。
 估計 > 已分配給它的 \Rightarrow 增加 frame
 估計 < 已分配給它的 \Rightarrow 減少 frame
 此外, $D = \sum w_{ss_i}$ (所有程式估計需要的) < 電腦裡的頁框
 \Rightarrow 增加 Multiprogramming degree
 若 $D >$ 電腦裡的頁框, \Rightarrow 減少 Multiprogramming degree

3. 指令可能在某一 page, A, B 分別在另外其它 page, 且 A, B 可能會跨 page 儲存
 \Rightarrow worst case: 讀指令 \rightarrow 1 次, 讀 A \rightarrow 2 次, 讀 B \rightarrow 2 次
 Ans: 5 次 page fault.

4(a). 1, 3, 2, 4, 1, 6, 3, 1, 2, 5, 1, 4, 1, 5, 2, 3



Ans: 13 次

4(b), 1, 3, 2, 4, 1, 6, 3, 1, 2, 5, 1, 4, 1, 5, 2, 3

LRU

1

1
3

1
3
2

4
3
2

4
1
2

4
1
6

3
1
2

5
1
2

5
1
4

5
1
2

5
3
2

 Ans: 12次

6(a), 有多個程式同時去存取共同的資料(變數)

race condition 可能會造成共同共享資料產生非預期結果。

thread 1

thread 2

$C = C + 1$

$C = C - 1$

假設 C 初值為 5, 程式執行後, C 可能是: 6, 4, 5.

6(b): progress: 不想進入 critical-section 之程式不該參與, 影響決策過程, 並且決定哪一個程式進入 critical-section 必須在有限時間內完成。

bounded waiting: 若某程式不能進入 critical-section, 必須在有限時間內
之內, 進入到 critical-section.

4 6(a): `bool TestAndSet(bool &lock)`
`{ bool temp = *lock;`
`*lock = true;`
`return temp;`
`}`

6(b) 進入 critical section
`number ++;`
`if(number > 0)`
`{`
 將之前被 block 的程式恢復執行
`}`

離開 critical section

7
wait(mutex);
readcount --;
if (readcount == 0) signal(wrt);
signal(mutex);

(插隊)
b(a). no preemption: 不可搶先的, 若一個資源分配給某一 process.
除非該 process 自願放棄該資源 (ex: 該 process terminate)
要不然其它 process 不可搶走該資源

hold and wait: process 握有一部分資源, 並且正在等待其它所需資源.

依照資源編號
5 給每一資源一個 unique number, process 必須由小至大向 OS 提出申請
這樣就可防止 circular wait

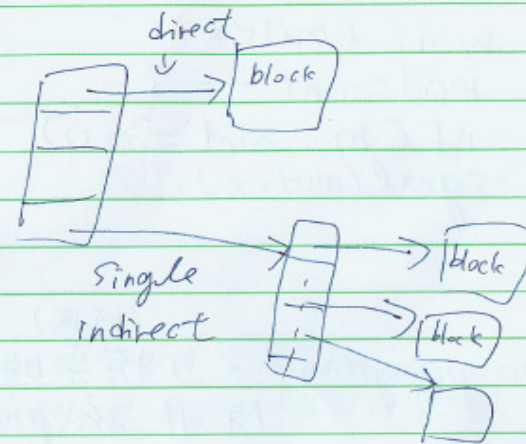
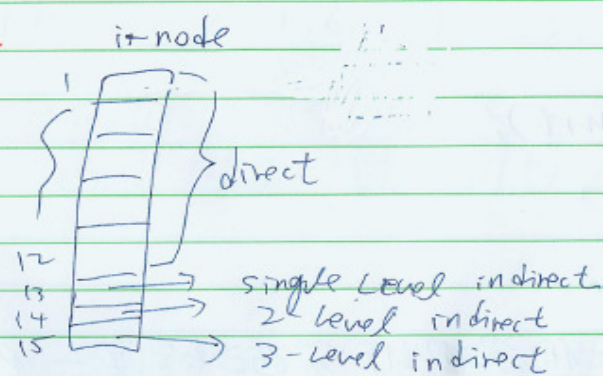
P _i	Allocation			MAX			Need	Finish	Available		
	A	B	C	A	B	C			A	B	C
P ₀	0	3	0	7	5	3	7 2 3	F	2	1	0
P ₁	3	0	2	3	2	2	0 2 0	F	Work		
P ₂	3	0	2	9	0	2	6 0 0	F			
P ₃	2	1	1	2	2	2	0 1 1	F	A	B	C
P ₄	0	0	2	4	3	3	4 3 1	F	2	1	0

Ans: 不安全

10. (a) ① 可靠度不佳
 ② 只能循序存取, 無法 Random Access

- (b) ① 有外部碎片
 ② 必須一開始宣告大小, 彈性不佳
 ③ 檔案修改不易
 (resize)

11 (a) 4



目的, 可以適用於大、小 file

(全 direct, 不適合大 file)
(全 indirect, 不適合小 file)

11 (b). FAT \Rightarrow File Allocation Table

4 原本 linked 的方式, 鏈結是放在 block 裡, 所以只能循序存取
而 FAT 是新增一 table, 將 link 放在該 table 中。

因此，在 table 中先循序找出該 block 的位置，
然後就可去存取該 block。

因此，感覺上有點直接去存取。

只是要先在 File Allocation Table 循序尋找出 block 的位置。