OS hw5 part1. B03304045 李亦晴

1. hard link 和 symbolic link 的差別？分別給出一個優點。

→ hard link:

hard link是資料真實的聯結，它的指令是 ln (原始檔案)(目的檔案)。不能是目錄。 → 限檔案。

所有基於 file system 的 directory 至少要有一個 hard link 給每個檔案一個原始檔名。通常使用 hard link 指令是用於備份。

* 不能跨 Filesystem 使用。

優點：不用擔心原始檔案被刪除。

→ symbolic link.

symbolic link 是建立一個新的 i-node，指向同一個硬碟位置。並且 symbolic link 存的是一個字串構成的檔案，紀錄所指向檔案的相對或絕對路徑，如果所指向的檔案被移動，連結仍會存在，但指向的就不是原向的檔案了。

優點：可以跨越 Filesystem 指到其他位置。

2. disk 有 4000 cylinders each with 8 tracks of 512 blocks. 每個 cylinder 移動需 1 msec (毫秒) → seek

如果沒有把 file block 放近一點，那連續的2塊需要5msec；但若OS將相关的block群聚在一起，平均間距可以減少到2 cylinders，seek time 減少到100微秒(microsec)

求如果讀入一個100塊的file在以上2种情況，且 rotational latency : 10 msec, transfer time : 20 microsec/block

read time = seek + rotate + transfer. (的time)

→ Non-adjacent

$(5+10+20\times10^{-3})\times100 = 1502$ msec (毫秒) *.

→ adjacent

$(100\times10^{-3}\times2+10+20\times10^{-3})\times100 = 1022$ msec (毫秒) *

3 page reference string = 1,2,3,4,2,1,5,6,2,1,2,3,7,6,3,2,1,2,3,6.

How many page fault 會發生在以下的演算法中，並且分別在1~7 frames 情況

* 每個 frame 最開始都是空的，所以 first unique pages 都會 cost one fault.

| (a) LRU replacement | (b) Optimal replacement | frame(s) |
|---|---|---|
| 20 | 20 | 1 |
| 18 | 15 | 2 |
| 15 | 11 | 3 |
| 10 | 8 | 4 |
| 8 | 7 | 5 |
| 7 | 7 | 6 |
| 7 | 7. | 7. |

(4). 假設 File 有 100 个 blocks, 且 file control block (和 index block, 在 indexed allocated 情況) 已存在 memory. 計算有多少 disk I/O 操作, 在 contiguous, linked, indexed (single level) allocation strategies.

contiguous → 假設 no room to grow in begining 但有 in the end.

(a) The block is added at the begining.

→ contiguous = 201 → 100 (read entire file) + 1 (write the new block at 1st block) + 100
→ linked = 1 → Change pointer in file control block to new block (write to 2nd block)
→ indexed = 1 → write new block and update index.

(b). The block is added in middle

→ contiguous = 101 → 50 (write half file) + 1 (new block) + 50 (last 50 block).
→ link = 52 → 50 (read first 50) + 1 (modify point to 50 block) + 1 (new one)
→ indexed = 1 → same as (a).

(c). The block is added in the end

→ contiguous = 1 → write in the end
→ linked = 3 → 1 (read the last one) + 1 (modify) + 1 (new one)
→ indexed = 1 → same as (a).

(d). Removed from the begining

→ contiguous = 198 → 99 (read last 99 blocks) + 99 (write back at beginning)
→ linked = 1 → 1 (read the first block and copy it)
→ indexed = 0 → just update indexed in memory

(e). Removed from middle

→ contiguous = 98 → 49 (read last 49 blocks) + 49 (write back stat from 51th block)
→ linked = 52 → 51 (read the first 51 block) + 1 (copy 51th → 50th).
→ indexed = 0. → same as (d).

(5) 某台 PC 有 virtual memory space $2^{32}$ bytes, 218 bytes physical memory. virtual memory 實做 by paging, 且 page size 是 4096 bytes. virtual address 是 11123456.

解釋 system 是如何建立相对的 physical location, 区分 software & hardware operations.

$(11123456)_{10} = (\underbrace{0001\ 0001\ 0001\ 0010\ 0011}_{\text{page table offset}}\ \underbrace{0100\ 0101\ 0110}_{\text{page offset}})_2$

page size = 4096 bytes = $2^{12}$ bytes.

page table size = $2^{32-12} = 2^{20}$ bytes.

因此, 後 12 bits (0100 0101 0110)$_2$ 用作於 page offset

而剩下的 20 bits (0001 0001 0001 0010 0011)$_2$ 作為 page table offset.