Operating-system homework#3

Written exercises

Chap. 10

10.21: Assume that we have a demand-paged memory.
- The page table is held in registers.
- It takes 8 milliseconds to service a page fault if an empty frame is available or if the replaced page is not modified and 20 milliseconds if the replaced page is modified.
- Memory-access time is 100 nanoseconds.

– Assume that the page to be replaced is modified 70 percent of the time.
– What is the maximum acceptable page-fault rate for an effective access time of no more than 200 nanoseconds?

memory-access: 100n

page fault
- empty frame: 8 ms
- modified: 20 ms

total time <= 200n

x = page fault rate

modified time : $((x * 70\%) * 20m)$

empty frame time : $(x - (x * 70\%) * 8m)$

no page fault time : $(1 - x) * 100n$

total time = modified time + empty frame time + no page fault time $\leq 200n$

$x \leq 6.10 * 10^{-6}$

10.24: Apply the (1) FIFO (2) LRU (3) Optimal (OPT) replacement algorithms for the following page reference string:
3, 1, 4, 2, 5, 4, 1, 3, 5, 2, 0, 1, 1, 0, 2, 3, 4, 5, 0, 1.
Indicate the number of page faults for each algorithm assuming demand paging with three frames.
- FIFO : 15
- LRU : 16
- OPT : 11

10.37: What is the cause of thrashing? How does the system detect thrashing? Once it detects thrashing, what can the system do to eliminate this problem?
thrashing: 需要的 page 數量超過了可用的 page 數量，導致大量的 page fault，使得 CPU 利用率下降。
detect: CPU 利用率下降，page fault rate 上升
eliminate: 增加可用的 page 數量，減少 CPU 利用率，或是增加 CPU 的速度，或實現 Working Set Model。

Chap. 11

11.13: Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4,999.
– The drive is currently serving a request at cylinder 2,150, and the previous request was at cylinder 1,805.
– The queue of pending requests, in FIFO order, is 2,069; 1,212; 2,296; 2,800; 544; 1,618; 356; 1,523; 4,965; 3,681

Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests, for each of the following disk-scheduling algorithms?
- (a) FCFS

2150-> 2,069-> 1,212-> 2,296-> 2,800-> 544-> 1,618-> 356-> 1,523-> 4,965-> 3,681
81 + 857 + 1084 + 504 + 2256 + 1074 + 1262 + 1167 + 3442 + 1284 = 13011

- (b) SCAN

2150 -> 2069 -> 1618 -> 1532 -> 1212 -> 544 -> 356 -> 0 -> 2296 -> 2800 -> 3681 -> 4965
=7492

- (c) C-SCAN

2150 -> 2296 -> 2800 -> 3681 -> 4965 -> 4999 -> 0 -> 356 -> 544 -> 1212 -> 1532 -> 1618 -> 2069
=4918

11.20: Consider a RAID level 5 organization comprising five disks, with the parity for sets of four blocks on four disks stored on the fifth disk.– How many blocks are accessed in order to perform the following?
- (a) A write of one block of data.

1 + 1 = 2
- (b) A write of seven contiguous blocks of data

7 + 2 = 9

11.21: Compare the throughput achieved by a RAID level 5 organization with that achieved by a RAID level 1 organization.
- (a) Read operations on single blocks.

RAID1：因儲存格式，可支援分散讀取
RAID5：相較於 RAID1，RAID5 的讀取速度較慢
- (b) Read operations on multiple contiguous blocks.

RAID1：如果 n 大於 disk 數量，RAID1 的讀取速度較慢
RAID5：連續讀取速度較快

Chap. 14

14.14: Consider a file system on a disk that has both logical and physical block sizes of 512 bytes.
– Assume that the information about each file is already in memory
– For each of the three allocation strategies (contiguous, linked, and indexed), answer these questions:
- (a) How is the logical-to-physical address mapping accomplished in this system? (For indexed allocation, assume that a file is always less than 512 blocks long)

contiguous : 存 start block 和 block 數量

linked : 存 start 接著在每個 block 存下一個 block 的 index

index : 使用 index block，將 physical block 的 index 存入 index block。

- (b) If we are currently at logical block 10 (the last block accessed was block 10) and want to access logical block 4, how many physical blocks must be read from the disk?

contiguous : 1

linked : 5

index : 2

14.15: Consider a file system that uses inodes to represent files

– Disk blocks are 8KB in size, and a pointer to a disk block requires 4 bytes

– This file system has 12 direct disk blocks, as well as single, double, and triple indirect disk blocks

– What is the maximum size of a file that can be stored in this file system?

$12 * 8K = 96K$

$\frac{8K}{4} \cdot 8K = 2K \cdot 8K = 16M$

$\frac{16M}{4} \cdot 8K = 4M \cdot 8K = 32G$

$\frac{32G}{4} \cdot 8K = 8G \cdot 8K = 64T$

$0.96M + 16M + 32000M + 64000000M = 64032016.96M$