Homework Assignment #4

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?

$$ \begin{aligned} EAT &= p * (8,000,000 * 30\% + 20,000,000 * 70\%) + (1 - p) * 100 \\ &=16400000p + 100 - 100p \\ &= 16399900p + 100 \\ EAT &\leq 200 \\ p &\leq 6 \times 10^{-6} \end{aligned} $$

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.

(1) 15

```
3 3
1 3,1
4 3,1,4
2 1,4,2
5 4,2,5
4
1 2,5,1
3 5,1,3
5
2 1,3,2
0 3,2,0
1 2,0,1
1
0
2
3 0,1,3
4 1,3,4
5 3,4,5
0 4,5,0
1 5,0,1
```

(2) 16

```
3 3
1 3,1
4 3,1,4
```

```
2 2,1,4
5 2,5,4
4
1 1,5,4
3 1,3,4
5 1,3,5
2 2,3,5
0 2,0,5
1 2,0,1
1
0
2
3 2,0,3
4 2,4,3
5 5,4,3
0 5,4,0
1 1,4,0
```

(3) 10

```
3 3
1 3,1
4 3,1,4
2 2,1,4
5 5,1,4
4
1
3 5,1,3
5
2 2,1,3
0 2,1,0
1
1
0
2
3 3,1,0
4 4,1,0
5 5,1,0
0
1
```

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?

cause:

- When the page is not enough, it will cause numerous page fault, thus decrease the CPU utilization.
- OS will increase the degree of multi-programming since low CPU utilization.

- if cpu spent most of the time dealing with page fault, it will cause thrashing

detect:

- When page fault rate is too high
- When working set size (WSS) > avalible frames

eliminate:

- Increase frames that each process can use to control page fault rate
- Suspent the process

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: 2069; 1212; 2296; 2800; 544; 1618; 356; 1523; 4965; 3681

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 (b) SCAN (c) C-SCAN

(a) FCFS: 13011

```
2150; 2069; 1212; 2296; 2800; 544; 1618; 356; 1523; 4965; 3681
```

(b) SCAN: 7492

```
2150; 2296; 2800; 3681; 4965; 4999; 2069; 1618; 1523; 1212; 544; 356
```

(c) C-SCAN: 4918

```
2150; 2296; 2800; 3681; 4965; 4999; 0; 356; 544; 1212; 1523; 1618; 2069
```

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.

2 blocks. 1 target block + pariry block

(b) A write of seven contiguous blocks of data.

9 blocks. 4 target block + 1 parity block 3 target block + 1 parity block

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.

RAID 1 和 RAID 5 在讀取 single block 上效能與單顆磁碟一致，但 RAID 1 可以較為彈性的選擇要讀取哪個 disk

(b) Read operations on multiple contiguous blocks.

若 RAID 1 包含兩顆 disk，則可用兩倍時間平行讀取連續區塊 RAID 5 可以同時在不同 disk 上讀取不同的 block，隨著 disk 增多可以獲得更高的讀取效能

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)

Countiguous: logical address / 512 = Q ... R Block to be accessed = Q + starting address Displacement into block = R

Linked: logical address / 511 = Q ... R Block to be accessed is the Qth block in the linked chain Displacement into block = R + 1

Index: logical address / 512 = Q ... R First get the index block into memory. The physical block address is contained in the index block at location Q. R is the displacement into the desired physical 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?

Countiguous: 1 block

Linked: 4 blocks

Indexed: 2 blocks (index table + target)

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?

64 TB $$ \begin{aligned} &8kb \div 4bytes = 2^{11} \ & 8kb * 12 + 8kb * 2^{11} + 8kb * 2^{22} + 8kb * 2^{33}\ =& 68753047648 kb \ =& 64 TB \end{aligned} $$