109370210 工管四乙 黃鈺凱

OS #4

- **10.21**

Time taken to service page Fault for empty page or unmodified page = 8ms.

Time taken to service page Fault for modified page = 20ms

Memory access time = 100ns

Effective Access time = 200ns

EAT = (1-p)*(100) + (p)*(100 + (1-.7)*(8msec) + (.7)*(20msec))

= 100 - 100p + 100p + (2.4e6)*p + (14e6)*p

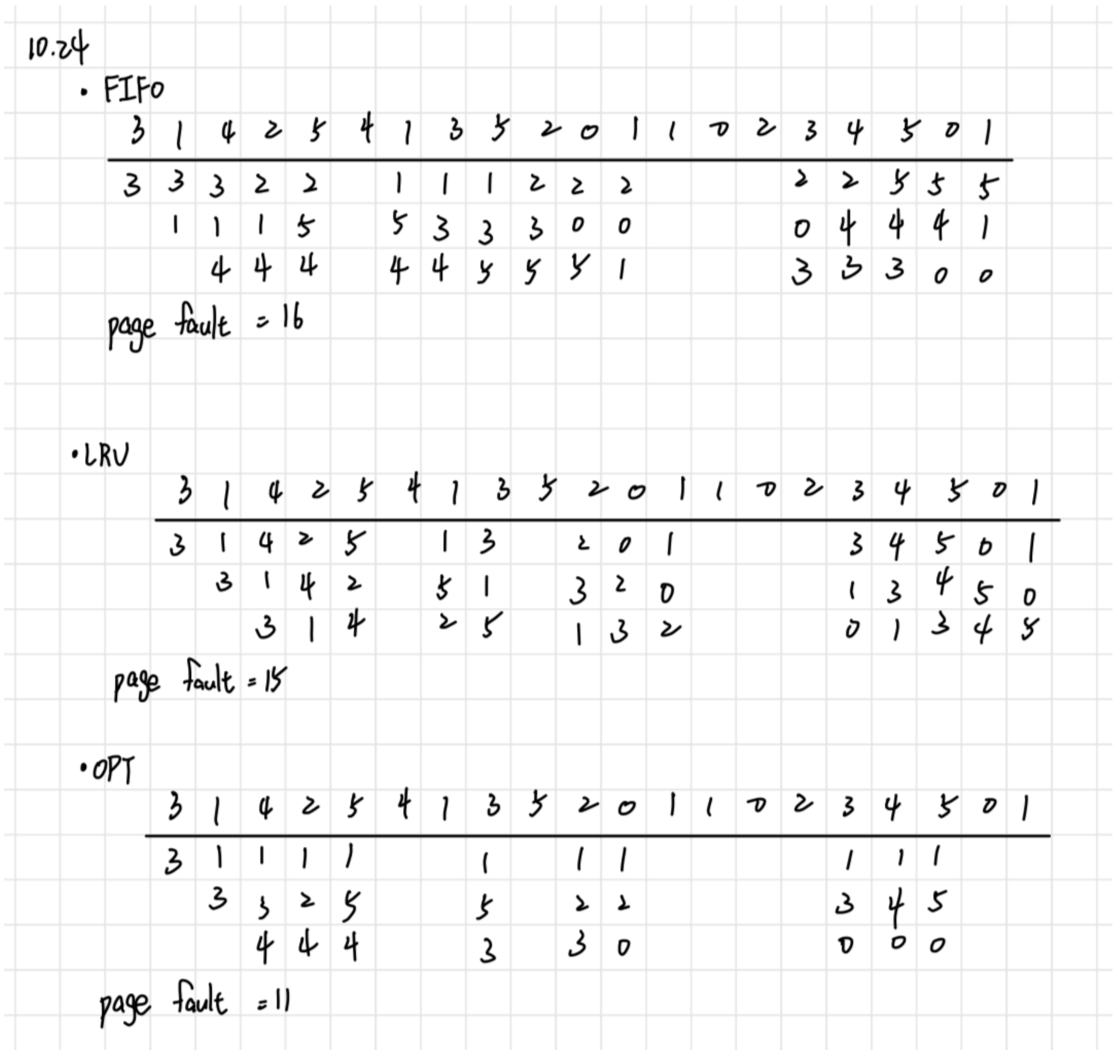
= 100 + (16.4e6)*p

200 = 100 + (16.4e6)*p

p = 100/16.4e6 = 6.097560975609756097560975609756097561e-6 ~ 6.01e-6

p-->page Fault Rate

- **10.24**



10.24

• FIFO

3	1	4	2	5	4	1	3	5	2	0	1	1	0	2	3	4	5	0	1
3	3	3	2	2		1	1	1	2	2	2				2	2	5	5	5
	1	1	1	5		5	3	3	3	0	0				0	4	4	4	1
		4	4	4		4	4	5	5	5	1				3	3	3	0	0

page fault = 16

• LRU

3	1	4	2	5	4	1	3	5	2	0	1	1	0	2	3	4	5	0	1
3	1	4	2	5		1	3		2	0	1				3	4	5	0	1
	3	1	4	2		5	1		3	2	0				1	3	4	5	0
		3	1	4		2	5		1	3	2				0	1	3	4	5

page fault = 15

• OPT

3	1	4	2	5	4	1	3	5	2	0	1	1	0	2	3	4	5	0	1
3	1	1	1	1			1		1	1					1	1	1		
	3	3	2	5			5		2	2					3	4	5		
		4	4	4			3		3	0					0	0	0		

page fault = 11

- **10.37**

1. Causes of Thrashing

Thrashing is typically caused by:

- **Insufficient physical memory**: When the sum of the working sets of all processes exceeds the physical memory available.
- **High degree of multiprogramming**: Running too many processes concurrently.
- **Poor locality of reference**: When processes do not exhibit good locality of reference, leading to frequent page faults.

2. Detecting Thrashing

A system can detect thrashing by:

- **Monitoring the page fault rate**: If the page fault rate is very high and the system throughput is low, thrashing may be occurring.
- **Using the Working Set Model**: By monitoring the working set of each process, the system can determine if the total demand for frames is greater than the number of available frames.
- **CPU Utilization**: Thrashing often leads to a significant drop in CPU utilization since more time is spent swapping pages than executing instructions.

3. Eliminating Thrashing

Once thrashing is detected, the system can take the following actions to mitigate it:

- **Reducing the level of multiprogramming**: Temporarily suspend or swap out some processes to decrease the load on memory.
- **Increasing physical memory**: Adding more RAM can help accommodate more pages and reduce the need for frequent paging.
- **Using a better page replacement algorithm**: Employing algorithms like Least Recently Used (LRU) or variants that can better predict which pages to replace.
- **Adjusting the working set**: Dynamically adjusting the working set size to reflect the actual needs of processes, thereby optimizing memory usage.

- **11.13**

  (a) FCFS : 2150 -> 2069 -> 1212 -> 2296 -> 2800 -> 544 -> 1618 -> 356 -> 1523 -> 4965 -> 3681

  Total: 13011

  (b) SCAN: 2150 -> 2296 -> 2800 -> 3681 -> 4965 -> 4999 -> 2069 -> 1618 -> 1523 -> 1212 -> 544 -> 356

  Total: 7492

  (c) C-SCAN: 2150 -> 2296 -> 2800 -> 3681 -> 4965 -> 4999 -> 0 -> 356 -> 544 ->1212 -> 1523 -> 1618 -> 2069

  Total: 9917

- **11.20**

  (a) A write of one block of data requires the following: read of the parity block, read of the old data stored in the target block, computation of the new parity based on the differences between the new and old contents of the target block, and write of the parity block and the target block.

  (b) Assume that the seven contiguous blocks begin at a four-block boundary. A write of seven contiguous blocks of data could be performed by writing the seven contiguous blocks, writing the parity block of the first four blocks, reading the eight block, computing the parity for the next set of four blocks and writing the corresponding parity block onto disk.

- **11.21**

  (a) The amount of throughput depends on the number of disks in the RAID system. A RAID Level 5 comprising of a parity block for every set of four blocks spread over five disks can support four to five operations simultaneously. A RAID Level 1 comprising of two disks can support two simultaneous operations. Of course, there is greater flexibility in RAID Level 1 as to which copy of a block could be accessed and that could provide performance benefits by taking into account position of disk head.

  (b) RAID Level 5 organization achieves greater bandwidth for accesses to multiple contiguous blocks since the adjacent blocks could be simultaneously accessed. Such bandwidth improvements are not possible in RAID Level 1.

- **14.14**

  Let Z be the starting file address (block number).

  Contiguous: (a) Divide the logical address by 512 with X and Y the resulting quotient and remainder respectively. Add X to Z to obtain the physical block number. Then Y is the displacement into that block. (b) 1.

  Linked: (a) Divide the logical physical address by 511 with X and Y the resulting quotient and remainder respectively. Chase down the linked list (getting X+1 blocks). Y +1 is the displacement into the last physical block. (b) 4.

  Indexed: (a) Divide the logical address by 512 with X and Y the resulting quotient and remainder respectively. First get the index block into memory. The physical block address is contained in the index block at location X. Y is the displacement into the desired physical block. (b) 2.

- **14.15**

  (12 * 8 KB) + (2048 * 8 KB) + (2048 * 2048 * 8 KB) + (2048 * 2048 * 2048 * 8 KB) = 64 terabytes