Eddie Gurnee
COSC 592
Operating Systems

Chapter 9 & 10 Reading Questions:

1. Consider the following page reference string:

   1, 2, 3, 1, 4, 2, 1, 5, 6, 2, 2, 3, 5, 6, 3, 2, 1, 2, 3, 6

How many page faults would occur for the following replacement algorithms, assuming one, three, five, and six frames? Remember that all frames are initially empty, so your first unique pages will cost one fault each. Provide a diagram/table similar to Figure 9.12 (or to the programming assignment) for each of the replacement algorithms assuming three frames.

LRU replacement

FIFO replacement

Optimal replacement

Response:

LRU/FIFO/Optimal																			
1	2	3	1	4	2	1	5	6	2	2	3	5	6	3	2	1	2	3	6
1	2	3	1	4	2	1	5	6	2	2	3	5	6	3	2	1	2	3	6

Page Faults:

LRU	19
FIFO	19
Optimal	19

LRU																			
1	2	3	1	4	2	1	5	6	2	2	3	5	6	3	2	1	2	3	6
1	1	1		1	1		1	1	2		2	2	6		6	1			6
0	2	2		4	4		5	5	5		3	3	3		3	3			3
0	0	3		3	2		2	6	6		6	5	5		2	2			2

FIFO																			
1	2	3	1	4	2	1	5	6	2	2	3	5	6	3	2	1	2	3	6
1	1	1		4		4	4	6	6		6	5	5		5	1		1	1
0	2	2		2		1	1	1	2		2	2	6		6	6		3	3
0	0	3		3		3	5	5	5		3	3	3		2	2		2	6

Optimal																			
1	2	3	1	4	2	1	5	6	2	2	3	5	6	3	2	1	2	3	6
1	1	1		1			1	6			6				6	1			6
0	2	2		2			2	2			3				3	3			3
0	0	3		4			5	5			5				2	2			2

Page Faults:

LRU	14
FIFO	15
Optimal	10

LRU																			
1	2	3	1	4	2	1	5	6	2	2	3	5	6	3	2	1	2	3	6
1	1	1		1			1	1			1								
0	2	2		2			2	2			2								
0	0	3		3			3	6			6								
0	0	0		4			4	4			3								
0	0	0		0			5	5			5								

FIFO																		
1	2	3	1	4	2	1	5	6	2	2	3	5	6	3	2	1	2	3	6
1	1	1		1			1	6								6	6	6	
0	2	2		2			2	2								1	1	1	
0	0	3		3			3	3								3	2	2	
0	0	0		4			4	4								4	4	3	
0	0	0		0			5	5								5	5	5	

Optimal																		
1	2	3	1	4	2	1	5	6	2	2	3	5	6	3	2	1	2	3	6
1	1	1		1			1	1										
0	2	2		2			2	2										
0	0	3		3			3	3										
0	0	0		4			5	5										
0	0	0		0			0	6										

Page Faults:

LRU	7
FIFO	9
Optimal	6

LRU/FIFO																		
1	2	3	1	4	2	1	5	6	2	2	3	5	6	3	2	1	2	3	6
1	1	1		1			1	1										
0	2	2		2			2	2										
0	0	3		3			3	3										
0	0	0		4			4	4										
0	0	0		0			5	5										
0	0	0		0			0	6										

Optimal																		
1	2	3	1	4	2	1	5	6	2	2	3	5	6	3	2	1	2	3	6
1	1	1		1			1	1										
0	2	2		2			2	2										
0	0	3		3			3	3										
0	0	0		4			5	5										
0	0	0		0			0	6										
0	0	0		0			0	0										

Page Faults:

LRU	6
FIFO	6
Optimal	6

2. How does manipulating the degree of multiprogramming affect the likelihood of thrashing?

Response: As the degree of multiprogramming increases, CPU utilization will reach a plateau of effectiveness, at which further increases to the multiprogramming can result in thrashing.

3. Assume a system is using demand paging with a disk that has an average access and transfer time of 30 milliseconds. Addresses are translated through a page table in main memory, with an access time of 2 microseconds per memory access. Thus, each memory reference through the page table takes two accesses (one to access the page table plus one to access the final address). To improve this time, we add some associative memory that can complete a memory reference with only a single access if the page-table entry is in the associative memory (because we don't need to access the page table itself). Assume that 90 percent of the accesses are in the associative memory and that, of those remaining, 10 percent (or 1 percent of the total) cause page faults. What is the effective memory access time?

Response:

Associative memory	+ other memory	+ page faults
= (2µs * 0.9)	+ (4µs * 0.09)	+ (30_004µs * 0.01)
= 1.8µs	+ 0.36µs	+ 300.04µs
= 302.2µs		

4. 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:

2296, 2069, 1212, 3681, 2800, 544, 1618, 356, 1523, 4965

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?

FCFS
SSTF
SCAN
LOOK
C-SCAN
C-LOOK

Response:

a) FCFS: [type=FCFS, order=[2296, 2069, 1212, 3681, 2800, 544, 1618, 356, 1523, 4965], totalDistance=13781]
b) SSTF: [type=SSTF, order=[2069, 2296, 2800, 3681, 4965, 1618, 1523, 1212, 544, 356], totalDistance=7586]
c) SCAN: [type=SCAN, order=[2296, 2800, 3681, 4965, 4999, 2069, 1618, 1523, 1212, 544, 356], totalDistance=7492]
d) LOOK: [type=LOOK, order=[2296, 2800, 3681, 4965, 2069, 1618, 1523, 1212, 544, 356], totalDistance=7424]
e) C-SCAN: [type=C_SCAN, order=[2296, 2800, 3681, 4965, 4999, 0, 356, 544, 1212, 1523, 1618, 2069], totalDistance=9917]
f) C-LOOK: [type=C_LOOK, order=[2296, 2800, 3681, 4965, 356, 544, 1212, 1523, 1618, 2069], totalDistance=9137]

ASIDE: Yes I coded up all of these algorithms in Java, obviously the best use of my time. I definitely wasn't just angling for extra credit.

5. When might we prefer sector sparing over sector slipping? Why?

Response: Using sector slipping requires the movement of the data on all of the blocks between a bad block and an open block; in situations where the resources required to move all that data is greater than the occasional remapped access that sector sparing would provide, it would be logical to choose sector sparing (captain).

6. Consider a RAID level 5 organization comprising four disks, with the parity for sets of three blocks on three disks stored on the fourth disk. How many blocks are accessed in order to perform the following?

a. A write of one block of data

b. A write of nine continuous blocks of data

Response:

a) write of one block of data:
    i. read parity bit
    ii. old data stored in target location is read
    iii. compute the new parity block from diff between old data and incoming data
    iv. write to parity and target blocks

    => 4 block accesses total

b) nine continuous blocks of data
    i. write the nine blocks
    ii. compute/write parity of the first 3
    iii. compute/write parity of the second 3
    iv. compute/write parity of the last 3

    => 12 block accesses total