Kendrick Kwok - CSC 415 - 12/1/16 - 912351666

HOMEWORK #2 :

1.(10 Points) Assume thread 1 calls x.signal() and thread 2 later calls x.wait() (and assume no other operations are executed on x). Will the behavior of the threads differ depending on whether x is a semaphore (initialized to 0) or a monitor condition variable? If so, how?

A: Semaphores and condition variables are built on top of mutual exclusion and can be used for similar purposes. The difference between them is that a condition variable is generally used to avoid busy waiting (looping repeadly to check a condition) until something happens for it to become available.A conditional variable is a waiting-queue, that goes to acquire mutex, check the condition, and then release the mutex. However, with a condition variable, it wastes processing time to have the thread of the condition variable repeatedly checked. A semaphore is used similarly, but better used when you have a shared resource. A semaphore is better in this case because it can signal to the other thread when it is finished, and can be used without external dependencies. Furthermore, another difference is that a semaphore is used to control the number of threads executing. You can treat semaphore to be a more organized structure than conditional variable. You cannot implement synchronization with mutexes, that's why there is condition variables.

2. (10 Points) Consider the following segment table: What are the physical addresses for the following logical addresses? a. 0,430 b. 1,10 c. 2,500 d. 3,400 e. 4,112

If the address is outside of the range determined by the two registers, it results in trap to OS

a) (0,430) = 430 + 219 = 649
b) (1, 10) = 2300 + 10 = 2310
c) (2,500) = Illegal reference: trapping to operating system
d) (3,400) =1327+400 = 1727
e) (4,112) = Illegal reference: trapping to operating system

3. What is internal Fragmentation and when does it occur?

A: Internal fragmentation occurs when the allocator for memory leaves extra space empty inside a block of memory that has been allocated for a client. This occurs when blocks need to be cut into blocks of certain sizes. For example, what happens if the memory block needs to be divided into four equal parts? If you have a 59 bytes and it is divided into four parts, there is still a chunk of memory left unused. All of the unused memory can be built up and create large quantities of memory that cannot be used by an allocator. Because of useless bytes inside large memory blocks, that is the reason why this fragmentation is called internal.

4. Translate the following logical addresses into physical addresses using the provided page table:

Process :
2 MB = 2^21
24 bit logical and physical address
24 - 21 = 3 in pages
|p|p|p|o|o|o|o|o|o|o|o|o|o|o|o|o|o|o|o|o|o|o|o||
3 bit pages, 21 bit offset

a.0x234800 = **001**000110100100000000000
Page #1 = Frame #4
Physical Address = (Interchange the page with frame and keep the offset)
1000 0011 0100 1000 0000 0000
Physical Address = 0x834800
‘
b. 0xBB4400 = **101**1 1011 0100 0100 0000 0000
Page # 5 = Frame #5
0xBB4400 = (Interchange the page with frame and keep the offset)
1011 1011 0100 0100 0000 0000
Physical Address = 0xBB4400

5. Consider the following page reference string: 1, 3, 2, 4, 3, 5, 6, 7, 2, 3, 2, 1, 7, 6, 5, 4, 7, 2, 5, 6
How many page faults would occur for the page replacement algorithms FIFO, Optimal, and LRU, assuming one, four, and seven free frames at the beginning?

REFERENCE STRING : 1, 3, 2, 4, 3, 5, 6, 7, 2, 3, 2, 1, 7, 6, 5, 4, 7, 2, 5
Bolded numbers = Page Faults

<u>FIFO</u>
1 Free Frame
1 | 3 | 2 | 4 | 3 | 5 | 6 | 7 | 2 | 3 | 2 | 1 | 7 | 6 | 5 | 4 | 7 | 2 | 5 | 6
1 3 2 4 3 5 6 7 2 3 2 1 7 6 5 4 7 2 5 6
Page Fault: 20

4 Free frame
1 | 1 3 | 1 3 2 | 1 3 2 4 | 1 3 2 4 | 5 3 2 4 | 5 6 2 4 | 5 6 7 4 | 5 6 7 2 | 3 6 7 2 | 3 6 7 2 |
1 3 2 4 3 **5 6 7** **2 3** 2
| 3 1 7 2 | 3 1 7 2 | 3 1 6 2 | 3 1 6 5 | 4 1 6 5 | 4 7 6 5 | 4 7 2 5 | 4 7 2 5 | 4 7 2 6
1 7 **6 5 4 7 2** 5 **6**
Page Fault: 16

7 Free Frame

1|1 3 |1 3 2 | 1 3 2 4 | 1 3 2 4 | 1 3 2 4 5 | 1 3 2 4 5 6 | 1 3 2 4 5 6 7 | 1 3 2 4 5 6 7 | 1 3 2 4 5 6 7 |

1 3 2 4 3 **5 6 7** 2 3

1 3 2 4 5 6 7 | 1 3 2 4 5 6 7 | 1 3 2 4 5 6 7 |1 3 2 4 5 6 7 |

2 1 7 6

1 3 2 4 5 6 7 | 1 3 2 4 5 6 7 | 1 3 2 4 5 6 7 | 1 3 2 4 5 6 7 |1 3 2 4 5 6 7 | 1 3 2 4 5 6 7 |

5 4 7 2 5 6

Page Fault: 7

<u>Optimal</u>

1 Free Frame

1 | 3 | 2 | 4 | 3 | 5 | 6 | 7 | 2 | 3 | 2 | 1 | 7 | 6 | 5 | 4 | 7 | 2 | 5 | 6

1 3 2 4 3 5 6 7 2 3 2 1 7 6 5 4 7 2 5 6

Page Fault: 20

4 Free frame

1 | 1 3 | 1 3 2 | 1 3 2 4 | 1 3 2 4 | 1 3 2 5 | 1 3 2 6 | 1 3 2 7 | 1 3 2 7 | 1 3 2 7 | 1 3 2 7 |

1 3 2 4 3 **5 6 7** 2 3 2

| 1 3 2 7 | 1 3 2 7 | 6 3 2 7 | 5 3 2 7 | 5 4 2 7 | 5 4 2 7 | 5 4 2 7 | 6 4 2 7

1 7 **6 5 4** 7 2 5 **6**

Page Fault: 11

7 Free Frame

1|1 3 |1 3 2 | 1 3 2 4 | 1 3 2 4 | 1 3 2 4 5 | 1 3 2 4 5 6 | 1 3 2 4 5 6 7 | 1 3 2 4 5 6 7 | 1 3 2 4 5 6 7 |

1 3 2 4 3 **5 6 7** 2 3

1 3 2 4 5 6 7 | 1 3 2 4 5 6 7 | 1 3 2 4 5 6 7 |1 3 2 4 5 6 7 |

2 1 7 6

1 3 2 4 5 6 7 | 1 3 2 4 5 6 7 | 1 3 2 4 5 6 7 | 1 3 2 4 5 6 7 |1 3 2 4 5 6 7 | 1 3 2 4 5 6 7|

5 4 7 2 5 6

Page Fault: 7

<u>LRU</u>

1 Free Frame

1 | 3 | 2 | 4 | 3 | 5 | 6 | 7 | 2 | 3 | 2 | 1 | 7 | 6 | 5 | 4 | 7 | 2 | 5 | 6

1 3 2 4 3 5 6 7 2 3 2 1 7 6 5 4 7 2 5 6

Page Fault: 20

4 Free frame

1 | 1 3 | 1 3 2 | 1 3 2 4 | 1 3 2 4 | 5 3 2 4 | 5 3 6 4 | 5 3 6 7 | 5 2 6 7 | 3 2 6 7| 3 2 6 7 |

1 3 2 4 3 **5 6 7 2 3** 2

| 3 2 1 7 | 3 2 1 7 | 6 2 1 7 | 6 5 1 7 | 6 5 4 7 | 6 5 4 7 | 2 5 4 7 | 2 5 4 7 | 2 5 6 7

1 7 **6** **5** **4** 7 **2** 5 **6**
Page Fault: 15

7 Free Frame
1|1 3 |1 3 2 | 1 3 2 4 | 1 3 2 4 | 1 3 2 4 5 | 1 3 2 4 5 6 | 1 3 2 4 5 6 7 | 1 3 2 4 5 6 7 | 1 3 2 4 5 6 7 |
1 **3** **2** **4** 3 **5** **6** **7** 2 3
1 3 2 4 5 6 7 | 1 3 2 4 5 6 7 | 1 3 2 4 5 6 7 |1 3 2 4 5 6 7 |
2 1 7 6
1 3 2 4 5 6 7 | 1 3 2 4 5 6 7 | 1 3 2 4 5 6 7 | 1 3 2 4 5 6 7 |1 3 2 4 5 6 7 | 1 3 2 4 5 6 7|
5 4 7 2 5 6
Page Fault: 7

6. 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?

A: What would occur if a process does not have enough frames? Thrashing is a process that spends more time paging then executing, causing page faults. The system can detect thrashing by evaluating the level of multiprogramming to the level of CPU utilization. To prevent thrashing, the system can reduce the level of multiprogramming. Measure and control the page-fault rate to prevent thrashing.

7. Starting from the current head position, what is the order in which the requests will be serviced for each of the following disk-scheduling algorithms (assuming no other requests arrive and the head's initial movement is toward higher numbered cylinders for c-f)?
333, 1200, 922, 4545, 3786, 3605

Intial start: 1056

A. FCFS - 1056, 333, 1200, 922, 4545, 3786, 3605
B. SSTF - 1056, 922, 1200, 333, 3605, 3786, 4545
C. SCAN - 1056, 333, (HITS 0), 922, 1200, 3605, 3786, 4545
D. LOOK- 1056, 1200, 3605, 3786, 4545, 922, 333
E. C-SCAN - 1056, 1200, 3605, 3786, 4545, (HITS 4999), (HITS 0), 333, 922
F. C-LOOK - 1056, 333,922, 1200, 3605, 3786, 4545

8. Describe the procedure for handling the page fault when there is always a free frame in demand paging. (9 Points)

A: Page fault occurs if an invalid page is accessed during MMU address translations. First, the operating system determines whether the reference is a valid or invalid memory access. If invalid, terminate the process. If valid, but the page is not in memory, it is then paged in. We then read the page into the free frame, and update the page table to indicate that the page is in memory. Restart the instruction that has caused page faults.