CS4328: Homework #4

Due on April 30, 2019 at 11:59PM

Jamal Rasool

In Collaboration with:

Kristof York, John Daloni, Fernando Valdez

Jamal Rasool
Thursday, April 18, 2019

Problem 1

Consider a simple paging system with the following parameters: 2^{32} bytes of physical memory; page size of 2^{10} bytes; 2^{16} pages of logical address space. Answer the following questions: [10 pts]

(a) How many bits are in a logical address?
$2^{16} + 2^{10} = 2^{26}$
As a result of this the number of bits in logical address is 26

(b) How many bytes are in frame?

The size of the frame is the same size as the page hence the value for how many bytes would be 2^{10}

(c) How many bits in the physical address specify a frame?

The number of bits required to specify a frame is 22 within the physical address. This is done by the following equation:

$$\frac{2^{32}}{2^{10}} = 2^{22}$$

(d) How many entries in the page table

Due to the number of entries in the page table being the number of pages in the virtual memory, the number of entries within the page table is 2^{16}

(e) How many bits in each page table entry? Assume each page table entry includes a valid/invalid bit.

23 bits are in each page table due to the number of bits required to specify the frame location in physical address is 22, the number of bits in each page table is one on top of that value hence 23.

Problem 2

Consider a paging system with page table stored in memory, answer the following [3 pts each]:

(a) If a memory reference takes 1.2 microseconds, how long does a paged memory reference take?

Time = Checking page lookup table + Accessing Instruction from memory

Therefore, with the memory reference taking 1.2 microseconds, we find that to access the instruction from the page table it's the same amount of time in which is 1.2 microseconds also. Hence the total time will be 1.2 + 1.2 = 2.4 microseconds.

Jamal Rasool

(b) If we add 8 associative registers and 75% of all page table references are found in the associative registers, what is the effective memory reference time? Assume that checking a page table entry in the associative registers takes 0.1 microsecond.

With 1.2 microseconds to access memory and 0.1 microseconds to search the page table we get the following equation

(75%) -> Beginning Portion

$\mathbf{0.75 * (1.2 + 0.1) = 9.75}$ ***microseconds***

then

(25%) -> Remaining portion

$\mathbf{0.25 * (2.4) = 0.6}$ ***microseconds***

Therefore, to total time is going to be the sum of the two values to get the effective memory reference time of 10.35 microseconds

Problem 3

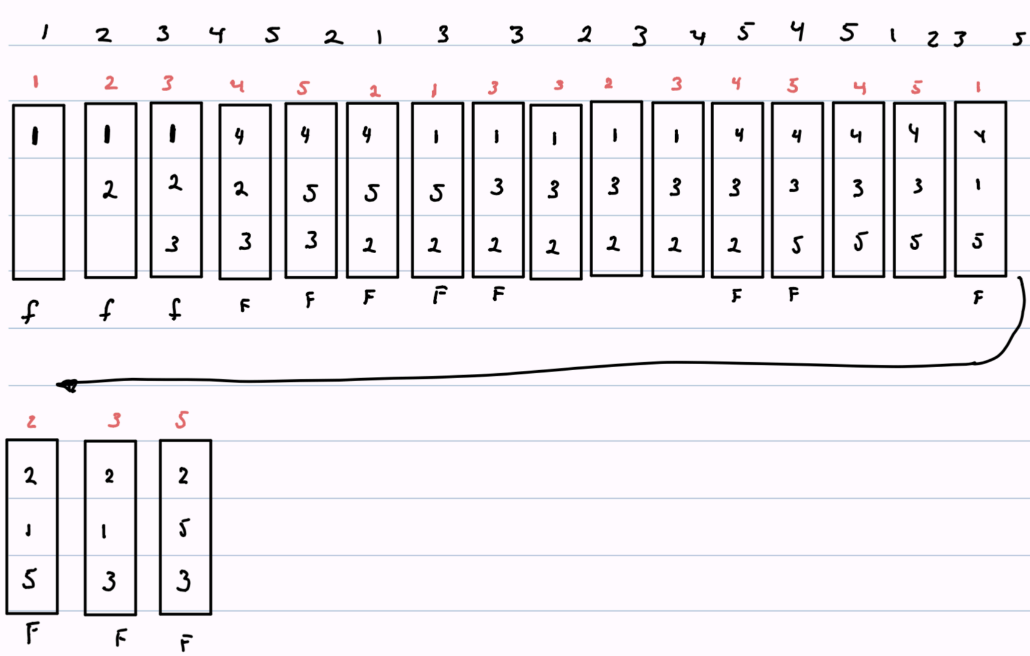
Consider the following sequence of page references (each element represents a page number in a virtual memory system):

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

Show how many page faults would occur under each of the following policies:
Assume only 3 frames are available and that they were initially empty.

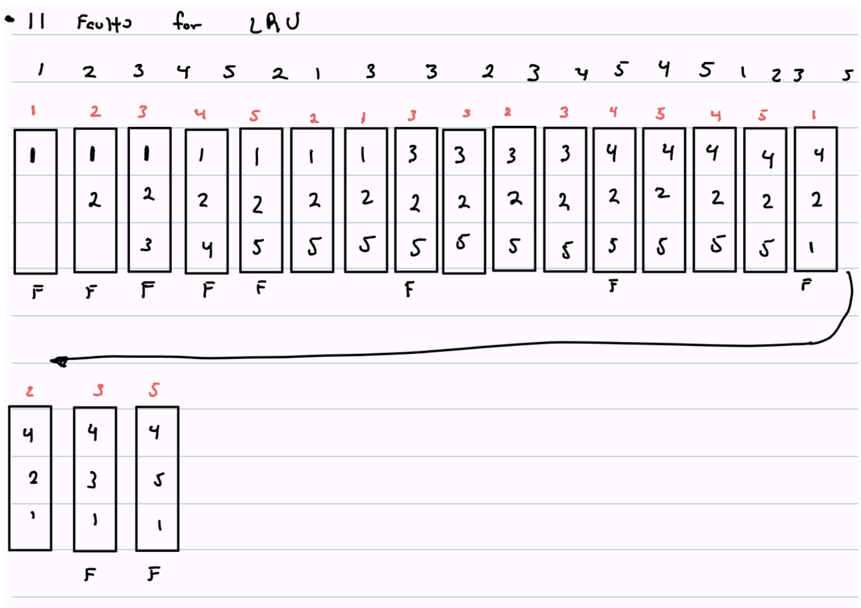
(a) FIFO. [3 pts]

• 14 Faults for FIFO

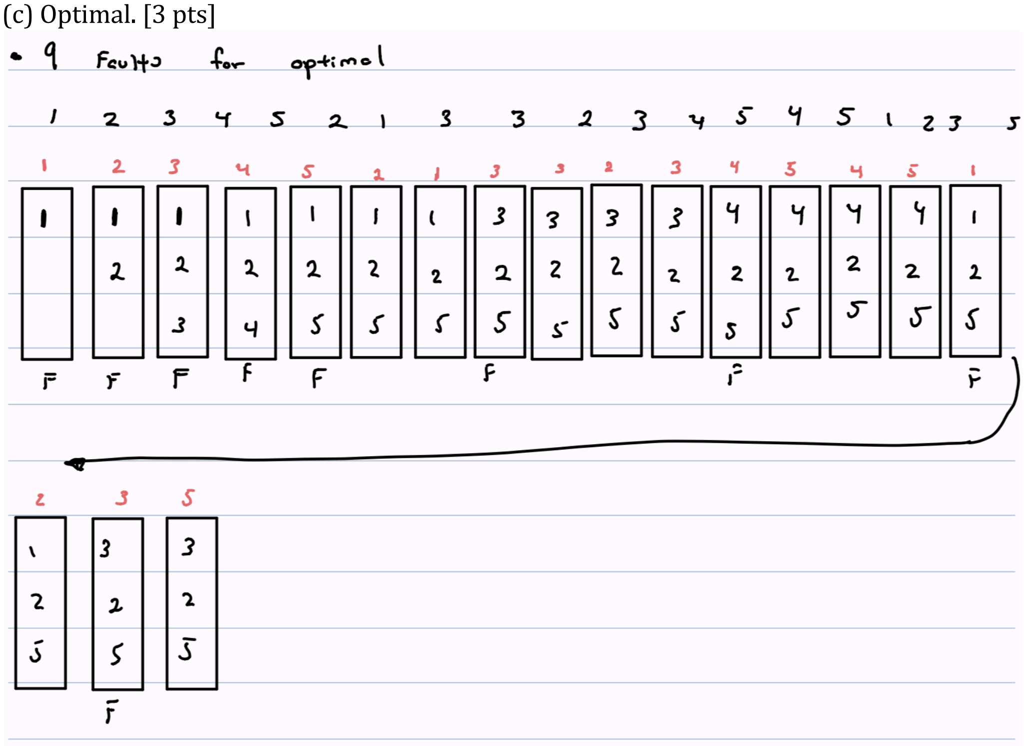


(b) LRU. [3 pts]

Jamal Rasool
Thursday, April 18, 2019



(c) Optimal. [3 pts]



Problem 4

Consider a page reference string for a process with a working set of M frames initially all empty. The page reference string is of length P with N distinct page numbers in it. For any page replacement algorithm, answer the following [3 pts each]:

(a) What is the lower bound on the number of page faults?
N will be the lower bound number in this case

(b) What is the upper bound on the number of page faults?
P is the upper bound case based upon the algorithm

Problem 5

Consider the following requests for tracks that arrived to a hard disk in that order: 98, 183, 37, 122, 14, 124, 65, 67. The hard disk has 200 tracks total and the head is currently positioned on track 53. Answer the following [3 pts each]:

(a) What is the average seek length under FIFO scheduling?
When looking at the distance between the points we find that the following equation calculates the average seek length:

45+85+146+85+108+ 110+59+2 = 640 tracks

To determine the average, seek length divide it by 8 to determine the seek length in which results in around

640 / 8 = 80 tracks per access

(b) What is the average seek length under SSTF scheduling?
12 + 2 + 30 + 23 + 84 + 24 + 2 + 59 = 236

then divide by 8

then you get around 29.5 tracks or 30 tracks per access as being the average seek length

(c) What is the average seek length under SCAN scheduling? Assume SCAN goes towards track 0 first and does not use LOOK.

16 + 23 + 14+ 65 + 2 + 31 + 24 + 2 + 59 = 236

Then again divide by 8

Then you get around 29.5 also similar to SSTF in this case in which yields a value of 30 tracks per access