CSCE 410/611 Homework #3

Alexia Perez
alexia_perezv@tamu.edu
List of people you worked with on homework – None.

Question 1: (CSCE 611 – 5 points, CSCE 410 – 7.5 points)

Consider an MMU with 16 4-KB pages. The incoming virtual address's lower 12 bits is the offset, and the upper 4 bits is the page table index. The middle column lists physical frame number.

Compute the sixteen-bit physical address (and convert to base 10) given the following page table and incoming virtual address:

HINT: Work in binary then covert to decimal.
Incoming virtual address: 12296 or 0011 0000 0000 1000

Page #	Frame #	present /absent
15	000	0
14	000	0
13	000	0
12	000	0
11	111	1
10	000	0
9	101	1
8	000	0
7	000	0
6	000	0
5	011	1
4	100	1
3	001	1
2	110	1
1	001	1
0	010	1

The page table index for the given incoming virtual address = 0011 = 3 (decimal). Therefore, the frame number = 001.

We know that physical address is = frame number + offset. Therefore, the physical address in binary = 001 0000 0000 1000.

We can easily convert this number to HEX = 0x1008 and going from HEX to decimal is much easier: $16^3 * 1 + 16^2 * 0 + 16^1 * 0 + 16^0 * 8 = 16^3 + 8 = 4096 + 8 =$ 4104.

Question 2: (CSCE 611 – 5 points, CSCE 410 – 10 points)

A computer has four page frames. The time of loading, time of last access and the R(referenced) and M(modified) bits are shown below (the times are in clock ticks).

Page	Loaded	Last Ref. (ticks)	R bit	M bit
0	126	280	0	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	0

(a) Which page will Not Recently Used (NRU) replace?
NRU will replace pages 0 and 2 since they both have R and M bits = 0.

(b) Which page will Least Recently Used (LRU) replace?
LRU will replace page 3 since it was used less recently than the others.

(c) Which page will second chance replace?
2nd change would replace page 0 since it has reference bit = 0 and its loaded time is the least (out of the pages with R = 0).

(d) Which page is computationally the most expensive to replace?
Page 3, since its R = 1 but M = 0, meaning it was recently used but clean, and therefore will probably be used again soon, so deleting it would be very unproductive.

Question 3: (CSCE 611 – 10 points, CSCE 410 – 12.5 points)

Given reference string R = 0 2 1 3 5 4 6 3 7 4 7 3 3 5 5 3 1 1 1 7 1 3 4 1 and memory M below; where M is has four page frames represented by the heavy outlined boxes. Pages (represented by their page numbers) are in memory if they appear in the upper heavy outlined boxes and have been paged out to disk if they are in the lower boxes.

R=	0	2	1	3	5	4	6	3	7	4	7	3	3	5	5	3	1	1	1	7	1	3	4	1
	0	2	1	3	5	4	6	3	7	4	7	3	3	5	5	3	1	1	1	7	1	3	4	1
		0	2	1	3	5	4	6	3	7	4	7	7	3	3	5	3	3	3	1	7	1	3	4
			0	2	1	3	5	4	6	3	3	4	4	7	7	7	5	5	5	3	3	7	1	3
				0	2	1	3	5	4	6	6	6	6	4	4	4	7	7	7	5	5	5	7	7
					0	2	1	1	5	5	5	5	5	6	6	6	4	4	4	4	4	4	5	5
						0	2	2	1	1	1	1	1	1	1	1	6	6	6	6	6	6	6	6
							0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Answer the following questions:

a) **How many page faults occur?**
   11 page faults
b) **What paging algorithm is being used?**
   Least Recently Used page replacement algorithm (LRU).
c) **Is the paging algorithm using a stack algorithm? Why or why not?**
   Yes, the LRU page replacement algorithm is a stack-based algorithm, as "every time a page is referenced it is moved to the top of the stack, so the top n pages of the stack are the n most recently used pages. If the number of frames is incremented to n+1, the top of the stack will still have the n+1 most recently used pages."
d) **Briefly describe Belady's anomaly.**
   "In general, you would expect the cache hit rate to increase (get better) when the cache gets larger. But in this case, with FIFO, it gets worse! Cal   culate the hits and misses yourself and see. This odd behavior is generally referred to as Belady's Anomaly."
e) **Does the paging algorithm in this problem exhibit Belady's anomaly? Why or why not?**
   No, it does not, as this is a stack-based algorithm and it "assigns priority to a page that is independent of the number of page frames."

Sources:

Question #3:

Belady's anomaly in page replacement algorithms. GeeksforGeeks. (2021, September 27). Retrieved February 12, 2022, from https://www.geeksforgeeks.org/beladys-anomaly-in-page-replacement-algorithms/#:~:text=In%20LRU%20algorithm%20every%20time,1%20most%20recently%20used%20pages.

Arpaci-Dusseau, R. H., & Arpaci-Dusseau, A. C. (2018). *Operating systems: Three easy pieces*. Arpaci-Dusseau Books, LLC.