Stamatios Morellas
COM S 352 - Homework 9
Due 4/17/20 @ 11:59pm

(8 points) Exercise 10.16 [a and c]

a) Yes, the thread state will change if it acquires a page fault. When a page fault occurs, a thread will change states from running to blocked. While the process waits for the I/O operation to finish, the OS will load the page table onto a free memory frame in order to then update the table with new mapping. Once the OS is finished with this, it will change the process state from blocked to ready, as shown in the diagram.

c) No. If an address reference is resolved in the page table, then this means that an I/O operation is not needed since the page will already be loaded in the main memory.

(12 points) Exercise 10.18

Virtual address space → 12-bit
Physical address space → 12-bit
Page size → 256 bytes

- 0x2A1 → 0xAB
- 0x4E6 → 0xF5
- 0x94A → 0x4B
- 0x316 → 0x1F

(30 points) Problem 3: Consider the following page reference string:

7, 2, 3, 1, 2, 5, 3, 4, 6, 7, 7, 1, 0, 5, 4, 6, 2, 3, 0, 1.

Assuming demand paging with three frames, how many page faults would occur for the following replacement algorithms? You must show your work to receive full credit.

- **LRU replacement**
- **FIFO replacement**
- **Optimal replacement**

For LRU replacement:

7	7	7	1	1	1	3	3	3	7	7	7	7	5	5	5	2	2	2	1
	2	2	2	2	2	2	4	4	4	4	1	1	1	4	4	4	3	3	3
		3	3	3	5	5	5	6	6	6	6	0	0	0	6	6	6	0	0
Pf	Pf	Pf	Pf		Pf	Pf	Pf	Pf	Pf		Pf	Pf	Pf	Pf	Pf	Pf	Pf	Pf	Pf

Page faults for LRU: **18**

For FIFO replacement:

7	7	7	1	1	1	1	1	6	6	6	6	0	0	0	6	6	6	0	0
	2	2	2	2	5	5	5	5	7	7	7	7	5	5	5	2	2	2	1
		3	3	3	3	3	4	4	4	4	1	1	1	4	4	4	3	3	3
Pf	Pf	Pf	Pf		Pf		Pf	Pf	Pf		Pf	Pf	Pf	Pf	Pf	Pf	Pf	Pf	Pf

Page faults for FIFO: **17**

For Optimal replacement:

7	**7**	**7**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	2	**2**	**2**	**2**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**4**	**6**	**2**	**3**	**3**	**3**
		3	**3**	**3**	**3**	**3**	**4**	**6**	**7**	**7**	**7**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Pf	Pf	Pf	Pf		Pf		Pf	Pf	Pf			Pf		Pf	Pf	Pf	Pf		

Page faults for Optimal: **13**

(18 points) Exercise 10.29 [a through f]

a) Install a faster CPU – No
b) Install a bigger paging disk – No
c) Increase the degree of multiprogramming – No
d) Decrease the degree of multiprogramming – Yes
e) Install more main memory – Yes, paging not required from disks
f) Install a faster hard disk or multiple controllers with multiple hard disks – Yes, higher rate of production from disk so CPU will get data faster.

(12 points) Exercise 10.37

Thrashing is caused by not allocating enough for the minimum number of pages that are required by a process, which will force it to page fault indefinitely. The system can detect it by monitoring the levels of CPU utilization. Thrashing can be eliminated by reducing the level of multiprogramming.

(10 points) Exercise 10.39

When delta is small, the set of pages for a process could be less than actual value, which will allow a process to be scheduled even though all of its pages are not resident. This can lead to indefinite page faulting for a process. When delta is large, the opposite will occur; the resident set of pages for a particular process will be overestimated, which could prevent other processes from being scheduled even if they have all of the required resident pages. This will not likely cause page faulting.