Homework 9

Gavin Monroe - ComS 352

1)A simplified view of thread states is ready, running, and blocked, where a thread is either ready and waiting to be scheduled, is running on the processor, or is blocked (for example, waiting for I/O).
Assuming a thread is in the running state, answer the following ques- tions, and explain your answers:
a. Will the thread change state if it incurs a page fault? If so, to what state will it change?
c. Will the thread change state if an address reference is resolved in the page table? If so, to what state will it change?

a) On the off chance that a page shortcoming brings about, at that point it goes to blocked state from the running in light of the fact that here the page deficiency occurred and for that it is expected to happen an I/O activity , so that by this another page can occur in memory.
b) C = In the TLB if the page table passage is missing so that is fundamentally called tlb miss and if this is settled in page table that is we will utilize the page no. for the ordering of the table and page is already in fundamental stockpiling so string will stay in running state and let say in the event that page is absent from the primary stockpiling, at that point this suggests a page shortcoming happened then it expected to go to blocked state.

2)Convert the following virtual addresses to their equivalent physical addresses in hexadecimal. All numbers are given in hexadecimal. In the case of a page fault, you must use one of the free frames to update the page table and resolve the logical address to its corresponding physical address.
• 0x2A1 • 0x4E6 • 0x94A • 0x316

• 0x2A1
160 + 1 = 161, 161 + 10 = 170 = A1+ A = AB
• 0x4E6
0XF + 0xE6 = 0x0f5
• 0x94A
0x1+ 0x4A = 0x4B
• 0x316
0X9 + 0x16 = 0x1F

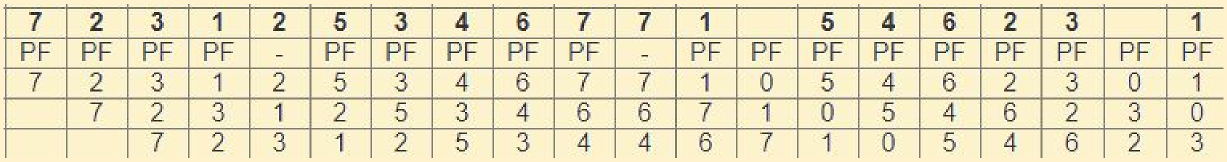
3)Consider the following page reference string:

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.

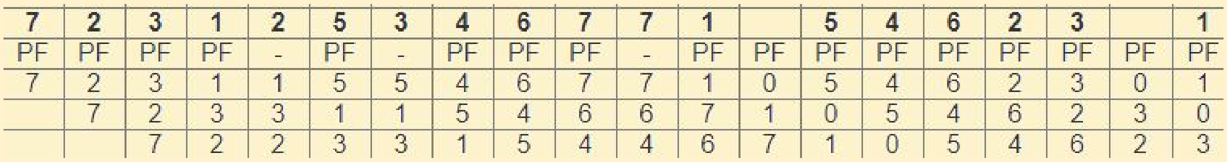
- **LRU replacement**
- **FIFO replacement**
- **Optimal replacement**

• LRU replacement = Page Faults: 18



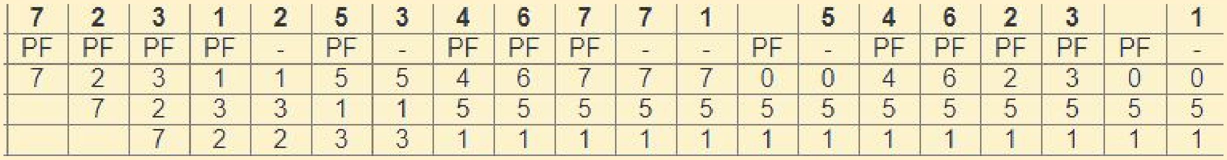
7	2	3	1	2	5	3	4	6	7	7	1		5	4	6	2	3		1
PF	PF	PF	PF	-	PF	PF	PF	PF	PF	-	PF	PF	PF	PF	PF	PF	PF	PF	PF
7	2	3	1	2	5	3	4	6	7	7	1	0	5	4	6	2	3	0	1
	7	2	3	1	2	5	3	4	6	6	7	1	0	5	4	6	2	3	0
		7	2	3	1	2	5	3	4	4	6	7	1	0	5	4	6	2	3

• FIFO replacement = Page Faults: 17



7	2	3	1	2	5	3	4	6	7	7	1		5	4	6	2	3		1
PF	PF	PF	PF	-	PF	-	PF	PF	PF	-	PF	PF	PF	PF	PF	PF	PF	PF	PF
7	2	3	1	1	5	5	4	6	7	7	1	0	5	4	6	2	3	0	1
	7	2	3	3	1	1	5	4	6	6	7	1	0	5	4	6	2	3	0
		7	2	2	3	3	1	5	4	4	6	7	1	0	5	4	6	2	3

• Optimal replacement = Page Faults: 13



7	2	3	1	2	5	3	4	6	7	7	1		5	4	6	2	3		1
PF	PF	PF	PF	-	PF	-	PF	PF	PF	-	-	PF	-	PF	PF	PF	PF	PF	-
7	2	3	1	1	5	5	4	6	7	7	7	0	0	4	6	2	3	0	0
	7	2	3	3	1	1	5	5	5	5	5	5	5	5	5	5	5	5	5
		7	2	2	3	3	1	1	1	1	1	1	1	1	1	1	1	1	1

4)Consider a demand-paging system with the following time-measured utilizations:
CPU utilization 20% Paging disk 97.7% Other I/O devices 5%
For each of the following, indicate whether it will (or is likely to) improve CPU utilization. Explain your answers.
a. Install a faster CPU.
f. Install a faster hard disk or multiple controllers with multiple hard disks.

a) Introduce a quicker CPU. Answer:NO
(Discretionary: a quicker CPU lessens the CPU usage further since the CPU will invest more energy trusting that a procedure will enter in the prepared line.)
b) Introduce a greater paging plate. Answer:NO
(Discretionary: the size of the paging plate doesn't influence the measure of memory that is expected to diminish the page shortcomings.)
c) Increment the level of multiprogramming Answer:NO
(Since each procedure would have less casings accessible and the page shortcoming rate would increment)
d) Abatement the level of multiprogramming.Answer: YES
(Discretionary: by suspending a portion of the procedures, different procedures will have more edges so as to get their pages them, subsequently lessening the page flaws.)

e) Introduce increasingly primary memory. Answer:Likely
(Discretionary: more pages can stay occupant and don't expect paging to or from the circles.)
f) Introduce a quicker hard circle or various controllers with different hard plates.
Answer:Likely

5)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?

Thrashing is a phenomenon caused when the number of pages allocated to the process are less than required leading to continuous page faults, the system is able to detect it by comparing the utilization of the cpu to the multiprogramming levels, once the system detects thrashing reducing the certain levels of multiprogramming will allow the cpu to eliminate this problem.

6)

The Main source of thrashing is under designation of the quantity of pages a procedure required to finish its execution. To finish the execution of a procedure CPU looks for the page of the procedure in primary memory on the off chance that the page isn't discovered, at that point it's a page issue. To get the necessary page CPU to look in optional memory and carry it to the fundamental memory. So due to under distribution of pages, CPU invests the greater part of the energy in swapping and thrashing happens.

At the point when the size of the working set is littler, at that point the quantity of the accessible casings then there will be less page flaw however on the off chance that delta is more prominent, at that point the quantity of accessible edges then there will be a high page issue rate. The working set model depends on the idea of territory. As per this model, in view of parameter suppose X. the working set is characterized as the arrangement of pages in the latest 'X' page references. So the most late dynamic page will be in the working set. In the event that the estimation of X is little, at that point the whole region may not be secured and in the event that the estimation of X is excessively huge, at that point the working set may cover.

on the off chance that D is the all out interest for edge and k is the size of working set then D = k

m - > Number of edges accessible in fundamental memory

on the off chance that D <= m, at that point no thrashing

on the off chance that D > m, at that point thrashing would happen as some procedure would not get enough edge.