ASSIGNMENT 5

1) **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, say whether it will (or is likely to) improve CPU utilization. Explain your answers.

Install a faster CPU. – NO
The faster CPU will reduce CPU utilization which will spend more time in the CPU.
Install a bigger paging disk. – NO
Installing a bigger paging disk does not affect the CPU utilization since the paging disk does not have any improvement on the CPU utilization.
Increase the degree of multiprogramming. – NO
Increasing degree of multiprogramming will increase chances for page faults because it will decrease CPU Utilization.
Decrease the degree of multiprogramming. – YES
This will improve CPU utilization because there will be more frames for pages which will also reduce page faults.
Install more main memory. – Likely to Improve
This is likely to improve because more pages can remain in memory which will not require paging.
Install a faster hard disk or multiple controllers with multiple hard disks.-Likely to Improve
This is likely to improve the CPU utilization because the system will be able to retrieve from the disk more quickly from the hard disk.
Add prepaging to the page fetch algorithms. – Likely to Improve
This is likely to improve because this will reduce the number of page faults since the pages are fetched before they are needed.
Increase the page size. – NO
This is not going to improve the CPU utilization because the increase in page size will lead to fewer pages in memory which will increase page faults.

2) **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 caused when there is under allocation of the minimum number of pages required by a process which forces continuous page faults.

Thrashing can be detected by comparing and evaluating the level of CPU use to the level of multiprogramming.

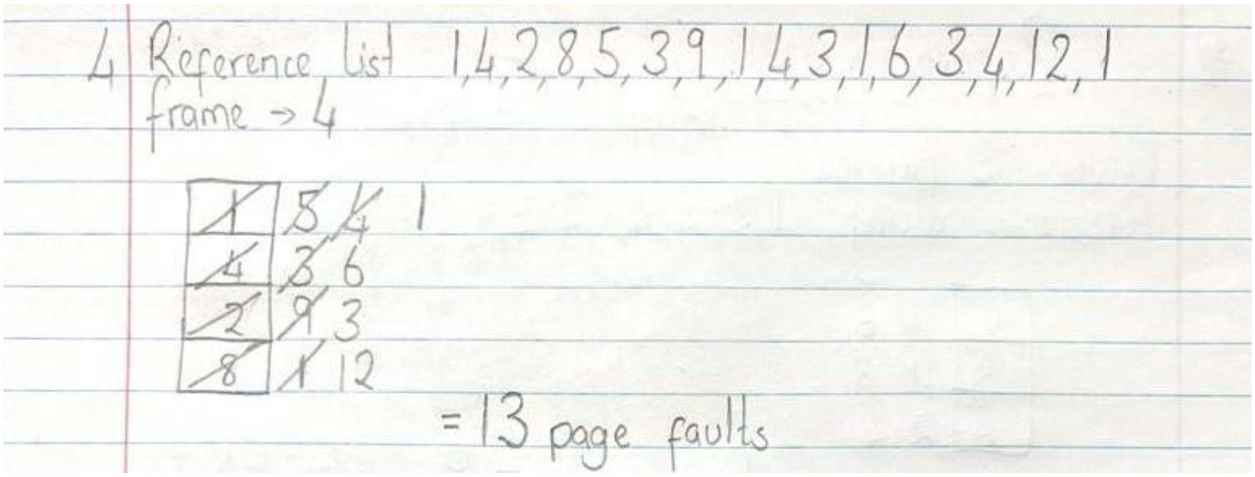
Thrashing can be eliminated by providing a process with as many frames as it requires.

3) **A process needs to access its pages in the order of 1, 4, 2, 8, 5, 3, 9, 1, 4, 3, 1, 6, 3, 4, 12, 1. The system will be using pure demand paging. What is the minimum number of pages faults that this process will generate? What is the maximum number of page faults that could be generated?**

The minimum number of page faults that the process will generate are the unique references in the reference string in our case they are 9 page faults and the maximum number of page faults are 16 page faults since in the reference string all numbers are different following each other.

Minimum = 9 page faults and maximum = 16 page faults

4) **Given the page reference list of 1, 4, 2, 8, 5, 3, 9, 1, 4, 3, 1, 6, 3, 4, 12, 1; How many page faults will be generated if the processes is assigned 4 frames of system memory and the system uses a First in First out page replacement algorithm?**

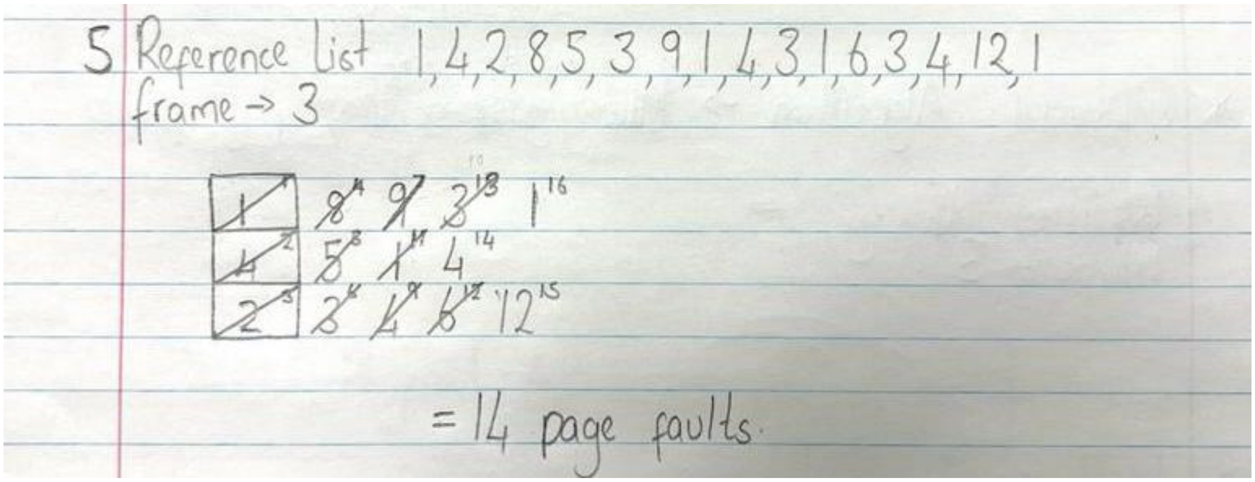


4 Reference list 1,4,2,8,5,3,9,1,4,3,1,6,3,4,12,1
frame → 4

~~1~~ ~~5~~ ~~4~~ 1
~~4~~ ~~3~~ 6
~~2~~ ~~9~~ 3
~~8~~ ~~1~~ 12

= 13 page faults

5) **Given the page reference list of 1, 4, 2, 8, 5, 3, 9, 1, 4, 3, 1, 6, 3, 4, 12, 1; How many page faults will be generated if the processes is assigned 3 frames and the system using a Least recently used page replacement algorithm?**



5 Reference list 1,4,2,8,5,3,9,1,4,3,1,6,3,4,12,1
frame → 3

~~1~~ ~~8~~^{4} ~~9~~^{7} ~~3~~^{13} 1^{16}
~~4~~^{2} ~~5~~^{5} ~~1~~^{11} 4^{14}
~~2~~^{3} ~~3~~^{6} ~~4~~^{9} ~~6~~^{12} 12^{15}

= 14 page faults.