1.What is Thrashing? When a process said to be thrashed and discuss the   in details about the various solutions for thrashing?

Thrashing is a situation where the system spends more time on paging (transferring pages between main memory and secondary storage) than on executing processes. [Thrashing occurs when the degree of multiprogramming (the number of processes in memory) is too high and the available frames (fixed-size units of physical memory) are too few to accommodate the current working set (the set of pages that are actively used by the processes) 1](https://www.geeksforgeeks.org/techniques-to-handle-thrashing/)[2](https://www.javatpoint.com/what-is-thrash)[3](https://www.indiastudychannel.com/resources/118085-Thrashing-Causes-of-Thrashing-and-How-to-Prevent-Thrashing-in-Operating-System.aspx).

Thrashing causes several problems, such as:

* It reduces CPU utilization by increasing the number of page faults and context switches.
* It increases disk I/O and swapping activity by transferring more pages between main memory and secondary storage.
* It degrades system performance and responsiveness by delaying the execution of processes.

A process is said to be thrashed when it does not have enough frames to hold its current working set. As a result, it suffers from frequent page faults and has to swap in and out pages from secondary storage. This reduces its progress and increases its effective access time (the average time required to access a page in memory).

There are different solutions for thrashing, such as:

* Working set model: This model is based on the concept of locality, which states that a process tends to access a relatively small and contiguous set of pages over a period of time. The working set model tries to estimate the current working set of each process by keeping track of the pages that have been accessed recently. The system then allocates enough frames to each process to accommodate its working set. If a process exceeds its allocated frames, some pages are replaced using a local replacement policy (which only selects pages that belong to the same process). If a process does not use all its allocated frames, some frames are reclaimed by the system and given to other processes. The working set model aims to prevent thrashing by adjusting the degree of multiprogramming according to the size and variation of the working sets [1](https://www.geeksforgeeks.org/techniques-to-handle-thrashing/)[2](https://www.javatpoint.com/what-is-thrash)[3](https://www.indiastudychannel.com/resources/118085-Thrashing-Causes-of-Thrashing-and-How-to-Prevent-Thrashing-in-Operating-System.aspx).
* Page fault frequency: This scheme is based on the observation that the page fault rate of a process is inversely proportional to the number of frames allocated to it. If a process has too few frames, it will have a high page fault rate. If a process has too many frames, it will have a low page fault rate. The system tries to maintain the page fault rate of each process within a predefined range by increasing or decreasing the number of frames allocated to it. If the page fault rate is too high, more frames are allocated. If the page fault rate is too low, some frames are deallocated. The page fault frequency scheme aims to prevent thrashing by balancing the frame allocation among the processes according to their page fault rates [1](https://www.geeksforgeeks.org/techniques-to-handle-thrashing/)[2](https://www.javatpoint.com/what-is-thrash)[3](https://www.indiastudychannel.com/resources/118085-Thrashing-Causes-of-Thrashing-and-How-to-Prevent-Thrashing-in-Operating-System.aspx).
* Load control: This method is based on the idea that thrashing can be avoided by limiting the number of processes that are loaded into memory at any given time. The system monitors the CPU utilization and the degree of multiprogramming and adjusts them accordingly. If the CPU utilization is too low, more processes are admitted into memory to increase the degree of multiprogramming. If the CPU utilization is too high, some processes are suspended or swapped out to reduce the degree of multiprogramming. [The load control scheme aims to prevent thrashing by regulating the level of concurrency in the system according to the available resources](https://www.geeksforgeeks.org/techniques-to-handle-thrashing/)

2.Given the reference to the following pages by a program 0, 9, 0, 1, 8, 1, 8, 7, 8, 7, 1, 2, 8, 2, 7, 8, 2, 3, 8, 3. There are 3 page frames in memory. Calculate the number of page faults occurs with respect to Optimal Page Replacement Algorithm and FIFO Page Replacement Algorithm.

Optimal Page Replacement Algorithm:

This algorithm replaces the page that will not be used for the longest time in the future. It is the best possible algorithm to minimize the number of page faults, but it is impossible to implement in practice because it requires the knowledge of future page references.

The following table shows how the optimal page replacement algorithm works for the given page reference string and 3 page frames:

| **Page Reference** | **Page Frames** | **Page Fault** |
| --- | --- | --- |
| 0 | 0 \_ \_ | Yes |
| 9 | 0 9 \_ | Yes |
| 0 | 0 9 \_ | No |
| 1 | 0 9 1 | Yes |
| 8 | 8 9 1 | Yes |
| 1 | 8 9 1 | No |
| 8 | 8 9 1 | No |
| 7 | 8 7 1 | Yes |
| 8 | 8 7 1 | No |
| 7 | 8 7 1 | No |
| 1 | 8 7 1 | No |
| 2 | 2 7 1 | Yes |
| 8 | 2 7 8 | Yes |
| 2 | 2 7 8 | No |
| 7 | 2 7 8 | No |
| 8 | 2 7 8 | No |
| 2 | 2 \_ \_ | Yes |
| 3 | \_ \_ \_ | Yes |
| \_ \_ \_ |  |  |

The number of page faults with optimal page replacement algorithm is **10**.

FIFO Page Replacement Algorithm:

This algorithm replaces the page that has been in memory for the longest time. It is simple and easy to implement, but it may not perform well because it does not consider the frequency or recency of page references.