



DEPARTMENT OF AIML
23CST202- OPERATING SYSTEMS
II YEAR IV SEM AIML-B
UNIT 3-MEMORY MANAGEMENT
TOPIC –THRASHING

THRASHING

In case, if the page fault and swapping happens very frequently at a higher rate, then the operating system has to spend more time swapping these pages. This state in the operating system is termed thrashing. Because of thrashing the CPU utilization is going to be reduced.

Let's understand by an example, if any process does not have the number of frames that it needs to support pages in active use then it will quickly page fault. And at this point, the process must replace some pages. As all the pages of the process are actively in use, it must replace a page that will be needed again right away. Consequently, the process will quickly fault again, and again, and again, replacing pages that it must bring back in immediately. This high paging activity by a process is called thrashing.

During thrashing, the CPU spends less time on some actual productive work spend more time swapping.

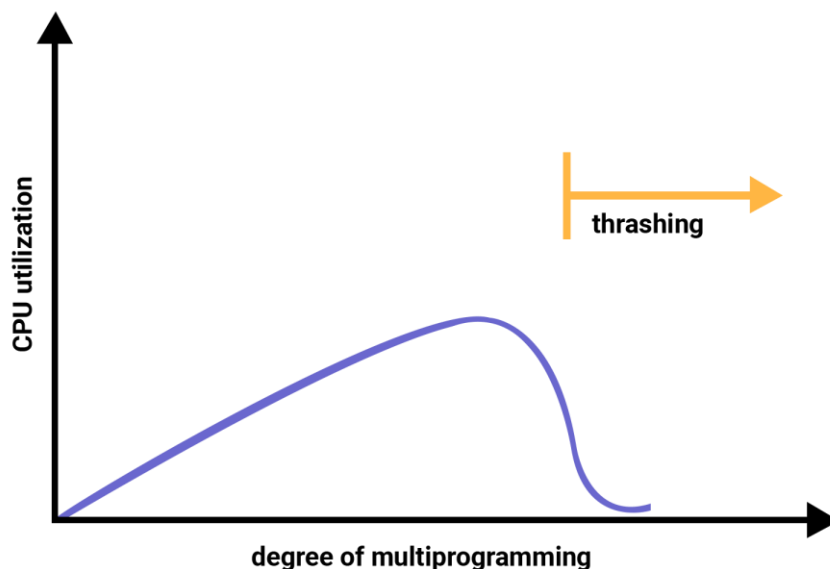


Figure: Thrashing

Causes of Thrashing



Thrashing affects the performance of execution in the Operating system. Also, thrashing results in severe performance problems in the Operating system.

When the utilization of CPU is low, then the process scheduling mechanism tries to load many processes into the memory at the same time due to which degree of Multiprogramming can be increased. Now in this situation, there are more processes in the memory as compared to the available number of frames in the memory. Allocation of the limited amount of frames to each process.

Whenever any process with high priority arrives in the memory and if the frame is not freely available at that time then the other process that has occupied the frame is residing in the frame will move to secondary storage and after that this free frame will be allocated to higher priority process.

We can also say that as soon as the memory fills up, the process starts spending a lot of time for the required pages to be swapped in. Again the utilization of the CPU becomes low because most of the processes are waiting for pages.

Thus a high degree of multiprogramming and lack of frames are two main causes of thrashing in the Operating system.

Effect of Thrashing

At the time, when thrashing starts then the operating system tries to apply either the **Global page replacement** Algorithm or the **Local page replacement** algorithm.

Global Page Replacement

The Global Page replacement has access to bring any page, whenever thrashing found it tries to bring more pages. Actually, due to this, no process can get enough frames and as a result, the thrashing will increase more and more. Thus the global page replacement algorithm is not suitable whenever thrashing happens.

Local Page Replacement

Unlike the Global Page replacement, the local page replacement will select pages which only belongs to that process. Due to this, there is a chance of a reduction in the thrashing. As it is also proved that there are many disadvantages of Local Page replacement. Thus local page replacement is simply an alternative to Global Page replacement.

Techniques used to handle the thrashing

As we have already told you the Local Page replacement is better than the Global Page replacement but local page replacement has many disadvantages too, so it is not suggestible. Thus given below are some other techniques that are used:



Working-Set Model

This model is based on the assumption of the locality. It makes the use of the parameter w in order to define the working-set window. The main idea is to examine the most recent w page reference. What locality is saying, the recently used page can be used again, and also the pages that are nearby this page will also be used?

1. Working Set

The set of the pages in the most recent w page reference is known as the working set. If a page is in active use, then it will be in the working set. In case if the page is no longer being used then it will drop from the working set w times after its last reference.

The working set mainly gives the approximation of the locality of the program.

The accuracy of the working set mainly depends on w what is chosen?

This working set model avoids thrashing while keeping the degree of multiprogramming as high as possible.

2. Page Fault Frequency

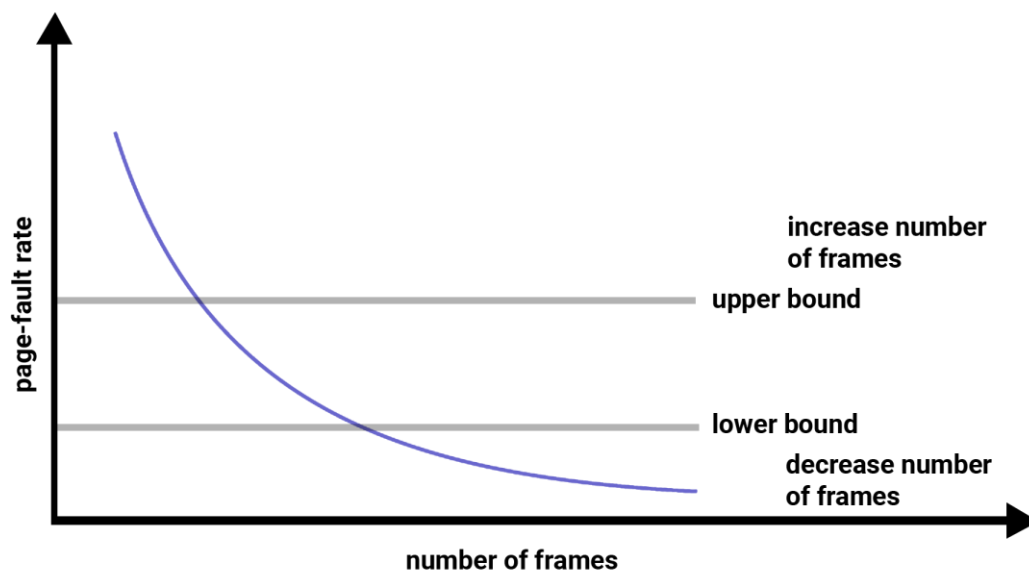


Figure: Page fault frequency

The working-set model is successful and its knowledge can be useful in preparing but it is a very clumpy approach in order to avoid thrashing. There is another technique that is used to avoid thrashing and it is Page Fault Frequency(PFF) and it is a more direct approach.



The main problem is how to prevent thrashing. As thrashing has a high page fault rate and also we want to control the page fault rate.

When the Page fault is too high, then we know that the process needs more frames. Conversely, if the page fault-rate is too low then the process may have too many frames.

We can establish upper and lower bounds on the desired page faults. If the actual page-fault rate exceeds the upper limit then we will allocate the process to another frame. And if the page fault rate falls below the lower limit then we can remove the frame from the process.

Thus with this, we can directly measure and control the page fault rate in order to prevent thrashing.