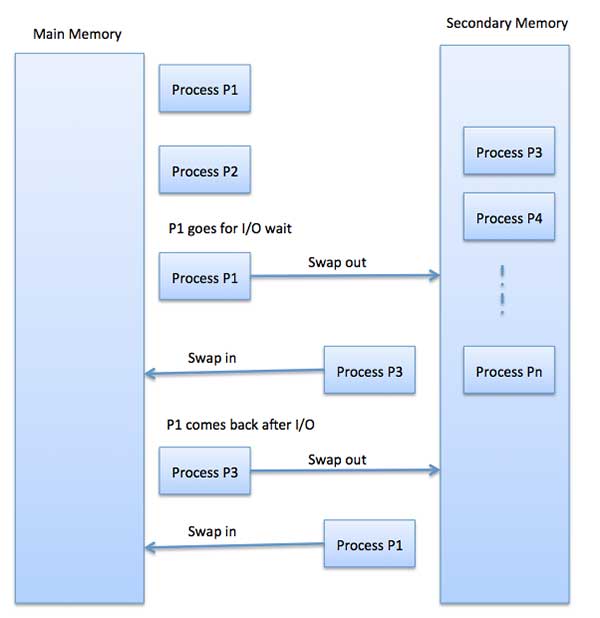
## Q1 Swapping

Swapping is a mechanism in which a process can be swapped temporarily out of main memory (or moved) to secondary storage (disk) and make that memory available to other processes. Later, the system swaps back the process from the secondary storage to main memory. Though the swapping process usually affects performance, it also helps in running multiple and big processes in parallel and that is the reason, not possible in practice, as operating system cannot know future requests.

The total time taken by swapping process includes the time it takes to move the entire process to a secondary disk and then to copy the process back to memory, as well as the time the process takes to regain main memory.



## Q2 Memory Fragmentation

As processes are loaded and removed from memory, the free memory space is broken into little pieces. It happens after sometimes that processes cannot be allocated to memory blocks considering their small size and memory blocks remains unused. This problem is known as Fragmentation. It is of two types −

* External – Total memory space is enough to satisfy a request or to reside a process in it, but it is not contiguous, so it cannot be used. This can be reduced by compaction or shuffling memory contents to place all free memory together in one large block. To make compaction feasible, relocation should be dynamic.
* Internal – Memory block assigned to process is bigger. Some portion of memory is left unused, so another process cannot use it. It can be reduced by effectively assigning the smallest partition but large enough for the process.

## Q3 Virtual Memory

A computer can address more memory than the amount physically installed on the system. This extra memory is called virtual memory and it is a section of a hard disk set up to emulate the computer's RAM.

The main visible advantage of this scheme is that programs can be larger than physical memory. Virtual memory serves two purposes. First, it allows us to extend the use of physical memory by using disk. Second, it allows us to have memory protection, as each virtual address is translated to a physical address.

## Q4 Page Scheduling Algorithms

Page replacement algorithms are the techniques using which an Operating System decides which memory pages to swap out, when a page of memory needs to be allocated.

1. **First In First Out (FIFO) algorithm** – In this algorithm, operating system keeps track of all pages in the memory in a queue, oldest page is in the front of the queue. When a page needs to be replaced, the page in the front of the queue is selected for removal.
2. **Optimal Page algorithm** – In this algorithm, pages that will not be used for the longest period in the future are replaced. An optimal page-replacement algorithm has the lowest page-fault rate of all algorithms. This is not applicable in practice, as operating system cannot know future requests.
3. **Least Recently Used (LRU) algorithm** – In this algorithm, the page that has not been used for the longest time in main memory is the one selected for replacement. This algorithm works on the idea that the pages that have been the most used in the past are more likely to be used again in the near future.

## Q5 Demand Paging

A demand paging system is quite similar to a paging system with swapping where processes reside in secondary memory and pages are loaded only on demand, not in advance. When a context switch occurs, the operating system does not copy any of the old program’s pages out to the disk or any of the new program’s pages into the main memory. Instead, it just begins executing the new program after loading the first page and fetches that program’s pages as they are referenced.

While executing a program, if the program references a page, which is not available in the main memory because it was swapped out a little while ago, the processor treats this invalid memory reference as a page fault and transfers control from the program to the operating system to demand the page back into the memory.