

# National Institute of Technology Raipur

**Operating System**

**Term Paper**

**Swapping**

**By**

**Kunal Sachdeva**

**Roll No. 19115045**

**(5th Semester Computer Science & Engineering 2020-21)**

**Abstract**

The aim of swapping in the operating system is to access data stored on the hard disc and bring it into RAM for usage by application programs. In this paper I’m discussing the whole concept of swapping and its relation with memory management.

**Introduction**

A collection of data in a specified format might be defined as memory. It's utilized to keep track of instructions and data that's been processed. A vast array or group of words or bytes, each with its own position, makes up the memory. A computer system's principal goal is to run programs. During execution, these programs, as well as the data they access, should reside in main memory. The CPU retrieves instructions from memory based on the program counter value.

Memory management is critical for achieving multiprogramming and proper memory use. There are numerous memory management strategies available, each reflecting a different approach, and the success of each algorithm varies depending on the situation.

Why Memory Management is required:

Before and after the procedure, allocate and de-allocate memory.

To keep track of how much memory is being consumed by processes.

To keep fragmentation to a minimum.

To make the best use of the main memory.

To keep data safe while the process is running.

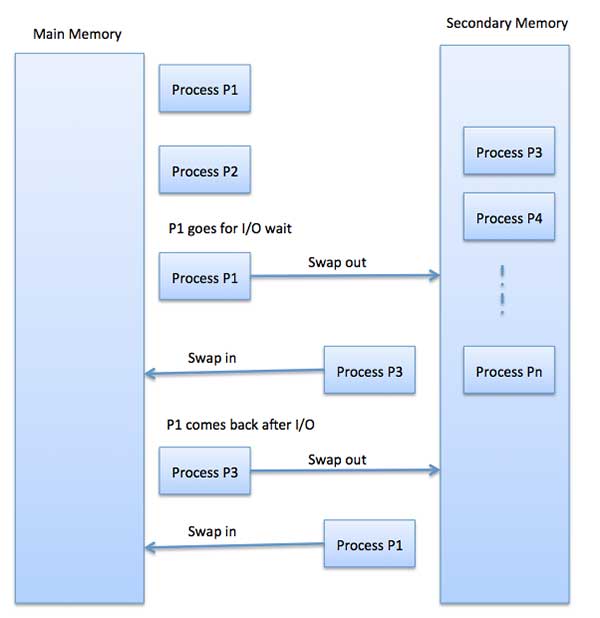
Memory management is an operating system feature that handles or manages primary memory and moves processes between main memory and disc during execution. Memory management keeps track of every memory location, regardless of whether it is used by a process or is free. It determines how much memory should be assigned to each process. It determines which processes will be given memory and when. It keeps track of when memory is freed or unallocated and updates the state accordingly.

**Purpose of using Swapping**

Swapping is a mechanism for temporarily moving a process from main memory to secondary storage (disc) and making that memory available to other processes. The system switches the process from secondary storage to main memory at a later time.

Swapping is a memory management technique that allows any process to be temporarily switched from main memory to secondary memory, freeing up main memory for other tasks. It is used to increase the use of main memory. Swap space is the location in secondary memory where the swapped-out process is stored.

The aim of swapping in the operating system is to access data stored on the hard disc and bring it into RAM for usage by application programs. It's important to note that swapping is only done when data isn't available in RAM.

Although swapping has an impact on the system's performance, it allows larger and multiple processes to operate simultaneously. Swapping is also referred to as memory compaction because of this.

Swapping has been subdivided into two concepts: Swap-in and Swap-out.

* Swap-out is a technique for transferring a process from RAM to the hard disc.
* Swap-in refers to the process of deleting a program from a hard disc and reinstalling it in main memory, or RAM.

**Example:** The time it takes to relocate the entire process to a secondary disc and then copy it back to memory, as well as the time it takes to reclaim main memory, is included in the total time taken by the swapping operation.

Assume that the user process is 2048KB in size and that the data transmission rate on the standard hard disc where the swapping would take place is roughly 1 MB per second. The 1000K procedure will take some time to move to or from memory.

2048KB / 1024KB per second

= 2 seconds

= 2000 milliseconds

In terms of in and out time, it will take a total of 4000 milliseconds, plus additional overhead while the process fights for main memory.

**Advantages of Swapping**

The Swapping approach has the following advantages and benefits:

1. The CPU uses the swapping approach to manage several tasks in a single main memory.
2. This method aids in the creation and utilization of virtual memory.
3. The CPU can accomplish multiple jobs at the same time using this technique. As a result, processes do not need to be executed for an extended period of time.
4. This method is cost-effective.
5. To increase the performance of priority-based scheduling, this strategy can be easily deployed.

**Disadvantages of Swapping**

The following are the disadvantages of the swapping technique:

1. If a resource or a variable is frequently used by the processes that are involved in the swapping process, inefficiencies may result.
2. If the swapping technique is bad, the overall strategy can increase the amount of page faults, lowering processing performance.
3. If the computer system loses power during a period of intense swapping activity, the user may lose all of the program's data.

**Note:**

* Only one process occupies the user program area of memory in a single tasking operating system, and it remains in memory until the process is completed.
* When all of the active processes in a multitasking operating system are unable to coordinate in the main memory, a process is swapped out of the main memory to allow additional processes to enter.

**Contiguous Memory Allocation:**

Both the operating system and the many client processes should rely on the main memory. As a result, memory allocation becomes a critical duty in the operating system. In most cases, memory is divided into two partitions: one for the resident operating system and another for user processes. In most cases, many user processes must be kept in memory at the same time. As a result, we must evaluate how to assign available memory to the processes waiting to be brought into memory from the input queue. Each process in adjacent memory allotment is confined in a single contiguous memory segment.

**Memory Allocation**

There are normally two partitions in main memory.

* Low Memory − The operating system is stored in this memory.
* High Memory − User processes are stored in a large amount of memory.

The following memory allocation mechanism is used by the operating system.

**Single-partition allocation**

The relocation-register strategy is employed in this type of allocation to protect user processes from each other as well as from changing operating-system code and data. The value of the shortest physical address is stored in the relocation register, whereas the range of logical addresses is stored in the limit register. The limit register must be less than each logical address.

**Multiple-partition allocation**

This method of allocation divides main memory into a number of fixed-size divisions, each of which should only contain one process. When a partition becomes available, a process is chosen from the input queue and loaded into it. The partition becomes available for another process when the process finishes.

**Difference between Paging and Swapping**

**Swapping** is the process of temporarily moving a process from main memory to secondary memory, which is faster than secondary memory. However, because RAM is limited in size, dormant processes are moved to secondary memory. The transferred time is the most important aspect of swapping, and the overall time is proportional to the quantity of memory swapped.

**Paging** is a memory allocation process in which separate non-contiguous memory blocks are given a fixed size. The average file size is 4KB. Between active pages, paging is always performed.

|  |  |
| --- | --- |
| **Swapping** | **Paging** |
| It is the process of copying the complete procedure. | It is a memory allocation technique. |
| When the entire operation is shifted to disc, it is referred to as swapping. | When a portion of a process is sent to disc, it is referred to as paging. |
| During this operation, data is moved from main memory to secondary memory. | The contiguous block of memory is made non-contiguous but of a constant size, referred to as frame or pages, in this method. |
| Without any memory management, swapping can be done. | Non-contiguous Memory Management is a technique for managing non-contiguous memory. |
| Inactive processes do the swapping. | Paging can only be done by active processes. |
| It gives instructions on how to solve the problem. | There are no recommendations for a solution in it. |