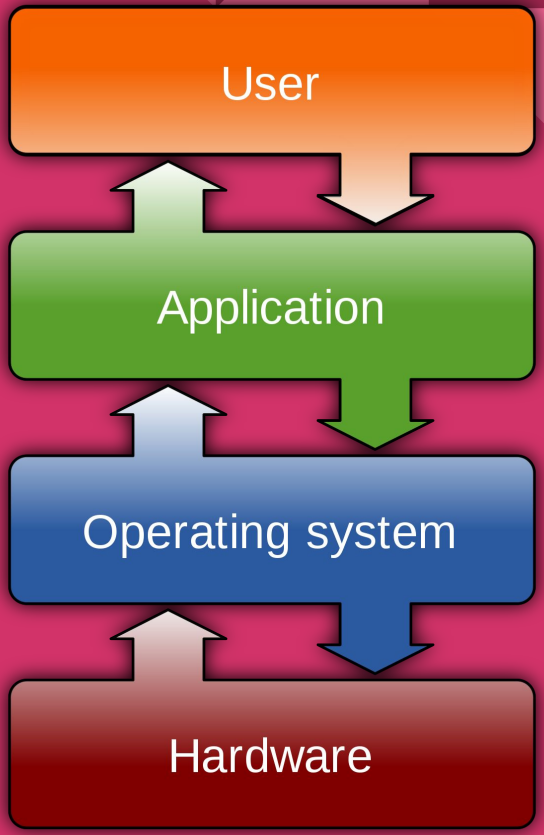
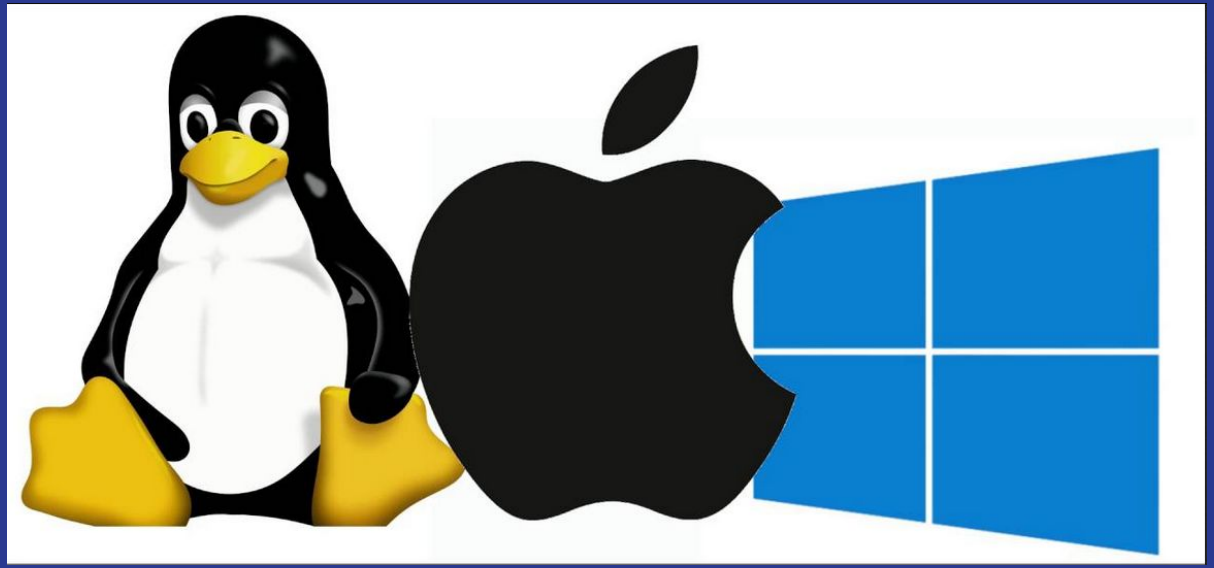
Memory in Operating System

Operating System

An Operating System (OS) is the most important system of the computer that interacts with the user and hardware.

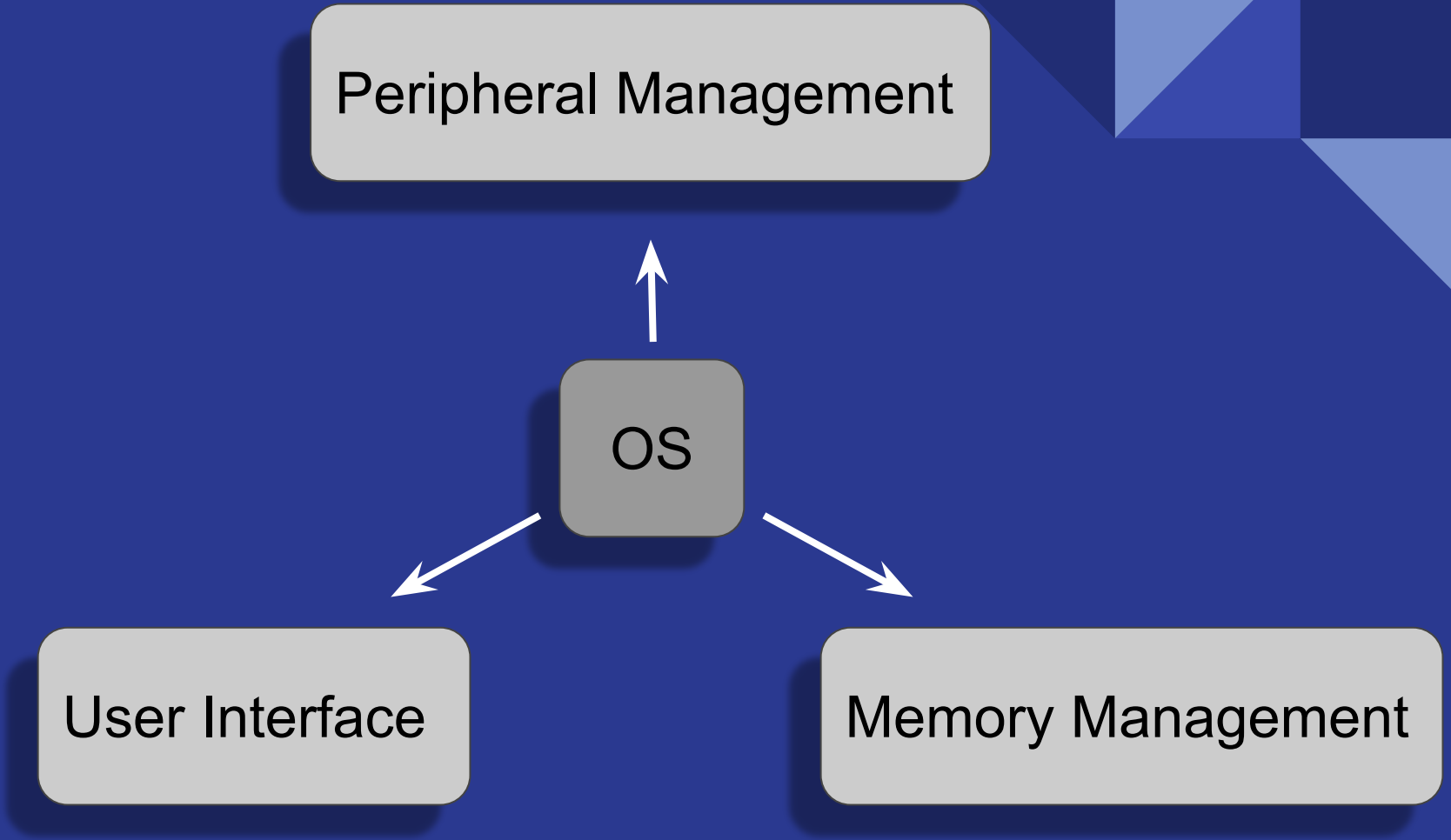


Operating System in Computers



Linux Mac, & Windows

Operating System in Smartphones


Peripheral Management
OS
User Interface
Memory Management

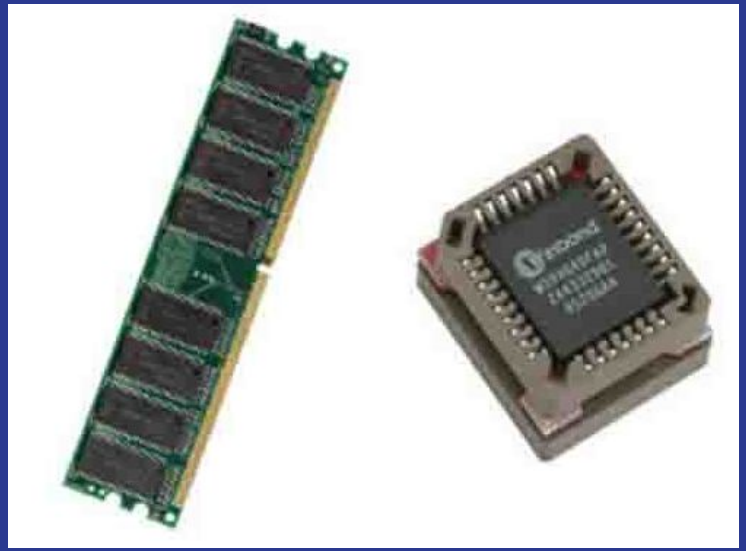
Memory in OS

Memory refers to the processes that are used to acquire, store, retain, and later retrieve information.

Primary Memory

Random-Access Memory

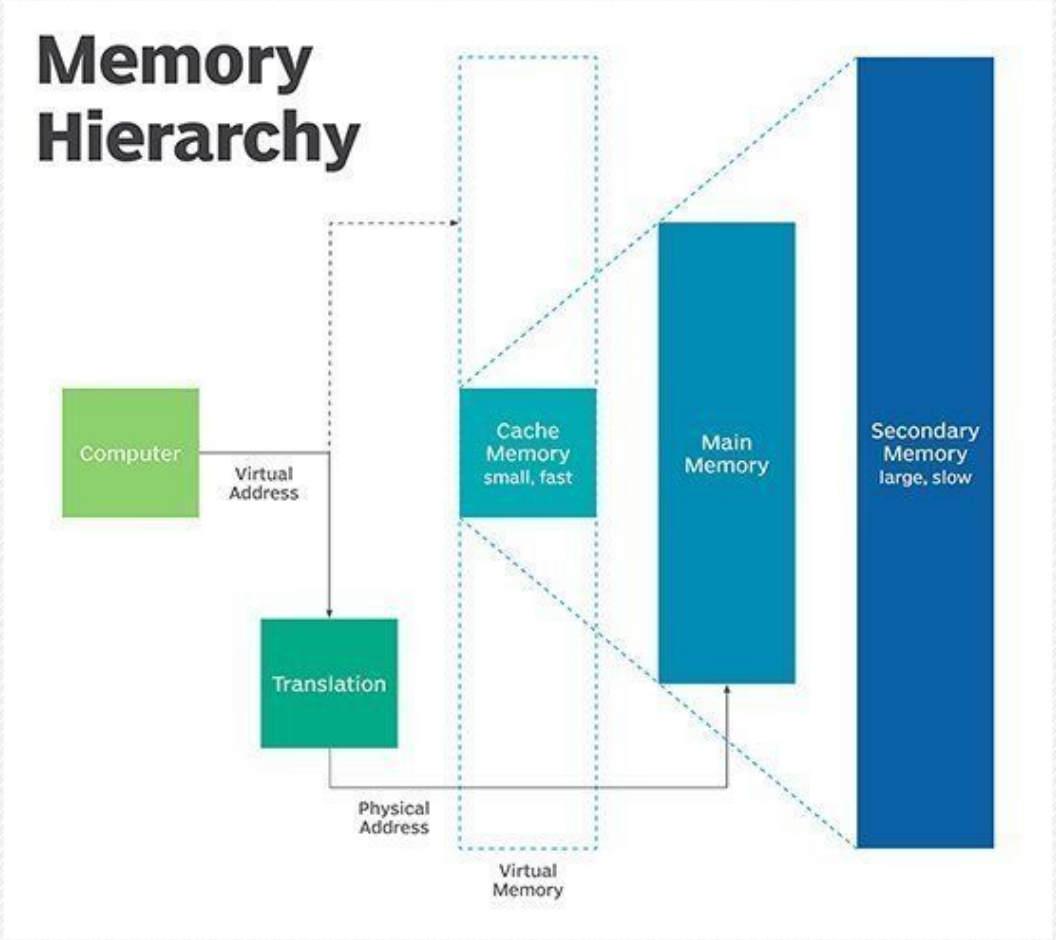
RAM



Read-Only Memory

ROM

Secondary Memory


Memory Hierarchy
Computer
Virtual Address
Translation
Physical Address
Cache Memory
small, fast
Main Memory
Secondary Memory
large, slow
Virtual Memory

Memory Management

The function of memory management is to keep track of the status of each memory location if it is free or occupied. By memory management it checks how much memory is needed to process. And it sets which process will get memory at what time.

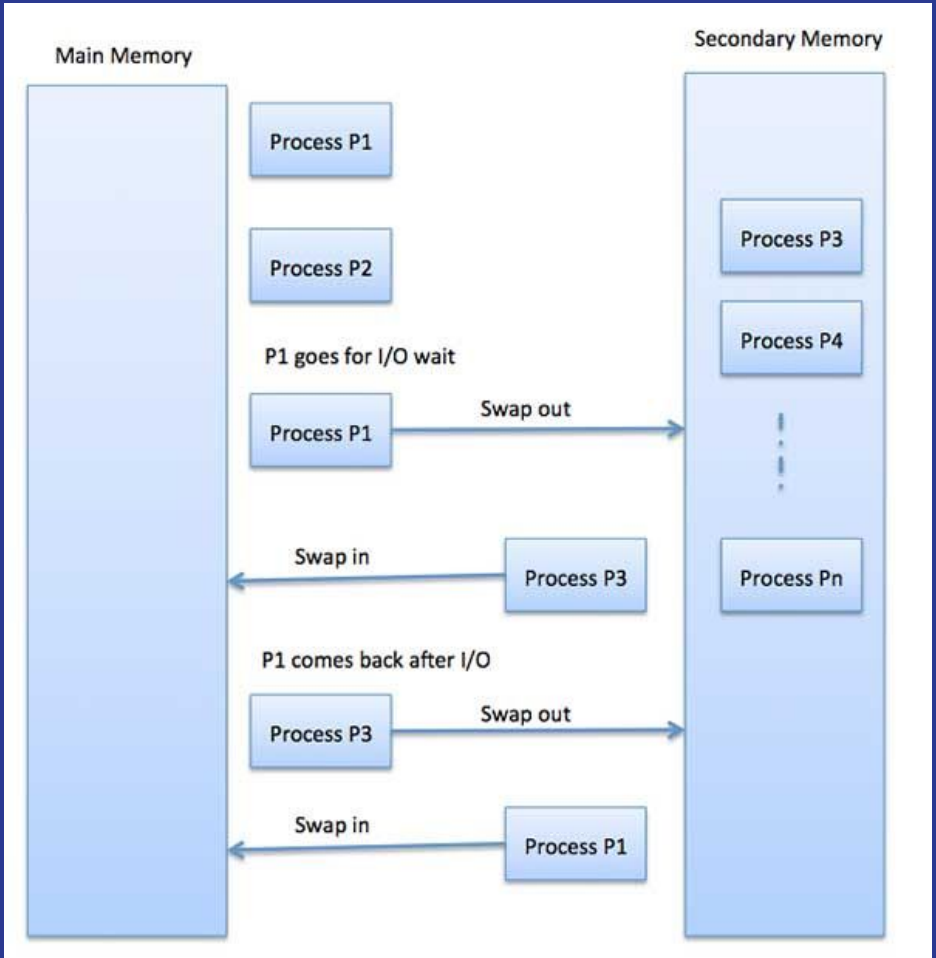
Types of Memory management

- Swapping
- Partitioning
- Paging
- Virtual Memory
- Segmentation

Swapping

It is a **temporary exchange** of a process from the main memory to the secondary memory, and is used to **improve memory utilization.**

Swapping is also referred to as a technique for **memory compaction.**



Swapping is divided into two concepts called **Swap-out** and **Swap-in**.

- **Swap-out** is to remove a process from the main memory and add them to the secondary memory.
- **Swap-in** is to remove a process from the secondary memory then bringing it back to the main memory.

Partitioning

It is a contiguous memory allocation technique that is divided into two called **Fixed Partitioning** and **Dynamic Partitioning**.

Fixed Partitioning

Equal Sizes

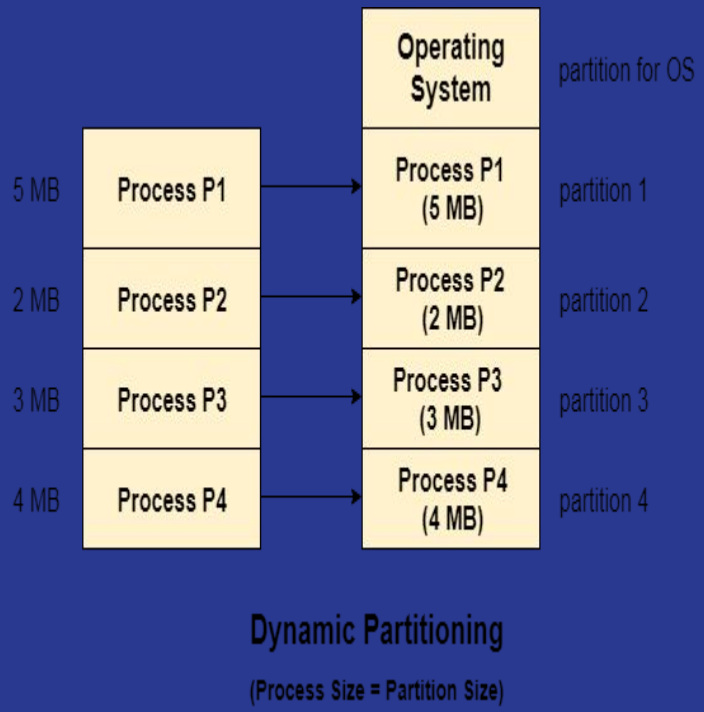
Unequal Sizes

Dividing the main memory into partitions of **equal** or **unequal sizes**.

In Fixed partitioning:

- The partition cannot overlap.
- A process must be contiguously present in a partition for the execution.

Dynamic Partitioning



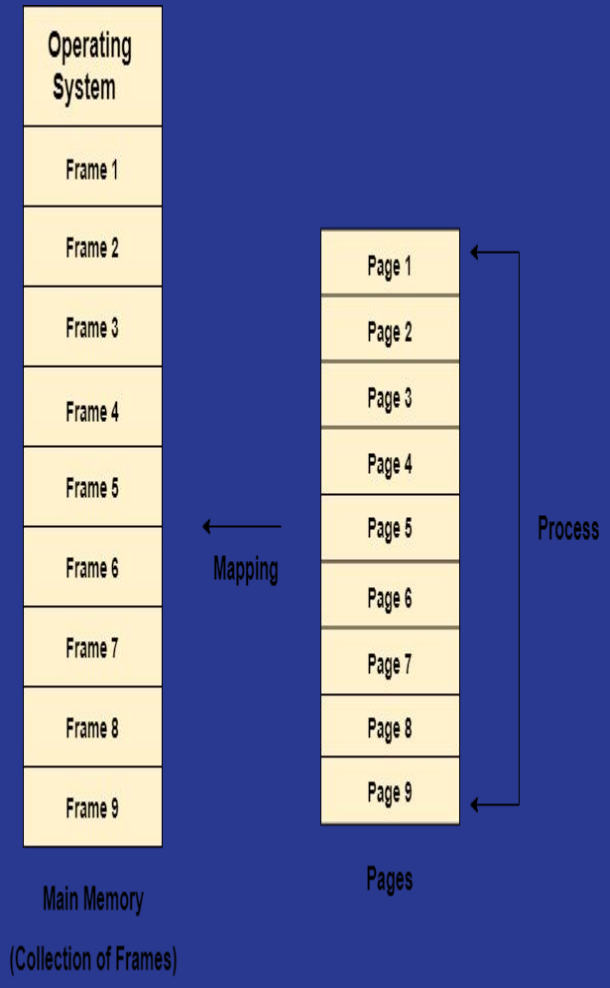
The partition is not initially declared and only declared at the time of process loading.

Each size of the partition is equal to the size of the process.

Paging

Allows the operating system to retrieve processes from the secondary memory to the main memory.

Each process is divided in the form of **pages** and the main memory is also divided into **small fixed-size blocks** of physical memory called **frames**, which are similar in size to pages.



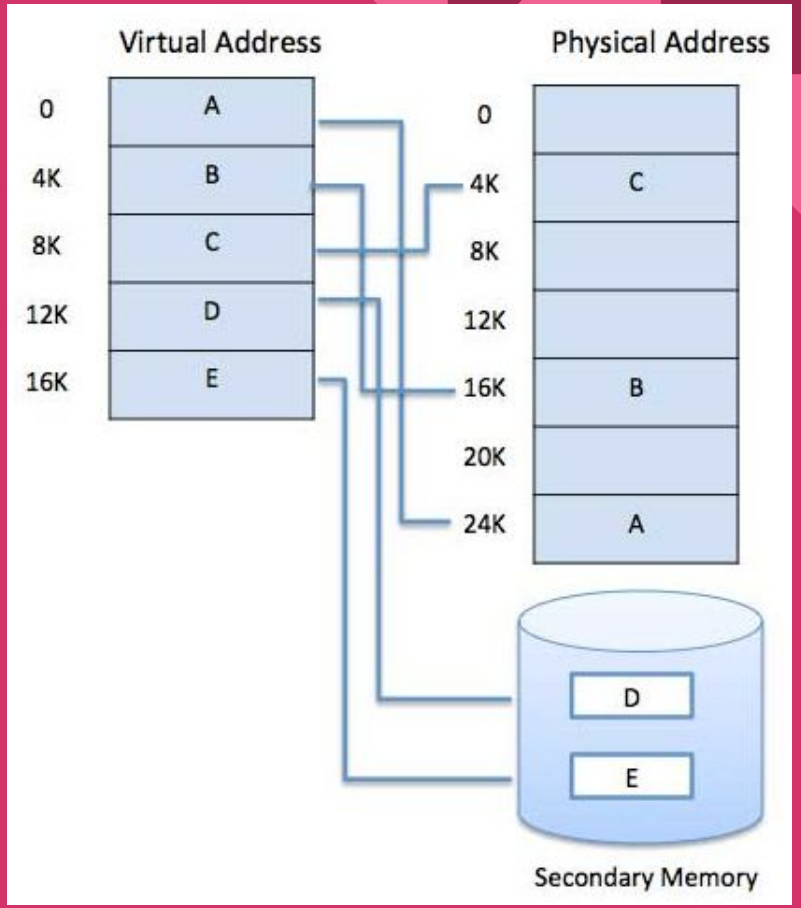
A single page of a process is to be stored in a single frame of the memory and can be in different locations but, the priority is always to find the contiguous frames.

The pages of the process are carried to the main memory only when they are needed. Otherwise, they stay in the secondary storage.

The various operating systems define different frame sizes, and each frame must be equal. Considering that the pages are mapped to the frames, the page size must be the same as the frame size.

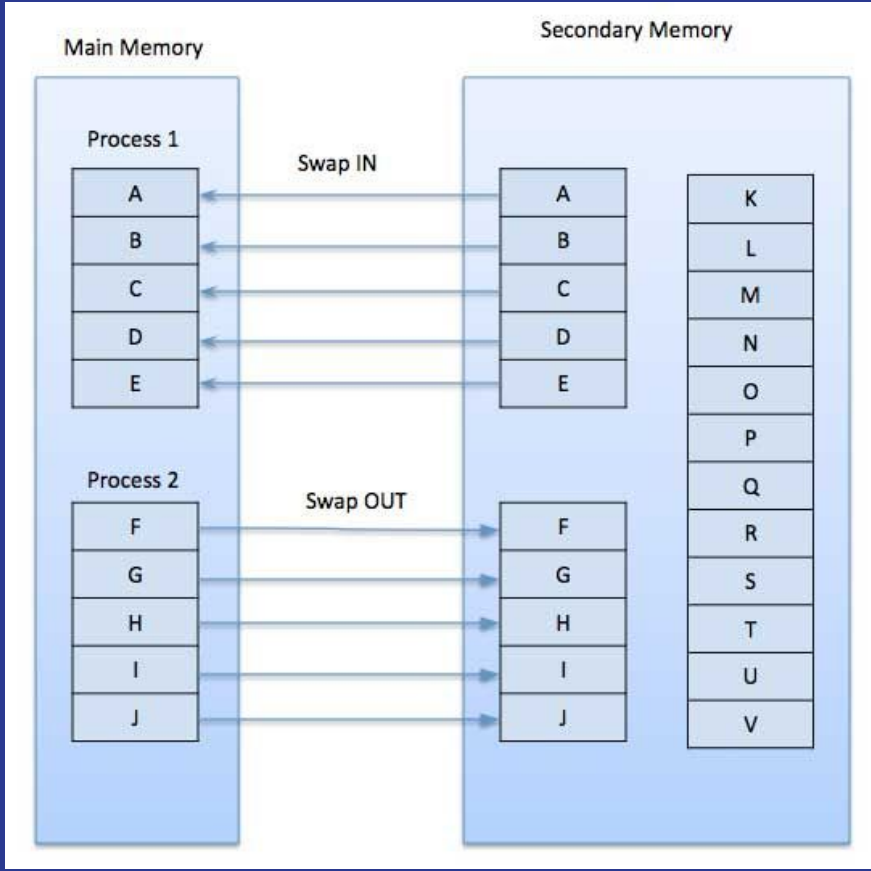
Virtual Memory

The Virtual Memory is a memory management that offers the user to have extra additional main memory where the second memory can be used as a part of main memory




USB Drive (E:) Properties
General
Tools
Hardware
Sharing
ReadyBoost
Customize
Speed up your system by utilizing the available space on this device.
Do not use this device.
Dedicate this device to ReadyBoost.
Use this device.
Space to reserve for system speed (may be less than actual free space due to file system limits):
4094
MB
While the device is being used for system speed the reserved space will not be available for file storage.
Windows recommends reserving 4094 MB for optimal performance.
Privacy statement
OK
Cancel
Apply

Demand Paging

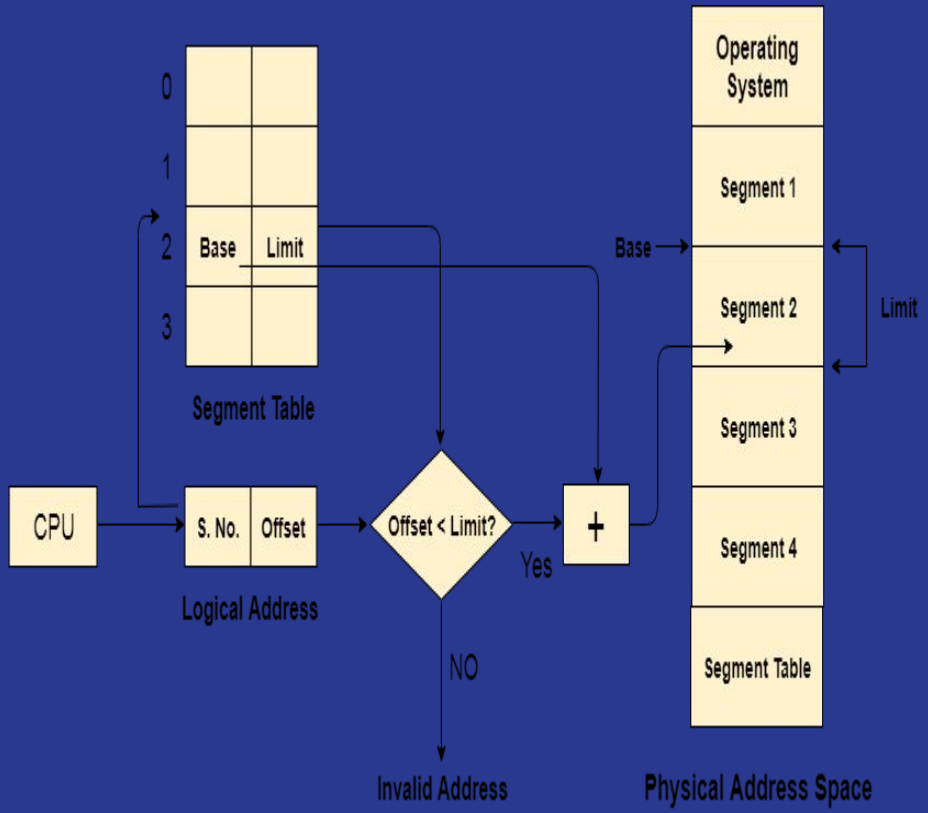


The demand paging system is like a paging system with swapping where the processes are in secondary memory and pages are loaded on demand and not in advance.

Segmentation

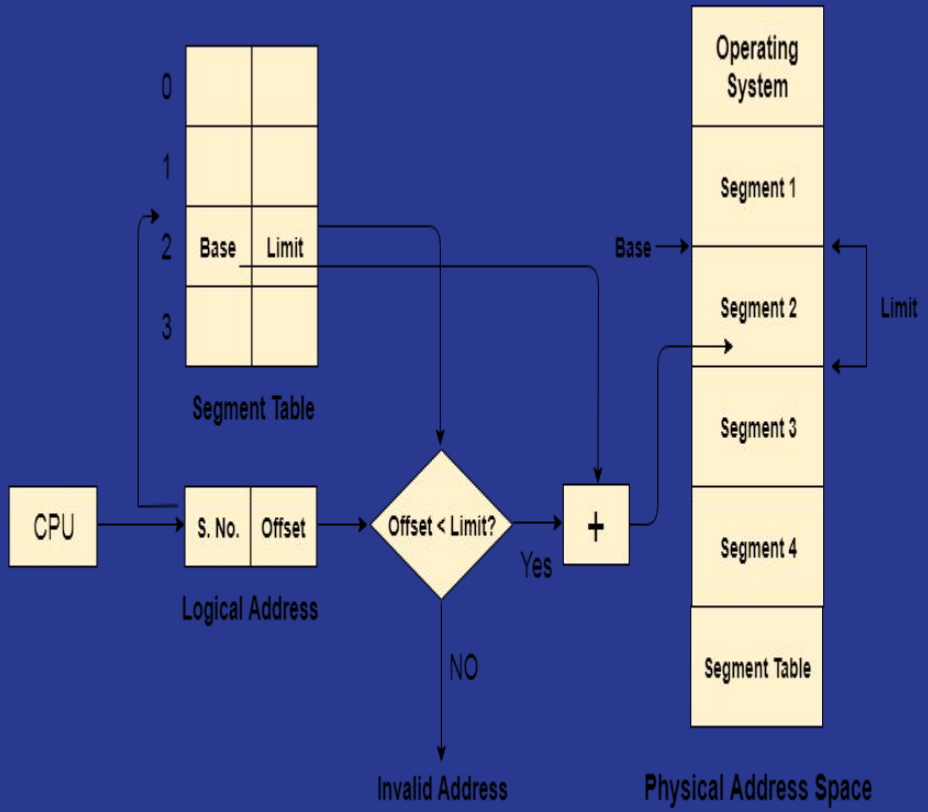
Generally supports the user view of memory in an operating system.

Similar to paging, it divides the memory but into variable-size blocks called **segments**.



Each of the segment details is put away in a table called the **segment table**.

This segment table is stored in one or many of the segments and mainly contains two pieces of information called the **base**, which is the base address of the segment, and the **limit**, which is the length of the segment.



In a logical address to a physical address by the segment table, the CPU generates a local address that contains the **segment number** and the **offset**.

The segment number is mapped to the segment table, while the limit of the corresponding segment is compared to the offset. On the condition that the offset is lesser than the limit, it is a valid address, or else it will be invalid.

If a valid address occurs, the base address of the segment is added to the offset that results in getting the physical address of the actual word in the main memory.