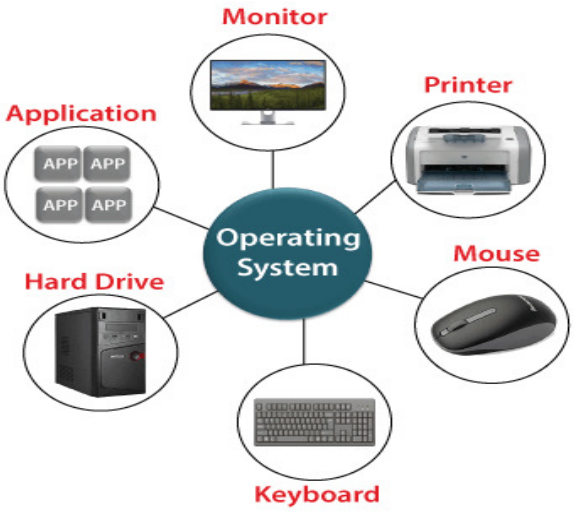
UNIT1 : Introduction to Operating System &Processor Management

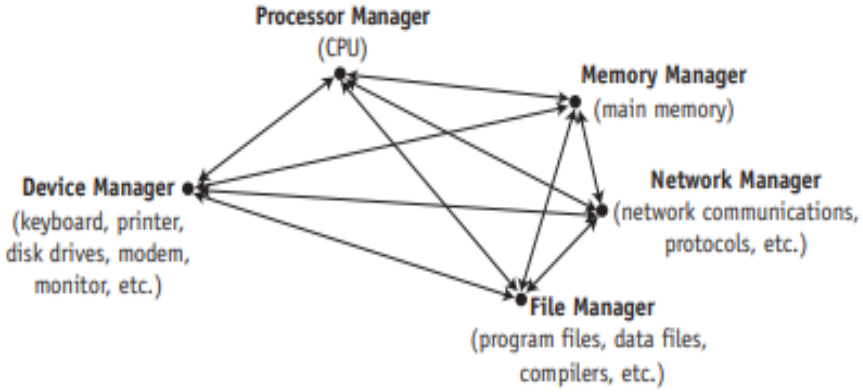
Que 1 : Introduction to Operating System

1. What Is an Operating System?

- A computer system consists of software (programs) and hardware (the physical machine and its electronic components).
- The operating system software is the chief piece of software, the portion of the computing system that manages all of the hardware and all of the other software.
- To be specific, it controls every file, every device, every section of main memory, and every nanosecond of processing time.
- It controls who can use the system and how. In short, it's the boss.
- Therefore, each time the user sends a command, the operating system must make sure that the command is executed; or, if it's not executed, it must arrange for the user to get a message explaining the error. Remember:
- This doesn't necessarily mean that the operating system executes the command or sends the error message—but it does control the parts of the system that do.



2. Operating System Software



Memory management:

- It manages both the primary and secondary memory such as RAM, ROM, hard disk, pen drive, etc.
- It checks and decides the allocations and deallocation of memory space to different processes.
- When a user interacts with a system, the CPU is supposed to read or write operations, in this case, OS decides the amount of memory to be allocated for loading the program instructions and data into RAM.
- After this program is terminated, the memory area is again free and is ready to be allocated to other programs by the OS.
-

processor management

- It facilitates processor management, where it decides the order for the processes to access the processor as well as decides the processing time to be allocated for each process. Besides this, it monitors the status of processes, frees the processor when a process is executed then allocates it to a new process.
-

Device Management

- The operating system also contains drivers to manage devices. A driver is a type of translation software that allows the operating system to communicate with devices, and there are different drivers for different devices as each device speaks a different language.

File Management

- The File Manager (the subject of Chapter 8) keeps track of every file in the system, including data files, program files, compilers, and applications. By using predetermined access policies, it enforces restrictions on who has access to which files. The File Manager also controls what users are allowed to do with files once they access them. For example, a user might have read-only access, read-and-write access, or the authority to create and delete files. Managing access control is a key part of file management. Finally, the File Manager allocates the necessary resources and later deallocates them.

Network Management

- Operating systems with Internet or networking capability have a fifth essential manager called the Network Manager (the subject of Chapters 9–10) that provides a convenient way for users to share resources while controlling users' access to them. These resources include hardware (such as CPUs, memory areas, printers, tape drives, modems, and disk drives) and software (such as compilers, application programs, and data files).

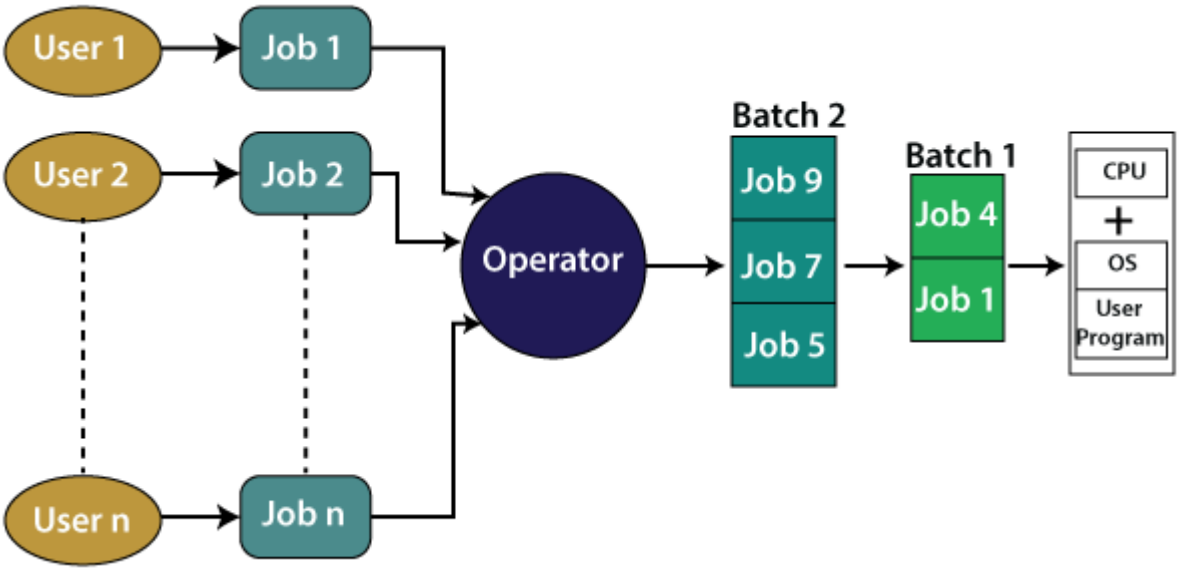
User Interface

- The user interface is the portion of the operating system that users interact with directly. In the old days, the user interface consisted of commands typed on a keyboard and displayed on a monitor. Now most systems allow users to choose a menu option from a list. The user interface, desktops, and formats vary widely from one operating system to another.

3 .Types of Operating System

Types of Operating System:

1) Batch Processing Operating System:



- The interaction between a user and the computer does not occur in this system.
- The user is required to prepare jobs on punch cards in the form of batches and submit them to the computer operator.
- The computer operator sorts the jobs or programs and keeps similar programs or jobs in the same batch and run as a group to speed up processing.
- It is designed to execute one job at a time. Jobs are processed on a first-come, first-serve basis, i.e., in the order of their submission without any human interference.

For example, the credit card bill generated by banks is an example of batch processing. A separate bill is not generated for each credit card purchase, rather a single bill that includes all purchases in a month is generated through batch processing. The bill details are collected and held as a batch, and then it is processed to generate the bill at the end of the billing cycle. Similarly, in a payroll system,

the salaries of employees of the company are calculated and generated through the batch processing system at the end of each month.

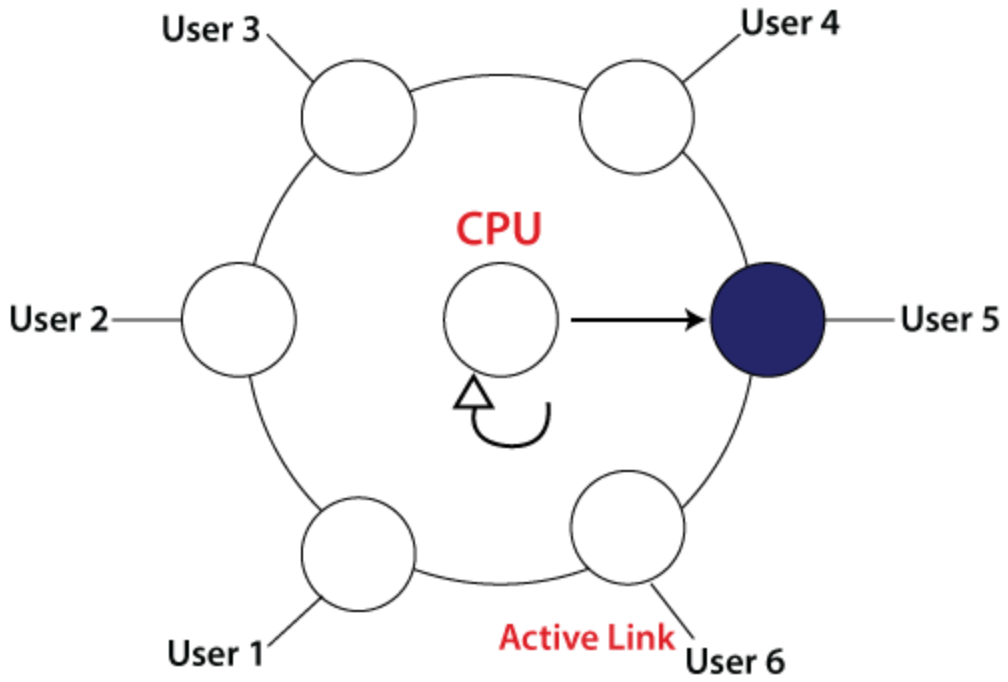
Advantages of Batch processing operating system:

- Repeated jobs can be completed easily without any human involvement
- Hardware or system support is not required to input data in batch systems
- It can work offline, so it causes less stress on the processor as it knows which task to process next and how long the task will last.
- It can be shared among multiple users.
- You can set the timing of batch jobs so that when the computer is not busy, it can start processing the batch jobs such as at night or any other free time.

Disadvantages of batch processing operating systems:

- You need to train the computer operators for using the batch system.
- It is not easy to debug this system.
- If any error occurs in one job, the other jobs may have to wait for an uncertain time.

2) Time Sharing Operating System:



- As the name suggests, it enables multiple users located at different terminals to use a computer system and to share the processor's time simultaneously.
- In other words, each task gets time to get executed, and thus all tasks are executed smoothly.
- Each user gets the processor's time as they get while using a single system.
- The duration of time allocated to a task is called quantum or time slice; when this duration is over, OS starts the next task.

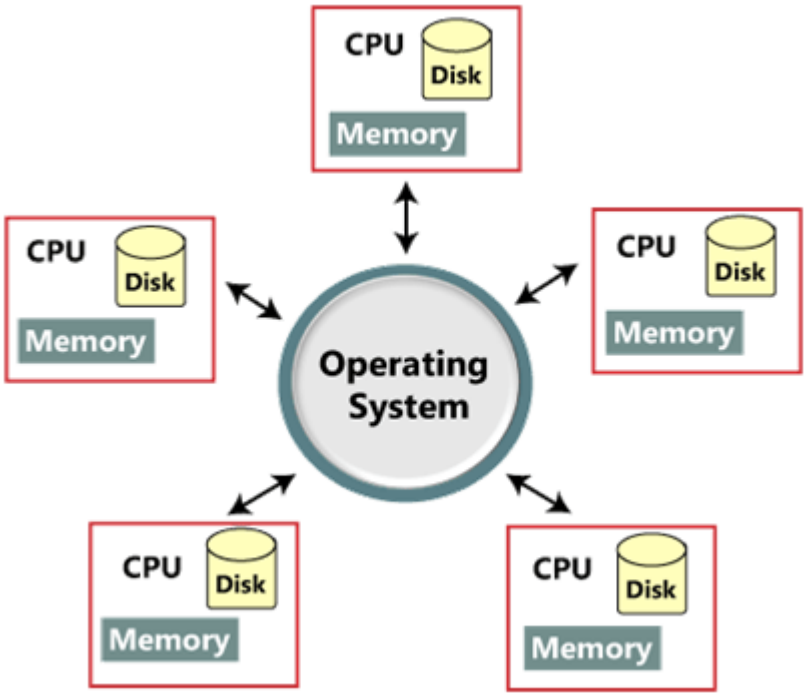
Advantages of time sharing operating system:

- It reduces CPU idle time and thus makes it more productive.
- Each process gets the chance to use the CPU.
- It allowed different applications run simultaneously.

Disadvantages of time sharing operating system:

- It requires a special operating system as it consumes more resources.
- Switching between tasks may hang up the system as it serves lots of users and runs lots of applications at the same time, so it requires hardware with high specifications.
- It is less reliable.

3) Distributed Operating System:



- It uses or runs on multiple independent processors (CPUs) to serve multiple users and multiple real-time applications.
- The communication between processors is established through many communication lines such as telephone lines and high-speed buses.
- The processors may differ from each other in terms of size and function.
- The availability of powerful microprocessor and advanced communication technology have made it possible to design, develop, and use the distributed operating system.
- in addition it is an extension of a network operating system that supports a high level of communication and integration of machines on the network.

Advantages of distributed operating system:

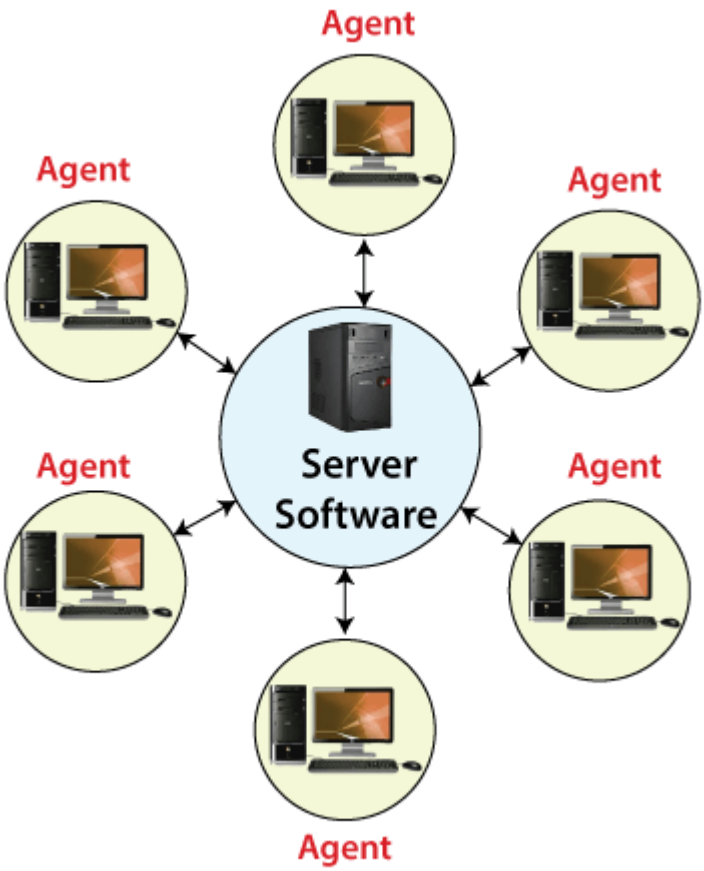
- Its performance is higher than a single system as resources are being shared.

- If one system stops working, malfunctions, or breaks down, other nodes are not affected.
- Additional resources can be added easily.
- Shared access to resources like printer can be established.
- Delay in processing is reduced to a greater extent.
- Data sharing or exchange speed is high, owing to the use of electronic mail.

Disadvantages of distributed operating system:

- Security issue may arise due to sharing of resources
- Few messages may be lost in the system
- Higher bandwidth is required in case of handling a large amount of data
- Overloading issue may arise
- The performance may be low
- The languages which are used to set up a distributed system are not well defined yet
- They are very costly, so they are not easily available.

4)Network Operating System:



- As the name suggests, this OS connects computers and devices to a local area network and manages network resources.

- The software in a NOS enables the devices of the network to share resources and communicate with each other.
- It runs on a server and allows shared access to printers, files, applications, files, and other networking resources and functions over a LAN.
- Besides this, all users in the network are aware of each other's underlying configuration and individual connections. Examples: Ms Windows Server 2003 and 2008, Linux, UNIX, Novell NetWare, Mac OS X, etc.

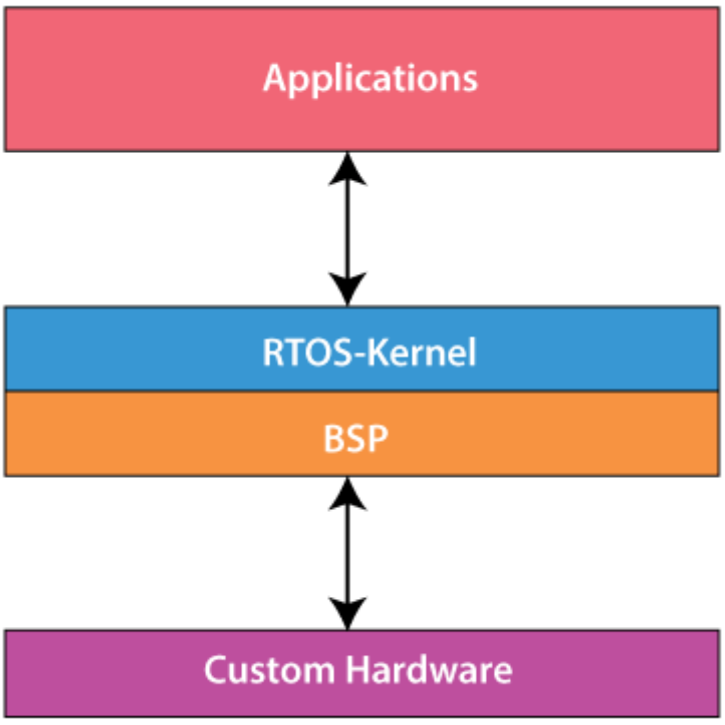
Advantages of network operating system:

- The servers are centralized that can be accessed remotely from distant locations and different systems.
- It is easy to integrate advanced and recent technologies and hardware in this system.

Disadvantages of network operating system:

- The servers used in the system may be expensive.
- The system depends on the central location and requires regular monitoring and maintenance.

5) Real-Time Operating System:



(Broad support package)

- It is developed for real-time applications where data should be processed in a fixed, small duration of time.
- It is used in an environment where multiple processes are supposed to be accepted and processed in a short time.

- RTOS requires quick input and immediate response, e.g., in a petroleum refinery, if the temperate gets too high and crosses the threshold value, there should be an immediate response to this situation to avoid the explosion. Similarly, this system is used to control scientific instruments, missile launch systems, traffic lights control systems, air traffic control systems, etc.

This system is further divided into two types based on the time constraints:

Hard Real-Time Systems:

These are used for the applications where timing is critical or response time is a major factor; even a delay of a fraction of the second can result in a disaster. For example, airbags and automatic parachutes that open instantly in case of an accident. Besides this, these systems lack virtual memory.

Soft Real-Time Systems:

These are used for application where timing or response time is less critical. Here, the failure to meet the deadline may result in a degraded performance instead of a disaster. For example, video observation (cctv), video player, virtual reality, etc. Here, the deadlines are not critical for every task every time.

Advantages of real-time operating system:

- The output is more and quick owing to the maximum utilization of devices and system
- Task shifting is very quick, e.g., 3 microseconds, due to which it seems that several tasks are executed simultaneously
- Gives more importance to the currently running applications than the queued application
- It can be used in embedded systems like in transport and others.
- It is free of errors.
- Memory is allocated appropriately.

Disadvantages of real-time operating system:

- A fewer number of tasks can run simultaneously to avoid errors.
- It is not easy for a designer to write complex and difficult algorithms or proficient programs required to get the desired output.
- Specific drivers and interrupt signals are required to respond to interrupts quickly.
- It may be very expensive due to the involvement of the resources required to work.

Que 2 : Memory Management: Early System

1. Single User Contiguous Scheme

- In this allocation scheme,The entire program or job is first loaded into memory contiguously as much as possible before the execution.

- If the size of program is larger than the size of the memory available, the execution will not begin.
- To execute the program either the size of the memory available needs to be increased or the size of the program needs to be decreased to fit in the available memory.

Operating System	10K
Program 1 (40K)	50K
Unused Main Memory	

Algorithm to Load a Job in a Single-User System

- Store first memory location of program into base register
- Set program counter equal to address of first memory location
- Read first instruction of program
- Increment program counter by number of bytes in instruction
- 5 Has the last instruction been reached? if yes, then stop loading program if no, then continue with step 6
- Is program counter greater than memory size? if yes, then stop loading program if no, then continue with step 7
- 7 Load instruction in memory 8 Read next instruction of program 9 Go to step 4

Advantages:

- The jobs are allocated sequentially therefore the requirement of operating system's work is less.
- The Operating system just needs to check if the incoming job fits into the available memory space else reject the job and proceed to the next incoming job.

Disadvantages:

- Only one job or program is loaded in to memory at any point of time.
- Memory sharing is not possible among jobs
- Even though there is a space to accommodate the waiting jobs, it is not possible accommodate more than one job.
- Does not support multi-programming

2. Fixed Partitions

- The earliest and one of the simplest technique which can be used to load more than one processes into the main memory is Fixed partitioning or Contiguous memory allocation.
- In this technique, the main memory is divided into partitions of equal or different sizes.
- The operating system always resides in the first partition while the other partitions can be used to store user processes.
- The memory is assigned to the processes in contiguous way.

- In fixed partitioning,The partitions cannot overlap And A process must be contiguously present in a partition for the execution.

There are various cons of using this technique.

1. Internal Fragmentation

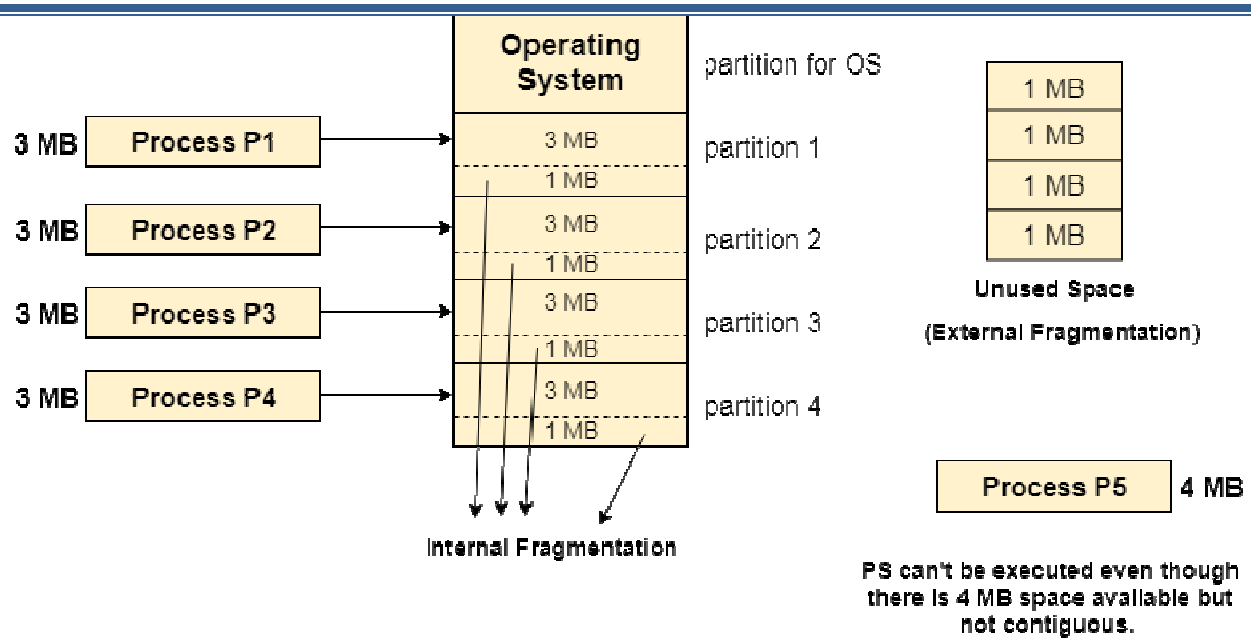
- If the size of the process is lesser then the total size of the partition then some size of the partition get wasted and remain unused.
- This is wastage of the memory and called internal fragmentation.
- As shown in the image below, the 4 MB partition is used to load only 3 MB process and the remaining 1 MB got wasted.

2. External Fragmentation

- The total unused space of various partitions cannot be used to load the processes even though there is space available but not in the contiguous form.
- As shown in the image below, the remaining 1 MB space of each partition cannot be used as a unit to store a 4 MB process.
- Despite of the fact that the sufficient space is available to load the process, process will not be loaded.

3. Limitation on the size of the process

- If the process size is larger than the size of maximum sized partition then that process cannot be loaded into the memory.
- Therefore, a limitation can be imposed on the process size that is it cannot be larger than the size of the largest partition.

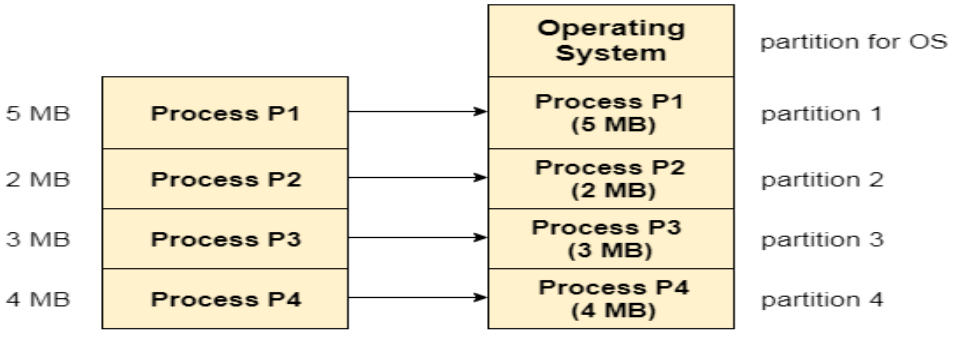


Fixed Partitioning

(Contiguous memory allocation)

3. Dynamic Partitions

- Dynamic partitioning tries to overcome the problems caused by fixed partitioning.
- In this technique, the partition size is not declared initially.
- It is declared at the time of process loading.
- The first partition is reserved for the operating system.
- The remaining space is divided into parts.
- The size of each partition will be equal to the size of the process.
- The partition size varies according to the need of the process so that the internal fragmentation can be avoided.



Dynamic Partitioning

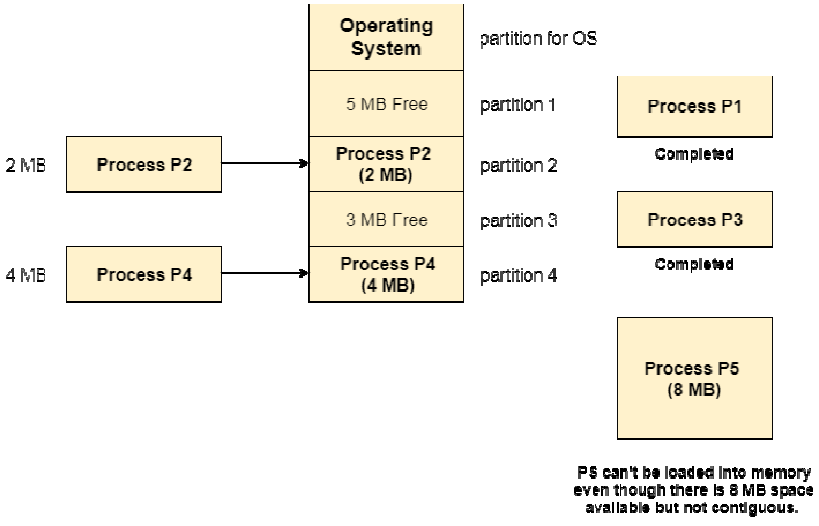
(Process Size = Partition Size)

Advantages

- No Internal Fragmentation
- No Limitation on the size of the process
- Due to the absence of internal fragmentation, there will not be any unused space in the partition hence more processes can be loaded in the memory at the same time.

Disadvantages

- **External Fragmentation**
  - Absence of internal fragmentation doesn't mean that there will not be external fragmentation.
  - Let's consider three processes P1 (1 MB) and P2 (3 MB) and P3 (1 MB) are being loaded in the respective partitions of the main memory.
  - After some time P1 and P3 got completed and their assigned space is freed. Now there are two unused partitions (1 MB and 1 MB) available in the main memory but they cannot be used to load a 2 MB process in the memory since they are not contiguously located.
  - The rule says that the process must be contiguously present in the main memory to get executed. We need to change this rule to avoid external fragmentation.
- **Complex Memory Allocation**
  - In Fixed partitioning, the list of partitions is made once and will never change but in dynamic partitioning, the allocation and deallocation is very complex since the partition size will be varied every time when it is assigned to a new process.
  - OS has to keep track of all the partitions.
  - Due to the fact that the allocation and deallocation are done very frequently in dynamic memory allocation and the partition size will be changed at each time, it is going to be very difficult for OS to manage everything.



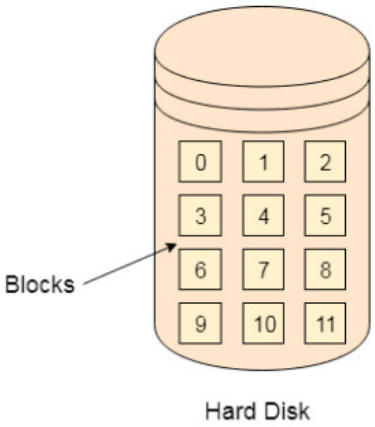
External Fragmentation in Dynamic Partitioning

4. Allocation and deallocation methods

- There are various methods which can be used to allocate disk space to the files. Selection of an appropriate allocation method will significantly affect the performance and efficiency of the system.
- Allocation method provides a way in which the disk will be utilized and the files will be accessed.
- There are following methods which can be used for allocation.

Contiguous Allocation.

- such a way that all the logical blocks of the file get the contiguous physical block in the hard disk then such allocation scheme is known as contiguous allocation.
- In the image shown below, there are three files in the directory.
- The starting block and the length of each file are mentioned in the table.
- We can check in the table that the contiguous blocks are assigned to each file as per its need.



File Name	Start	Length	Allocated Blocks
abc.text	0	3	0,1,2
video.mp4	4	2	4,5
jtp.docx	9	3	9,10,11

Directory

Contiguous Allocation

Advantages

1. It is simple to implement.
2. We will get Excellent read performance.
3. Supports Random Access into files.

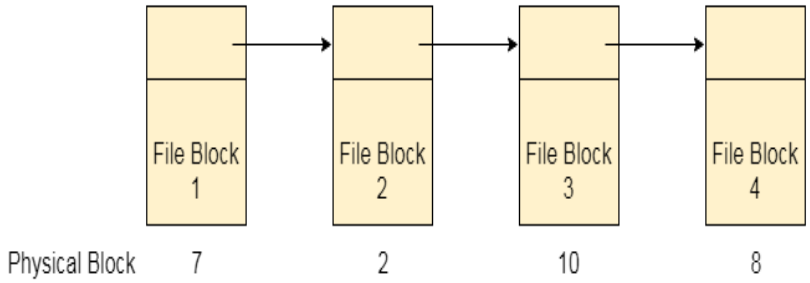
Disadvantages

1. The disk will become fragmented.
2. It may be difficult to have a file grow.

Linked Allocation

- Linked List allocation solves all problems of contiguous allocation.

- In linked list allocation, each file is considered as the linked list of disk blocks. However, the disks blocks allocated to a particular file need not to be contiguous on the disk.
- Each disk block allocated to a file contains a pointer which points to the next disk block allocated to the same file.



Linked List Allocation

Advantages

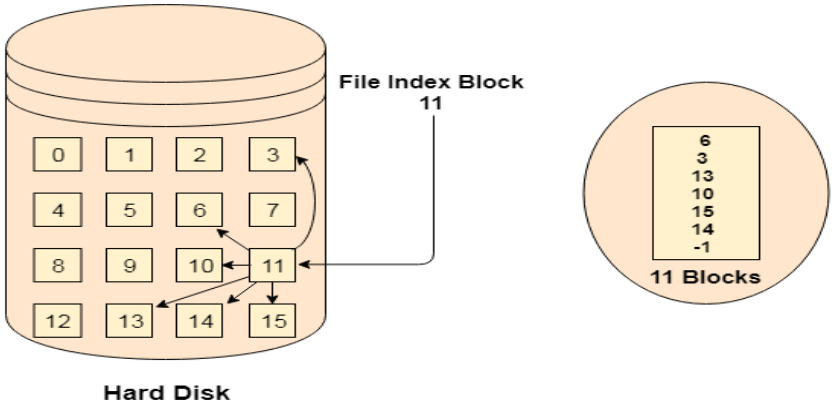
- There is no external fragmentation with linked allocation.
- No file can increse

Disadvantages

- Random Access is not provided.
- Pointers require some space in the disk blocks.
- Any of the pointers in the linked list must not be broken otherwise the file will get corrupted.
- Need to traverse each block.

Indexed Allocation

- Instead of maintaining a file allocation table of all the disk pointers, Indexed allocation scheme stores all the disk pointers in one of the blocks called as indexed block.
- Indexed block doesn't hold the file data, but it holds the pointers to all the disk blocks allocated to that particular file.
- Directory entry will only contain the index block address.



Advantages

- There is no external fragmentation
- Support direct accsess

DisAdvantages

- Pointer Overhead
- Multiple Index

Deallocation of Memory

- Deallocation of memory by the Operating System (OS) is a way to free the Random Access Memory (RAM) of finished processes and allocate new ones.
- the computer memory comes with a specific size.
- computer manages everything with the help of a robust memory management technique that handles all the allocations and deallocations of processes.
- A process has to be loaded into the RAM for its execution and remains in the RAM until its completion. Finished processes are deallocated or removed from the memory and new processes are allocated again.
- This is how the OS works with allocation and deallocation.
- In high-level, programming memory deallocation is done by garbage collection.

5. Relocatable Dynamic Partitions

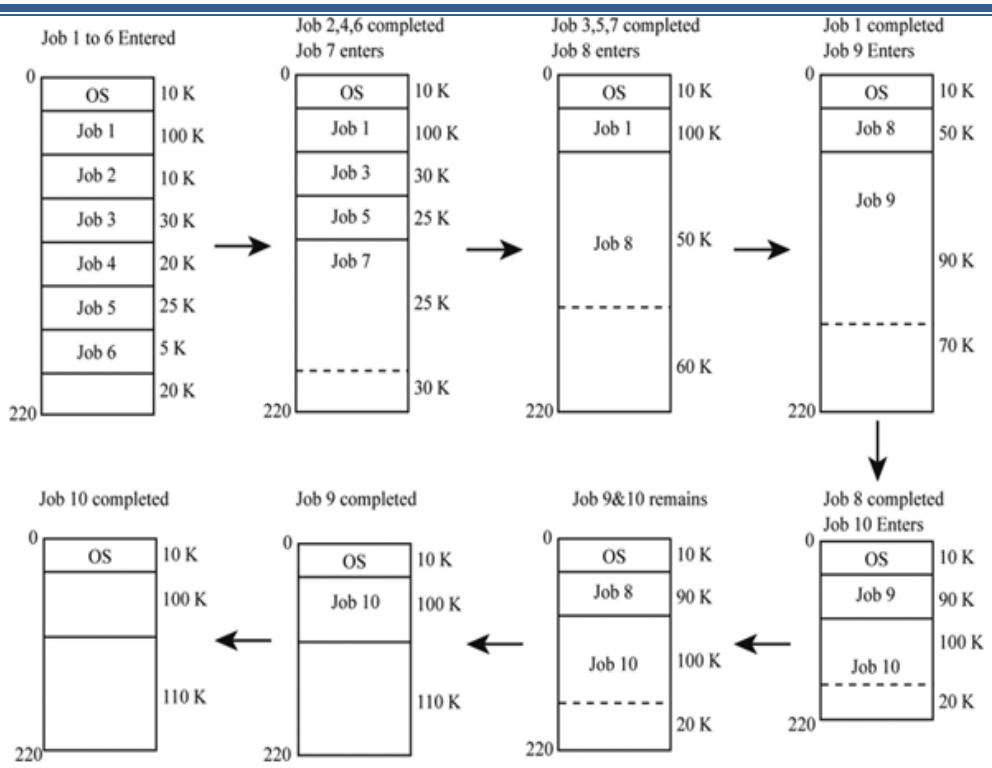
- In relocatable dynamic partition, memory manager in operating system relocates the program, that is all empty blocks are gathered to form one single block of large memory enough to accommodate some or all of the jobs waiting in the queue.

Example:

Job	Memory utilized	Turnaround time
Job1	100K	3
Job2	10K	1
Job3	30K	2
Job4	20K	1
Job5	25K	2
Job6	5K	1
Job7	25K	1
Job8	50K	2
Job9	90K	3
Job10	100K	3

- Let us assume the memory size is 220k with 10k allocated to operating system programs. The above table gives the data of number of jobs, memory utilized by individual job and their respective turnaround time. There are 10 jobs to be loaded into memory.
- Initially with 220k of memory, we can load job1 to job6 with free memory of 20k.
- At first turnaround time, job2 of 10k, job4 of 20k, and job6 of 5k are completed leaving job 1, job 3, and job 5 as it is with free memory of 55k, in other words job1, job3, and job5 are still processing. Job 7 of 25k now enters into the memory partition leaving only 30k free memory where neither of the remaining jobs (job 8, job 9, and job 10) can accommodate.
- At second turnaround time job3 of 30k, job5 of 25k, and job7 of 25k are completed leaving job 1 of 100k with free memory of 110k. Now job 8 of 50K is entered where 50k is allocated leaving 60K of free memory space where either job 9 or job 10 can accommodate.
- Note: for job3 and job5 turnaround time is finished and at the same time turnaround time 1 of job 7 is also finished.
- At third turnaround time, job1 of 100k is completed leaving free memory space of 160k.
- Job9 now can be allocated with 90k leaving remaining 70K of free memory where job10 cannot accommodate.
- Once job 8 finishes turnaround time of 2, it is completed leaving 120k of free memory space. Now job10 can be allocated with 100k of memory leaving 20K of free memory space.
- Process repeats until all the memory locations are freed leaving 10k allocated to operating system processes.

Diagrammatic representation of above example is shown below:



Que 3 : Memory Management: Virtual Memory

1. Virtual Memory

- Virtual Memory is a storage scheme that provides user an illusion of having a very big main memory.
- This is done by treating a part of secondary memory as the main memory.
- In this scheme, User can load the bigger size processes than the available main memory by having the illusion that the memory is available to load the process.
- Instead of loading one big process in the main memory, the Operating System loads the different parts of more than one process in the main memory.
- By doing this, the degree of multiprogramming will be increased and therefore, the CPU utilization will also be increased.

How Virtual Memory Works?

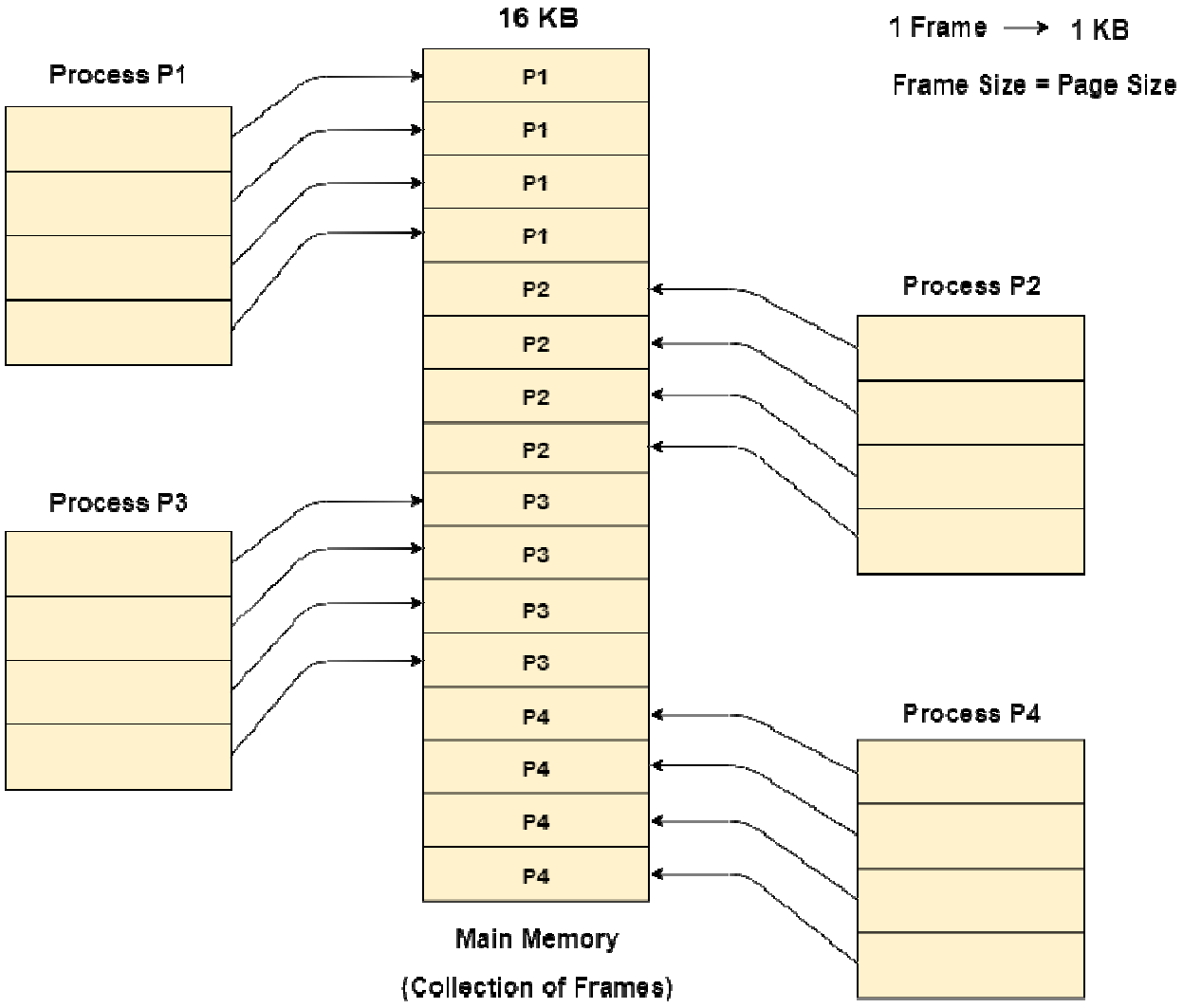
- Whenever some pages needs to be loaded in the main memory for the execution and the memory is not available for those many pages, then in that case, instead of stopping the pages from entering in the main memory, the OS search for the RAM area that are least used in the recent times or that are not referenced and copy that into the secondary memory to make the space for the new pages in the main memory.
- Since all this procedure happens automatically, therefore it makes the computer feel like it is having the unlimited RAM.

2. Paged Memory Allocation

- In Operating Systems, Paging is a storage mechanism used to retrieve processes from the secondary storage into the main memory in the form of pages.
- The main idea behind the paging is to divide each process in the form of pages.
- The main memory will also be divided in the form of frames.
- One page of the process is to be stored in one of the frames of the memory.
- The pages can be stored at the different locations of the memory but the priority is always to find the contiguous frames .
- Pages of the process are brought into the main memory only when they are required otherwise they reside in the secondary storage.
- Different operating system defines different frame sizes.
- The sizes of each frame must be equal.
- Considering the fact that the pages are mapped to the frames in Paging, page size needs to be as same as frame size.

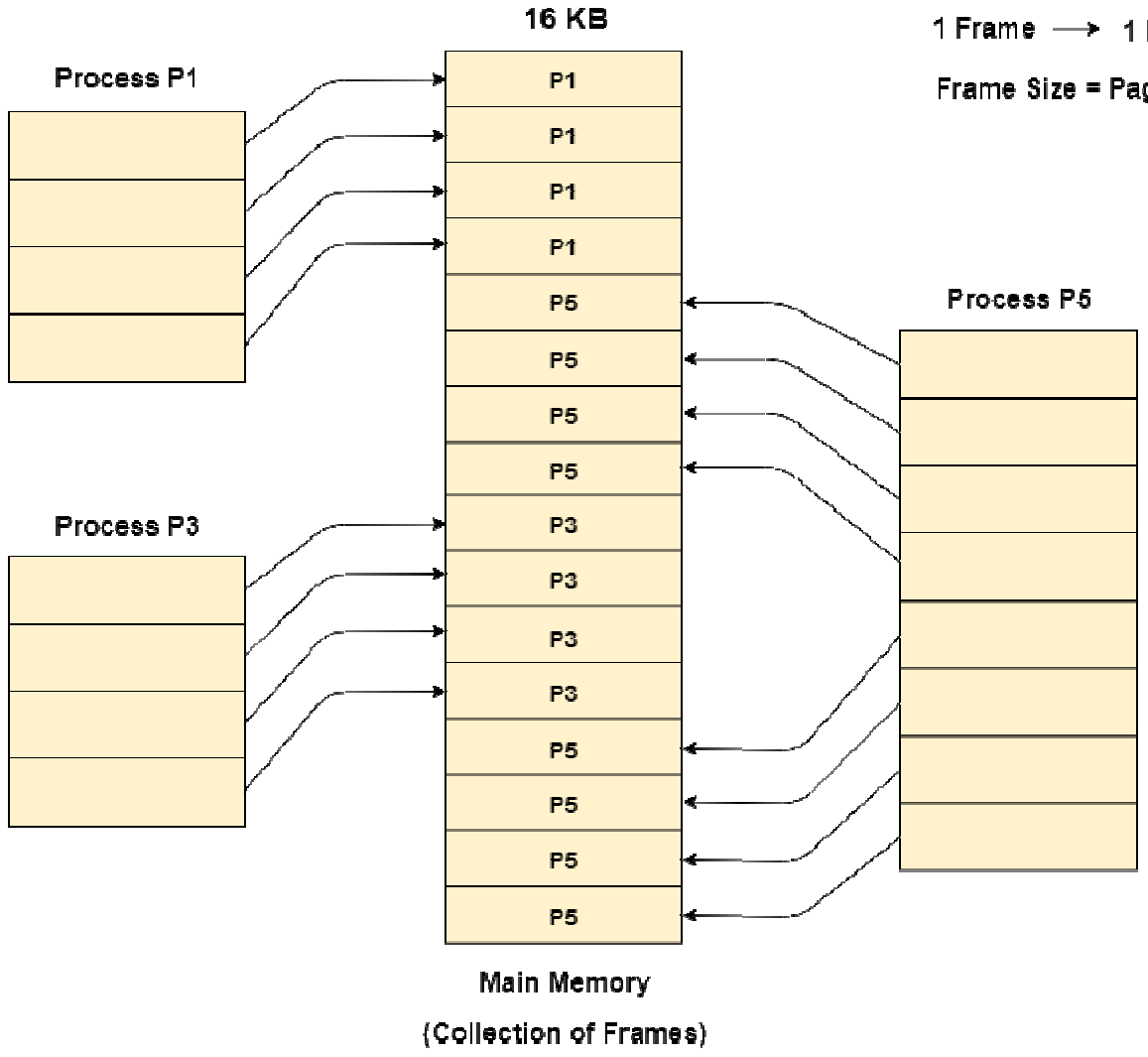
Example

- Let us consider the main memory size 16 Kb and Frame size is 1 KB therefore the main memory will be divided into the collection of 16 frames of 1 KB each.
- There are 4 processes in the system that is P1, P2, P3 and P4 of 4 KB each.
- Each process is divided into pages of 1 KB each so that one page can be stored in one frame.
- Initially, all the frames are empty therefore pages of the processes will get stored in the contiguous way.
- Frames, pages and the mapping between the two is shown in the image below.



Paging

- Let us consider that, P2 and P4 are moved to waiting state after some time.
- Now, 8 frames become empty and therefore other pages can be loaded in that empty place. The process P5 of size 8 KB (8 pages) is waiting inside the ready queue.
- Given the fact that, we have 8 non contiguous frames available in the memory and paging provides the flexibility of storing the process at the different places. Therefore, we can load the pages of process P5 in the place of P2 and P4.



Paging

Memory Management Unit

- The purpose of Memory Management Unit (MMU) is to convert the logical address (**virtual address and can be viewed by the user)**into the physical address(memory address).
- The logical address is the address generated by the CPU for every page while the physical address is the actual address of the frame where each page will be stored.
- When a page is to be accessed by the CPU by using the logical address, the operating system needs to obtain the physical address to access that page physically.

  The logical address has two parts.

  Page Number

  Offset(Last bit of virtual address)

- Memory management unit of OS needs to convert the page number to the frame number.

Example

- Considering the above image, let's say that the CPU demands 10th word of 4th page of process P3.
- Since the page number 4 of process P1 gets stored at frame number 9 therefore the 10th word of 9th frame will be returned as the physical address.

3.Demand Paging

- The process of loading the page into memory on demand (whenever page fault occurs) is known as demand paging.
- Every process in the virtual memory contains lots of pages and in some cases, it might not be efficient to swap all the pages for the process at once. Because it might be possible that the program may need only a certain page for the application to run.
- Let us take an example here, suppose there is a 500 MB application and it may need as little as 100MB pages to be swapped, so in this case, there is no need to swap all pages at once.
- The demand paging system is somehow similar to the paging system with swapping where processes mainly reside in the main memory(usually in the hard disk).
- Thus demand paging is the process that solves the above problem only by swapping the pages on Demand.
- This is also known as lazy swapper(It never swaps the page into the memory unless it is needed).
- Swapper that deals with the individual pages of a process are referred to as Pager.
- Demand Paging is a technique in which a page is usually brought into the main memory only when it is needed or demanded by the CPU.
- Initially, only those pages are loaded that are required by the process immediately. Those pages that are never accessed are thus never loaded into the physical memory.

Valid-Invalid Bit

- Some form of hardware support is used to differentiate between the pages that are in the memory and the pages that are on the disk.
- Thus for this purpose Valid-Invalid scheme is used:
- With each page table entry, a valid-invalid bit is associated(where **1** indicates **in the memory** and **0** indicates **not in the memory**)
- Initially, the valid-invalid bit is set to 0 for all table entries.
- If the bit is set to "**valid**", then the associated page is **both legal and is in memory.**
- If the bit is set to "**invalid**" then it indicates that the **page is either not valid** or the page is valid **but is currently not on the disk**
- For the **pages** that are **brought into the memory**, **the page table is set as usual.**

- But for the **pages** that are **not currently in the memory**, the **page table** is either **simply marked as invalid** or it contains the **address of the page on the disk**.
- During the translation of address, if the valid-invalid bit in the page table entry is 0 then it leads to **page fault.**
-

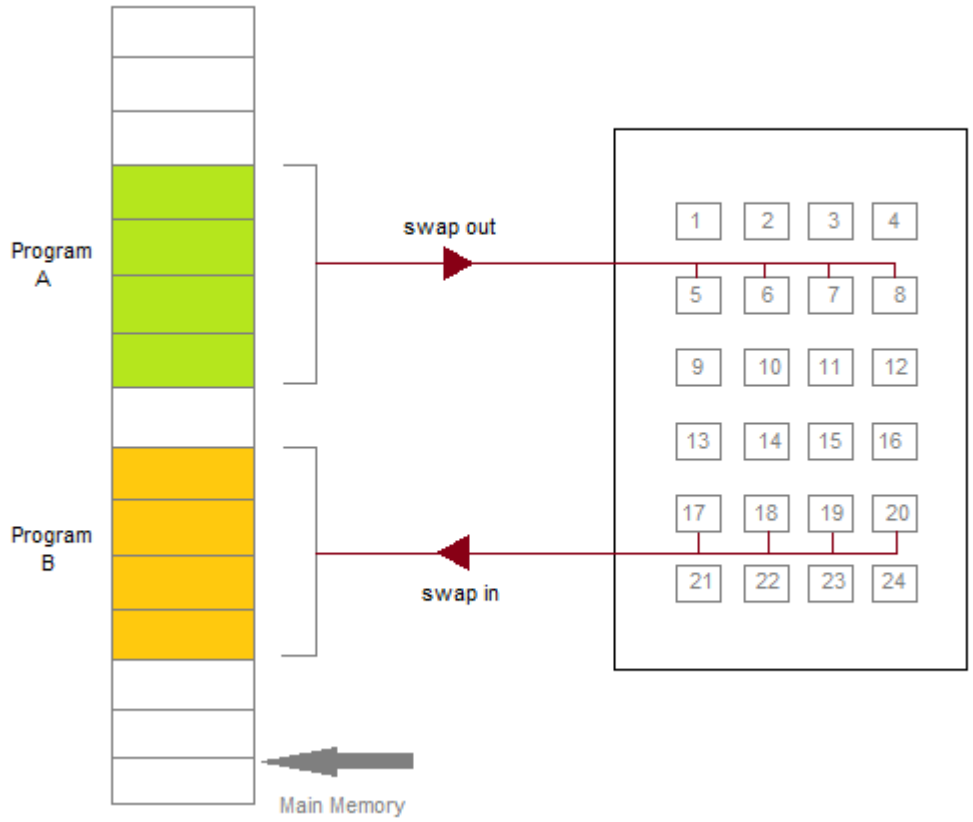


Figure: Transfer of a Paged Memory to the contiguous disk space.

- Whenever a page is needed? make a reference to it;
- If the reference is **invalid** then abort it.
- If the page is Not-in-memory then bring it to memory.

How Demand Paging Works?

First of all the components that are involved in the Demand paging process are as follows:

Main Memory
CPU
Secondary Memory
Interrupt
Physical Address space
Logical Address space
Operating System
Page Table

- If a page is not available in the main memory in its active state; then a request may be made to the CPU for that page.
- Thus for this purpose, it has to generate an interrupt.
- After that, the Operating system moves the process to the blocked state as an interrupt has occurred.
- Then after this, the Operating system searches the given page in the Logical address space.
- And Finally with the help of the page replacement algorithms, replacements are made in the physical address space.
- Page tables are updated simultaneously.
- After that, the CPU is informed about that update and then asked to go ahead with the execution and the process gets back into its ready state.
- When the process requires any of the pages that are not loaded into the memory, a page fault trap is triggered and the following steps are followed,
- The memory address which is requested by the process is first checked, to verify the request made by the process.
- If it is found to be invalid, the process is terminated.
- In case the request by the process is valid, a free frame is located, possibly from a free-frame list, where the required page will be moved.
- A new operation is scheduled to move the necessary page from the disk to the specified memory location. (This will usually block the process on an I/O wait, allowing some other process to use the CPU in the meantime.)
- When the I/O operation is complete, the process's page table is updated with the new frame number, and the invalid bit is changed to valid.
- The instruction that caused the page fault must now be restarted from the beginning.

Advantages of Demand Paging

- With the help of Demand Paging, memory is utilized efficiently.
- Demand paging avoids External Fragmentation.
- Less Input/Output is needed for Demand Paging.
- This process is not constrained by the size of physical memory.
- With Demand Paging it becomes easier to share the pages.
- With this technique, portions of the process that are never called are never loaded.
- No compaction is required in demand Paging.

Disadvantages of Demand paging

- There is an increase in overheads due to interrupts and page tables.
- Memory access time in demand paging is longer.

4. Page Replacement Algorithms

- The page replacement algorithm decides which memory page is to be replaced.
- The process of replacement is sometimes called swap out or write to disk.
- Page replacement is done when the requested page is not found in the main memory (page fault).

What is a Page Fault?

- If the referred page is not present in the main memory then there will be a miss and the concept is called Page miss or page fault.
- The CPU has to access the missed page from the secondary memory. If the number of page fault is very high then the effective access time of the system will become very high.

Page Replacement Algorithms :

1. FIFO Page Replacement Algorithm

- It is a very simple way of Page replacement and is referred to as First in First Out. This algorithm mainly replaces the oldest page that has been present in the main memory for the longest time.
- This algorithm is implemented by keeping the track of all the pages in the queue.
- As new pages are requested and are swapped in, they are added to the tail of a queue and the page which is at the head becomes the victim.
- This is not an effective way of page replacement but it can be used for small systems.

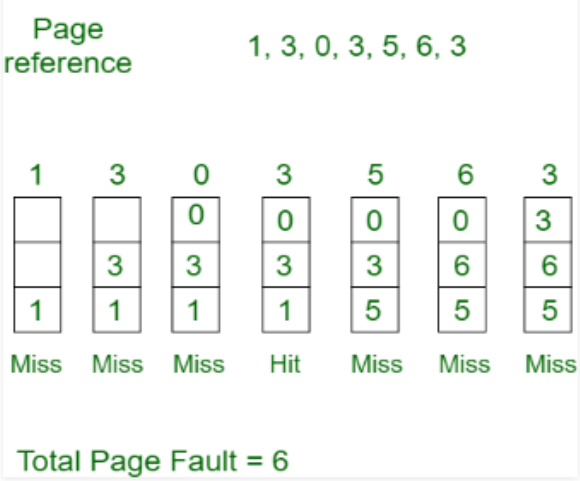
Advantages

- This algorithm is simple and easy to use.
- FIFO does not cause more overhead.

Disadvantages

- This algorithm does not make the use of the frequency of last used time rather it just replaces the Oldest Page.
- There is an increase in page faults as page frames increases.
- The performance of this algorithm is the worst.

Example-1Consider page reference string 1, 3, 0, 3, 5, 6 with 3 page frames.Find number of page faults.



- Initially all slots are empty, so when 1, 3, 0 came they are allocated to the empty slots —> 3 Page Faults.
- when 3 comes, it is already in memory so —> 0 Page Faults.
- Then 5 comes, it is not available in memory so it replaces the oldest page slot i.e 1. —>1 Page Fault.
- 6 comes, it is also not available in memory so it replaces the oldest page slot i.e 3 —>1 Page Fault.
- Finally when 3 come it is not avilable so it replaces 0 1 page fault

Least Recently Used

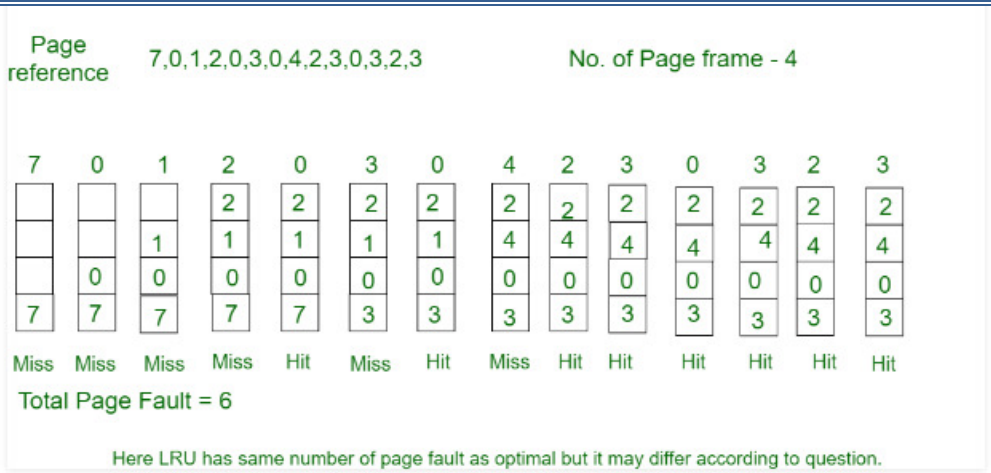
- This algorithm stands for "Least recent used" and this algorithm helps the Operating system to search those pages that are used over a short duration of time frame.
- The page that has not been used for the longest time in the main memory will be selected for replacement.
- This algorithm is easy to implement.
- This algorithm makes use of the counter along with the even-page.

Advantages of LRU

- It is an efficient technique.
- With this algorithm, it becomes easy to identify the faulty pages that are not needed for a long time.
- It helps in Full analysis.

Disadvantages of LRU

- It is expensive and has more complexity.
- There is a need for an additional data structure.



- Initially all slots are empty, so when 7 0 1 2 are allocated to the empty slots —> 4 Page fault 0 is already their so —> 0 Page fault.
- when 3 came it will take the place of 7 because it is least recently used —>1 Page fault 0 is already in memory so —> 0 Page fault.
- 4 will takes place of 1 —> 1 Page Fault
- Now for the further page reference string —> 0 Page fault because they are already available in the memory.

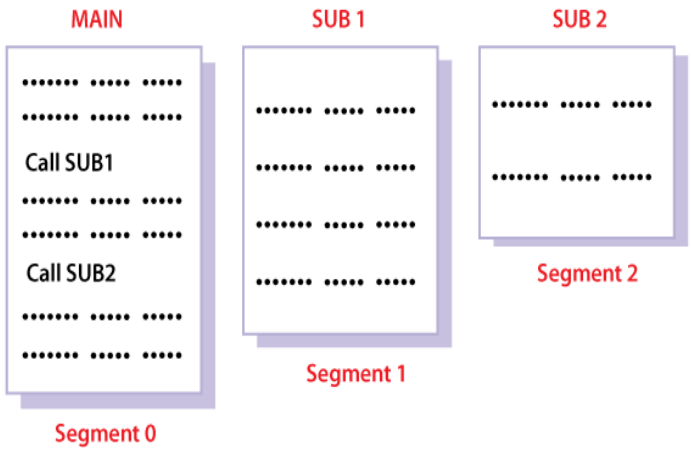
5 Segmented Memory allocation

- In Operating Systems, Segmentation is a memory management technique in which the memory is divided into the variable size parts.
- Each part is known as a segment which can be allocated to a process.
- The details about each segment are stored in a table called a segment table.
- Segment table is stored in one (or many) of the segments.
- Segment table contains mainly two information about segment:

1. Base: It is the base address of the segment
2. Limit: It is the length of the segment.

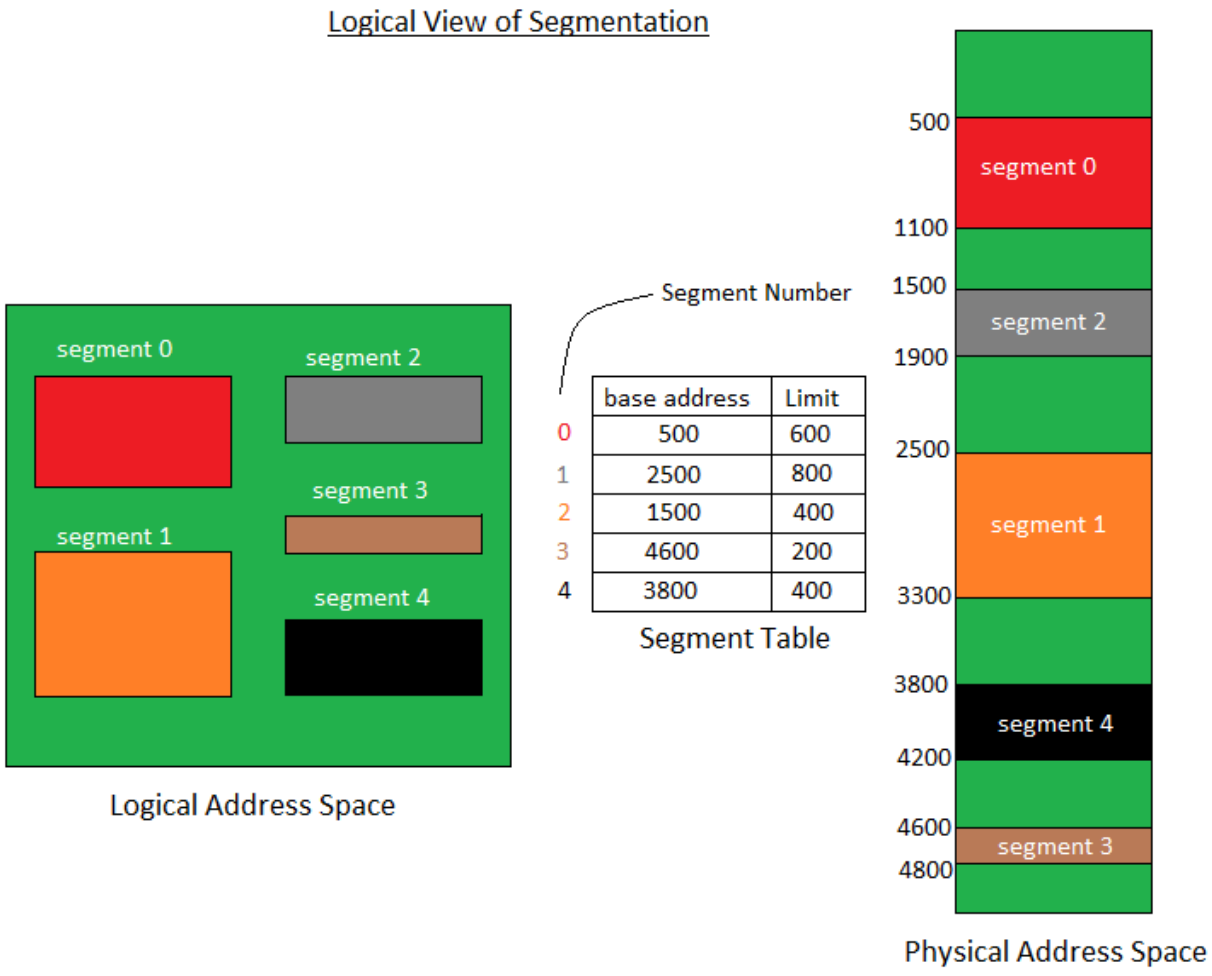
Why Segmentation is required?

- Till now, we were using Paging as our main memory management technique.
- Paging is more close to the Operating system rather than the User.
- It divides all the processes into the form of pages in spite of the fact that a process can have some relative parts of functions which need to be loaded in the same page.
- Operating system doesn't care about the User's view of the process.
- It may divide the same function into different pages and those pages may or may not be loaded at the same time into the memory.
- It decreases the efficiency of the system.
- It is better to have segmentation which divides the process into the segments.
- Each segment contains the same type of functions such as the main function can be included in one segment and the library functions can be included in the other segment.



Translation of Logical address into physical address by segment table

1. The operating system also generates a segment map table for each program.



Advantages of Segmentation

1. No internal fragmentation
2. Average Segment Size is larger than the actual page size.
3. Less overhead(Processing time)
4. It is easier to relocate segments than entire address space.
5. The segment table is of lesser size as compared to the page table in paging.

Disadvantages

1. It can have external fragmentation.
2. it is difficult to allocate contiguous memory to variable sized partition.
3. Costly memory management algorithms.

Segmented/Demand Paged Memory allocation

S.No. Demand Paging Segmentation

S.No.	Demand Paging	Segmentation
1.	In demand paging, the pages are of equal size.	While in segmentation, segments can be of different size.
3.	It does not allows sharing of the pages.	While segments can be shared in segmentation.
4.	In demand paging, on demand pages are loaded in the memory.	In segmentation, during compilation segments are allocated to the program.
5.	Page table in demand paging manages record of pages in memory.	Segment table in segmentation demonstrates every segment address in the memory.
6.	It provides large virtual memory and have more efficient use of memory.	It provides virtual memory and maximum size of segment is defined by the size of memory.