**What is a Process?**

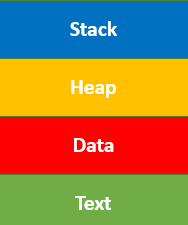
**Process** is the execution of a program that performs the actions specified in that program. It can be defined as an execution unit where a program runs. The OS helps you to create, schedule, and terminates the processes which is used by CPU. A process created by the main process is called a child process.

Process operations can be easily controlled with the help of PCB(Process Control Block).

## What is Process Management?

Process management involves various tasks like creation, scheduling, termination of processes, and a dead lock. Process is a program that is under execution, which is an important part of modern-day operating systems. The OS must allocate resources that enable processes to share and exchange information. It also protects the resources of each process from other methods and allows synchronization among processes.

It is the job of OS to manage all the running processes of the system. It handles operations by performing tasks like process scheduling and such as resource allocation.



* **Stack:** The Stack stores temporary data like function parameters, returns addresses, and local variables.
* **Heap** Allocates memory, which may be processed during its run time.
* **Data:** It contains the variable.
* **Text:** Text Section includes the current activity, which is represented by the value of the Program Counter.

## Process Control Blocks

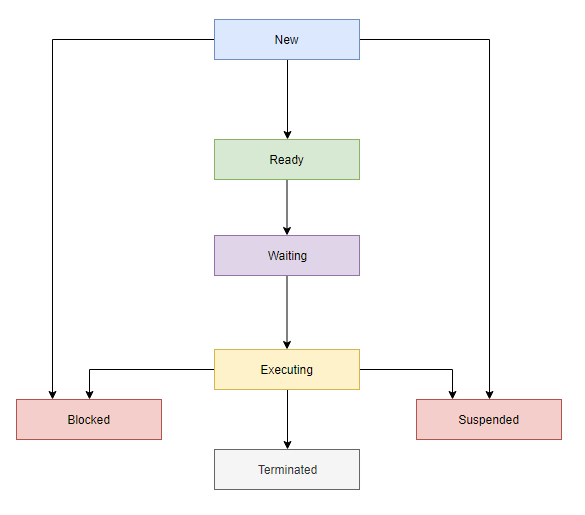
The PCB is a full form of Process Control Block. It is a data structure that is maintained by the Operating System for every process. The PCB should be identified by an integer Process ID (PID). It helps you to store all the information required to keep track of all the running processes. It is also accountable for storing the contents of processor registers. These are saved when the process moves from the running state and then returns back to it. The information is quickly updated in the PCB by the OS as soon as the process makes the state transition.

## Process States

A process state is a condition of the process at a specific instant of time. It also defines the current position of the process.

There are mainly seven stages of a process which are:

* New: The new process is created when a specific program calls from secondary memory/ hard disk to primary memory/ RAM a
* Ready: In a ready state, the process should be loaded into the primary memory, which is ready for execution.
* Waiting: The process is waiting for the allocation of CPU time and other resources for execution.
* Executing: The process is an execution state.
* Blocked: It is a time interval when a process is waiting for an event like I/O operations to complete.
* Suspended: Suspended state defines the time when a process is ready for execution but has not been placed in the ready queue by OS.
* Terminated: Terminated state specifies the time when a process is terminated



## Process Control Block(PCB)

Every process is represented in the operating system by a process control block, which is also called a task control block.

Important components of PCB are:-

* **Process state:**A process can be new, ready, running, waiting, etc.
* **Program counter:**The program counter lets you know the address of the next instruction, which should be executed for that process.
* **CPU registers:**This componentincludes accumulators, index and general-purpose registers, and information of condition code.
* **CPU scheduling information:**This componentincludes a process priority, pointers for scheduling queues, and various other scheduling parameters.
* **Accounting and business information:**It includes the amount of CPU and time utilities like real time used, job or process numbers, etc.
* **Memory-management information:**This information includes the value of the base and limit registers, the page, or segment tables. This depends on the memory system, which is used by the operating system.
* **I/O status information:**This block includes a list of open files, the list of I/O devices that are allocated to the process, etc

## What is Deadlock?

**Deadlock** is a situation that occurs in OS when any process enters a waiting state because another waiting process is holding the demanded resource. Deadlock is a common problem in multi-processing where several processes share a specific type of mutually exclusive resource known as a soft lock or software. A real-world example would be traffic, which is going only in one direction.

**Advantages of Deadlock**

Here, are pros/benefits of using Deadlock method

* This situation works well for processes which perform a single burst of activity
* No preemption needed for Deadlock.
* Convenient method when applied to resources whose state can be saved and restored easily
* Feasible to enforce via compile-time checks
* Needs no run-time computation since the problem is solved in system design

**Disadvantages of Deadlock method**

Here, are cons/ drawback of using deadlock method

* Delays process initiation
* Processes must know future resource need
* Pre-empts more often than necessary
* Dis-allows incremental resource requests
* Inherent preemption losses.

## What is First Come First Serve or First In First Out Method?

**First Come First Serve (FCFS)** is an operating system scheduling algorithm that automatically executes queued requests and processes in order of their arrival. It is the easiest and simplest CPU scheduling algorithm. In this type of algorithm, processes which request the CPU first get the CPU allocation first. This is managed with a FIFO queue. The full form of FCFS is First Come First Serve.

As the process enters the ready queue, its PCB (Process Control Block) is linked with the tail of the queue and, when the CPU becomes free, it should be assigned to the process at the beginning of the queue.

**Characteristics of FCFS method**

* It supports non-preemptive and pre-emptive scheduling algorithm.
* Jobs are always executed on a first-come, first-serve basis.
* It is easy to implement and use.
* This method is poor in performance, and the general wait time is quite high.

**Advantages of FCFS**

Here, are pros/benefits of using FCFS scheduling algorithm:

* The simplest form of a CPU scheduling algorithm
* Easy to program
* First come first served

**Disadvantages of FCFS**

Here, are cons/ drawbacks of using FCFS scheduling algorithm:

* It is a Non-Preemptive CPU scheduling algorithm, so after the process has been allocated to the CPU, it will never release the CPU until it finishes executing.
* The Average Waiting Time is high.
* Short processes that are at the back of the queue have to wait for the long process at the front to finish.
* Not an ideal technique for time-sharing systems.
* Because of its simplicity, FCFS is not very efficient.

**What is Paging?**

**Paging** is a storage mechanism that allows OS to retrieve processes from the secondary storage into the main memory in the form of pages. In the Paging method, the main memory is divided into small fixed-size blocks of physical memory, which is called frames. The size of a frame should be kept the same as that of a page to have maximum utilization of the main memory and to avoid external fragmentation. Paging is used for faster access to data, and it is a logical concept.

**What is Paging Protection?**

The paging process should be protected by using the concept of insertion of an additional bit called Valid/Invalid bit. Paging Memory protection in paging is achieved by associating protection bits with each page. These bits are associated with each page table entry and specify protection on the corresponding page.

**Advantages of Paging**

Here, are advantages of using Paging method:

* Easy to use memory management algorithm
* No need for external Fragmentation
* Swapping is easy between equal-sized pages and page frames.

**Disadvantages of Paging**

Here, are drawback/ cons of Paging:

* May cause Internal fragmentation
* Complex memory management algorithm
* Page tables consume additonal memory.
* Multi-level paging may lead to memory reference overhead.

**What is Segmentation?**

Segmentation method works almost similarly to paging, only difference between the two is that segments are of variable-length whereas, in the paging method, pages are always of fixed size.

A program segment includes the program's main function, data structures, utility functions, etc. The OS maintains a segment map table for all the processes. It also includes a list of free memory blocks along with its size, segment numbers, and it's memory locations in the main memory or virtual memory.

**Advantages of a Segmentation method**

Here, are pros/benefits of Segmentation

* Offer protection within the segments
* You can achieve sharing by segments referencing multiple processes.
* Not offers internal fragmentation
* Segment tables use lesser memory than paging

**Disadvantages of Segmentation**

Here are cons/drawback of Segmentation

* In segmentation method, processes are loaded/ removed from the main memory. Therefore, the free memory space is separated into small pieces which may create a problem of external fragmentation
* Costly memory management algorithm

**What is Round-Robin Scheduling?**

The name of this algorithm comes from the round-robin principle, where each person gets an equal share of something in turns. It is the oldest, simplest scheduling algorithm, which is mostly used for multitasking.

In Round-robin scheduling, each ready task runs turn by turn only in a cyclic queue for a limited time slice. This algorithm also offers starvation free execution of processes.

**Characteristics of Round-Robin Scheduling**

Here are the important characteristics of Round-Robin Scheduling:

* Round robin is a pre-emptive algorithm
* The CPU is shifted to the next process after fixed interval time, which is called time quantum/time slice.
* The process that is preempted is added to the end of the queue.
* Round robin is a hybrid model which is clock-driven
* Time slice should be minimum, which is assigned for a specific task that needs to be processed. However, it may differ OS to OS.
* It is a real time algorithm which responds to the event within a specific time limit.
* Round robin is one of the oldest, fairest, and easiest algorithm.
* Widely used scheduling method in traditional OS.

**Advantage of Round-robin Scheduling**

Here, are pros/benefits of Round-robin scheduling method:

* It doesn't face the issues of starvation or convoy effect.
* All the jobs get a fair allocation of CPU.
* It deals with all process without any priority
* If you know the total number of processes on the run queue, then you can also assume the worst-case response time for the same process.
* This scheduling method does not depend upon burst time. That's why it is easily implementable on the system.
* Once a process is executed for a specific set of the period, the process is preempted, and another process executes for that given time period.
* Allows OS to use the Context switching method to save states of preempted processes.
* It gives the best performance in terms of average response time.

**Disadvantages of Round-robin Scheduling**

Here, are drawbacks/cons of using Round-robin scheduling:

* If slicing time of OS is low, the processor output will be reduced.
* This method spends more time on context switching
* Its performance heavily depends on time quantum.
* Priorities cannot be set for the processes.
* Round-robin scheduling doesn't give special priority to more important tasks.
* Decreases comprehension
* Lower time quantum results in higher the context switching overhead in the system.
* Finding a correct time quantum is a quite difficult task in this system.

**What is Process Scheduling?**

**Process Scheduling** is an OS task that schedules processes of different states like ready, waiting, and running.

Process scheduling allows OS to allocate a time interval of CPU execution for each process. Another important reason for using a process scheduling system is that it keeps the CPU busy all the time. This allows you to get the minimum response time for programs.

**Process Scheduling Queues**

Process Scheduling Queues help you to maintain a distinct queue for each and every process states and PCBs. All the process of the same execution state are placed in the same queue. Therefore, whenever the state of a process is modified, its PCB needs to be unlinked from its existing queue, which moves back to the new state queue.

Three types of operating system queues are:

1. **Job queue** – It helps you to store all the processes in the system.
2. **Ready queue** – This type of queue helps you to set every process residing in the main memory, which is ready and waiting to execute.
3. **Device queues** – It is a process that is blocked because of the absence of an I/O device.

## Two State Process Model

Two-state process models are:

* Running
* Not Running

### Running

In the Operating system, whenever a new process is built, it is entered into the system, which should be running.

### Not Running

The process that are not running are kept in a queue, which is waiting for their turn to execute. Each entry in the queue is a point to a specific process.

**cheduling Objectives**

Here, are important objectives of Process scheduling

* Maximize the number of interactive users within acceptable response times.
* Achieve a balance between response and utilization.
* Avoid indefinite postponement and enforce priorities.
* It also should give reference to the processes holding the key resources.

**Type of Process Schedulers**

A scheduler is a type of system software that allows you to handle process scheduling.

There are mainly three types of Process Schedulers:

1. Long Term
2. Short Term
3. Medium Term

**Long Term Scheduler**

Long term scheduler is also known as a **job scheduler**. This scheduler regulates the program and select process from the queue and loads them into memory for execution. It also regulates the degree of multi-programing.

However, the main goal of this type of scheduler is to offer a balanced mix of jobs, like Processor, I/O jobs., that allows managing multiprogramming.

**Medium Term Scheduler**

Medium-term scheduling is an important part of **swapping**. It enables you to handle the swapped out-processes. In this scheduler, a running process can become suspended, which makes an I/O request.

A running process can become suspended if it makes an I/O request. A suspended processes can't make any progress towards completion. In order to remove the process from memory and make space for other processes, the suspended process should be moved to secondary storage.

**Short Term Scheduler**

Short term scheduling is also known as **CPU scheduler**. The main goal of this scheduler is to boost the system performance according to set criteria. This helps you to select from a group of processes that are ready to execute and allocates CPU to one of them. The dispatcher gives control of the CPU to the process selected by the short term scheduler.

**Difference between Schedulers**

Long-Term Vs. Short Term Vs. Medium-Term

| **Long-Term** | **Short-Term** | **Medium-Term** |
| --- | --- | --- |
| Long term is also known as a job scheduler | Short term is also known as CPU scheduler | Medium-term is also called swapping scheduler. |
| It is either absent or minimal in a time-sharing system. | It is insignificant in the time-sharing order. | This scheduler is an element of Time-sharing systems. |
| Speed is less compared to the short term scheduler. | Speed is the fastest compared to the short-term and medium-term scheduler. | It offers medium speed. |
| Allow you to select processes from the loads and pool back into the memory | It only selects processes that is in a ready state of the execution. | It helps you to send process back to memory. |
| Offers full control | Offers less control | Reduce the level of multiprogramming. |

**What is Context switch?**

It is a method to store/restore the state or of a CPU in PCB. So that process execution can be resumed from the same point at a later time. The context switching method is important for multitasking OS.

**What is Priority Scheduling?**

**Priority Scheduling** is a method of scheduling processes that is based on priority. In this algorithm, the scheduler selects the tasks to work as per the priority.

The processes with higher priority should be carried out first, whereas jobs with equal priorities are carried out on a round-robin or FCFS basis. Priority depends upon memory requirements, time requirements, etc.

## Types of Priority Scheduling

Priority scheduling divided into two main types:

### Preemptive Scheduling

In Preemptive Scheduling, the tasks are mostly assigned with their priorities. Sometimes it is important to run a task with a higher priority before another lower priority task, even if the lower priority task is still running. The lower priority task holds for some time and resumes when the higher priority task finishes its execution.

### Non-Preemptive Scheduling

In this type of scheduling method, the CPU has been allocated to a specific process. The process that keeps the CPU busy, will release the CPU either by switching context or terminating. It is the only method that can be used for various hardware platforms. That's because it doesn't need special hardware (for example, a timer) like preemptive scheduling.

## Characteristics of Priority Scheduling

* A CPU algorithm that schedules processes based on priority.
* It used in Operating systems for performing batch processes.
* If two jobs having the same priority are READY, it works on a FIRST COME, FIRST SERVED basis.
* In priority scheduling, a number is assigned to each process that indicates its priority level.
* Lower the number, higher is the priority.
* In this type of scheduling algorithm, if a newer process arrives, that is having a higher priority than the currently running process, then the currently running process is pre

**Advantages of priority scheduling**

Here, are benefits/pros of using priority scheduling method:

* Easy to use scheduling method
* Processes are executed on the basis of priority so high priority does not need to wait for long which saves time
* This method provides a good mechanism where the relative important of each process may be precisely defined.
* Suitable for applications with fluctuating time and resource requirements.

**Disadvantages of priority scheduling**

Here, are cons/drawbacks of priority scheduling

* If the system eventually crashes, all low priority processes get lost.
* If high priority processes take lots of CPU time, then the lower priority processes may starve and will be postponed for an indefinite time.
* This scheduling algorithm may leave some low priority processes waiting indefinitely.
* A process will be blocked when it is ready to run but has to wait for the CPU because some other process is running currently.
* If a new higher priority process keeps on coming in the ready queue, then the process which is in the waiting state may need to wait for a long duration of time.

**What is Memory Management?**

**Memory Management** is the process of controlling and coordinating computer memory, assigning portions known as blocks to various running programs to optimize the overall performance of the system.

It is the most important function of an operating system that manages primary memory. It helps processes to move back and forward between the main memory and execution disk. It helps OS to keep track of every memory location, irrespective of whether it is allocated to some process or it remains free.

## Why Use Memory Management?

Here, are reasons for using memory management:

* It allows you to check how much memory needs to be allocated to processes that decide which processor should get memory at what time.
* Tracks whenever inventory gets freed or unallocated. According to it will update the status.
* It allocates the space to application routines.
* It also make sure that these applications do not interfere with each other.
* Helps protect different processes from each other
* It places the programs in memory so that memory is utilized to its full extent.

## Memory Management Techniques

Here, are some most crucial memory management techniques:

### Single Contiguous Allocation

It is the easiest memory management technique. In this method, all types of computer's memory except a small portion which is reserved for the OS is available for one application. For example, MS-DOS operating system allocates memory in this way. An embedded system also runs on a single application.

### Partitioned Allocation

It divides primary memory into various memory partitions, which is mostly contiguous areas of memory. Every partition stores all the information for a specific task or job. This method consists of allotting a partition to a job when it starts & unallocate when it ends.

### Paged Memory Management

This method divides the computer's main memory into fixed-size units known as page frames. This hardware memory management unit maps pages into frames which should be allocated on a page basis.

### Segmented Memory Management

Segmented memory is the only memory management method that does not provide the user's program with a linear and contiguous address space.

Segments need hardware support in the form of a segment table. It contains the physical address of the section in memory, size, and other data like access protection bits and status.

## What is Swapping?

Swapping is a method in which the process should be swapped temporarily from the main memory to the backing store. It will be later brought back into the memory for continue execution.

Backing store is a hard disk or some other secondary storage device that should be big enough inorder to accommodate copies of all memory images for all users. It is also capable of offering direct access to these memory images.

### Benefits of Swapping

Here, are major benefits/pros of swapping:

* It offers a higher degree of multiprogramming.
* Allows dynamic relocation. For example, if address binding at execution time is being used, then processes can be swap in different locations. Else in case of compile and load time bindings, processes should be moved to the same location.
* It helps to get better utilization of memory.
* Minimum wastage of CPU time on completion so it can easily be applied to a priority-based scheduling method to improve its performance.

**What is Memory allocation?**

Memory allocation is a process by which computer programs are assigned memory or space.

Here, main memory is divided into two types of partitions

1. **Low Memory** - Operating system resides in this type of memory.
2. **High Memory**- User processes are held in high memory.

**Partition Allocation**

Memory is divided into different blocks or partitions. Each process is allocated according to the requirement. Partition allocation is an ideal method to avoid internal fragmentation.

Below are the various partition allocation schemes :

* **First Fit**: In this type fit, the partition is allocated, which is the first sufficient block from the beginning of the main memory.
* **Best Fit:**It allocates the process to the partition that is the first smallest partition among the free partitions.
* **Worst Fit:**It allocates the process to thepartition, which is the largest sufficient freely available partition in the main memory.
* **Next Fit:** It is mostly similar to the first Fit, but this Fit, searches for the first sufficient partition from the last allocation point.

**What is Paging?**

Paging is a storage mechanism that allows OS to retrieve processes from the secondary storage into the main memory in the form of pages. In the Paging method, the main memory is divided into small fixed-size blocks of physical memory, which is called frames. The size of a frame should be kept the same as that of a page to have maximum utilization of the main memory and to avoid external fragmentation. Paging is used for faster access to data, and it is a logical concept.

**What is Fragmentation?**

Processes are stored and removed from memory, which creates free memory space, which are too small to use by other processes.

After sometimes, that processes not able to allocate to memory blocks because its small size and memory blocks always remain unused is called fragmentation. This type of problem happens during a dynamic memory allocation system when free blocks are quite small, so it is not able to fulfill any request.

Two types of Fragmentation methods are:

1. External fragmentation
2. Internal fragmentation

* External fragmentation can be reduced by rearranging memory contents to place all free memory together in a single block.
* The internal fragmentation can be reduced by assigning the smallest partition, which is still good enough to carry the entire process.

**What is Segmentation?**

Segmentation method works almost similarly to paging. The only difference between the two is that segments are of variable-length, whereas, in the paging method, pages are always of fixed size.

A program segment includes the program's main function, data structures, utility functions, etc. The OS maintains a segment map table for all the processes. It also includes a list of free memory blocks along with its size, segment numbers, and its memory locations in the main memory or virtual memory.

**What is Dynamic Loading?**

Dynamic loading is a routine of a program which is not loaded until the program calls it. All routines should be contained on disk in a relocatable load format. The main program will be loaded into memory and will be executed. Dynamic loading also provides better memory space utilization.

**What is Dynamic Linking?**

Linking is a method that helps OS to collect and merge various modules of code and data into a single executable file. The file can be loaded into memory and executed. OS can link system-level libraries into a program that combines the libraries at load time. In Dynamic linking method, libraries are linked at execution time, so program code size can remain small.

**Difference Between Static and Dynamic Loading**

|  |  |
| --- | --- |
| **Static Loading** | **Dynamic Loading** |
| Static loading is used when you want to load your program statically. Then at the time of compilation, the entire program will be linked and compiled without need of any external module or program dependency. | In a Dynamically loaded program, references will be provided and the loading will be done at the time of execution. |
| At loading time, the entire program is loaded into memory and starts its execution. | Routines of the library are loaded into memory only when they are required in the program. |

**Difference Between Static and Dynamic Linking**

Here, are main difference between Static vs. Dynamic Linking:

|  |  |
| --- | --- |
| **Static Linking** | **Dynamic Linking** |
| Static linking is used to combine all other modules, which are required by a program into a single executable code. This helps OS prevent any runtime dependency. | When dynamic linking is used, it does not need to link the actual module or library with the program. Instead of it use a reference to the dynamic module provided at the time of compilation and linking. |

**What is Shortest Job First Scheduling?**

**Shortest Job First (SJF)** is an algorithm in which the process having the smallest execution time is chosen for the next execution. This scheduling method can be preemptive or non-preemptive. It significantly reduces the average waiting time for other processes awaiting execution. The full form of SJF is Shortest Job First.

**There are basically two types of SJF methods:**

* Non-Preemptive SJF
* Preemptive SJF

**Characteristics of SJF Scheduling**

* It is associated with each job as a unit of time to complete.
* This algorithm method is helpful for batch-type processing, where waiting for jobs to complete is not critical.
* It can improve process throughput by making sure that shorter jobs are executed first, hence possibly have a short turnaround time.
* It improves job output by offering shorter jobs, which should be executed first, which mostly have a shorter turnaround time.

**Non-Preemptive SJF**

In non-preemptive scheduling, once the CPU cycle is allocated to process, the process holds it till it reaches a waiting state or terminated.

**Preemptive SJF**

In Preemptive SJF Scheduling, jobs are put into the ready queue as they come. A process with shortest burst time begins execution. If a process with even a shorter burst time arrives, the current process is removed or preempted from execution, and the shorter job is allocated CPU cycle.

**Advantages of SJF**

Here are the benefits/pros of using SJF method:

* SJF is frequently used for long term scheduling.
* It reduces the average waiting time over FIFO (First in First Out) algorithm.
* SJF method gives the lowest average waiting time for a specific set of processes.
* It is appropriate for the jobs running in batch, where run times are known in advance.
* For the batch system of long-term scheduling, a burst time estimate can be obtained from the job description.
* For Short-Term Scheduling, we need to predict the value of the next burst time.
* Probably optimal with regard to average turnaround time.

**Disadvantages/Cons of SJF**

Here are some drawbacks/cons of SJF algorithm:

* Job completion time must be known earlier, but it is hard to predict.
* It is often used in a batch system for long term scheduling.
* SJF can't be implemented for CPU scheduling for the short term. It is because there is no specific method to predict the length of the upcoming CPU burst.
* This algorithm may cause very long turnaround times or starvation.
* Requires knowledge of how long a process or job will run.
* It leads to the starvation that does not reduce average turnaround time.
* It is hard to know the length of the upcoming CPU request.
* Elapsed time should be recorded, that results in more overhead on the proce

**What is Virtual Memory?**

**Virtual Memory** is a storage mechanism which offers user an illusion of having a very big main memory. It is done by treating a part of secondary memory as the main memory. In Virtual memory, the user can store processes with a bigger size than the available main memory.

Therefore, instead of loading one long process in the main memory, the OS loads the various parts of more than one process in the main memory. Virtual memory is mostly implemented with demand paging and demand segmentation.

**Why Need Virtual Memory?**

Here, are reasons for using virtual memory:

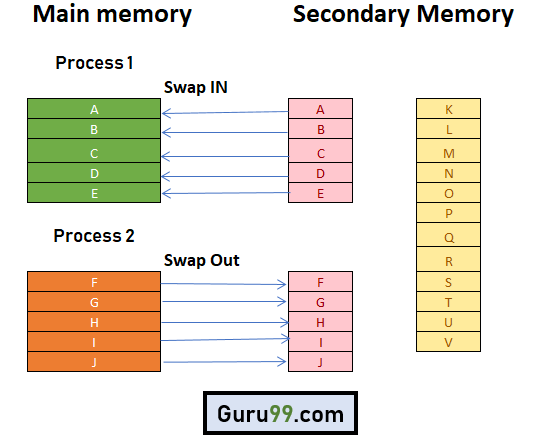
* Whenever your computer doesn't have space in the physical memory it writes what it needs to remember to the hard disk in a swap file as virtual memory.
* If a computer running Windows needs more memory/RAM, then installed in the system, it uses a small portion of the hard drive for this purpose.

**How Virtual Memory Works?**

In the modern world, virtual memory has become quite common these days. It is used whenever some pages require to be loaded in the main memory for the execution, and the memory is not available for those many pages.

So, in that case, instead of preventing pages from entering in the main memory, the OS searches for the RAM space that are minimum used in the recent times or that are not referenced into the secondary memory to make the space for the new pages in the main memory.

**What is Demand Paging?**



A demand paging mechanism is very much similar to a paging system with swapping where processes stored in the secondary memory and pages are loaded only on demand, not in advance.

So, when a context switch occurs, the OS never copy any of the old program's pages from the disk or any of the new program's pages into the main memory. Instead, it will start executing the new program after loading the first page and fetches the program's pages, which are referenced.

During the program execution, if the program references a page that may not be available in the main memory because it was swapped, then the processor considers it as an invalid memory reference. That's because the page fault and transfers send control back from the program to the OS, which demands to store page back into the memory.

## Types of Page Replacement Methods

Here, are some important Page replacement methods

* FIFO
* Optimal Algorithm
* LRU Page Replacement

## FIFO Page Replacement

FIFO (First-in-first-out) is a simple implementation method. In this method, memory selects the page for a replacement that has been in the virtual address of the memory for the longest time.

### Features:

* Whenever a new page loaded, the page recently comes in the memory is removed. So, it is easy to decide which page requires to be removed as its identification number is always at the FIFO stack.
* The oldest page in the main memory is one that should be selected for replacement first.

## Optimal Algorithm

The optimal page replacement method selects that page for a replacement for which the time to the next reference is the longest.

### Features:

* Optimal algorithm results in the fewest number of page faults. This algorithm is difficult to implement.
* An optimal page-replacement algorithm method has the lowest page-fault rate of all algorithms. This algorithm exists and which should be called MIN or OPT.
* Replace the page which unlike to use for a longer period of time. It only uses the time when a page needs to be used.

## LRU Page Replacement

The full form of LRU is the Least Recently Used page. This method helps OS to find page usage over a short period of time. This algorithm should be implemented by associating a counter with an even- page.

### How does it work?

* Page, which has not been used for the longest time in the main memory, is the one that will be selected for replacement.
* Easy to implement, keep a list, replace pages by looking back into time.

### Features:

* The LRU replacement method has the highest count. This counter is also called aging registers, which specify their age and how much their associated pages should also be referenced.
* The page which hasn't been used for the longest time in the main memory is the one that should be selected for replacement.
* It also keeps a list and replaces pages by looking back into time.

### Fault rate

Fault rate is a frequency with which a designed system or component fails. It is expressed in failures per unit of time. It is denoted by the Greek letter λ (lambda).

## Advantages of Virtual Memory

Here, are pros/benefits of using Virtual Memory:

* Virtual memory helps to gain speed when only a particular segment of the program is required for the execution of the program.
* It is very helpful in implementing a multiprogramming environment.
* It allows you to run more applications at once.
* It helps you to fit many large programs into smaller programs.
* Common data or code may be shared between memory.
* Process may become even larger than all of the physical memory.
* Data / code should be read from disk whenever required.
* The code can be placed anywhere in physical memory without requiring relocation.
* More processes should be maintained in the main memory, which increases the effective use of CPU.
* Each page is stored on a disk until it is required after that, it will be removed.
* It allows more applications to be run at the same time.
* There is no specific limit on the degree of multiprogramming.
* Large programs should be written, as virtual address space available is more compared to physical memory.

## Disadvantages of Virtual Memory

Here, are drawbacks/cons of using virtual memory:

* Applications may run slower if the system is using virtual memory.
* Likely takes more time to switch between applications.
* Offers lesser hard drive space for your use.
* It reduces system stability.
* It allows larger applications to run in systems that don't offer enough physical RAM alone to run them.
* It doesn't offer the same performance as RAM.
* It negatively affects the overall performance of a system.
* Occupy the storage space, which may be used otherwise for long term data storage.

**What is Paging?**

Paging is a storage mechanism that allows OS to retrieve processes from the secondary storage into the main memory in the form of pages. In the Paging method, the main memory is divided into small fixed-size blocks of physical address, which is called frames.

The size of a frame should be kept the same as that of a page to have maximum utilization of the main memory and to avoid external fragmentation. Paging is used for faster access to data, and it is a logical concept.

**What is Segmentation?**

Segmentation method works almost similarly to paging, only difference between the two is that segments are of variable-length whereas, in the paging method, pages are always of fixed size.

A program segment includes the program's main function, data structures, utility functions, etc. The OS maintains a segment map table for all the processes. It also includes a list of free memory blocks along with its size, segment numbers, and it's memory locations in the main memory or virtual memory.

Here, are differences between Paging and Segmentation method:

|  |  |
| --- | --- |
| **Paging** | **Segmentation** |
| A page is of the fixed block size. | A segment is of variable size. |
| It may lead to internal fragmentation. | It may lead to external fragmentation. |
| In Paging, the hardware decides the page size. | The segment size is specified by the user. |
| A process address space is broken into fixed-sized blocks, which is called pages. | A process address space Is broken in differing sized blocks called sections. |
| The paging technique is faster for memory access. | Segmentation is slower than paging method. |
| Page table stores the page data | Segmentation table stores the segmentation data. |
| Paging does not facilitate any sharing of procedures. | Segmentation allows for the sharing of procedures. |
| Paging fails to distinguish and secure procedures and data separately. | Segmentation can be able to separate secure procedures and data. |
| Paging address space is one dimensional | In segmentation, there is the availability of many independent address spaces |
| In paging, the user just provides a single integer as the address, that is divided by the hardware into a page number and offset. | In the segmentation method, the user specifies the address in two quantities 1)segment number 2)offset. |

**Advantages of Paging**

Here, are pros/benefits of using Paging method

* On the programmer level, paging is a transparent function and does not require intervention.
* Frames do not have to be contiguous.
* Easy to use memory management algorithm
* Swapping is easy between equal-sized pages and page frames.

**Advantage of Segmentation**

Here, are pros/benefits of Segmentation:

* Simple to relocate segments than the entire address space.
* The absence of internal fragmentation as external fragmentation has to be done.
* The segment table is of lesser size compared with the page table in paging.
* The average size of the segment is larger to the actual size of the page
* Offer protection within the segment
* Not offers internal fragmentation
* Segment tables use lesser memory than paging
* As it offers the small segment table, memory reference is simple, which lends itself to sharing data among processes.

**Disadvantages of Paging**

Here, are cons/drawbacks of Paging:

* Paging causes internal fragmentation on older systems.
* Longer memory lookup times compared to segmentation
* It may cause internal fragmentation
* Complex memory management algorithm
* Page tables consume additional memory.
* Multi-level paging may lead to memory reference overhead.

**Disadvantages of Segmentation**

Here, are cons/drawbacks of Segmentation:

* Un-equal size of segments is not good in the case of swapping.
* Porting Linux to different architectures is difficult to process because it offers very limited support for segmentation.
* It demands programmer intervention.
* It is hard to allocate contagious memory to partition as it is of its variable size.
* This is costly memory management algorithm.

**KEY DIFFERENCES:**

* A page is of the fixed block size whereas a segment is of variable size.
* In Paging, the hardware decides the page size while the segment size is specified by the user.
* The paging technique is faster for memory access but segmentation is slower than paging method.
* In paging, the user just provides a single integer as the address while in segmentation, there is the availability of many independent address spaces
* Page table stores the page data while Segmentation table stores the segmentat

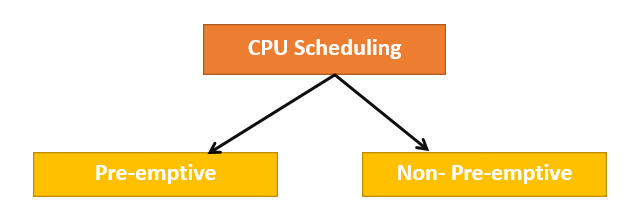
( SEE THIS LINK FOR BETTER UNDERSTAND all <https://www.guru99.com/process-management-pcb.html> )

**What is CPU Scheduling?**

**CPU Scheduling** is a process of determining which process will own CPU for execution while another process is on hold. The main task of CPU scheduling is to make sure that whenever the CPU remains idle, the OS at least select one of the processes available in the ready queue for execution. The selection process will be carried out by the CPU scheduler. It selects one of the processes in memory that are ready for execution.

## Types of CPU Scheduling

## Here are two kinds of Scheduling methods:

[](https://www.guru99.com/images/1/122519_0449_CPUscheduli1.png)

### Preemptive Scheduling

In Preemptive Scheduling, the tasks are mostly assigned with their priorities. Sometimes it is important to run a task with a higher priority before another lower priority task, even if the lower priority task is still running. The lower priority task holds for some time and resumes when the higher priority task finishes its execution.

### Non-Preemptive Scheduling

In this type of scheduling method, the CPU has been allocated to a specific process. The process that keeps the CPU busy will release the CPU either by switching context or terminating. It is the only method that can be used for various hardware platforms. That's because it doesn't need special hardware (for example, a timer) like preemptive scheduling.

### When scheduling is Preemptive or Non-Preemptive?

To determine if scheduling is preemptive or non-preemptive, consider these four parameters:

1. A process switches from the running to the waiting state.
2. Specific process switches from the running state to the ready state.
3. Specific process switches from the waiting state to the ready state.
4. Process finished its execution and terminated.

**Only conditions 1 and 4 apply, the scheduling is called non- preemptive.**

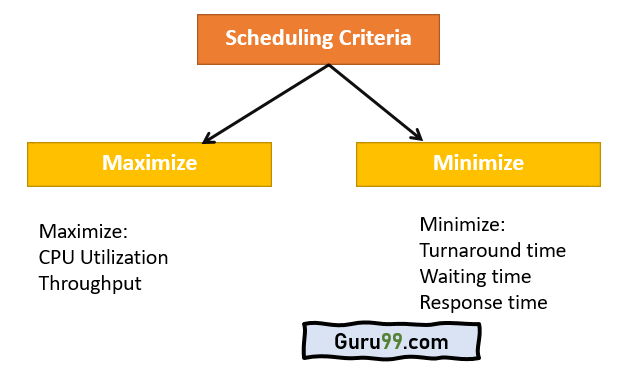
**All other scheduling are preemptive.**

## Important CPU scheduling Terminologies

* **Burst Time/Execution Time:** It is a time required by the process to complete execution. It is also called running time.
* **Arrival Time:** when a process enters in a ready state
* **Finish Time:** when process complete and exit from a system
* **Multiprogramming:** A number of programs which can be present in memory at the same time.
* **Jobs:** It is a type of program without any kind of user interaction.
* **User:** It is a kind of program having user interaction.
* **Process:** It is the reference that is used for both job and user.
* **CPU/IO burst cycle:** Characterizes process execution, which alternates between CPU and I/O activity. CPU times are usually shorter than the time of I/O.

## CPU Scheduling Criteria

A CPU scheduling algorithm tries to maximize and minimize the following:

[](https://www.guru99.com/images/1/122519_0449_CPUscheduli2.png)

### Maximize:

**CPU utilization:**CPU utilization is the main task in which the operating system needs to make sure that CPU remains as busy as possible. It can range from 0 to 100 percent. However, for the RTOS, it can be range from 40 percent for low-level and 90 percent for the high-level system.

**Throughput:**The number of processes that finish their execution per unit time is known Throughput. So, when the CPU is busy executing the process, at that time, work is being done, and the work completed per unit time is called Throughput.

### Minimize:

**Waiting time:**Waiting time is an amount that specific process needs to wait in the ready queue.

**Response time:**It is an amount to time in which the request was submitted until the first response is produced.

**Turnaround Time:**Turnaround time is an amount of time to execute a specific process. It is the calculation of the total time spent waiting to get into the memory, waiting in the queue and, executing on the CPU. The period between the time of process submission to the completion time is the turnaround time.

**Interval Timer**

Timer interruption is a method that is closely related to preemption. When a certain process gets the CPU allocation, a timer may be set to a specified interval. Both timer interruption and preemption force a process to return the CPU before its CPU burst is complete.

Most of the multi-programmed operating system uses some form of a timer to prevent a process from tying up the system forever.

**What is Dispatcher?**

It is a module that provides control of the CPU to the process. The Dispatcher should be fast so that it can run on every context switch. Dispatch latency is the amount of time needed by the CPU scheduler to stop one process and start another.

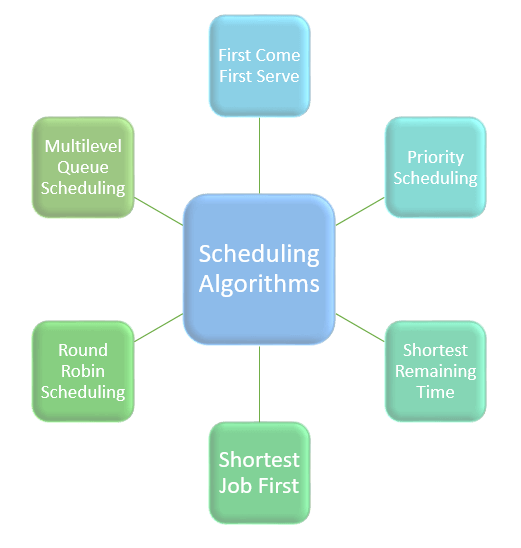
Functions performed by Dispatcher:

* Context Switching
* Switching to user mode
* Moving to the correct location in the newly loaded program.

**Types of CPU scheduling Algorithm**

There are mainly six types of process scheduling algorithms

1. First Come First Serve (FCFS)
2. Shortest-Job-First (SJF) Scheduling
3. Shortest Remaining Time
4. Priority Scheduling
5. Round Robin Scheduling
6. Multilevel Queue Scheduling



## First Come First Serve

First Come First Serve is the full form of FCFS. It is the easiest and most simple CPU scheduling algorithm. In this type of algorithm, the process which requests the CPU gets the CPU allocation first. This scheduling method can be managed with a FIFO queue.

As the process enters the ready queue, its PCB (Process Control Block) is linked with the tail of the queue. So, when CPU becomes free, it should be assigned to the process at the beginning of the queue.

### Characteristics of FCFS method:

* It offers non-preemptive and pre-emptive scheduling algorithm.
* Jobs are always executed on a first-come, first-serve basis
* It is easy to implement and use.
* However, this method is poor in performance, and the general wait time is quite high.

## Shortest Remaining Time

The full form of SRT is Shortest remaining time. It is also known as SJF preemptive scheduling. In this method, the process will be allocated to the task, which is closest to its completion. This method prevents a newer ready state process from holding the completion of an older process.

### Characteristics of SRT scheduling method:

* This method is mostly applied in batch environments where short jobs are required to be given preference.
* This is not an ideal method to implement it in a shared system where the required CPU time is unknown.
* Associate with each process as the length of its next CPU burst. So that operating system uses these lengths, which helps to schedule the process with the shortest possible time.

## Priority Based Scheduling

Priority scheduling is a method of scheduling processes based on priority. In this method, the scheduler selects the tasks to work as per the priority.

Priority scheduling also helps OS to involve priority assignments. The processes with higher priority should be carried out first, whereas jobs with equal priorities are carried out on a round-robin or FCFS basis. Priority can be decided based on memory requirements, time requirements, etc.

## Round-Robin Scheduling

Round robin is the oldest, simplest scheduling algorithm. The name of this algorithm comes from the round-robin principle, where each person gets an equal share of something in turn. It is mostly used for scheduling algorithms in multitasking. This algorithm method helps for starvation free execution of processes.

### Characteristics of Round-Robin Scheduling

* Round robin is a hybrid model which is clock-driven
* Time slice should be minimum, which is assigned for a specific task to be processed. However, it may vary for different processes.
* It is a real time system which responds to the event within a specific time limit.

## Shortest Job First

SJF is a full form of (Shortest job first) is a scheduling algorithm in which the process with the shortest execution time should be selected for execution next. This scheduling method can be preemptive or non-preemptive. It significantly reduces the average waiting time for other processes awaiting execution.

### Characteristics of SJF Scheduling

* It is associated with each job as a unit of time to complete.
* In this method, when the CPU is available, the next process or job with the shortest completion time will be executed first.
* It is Implemented with non-preemptive policy.
* This algorithm method is useful for batch-type processing, where waiting for jobs to complete is not critical.
* It improves job output by offering shorter jobs, which should be executed first, which mostly have a shorter turnaround time.

## Multiple-Level Queues Scheduling

This algorithm separates the ready queue into various separate queues. In this method, processes are assigned to a queue based on a specific property of the process, like the process priority, size of the memory, etc.

However, this is not an independent scheduling OS algorithm as it needs to use other types of algorithms in order to schedule the jobs.

### Characteristic of Multiple-Level Queues Scheduling:

* Multiple queues should be maintained for processes with some characteristics.
* Every queue may have its separate scheduling algorithms.
* Priorities are given for each queue.

## The Purpose of a Scheduling algorithm

Here are the reasons for using a scheduling algorithm:

* The CPU uses scheduling to improve its efficiency.
* It helps you to allocate resources among competing processes.
* The maximum utilization of CPU can be obtained with multi-programming.
* The processes which are to be executed are in ready queue.

# Logical and Physical Address in Operating System

**Logical Address** is generated by CPU while a program is running. The logical address is virtual address as it does not exist physically, therefore, it is also known as Virtual Address. This address is used as a reference to access the physical memory location by CPU. The term Logical Address Space is used for the set of all logical addresses generated by a program’s perspective.  
The hardware device called Memory-Management Unit is used for mapping logical address to its corresponding physical address.

**Physical Address** identifies a physical location of required data in a memory. The user never directly deals with the physical address but can access by its corresponding logical address. The user program generates the logical address and thinks that the program is running in this logical address but the program needs physical memory for its execution, therefore, the logical address must be mapped to the physical address by MMU before they are used. The term Physical Address Space is used for all physical addresses corresponding to the logical addresses in a Logical address space.

**Differences Between Logical and Physical Address in Operating System**

1. The basic difference between Logical and physical address is that Logical address is generated by CPU in perspective of a program whereas the physical address is a location that exists in the memory unit.
2. Logical Address Space is the set of all logical addresses generated by CPU for a program whereas the set of all physical address mapped to corresponding logical addresses is called Physical Address Space.
3. The logical address does not exist physically in the memory whereas physical address is a location in the memory that can be accessed physically.
4. Identical logical addresses are generated by Compile-time and Load time address binding methods whereas they differs from each other in run-time address binding method. Please refer [this](https://www.geeksforgeeks.org/memory-management-mapping-virtual-address-physical-addresses/) for details.
5. The logical address is generated by the CPU while the program is running whereas the physical address is computed by the Memory Management Unit (MMU).

## File

A file is a named collection of related information that is recorded on secondary storage such as magnetic disks, magnetic tapes and optical disks. In general, a file is a sequence of bits, bytes, lines or records whose meaning is defined by the files creator and user.

## File Structure

A File Structure should be according to a required format that the operating system can understand.

* A file has a certain defined structure according to its type.
* A text file is a sequence of characters organized into lines.
* A source file is a sequence of procedures and functions.
* An object file is a sequence of bytes organized into blocks that are understandable by the machine.
* When operating system defines different file structures, it also contains the code to support these file structure. Unix, MS-DOS support minimum number of file structure.

## File Type

File type refers to the ability of the operating system to distinguish different types of file such as text files source files and binary files etc. Many operating systems support many types of files. Operating system like MS-DOS and UNIX have the following types of files −

### Ordinary files

* These are the files that contain user information.
* These may have text, databases or executable program.
* The user can apply various operations on such files like add, modify, delete or even remove the entire file.

### Directory files

* These files contain list of file names and other information related to these files.

### Special files

* These files are also known as device files.
* These files represent physical device like disks, terminals, printers, networks, tape drive etc.

These files are of two types −

* **Character special files** − data is handled character by character as in case of terminals or printers.
* **Block special files** − data is handled in blocks as in the case of disks and tapes.

## Space Allocation

Files are allocated disk spaces by operating system. Operating systems deploy following three main ways to allocate disk space to files.

* Contiguous Allocation
* Linked Allocation
* Indexed Allocation

### Contiguous Allocation

* Each file occupies a contiguous address space on disk.
* Assigned disk address is in linear order.
* Easy to implement.
* External fragmentation is a major issue with this type of allocation technique.

### Linked Allocation

* Each file carries a list of links to disk blocks.
* Directory contains link / pointer to first block of a file.
* No external fragmentation
* Effectively used in sequential access file.
* Inefficient in case of direct access file.

### Indexed Allocation

* Provides solutions to problems of contiguous and linked allocation.
* A index block is created having all pointers to files.
* Each file has its own index block which stores the addresses of disk space occupied by the file.
* Directory contains the addresses of index blocks of files.

### Operations of the file

A file has many properties so to defile the file property operating system provides a lot of information that can be performed on the file. There are many simple and easy operations of file like create, delete, update and some others like rename.

1. **Creating a file** - for creating any file there are mainly two steps first the free space is available in the system and second is the new entry of file must be made in the directory.
2. **Writing a file** - For writing a file the system call specifies the two things name of the file and the information that is written on the file. When the name of the file is given the system search that file in the directory, the write pointer point the location where the next write is to take place and write pointer is updated.
3. **Reading a file** - For reading a file the system call also specifies the two things name of the file and the read pointer. The directory is searched for the given entry and system keep track on the read pointer. The read pointer is updated once the read is completed. The same pointer is used by both the read and write operation on the file.
4. **Deleting a file** - For deleting a file first we search for the directory and then erase the directory. After that, we release the space so that it can be reused by another file.
5. **Repositioning of the file** - First, the directory is searched for the file and the current position of the file is changed by the new position.

### File Access Method

The file contains the information but when it required to used this information can be access by the access methods and reads into the computer memory. Some system provides only one access method and some provide more than on access method to access the file,

**1. Sequential Access Method**

A sequential access is that in which the records are accessed in some sequence, i.e., the information in the file is processed in order, one record after the other. This access method is the most primitive one.

The idea of Sequential access is based on the tape model which is a sequential access device. We consider Sequential access method is best because most of the records in a file are to be processed. For example, transaction files.

**Example:** Compilers usually access files in this fashion.

**2. Direct or Random Access Methods**

Sometimes it is not necessary to process every record in a file. It is not necessary to process all the records in the order in which they are present in the memory. In all such cases, direct access is used.

The disk is a direct access device which gives us the reliability to random access of any file block. In the file, there is a collection of physical blocks and the records of that blocks.

Eg. Databases are often of this type since they allow query processing that involves immediate access to large amounts of information. All reservation systems fall into this category.

Not all operating systems support direct access files. The sequential and direct access of the file is defined at the time of creation and accessed accordingly later. The direct access of a sequential file is not possible but Sequential access to a direct access file is possible.

**Index Access Method**

An indexed file is a computer file with an index that allows easy random access to any record given its file key. The key is an attribute that uniquely identifies a record. We can say that If more than one index is present the other ones are alternate indexes. The creation of the indexes is done with the file but maintained by the system.

**4. Index sequential Access Method**

The index sequential access method is a modification of the direct access method. Basically, it is kind of combination of both the sequential access as well as direct access. The main idea of this method is to first access the file directly and then it accesses sequentially. In this access method, it is necessary for maintaining an index. The index is nothing but a pointer to a block. The direct access of the index is made to access a record in a file. The information which is obtained from this access is used to access the file. Sometimes the indexes are very big. So to maintain all these hierarchies of indexes are built in which one direct access of an index leads to information of another index access.

# Structures of Directory in Operating System

A **directory** is a container that is used to contain folders and file. It organizes files and folders into a hierarchical manner.

https://media.geeksforgeeks.org/wp-content/uploads/111-11.png

* 1. **Single-level directory –**  
     Single level directory is simplest directory structure.In it all files are contained in same directory which make it easy to support and understand.

A single level directory has a significant limitation, however, when the number of files increases or when the system has more than one user. Since all the files are in the same directory, they must have the unique name . if two users call their dataset test, then the unique name rule violated.

https://media.geeksforgeeks.org/wp-content/uploads/222-13.png

1. **Advantages:**
   * Since it is a single directory, so its implementation is very easy.
   * If the files are smaller in size, searching will become faster.
   * The operations like file creation, searching, deletion, updating are very easy in such a directory structure.

**Disadvantages:**

* + There may chance of name collision because two files can not have the same name.
  + Searching will become time taking if the directory is large.
  + In this can not group the same type of files together.

1. **Two-level directory –**  
   As we have seen, a single level directory often leads to confusion of files names among different users. the solution to this problem is to create a separate directory for each user. In the two-level directory structure, each user has there own *user files directory (UFD)*. The UFDs has similar structures, but each lists only the files of a single user. system’s *master file directory (MFD)* is searches whenever a new user id=s logged in. The MFD is indexed by username or account number, and each entry points to the UFD for that user.

https://media.geeksforgeeks.org/wp-content/uploads/222-2-1.png

**Advantages:**

* We can give full path like /User-name/directory-name/.
* Different users can have same directory as well as file name.
* Searching of files become more easy due to path name and user-grouping.

**Disadvantages:**

* + A user is not allowed to share files with other users.
  + Still it not very scalable, two files of the same type cannot be grouped together in the same user.

1. **Tree-structured directory –**  
   Once we have seen a two-level directory as a tree of height 2, the natural generalization is to extend the directory structure to a tree of arbitrary height.  
   This generalization allows the user to create there own subdirectories and to organize on their files accordingly.

https://media.geeksforgeeks.org/wp-content/uploads/222-3-1.png

1. A tree structure is the most common directory structure. The tree has a root directory, and every file in the system have a unique path.

**Advantages:**

* + Very generalize, since full path name can be given.
  + Very scalable, the probability of name collision is less.
  + Searching becomes very easy, we can use both absolute path as well as relative.

**Disadvantages:**

* + Every file does not fit into the hierarchical model, files may be saved into multiple directories.
  + We can not share files.
  + It is inefficient, because accessing a file may go under multiple directories.

1. **Acyclic graph directory –**  
   An acyclic graph is a graph with no cycle and allows to share subdirectories and files. The same file or subdirectories may be in two different directories. It is a natural generalization of the tree-structured directory.

It is used in the situation like when two programmers are working on a joint project and they need to access files. The associated files are stored in a subdirectory, separating them from other projects and files of other programmers, since they are working on a joint project so they want the subdirectories to be into their own directories. The common subdirectories should be shared. So here we use Acyclic directories.

It is the point to note that shared file is not the same as copy file . If any programmer makes some changes in the subdirectory it will reflect in both subdirectories.

https://media.geeksforgeeks.org/wp-content/uploads/222-4-1.png

1. **Advantages:**  
   * We can share files.
   * Searching is easy due to different-different paths.

**Disadvantages:**

* + We share the files via linking, in case of deleting it may create the problem,
  + If the link is softlink then after deleting the file we left with a dangling pointer.
  + In case of hardlink, to delete a file we have to delete all the reference associated with it.

1. **General graph directory structure –**  
   In general graph directory structure, cycles are allowed within a directory structure where multiple directories can be derived from more than one parent directory.  
   The main problem with this kind of directory structure is to calculate total size or space that has been taken by the files and directories.

https://media.geeksforgeeks.org/wp-content/uploads/333-5.png

**Advantages:**

* It allows cycles.
* It is more flexible than other directories structure.

**Disadvantages:**

* It is more costly than others.
* It needs garbage collection.

Security refers to providing a protection system to computer system resources such as CPU, memory, disk, software programs and most importantly data/information stored in the computer system. If a computer program is run by an unauthorized user, then he/she may cause severe damage to computer or data stored in it. So a computer system must be protected against unauthorized access, malicious access to system memory, viruses, worms etc. We're going to discuss following topics in this chapter.

* Authentication
* One Time passwords
* Program Threats
* System Threats
* Computer Security Classifications

Authentication

Authentication refers to identifying each user of the system and associating the executing programs with those users. It is the responsibility of the Operating System to create a protection system which ensures that a user who is running a particular program is authentic. Operating Systems generally identifies/authenticates users using following three ways −

* **Username / Password** − User need to enter a registered username and password with Operating system to login into the system.
* **User card/key** − User need to punch card in card slot, or enter key generated by key generator in option provided by operating system to login into the system.
* **User attribute - fingerprint/ eye retina pattern/ signature** − User need to pass his/her attribute via designated input device used by operating system to login into the system.

One Time passwords

One-time passwords provide additional security along with normal authentication. In One-Time Password system, a unique password is required every time user tries to login into the system. Once a one-time password is used, then it cannot be used again. One-time password are implemented in various ways.

* **Random numbers** − Users are provided cards having numbers printed along with corresponding alphabets. System asks for numbers corresponding to few alphabets randomly chosen.
* **Secret key** − User are provided a hardware device which can create a secret id mapped with user id. System asks for such secret id which is to be generated every time prior to login.
* **Network password** − Some commercial applications send one-time passwords to user on registered mobile/ email which is required to be entered prior to login.

Program Threats

Operating system's processes and kernel do the designated task as instructed. If a user program made these process do malicious tasks, then it is known as **Program Threats**. One of the common example of program threat is a program installed in a computer which can store and send user credentials via network to some hacker. Following is the list of some well-known program threats.

* **Trojan Horse** − Such program traps user login credentials and stores them to send to malicious user who can later on login to computer and can access system resources.
* **Trap Door** − If a program which is designed to work as required, have a security hole in its code and perform illegal action without knowledge of user then it is called to have a trap door.
* **Logic Bomb** − Logic bomb is a situation when a program misbehaves only when certain conditions met otherwise it works as a genuine program. It is harder to detect.
* **Virus** − Virus as name suggest can replicate themselves on computer system. They are highly dangerous and can modify/delete user files, crash systems. A virus is generatlly a small code embedded in a program. As user accesses the program, the virus starts getting embedded in other files/ programs and can make system unusable for user

System Threats

System threats refers to misuse of system services and network connections to put user in trouble. System threats can be used to launch program threats on a complete network called as program attack. System threats creates such an environment that operating system resources/ user files are misused. Following is the list of some well-known system threats.

* **Worm** − Worm is a process which can choked down a system performance by using system resources to extreme levels. A Worm process generates its multiple copies where each copy uses system resources, prevents all other processes to get required resources. Worms processes can even shut down an entire network.
* **Port Scanning** − Port scanning is a mechanism or means by which a hacker can detects system vulnerabilities to make an attack on the system.
* **Denial of Service** − Denial of service attacks normally prevents user to make legitimate use of the system. For example, a user may not be able to use internet if denial of service attacks browser's content settings.

Computer Security Classifications

As per the U.S. Department of Defense Trusted Computer System's Evaluation Criteria there are four security classifications in computer systems: A, B, C, and D. This is widely used specifications to determine and model the security of systems and of security solutions. Following is the brief description of each classification.