**Linker**: A **linker** tool is used to **link** all the parts of the program together for execution (executable machine code).

**Loader**: A **loader** loads all of them into memory and then the program is executed.

* Compiler: A compiler is a computer program (or a set of programs) that transforms source code written in a programming language (the source language) into another computer language (the target language).

**Assembler**: An **assembler** then translates the assembly program into machine code (object).

Macro is a single line abbreviation for group of instructions.

Fragmentation

Segmentation

semaphore

page fault

**Dijkstra's banker's algorithm: The Banker's algorithm is a resource allocation & deadlock avoidance algorithm developed by Edsger Dijkstra that**

|  |
| --- |
| Round-robin scheduling |
|  |

Need Of Paging:Start Here: <https://www.javatpoint.com/os-need-for-paging>

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.

What is Thrashing?

**Thrashing**:- can be caused by poor paging algorithms.

If the number of page faults is equal to the number of referred pages or the number of page faults are so high so that the CPU remains busy in just reading the pages from the secondary memory then the effective access time will be the time taken by the CPU to read one word from the secondary memory and it will be so high. The concept is called thrashing.

If the page fault rate is PF %, the time taken in getting a page from the secondary memory and again restarting is S (service time) and the memory access time is ma then the effective access time can be given as;

|  |
| --- |
| Thrashing can be avoided if |
| |  |  | | --- | --- | | [**A.**](javascript:%20void%200;) | the pages, belonging to the working set of the programs, are in main memory | |

**Demand Paging**

According to the concept of Virtual Memory, in order to execute some process, only a part of the process needs to be present in the main memory which means that only a few pages will only be present in the main memory at any time.

However, deciding, which pages need to be kept in the main memory and which need to be kept in the secondary memory, is going to be difficult because we cannot say in advance that a process will require a particular page at particular time.

Therefore, to overcome this problem, there is a concept called Demand Paging is introduced. It suggests keeping all pages of the frames in the secondary memory until they are required. In other words, it says that do not load any page in the main memory until it is required.

# **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).

1. The basic reason behind the occurrences of internal and external fragmentation is that internal fragmentation occurs when memory is partitioned into **fixed-sized blocks** whereas external fragmentation occurs when memory is partitioned into **variable size blocks(Dynamically)**.
2. When the memory block allotted to the process comes out to be slightly larger than requested memory, then the free space left in the allotted memory block causes internal fragmentation. On the other hands, when the process is removed from the memory it creates free space causing a hole in the memory which is called external fragmentation.
3. The problem of internal fragmentation can be solved by partitioning the memory into variable sized blocks and assign the **best fit block** to the requesting process. However, the solution for external fragmentation is compaction, but it is expensive to implement, so the processes must be allowed to acquire physical memory in a non-contiguous manner, to achieve this the technique of **paging and segmentation** is introduced.

# **Scheduling Algorithms**

There are various algorithms which are used by the Operating System to schedule the processes on the processor in an efficient way.

## **The Purpose of a Scheduling algorithm**

1. Maximum CPU utilization
2. Fare allocation of CPU
3. Maximum throughput
4. Minimum turnaround time
5. Minimum waiting time
6. Minimum response time

There are the following algorithms which can be used to schedule the jobs.

### 1. First Come First Serve(Arrival time matter otherwise continue upto terminate).

It is the simplest algorithm to implement. The process with the **minimal arrival time** will get the CPU first. The lesser the arrival time, the sooner will the process gets the CPU. It is the non-preemptive type of scheduling.

### 2. Round Robin

In the Round Robin scheduling algorithm, the OS defines a time quantum (slice). All the processes will get executed in the cyclic way. Each of the process will get the CPU for a small amount of time (called time quantum) and then get back to the ready queue to wait for its next turn. **It is a preemptive type of scheduling.**

### 3. Shortest Job First(Arrival time matter otherwise continue upto terminate)

The job with the **shortest burst time** will get the CPU first. The lesser the burst time, the sooner will the process get the CPU**. It is the non-preemptive type of scheduling**.

### 4. Shortest remaining time first(Arrival time matter and check burst time after every 1 time slot)

It is the preemptive form of SJF. In this algorithm, the OS schedules the Job according to the remaining time of the execution.

### 5. Priority based scheduling

Priority scheduling is a **non-preemptive algorithm.**In this algorithm, the priority will be assigned to each of the processes. The higher the priority, the sooner will the process get the CPU. If the priority of the two processes is same then they will be scheduled according to their arrival time.

### 6. Highest Response Ratio Next

In this scheduling Algorithm, the process with highest response ratio will be scheduled next. This reduces the starvation in the system.

# **Disk Scheduling**

As we know, a process needs two type of time, CPU time and IO time. For I/O, it requests the Operating system to access the disk.

However, the operating system must be fare enough to satisfy each request and at the same time, operating system must maintain the efficiency and speed of process execution.

The technique that operating system uses to determine the request which is to be satisfied next is called disk scheduling.

Let's discuss some important terms related to disk scheduling.

### Seek Time

Seek time is the time taken in locating the disk arm to a specified track where the read/write request will be satisfied.

### Rotational Latency

It is the time taken by the desired sector to rotate itself to the position from where it can access the R/W heads.

### Transfer Time

It is the time taken to transfer the data.

### Disk Access Time

Disk access time is given as,

Disk Access Time = Rotational Latency + Seek Time + Transfer Time

### Disk Response Time

It is the average of time spent by each request waiting for the IO operation.

### Purpose of Disk Scheduling

The main purpose of disk scheduling algorithm is to select a disk request from the queue of IO requests and decide the schedule when this request will be processed.

Let us first understand the 2 modes in which a program executes.

* User mode
* Kernel mode

When a program is executing in user mode, it is not in privileged mode. So whenever it needs any hardware resource like RAM or printer, it needs to make a call to the kernel and this is known as **SYSTEM CALL**.

When a program is executing in kernel mode, it is executing in privileged mode. So it can access any hardware resource. So when a program needs to access any resource while it is running in user mode it makes a System Call to the kernel then a context switch occurs which takes the program from user mode to kernel mode. After the resource is accessed one more context switch occurs which takes back the program’s execution to user mode.

Now you may wonder why aren’t all programs occurring in kernel mode so we can skip the context switching. This is because if a program crashes in kernel mode entire system will be halted. So most programs are executed in user mode because if it crashes there, entire system won’t be affected.

Starvation is a condition where a process does not get the resources for a long time because the resources are being allocated to other processes. It generally occurs in a Priority based scheduling System.

**In Deadlock the resources are blocked by the process whereas, in starvation, the resources are continuously being used by the processes with high priorities. On the other hand, Starvation can be prevented by aging.**

**Long term Scheduler:job of it is to select the next process from job queue and put it into Ready Queue**

**Short term Scheduler:job of it is to select the next process from ready queue and assign a CPU to it.**

What is **dispatcher** OS? When the scheduler completes its job of selecting a process, it is the dispatcher which takes that process to the desired state/queue. **The dispatcher is the module that gives a process control over the CPU after it has been selected by the short-term scheduler.** This function involves the following: Switching context  
Shedulaer: <https://www.javatpoint.com/os-process-schedulers>

**Race Condition:**

Race Condition occur when two or more processes try to access shared resources (memory) concurrently.

**Critical** **section**.  
In concurrent programming, concurrent accesses of resources can lead to unexpected behavior, so parts of the program where the shared resource is accessed are protected. This protected section is the critical section or critical region.  
Avoiding Race Conditions: Critical Section: To avoid race condition we need **Mutual Exclusion**. Mutual Exclusion is some way of making sure that if one process is using a shared variable or file, the other processes will be excluded from doing the same things.

**Memory Management:**

The **degree of multiprogramming** describes the maximum number of processes that a single-processor system can accommodate efficiently. The primary factor affecting the **degree of multiprogramming** is the amount of memory available to be allocated to executing processes.

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 fixed partitioning,

1. The partitions cannot overlap.
2. 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.

**Degree of multiprogramming is less**

The **degree of multiprogramming** describes the maximum number of processes that a single-processor system can accommodate efficiently so it is less

**Dynamic partitioning**

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.

Disadvantage of Dymanic:

1.external fragmentation

Absence of internal fragmentation doesn't mean that there will not be 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.

# **Compaction**

We got to know that the dynamic partitioning suffers from external fragmentation. However, this can cause some serious problems.

By applying this technique, we can store the bigger processes in the memory. The free partitions are merged which can now be allocated according to the needs of new processes. This technique is also called defragmentation.

Segmentation:

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 segment which can be allocated to a process.

The details about each segment are stored in a table called as segment table. Segment table is stored in one (or many) of the segments. Segmentation is slower than paging

Spooling is a process in which data is temporarily held to be used and executed by a device, program or the system. Data is sent to and stored in memory or other volatile storage until the program or computer requests it for execution.

"Spool" is technically for simultaneous peripheral operations online.

For example, in printer spooling, the documents/files that are sent to the printer are first stored in the memory or printer spooler. Once the printer is ready, it fetches the data from that spool and prints it.

Print **spooler**. A software program in Microsoft Windows that is responsible for managing all print jobs currently being sent to the computer printer

Types of locality:

spatial locality: Programs tend to make memory accesses that are in proximity of previous access this is called

temporal locality: Which term means to access some data and instructions repeatedly?

reference locality: Instructions tend to be accessed sequentially.

access locality:

MCQ

Loading an OS from secondary memory to primary memory:- Booting

Moving process from main memory to disk is called :-Swapping

Windows is :-Pre-emptive

 The page replacement policy that suffers from belady's anamoly is :-FIFO

Cache memory is: High speed RAM

Where does the swap space residable:- Disk

Which of the following scheduler reduces the degree of multiprogramming? :- **Mid term scheduler**

13.  How many primary partitions can exist on one drive?:-4 partition

What command do you use to create Linux file systems?:mkfs

Which of the following command can you execute to count the number of lines in a file?:-wc –I

What command is used to display the characteristics of a process?:ps

What command is not used to list the files chap01, chap02 and chap04?: ls chap[124]

What does GNU stand for? GNU&#39;s not Unix

What shell&#39;s wild-card is used to match any number of characters including none?:-\*

Imp mcq(read all module):- <http://www.brightways.org/ugc-net-gate-mcq-operating-system.php?page=8#!>

Use the kill command followed by the pid to terminate that process. To terminate all process at once, use kill 0.

To calculate the size of a folder uses the command **du –sh folder1.**

The pwd prints the path of the working directory, starting from the root

Used to make typescript or record all the terminal activities **script command** in Linux is used to make typescript or record all the terminal activities. After executing the **script command** it starts recording everything printed on the screen including the inputs and outputs until exit.

Increasing the RAM of a computer typically improves performance because: Fewer page faults occur

Which of the following is major part of time taken when accessing data on the disk?Ans:Seek Time

Put the following disk scheduling policies results in minimum amount of head movement.  
**(A)** FCFS  
**(B)** Circular SCAN-No Starvation  
**(C)** Elevator

D)SSTF-Starvation

Ans: Circular SCAN has more head movement than SCAN (elevator) because Circular SCAN has circular jump and it does count as a head movement.  
**SCAN (elevator)** is the best choice here.

**Shortest Job first has the advantage of having a minimum average waiting time among all scheduling algorithms.**

**A process executes the following code**

for (i = 0; i < n; i++) fork();

Ans: 2^n – 1

Group I Group II

(P) Gang Scheduling (1) Guaranteed Scheduling

(Q) Rate Monotonic Scheduling (2) Real-time Scheduling

(R) Fair Share Scheduling (3) Thread Scheduling

Ans:-P – 3 Q – 2 R – 1

Swap space is typically used to store process data.Swap space created in HardDissk/Disk

Semaphore can be used for solving \_\_\_\_\_\_\_\_\_\_.3 Synchronization

SJRF Problem:-<https://www.geeksforgeeks.org/cpu-scheduling-in-operating-systems/>

The monitor is one of the ways to achieve Process synchronization.

Direct memory access (**DMA**) is a method that allows an input/output (I/O) device to send or receive data directly to or from the main memory, bypassing the CPU to speed up memory **operations**. The process is managed by a chip known as a **DMA** controller (DMAC).

In computing, traditionally **cycle stealing** is a method of accessing computer memory (RAM) or bus without interfering with the CPU. It is similar to direct memory access (DMA) for allowing I/O controllers to read or write RAM without CPU intervention.