CPU Scheduling Algorithm:

CPU Scheduling deals with the problem of deciding which of the processes in the ready queue is to be allocated first to the CPU. There are four types of CPU scheduling that exist.

1. **First Come, First Served Scheduling (FCFS) Algorithm:**This is the simplest CPU scheduling algorithm. In this scheme, the process which requests the CPU first, that is allocated to the CPU first. The implementation of the FCFS algorithm is easily managed with a FIFO queue. When a process enters the ready queue its PCB is linked onto the rear of the queue. The average waiting time under FCFS policy is quiet long. Consider the following example:

Process	**CPU time**
P_1	3
P_2	5
P_3	2
P_4	4

Using FCFS algorithm find the average waiting time and average turnaround time if the order is P_1, P_2, P_3, P_4.

Solution: If the process arrived in the order P_1, P_2, P_3, P_4 then according to the FCFS the Gantt chart will be:

P_1	P_2	P_3	P_4

0 3 8 10 14

The waiting time for process $P_1 = 0$, $P_2 = 3$, $P_3 = 8$, $P_4 = 10$ then the turnaround time for process $P_1 = 0 + 3 = 3$, $P_2 = 3 + 5 = 8$, $P_3 = 8 + 2 = 10$, $P_4 = 10 + 4 = 14$.

Then average waiting time = $(0 + 3 + 8 + 10)/4 = 21/4 = 5.25$ Average turnaround time = $(3 + 8 + 10 + 14)/4 = 35/4 = 8.75$

The FCFS algorithm is non preemptive means once the CPU has been allocated to a process then the process keeps the CPU until the release the CPU either by terminating or requesting I/O

2. **Shortest Job First Scheduling (SJF) Algorithm:** This algorithm associates with each process if the CPU is available. This scheduling is also known as shortest next CPU burst, because the scheduling is done by examining the length of the next CPU burst of the process rather than its total length. Consider the following example:

Process	**CPU time**
P_1	3
P_2	5
P_3	2
P4	4

Solution: According to the SJF the Gantt chart will be

P_3	P_1	P_2	P_4

0 2 5 9 14

The waiting time for process $P_1 = 0$, $P_2 = 2$, $P_3 = 5$, $P_4 = 9$ then the turnaround time for process $P_3 = 0 + 2 = 2$, $P_1 = 2 + 3 = 5$, $P_4 = 5 + 4 = 9$, $P_2 = 9 + 5 = 14$.

Then average waiting time $= (0 + 2 + 5 + 9)/4 = 16/4 = 4$ Average turnaround time $= (2 + 5 + 9 + 14)/4 = 30/4 = 7.5$

The SJF algorithm may be either preemptive or non preemptive algorithm. The preemptive SJF is also known as shortest remaining time first.

Consider the following example.

Process	**Arrival Time**	**CPU time**
P_1	0	8
P_2	1	4
P_3	2	9
P_4	3	5

In this case the Gantt chart will be

P_1	P_2	P_4	P_1	P_3

0 1 5 10 17 26

The waiting time for process $P_1 = 10 - 1$

$= 9$

$P_2 = 1 – 1 = 0$

$P_3 = 17 – 2 = 15$

$P_4 = 5 – 3 = 2$

The average waiting time $= (9 + 0 + 15 + 2)/4 = 26/4 = 6.5$

3. **Priority Scheduling Algorithm:** In this scheduling a priority is associated with each process and the CPU is allocated to the process with the highest priority. Equal priority processes are scheduled in FCFS manner. Consider the following example:

Process	**Arrival Time**	**CPU time**
P_1	10	3
P_2	1	1
P_3	2	3

P_4	1	4
P_5	5	2

According to the priority scheduling the Gantt chart will be

P_2	P_5	P_1	P_3	P_4

0 1 6 16 18 19

The waiting time for process $P_1 = 6$

$P_2 = 0$

$P_3 = 16$

$P_4 = 18$

$P_4 = 1$

The average waiting time = (0 + 1 + 6 + 16 + 18)/5 = 41/5 = 8.2

4. **Round Robin Scheduling Algorithm:** This type of algorithm is designed only for the time sharing system. It is similar to FCFS scheduling with preemption condition to switch between processes. A small unit of time called quantum time or time slice is used to switch between the processes. The average waiting time under the round robin policy is quiet long. Consider the following example:

Process	**CPU time**
P_1	3
P_2	5
P_3	2
P_4	4

Time Slice = 1 millisecond.

P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4	P_1	P_2	P_4	P_2	P_4	P_2

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

The waiting time for process

$P_1 = 0 + (4 – 1) + (8 – 5) = 0 + 3 + 3 = 6$

$P_2 = 1 + (5 – 2) + (9 – 6) + (11 – 10) + (12 – 11) + (13 – 12) = 1 + 3 + 3 + 1 + 1 + 1 = 10$

$P_3 = 2 + (6 – 3) = 2 + 3 = 5$

$P_4 = 3 + (7 – 4) + (10 – 8) + (12 – 11) = 3 + 3 + 2 + 1 = 9$

The average waiting time = (6 + 10 + 5 + 9)/4 = 7.5

Operation on Processes

Several operations are possible on the process. Process must be created and deleted dynamically. Operating system must provide the environment for the process operation. We discuss the two main operations on processes.

1. Create a process

2. Terminate a process

- **Create Process**

Operating system creates a new process with the specified or default attributes and identifier. A process may create several new subprocesses. Syntax for creating new process is :

CREATE (processed, attributes)

Two names are used in the process they are parent process and child process.

Parent process is a creating process. Child process is created by the parent process. Child process may create another subprocess. So it forms a tree of processes. When operating system issues a CREATE system call, it obtains a new process control block from the pool of free memory, fills the fields with provided and default parameters, and insert the PCB into the ready list. Thus it makes the specified process eligible to run the process.

When a process is created, it requires some parameters. These are priority, level of privilege, requirement of memory, access right, memory protection information etc. Process will need certain resources, such as CPU time, memory, files and I/O devices to complete the operation. When process creates a subprocess, that subprocess may obtain its resources directly from the operating system. Otherwise it uses the resources of parent process.

When a process creates a new process, two possibilities exist in terms of execution.

1. The parent continues to execute concurrently with its children.

2. The parent waits until some or all of its children have terminated.

For address space, two possibilities occur:

1. The child process is a duplicate of the parent process.

2. The child process has a program loaded into it.

- **Terminate a Process**