**Operating System 19115045**

**Assignment-1 Kunal Sachdeva**

**5th Semester CSE**

**Topic: What are the mechanisms to evaluate an algorithm related to CPU scheduling? Discuss any one of them.**

**Solution:**

CPU Scheduling is the process of determining which process will have exclusive use of the CPU while another is paused. The basic goal of CPU scheduling is to ensure that whenever the CPU is idle, the OS chooses at least one of the programmes in the ready queue to run. The CPU scheduler will be in charge of the selecting process. It chooses from among the processes in memory that are ready to run.

A Process Scheduler assigns distinct tasks to the CPU according to specific scheduling methods. There are eight widely used process scheduling algorithms:

* First Come First Serve (FCFS)
* Shortest Job First (SJF)
* Longest Job First (LJF)
* Shortest Remaining Time First (SRTF)
* Longest Remaining Time First (LRTF)
* Round Robin Scheduling
* Priority Based scheduling (non-pre-emptive)
* Highest Response Ratio Next (HRRN)

**Round Robin** is a CPU scheduling system in which each process is cyclically assigned a set time slot.

Round-Robin Scheduling has the following crucial characteristics:

* Round robin is a pre-emptive algorithm
* After a defined interval of time, the CPU is transferred to the next process, which is known as time quantum/time slice.
* The pre-empted process gets moved to the end of the queue.
* The round robin model is a clock-driven hybrid model.
* A minimum time slice should be set to a specific task that needs to be processed. It may, however, differ from one OS to the next.
* It's a real-time algorithm that reacts to an event within a set amount of time.
* Round robin is one of the most popular, fair, and simple algorithms.
* In classic operating systems, this is a widely used scheduling mechanism.

It is the simplest and oldest scheduling method, and it is primarily used for multitasking. A round-robin scheduler uses time-sharing to ensure that processes are scheduled equitably, assigning each job a time slot or quantum (its allocation of CPU time) and interrupting the job if it is not completed by that time. When a time slot is assigned to that process, the job is resumed. The scheduler chooses the first process in the ready queue to execute if the process terminates or changes its state to waiting during its allocated time quantum. In the absence of time-sharing, or if the quanta were enormous in comparison to the work sizes, a process that created large jobs would be preferred.

The scheduler forces the process out of the CPU after the time quota expires, making the round-robin method a pre-emptive algorithm. Round-robin scheduling allows each ready task to run in a cyclic queue for a set amount of time. This technique also allows for process execution without starvation.

For example, If the time slot is 100 milliseconds and job1 takes 250 milliseconds to finish, the round-robin scheduler will suspend the job after 100 milliseconds and offer other jobs CPU time. Job1 will receive another allocation of CPU time when the other jobs have received their fair share (100 ms apiece), and the cycle will resume. This process continues until the work is completed and no more CPU time is required.

**Advantage of Round-robin Scheduling**

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

1. It is not affected by famine or the convoy effect.
2. Each work is given an equal amount of CPU time.
3. It handles all processes without regard to priority.
4. You can estimate the worst-case response time for the same process if you know the total number of processes on the run queue.
5. This type of scheduling is independent of burst time. As a result, it is simple to integrate into the system.
6. After a process has run for a certain amount of time, it is pre-empted, and another process runs for the same amount of time.
7. Allows the OS to save the states of pre-empted processes using the Context switching mechanism.
8. In terms of average response time, it performs the best.

**Disadvantages of Round-robin Scheduling**

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

1. The CPU output will be lowered if the OS slicing time is short.
2. This technique spends more time transitioning between contexts.
3. Its performance is mostly determined by the time quantum.
4. Processes cannot have priorities defined for them.
5. Round-robin scheduling does not give more critical work additional consideration.
6. Reduces understanding
7. The context switching overhead in the system increases as the time quantum decreases.
8. In this system, finding a right time quantum is a challenging task.

**Worst Case Latency**

This word refers to the total amount of time it takes to complete all of the jobs.

dt = When a task is added to the list, the detection time is noted.

st = Indicate the amount of time it takes to switch from one task to another.

et = Indicate the time it takes to complete a task.

Formula:

T worst = {(dti+ sti + eti ), + (dti+ sti + eti )2 +...+ (dti+ sti + eti )N., + (dti+ sti + eti + eti) N} + tISR

t, SR = sum of all execution times

**Summary:**

* The round-robin principle, in which each person gets an equal amount of something in turn, inspired the name of this method.
* Round robin is one of the oldest, fairest, and most extensively utilised scheduling algorithms in traditional operating systems.
* The round robin algorithm is a forewarning algorithm.
* The most significant benefit of the round-robin scheduling strategy is that if you know the total number of processes in the run queue, you can also guess the worst-case response time for each of them.
* This technique spends more time transitioning between contexts.
* The term "worst-case latency" refers to the total amount of time it takes to complete all of the processes.