International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 10 Oct. 2022, pp: 399-400 www.ijaem.net ISSN: 2395-5252

Operating System

Date of Submission: 01-10-2022 Date of Acceptance: 12-10-2022

PROCESS SCHEDULING:

CPU is always busy in **Multiprogramming**. Because CPU switches from one job to another job. But in **simple computers** CPU sit idle until the I/O request granted. **scheduling** is a important OS function. All resources are scheduled before use.(cpu, memory, devices etc)

Process scheduling is an essential part of a Multiprogramming operating systems. Such operating systems allow more than one process to be loaded into the executable memory at a time and the loaded process shares the CPU using time multiplexing.

Scheduling Objectives

Scheduling objectives:

1. **CPU utilization** – keep the CPU as busy as possible
2. **Throughput** – Number of processes that complete their execution per time unit
3. **Turnaround time** – amount of time to execute a particular process
4. **Waiting time** – amount of time a process has been waiting in the ready queue
5. **Response time** – amount of time it takes from when a request was submitted until the first response is produced, **not** output (for time-sharing environment)

General Goals

Fairness

Fairness is important under all circumstances. A scheduler makes sure that each process gets its fair share of the CPU and no process can suffer indefinite postponement.

Policy Enforcement

The scheduler has to make sure that system's policy is enforced. For example, if the local policy is safety then the safety control processes must be able to run whenever they want to, even if it means delay in payroll processes.

Efficiency

Scheduler should keep the system (or in particular CPU) busy cent percent of the time when possible. If the CPU and all the Input/Output devices can be kept running all the time, more work gets done per second than if some components are idle.

Response Time

A scheduler should minimize the response time for interactive user.

Turnaround

A scheduler should minimize the time batch users must wait for an output.

Throughput

A scheduler should maximize the number of jobs processed per unit time.

A little thought will show that some of these goals are contradictory. It can be shown that any scheduling algorithm that favors some class of jobs

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 10 Oct. 2022, pp: 399-400 www.ijaem.net ISSN: 2395-5252

hurts another class of jobs. The amount of CPU time available is finite, after all.

Preemptive Vs Non-preemptive Scheduling

The Scheduling algorithms can be divided into two categories with respect to how they deal with clock interrupts.

Non-preemptive Scheduling

A scheduling discipline is non-preemptive if, once a process has been given the CPU, the CPU cannot be taken away from that process.