**PROJECT**

**PROJECT TITLE:** Process Scheduling Algorithms in OS

**Introduction:**

In an operating system (OS), process scheduling is a critical function that manages the execution of processes on a CPU. The goal of process scheduling is to allocate the system's processing power efficiently among various processes, ensuring that each process gets executed in a fair and optimal manner. By using scheduling algorithms, the OS determines the order in which processes are executed based on factors like priority, execution time, and resource requirements. The effectiveness of scheduling algorithms directly impacts system performance, resource utilization, and user experience.

**Problem:**

* Inefficient Resource Utilization: Without proper scheduling algorithms, an OS may lead to CPU underutilization or overuse, leading to performance bottlenecks and inefficiencies.
* Process Starvation: Some processes may never get executed if they are always pre-empted by higher-priority processes, leading to "starvation" or delays in process completion.
* Fairness Issues: A lack of a good scheduling strategy can cause certain processes to monopolize system resources, making it difficult for others to get timely CPU access.

**Solution:**

The solution lies in the implementation of efficient process scheduling algorithms that balance CPU utilization, fairness, and process prioritization. Different types of scheduling algorithms, such as:

* First-Come, First-Served (FCFS): Processes are executed in the order they arrive.
* Shortest Job Next (SJN): Prioritizes processes with the shortest execution times to minimize waiting time.
* Round Robin (RR): Allocates equal CPU time to each process in a circular manner, ensuring fairness.
* Priority Scheduling: Assigns priorities to processes, executing higher-priority processes first.
* Multilevel Queue Scheduling: Organizes processes into different queues based on their priority or type, optimizing resource allocation.
* These algorithms will be implemented and compared to determine their effectiveness in minimizing wait time, CPU utilization, and ensuring fairness.

**Objectives:**

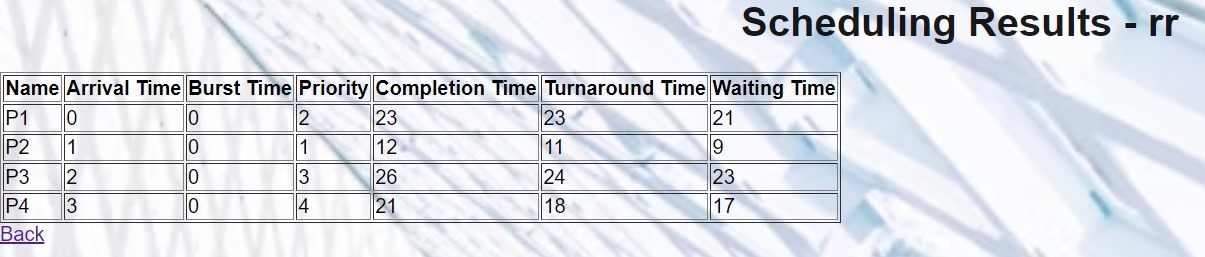
* Understand Different Scheduling Algorithms: Research and analyse various CPU scheduling algorithms to understand their strengths, weaknesses, and applicable scenarios.
* Implement Algorithms: Develop and implement multiple process scheduling algorithms (FCFS, SJN, RR, Priority Scheduling, etc.) in an OS simulation.
* Comparison and Analysis: Compare the performance of different scheduling algorithms in terms of CPU utilization, wait time, turnaround time, and fairness.
* Optimization: Investigate the possibility of optimizing algorithms for better performance based on system requirements (e.g., maximizing throughput or reducing latency).
* Fairness and Resource Allocation: Ensure that the selected algorithms provide fair resource allocation without causing process starvation.
* Simulation and Testing: Develop a simulation environment to test the algorithms with different workloads and input conditions to measure their performance.

**Technologies:**

* Front-end: HTML, CSS, JavaScript, Flask (Python web framework)
* Back-end: Python, Flask
* DSA: For storing data

**PHOTOS :**

****

****