Name: JATIN
Enrollment no. : 21113068

CPU SCHEDULER

1.Introduction

The objective of the "CPU Scheduler" project is to create and deploy an effective CPU scheduling algorithm aimed at optimizing the distribution of CPU resources among multiple processes within a computer system.
The project enriches comprehension of operating system process management, offers practical coding experience with scheduling algorithms, and enhances problem-solving abilities in optimizing resource allocation and system performance.
The implemented algorithms are First Come, First Serve (FCFS), Shortest Job First (SJF), Priority Scheduling, and Round Robin (RR).

2. Algorithms

- FCFS

FCFS is a non-preemptive scheduling algorithm where processes are executed in the order they arrive. Once a process starts executing, it continues until it is complete. If two processes have the same arrival time, the one that appears first in the input is executed first. Finish time is calculated during execution, and Turnaround Time and Wait Time are derived from it.
FCFS is suitable for batch processing where all jobs are known in advance, and equal priority is given to all processes.

- SJF

SJF is a non-preemptive or preemptive scheduling algorithm where the process with the shortest execution time is selected for execution next.
SJF is suitable for environments where burst times are known in advance or can be estimated accurately, such as interactive systems or when processes arrive at random intervals.

- Priority Scheduling

Priority scheduling is a scheduling algorithm where each process is assigned a priority. The scheduler selects the process with the highest priority for execution.
Priority scheduling is used in real-time systems where certain processes require immediate attention or when there are specific service level agreements (SLAs) to be met.

- Round Robin

  Round Robin is a preemptive scheduling algorithm where each process is assigned a fixed time slice or quantum. The scheduler cycles through ready processes, allocating each one the CPU for the specified time.
  Round Robin is commonly used in time-sharing systems, interactive systems, and environments where fairness and responsiveness are prioritized.

3. Implementation

Obtain the project code by cloning the repository from its GitHub location given below ➖
Link: https://github.com/JATIN-IITR/CPU-Scheduler

4. Learning Takeaways

Throughout this project, several key insights and skills were acquired:

- Comprehensive Understanding of Scheduling Algorithms: Developed a thorough grasp of different scheduling algorithms, including their strengths and weaknesses.

- Significance of Scheduling Terms and Selection Criteria: Recognized the importance of scheduling terms and criteria in determining the most suitable scheduler for specific scenarios.

- Critical Role of Schedulers in Operating Systems: Appreciated the pivotal role of schedulers in optimizing task execution and resource allocation within an operating system.

- Enhanced C++ Coding Proficiency: Improved skills in writing structured and efficient C++ code tailored to implement specific algorithms effectively.

- Integration of Front-end and Back-end Systems: Acquired knowledge in integrating front-end interfaces with back-end systems, facilitating the display of back-end data to users.

These insights and skills collectively contribute to a deeper understanding of system optimization through CPU scheduling algorithms and enhance proficiency in software development practices.

5. Potential applications

- Operating Systems Enhancement:-

Operating system efficiency can be improved by integrating advanced scheduling algorithms to manage concurrent processes more effectively.

- Embedded Systems Efficiency:-

Lightweight scheduling algorithms can be implemented in embedded systems to ensure efficient task management and resource allocation.

- Industrial Automation Advancements:-

Deterministic scheduling policies can be implemented in industrial automation systems to ensure the timely execution of critical tasks and reduce latency.

6. Future work

- Algorithm Optimization:

Further refining existing scheduling algorithms (FCFS, SJF, Priority Scheduling, Round Robin) to enhance performance metrics such as turnaround time and CPU utilization.

- Advanced Scheduling Policies:

Exploration and implementing advanced policies like Multilevel Feedback Queue (MLFQ) and Shortest Remaining Time (SRT) to handle diverse workload scenarios more effectively.

- Real-time Scheduling Enhancements:

Develop capabilities for real-time scheduling to support time-sensitive applications and ensure predictable task completion times.

7. References

- https://www.geeksforgeeks.org/cpu-scheduling-in-operating-systems/
- https://www.youtube.com/playlistlist=PLBlnK6fEyqRitWSE_AyyySWfhRgyA-rHk
- Operating System Concepts by Silberschatz, Galvin, and Gagne