Multilevel queue scheduling

Assignment-8

W.A.K.P.A Wanniarachchi
22002121

Context:

The provided code presents an implementation of various CPU scheduling algorithms in the context of operating systems. CPU scheduling is a crucial aspect of operating system design, responsible for efficiently allocating CPU resources to processes competing for execution. Different scheduling algorithms prioritize processes differently, aiming to optimize various performance metrics such as turnaround time, waiting time, and CPU utilization.

In this implementation, four common CPU scheduling algorithms are explored: Round Robin (RR), Shortest Job First (SJF) with two variations, and First In First Out (FIFO). These algorithms are fundamental components of modern operating systems and are widely used in both batch and interactive computing environments.

The code simulates the scheduling of processes with varying burst times and priorities. It employs a simple queue data structure to organize processes into different priority levels, with each algorithm operating on its designated queue. The implementation allows for the comparison of different scheduling algorithms by analyzing their respective performance metrics, such as waiting time and turnaround time, under various workload scenarios.

CPU scheduling algorithms play a crucial role in system performance and user experience. By implementing and evaluating these algorithms, developers and system administrators can gain_insights into their strengths, weaknesses, and suitability for different computing environments. This code serves as a learning tool for understanding the principles of CPU scheduling and exploring the trade-offs involved in selecting an appropriate scheduling algorithm for a given system. Additionally, it provides a foundation for further experimentation and optimization of scheduling strategies in real-world operating systems.

Objective: The objective of this document is to provide a comprehensive overview of the implementation and analysis of various CPU

scheduling algorithms within the context of operating system design. The document aims to explore different scheduling strategies, including Round Robin (RR), Shortest Job First (SJF), and First In First Out (FIFO), highlighting their respective strengths, weaknesses, and performance characteristics.

Scope Overview:

1. Introduction to CPU Scheduling:
    - Brief overview of the importance of CPU scheduling in operating system design.
    - Explanation of the role of CPU scheduling algorithms in managing process execution.
2. Implemented Algorithms:
    - Description of the four implemented CPU scheduling algorithms: RR, SJF1, SJF2, and FIFO.
    - Overview of each algorithm's principles, scheduling strategy, and key characteristics.
3. Implementation Details:
    - Overview of the code structure and organization.
    - Explanation of the data structures and variables used in the implementation.
    - Description of the input process data and how it is utilized in the scheduling algorithms.
4. Execution and Results:
    - Explanation of how the scheduling algorithms are executed and evaluated.
    - Presentation of the results, including waiting time, turnaround time, and other performance metrics.
    - Comparative analysis of the performance of each scheduling algorithm under various workload scenarios.

5. Pros and Cons of Each Algorithm:
    - Discussion of the strengths and weaknesses of each scheduling algorithm.
    - Analysis of the trade-offs involved in selecting an appropriate scheduling strategy for different system requirements and constraints.
6. Conclusion and Recommendations:
    - Summary of the findings from the analysis of the implemented scheduling algorithms.
    - Recommendations for selecting an optimal scheduling algorithm based on specific system characteristics and workload patterns.
    - Suggestions for further experimentation and optimization of CPU scheduling strategies.

Objective	Demonstrate multilevel queue scheduling using four queues
Lines	300
Programming language	C++

Implementation Overview:

The provided program implements four different CPU scheduling algorithms:

- Round Robin (RR)
- Shortest Job First (SJF)
- First In First Out (FIFO)

These algorithms are implemented using a simple queue data structure, with each queue representing a different priority level.

Results of each scheduling algorithm:

1. **Round Robin (RR):**
    - This algorithm allocates a fixed slice (T) to each process in a cyclic manner.
    - Results are displayed with process IDs, waiting time, and tourn around time.
    - The program calculates waiting time and tourn around time for each process.
2. **Shortest Job First (SJF):**
    - Implemented in two variations (SJF1 and SJF2), sorting processes based on their burst time.
    - Results include waiting time for each process.
3. **First In First Out (FIFO):**
    - Processes are executed in the order they arrive in the system.
    - Results include waiting time and tourn around time for each process.

Pros and Cons of each scheduling algorithm:

1. **Round Robin (RR):**
   - Pros:
     - Provide fair CPU time allocation.
     - Suitable for time-sharing systems.
   - Cons:
     - May result in higher waiting times for longer processes.
     - High context-switching overhead with small time slices.
2. **Shortest Job First (SJF)**:
   - Pros:
     - Minimizes average waiting time by prioritizing shorter jobs.
     - Efficient for minimizing turnaround time.
   - Cons:
     - Requires knowledge of burst times, which may not be available.
     - Can lead to starvation for longer processes.
3. **First In First Out (FIFO)**:
   - Pros:
     - Simple and easy to implement.
     - Suitable for non-preemptive environments.
   - Cons:
     - May result in longer average waiting times for longer processes.
     - Can lead to convoy effect, where short processes are delayed by long ones.

Brief Conclusion:

The analysis of the implemented CPU scheduling algorithms demonstrates that each algorithm has its strengths and weaknesses. Round Robin offers fairness but may not be optimal for minimizing waiting times. SJF is efficient for minimizing

waiting times but may lead to starvation. FIFO is simple but may not optimize waiting times efficiently. The choice of scheduling algorithm depends on the specific requirements and constraints of the system.

Limitations of the Program:

1. **Simplistic Model**: The program implements basic versions of scheduling algorithms without considering various real-world complexities such as process arrival patterns, I/O bursts, and priority aging.
2. **Lack of Visualization**: Results are displayed in a textual format, which may make it difficult to analyze performance intuitively.
3. **Limited Scalability**: The program is designed for a small number of processes and may not scale well for larger systems.

In conclusion, while the implemented program provides insights into different CPU scheduling algorithms, further enhancements and considerations are necessary to address real-world complexities and scalability issues.