

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

# OPERATING SYSTEMS - CS235AI

**REPORT**

# Submitted by

**PALLAVI BG 1RV22CS134**

**SANJANA S 1RV22CS177**

**Computer Science and Engineering 2023-2024**

**INTRODUCTION**

~CPU scheduling in operating systems is a crucial component that manages the allocation of CPU resources among processes.

~The primary goal of CPU scheduling is to maximise CPU utilisation, ensure fairness among processes, and optimise system performance.

~When a process is ready to execute, it is placed in a queue called the ready queue. The operating system uses scheduling algorithms to select the next process from the ready queue for execution.

~Common scheduling algorithms include First-Come, First-Served (FCFS), Shortest Job Next (SJN) or Shortest Job First (SJF), Priority Scheduling, Round Robin (RR), and Multilevel Queue Scheduling.

~The selected process is then dispatched, and its context is loaded into the CPU for execution. Context switching occurs when the CPU switches from executing one process to another, involving saving the current process's state and loading the state of the new process.

~CPU scheduling algorithms are evaluated based on criteria such as CPU utilisation, throughput, turnaround time, waiting time, and response time, with the aim of optimising system performance and resource utilisation.

**PROBLEM STATEMENT**

* In a computer system with multiple processes competing for CPU time, the CPU scheduling algorithm plays a critical role in determining the order in which processes are executed.
* The goal of CPU scheduling is to maximize CPU utilization, ensure fairness among processes, minimize response time, and optimize system throughput.
* CPU scheduling algorithms are evaluated based on criteria such as CPU utilization, throughput, turnaround time, waiting time, and response time, with the aim of optimizing system performance and resource utilization
* However, designing an efficient CPU scheduling algorithm is challenging due to the trade-offs between various scheduling criteria such as throughput, turnaround time, waiting time, and response time.
* Therefore, the problem statement is to develop a CPU scheduling algorithm that balances these criteria to improve system performance and resource utilization.

**SYSTEM ARCHITECTURE**

**METHODOLOGY**

1. Initialization: Set up the initial frame values and counters.
2. Iteration: Iterate through the pages, checking for existing values in the frames.
3. Handling Hits and Faults: Manage page hits and faults based on whether the page is already in the frames or needs replacement
4. Output Display: Conclude the process by displaying the final frame array and counters.
5. This flow ensures a systematic implementation of the FIFO algorithm, visualized through a block diagram or flow chart.

**SYSTEM CALLS**

**OUTPUT**

**CONCLUSION**

CPU Scheduling is vital since it enhances the following functions:

1. Resource Utilization: Efficient CPU scheduling ensures that the CPU is utilized to its fullest potential, minimizing idle time and maximizing throughput.
2. Response Time: Effective scheduling algorithms reduce the response time for interactive tasks, improving user experience and system responsiveness.
3. Fairness: Fair CPU scheduling ensures that all processes get a fair share of CPU time, preventing starvation and ensuring equitable resource allocation.
4. Priority Management: Scheduling allows for prioritizing critical tasks over non- critical ones, ensuring that important processes are executed in a timely manner.
5. Throughput:By efficiently scheduling processes, the system can achieve higher throughput, processing more tasks in a given time period.
6. Predictability:Scheduling algorithms provide predictability in task execution, which is crucial for real-time systems and applications with strict timing requirements.
7. Resource Management: CPU scheduling plays a vital role in managing system resources, coordinating CPU access among multiple processes and preventing resource contention.
8. Multiprogramming: In a multiprogramming environment, scheduling ensures that multiple processes can run concurrently on a single CPU, maximizing system efficiency.
9. Adaptability: Modern scheduling algorithms can adapt to varying workloads and system conditions, dynamically adjusting scheduling parameters to optimize performance.
10. Overall System Performance: Effective CPU scheduling contributes significantly to the overall performance and stability of the operating system, influencing factors such as throughput, latency, and resource utilization.