Lecture#07 Scheduling Algorithms

Longest Job First (LJF) Algorithm

Introduction to CPU Scheduling

- • CPU scheduling is a fundamental concept in operating systems
- • Determines which process runs on the CPU at any given time
- • Goals: maximize CPU utilization and throughput
- • Multiple scheduling algorithms exist for different scenarios
- • Longest Job First (LJF) is one non-preemptive scheduling approach

Understanding Longest Job First

- • Non-preemptive scheduling algorithm
- • Selects process with longest burst time first
- • Burst time: estimated CPU execution time required
- • Once process begins, it runs to completion
- • Also known as Longest Process First (LPF)

LJF Characteristics

- • Prioritizes longer processes over shorter ones
- • Can lead to convoy effect
- • Not typically used in real-time systems
- • Better suited for batch processing systems
- • Maximizes CPU utilization for specific workloads

Advantages of LJF

- • Simple to implement
- • Reduces context switching overhead
- • Optimal for batch systems with known processing times
- • Maximum CPU utilization for compute-intensive tasks
- • Good for background processing systems

Disadvantages of LJF

- • Poor average waiting time
- • Can cause starvation of shorter processes
- • Not suitable for interactive systems
- • Requires accurate burst time estimation
- • Performance depends heavily on arrival patterns

Example Implementation

Process	Burst Time	Completion Time
P1	6	6 units
P2	8	14 units
P3	4	18 units
P4	2	20 units

Comparison with Other Algorithms

- • Shortest Job First (SJF) - opposite approach
- • First Come First Served (FCFS) - simpler alternative
- • Priority Scheduling - different criteria
- • Round Robin - time quantum based
- • Each serves different scheduling needs

Real-world Applications

- • Batch processing systems
- • Scientific computing environments
- • Background processing tasks
- • Data center workload management
- • High-performance computing scenarios

Best Practices and Considerations

- • Use when burst times are known in advance
- • Consider system requirements carefully
- • Monitor for process starvation
- • Implement aging mechanism if needed
- • Balance with other scheduling algorithms

Summary and Key Takeaways

- • LJF prioritizes longest jobs first
- • Non-preemptive scheduling algorithm
- • Best for batch processing
- • Consider trade-offs carefully
- • Important part of CPU scheduling knowledge

Highest Response Ratio Next (HRRN) Algorithm

Introduction to HRRN

- • CPU scheduling algorithm that prioritizes processes based on their waiting time and burst time
- • Non-preemptive scheduling technique
- • Aims to prevent starvation and provide fair process execution
- • Developed to improve upon limitations of other scheduling algorithms

HRRN Formula and Calculation

- Response Ratio = (Waiting Time + Burst Time) / Burst Time
- Where:
- W = Waiting Time
- B = Burst Time
- Higher response ratio = Higher priority

Key Components

- • Arrival Time: When process enters ready queue
- • Burst Time: CPU time required for process execution
- • Waiting Time: Time spent waiting in ready queue
- • Response Ratio: Priority calculation metric
- • Completion Time: When process finishes execution

Algorithm Steps

- 1. Calculate response ratio for all ready processes
- 2. Select process with highest response ratio
- 3. Execute process to completion (non-preemptive)
- 4. Recalculate ratios for remaining processes
- 5. Repeat until all processes complete

Advantages of HRRN

- • Prevents process starvation
- • Considers both waiting time and burst time
- • Provides better balance than SJF
- • Fair treatment of long and short processes
- • Improved average waiting time

Example Calculation

- Process P1:
- Waiting Time = 10 units
- Burst Time = 4 units
- Response Ratio = (10 + 4) / 4 = 3.5

- Process P2:
- Waiting Time = 8 units
- Burst Time = 2 units
- Response Ratio = (8 + 2) / 2 = 5.0

Implementation Considerations

- • Overhead of calculating response ratios
- • Need for accurate burst time estimates
- • System resource requirements
- • Queue management
- • Process state tracking

Comparison with Other Algorithms

- • FCFS: More fair than simple first-come-first-serve
- • SJF: Better handling of longer processes
- • Priority: Dynamic priority based on waiting time
- • Round Robin: No time quantum needed
- • Aging concept naturally implemented

Real-world Applications

- • Batch processing systems
- • Operating system schedulers
- • Job scheduling in distributed systems
- • Resource allocation in cloud computing
- • Process management in multi-user systems

END OF LECTURE!