Lecture#05 Process Scheduling

Introduction to Process Scheduling?

- Process scheduling is a fundamental operating system function:
    - In an operating system, a process represents a program in execution. Process scheduling is the method by which the OS determines which process runs at any given time.
- Manages allocation of CPU time to different processes:
    - The operating system allocates CPU time to various processes using scheduling algorithms (like First-Come, First-Served, Shortest Job Next, Round Robin, etc.).
- Ensures efficient utilization of system resources:
    - By managing resources like CPU, memory, and I/O devices, the operating system ensures that they are used effectively.
- Essential for multitasking environments:
    - Multitasking allows multiple processes to run seemingly simultaneously by rapidly switching the CPU's attention between them.

Why Process Scheduling Matters

- Enables multitasking: Process scheduling allows an operating system to switch between multiple tasks quickly, giving the illusion that all processes are running simultaneously.
- Maximizes CPU utilization: Effective process scheduling ensures that the CPU is rarely idle. By allocating CPU time to processes based on their states (e.g., ready or waiting), the operating system minimizes downtime,
- Reduces process waiting time: Scheduling algorithms prioritize tasks in a way that minimizes the amount of time processes spend waiting in the queue for execution.
- Improves system performance: By minimizing delays, optimizing resource allocation, and preventing bottlenecks, process scheduling enhances the overall efficiency and speed of the system.
- Ensures fair resource allocation: Process scheduling ensures that all processes get a fair share of resources.

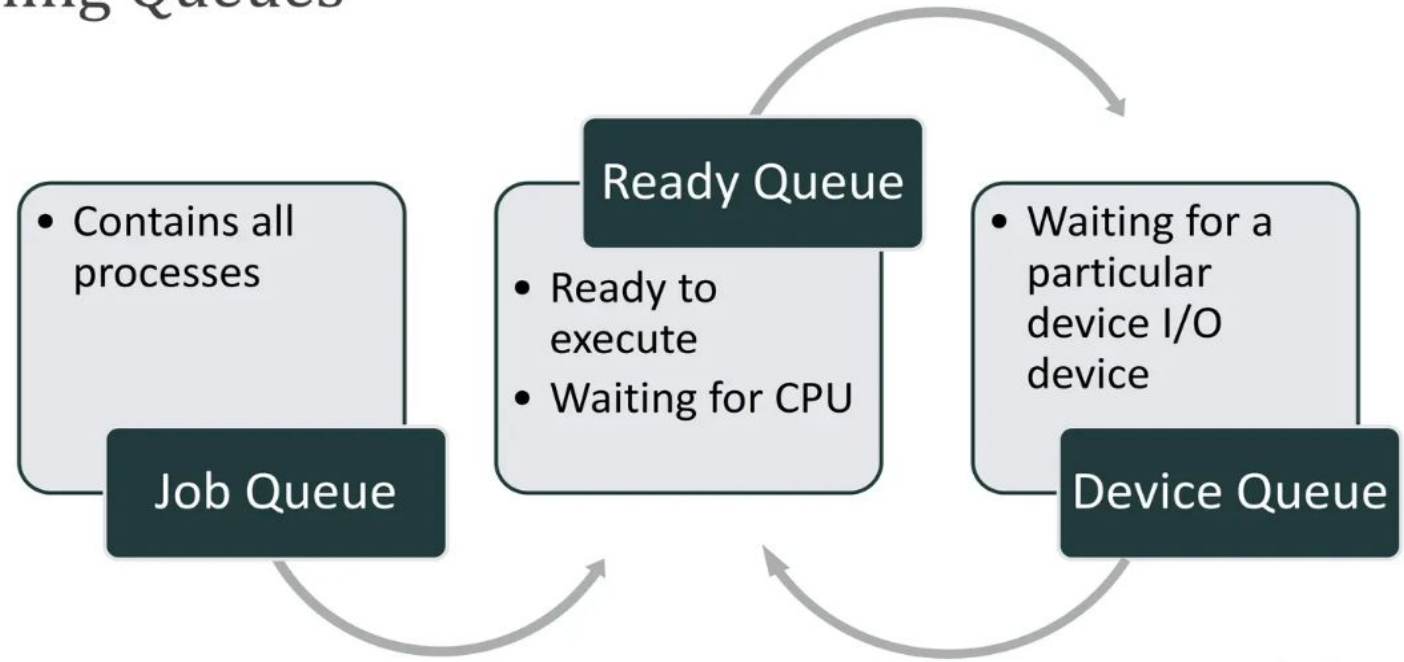
Process behavior for Scheduling

- **CPU** BURST-when a process has long computations to be executed by the processor.
- I/O BURST-The occurrence of I/O operation.
- CPU BOUND-process with intensive CPU-burst i.e., longer CPU cycles and a low number of I/O bursts
- I/O BOUND-process has a large number of frequent I/O bursts within the smaller CPU-bursts

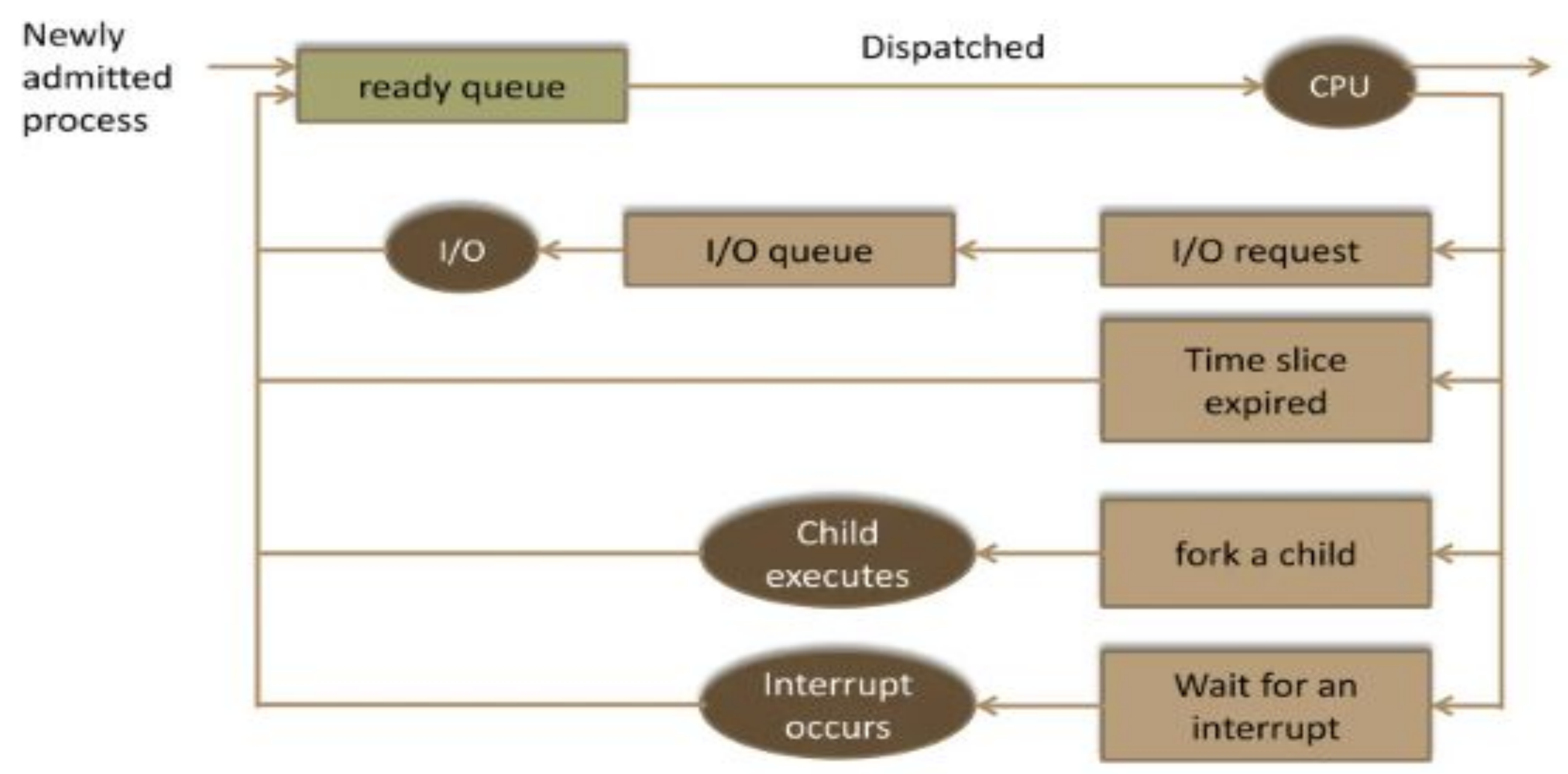
Scheduling Levels

- **Long-term** scheduling: Done when a new process is created. It initiates processes and so controls the degree of multi-programming (number of processes in memory).
- **Medium-term scheduling**: Involves suspending or resuming processes by swapping (rolling) them out of or into memory
- **Short-term scheduling**: Occurs most frequently and decides which process to execute next.

Scheduling Queues



How a Process Scheduled?



Who Schedules A Process?

- Scheduler
- Selects processes from different queues for scheduling purpose
- Types of schedulers
- Long-term Scheduler (Job Scheduler)
  - Selects processes from job pool (job queue) and loads them into main memory for execution
  - Less frequently executed
  - Controls the degree of multiprogramming (no. of processes in main memory)
- Short-term Scheduler (CPU Scheduler)
  - Selects processes from ready queue and allocates CPU to one of them.
  - Most frequently executed "PROCESS AND PROCESS SCHEDULING"

Context Switching

- When interrupt occurs, the system saves the current context of a process running on the CPU
- **Process context** is stored in the **Process Control Block (PCB).**
- Two major operations are involved in context switching:
  - State save: Storing the current state of the process before switching to another process.
  - State restore: Retrieving the saved state to resume the previous process.
- Context switch
  - **Switching the CPU to another process** requires both a **state save** of the current process and a **state restore** of a different process.
  - **Context switch time** is an **overhead**, meaning that it does not contribute to the actual execution of user programs but is necessary for multitasking.
  - **Hardware dependency**: The efficiency of context switching depends on hardware support for "PROCESS AND PROCESS SCHEDULING.

Key Scheduling Criteria: CPU Utilization

- Definition: Percentage of time CPU is actively processing:
  - A metric that measures how effectively the CPU is being used
- Goal: Keep CPU as busy as possible:
  - CPU remains engaged with tasks, minimizing idle periods.
- Typical target: 40% (IO-bound) to 90% (CPU-bound):
  - I/O-bound achieve lower CPU utilization around 40%.
  - CPU-bound, involve intensive computations reaching up to 90%.
- Measurement methods:
  - Measured through tools like system monitors, profilers, or performance analytics software
- Impact on system performance:
  - High CPU utilization correlates with better system performance.

Key Scheduling Criteria: Throughput

- Definition: Number of processes completed per time unit:
- Factors affecting throughput: Scheduling Algorithms, I/O Bound, Resources availability, Context switching
- Relationship with CPU utilization: CPU utilization often leads to increased throughput. If the CPU is overburdened, it may lead to bottlenecks, reducing throughput.
- Optimization strategies: Efficient scheduling algorithms, Reducing context switching, Implementing resource optimization techniques.
- Performance metrics: evaluate overall system performance combined with other metrics

Key Scheduling Criteria: Turnaround Time

Definition: Time from process submission to completion.

Components:

- Waiting time: Process spends time in the ready queue for CPU allocation.
- Execution time: Actual time the CPU spends executing the instructions of the process.
- I/O time: The time taken for input/output operations required by the process.

Impact on user experience

Factors Affecting Waiting Time

Scheduling algorithm: Affect the amount of time a process waits.

Process priority: High-priority processes may reduce waiting times for critical tasks while increasing them for lower-priority tasks.

System load: A high number of processes in the ready queue can increase waiting

Context switching overhead: Lead to increased waiting times.

Relationship with other metrics: Waiting time directly influences other performance metrics.

Minimization strategies: Employing efficient scheduling algorithms, Reducing context switching, Allocating resources dynamically .

Response Time

- Definition: Time from request to first response
- Critical for interactive systems: For web applications or real-time software, a quick response time is crucial .
- Systems like ATMs, customer service chatbots, and gaming servers rely heavily on minimal response times.
- User experience implications: Long response times can frustrate users.
- Measurement methods: Tools like performance monitors or system logs are often used to calculate this metric.
- Optimization techniques: Efficient scheduling algorithms, Load balancing, Resource prioritization, Reducing context switching.

END OF LECTURE!