Sunbeam Institute of Information Technology
Pune and Karad
PreCAT

Module – Operating System Concepts

Trainer - Devendra Dhande

Email – devendra.dhande@sunbeaminfo.com

Operating System – Types of Scheduler

- **Job scheduler - Long term scheduler**
  - Multi-programming system: multiple programs loaded in main memory.
    - Better CPU utilization
    - Should be "mixture" of CPU bound and IO bound processes
      - CPU bound: CPU burst > IO burst
      - IO bound: CPU burst < IO burst
  - Job scheduler load the programs into main memory.
  - Used in older mainframe systems.
- **CPU scheduler - Short term scheduler**
  - Multi-programming system: multiple programs loaded in main memory.
    - If multiple programs are CPU bound, they need to wait for current program completion (default implementation).
  - For better execution, we want to execute these programs concurrently.
  - Multi-tasking system: sharing CPU time among multiple tasks present in main memory and ready for execution.
    - CPU scheduler pick the process to be executed on CPU from ready processes.
    - Algorithm used by CPU scheduler to pick a process is called as CPU scheduling algorithm

Operating System – CPU Scheduler

- CPU Scheduler is invoked in the following four cases:
  - Case-1: Running -> Terminated
  - Case-2: Running -> Waiting
  - Case-3: Running -> Ready
  - Case-4: Waiting -> Ready

- **Types of Scheduling**
  - **Non-preemptive**
    - The current process gives up CPU voluntarily (for IO, terminate or yield).
    - Then CPU scheduler picks next process for the execution.
    - If each process yields CPU so that other process can get CPU for the execution, it is referred as "Co-operative scheduling".
  - **Preemptive**
    - The current process may give up CPU voluntarily or paused forcibly (for high priority process or upon completion of its time quantum)

Operating System – Scheduling criteria's

- **CPU utilization: Ideal - max**
  - On server systems, CPU utilization should be more than 90%.
  - On desktop systems, CPU utilization should around 70%.
- **Throughput: Ideal - max**
  - The amount of work done in unit time.
- **Waiting time: Ideal - min**
  - Time spent by the process in the ready queue to get scheduled on the CPU.
  - If waiting time is more (not getting CPU time for execution) -- Starvation.
- **Turn-around time: Ideal - CPU burst + IO burst**
  - Time from arrival of the process till completion of the process.
  - CPU burst + IO burst + (CPU) Waiting time + IO Waiting time
- **Response time: Ideal - min**
  - Time from arrival of process (in ready queue) till allocated CPU for first time.

Operating System – Scheduling Algorithms

- **FCFS**
  - Process added first in ready queue should be scheduled first.
  - Non-preemptive scheduling
    - Scheduler is invoked when process is terminated, blocked or gives up CPU is ready for execution.
  - Convoy Effect: Larger processes slow down execution of other processes.

- **SJF**
  - Process with lowest burst time is scheduled first.
  - Non-preemptive scheduling
  - Minimum waiting time

- **SRTF - Shortest Remaining Time First**
  - Similar to SJF - but Preemptive scheduling
  - Minimum waiting time

Operating System – Scheduling Algorithms

- **Priority**
  - Each process is associated with some priority level. Usually lower the number, higher is the priority.
  - Preemptive scheduling or Non Preemptive scheduling
  - Starvation
    - Problem may arise in priority scheduling.
    - Process not getting CPU time due to other high priority processes.
    - Process is in ready state (ready queue).
    - May be handled with aging -- dynamically increasing priority of the process.
- **Round-Robin**
  - Preemptive scheduling
  - Process is assigned a time quantum/slice.
  - Once time slice is completed/expired, then process is forcibly preempted and other process is scheduled.
  - Min response time.

Operating System – IPC Overview

- A process cannot access of memory of another process directly. OS provides IPC mechanisms so that processes can communicate with each other.
- IPC models
  - Shared memory model
    - Processes write/read from the memory region accessible to both the processes.
    - OS only provides access to the shared memory region.
  - Message passing model
    - Process send message to the OS and the other process receives message from the OS.
    - This is slower compared to shared memory model.
- Unix/Linux IPC mechanisms
  - Signals
  - Message queue
  - Pipe
  - Socket
  - Shared memory

Thank you!

Devendra Dhande
<devendra.dhande@sunbeaminfo.com>