Scheduling

Saroj Shivagunde

Overview

- Schedulers
- Preemptive vs. Non-preemptive Scheduling
- Scheduling algorithms
  - First-Come-First-Served (FCFS)
  - Shortest Job First (SJF)
  - Shortest Remaining Time First (SRTF)
  - Round Robin (RR)
  - Priority Scheduling

Process Scheduler

- Process scheduler decides which process to admit in the ready queue
- It is also called a long-term scheduler as it is used only to bring a process into the CPU
- It needs to be invoked less frequently for the same reason, hence, it is slower than the CPU scheduler
- The process scheduler decides how many processes to be admitted to the CPU
  - This is called as the ‘degree of multi-programming’
- The admission of a process may depend on the availability of the needed resources

CPU Scheduler

- CPU scheduler decides which process gets the CPU time next
- It is called a short-term scheduler as it only schedules for the next CPU time
- It needs to be invoked more frequently for the same reason, hence, it must be faster than the Process scheduler
- The next process to be scheduled is decided by the scheduling policy of the operating system
  - **Example-** First-Come-First-Served (FCFS), Shortest Job First (SJF), Shortest Remaining Time First (SRTF), Round Robin (RR), Priority Scheduling, etc.
- Some systems also have a medium-Term scheduler which swaps processes in and out of CPU to reduce/increase the degree of multiprogramming

Preemptive vs. Non-preemptive Scheduling

- Each process, which has been allotted CPU time, is given some resources that are required to run that process
- An operating system may decide on a policy of resource allocation that is-
  - Preemptive- where the allocated resources can be revoked from a process
    Example- RR, SRTF, Preemptive priority scheduling
  - Non-preemptive- where the allocated resources cannot be revoked from a process
    Example- FCFS, SJF and non-preemptive priority scheduling
- Preemptive scheduling ensures that no process dominates the CPU and improves average response time
- Non-preemptive scheduling ensures that no process is starved due to priority preferences and has less context switches

Non-Preemptive Scheduling Algorithms

- First-Come-First-Served (FCFS)
  - The processes are scheduled for CPU in the same order that they arrive
  - Each process regardless of the required time or resources, completes its execution and then leaves the CPU
  - Only after the current process has completed its execution, next process can be admitted
- Shortest Job First (SJF)
  - Another simple non-preemptive scheduling algorithm
  - Here, in the beginning or after the completion of one process, CPU scheduler assesses the time required for all the processes in the ready queue and admits the process that needs the least amount of CPU time
- Non-Preemptive Priority Scheduling (NPS)
  - Processes are selected based on their priority values, the smaller the number, higher the priority
  - A process once dispatched to the CPU cannot be interrupted even if other higher priority process comes

Preemptive Scheduling Algorithms

- Shortest Remaining Time First (SRTF)
  - A preemptive version of SJF where, whenever a new process enter the ready queue, the remaining time of all the processes is compared and the process with shorter remaining time is granted CPU access
  - If a shorter process enter the ready queue, CPU access of currently running longer process is revoked
- Round Robin (RR)
  - Each process gets to access the CPU for a predetermined amount of time (Time Quantum)
  - Process exits the CPU one its TQ is over, or it has completed its execution (whichever time is smaller), then the next process in queue gets the CPU access for TQ amount of time
  - The process is followed in a loop over all the processes in the ready queue
- Preemptive Priority Scheduling (NPS)
  - A preemptive version of NPS, only here, a process once dispatched to the CPU can be interrupted when other higher priority process joins the ready queue

Exercise

- Apply all the scheduling algorithms on the following processes and compute the following-
  - Average and per process Waiting Time
  - Average and per process Response Time
  - Average and per process Turn-around Time
  - Completion Time of each process

Process Number	Arrival Time	Burst Time
P1	0	7
P2	1	6
P3	2	2
P4	4	3

End