PRINCIPLES OF OPERATING SYSTEMS

LECTURE 8
Principles of Operating Systems

CPU SCHEDULING

Scheduling Objectives

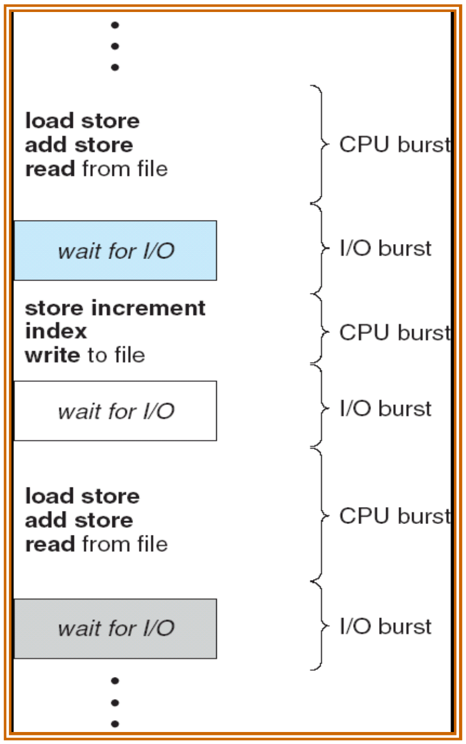
- Enforcement of fairness
  - in allocating resources to processes
- Enforcement of priorities
- Make best use of available system resources
- Give preference to processes holding key resources.
- Give preference to processes exhibiting good behavior.
- Degrade gracefully under heavy loads.

Program Behavior Issues

- I/O boundedness
  - short burst of CPU before blocking for I/O
- CPU boundedness
  - extensive use of CPU before blocking for I/O
- Urgency and Priorities
- Frequency of preemption
- Process execution time
- Time sharing
  - amount of execution time process has already received.

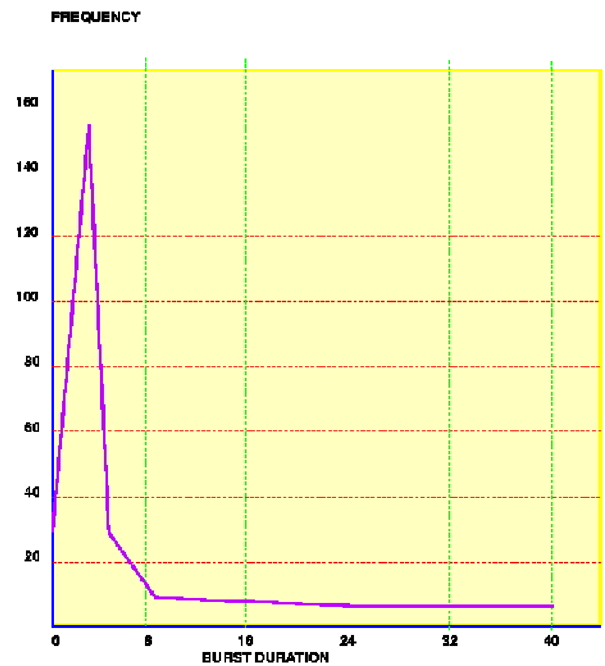
CPU and I/O Bursts

Maximum CPU utilization obtained with multiprogramming**.**



CPU-I/O Burst Cycle

Process execution consists of a cycle of CPU execution and a cycle of I/O wait.

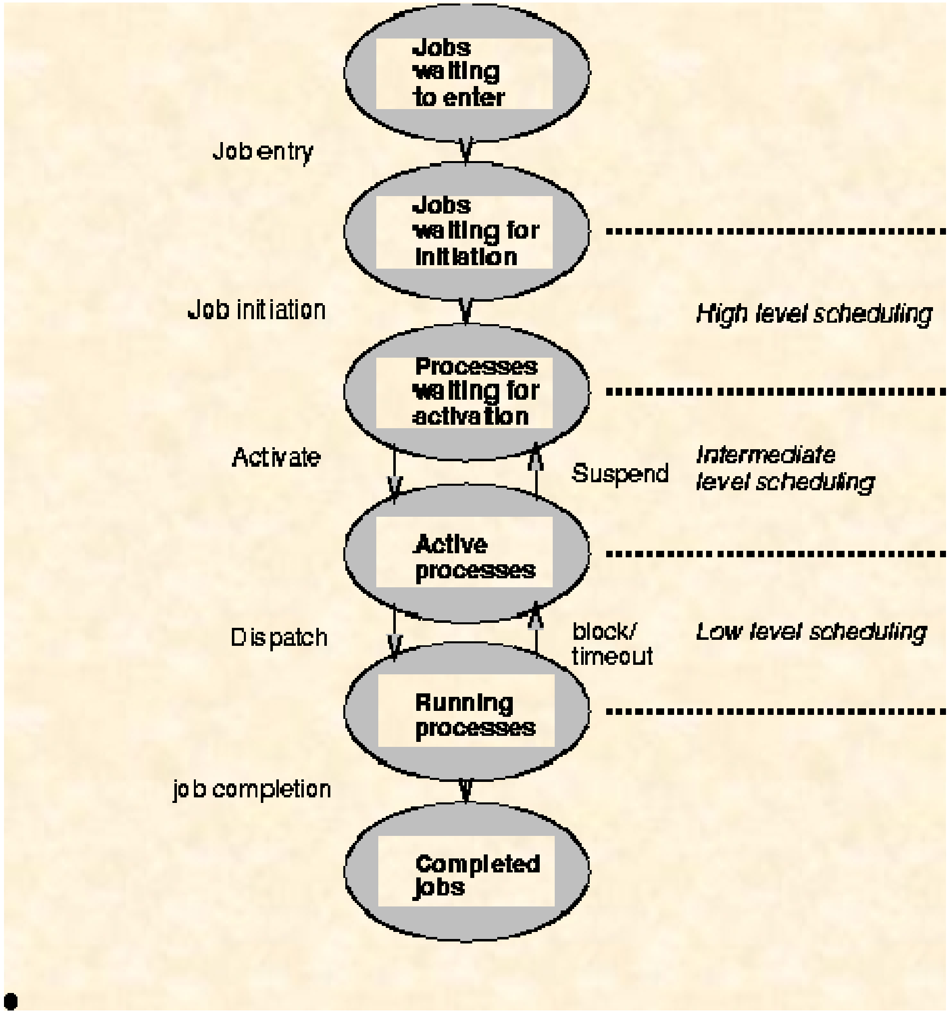


CPU Burst Distribution.

Levels of Scheduling

- High Level Scheduling or Job Scheduling
  - Selects jobs allowed to compete for CPU and other system resources.
- Intermediate Level Scheduling or Medium Term Scheduling
  - Selects which jobs to temporarily suspend/resume to smooth fluctuations in system load.
- Low Level (CPU) Scheduling or Dispatching
  - Selects the ready process that will be assigned the CPU.
  - Ready Queue contains PCBs of processes.

Levels of Scheduling(cont.)



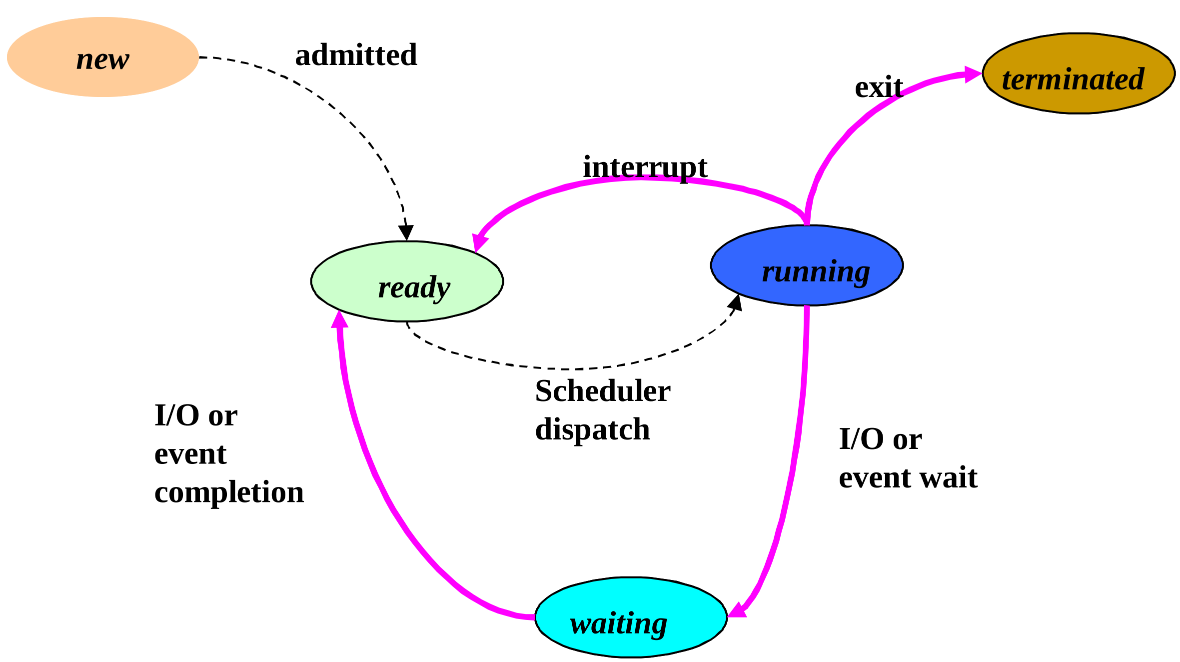
CPU Scheduler

- Selects from among the processes in memory that are ready to execute, and allocates the CPU to one of them.
  - Non-preemptive Scheduling
    - Once CPU has been allocated to a process, the process keeps the CPU until
      - Process exits OR
      - Process switches to waiting state
  - Preemptive Scheduling
    - Process can be interrupted and must release the CPU.
      - Need to coordinate access to shared data

CPU Scheduling Decisions

- CPU scheduling decisions may take place when a process:
  - switches from running state to waiting state
  - switches from running state to ready state
  - switches from waiting to ready
  - terminates
- Scheduling under 1 and 4 is non-preemptive.
- All other scheduling is preemptive.

CPU scheduling decisions



Dispatcher

- Dispatcher module gives control of the CPU to the process selected by the short-term scheduler. This involves:
  - switching context
  - switching to user mode
  - jumping to the proper location in the user program to restart that program
- Dispatch Latency:
  - time it takes for the dispatcher to stop one process and start another running.
  - Dispatcher must be fast.

Scheduling Criteria

- CPU Utilization
  - Keep the CPU and other resources as busy as possible
- Throughput
  - # of processes that complete their execution per time unit.
- Turnaround time
  - amount of time to execute a particular process from its entry time.
- Waiting time
  - amount of time a process has been waiting in the ready queue.
- Response Time (in a time-sharing environment)
  - amount of time it takes from when a request was submitted until the first response is produced, NOT output.

Optimization Criteria

- Maximize CPU Utilization
- Maximize Throughput
- Minimize Turnaround time
- Minimize Waiting time
- Minimize response time

Observations: Scheduling Criteria

- Throughput vs. response time
  - Throughput related to response time, but not identical:
    - Minimizing response time will lead to more context switching than if you only maximized throughput
  - Two parts to maximizing throughput
    - Minimize overhead (for example, context-switching)
    - Efficient use of resources (CPU, disk, memory, etc)
- Fairness vs. response time
  - Share CPU among users in some equitable way
  - Fairness is not minimizing average response time:
    - Better *average* response time by making system *less* fair