CISC 7310X
C05: CPU Scheduling

Hui Chen

Department of Computer & Information Science

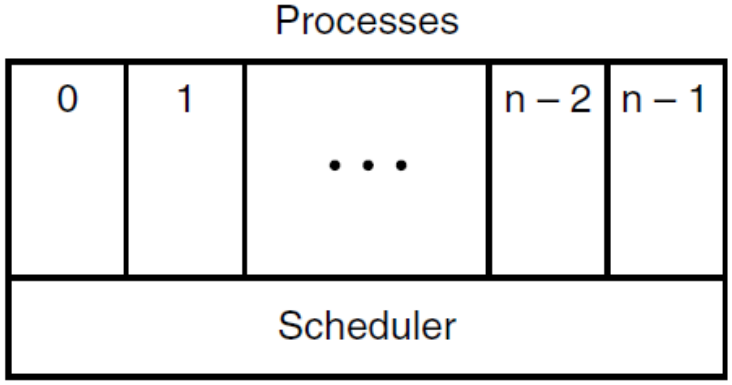
CUNY Brooklyn College

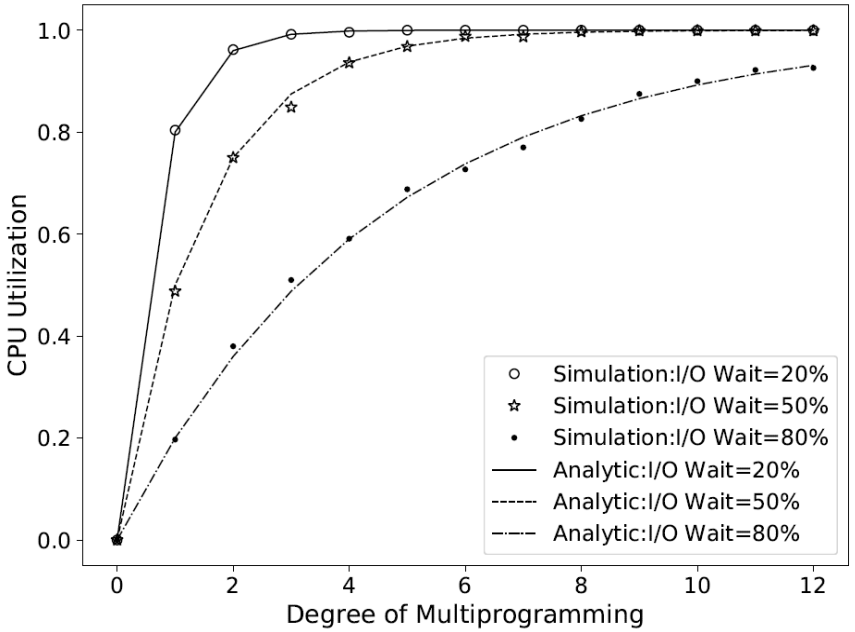
Outline

- Recap & issues
- CPU Scheduling
  - Concepts
  - Goals and criteria
  - Thread
  - Multiprocessor
- Project 2 discussion
- Assignment

Recall Process and Modeling of Multiprogramming

- Scheduler

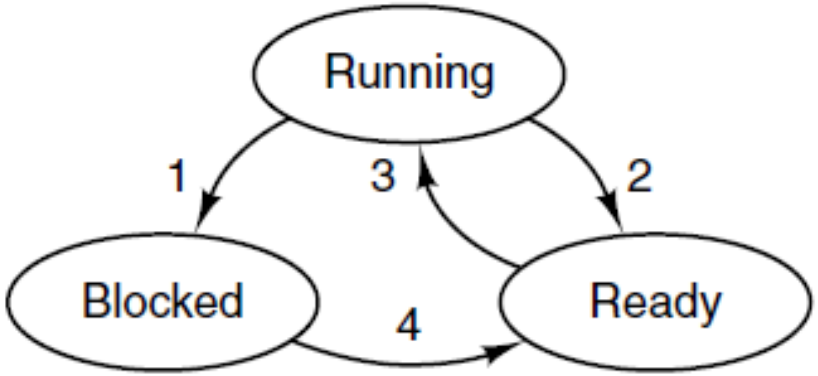




- Scheduler and processes [Figure 2-3 in Tanenbaum & Bos, 2014]
- Modeling of multiprogramming with an implicit assumption of a scheduler

Scheduler and Scheduling Algorithm

- CPU scheduler makes the decision on which process (or thread) to run next if more than one process is in the ready state with a scheduling algorithm

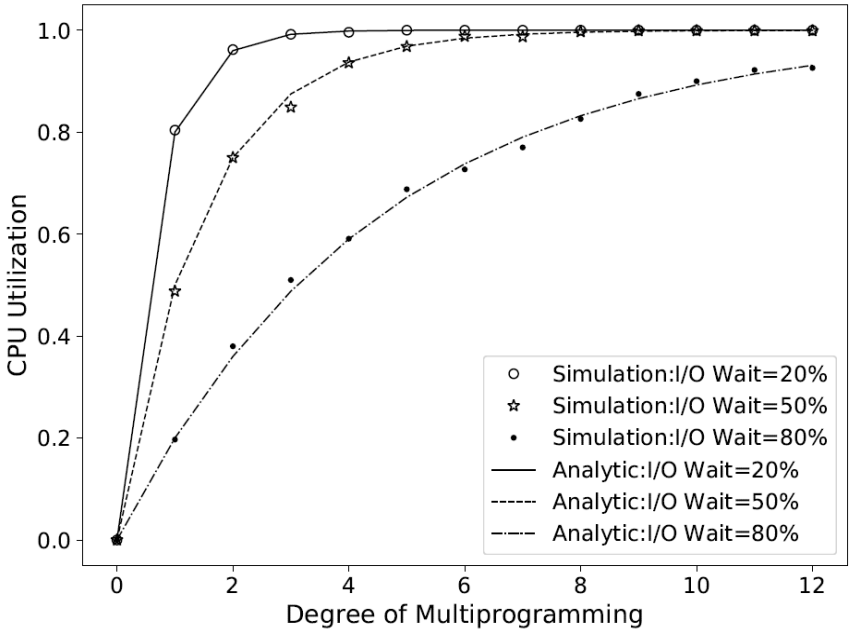


1. Process blocks for input
2. Scheduler picks another process
3. Scheduler picks this process
4. Input becomes available

- Process states [Figure 2-2 in Tanenbaum & Bos, 2014]

Assumptions?

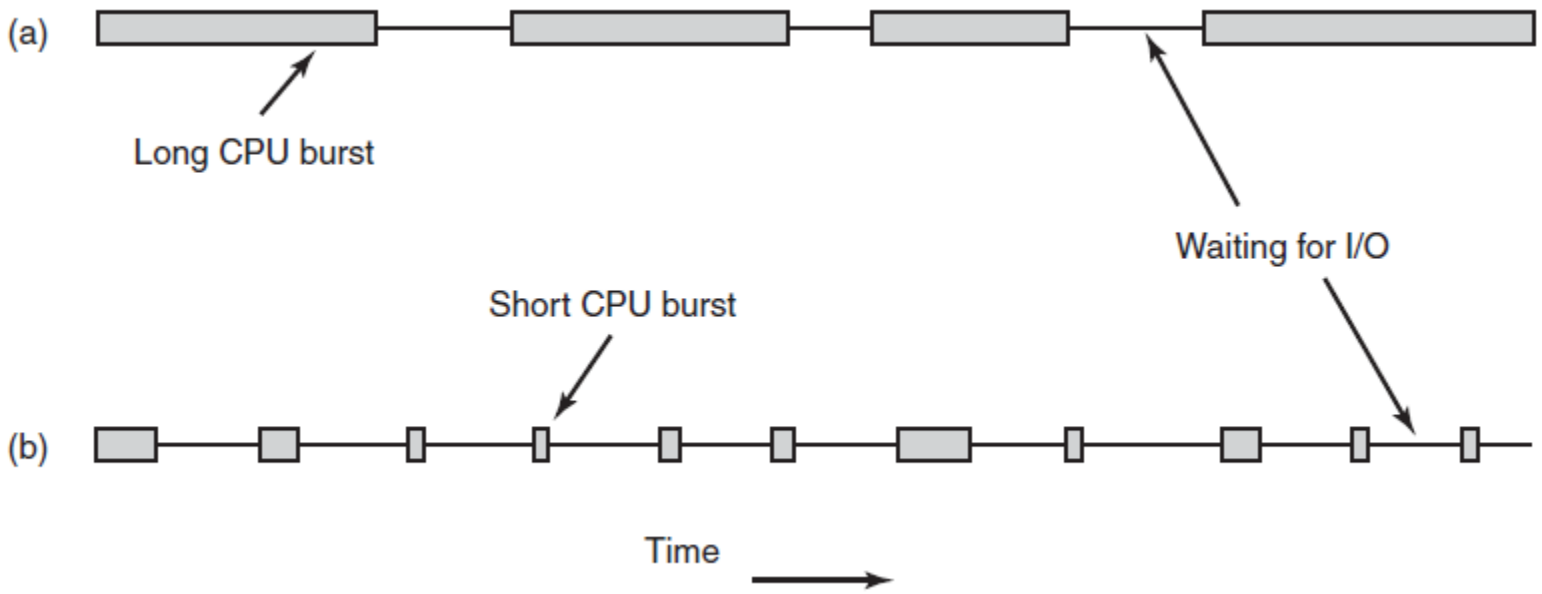
- I/O, scheduling?



  - Modeling of multiprogramming with an implicit assumption of a scheduler

Process Behavior

- A simple categorization: CPU-bound or I/O-bound?



- CPU- & I/O-bound [Figure 2-2 in Tanenbaum & Bos, 2014]

Scheduling Timing

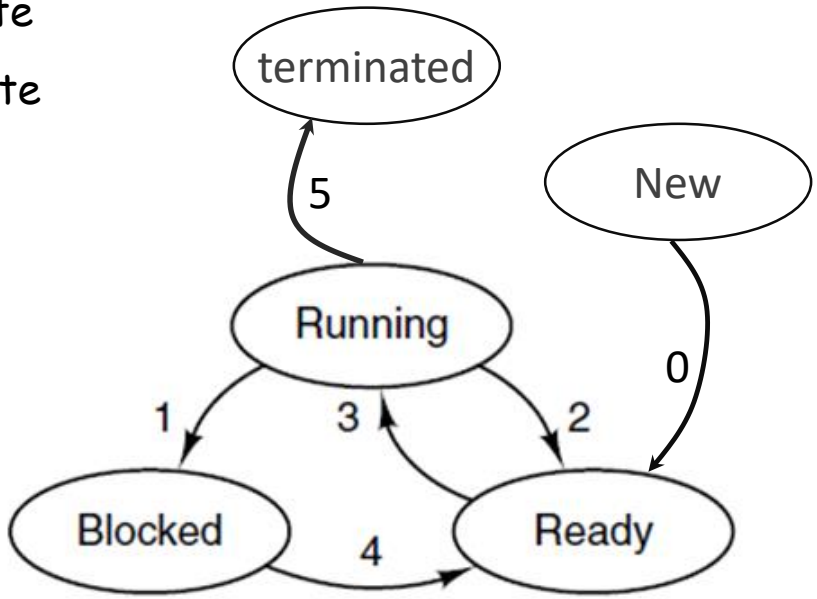
- When does a scheduler make scheduling decision?
  - When a new process is created
  - When a process exits
  - When a process blocks on I/O request or some other situations (e.g., Unix wait(2))
  - When an I/O interrupt occurs
  - When a scheduling quantum expires (clock interrupts)

Preemptive and Non-preemptive Scheduling

- Non-preemptive scheduling: selects a process to run until it blocks or voluntarily releases the CPU
- Preemptive scheduling: selects a process to run for a scheduling quantum, and suspends it to run another when the clock interrupts.

Process State and Scheduling

- Scheduler gets to run and makes a scheduling decision
  1. a process being created
  2. the running state to the blocked state
     - e.g., I/O request
  3. the running state to the ready state
  4. the blocked state to the ready state
     - e.g., I/O interrupt
  5. a process terminates
- Non-preemptive scheduling: 2 & 5
- Preemptive scheduling: 3 & 4
  - Hardware support (clock interrupts)



Preemptive Scheduling

- Require hardware support (clock interrupt)
- More complex
  - How to deal with shared data (2 processes share data, can the 2nd process read & write the data?)
  - Kernel code also need to maintain data structures, but interrupts can happen at any time

Categories of Scheduling Algorithms

- Batch
- Interactive
- Realtime

Scheduling Goals and Criteria

- General goals
- Scheduling criteria
- System specific criteria
  - Batch
  - Interactive
  - Realtime

General Goals

- Fairness
  - Giving each process a fair share of thee CPU
- Policy enforcement
  - Seeing that stated policy is carried out
- Balance
  - Keeping all parts of the system busy

Scheduling Criteria

- CPU utilization: percentage of time that CPU is busy
- Throughput: processes/jobs/tasks completed per unit of time
- Turnaround time: the interval from the time of submission of a process to the time of completion
- Response time: latency to begin a response to a request
- Waiting time: the sum of the time waiting in the ready state
- Proportionality: how well user's expectations are met
- Predictability: variance of various criteria and scheduling behavior
- Meeting deadlines: how well deadlines are met (or certain tasks)

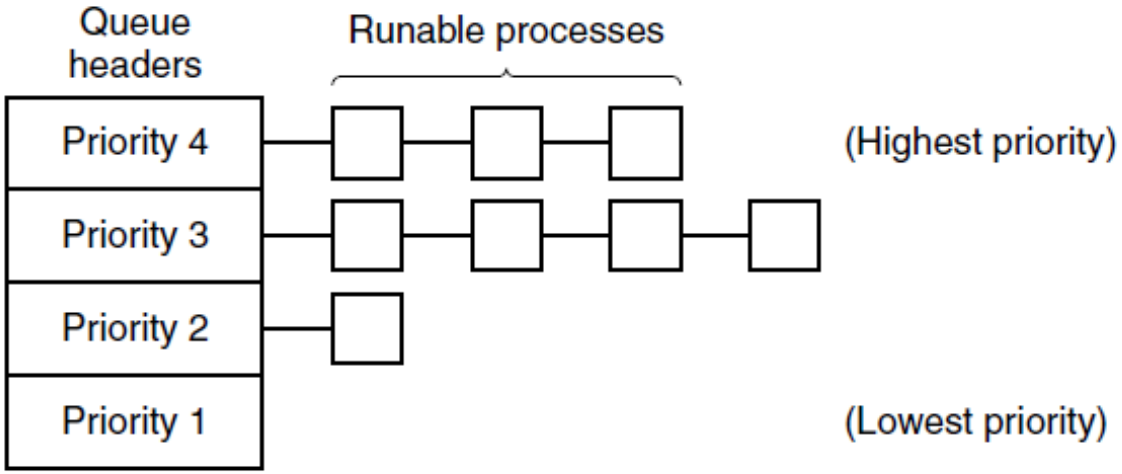
Shortest-Job-First

- (a) Running four jobs in the original order. (b) Running them in shortest job first order.[Figure 2-41 in Tanenbaum & Bos, 2014]

Round-Robin

- (a) The list of runnable processes. (b) The list of runnable processes after *B* uses up its quantum. [Figure 2-42 in Tanenbaum & Bos, 2014]

Priority-Scheduling



- A scheduling algorithm with four priority classes. [Figure 2-43 in Tanenbaum & Bos, 2014]

Batch Systems

- Goals
  - Maximize throughput
  - Minimize turnaround time
  - Maximize CPU utilization

Interactive Systems

- Goals
  - Minimize response time
  - Optimize proportionality to meet users' expectations

Realtime Systems

- Goals
  - Maximizes the chance to meet deadlines
  - Maximizes predictability of system behavior

Scheduling Algorithms

- First-Come, First-Served (FCFS)
- Shortest-Job-First (SJF)
- Shortest-Remaining-Time-First (SRTF)
- Round-Robin (RR)
- Priority Scheduling
- Multilevel Queue
- Shortest-Process-First
- Guaranteed Scheduling
- Lottery Scheduling
- Fair-Share Scheduling

Batch Systems

- First-Come, First-Served
- Shortest-Job-First
- Shortest-Remaining-Time-First

Interactive System

- Round-Robin Scheduling
- Priority Scheduling
- Multiple Queues
- Shortest Process Next
- Guaranteed Scheduling
- Lottery Scheduling
- Fair-Share Scheduling

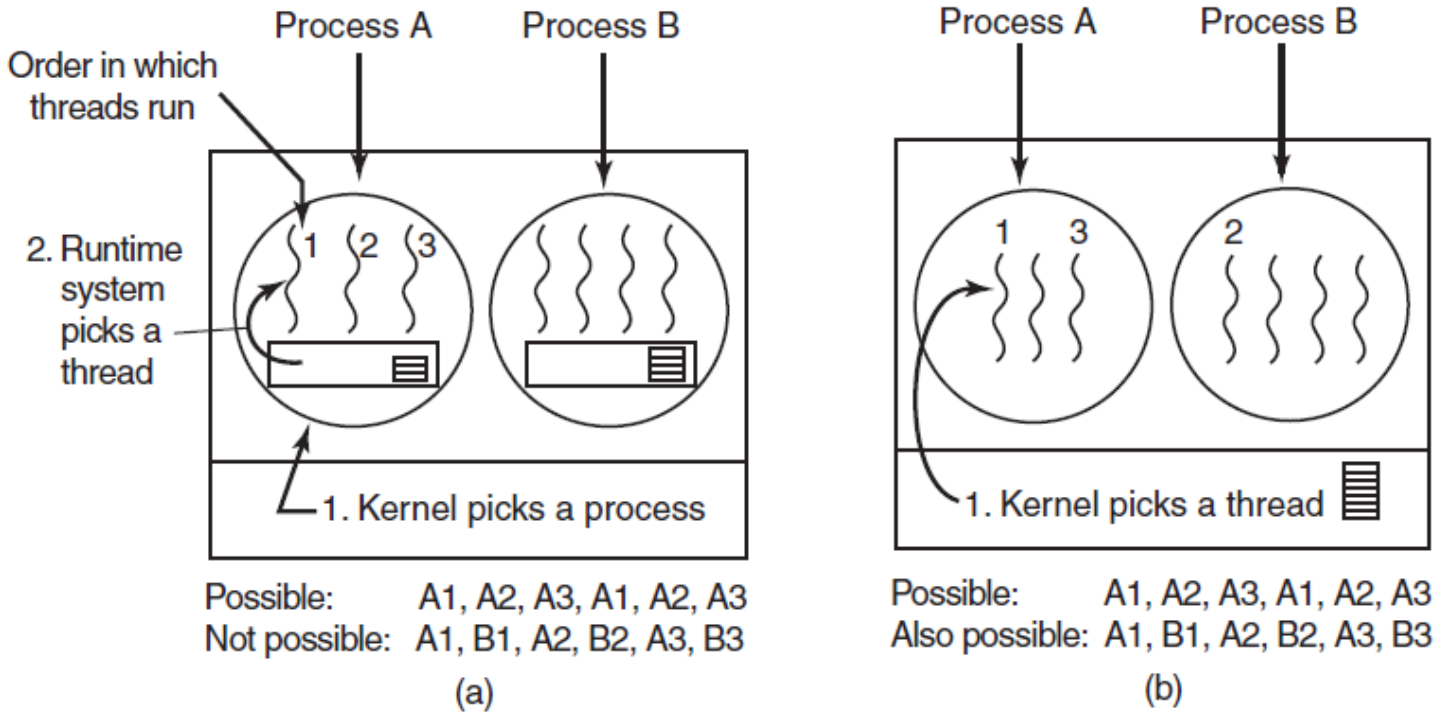
Realtime System

- Time plays essential role
  - Must or maximize chances to meet deadlines
- Categories
  - Hard real time
  - Soft real time
  - Periodic or aperiodic
- Schedulable condition $\sum_{i=1}^{m} \frac{C_i}{P_i} \leq 1$

Policy vs. Mechanism

- Mechanism enforces policy
  - Scheduling algorithms need to be parameterized
  - User processes can fill the parameters to meet policy goals

Thread Scheduling



- (a) Possible scheduling of user-level threads with a 50-msec process quantum and threads that run 5 ms per CPU burst. (b) Possible scheduling of kernel-level threads with the same characteristics as (a). [Figure 2-44 in Tanenbaum & Bos, 2014]

Contention Scope

- Process contention scope
  - User-level threads compete for CPU within a process
- System-contention scope
  - Kernel threads compete for CPU within the system

POSIX Thread

- POSIX thread library provides API to select contention scope
  - indicating whether a user-space thread is bound directly to a single kernel-scheduling entity
  - PTHREAD_SCOPE_PROCESS
    PTHREAD_SCOPE_SYSTEM

Multiple-Processor Scheduling

- Asymmetric multiprocessing
  - All scheduling decision, I/O processing, and other system activities are handled by a single processor
- Symmetric multiprocessing
  - Each processor is self-scheduling
    - common ready queue
    - private ready queue

Multiple-Processor Scheduling: Considerations

- Processor affinity
- Load balancing
- Multicore processors
- Virtualization and scheduling

Questions?

- Concepts of scheduling
- Scheduling goals and criteria
- Scheduling algorithms
- Process and thread scheduling
- Considerations in multiprocessor scheduling