CISC 7310X

C05a: Basic Concepts about Multiprogramming

Hui Chen

Department of Computer & Information Science

CUNY Brooklyn College

Acknowledgement

- This slides are a revision of the slides by the authors of the textbook

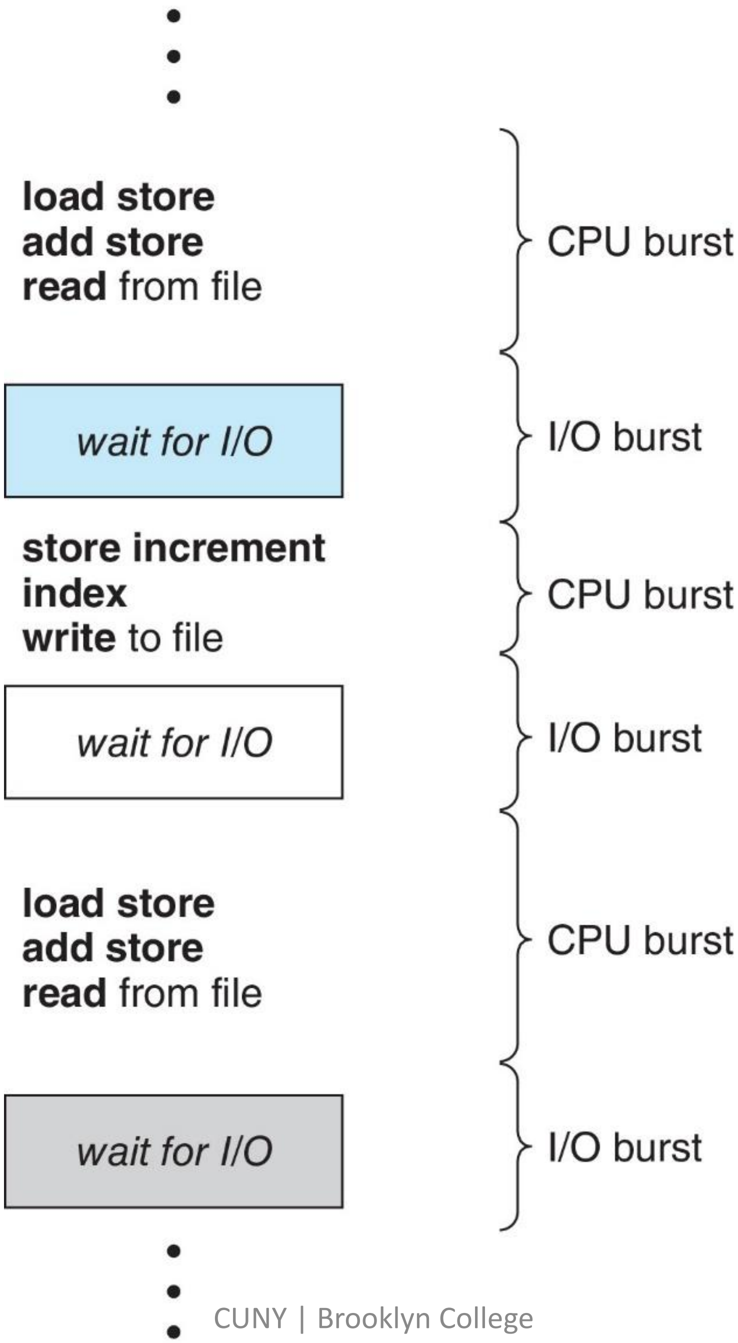
Outline

- Basic Concepts

- Scheduling Criteria
- Scheduling Algorithms
- Thread Scheduling
- Multi-Processor Scheduling
- Real-Time CPU Scheduling
- Operating Systems Examples
- Algorithm Evaluation

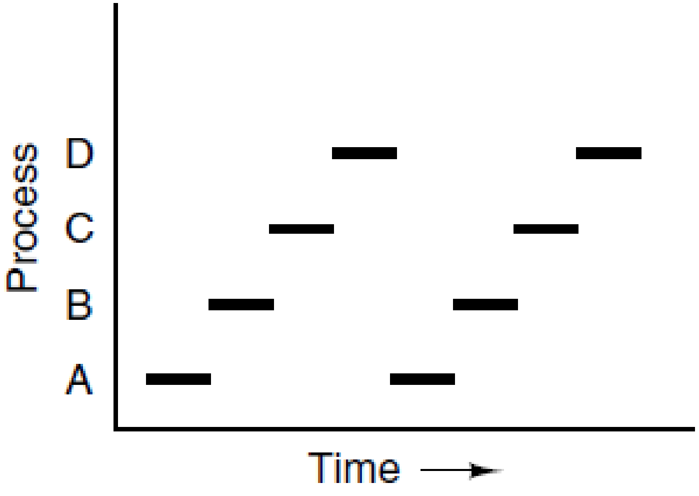
CPU-I/O Burst Cycle

- CPU–I/O Burst Cycle
  - Process execution consists of a **cycle** of CPU execution and I/O wait
- **CPU burst** followed by **I/O burst**


load store
add store
read from file
CPU burst
wait for I/O
I/O burst
store increment
index
write to file
CPU burst
wait for I/O
I/O burst
load store
add store
read from file
CPU burst
wait for I/O
I/O burst

Multiprogramming

- Process rapidly switching back and forth to share the CPU time



(c)

- In a single processor core system, only one program is active at once (pseudoparallelsim) [Figure 2-1(c) in Tanenbaum & Bos, 2014]

Benefit of Multiprogramming

- CPU utilization can be improved due to multiprogramming
- Intuition
  - When one process is waiting for I/O, another can be scheduled to CPU

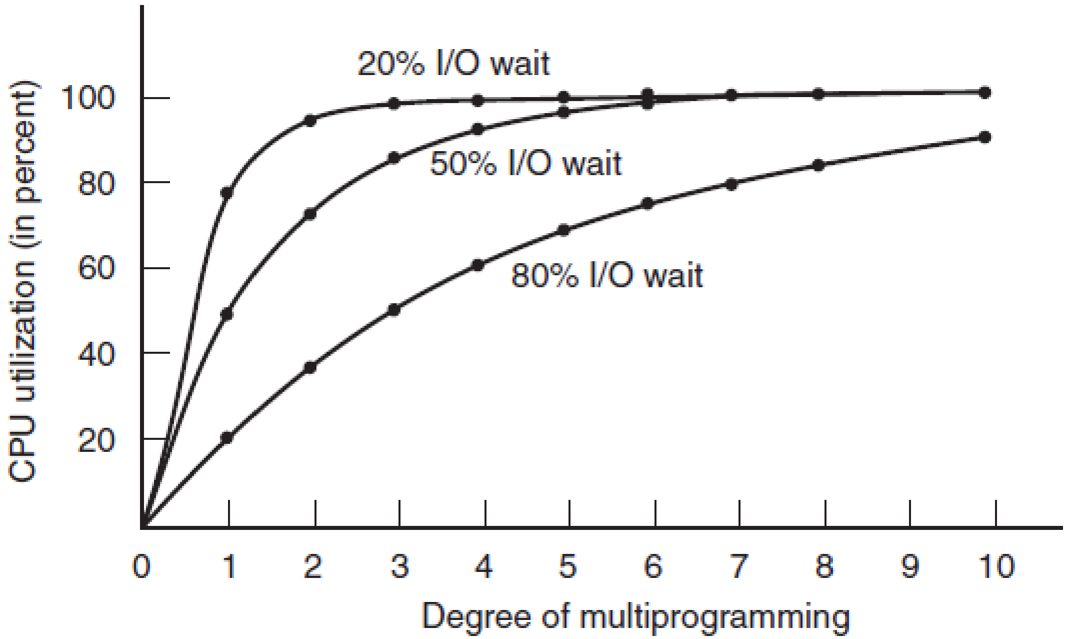
How much do we benefit?

Simple Multiprogramming Model

- Assumptions
  - **n** processes in the main memory;
  - a process spends a fraction **p** of its waiting for I/O independent of the others
- Analysis
  - CPU is idle when all processes are waiting for I/O
  - The probability that all n processes waiting for I/O is p^n
  - CPU Utilization = $1 - p^n$

Result from the Simple Modeling Multiprogramming

- CPU utilization



- CPU utilization [Figure 2-6 in Tanenbaum & Bos, 2014]

Questions

- Review the concepts of multiprogramming and process
- Benefit of multiprogramming
- Model and simulation of multiprogramming
- Examples of simulation and graphing
- What are the assumptions in the model?

What are the assumptions in the model?

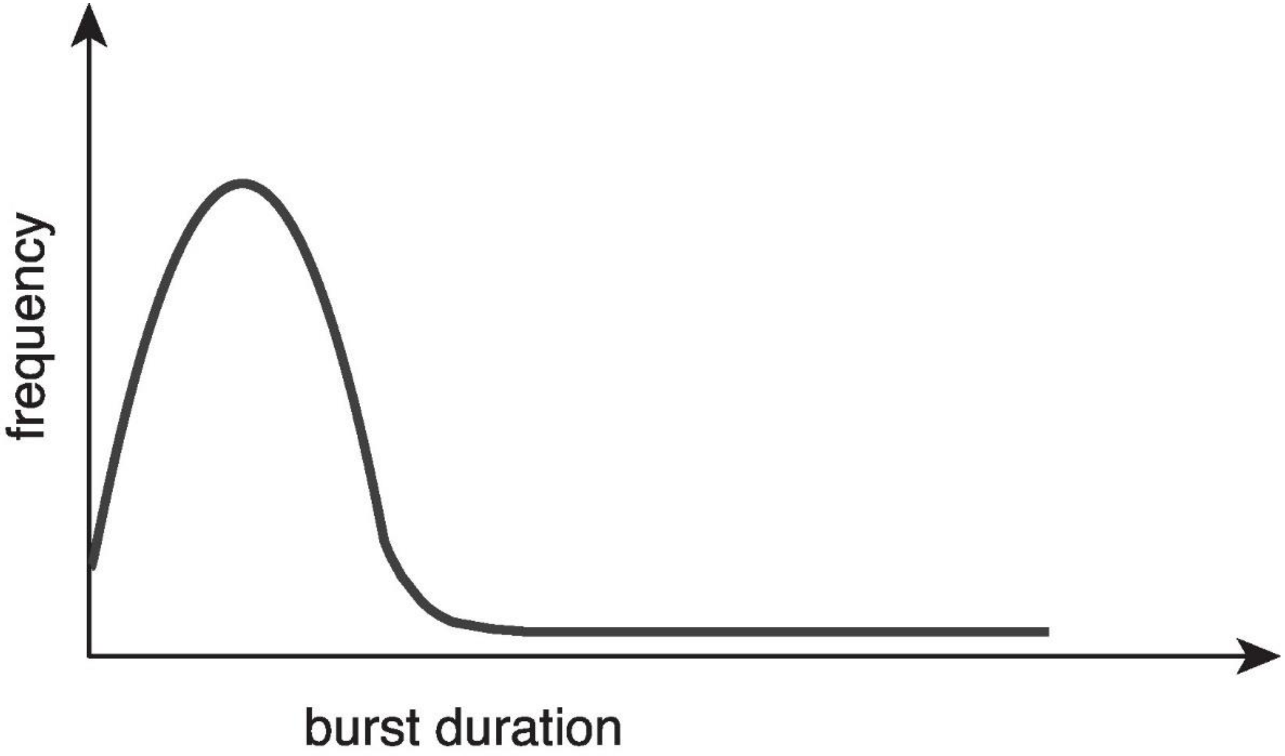
- **n** processes in the main memory and treated equally
- A process spends a fraction **p** of its waiting for I/O independent of the others

CPU Bursts Differ

- CPU burst distribution is of main concern

Histogram of CPU-burst Times

- Large number of short bursts, small number of longer bursts



CPU Scheduler: Intuition

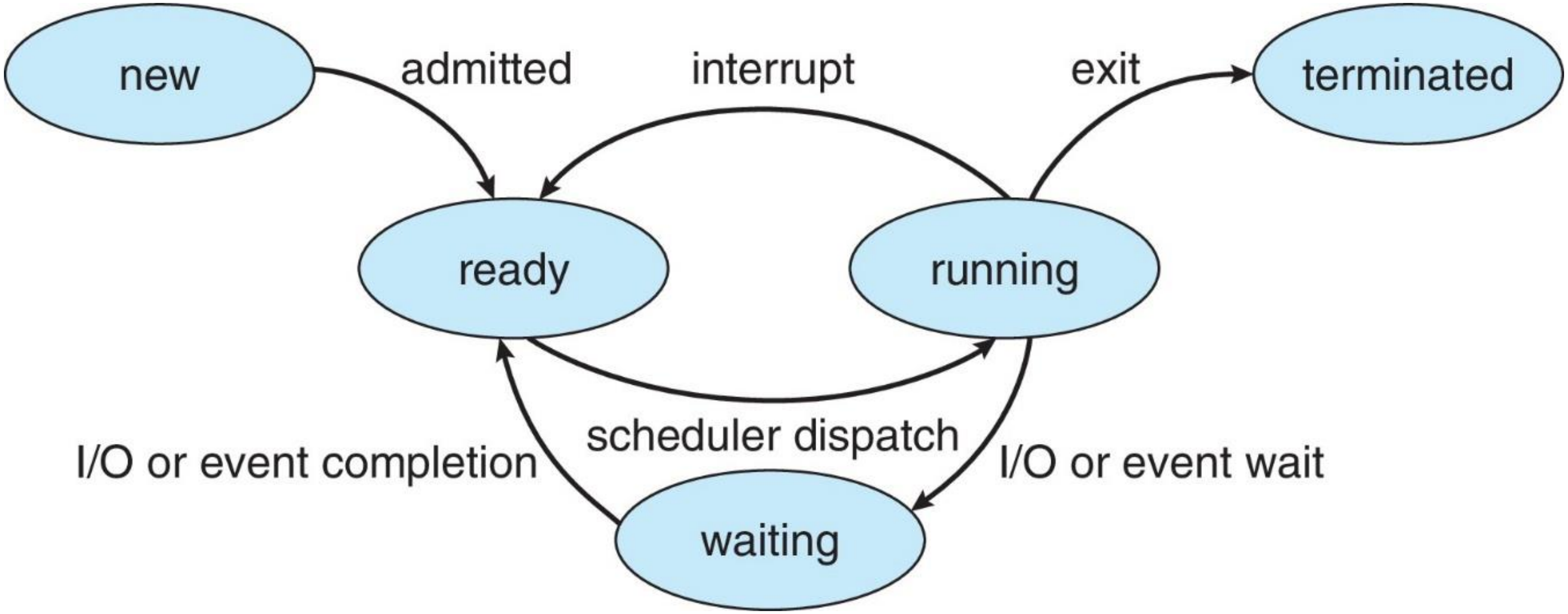
- Select processes to run on CPU to take advantage of the understanding of the CPU bursts distribution

CPU Scheduler

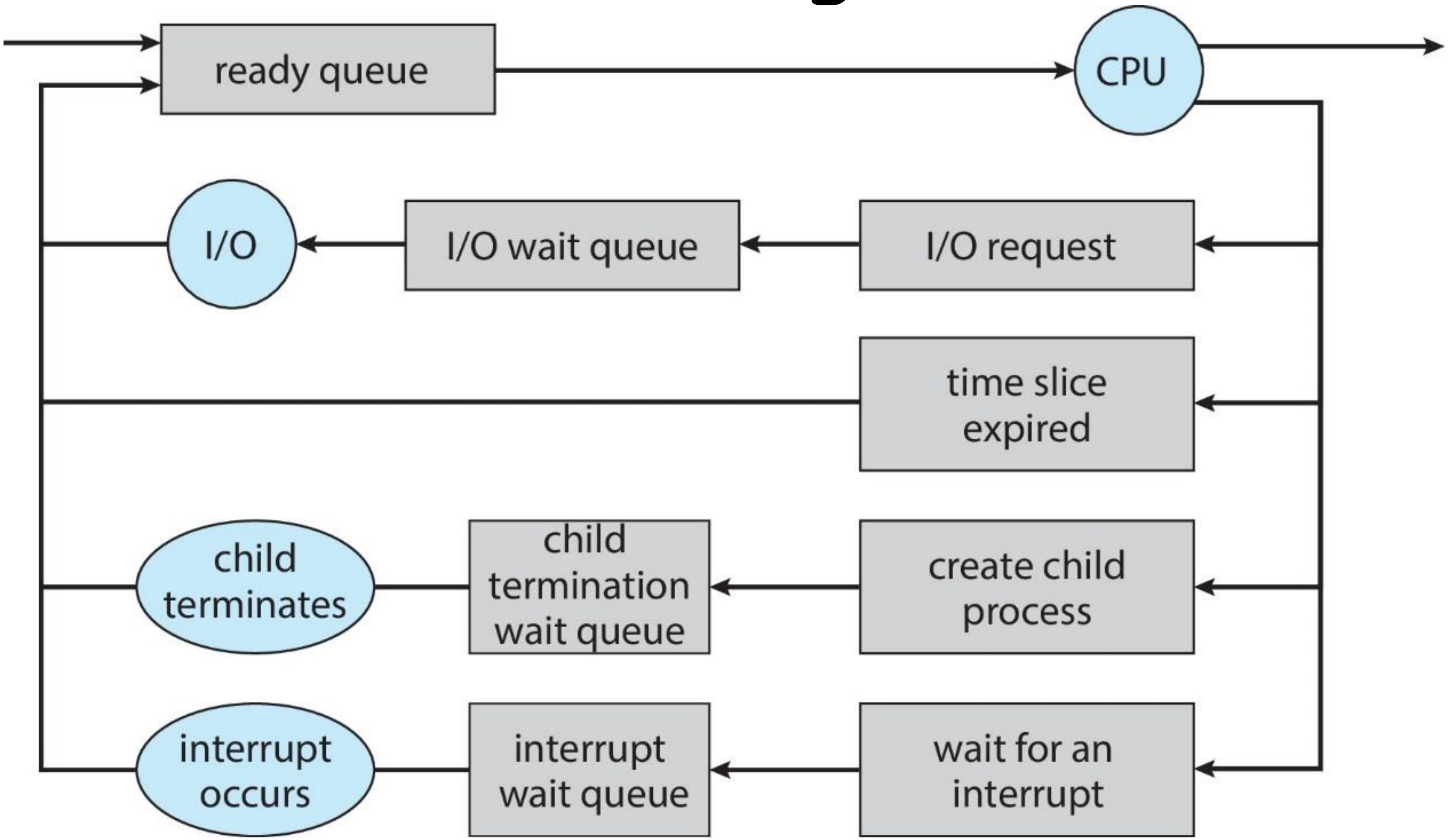
- The CPU scheduler selects from among the processes in ready queue, and allocates the a CPU core to one of them

CPU Scheduler: When, From and To?

Review: Process States



Review: Representation of Process Scheduling



CPU Scheduler: When, From and To?

- CPU scheduling decisions may take place when a process:
  - Switches from running to waiting state
  - Switches from running to ready state
  - Switches from waiting to ready
  - Terminates

CPU Scheduler

- Queue may be ordered in various ways

- Scheduling under 1 and 4 is nonpreemptive
- All other scheduling is preemptive
- Consider access to shared data
- Consider preemption while in kernel mode
- Consider interrupts occurring during crucial OS activities

Preemptive and Nonpreemptive Scheduling

- Scheduling under the following is nonpreemptive
  - Switches from running to waiting state
  - Terminates
- All other scheduling is preemptive, such as,
  - Switches from running to ready state
  - Switches from waiting to ready
- For preemptive scheduling,
  - Consider access to shared data
  - Consider preemption while in kernel mode
  - Consider interrupts occurring during crucial OS activities

Ready Queue

- Queue may be ordered in various ways

Questions?

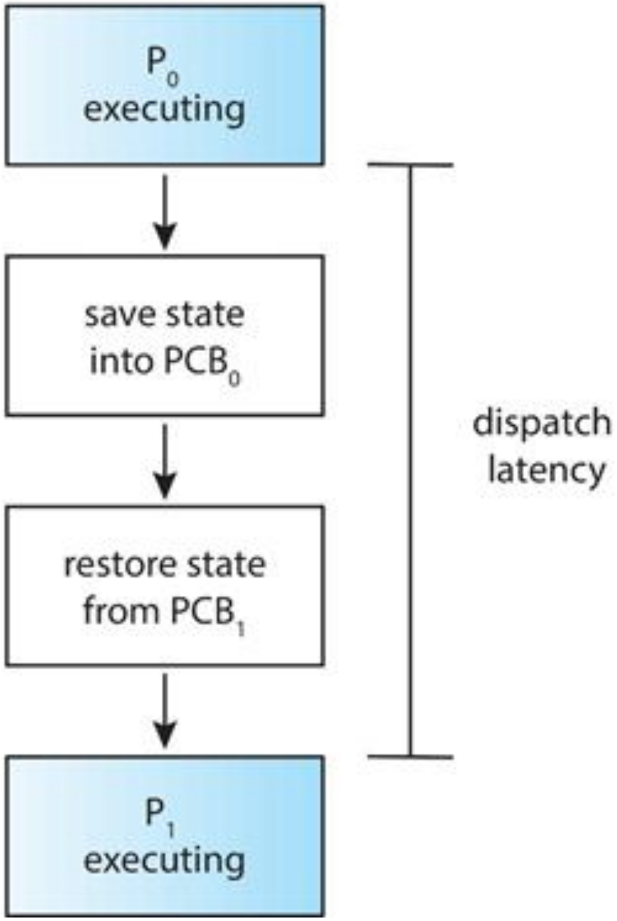
- CPU scheduler
- Preemptive and nonpreemptive scheduling

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



Questions?

- Dispatcher and dispatcher latency