CSc 360
Operating Systems
CPU Scheduling

Jianping Pan

Summer 2015

A reminder on all assignments

- encourage discussion and collaboration
  - connex chat room,wiki, (forum) etc
  - tutorial hours, office hours, etc
  - consultant office, help sessions, etc
- all you submitted work should be yours!
  - if you use anything out there, reference and credit, so your contribution can be evaluated properly; otherwise, academic integrity issues
  - do not short-cut assignments---it's part of the training process that you have paid for
- do submit your work on/before due date

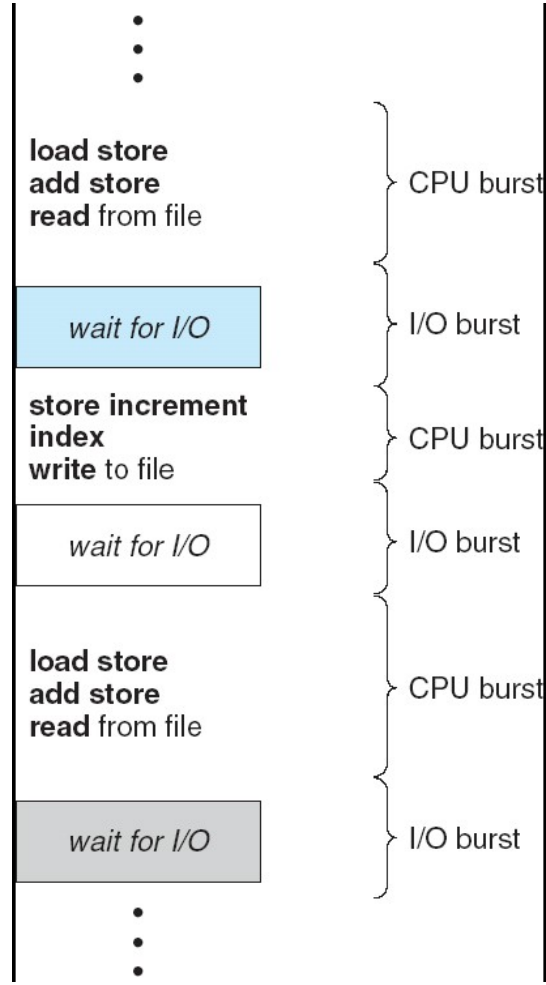
* p2 posted on connex already; deliverable 1 due this Thursday!

Review: process and thread

- Uniprogramming
- Multiprogramming
- Multitasking
- Multithreading
- How to handle many processes/threads?
  - state: ready, running, blocked
  - scheduling (PCB, TCB)

CPU scheduling

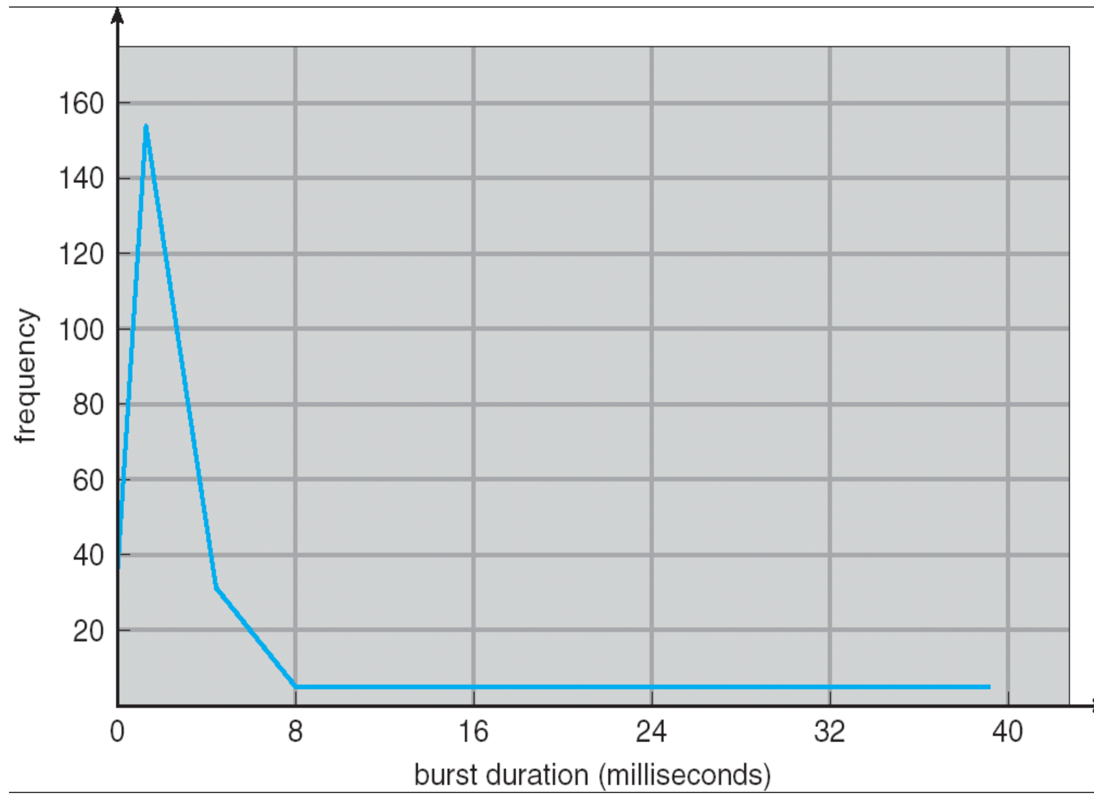
- Goal
  - maximize resource utilization
    - CPU, memory, storage
  - improve system responsiveness
- Example
  - CPU burst
  - I/O burst



Q: why scheduling?

CPU burst distribution

- Observation
  - many short bursts
  - a few long bursts
- Why is this important?



CPU scheduler

- CPU scheduler
  - short-term scheduling
- CPU scheduling
  - switch from running to waiting (blocked)
  - switch from running to ready
  - switch from waiting to ready
  - terminate (i.e., leave the system)
- Preemptive vs non-preemptive

state diagram on blackboard

CPU dispatcher

- Dispatcher
  - give control to the one selected by scheduler
- Procedures
  - context switching (save old, load new, etc)
  - mode switching (e.g., switch to user mode)
  - start to execute from the newly loaded PC
- Performance measure
  - dispatch latency/overhead

context switching diagram

Scheduling criteria

- "Who's next?"
  - CPU utilization: keep CPU as busy as possible
  - throughput: # of process done per unit time
  - turnaround time: from start to finish
  - waiting time: time spent in ready state
  - response time: interactive, request-reply
- Goal
  - max {…}, min {…}

Q: conflicting goals?

Scheduling algorithms

- First come, first serve (FCFS)
  - served by the order of arrival
- Example
  - P1(24), P2(3), P3(3)
  - schedule A
    - P1, P2, P3
  - schedule B (if they arrive at the same time)
    - P2, P3, P1

Gantt chart

Q: which schedule is better?

This lecture

- CPU scheduling
  - who’s who
    - scheduler, dispatcher
  - who’s next
    - FCFS (and many more)
- Explore further
  - what’s the best way (data structure + algorithm) to implement an FCFS scheduler?

Next lecture

- More scheduling algorithms to come
  – read OSC7 Chapter 5 (or OSC6 Chapter 6)