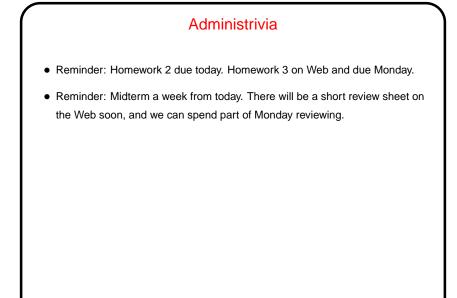
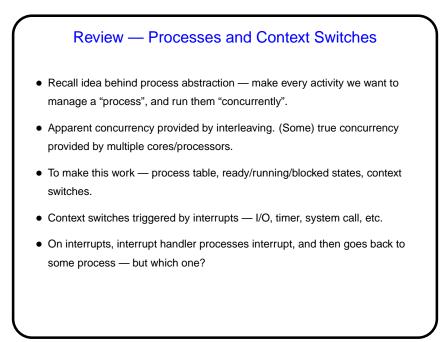
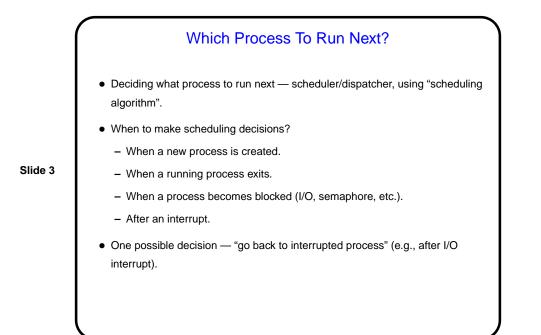
Slide 1

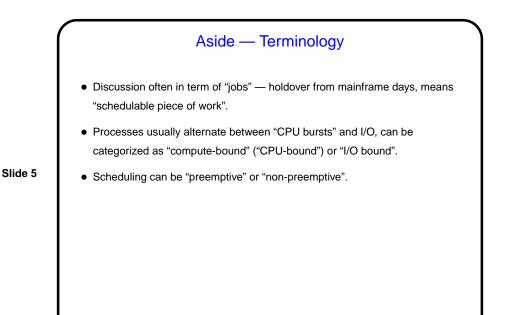




Slide 2



	Scheduler Goals
Slide 4	 Importance of scheduler can vary; extremes are Single-user system — often only one runnable process, complicated decision-making may not be necessary (though still might sometimes be a good idea). Mainframe system — many runnable processes, queue of "batch" jobs waiting, "who's next?" an important question.
	 Servers / workstations somewhere in the middle. First step is to be clear on goals — want to make "good decisions", but what does that mean? Typical goals for any system: Fairness — similar processes get similar service. Policy enforcement — "important" processes get better service. Balance — all parts of system (CPU, I/O devices) kept busy (assuming there is work for them).



Scheduler Goals By System Type
For batch (non-interactive) systems, possible goals (might conflict):

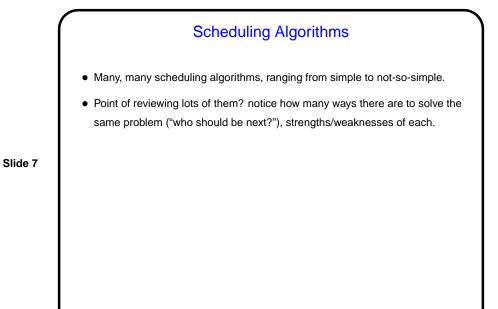
Maximize throughput — jobs per hour.
Minimize turnaround time.
Maximize CPU utilization.

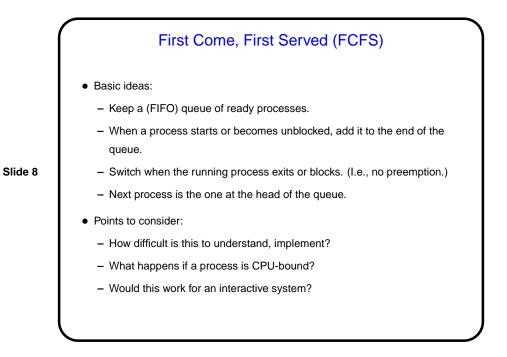
Preemptive scheduling may not be needed.
For interactive systems, possible goals:

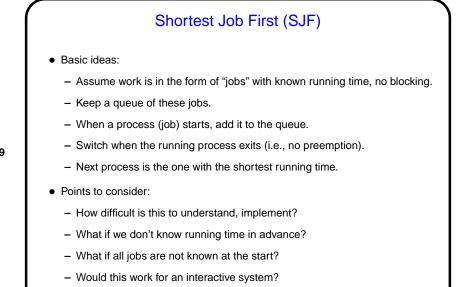
Minimize response time.
Make response time proportional (to user's perception of task difficulty).
Preemptive scheduling probably needed.

For real-time systems, possible goals:

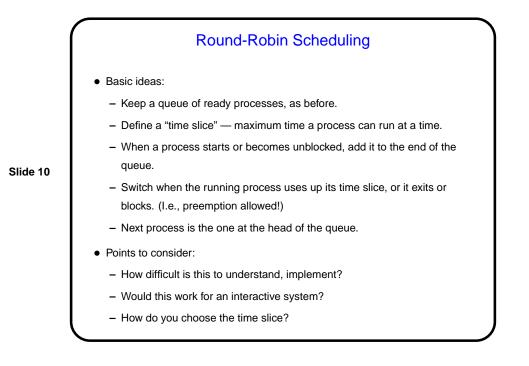
Meet time constraints/deadlines.
Behave predictably.



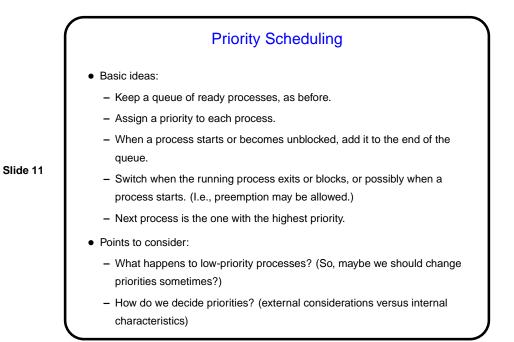




- What's the key advantage of this algorithm?

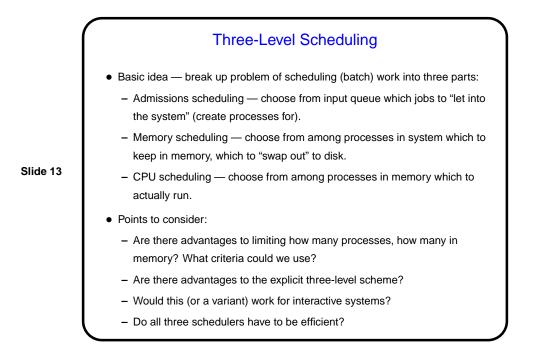


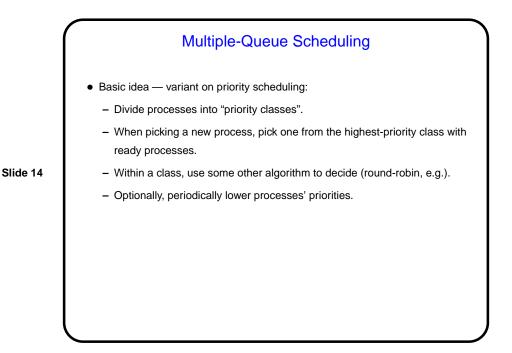
Slide 9



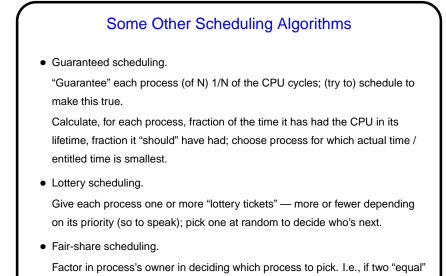
Shortest Remaining Time Next Basic idea — variant on SJF: Assume that for each process (job), we know how much longer it will take. Keep a queue of ready processes, as before; add to it as before. Switch when the running process exits or a new process starts. (I.e., preemption allowed — requires recomputing time left for preempted process.) Next process is the one with the shortest time left. Points to consider: How does this compare with SJF?

Slide 12

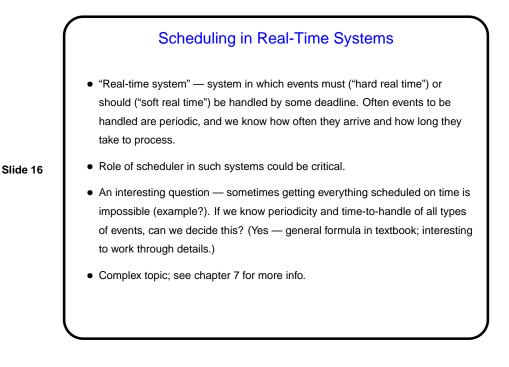


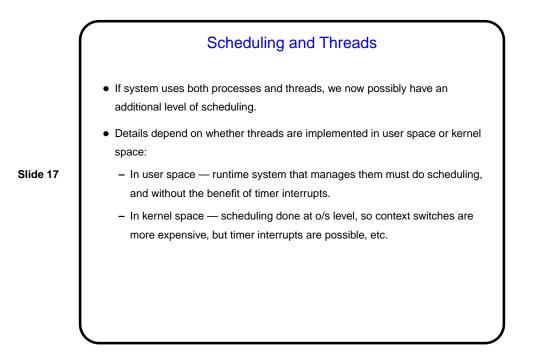


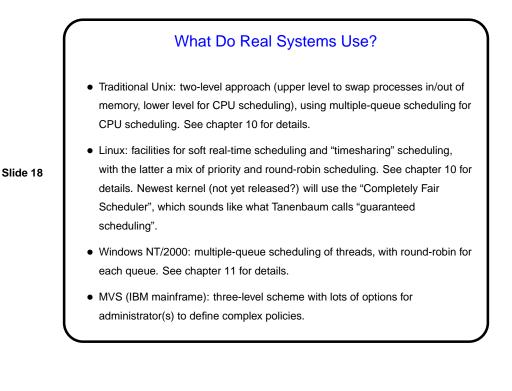
Slide 15

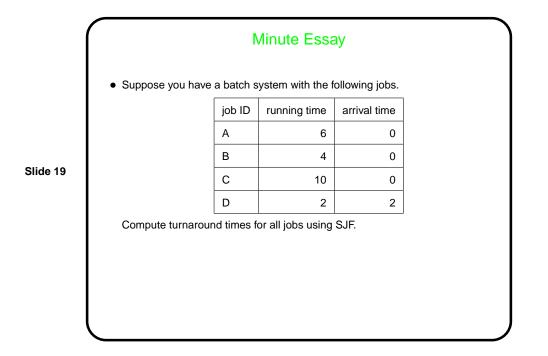


users, schedule processes such that user A's processes get about as much time as those of user B.









 Solution: job ID start time stop time turnaround time (SJ
job ID start time stop time turnaround time (SJ
A 6 12
B 0 4
lide 20 C 12 22
D 4 6