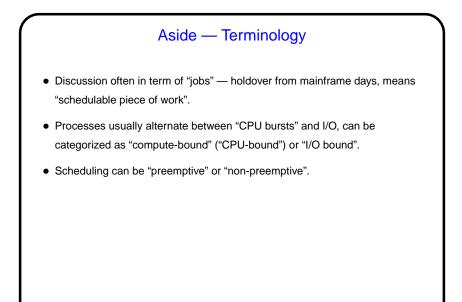
Administrivia

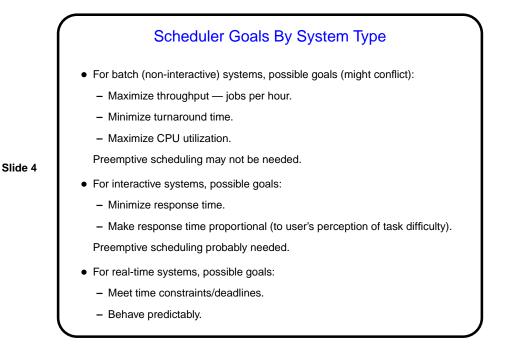
- Draft version of homework 2 on the Web. Questions probably will not change but point values might. Written problems due in a week, programming problem(s) the following Monday. (Whether there will be just the one, or a second one — I will let you know when I hear from more of you about language(s).)
- Reminder: Quiz 2 next Monday. Topics from chapter 2, not including scheduling algorithms.

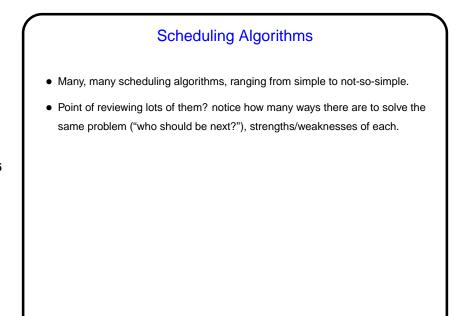


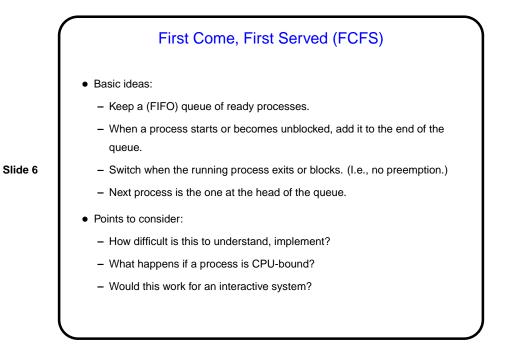
- Deciding what process to run next scheduler/dispatcher, using "scheduling algorithm".
- When to make scheduling decisions?
 - When a new process is created.

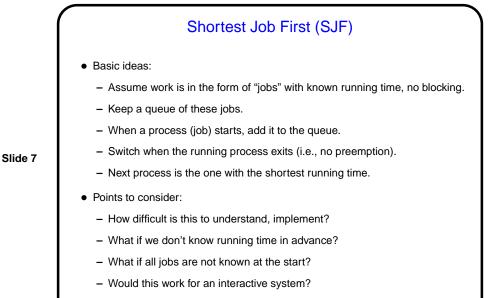
- When a running process exits.
- When a process becomes blocked (I/O, semaphore, etc.).
- After an interrupt.
- One possible decision "go back to interrupted process" (e.g., after I/O interrupt). But there are other choices.



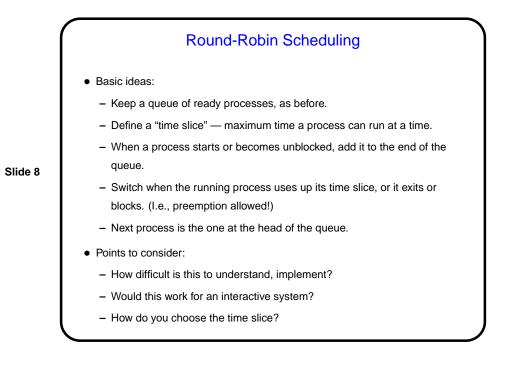


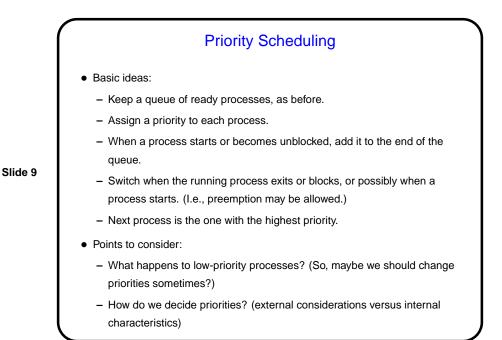




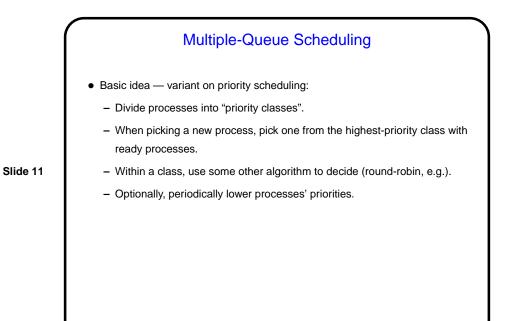


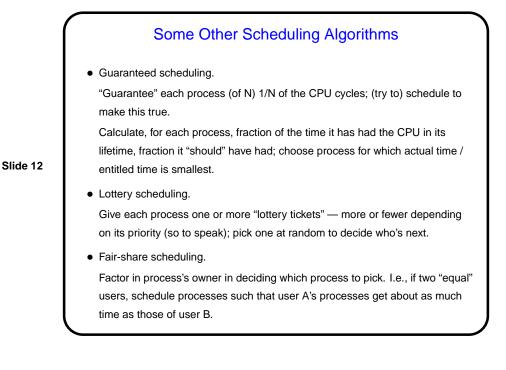
- What's the key advantage of this algorithm?





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?



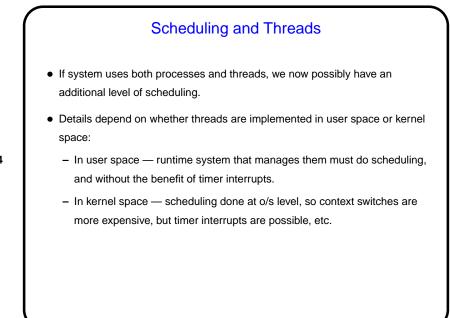


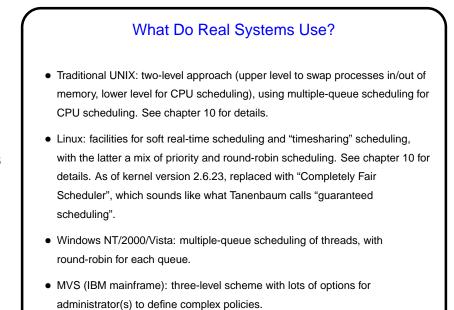
Scheduling in Real-Time Systems

 "Real-time system" — system in which events must ("hard real time") or should ("soft real time") be handled by some deadline. Often events to be handled are periodic, and we know how often they arrive and how long they take to process.

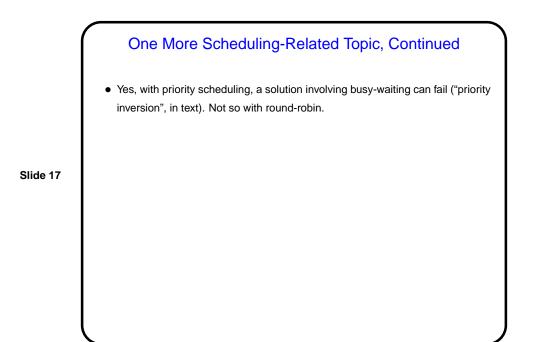
Slide 13

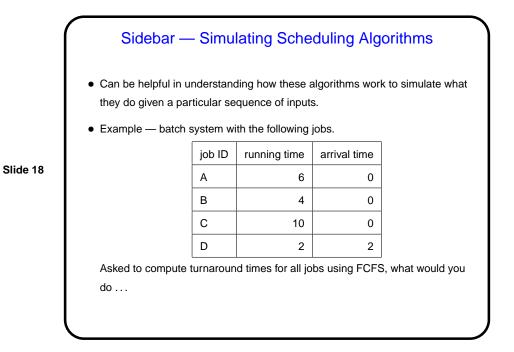
- Role of scheduler in such systems could be critical.
- An interesting question sometimes getting everything scheduled on time is impossible (example?). If we know periodicity and time-to-handle of all types of events, can we decide this? (Yes — general formula in textbook; can be interesting to work through details.)
- Complex topic; see chapter 7 for more info.

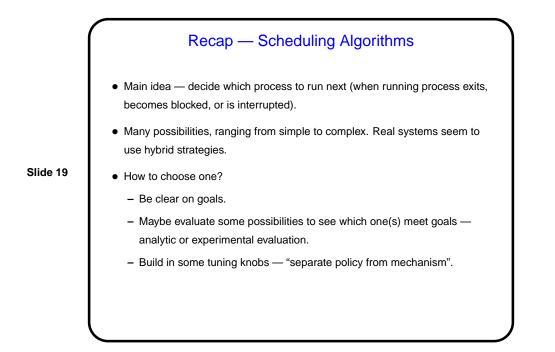




One More Scheduling-Related Topic • A question I used to use as homework: Recall that some proposed solutions to the mutual-exclusion problem (e.g., Peterson's algorithm) involve busy waiting. Do such solutions work if priority scheduling is being used and one of the processes involved has higher priority than the other(s)? Why or why not? How about if round-robin scheduling is being used? Why or why not? Notice that a process can be interrupted while in its critical region; if that happens, it is considered to still be in its critical region, and other processes wanting to be in their critical regions are supposed to busy-wait.







	Minute Essay			
	 Suppose you have 	a batch s	ystem with the f	ollowing jobs.
ilide 20		job ID	running time	arrival time
		А	6	0
		В	4	0
		с	10	0
		D	2	2
	Compute turnaround times for all jobs using SJF.			

SI

Solution: job ID start time stop time turnaround time (Sub-
job ID start time stop time turnaround time (S
A 6 12
B 0 4
de 21 C 12 22
D 4 6