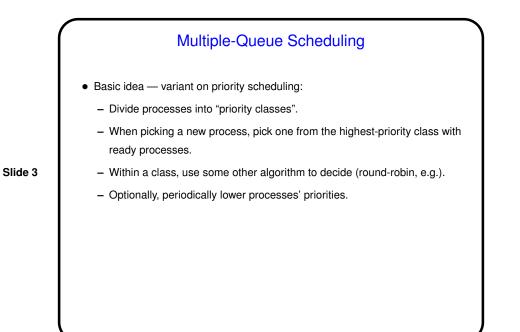
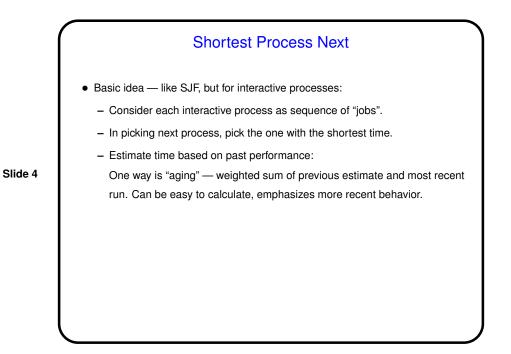


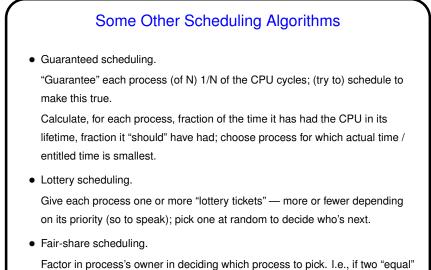
Slide 1

Recap — Scheduling Algorithms Main idea — decide which process to run next (when running process exits, becomes blocked, or is interrupted). Goal is to make decisions in a way that meets system objectives (minimize average turnaround time, maximize CPU usage, etc.) Some simple algorithms discussed last time. Some (e.g., FCFS) have limited practical value, but notice how many there are — many ways to approach the problem of "who's next?"

Slide 2





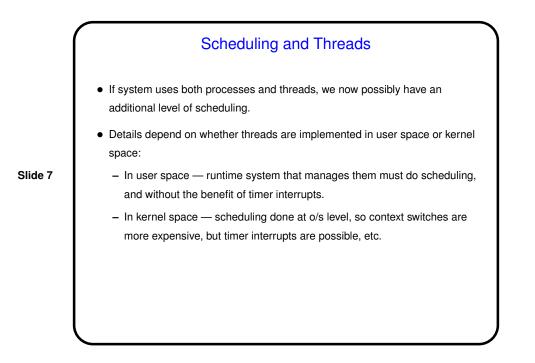


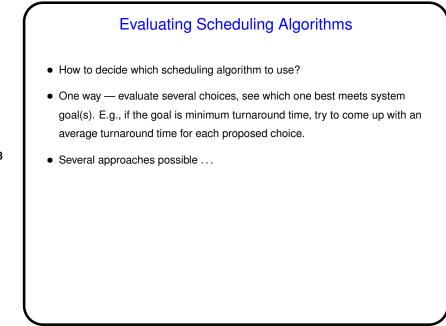
Scheduling in Real-Time Systems
"Real-time system" — system in which events must ("hard real time") or should ("soft real time") be handled by some deadline. Often events to be handled are periodic, and we know how often they arrive and how long they take to process.
Role of scheduler in such systems 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?
Suppose we have *m* types of events, and event type *i* has period *P_i* and time-to-handle *C_i*.
General formula on p. 149; can be derived by extending from case where *m* = 1 …
Complex topic, see chapter 7 for more info.

Slide 5

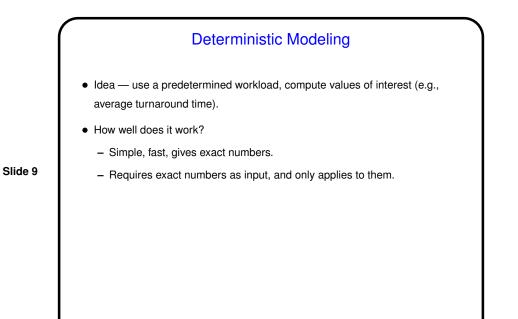
Factor in process's owner in deciding which process to pick. I.e., if two "equal" users, schedule processes such that user A's processes get about as much time as those of user B.

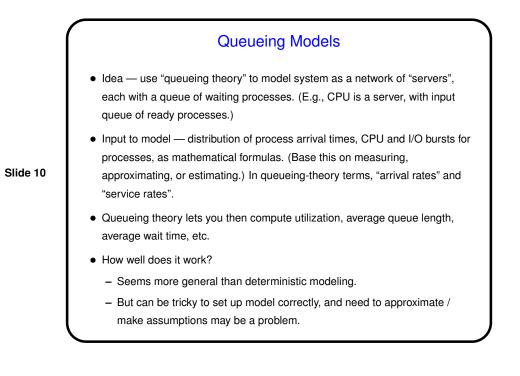


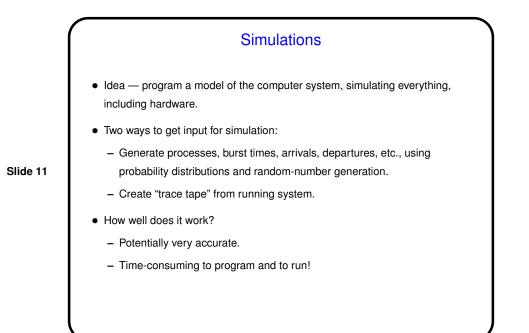


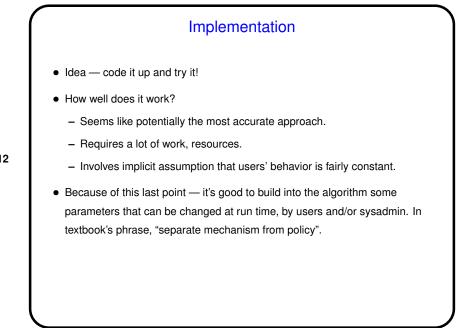


Slide 8









Slide 12

