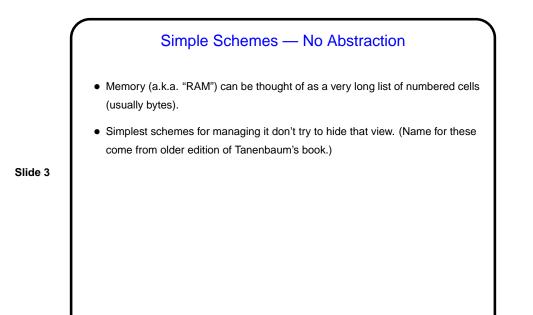
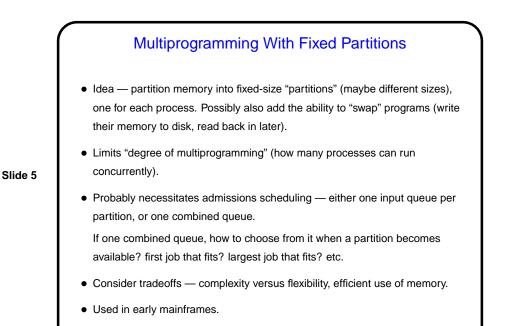


Memory Management, Introduction One job of operating system is to "manage memory" — assign sections of

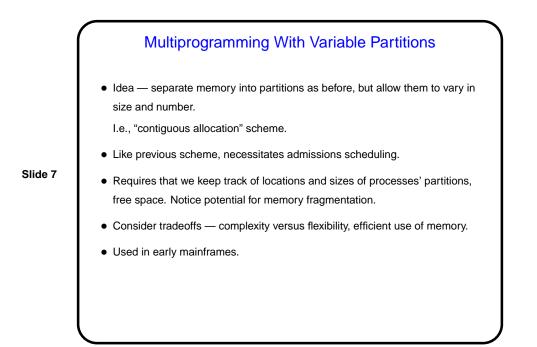
- One job of operating system is to manage memory assign sections of main memory to processes, keep track of who has what, protect processes' memory from other processes.
- As with CPU scheduling, we'll look at several schemes, starting with the very simple. For each scheme, think about how well it solves the problem, how it compares to others.
- As with processes, there's a tradeoff between simplicity and providing a nice abstraction to user programs.

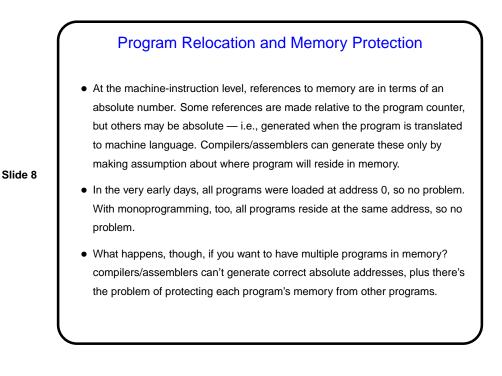


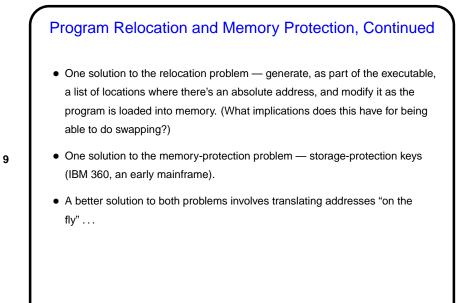
Monoprogramming
Idea — only one user program/process at a time, no swapping or paging. Only decision to make is how much memory to devote to o/s itself, where to put it.
Consider tradeoffs — complexity versus flexibility, efficient use of memory.
Used in very early mainframes, MS-DOS; still used in some embedded systems.



Sidebar: Three-Level Scheduling • Basic idea — break up problem of scheduling (batch) work into three parts: - Admissions scheduling - choose from input queue which jobs to "let into the system" (create processes for). - Memory scheduling - choose from among processes in system which to keep in memory, which to "swap out" to disk. - CPU scheduling - choose from among processes in memory which to actually run. · Points to consider: - Are there advantages to limiting how many processes, how many in memory? What criteria could we use? - Are there advantages to the explicit three-level scheme? - Would this (or a variant) work for interactive systems? - Do all three schedulers have to be efficient?







Sidebar: The "Address Space" Abstraction

- Basic idea is somewhat analogous to process abstraction, in which each process has its own simulated CPU. Here, each process has its own simulated memory.
- As with processes, implementing this abstraction is part of what an operating system can/should do.
- Usually, though, o/s needs help from hardware ...

