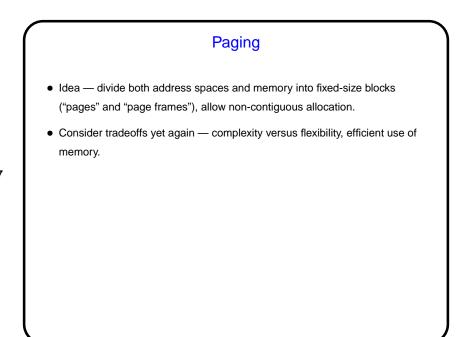


Simple Memory Management — Recap

- Contiguous-allocation schemes are simple to understand, implement.
- But they're not very flexible process's memory must be contiguous, swapping is all-or-nothing.

Slide 6

• Can we do better? yes, by relaxing one or both of those requirements — "paging".

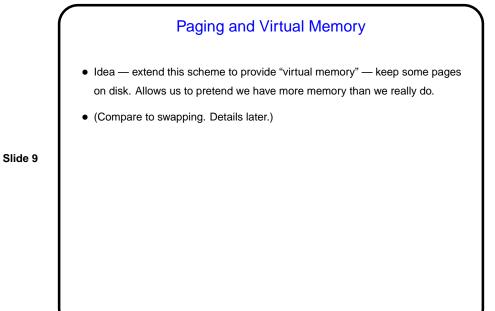


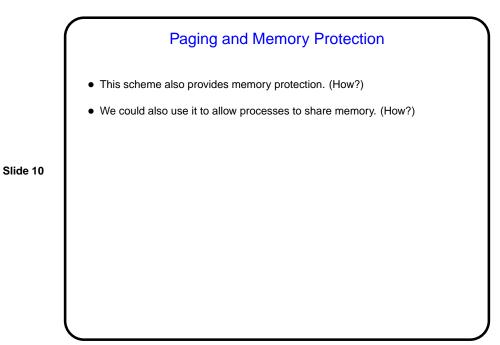
Slide 7

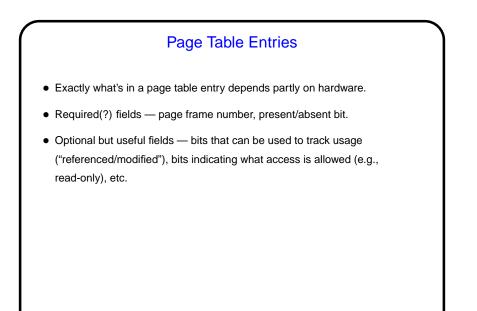
Paging — Mapping Program to Physical Addresses

- One consequence mapping from program addresses to physical addresses is much more complicated.
- How? "page table" for each process maps pages of address space to page frames; use this to calculate physical address from program address. (Are there page sizes for which this is easier?)

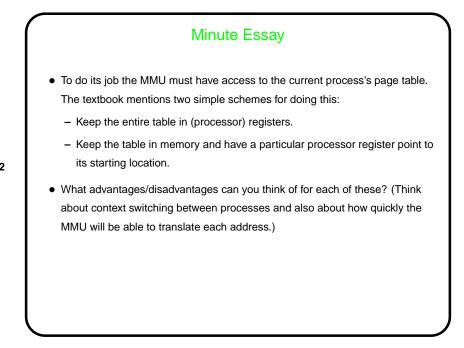
- As with base/limit scheme, makes more sense to implement this in MMU. (Notice again interaction between hardware design and o/s design.)
- Could let page table size vary, but easier to make them all the same (i.e., each process has the same size address space), have a bit to indicate valid/invalid for each entry. Attempt to access page with invalid bit "page fault".







Slide 11





• The first scheme almost surely makes for faster translations, but for a large page table it will require a lot of registers, which would make context switches slow. The second scheme won't slow down context switches, but as stated it isn't going to make for fast translation.