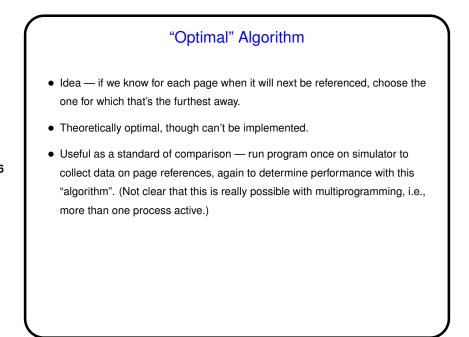
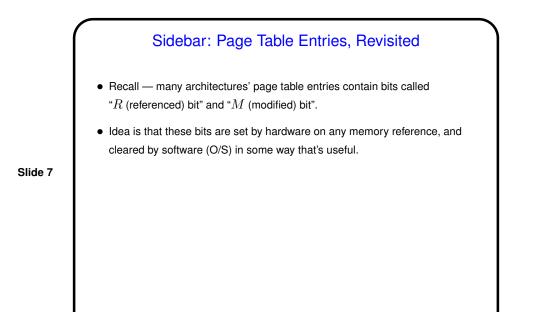


## Page Fault Interrupts, Continued One possible cause — an address that's not valid. You know (sort of) what happens then ... Another cause — an address that's valid, but the page is on disk rather than in real memory. So — do I/O to read it in. Where to put it? If there's a free page frame, choice is easy. What if there's not?



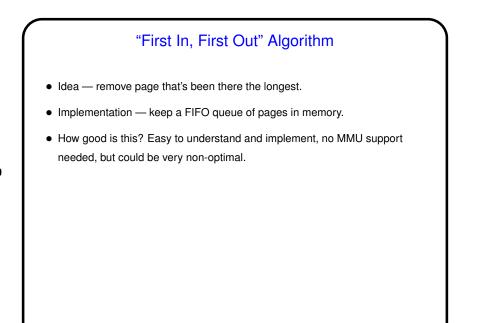
- Processing a page fault can involve finding a free page frame. Would be easy if the current set of processes aren't taking up all of main memory, but what if they are? Must steal a page frame from someone. How to choose one?
- Several ways to make choice (as with CPU scheduling) "page replacement algorithms".
- "Good" algorithms are those that result in few page faults. (What happens if there are many page faults?)
- Choice usually constrained by what MMU provides (though that is influenced by what would help O/S designers).
- Many choices (no surprise, right?) ....





"Not Recently Used" Algorithm
Idea — choose a page that hasn't been referenced/modified recently, hoping it won't be referenced again soon.
Implementation — use page table's *R* and *M* bits, group pages into four classes: *R* = 0, *M* = 0. *R* = 0, *M* = 1. *R* = 1, *M* = 0. *R* = 1, *M* = 1.

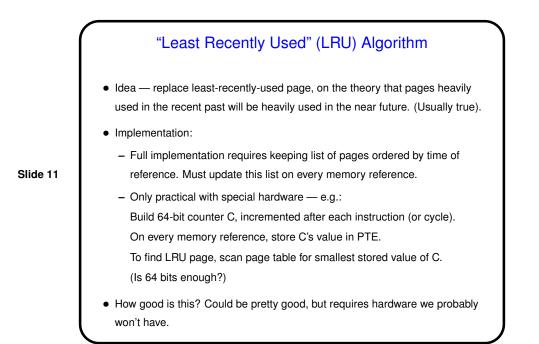
Choose page to replace at random from first non-empty class.
How good is this? Easy to understand, reasonably efficient to implement, often gives adequate performance.

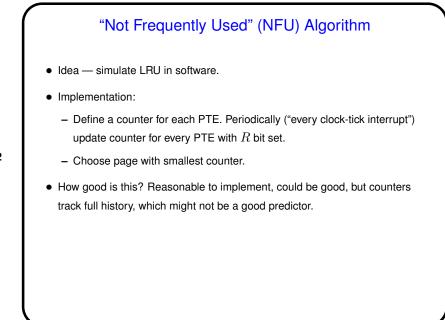


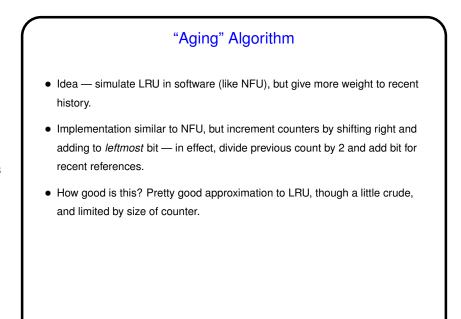
## "Second Chance" Algorithm

- Idea modify FIFO algorithm so it only removes the oldest page if it looks inactive.
- Implementation use page table's R and M bits, also keep FIFO queue. Choose page from head of FIFO queue, *but* if its R bit is set, just clear R bit and put page back on queue.

- Variant "clock" algorithm (same idea, keeps pages in a circular queue).
- How good is this? Easy to understand and implement, probably better than FIFO.







## Sidebar: Working Sets Most programs exhibit "locality of reference", so a process usually isn't using all its pages. A process's "working set" is the pages it's using. Changes over time, with size a function of time and also of how far back we look.

