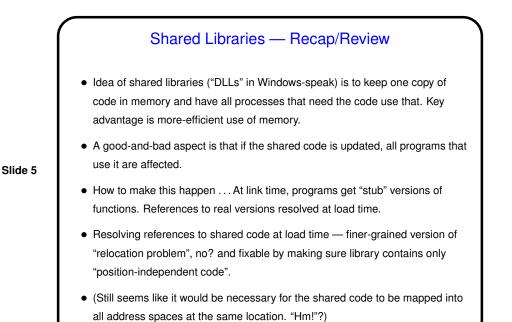


Slide 3

## Memory Management Versus(?) Caching Main memory (RAM) is not such a scarce resource on most current mainstream systems, so managing it carefully isn't as important (though it still *could* be). What might matter more from an application-programming perspective is not whether the whole program and its data fit into RAM (likely) but whether the current working set fits into cache. Performance differences can be noticeable! Example: matrix multiplication, where the difference between the "naive" (obvious) implementation and one that works by blocks can be noticeable, almost surely because of differences in what fits in cache.



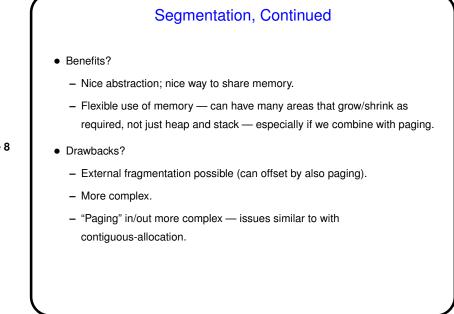
## Memory-Mapped File I/O

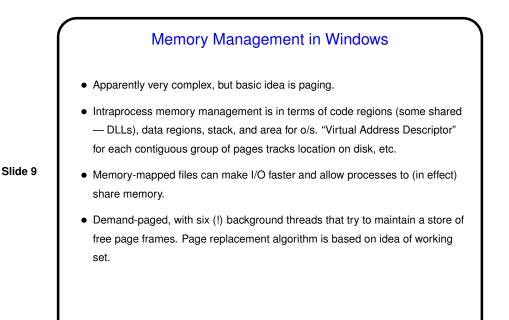
- Worth mentioning here that some systems also provide a mechanism (e.g., via system calls) to allow reading/writing whole files into/from memory. If there's enough memory, this could improve performance.
- Example of how this works in Linux man page for mmap.



- Idea make program address "two-dimensional" / separate address space into logical parts. So a virtual address has two parts, a segment and an offset.
- To map virtual address to memory location, need "segment table", like page table except each entry also requires a length/limit field. (So this is like a cross between contiguous-allocation schemes and paging.)

Slide 7





Memory Management in UNIX/Linux
Very early UNIX used contiguous-allocation or segmentation with swapping. Later versions use paging. Linux uses multi-level page tables; details depend on architecture (e.g., three levels for Alpha, two for Pentium).
Intraprocess memory management is in terms of text (code) segment, data segment, and stack segment. Linux reserves part of address space for O/S. For each contiguous group of pages, "vm\_area\_struct" tracks location on disk, etc.
Memory-mapped files can make I/O faster and allow processes to (in effect) share memory.
Demand-paged, with background process ("page daemon") that tries to maintain a store of free page frames. Page replacement algorithms are mostly variants of clock algorithm.

