

Slide 3

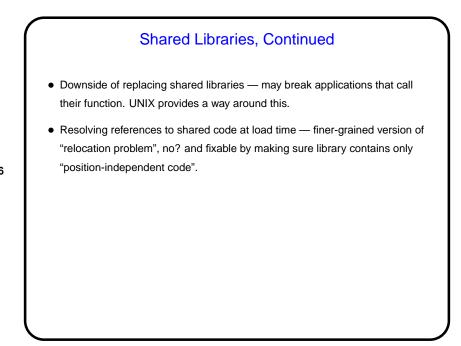
Sharing Pages, Continued One use for shared pages is multiple processes running the same program. What about sharing code at a level below whole programs (UNIX "shared libraries", Windows DLLs)? Seems attractive; are there potential problems?

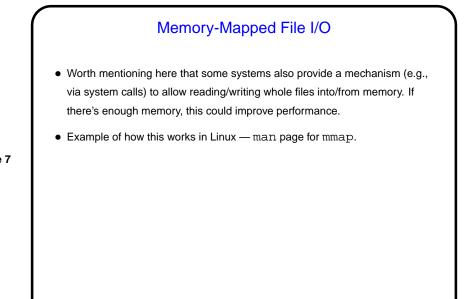


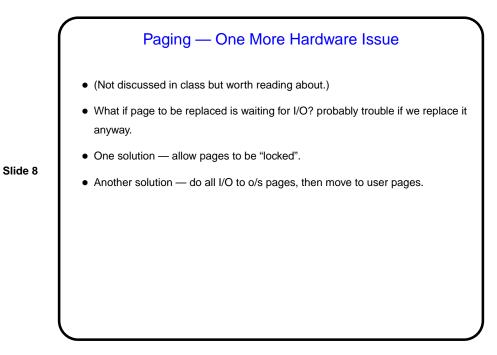
 One attraction is somewhat obvious — if code for library functions (e.g., printf) is statically linked into every program that uses it, programs need more memory — seems wasteful if processes can share one copy of code in memory.

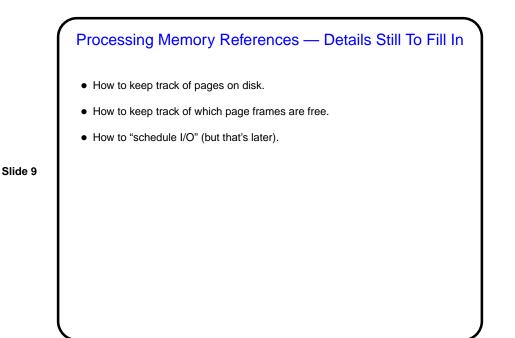
Slide 5

- Another attraction is that library code can be updated independently of programs that use it. (Is there a downside to that?)
- How to make this happen ... At link time, programs get "stub" versions of functions. References to real versions resolved at load time. Does this remind you of anything? and suggest a possible problem? how to fix?

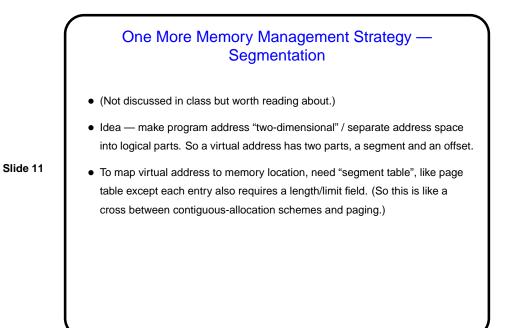


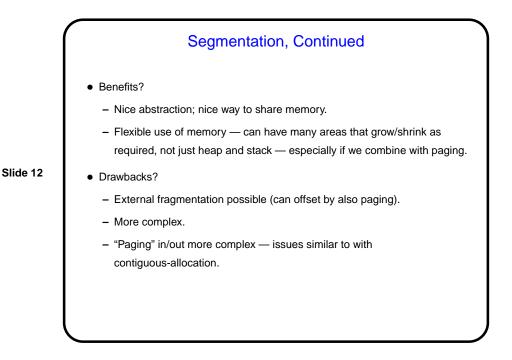


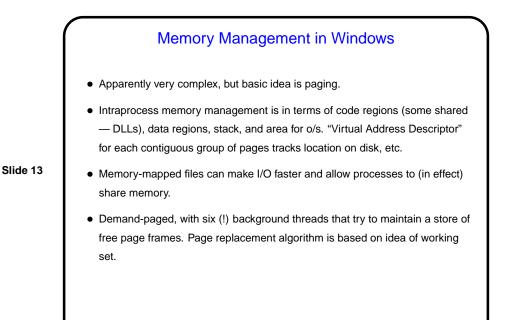




Keeping Track of Pages on Disk To implement virtual memory, need space on disk to keep pages not in main memory. Reserve part of disk for this purpose ("swap space"); (conceptually) divide it into page-sized chunks. How to keep track of which pages are where? Slide 10 One approach — give each process a contiguous piece of swap space. Advantages/disadvantages? Another approach — assign chunks of swap space individually. Advantages/disadvantages? Either way — processes must know where "their" pages are (via page table and some other data structure), operating system must know where free slots are (in memory and in swap space).







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.

