

Slide 1

Administrivia

- As mentioned in e-mail — Homework 4 on Web, due next Friday.

Slide 2

One More Memory Management Strategy — Segmentation

- 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.)

Segmentation, Continued

Slide 3

- Benefits?
 - Nice abstraction; nice way to share memory.
 - Flexible use of memory — can have many areas that grow/shrink as required, not just heap and stack — especially if we combine with paging.
- Drawbacks?
 - External fragmentation possible (can offset by also paging).
 - More complex.
 - “Paging” in/out more complex — issues similar to with contiguous-allocation.

Memory Management in Windows

Slide 4

- Apparently very complex, but basic idea is paging.
- Intraprocess memory management is in terms of code regions (some shared — DLLs), data regions, stack, and area for o/s. “Virtual Address Descriptor” for each contiguous group of pages tracks location on disk, etc.
- Memory-mapped files can make I/O faster and allow processes to (in effect) share memory.
- Demand-paged, with six (!) background threads that try to maintain a store of free page frames. Page replacement algorithm is based on idea of working set.
- (Also see comment on p. 823.)

Slide 5

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.

Slide 6

Minute Essay

- Anything about memory management you'd like to hear more about / have clarified?