Slide 1

Administrivia

- Everything turned in has been graded (!). Scores on the midterm were overall good.
- I will send out (by e-mail) a summary of your grades and a somewhat tentative letter grade. The only people who aren't passing are the ones who aren't consistently turning work in.

Slide 2

Memory Management — Review

- The problem we're solving — partition physical memory among processes. Two related issues — program relocation and memory protection — both nicely solved by defining "address space" abstraction and implementing with help from hardware (MMU).
- Contiguous-allocation schemes are simple but not very flexible.
- Paging is more flexible but more complex.

Slide 3

Paging — Recap

- Idea — divide both address spaces and memory into fixed-size blocks ("pages" and "page frames"), allow non-contiguous allocation.
- Makes for a much more flexible system but at a cost in complexity — keeping track of a process's memory requires a "page table" to be used by both hardware (MMU) and software (O/S).

Slide 4

Page Tables — Performance Issues

- One possibility is to keep the whole page table for the current process in registers. Could possibly use general-purpose registers for this but likely would not. Should make for fast translation of addresses, but — is this really feasible for a large table? and what about context switches?
- Another possibility is to keep the process table in memory and just have one register (probably a special-purpose one) point to it. Cost/benefit tradeoffs here seem like the opposite of the first scheme, no?

  The big downside is slow lookup. Can be mitigated with a "translation lookaside buffer" (TLB) — special-purpose cache.

Slide 5

Paging — Feasibility Issues

- Clearly page tables can be big. How to make this feasible?
- One approach — multilevel page tables.
- Another approach — inverted page tables (one entry per page frame).

Slide 6

Paging and Virtual Memory

- Idea — if we don't have room for all pages of all processes in main memory, keep some on disk ("pretend we have more memory than we really do").
- Or a simpler view: All address spaces live in secondary memory / swap space / backing store, and we "page in" as needed (demand paging).
- (Aside: Why are we even bothering? Can't the processor(s) access disk? Yes, but . . .)
- Making this work requires help from both hardware (MMU) and software (operating system).
- (To be continued.)

Slide 7

Minute Essay

- How did the midterm compare to your expectations (topics, level of difficulty, . . .).