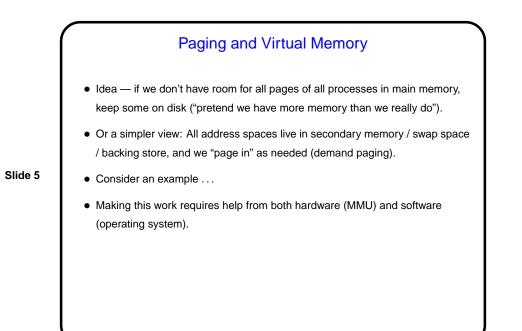
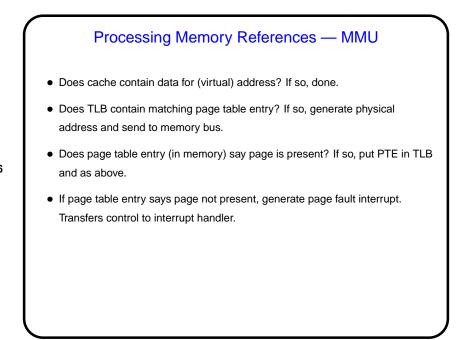


Large Address Spaces

- Clearly page tables can be big. How to make this feasible?
- One approach multilevel page tables. Figure on p. 208.
- Another approach inverted page tables (one entry per page frame). Figure on p. 214.







- Is page on disk or invalid (based on entry in process table, or other o/s data structure)? If invalid, error terminate process.
- Is there a free page frame? If not, choose one to steal. If it needs to be saved to disk, start I/O to do that. Update process table, PTE, etc., for "victim" process. Block process until I/O done.
- Start I/O to bring needed page in from swap space (or zero out new page). If I/O needed, block process until done.
- Update process table, etc., for process that caused the page fault, and restart it at instruction that generated page fault.



- How to keep track of pages on disk.
- How to keep track of which page frames are free.
- How to "schedule I/O" (but that's later).
- How to choose a page frame to "steal".

