Slide 1

Slide 2

CSCI 4320

## Administrivia

- Next week I will be at a conference. The plan is to have no class Tuesday and a guest lecturer Thursday. Check the "Lecture topics and assignments" page for possible changes.
- There will be a short homework on Chapter 1. It will be on the Web soon, linked from the "Lecture topics and assignments" page. Due date will be September 16.

# Hardware Overview, Recap

- Simplified view of basic hardware components (processor, memory, I/O devices) — their purpose from a user's perspective, how they work from an assembly-language programmer's perspective.
- Interaction between hardware design and o/s design what the hardware
- can do influences o/s design, but what o/s designers want also influence hardware design.

# Minute Essay From Last Lecture

- Question: I once had a learning experience about "how DOS is different from a real o/s". Summary version: A program using pointers (possibly uninitialized) caused the whole machine to lock up and need to be power-cycled. What do you think went wrong?
- Slide 3 Answer: The program changed memory at the addresses pointed to by the uninitialized pointers and this memory was being used by the o/s, possibly to store something related to interrupt handling. A "real" o/s wouldn't allow this!

# Operating System Services, Again Process management. Memory management. I/O subsystem. File systems. Slide 4 Security. Shell.

Slide 5

September 4, 2003

CSCI 4320

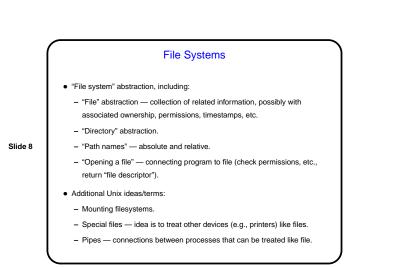
September 4, 2003

# **Process Management** I/O Subsystem · Encapsulates messy low-level details. · "Process" abstraction to represent one of a collection of "things happening at the same time". • Allows sharing of I/O devices among programs/users. A working definition — "program in execution" (program code plus associated variables, sequence of states tracking progress through code and changes in variables). Slide 7 • "Concurrent" execution via interleaving of actions. In effect, each process has a "virtual CPU", with the actual CPU repeatedly suspending one process to work on another ("context switch"). • O/s must provide a way to manage this, including ways to create processes, allow/force them to terminate, communicate among them (e.g., to coordinate/synchronize).

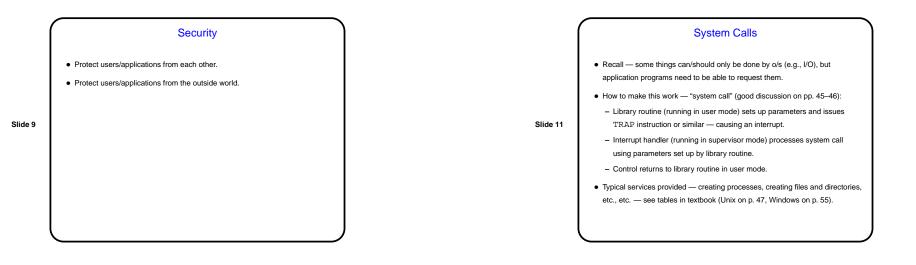
Memory Management

Managing physical memory:

- How to divide it up among processes/programs/users each has an "address space" of memory it can access.
- How to protect each process's memory from other processes (requires h/w support, but managed by o/s).
- Slide 6
- Managing address spaces (virtual memory):
  - Originally, address space limited by size of physical memory.
- "Virtual memory" allows bigger address spaces, by shuffling data between disk and physical memory.



CSCI 4320

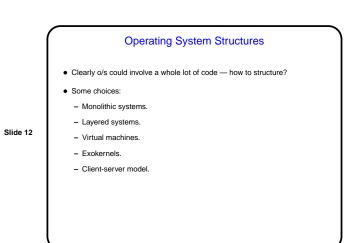




- History early batch systems had to interpret "control cards"; modern equivalent is to interpret "commands" (usually interactive).
- Not technically part of o/s, but important and related.
- Typical shell functionality:

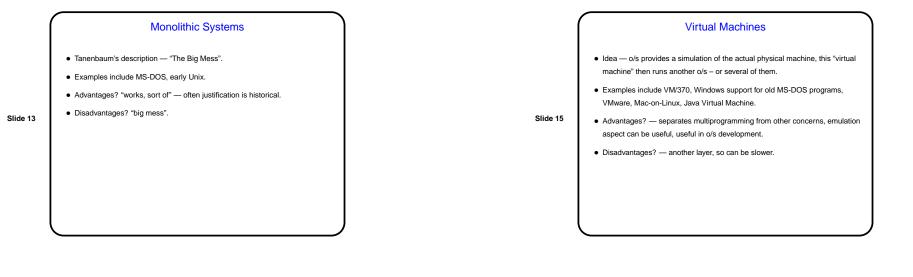
Slide 10

- Invocation of programs (optionally in background).
- Input/output redirection.
- Program-to-program connections (pipes).
- "Wildcard" capability.
- Scripting capability.
- $\bullet$  Examples MS-DOS <code>command.com</code>; <code>Unix sh</code>, <code>bash</code>, <code>csh</code>, <code>tcsh</code>, <code>ksh</code>, <code>zsh</code>, …



Slide 14

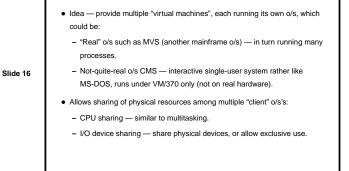
CSCI 4320





- Idea use layers of abstraction, just as one structures application programs.
- Examples include THE, MULTICS, OS/2, Windows NT (more so in early releases).
- Advantages? nice separation of concerns, modularity.
- Disadvantages? tricky to plan layers, performance can be slow.

VM/370



Slide 17

# September 4, 2003

# VM/370, Continued

- How does this work? briefly:
  - Client o/s's run native code, request o/s services in the usual way (interrupt or system call).
- Interrupt handler is part of VM/370 so it processes I/O
- requests/interrupts, errors, etc.
- Client o/s system code runs in simulated supervisor mode (really user mode).
- Successors to VM/370 (VM/ESA, z/VM) currently being used to run many copies of Linux on a mainframe (!).

# Minute Essay

- This wraps up lectures on chapter 1; is there anything that was particularly unclear or you want to know more about?
- Tell me one interesting or useful thing you've learned from the reading so far.

Slide 18