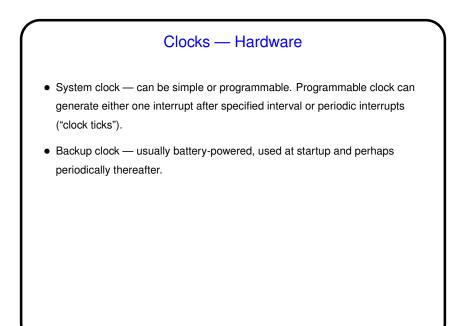
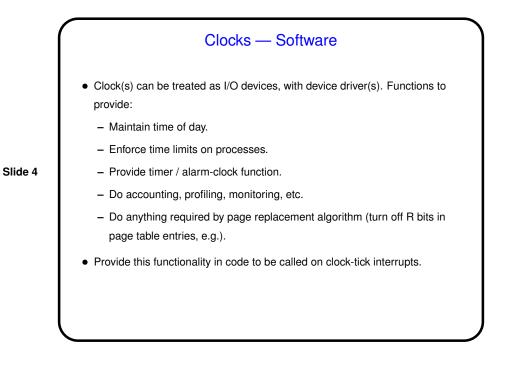


I/O Continued — Device Specifics

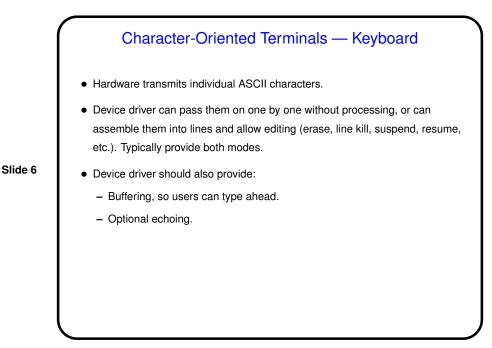
• Next, a tour of major classes of devices. For each, we look first at what the hardware can typically do, and then at what kinds of device-driver functionality we might want to provide.

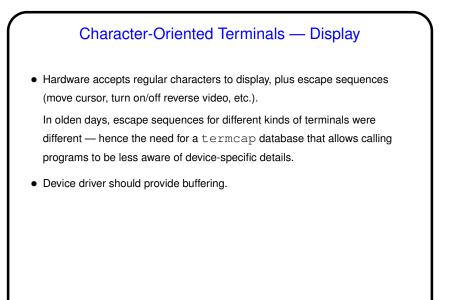






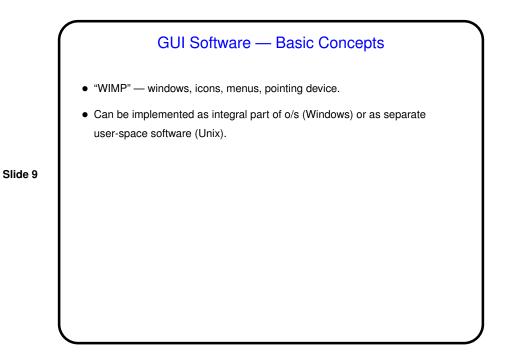
- Hardware consists of character-oriented display (fixed number of rows and columns) and keyboard, connected to CPU by serial line.
- Actual hardware no longer common (except in mainframe world), but emulated in software (e.g., Unix xterm) so old programs still work. (Why does anyone care? some of those old programs are still useful — e.g., text editors — and usually very stable.)

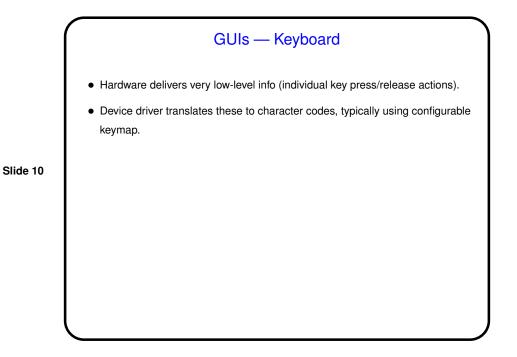


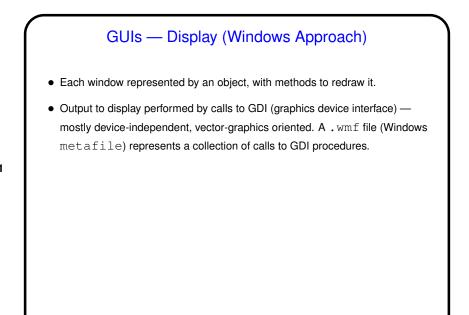


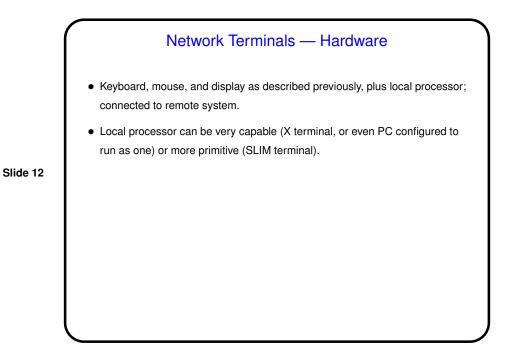
GUIs — Hardware Overview
PC keyboard — sends very low-level detailed info (keys pressed/released); contrast with keyboard for character-oriented terminal.
Mouse — sends (delta-x, delta-y, button status) events.
Display can be vector graphics device (rare now, works in terms of lines, points, text) or raster graphics device (works in terms of pixels). Raster graphics device uses graphics adapter, which includes:

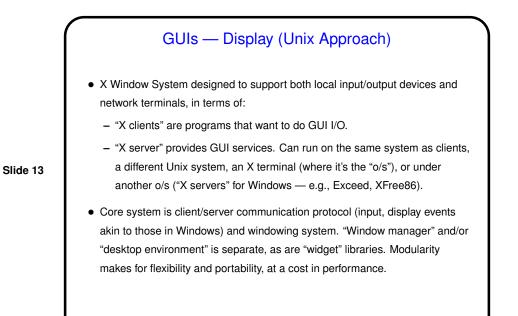
Video RAM, mapped to part of memory.
Video controller that translates contents of video RAM to display. Has two modes, text and bitmap.



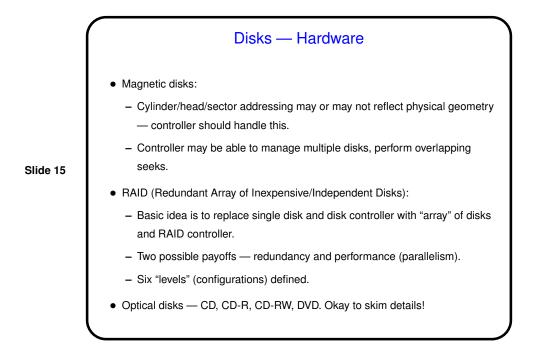


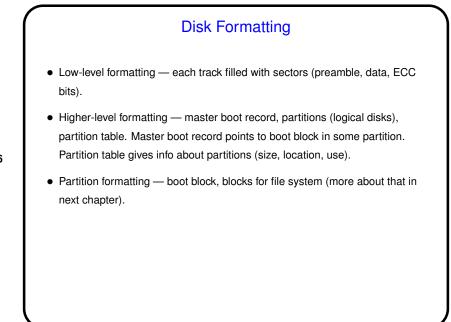


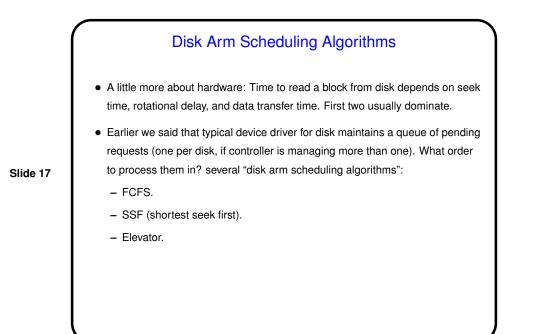




GUI-Based Programming Input from keyboard and mouse captured by o/s and turned into messages to process owning appropriate window. Typical structure of GUI-based program is a loop to receive and dispatch these messages — "event-driven" style of programming. Details vary between Windows and X, but overall idea is similar. See example programs in textbook.







Disk Error Handling
Almost all disks have sectors with defects. Some controllers can recognize them (repeated failures) and avoid them; if not, o/s (device driver) must do this.
Other kinds of errors also possible, e.g., failure to correctly position read/write head; also must be handled either by controller (if possible) or o/s.

