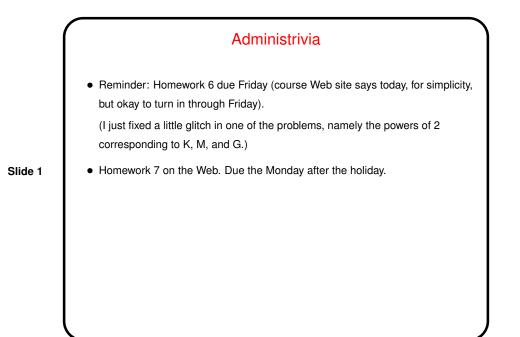
Slide 2

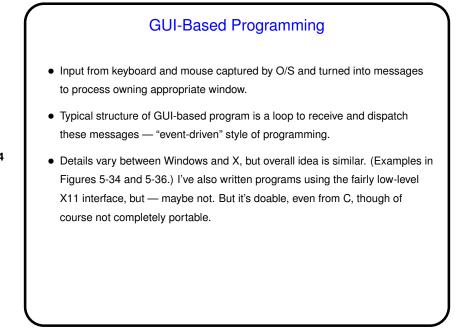


GUIs — Review/Recap
Keyboard: Hardware delivers very low-level info (individual key press/release actions). Device driver translates these to character codes, typically using configurable keymap.
Mouse: Hardware delivers very low-level info (change in coordinates, status of buttons).
Display: Hardware can be fairly simple ("raster graphics") or pretty sophisticated (independent processor capable of independent operation). How the O/S communicates with it varies by platform — Windows approach explicitly object-oriented, traditional UNIX/Linux based on protocol (X11) that works over network too.

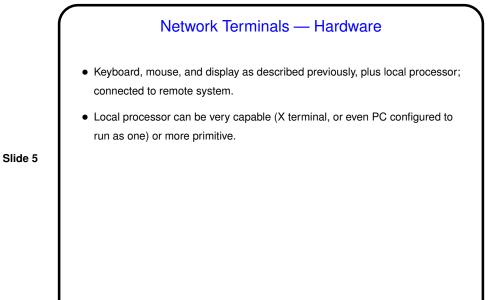


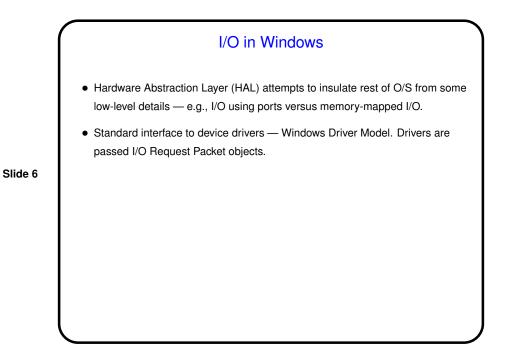
- Of course, many examples of software using this kind of device.
- Libraries for writing such software vary by language:
- Java and Scala include lots of library classes, mostly fairly high-level/abstract.
- Nothing standard in C, but most platforms offer various libraries. Lowest-level one in UNIXworld is "X11".

Slide 3



Slide 4





I/O in UNIX/Linux

Access to devices provided by special files (normally in /dev/*), to provide uniform interface for callers. Two categories, block and character. Each defines interface (set of functions) to device driver. Associated with each special file are major and minor device numbers, with major device number used to locate specific function. (Look at some output of ls -l /dev.)

Slide 7

- For block devices, buffer cache contains blocks recently/frequently used.
- For character devices, optional line-discipline layer provides some of what we described for text-terminal keyboard driver.
- Streams provide additional layer of abstraction for callers can interface to files, terminals, etc. (This is what you access with *scanf, *printf.)



- I mentioned the pseudofilesystem /proc? which supposedly you can read/write just as if it were a file?
- I wrote some throwaway code to access "files" within it and learned(?) that while C stream I/O (fopen, fgetc, etc.) didn't work well, the lower-level routines (open, read, etc.) did.

Slide 8

