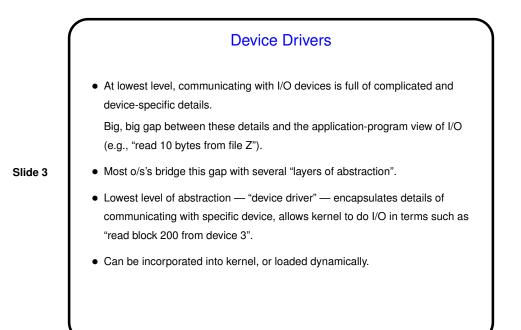
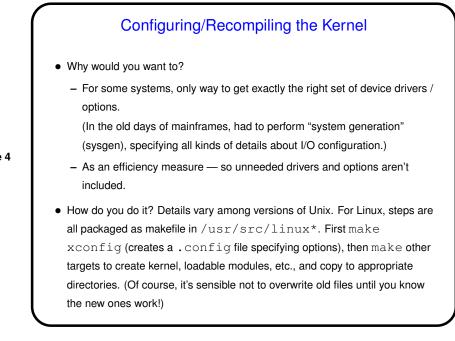


Kernels

- In general (as taught in o/s courses), the job of the operating system is to
 - Provide "virtual machine" abstractions processes, filesystems, etc.
 - Manage hardware resources on behalf of user programs.

- Kernel is heart of o/s the part that's always loaded (in virtual memory anyway). Responsible for many system activities:
 - Managing processes (timesharing, address-space protection).
 - Providing interprocess communication (signals and semaphores, pipes).
 - Implementing virtual memory (paging, swapping, etc.).
 - Managing filesystems.



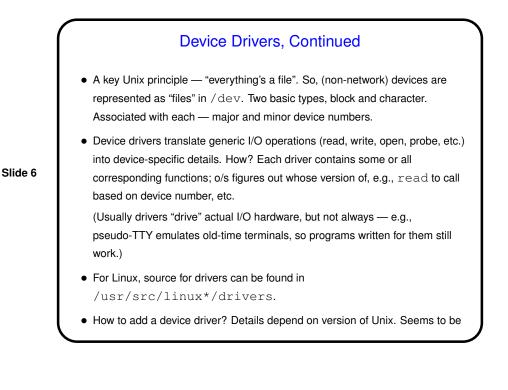


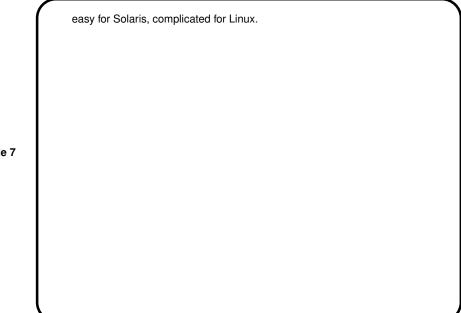
Tuning the Kernel

• Some systems also let you set some kernel parameters (e.g., maximum number of open files a process can have) "on the fly". Details vary among versions of Unix. For Linux, many are set by overwriting files in /proc. To do this "permanently", must add the overwrite commands to startup scripts (more about that later).

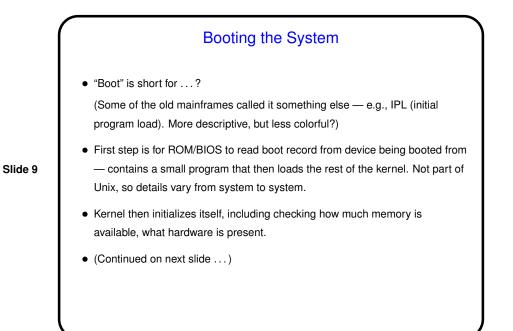
Slide 5

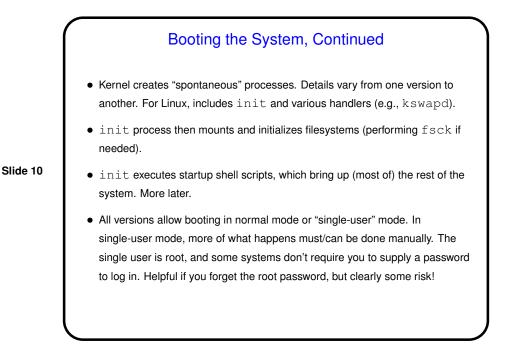
• (Aside: /proc filesystem is a sort of phony filesystem allowing you to examine/change o/s data structures. Take a look sometime!)





Loadable Kernel Modules Idea is to allow adding/removing kernel services (such as device drivers) without recompiling kernel. Support for this varies among versions of Unix; very good in Solaris, okay in Linux, etc. For Linux, lsmod lists currently loaded modules. insmod to add, rmmod to remove, modprobe to add/remove in a way that deals with dependencies, pre- and post- stuff.





Boot Loaders Revisited

- On PC hardware, MBR contains program that loads "second-stage boot loader" from selected disk partition, which then loads the rest ...
- For Linux systems, boot loaders include lilo and grub. Both make possible multiboot systems. To change options/configuration, must edit appropriate file and then reinstall boot loader.

