

Administrivia

Slide 1

- *Please do not* reboot the machines in HAS 340! If a previous user has left a machine in the "locked by screensaver" state, you can bail out by pressing control-alt-backspace to restart X (the graphical subsystem) without disturbing background processes.
- Are your prox cards giving you access to the labs? Supposedly all known problems have been resolved.
- Reminder: Homework 1 due Wednesday.

Recap — Operating System Functionality

Slide 2

- Two goals:
 - Bridge gap between what hardware will do (very primitive) and "virtual machine" useful for application-level programs.
 - Manage physical resources on behalf of multiple applications / users.
- Major functions:
 - Process management.
 - Memory management.
 - I/O subsystem.
 - File systems.
 - Security.
 - Shell.

Operating System Structures

Slide 3

- Clearly o/s could involve a whole lot of code (table of representative sizes on p. 771) — how to structure?
- Some choices:
 - Monolithic systems.
 - Layered systems.
 - Virtual machines.
 - Exokernels.
 - Client-server model.

Monolithic Systems

Slide 4

- Tanenbaum's description — "The Big Mess".
- Examples include MS-DOS, early Unix.
- Advantages? "works, sort of" — often justification is historical.
- Disadvantages? "big mess".

Layered Systems

Slide 5

- 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.

Virtual Machines

Slide 6

- Idea — o/s provides a simulation of the actual physical machine, this “virtual machine” then runs another o/s – or several of them.
- Examples include VM/370, Windows support for old MS-DOS programs, VMware, Mac-on-Linux, Java Virtual Machine.
- Advantages? — separates multiprogramming from other concerns, emulation aspect can be useful, useful in o/s development.
- Disadvantages? — another layer, so can be slower.

Slide 7

VM/370

- Idea — provide multiple “virtual machines”, each running its own o/s, which could be:
 - “Real” o/s such as MVS (another mainframe o/s) — in turn running many processes.
 - Not-quite-real o/s CMS — interactive single-user system rather like MS-DOS, runs under VM/370 only (not on real hardware).
- Allows sharing of physical resources among multiple “client” o/s's:
 - CPU sharing — similar to multitasking.
 - I/O device sharing — share physical devices, or allow exclusive use.

Slide 8

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

- Tell me something you've learned from what you've read in the textbook so far.
- (You can also tell me here about any problems you're having with Homework 1.)

Slide 9