

## Administrivia

- Reminder: Homework 1 due today at 5pm (programming problem by e-mail, rest hardcopy). I should be around most of this afternoon if there are questions.

Slide 1

## Operating System Structures

- Clearly o/s could involve a whole lot of code — how to structure?
- Some choices:
  - Monolithic systems.
  - Layered systems.
  - Virtual machines.
  - Exokernels.
  - Client-server model.

Slide 2

### Monolithic Systems

- Tanenbaum's description — "The Big Mess".
- Examples include MS-DOS, early Unix.
- Advantages? "works, sort of" — often justification is historical.
- Disadvantages? "big mess". (Not everyone agrees, though.)

Slide 3

### Layered Systems

- 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.

Slide 4

## Virtual Machines

Slide 5

- 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.

## VM/370

Slide 6

- 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.

### VM/370, Continued

Slide 7

- 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

Slide 8

- What do you think you learned from doing Homework 1? was anything particularly easy / difficult / interesting / annoying?