OPERATING SYSTEMS

STRUCTURES

Part 2

OPERATING SYSTEM STRUCTURES — System Calls

2. File management

- Create / delete files
- Open / close
- Read / write
- Get / set file attributes

OPERATING SYSTEM STRUCTURES — System Calls

3. Device management

- Request / release device
- Read / write
- Get / set device attributes
- Logically attach or detach a device

OPERATING SYSTEM STRUCTURES — System Calls

4. Information maintenance

- Get / set time or date
- Get / set system data
- Get process, file or device attributes
- Set process, file or device attributes

OPERATING SYSTEM STRUCTURES — System Calls

5. Communication

- Create / delete communication connection
- Send /receive messages
- Transfer status information
- Attach or detach remote devices

OPERATING SYSTEM STRUCTURES — System Calls

5. Communication

There are two ways of communications between processes:

i. Message passing (eg. Chat programs)

ii. Shared memory

OPERATING SYSTEM STRUCTURES — System Calls

Message passing:

communicating processes exchange information to transfer data

Eg. Process A needs to get some information from process B

-send (A, message)
-Receive (B, message)

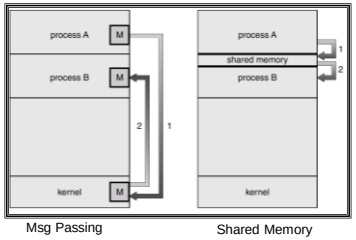
OPERATING SYSTEM STRUCTURES — System Calls

Shared memory:

processes use "shared memory create" and "shared memory attach" system calls to create and gain access to regions of memory used by other processes

OPERATING SYSTEM STRUCTURES — System Calls

Two ways of passing data between programs.



Msg Passing Shared Memory

OPERATING SYSTEM STRUCTURES — System Calls

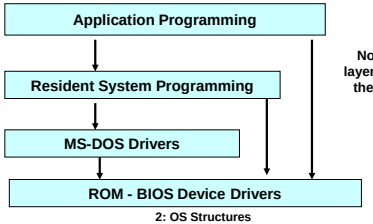
Message passing:
- Useful for exchanging smaller amounts of data (no conflicts)
- easier to implement when inter-computer communication is needed

Shared memory:
-Allows maximum speed & convenience since communication takes place at memory speeds
-Needs to consider protection & synchronization between the processes sharing the memory.

OPERATING SYSTEM STRUCTURES — How An Operating System Is Put Together

A SIMPLE STRUCTURE:

Example of MS-DOS.



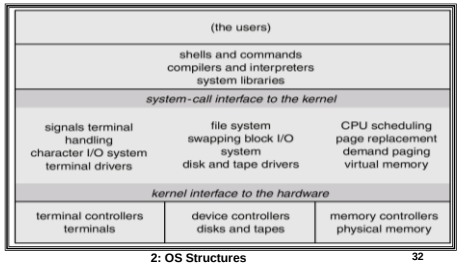
Note how all layers can touch the hardware.

OPERATING SYSTEM STRUCTURES

How An Operating System Is Put Together

A LAYERED STRUCTURE:

Example of UNIX.

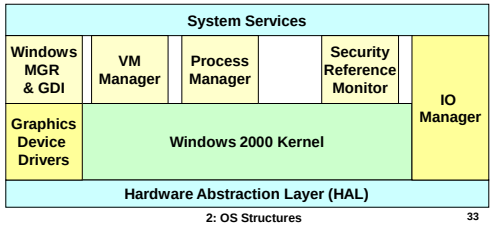


OPERATING SYSTEM STRUCTURES

How An Operating System Is Put Together

A LAYERED STRUCTURE:

Example of Windows 2000.

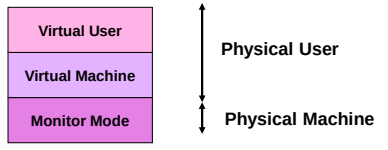


OPERATING SYSTEM STRUCTURES

Virtual Machine

In a Virtual Machine - each process "seems" to execute on its own processor with its own memory, devices, etc.

- The resources of the physical machine are shared. Virtual devices are sliced out of the physical ones. Virtual disks are subsets of physical ones.
- Useful for running different OS simultaneously on the same machine.
- Protection is excellent, but no sharing possible.
- Virtual privileged instructions are trapped.



OPERATING SYSTEM STRUCTURES

Virtual Machine

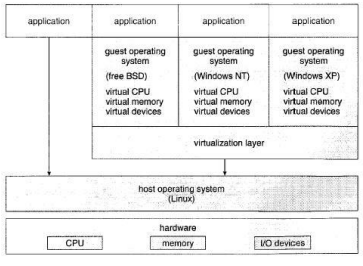
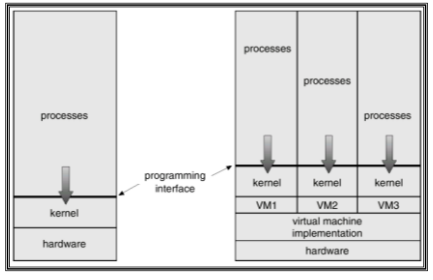


Figure 2.16 VMware architecture.

OPERATING SYSTEM STRUCTURES

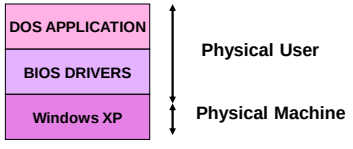
Virtual Machine



OPERATING SYSTEM STRUCTURES

Virtual Machine

Example of MS-DOS on top of Windows XP.

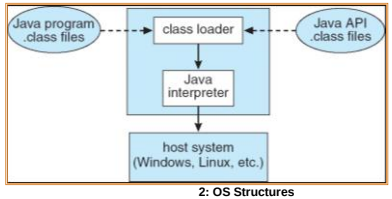


OPERATING SYSTEM STRUCTURES

Virtual Machine

Example of Java Virtual Machine

The Java Virtual Machine allows Java code to be portable between various hardware and OS platforms.



OPERATING SYSTEM STRUCTURES

WRAPUP

We've completed our second overview of an Operating System – this at the level of a high flying plane.

We've looked at the basic building blocks of an operating system – processes, memory management, file systems, and seen how they all connect together.

Now we'll get into more detailed explanations, spending considerable time on each of these pieces.