OPERATING SYSTEM STRUCTURES

System Calls

OPERATING SYSTEMS

STRUCTURES

Part 2

2: OS Structures 1

2: OS Structures

21

Open / closeRead / write

2. File management

· Get / set file attributes

· Create / delete files

2: OS Structures 22

OPERATING SYSTEM STRUCTURES

System Calls

OPERATING SYSTEM STRUCTURES

System Calls

OPERATING SYSTEM STRUCTURES

System Calls

3. Device management

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

4. Information maintenance

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

5. Communication

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

 2: OS Structures
 23
 2: OS Structures
 24
 2: OS Structures
 25

OPERATING SYSTEM STRUCTURES

System Calls

OPERATING SYSTEM STRUCTURES

System Calls

OPERATING SYSTEM STRUCTURES

System Calls

5. Communication

There are two ways of communications between processes:

- i. Message passing (eg. Chat programs)
- ii. Shared memory

2: OS Structures

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)

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

> 2: OS Structures 28

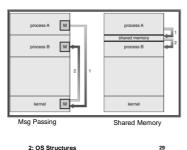
System Calls

2: OS Structures

27

OPERATING SYSTEM STRUCTURES

Two ways of passing data between programs.



2: OS Structures

OPERATING SYSTEM STRUCTURES

System Calls

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

Shared memory:

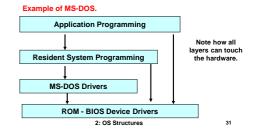
- -Allows maximum speed & convenience since communication takes place at memory speeds
- -Needs to consider protection & synchronization between the processes sharing the memory.

2: OS Structures

OPERATING SYSTEM STRUCTURES

How An Operating System Is Put **Together**

A SIMPLE STRUCTURE:



OPERATING SYSTEM STRUCTURES

How An Operating System Is Put Together

Virtual Machine

35

A LAYERED STRUCTURE: Example of UNIX.

> (the users) shells and commands compilers and interpreters system libraries system-call interface to the kernel file system CPU scheduling signals terminal swapping block I/O page replacement handling character I/O system system disk and tape drivers demand paging virtual memory terminal drivers kernel interface to the hardw terminal controllers device controllers memory controllers physical memory

> > 2: OS Structures

OPERATING SYSTEM STRUCTURES

How An Operating System Is Put Together

33

36

Virtual Machine

A LAYERED STRUCTURE:

Example of Windows 2000.

System Services				
Windows MGR & GDI	VM Manager	Process Manager	Security Reference Monitor	IO
Graphics Device Drivers	Windows 2000 Kernel			Manager
Hardware Abstraction Layer (HAL)				

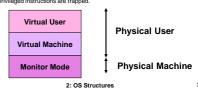
2: OS Structures

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.

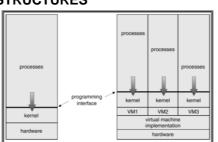


OPERATING SYSTEM STRUCTURES

application (Windows NT) application (Windows NT) are application (Windows NT) and application (Windows NT) are application (Windows NT) artificial memory virtual devices virtual devices virtual application (Linux) application (Windows NT) artificial memory virtual devices virtual application (Linux) application (Windows NT) artificial memory virtual devices virtual application (Linux) application (Windows NT) application (Windows NT) artificial memory virtual devices virtual devices virtual devices virtual devices (Linux) application appli

Figure 2.16 VMware architecture.
2: OS Structures

OPERATING SYSTEM STRUCTURES

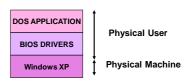


2: OS Structures

OPERATING SYSTEM STRUCTURES

Virtual Machine

Example of MS-DOS on top of Windows XP.



2: OS Structures 37

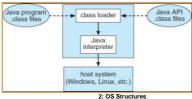
3

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.

2: OS Structures

39