

Operating System Organization

CS 256/456
Dept. of Computer Science, University of Rochester

1/26/2004 CSC 256/456 - Spring 2004 1

Recap of the Last Class

- Overall computer system architecture
 - CPU, memory, and I/O devices
- I/O architecture
 - interrupt-driven I/O
- Protection mechanisms provided by the hardware
 - Why? - protect against untrusted user programs
 - Distinguish between 2 exec modes: kernel vs. user
 - Protection for memory, I/O devices, and CPU resources
- When does an OS run in the kernel mode?

1/26/2004 CSC 256/456 - Spring 2004 2

Outline

Operating system organization: this concludes the first three introductory lectures.

- From OS designers' view: System Components
- From users' view: Operating System Services
- Back to the designers' view: Operating System Structure
- Virtual Machines

1/26/2004 CSC 256/456 - Spring 2004 3

Operating System Components

- Process Management
- Memory Management
- I/O System Management
- File and Secondary-Storage Management
- Networking

1/26/2004 CSC 256/456 - Spring 2004 4

Process Management

- A *process* is a program in execution. A process needs certain resources, including CPU time, memory, files, and I/O devices, to accomplish its task.
- OS responsibilities for process management:
 - Process creation and deletion.
 - Process suspension and resumption.
 - Process synchronization
 - Inter-process communication

1/26/2004

CSC 256/456 - Spring 2004

5

Memory Management

- Memory
 - A large array of addressable words or bytes.
 - A data repository shared by the CPU and I/O devices.
- OS responsibility for memory management:
 - Allocate and deallocate memory space as requested.
 - Keep track of which parts of memory are currently being used and by whom.
 - Efficient utilization when the memory resource is heavily competed.

1/26/2004

CSC 256/456 - Spring 2004

6

I/O System Management

- A computer needs I/O to interact with outside world:
 - Console/terminal
 - Non-volatile secondary storage - disks
 - Networking
- The I/O system consists of:
 - A buffer-caching system
 - A general device-driver interface
 - Drivers for specific hardware devices

1/26/2004

CSC 256/456 - Spring 2004

7

File and Secondary Storage Management

- A file is a collection of information defined by its creator. Commonly, files represent programs and data.
- OS responsibility for file management:
 - Manipulation of files and directories.
 - Map files onto (nonvolatile) secondary storage - disks.
- OS responsibility for disk management:
 - Free space management and storage allocation.
 - Disk scheduling.
- They are not all always together
 - Not all files are mapped to secondary storage!
 - Not all disk space are used for the file system!

1/26/2004

CSC 256/456 - Spring 2004

8

Networking (Distributed Systems)

- A *distributed system*
 - A collection of processors that do not share memory.
 - Processors are connected through a communication network.
 - Communication takes place using a *protocol*.
- A distributed system provides user access to various shared system resources, which allows:
 - Computation speed-up
 - Enhanced reliability
- How much of it is in OS?

1/26/2004

CSC 256/456 - Spring 2004

9

Outline

- From OS designers' view: System Components
- From users' view: Operating System Services
- Back to the designers' view: Operating System Structure
- Virtual Machines

1/26/2004

CSC 256/456 - Spring 2004

10

Operating System Services

- Program execution – system capability to load a program into memory and to run it.
- I/O operations – since user programs cannot execute I/O operations directly, the operating system must provide some means to perform I/O.
- File-system manipulation – program capability to read, write, create, and delete files.
- Inter-process Communications – exchange of information between processes.

1/26/2004

CSC 256/456 - Spring 2004

11

Additional Operating System Functions

Additional functions exist not for the OS role of extended machine, but rather for the OS role of resource manager.

- Resource allocation – allocating resources to multiple users or multiple processes running at the same time.
- Accounting – keep track of and record which users use how much and what kinds of computer resources.
- Protection – ensuring that all access to system resources is controlled.

1/26/2004

CSC 256/456 - Spring 2004

12

System Calls

- System calls provide the interface for a user program to seek OS services.
- Example on the right for: **read (fd, buffer, nbytes)**
- How does **TRAP** pass parameters to kernel?

The diagram shows the execution flow of a system call. In user space, a library procedure calls 'read', which then calls 'read' in the user program. The user program pushes parameters (nbytes, buffer, fd) onto the stack. A trap occurs, transferring control to the kernel. In kernel space, a dispatch routine calls a system call handler, which then returns control to the user program. The stack is shown with addresses from 0 to 0FFFFFFF.

1/26/2004 CSC 256/456 - Spring 2004 13

Inter-Process Communications

- Communication may take place using either message passing or shared memory.

The diagram compares two IPC methods. On the left, message passing is shown where process A sends a message (M) to process B via a kernel-mediated path. On the right, shared memory is shown where process A and process B access a common memory region. Arrows indicate the flow of data between processes and the kernel.

1/26/2004 CSC 256/456 - Spring 2004 14

Outline

- From OS designers' view: System Components
- From users' view: Operating System Services
- Back to the designers' view: Operating System Structure
- Virtual Machines

1/26/2004 CSC 256/456 - Spring 2004 15

Monolithic System Structure

The diagram illustrates the monolithic system structure. At the top, an Interactive User interacts with Application Programs. Below this is the OS System Call Interface, which includes a Trap Table and a Monolithic Kernel Module. The kernel module contains Process Management, Memory Management, File Management, and Device Mgmt Infrastructure. Device Drivers are connected to the kernel via a Driver Interface.

Most modern OSES fall into this category!

1/26/2004 CSC 256/456 - Spring 2004 16

Microkernel System Structure

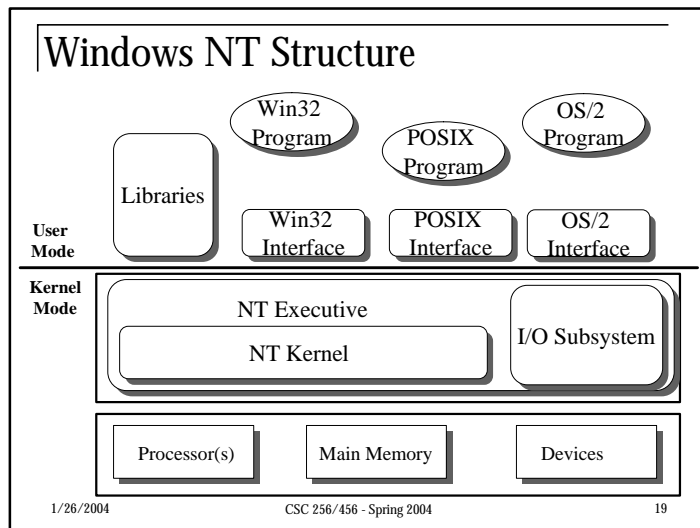
- Microkernel structure:
 - Moves as much from the kernel into "user" space.
 - Communication takes place between user modules using message passing.
- What should be in the kernel and what should be in user space?
 - Mechanisms determine how to do something.
 - Policies decide what will be done.
- Benefits:
 - More reliable (less code is running in kernel mode)
 - More secure (less code is running in kernel mode)
- Disadvantage?

1/26/2004 CSC 256/456 - Spring 2004 17

Layered Structure

- Layered structure
 - The operating system is divided into a number of layers (levels), each built on top of lower layers.
 - The bottom layer (layer 0), is the hardware.
 - The highest (layer N) is the user interface.
 - Decreased privileges for higher layers.
- Benefits:
 - more reliable
 - more secure
 - easier to extend
 - easier to port to new architectures
- Disadvantage? (along with the microkernel structure)
 - Weak integration results in performance penalty.

1/26/2004 CSC 256/456 - Spring 2004 18



Outline

- From OS designers' view: System Components
- From users' view: Operating System Services
- Back to the designers' view: Operating System Structure
- Virtual Machines

1/26/2004 CSC 256/456 - Spring 2004 20

Virtual Machines

- A virtual machine
 - Virtualization: It is a piece of software that provides an interface *identical* to the underlying bare hardware.
 - Multiplexing: It provides several copies of this interface on top of a single piece of hardware.
 - Resource management: The resources of the physical computer are shared to create the virtual machines.
 - e.g., CPU scheduling can create the appearance that each piece of upper-layer software has its own processor.
- It is not an operating system
 - It is a resource manager, but not an extended machine

1/26/2004 CSC 256/456 - Spring 2004 21

VM Models

user programs	user programs	user programs	user programs	user programs on native OS	user programs	user programs
OS	OS	OS	OS	OS	OS	OS
hardware	VM monitor			VM monitor		
hardware	hardware			native OS		
hardware	hardware			hardware		
Non-VM	Native VM			Hosted VM		

1/26/2004 CSC 256/456 - Spring 2004 22

Usage of Virtual Machines

- Running several different OSES on a single machine
- Research and development of operating systems and computer architecture
- Education
- Enhanced reliability and security

1/26/2004 CSC 256/456 - Spring 2004 23

Disclaimer

- Parts of the lecture slides contain original work of Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Andrew S. Tanenbaum, and Gary Nutt. The slides are intended for the sole purpose of instruction of operating systems at the University of Rochester. All copyrighted materials belong to their original owner(s).

1/26/2004 CSC 256/456 - Spring 2004 24