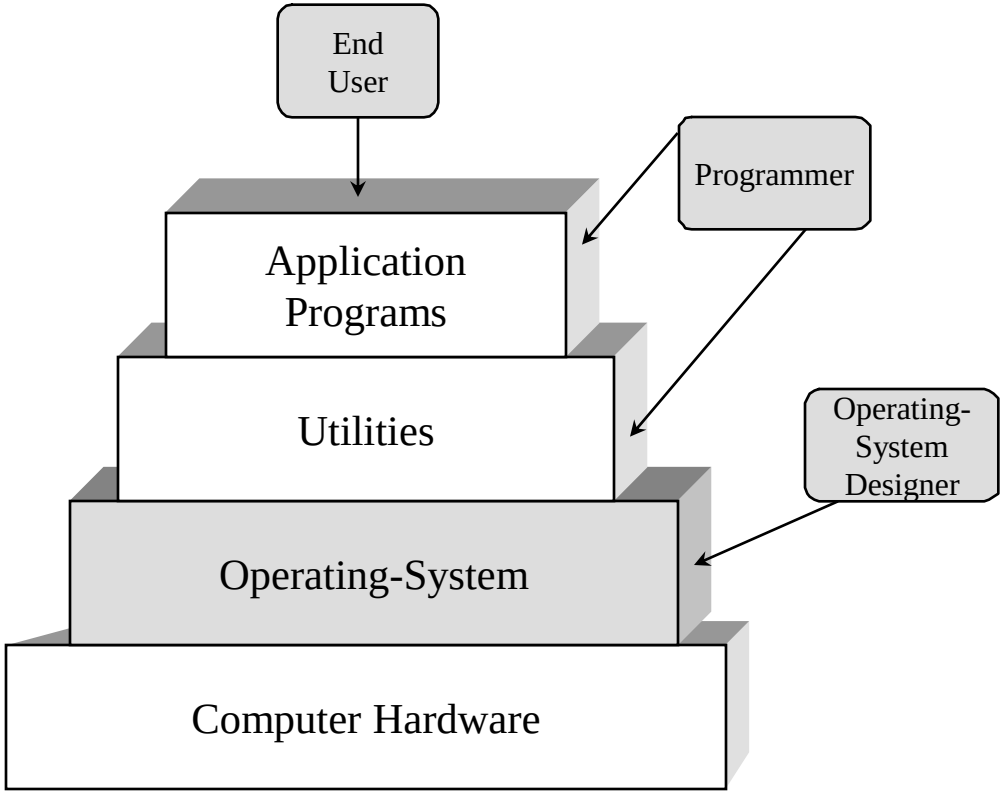
Operating Systems Overview

Chapter 2

Operating System

- A program that controls the execution of application programs
- An interface between the user and hardware
- Masks the details of the hardware

Layers and Views of a Computer System



Services Provided by the Operating System

- ✓ Program execution
- ✓ Access to I/O devices
- ✓ Controlled access to files
- ✓ System access

Services Provided by the Operating System

- ✓ Error detection and response
  - internal and external hardware errors
    - memory error
    - device failure
  - software errors
    - arithmetic overflow
    - access forbidden memory locations
  - operating system cannot grant request issued by an application

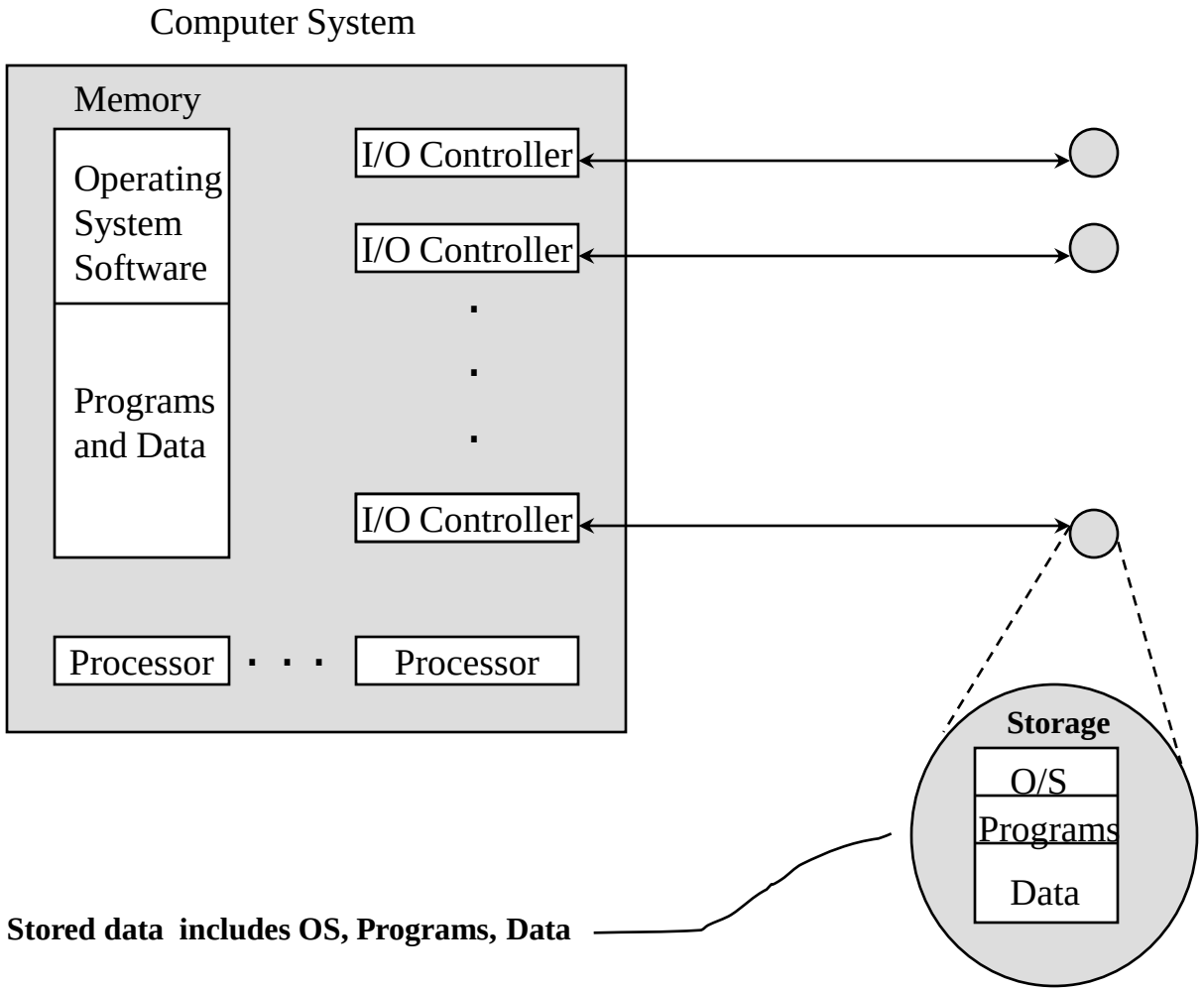
Services Provided by the Operating System

- ✓ Accounting
  - collect statistics
  - monitor performance
  - used to anticipate future enhancements
  - used for billing users

Operating System

- ✓ It is actually a program
- ✓ Directs the processor in the use of system resources
- ✓ Directs the processor when executing other programs
- ✓ Processor stops executing the operating system in order to execute other programs

Operating System as a Resource Manager



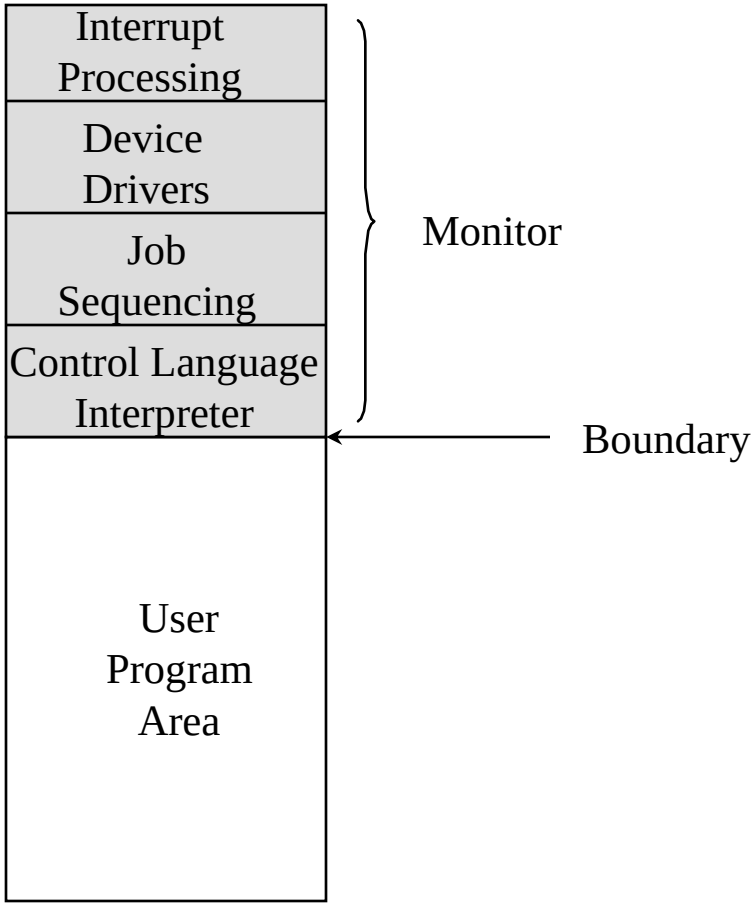
Evolution of an Operating System

- ✓ Hardware upgrades and new types of hardware
- ✓ New services
- ✓ Fixes

Monitors (an ancient type of OS)

- Software that controls the running programs
- Batch operating system
- Jobs are *batched* together
- Resident monitor is in main memory and available for execution
- Monitor utilities are *loaded* when needed

Memory Layout For a Resident Monitor



Job Control Language (JCL)

- ✓ Special type of programming language
- ✓ Provides instruction to the monitor about each job to be admitted into the system
  - what compiler to use
  - what data to use
  - job priority

Hardware Features

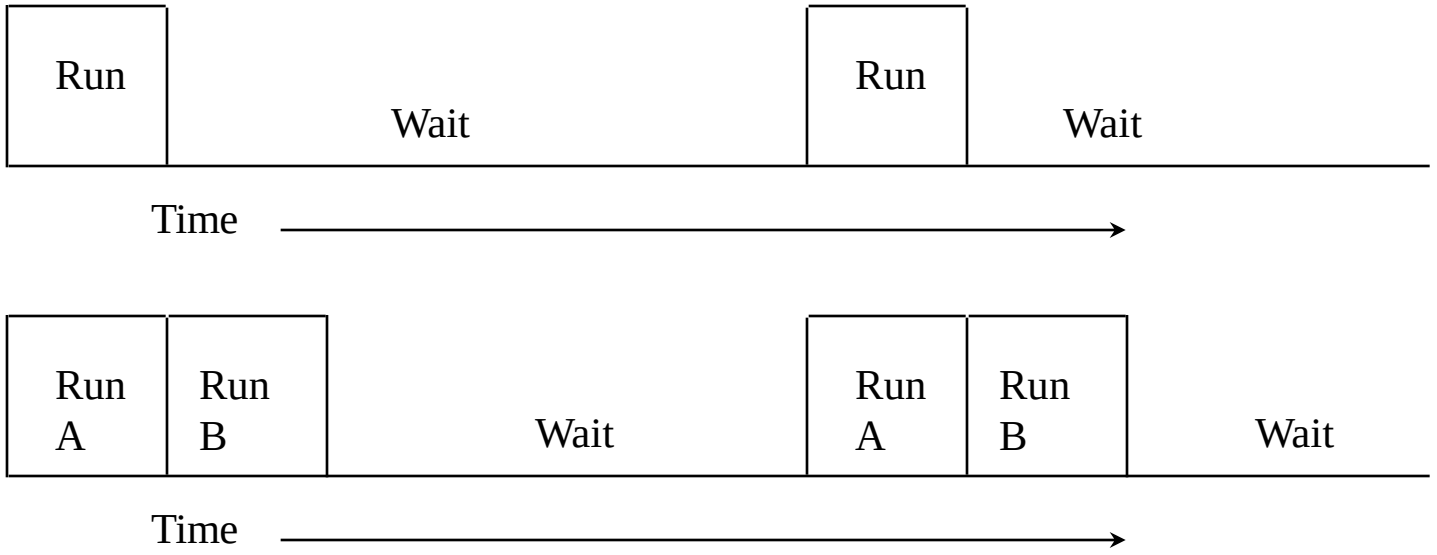
- ✓ Memory protection
  - do not allow the memory area containing the monitor to be altered
- ✓ Timer
  - prevents a job from monopolizing the system
  - an interrupt occurs when time expires

Hardware Features

- ✓ *Privileged instructions*
  - executed only by the monitor
  - an interrupt occurs if a program tries these instructions
- ✓ Interrupts
  - provide flexibility for controlling user programs

Multiprogramming or Multitasking

Allows the processor to execute another program while one program must wait for an I/O device.



Example

	JOB1	JOB2	JOB3
Type of job	Heavy compute	Heavy I/O	Heavy I/O
Duration	5 min.	15 min.	10 min.
Memory required	50K	100 K	80 K
Need disk?	No	No	Yes
Need terminal	No	Yes	No
Need printer?	No	No	Yes

Effects of Multiprogramming

	Uniprogramming	Multiprogramming
Processor use	17%	33%
Memory use	30%	67%
Disk use	33%	67%
Printer use	33%	67%
Elapsed time	30 min.	15 min.
Throughput rate	6 jobs/hr	12 jobs/hr
Mean response time	18 min.	10 min.

Time Sharing

- ✓ Using multiprogramming to handle multiple interactive jobs
- ✓ Processor's time is shared among multiple users
- ✓ Multiple users simultaneously access the system through terminals

Process

- More general term than a job
- Consists of an executable program, associated data, and execution context (registers, program counter, etc.)

Difficulties with Designing OS Software

- ✓ Improper synchronization
  - ensure a process waiting for an I/O device receives the signal
- ✓ Failed mutual exclusion
- ✓ Non-deterministic program operation
  - program should only depend on its input, not on common memory areas
- ✓ Deadlocks

Memory Management

- ✓ Process isolation
- ✓ Automatic allocation and management
- ✓ Support for modular programming
- ✓ Protection and access control
- ✓ Long-term storage

Virtual Memory

- Allows programmers to address memory from a logical point of view
- While program is running portions of the program and data are kept in blocks on disk (transparently to the application).

File System

- ✓ Implements long-term storage
- ✓ Information stored in named objects, called *files*

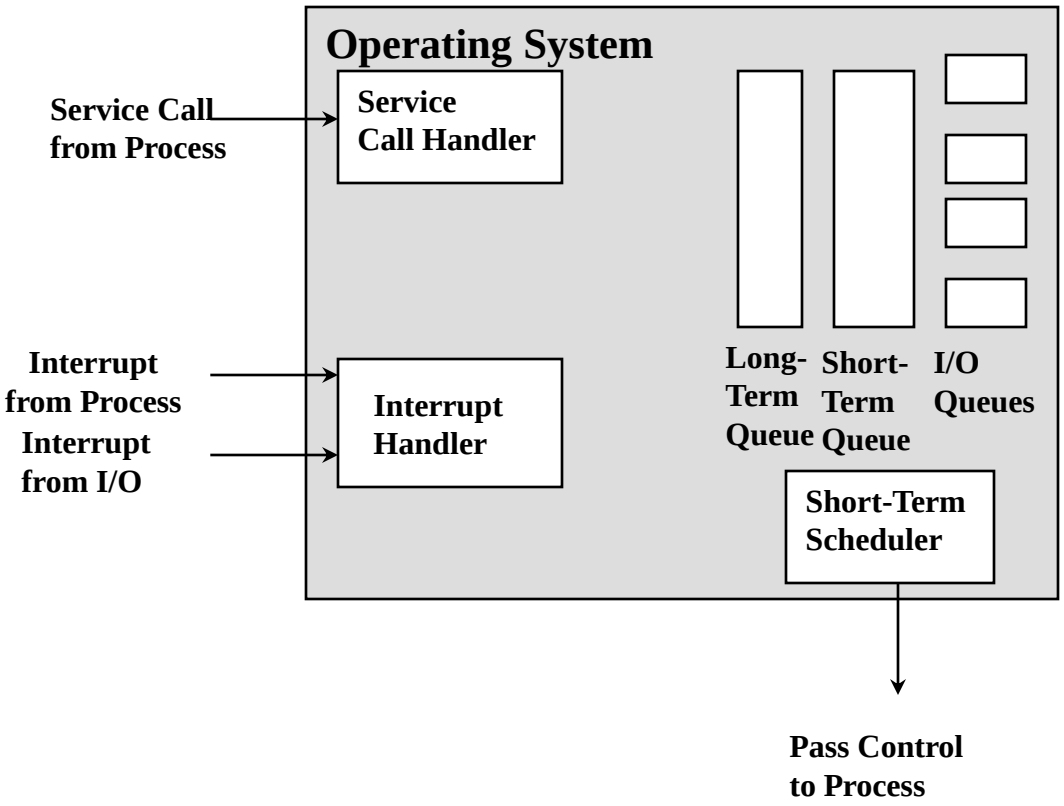
Categories of Security and Protection

- ✓ Access control
  - regulate user access to the system
- ✓ Information flow control
  - regulate flow of data within the system and its delivery to users
- ✓ Certification
  - proving that access and flow control perform according to specifications

Scheduling and Resource Management

- ✓ Fairness
  - give equal and fair access to all processes
- ✓ Differential responsiveness
  - discriminate between different classes of jobs
- ✓ Efficiency
  - maximize throughput, minimize response time, and accommodate as many users as possible

Major Elements Of Operating System



System Structure

- View the system as a series of levels
- Each level performs a related subset of functions
- Each level relies on the next lower level to perform more primitive functions
- This decomposes a problem into a number of more manageable sub-problems

Operating System Design Hierarchy

Level	Name	Objects	Example Operations
13	Shell	User programming environment	Statements in shell language
12	User processes	User processes	Quit, kill, suspend, resume
11	Directories	Directories	Create, destroy, attach, detach, search, list
10	Devices	External devices, such as printer, displays and keyboards	Create, destroy, open, close, read, write
9	File system	Files	Create, destroy, open, close read, write
8	Communications	Pipes	Create, destroy, open. close, read, write

Operating System Design Hierarchy

Level	Name	Objects	Example Operations
7	Virtual Memory	Segments, pages	Read, write, fetch
6	Local secondary store	Blocks of data, device channels	Read, write, allocate, free
5	Primitive processes	Primitive process, semaphores, ready list	Suspend, resume, wait, signal

Operating System Design Hierarchy

Level	Name	Objects	Example Operations
4	Interrupts	Interrupt-handling programs	Invoke, mask, unmask, retry
3	Procedures	Procedures, call stack	Mark stack, call, return
2	Instruction Set	Evaluation stack, micro-program interpreter, scalar and array data	Load, store, add, subtract, branch
1	Electronic circuits	Registers, gates, busses, etc.	Clear, transfer, activate, complement

Characteristics of Modern Operating Systems

- ✓ Microkernel architecture
  - assigns only a few essential functions to the kernel
    - address space
    - interprocess communication (IPC)
    - basic scheduling

Characteristics of Modern Operating Systems

- ✓ *Multithreading*
  - process is divided into threads that can run simultaneously
- ✓ *Thread*
  - dispatchable unit of work
  - executes sequentially and is interruptible
- ✓ *Process* (sometimes also called *task*) is a collection of one or more threads

Characteristics of Modern Operating Systems

- ✓ *Symmetric multiprocessing (SMP)*
  - there are multiple processors
  - these processors share same main memory and I/O facilities
  - Each processor can perform the same functions (i.e., processors are interchangeable)

Characteristics of Modern Operating Systems

- ✓ Distributed operating systems
  - distinct computers organized into a single whole
  - provides the illusion of a single main memory
  - provides distributed file system that appears to the user as a single file space

Characteristics of Modern Operating Systems

- ✓ Object-oriented design
  - used for adding modular extensions to a small kernel
  - enables programmers to customize an operating system without disrupting system integrity

Windows NT/.../XP/.../10/...

- ✓ Exploits the power of today's microprocessors
- ✓ Provides full multitasking in a single-user environment
- ✓ Can execute on several hardware platforms

Windows NT/.../10/...

- ✓ Modified microkernel architecture
  - many of the system functions outside of the microkernel still run in kernel mode
- ✓ Hardware abstraction layer (HAL)
  - makes the hardware look the same to the kernel, regardless of the platform
  - provides support for symmetric multiprocessing

Windows NT/.../10/...

- The main parts of the OS are
  - The Executive
    - Memory management, processes&threads (create, delete,track), I/O
  - (micro) Kernel
    - Thread scheduling, process switching, interrupts, synchronization
  - Hardware abstraction layer (HAL)
  - Device drivers
  - Window manager and graphics
- Executive is built using a client-server architecture
  - Eg, memory management, process creation/deletion, scheduling are services that wait for requests

Advantages of Client/Server

- ✓ Simplifies the base operating system, the executive
  - possible to construct a variety of APIs (application programming interface)
- ✓ Improves reliability
  - each server runs as a separate process with its own partition of memory
  - servers cannot directly access hardware
- ✓ Provides basis for distributed computing

Windows Features for SMP

- ✓ Operating system routines can run on any processor
- ✓ Different routines can execute simultaneously on different processors
- ✓ Multiple threads of execution within a single process
- ✓ Servers use multiple threads to process requests for more than one client simultaneously
- ✓ Flexible interprocess communication

Windows Objects

- Not all entities are objects
- Entities represented by objects are files, processes, threads, semaphores, times, and windows
- *Object manager* in the kernel is responsible for creating and destroying objects, for granting access to object's services and data

UNIX Architecture
Linux, Solaris, Mac OS X

- ✓ Hardware is surrounded by the operating-system
- ✓ Operating system is called the *kernel*
- ✓ Comes with a number of user services and interfaces
  - shell
  - compilers
  - inter-process communication
- ✓ Symmetric multi-processing
- ✓ Some use microkernels (Solaris, Mac OS X); some are modular-monolithic (Linux)

UNIX Architecture