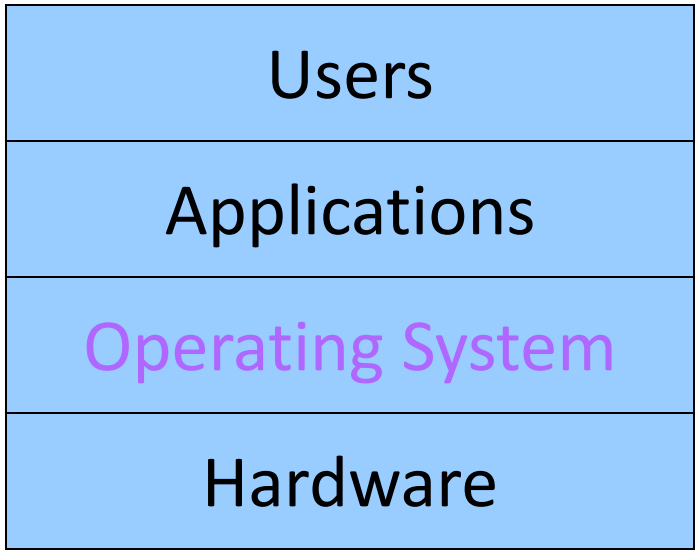
Introduction and Overview

Questions answered in this lecture:

- What is an operating system?
- What is the role of an OS?
- What is the functionality of an OS?
- How have operating systems evolved?
- Why study operating systems?

What is an Operating System?

Not easy to define precisely…



compilers
databases
word processors

CPU
memory
I/O devices

OS:

Everything in system that isn't an application or hardware

OS:

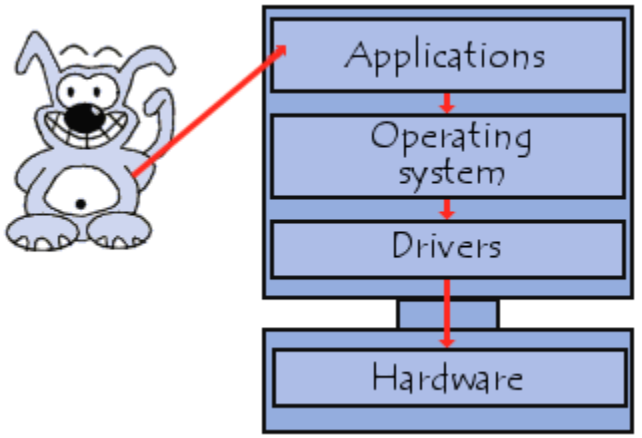
Software that converts hardware into a useful form for applications

What is an Operating System?

- A program that acts as an intermediary between a user of a computer and the computer hardware.
- Operating system goals:
  - Execute user programs and make solving user problems easier.
  - Make the computer system convenient to use.
- Use the computer hardware in an efficient manner.

OS – Good Definition!

- The **operating system** (sometimes referred to by its abbreviation *OS*), is responsible for creating the link between the material resources, the user and the applications (word processor, video game, etc.). When a programme wants to access a material resource, it does not need to send specific information to the peripheral device but it simply sends the information to the operating system, which conveys it to the relevant peripheral via its driver. If there are no drivers, each programme has to recognize and take into account the communication with each type of peripheral!



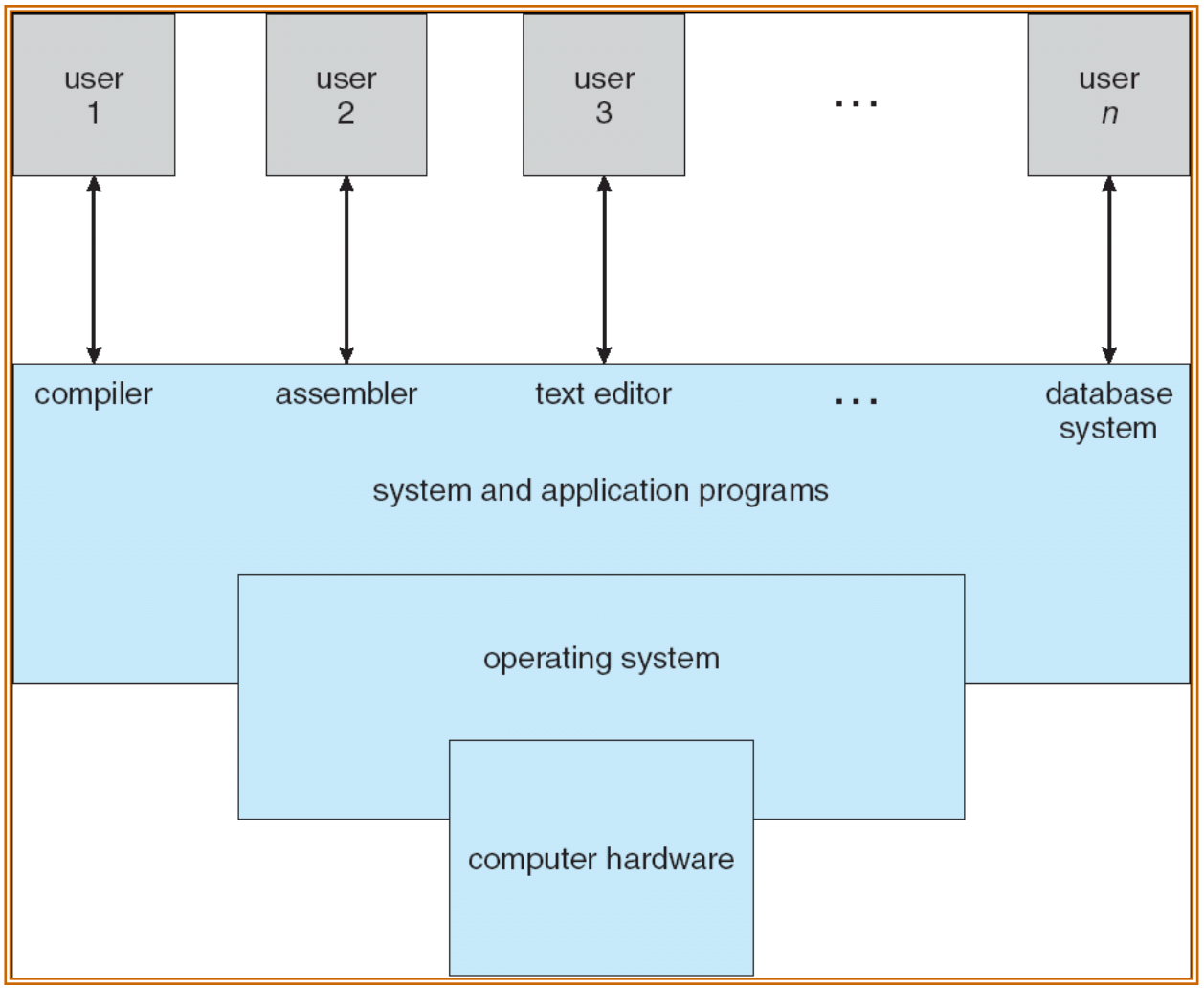
Operating System Definition

- OS is a **resource allocator**
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use
- OS is a **control program**
  - Controls execution of programs to prevent errors and improper use of the computer

Computer System Structure

- Computer system can be divided into four components
  - Hardware – provides basic computing resources
    - CPU, memory, I/O devices
  - Operating system
    - Controls and coordinates use of hardware among various applications and users
  - Application programs – define the ways in which the system resources are used to solve the computing problems of the users
    - Word processors, compilers, web browsers, database systems, video games
  - Users
    - People, machines, other computers

Four Components of a Computer System



What is the role of the OS?

Role #1: Provide standard Library (I.e., abstract resources)

What is a resource?

- Anything valuable (e.g., CPU, memory, disk)

Advantages of standard library

- Allow applications to reuse common facilities
- Make different devices look the same
- Provide higher-level abstractions

Challenges

- What are the correct abstractions?
- How much of hardware should be exposed?

What is the role of the OS?

Role #2: Resource coordinator (I.e., manager)

Advantages of resource coordinator

- Virtualize resources so multiple users or applications can share
- Protect applications from one another
- Provide efficient and fair access to resources

Challenges

- What are the correct mechanisms?
- What are the correct policies?

Operating System Services/ Functionality

- One set of operating-system services provides functions that are helpful to the user:
  - User interface - Almost all operating systems have a user interface (UI)
    - Varies between Command-Line (CLI), Graphics User Interface (GUI), Batch
  - Program execution - The system must be able to load a program into memory and to run that program, end execution, either normally or abnormally (indicating error)
  - I/O operations - A running program may require I/O, which may involve a file or an I/O device.
  - File-system manipulation - The file system is of particular interest. Obviously, programs need to read and write files and directories, create and delete them, search them, list file Information, permission management.

Operating System Services (Cont.)

- One set of operating-system services provides functions that are helpful to the user (Cont):
  - Communications – Processes may exchange information, on the same computer or between computers over a network
    - Communications may be via shared memory or through message passing (packets moved by the OS)
  - Error detection – OS needs to be constantly aware of possible errors
    - May occur in the CPU and memory hardware, in I/O devices, in user program
    - For each type of error, OS should take the appropriate action to ensure correct and consistent computing
    - Debugging facilities can greatly enhance the user's and programmer's abilities to efficiently use the system

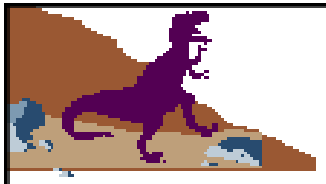
Operating System Services (Cont.)

- Another set of OS functions exists for ensuring the efficient operation of the system itself via resource sharing
  - **Resource allocation -** When multiple users or multiple jobs running concurrently, resources must be allocated to each of them
    - Many types of resources - Some (such as CPU cycles,mainmemory, and file storage) may have special allocation code, others (such as I/O devices) may have general request and release code.
  - **Accounting -** To keep track of which users use how much and what kinds of computer resources
  - **Protection and security -** The owners of information stored in a multiuser or networked computer system may want to control use of that information, concurrent processes should not interfere with each other
    - **Protection** involves ensuring that all access to system resources is controlled
    - **Security** of the system from outsiders requires user authentication, extends to defending external I/O devices from invalid access attempts
    - If a system is to be protected and secure, precautions must be instituted throughout it. A chain is only as strong as its weakest link.

History of the OS

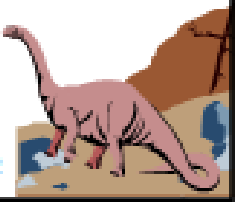
Two distinct phases of history

- Phase 1: Computers are expensive
  - Goal: Use computer's time efficiently
  - Maximize throughput (I.e., jobs per second)
  - Maximize utilization (I.e., percentage busy)
- Phase 2: Computers are inexpensive
  - Goal: Use people's time efficiently
  - Minimize response time

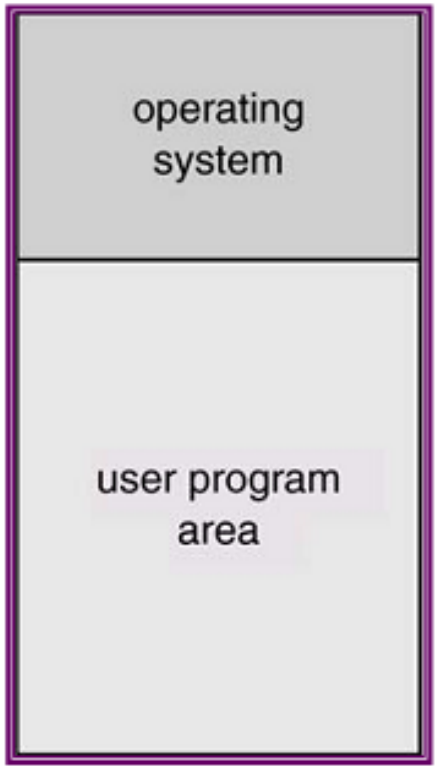


Simple Batch Systems

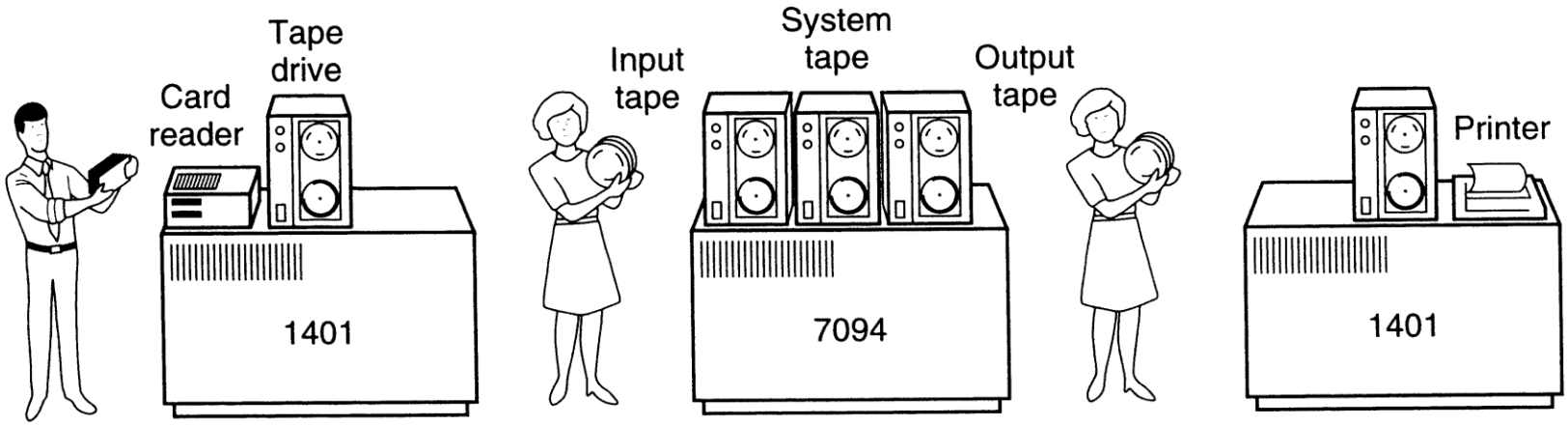
- User prepare a job and submit it to a computer operator
- User get output some time later
- No interaction between the user and the computer system
- Operator batches together jobs with similar needs to speedup processing
- Task of OS: automatically transfers control from one job to another.
- OS always resident in memory
- Disadvantages of one job at a time:
  - CPU idle during I/O
  - I/O devices idle when CPU busy



Memory Layout for a Simple Batch System



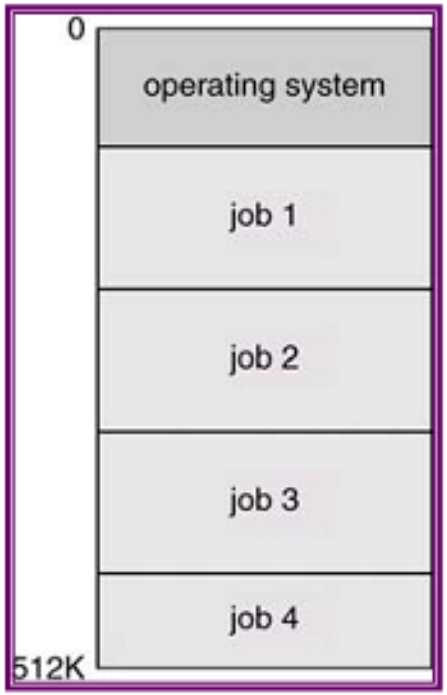
Batch Processing



Multiprogrammed Batch Systems

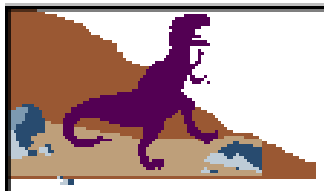
- Keep more than one job in memory simultaneously
- When a job performs I/O, OS switches to another job
  - Increase CPU utilization
- All jobs enter the system kept in the job pool on a disk, scheduler brings jobs from pool into memory

Multiprogrammed Batch Systems



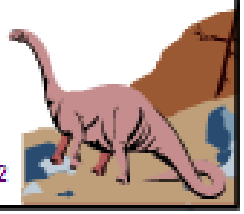
OS Features Needed for Multiprogramming

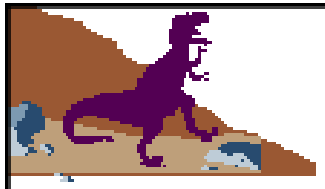
- Job scheduling: which jobs in the job pool should be brought into memory? (Ch6)
- Memory management: the system must allocate the memory to several jobs. (Ch 9, Ch 10)
- CPU scheduling: choose among jobs in memory that are ready to run. (Ch6)
- Allocation of devices: what if more than one job wants to use a device?
- Multiple jobs running concurrently should not affect one another



Time-Sharing Systems

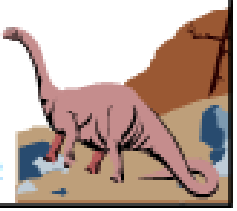
- Like multiprogrammed batch, except CPU switches between jobs occur so frequently that the users can interact with a running program
- User give instructions to the OS or a program, and wait for immediate results
  - Require low response time
- Allow many users to share the computer simultaneously, users have the impression that they have their own machine
- CPU is multiplexed among several jobs that are kept in memory and on disk





OS Features Needed for Time-Sharing

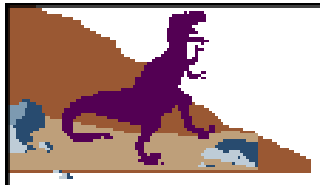
- Job synchronization and communication (Ch7)
- Deadlock handling (Ch8)
- File system (Ch11, Ch12)



Desktop Systems

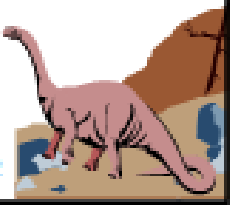
- Personal computers – computer system dedicated to a single user.
- CPU utilization not a prime concern, want maximize user convenience and responsiveness.
- Can adopt technologies developed for mainframe operating systems: virtual memory, file systems, multiprogramming
- File protection needed due to interconnections of computers
- Operating systems for PCs: Windows, Mac OS, Linux

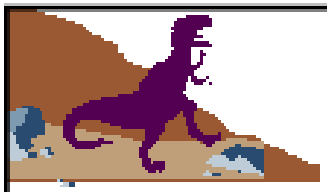
- **Multiprogramming** needed for efficiency
  - Single user cannot keep CPU and I/O devices busy at all times
  - Multiprogramming organizes jobs (code and data) so CPU always has one to execute
  - A subset of total jobs in system is kept in memory
  - One job selected and run via **job scheduling**
  - When it has to wait (for I/O for example), OS switches to another job
- **Timesharing (multitasking)** is logical extension in which CPU switches jobs so frequently that users can interact with each job while it is running, creating **interactive** computing
  - **Response time** should be < 1 second
  - Each user has at least one program executing in memory ⇨**process**
  - If several jobs ready to run at the same time ⇨ **CPU scheduling**
  - If processes don't fit in memory, **swapping** moves them in and out to run
  - **Virtual memory** allows execution of processes not completely in memory



Multiprocessor Systems

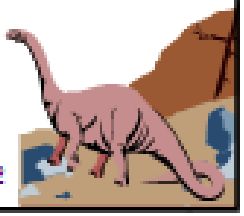
- Also known as parallel systems or tightly coupled systems
- More than one processor in close communication, sharing computer bus, clock, memory, and usually peripheral devices
  - Communication usually takes place through the shared memory.
- Advantages
  - Increased throughput: speed-up ratio with N processors < N
  - Economy of scale: cheaper than multiple single-processor systems
  - Increased reliability: graceful degradation, fault tolerant





Multiprocessor Systems

- Symmetric multiprocessing (SMP)
  - Each processor runs an identical copy of the operating system.
  - All processors are peers: any processor can work on any task
  - OS can distribute load evenly over the processors.
  - Most modern operating systems support SMP
- Asymmetric multiprocessing
  - Master-slave relationship: a master processor controls the system, assigns works to other processors
  - Each processor is assigned a specific task
    - Don't have the flexibility to assign processes to the least-loaded CPU
  - More common in extremely large systems



Clustered Systems

Clustered systems are typically constructed by combining multiple computers into a single system to perform a computational task distributed across the cluster.

Multiprocessor systems on the other hand could be a single physical entity comprising of multiple CPUs.

A clustered system is less tightly coupled than a multiprocessor system.

Clustered systems communicate using messages, while processors in a multiprocessor system could communicate using shared memory.

In order for two machines to provide a highly available service, the state on the two machines should be replicated and should be consistently updated.

When one of the machines fail, the other could then take-over the functionality of the failed machine.

Evolution of OS :

Major Phases	Technical Innovations	Operating Systems
Open Shop	**The idea of OS**	**IBM 701 open shop (1954)**
Batch Processing	**Tape batching,** **First-in, first-out scheduling.**	**BKS system (1961)**
Multi-programming	**Processor multiplexing,** **Indivisible operations,** **Demand paging,** **Input/output spooling,** **Priority scheduling,** **Remote job entry**	**Atlas supervisor (1961),** **Exec II system (1966)**

(Contd...)

Evolution of OS (contd..):

Timesharing	**Simultaneous user interaction,** **On-line file systems**	**Multics file system (1965),** **Unix (1974)**
Concurrent Programming	**Hierarchical systems, Extensible kernels, Parallel programming concepts, Secure parallel languages**	**RC 4000 system (1969),** **13 Venus system (1972),** **14 Boss 2 system (1975).**
Personal Computing	**Graphic user interfaces**	**OS 6 (1972)** **Pilot system (1980)**
Distributed Systems	**Remote servers**	**WFS file server (1979) Unix United RPC (1982)** **24 Amoeba system (1990)**

KINDS OF OS & PROPERTIES

Properties of the following types of operating systems:

- a. Batch
- b. Interactive
- c. Time sharing
- d. Real time
- e. Network
- f. Parallel
- g. Distributed
- h. Clustered
- i. Handheld

KINDS OF OS & PROPERTIES

- a. **Batch**. Jobs with similar needs are batched together and run through the computer as a group by an operator or automatic job sequencer. Performance is increased by attempting to keep CPU and I/O devices busy at all times through buffering, off-line operation, spooling, and multiprogramming. Batch is good for executing large jobs that need little interaction; it can be submitted and picked up later.

- b. **Interactive**. This system is composed of many short transactions where the results of the next transaction may be unpredictable. Response time needs to be short (seconds) since the user submits and waits for the result.

- c. **Time sharing**. This systems uses CPU scheduling and multiprogramming to provide economical interactive use of a system. The CPU switches rapidly from one user to another. Instead of having a job defined by spooled card images, each program reads its next control card from the terminal, and output is normally printed immediately to the screen.

KINDS OF OS & PROPERTIES

- d. **Real time**. Often used in a dedicated application, this system reads information from sensors and must respond within a fixed amount of time to ensure correct performance.

- e. **Network**. Provides operating system features across a network such as file sharing.

- f. **SMP**. Used in systems where there are multiple CPU's each running the same copy of the operating system. Communication takes place across the system bus.

- g. **Distributed**. This system distributes computation among several physical processors. The processors do not share memory or a clock. Instead, each processor has its own local memory. They communicate with each other through various communication lines, such as a high-speed bus or local area network.
- h. **Clustered**. A clustered system combines multiple computers into a single system to perform computational task distributed across the cluster.

- i. **Handheld**. A small computer system that performs simple tasks such as calendars, email, and web browsing. Handheld systems differ from traditional desktop systems with smaller memory and display screens and slower processors.

Why study Operating Systems?

Build, modify, or administer an operating system

Understand system performance

- Behavior of OS impacts entire machine
- Challenge to understand large, complex system
- Tune workload performance
- Apply knowledge across many areas
  - Computer architecture, programming languages, data structures and algorithms, and performance modeling

Reading Assignment:

1. Under what circumstances would a user be better off using a timesharing system rather than a PC or single-user workstation?

2. Which of the functionalities listed below need to be supported by the operating system for the following two settings: (a) handheld devices and (b) real-time systems.
a. Batch programming
b. Virtual memory
c. Time sharing

3. How are network computers different from traditional personal computers? Describe some usage scenarios in which it is advantageous to use network computers.