What is an Operating System?

- **Definition:** Set of software that controls the overall operation of a computer system.
  - Interface between application software and hardware so
  - Users and applications do not have to have direct communication with low-level hardware devices

- **Examples:**
  - Unix (Solaris and Linux)
  - Microsoft Windows 7 (XP, Vista)
  - MacOS (Snow Leopard, Mountain Lion)
  - Mobile device Oss (iOS, Android, Windows Mobile)
Functions of an Operating System

- User interface
- Software support for hardware devices
- Resource management

User Interface

- Graphical or text
- Algorithms:
  - Graphics
Support for Hardware Devices

- Device drivers
- Control signals to/from device to perform functions on behalf of applications
- *Algorithms:*
  - Low-level hardware control

Resource Management

- OS manages resources used by applications
- Resources include:
  - CPU
  - Memory
  - Hardware devices
  - Data
- How to allocate resources to applications that all need to use them to run
- *Algorithms:*
  - Similar to problems/solutions we deal with in our every day lives
Resource Management Problems

- CPU scheduling
- Memory management
- Concurrency control
- Deadlock management
- Present problems here
  - Discuss solutions (algorithms) in class

CPU Scheduling

- CPU (central processing unit)
  - Processor executes all program instructions
  - Can only execute one instruction at a time
  - Need a way to determine which instructions to execute when
- Scheduler
  - Part of the operating system that manages processes and schedules their time on the CPU
  - Every program that wants to run on the CPU is called a process.

- Process is in one of three states:
  - running - actively executed by CPU
  - ready - waiting for CPU
  - waiting - for some event such as keyboard input
Memory Management

- OS determines how much memory needed for program to run
- Allocates memory
- Must swap out other programs if not enough room
- Allocates pages
- Which pages to swap out?

Concurrency Control

- Several processes run concurrently
  - often need to access shared resources
  - disk, memory, printer, network interface, etc.
- OS must ensure that the resources shared in a non-interfering fashion
Sharing Resources

- Sometimes resources can’t be used at the same time
- Need to take turns
- Use locks to decide who gets it when

Bank Example

P1: Get balance
balance = balance – 100
Write balance to account

P2: Get balance
balance = balance – 100
Write balance to account

balance: $900
Solution

P1: Lock balance
Get balance
balance = balance – 100
Write balance to account
Unlock balance

P2: Lock balance
Get balance
balance = balance – 100
Write balance to account
Unlock balance

Another Problem

P1: Lock checking balance
Lock savings balance
Subtract $100 from checking
Add $100 to savings
Unlock accounts

P2: Lock savings balance
Lock checking balance
Subtract $100 from checking
Add $100 to savings
Unlock accounts

DEADLOCK!
Deadlock

- **Deadlock** occurs when two or more processes is each waiting for a resource that is locked by another process.

- Operating system must detect and break deadlocks.