# CSE321: Operating Systems Introduction

# **Course Logistics**

### Instructor:

Dr. Md Sadek Ferdous, Associate Professor, CSE, BRACU

### Course schedule:

Monday/Wednesday: 12:30pm to 1:50pm.

Room number: 09H-35C

### Textbook:

• Operating System Concepts, by Silberschatz, Galvin and Gagne, 9<sup>th</sup> Edition.

### **Introduction**

- Introduce yourself and state one interesting fact about yourself!
- Why are you taking this class now?
- What is your expectation out of this class?

### **Course Outcome**

- To understand the fundamental concepts of computer system organization and the structure of operating systems.
- To explore various aspects of process management in operating system
- To know how different CPU scheduling algorithm works and their respective importance
- To develop practical knowledge on the concept of threads
- To inspect process synchronization mechanisms and deadlocks
- To be able to analyze the management of main and virtual memory

## **Marks Distribution**

- Theory 80%
  - Class participation 5%
  - o Take Home Exam − 5%
  - Quiz 10% (n-1)
  - 。 Mid − 25%
  - ∘ Final 35%
- Lab 20%

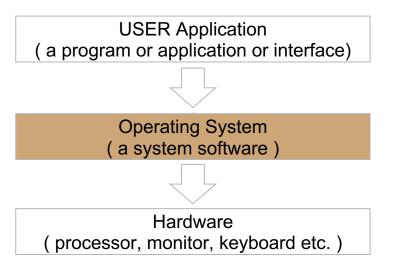
### **Operating Systems**

### "Actual" Introduction



# What is an Operating System?

A program that acts as an intermediary between a user of a computer and the computer hardware.



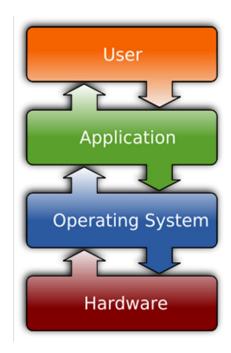
# System Software Vs Application Software

### **System Software:**

- System Software refers to the operating system and all utility programs that manage computer resources at a low level.
- Systems software includes compilers, loaders, linkers, and debuggers.

### **Application Software:**

 Applications software comprises programs designed for an end user, such as word processors, database systems, and spreadsheet programs.



# **Major Goals of OS**

- Execute user programs.
- Make the computer system convenient to use.
- Use the computer hardware in an efficient manner
- Manages and allocate all resources
- Controls the execution of user programs and operations of I/O devices

GM-NAA I/O, produced by General Motors for its **IBM 704** 

1956



Apple ][ released



MS-DOS is released by Microsoft

1977 1981



Linux is released by Linus **Torvalds** 

1991



Microsoft Windows 95 Windows 95 is released

1995



Android is released (based on a Linux kernel)

OpenShift released by Red Hat

2008

2011

2010s

### **Timeline of Operating Systems**



1960s

IBM develops a series of OSs for its 360 series. Multics is developed and abandoned but UNIX is developed as a consequence.



Unix becomes popular in academic circles and spawns many versions











### 1980s

The home computer revolution



### 1990s

Windows dominates the laptop and desktop market



Unix and then Linux dominate the Supercomputer Market



### 2000s

Smart phones become ubiquitous after the iPhone release in 2007















- Another revolution of OS came in the mobile computing domain, when Steve Jobs introduced iPhone with iOS in 2007
- The iPhone introduction video is now regarded as a classic advertise video



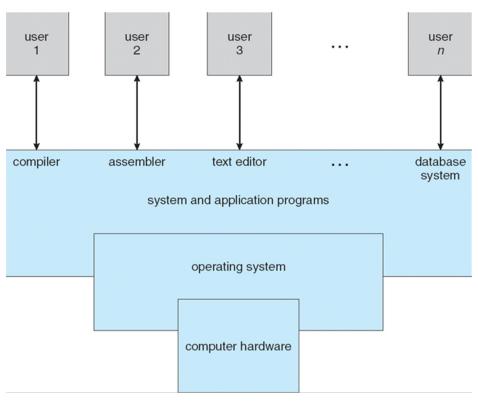


# AN UPDATE IS AVAILABLE FOR YOUR COMPUTER



# Operating Systems Computer System Organization

# **Components of a Computer System**



### Kernel

The one program running at all times.

- Kernel is the central module of an operating system
- Part of OS that loads first, and it remains in main memory.
- As small as possible
- Provide all the essential services required by other parts of the operating system and applications.
- Kernel code is usually loaded into a protected area of memory to prevent it from being overwritten.

## **Bootstrap Program**

An initial program executed when a computer starts running.

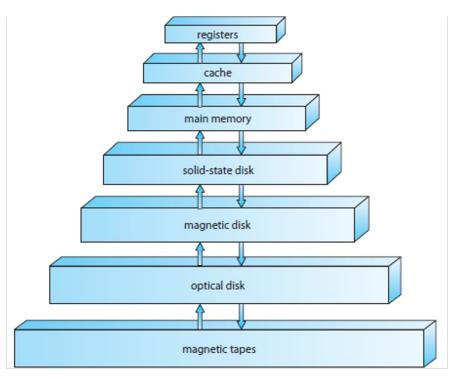
- When a computers is powered up or rebooted, it is executed first.
- Stored in the ROM or EEPROM, known as firmware
- Initializes all aspects of the system, from CPU registers to device controllers to memory contents
- bootstrap program must know how to load the operating system and how to start executing
- Once the OS kernel is loaded and executing, it can start providing services to the system and its users

# **Storage Structure**

- Main memory only large storage media that the CPU can access directly
  - □ Random access
  - ☐ Typically volatile
- ☐ Secondary storage extension of main memory that provides large nonvolatile storage capacity

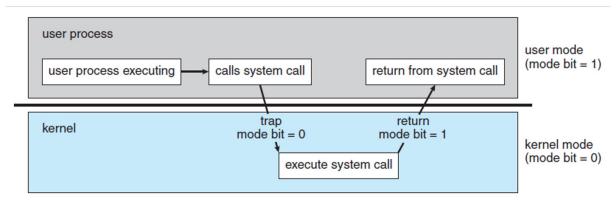
- CPU can load instructions only from main memory.
- General-purpose computers run most of their programs from rewritable memory, called main memory ( also called RAM)
- Computers use other forms of memory as well Read Only Memory (ROM) and electrically erasable programmable read-only memory (EEPROM)
- Only static programs, such as the bootstrap program described earlier, are stored here.

# **Storage Device Hierarchy**



# **Dual Mode Operation**

- Need to distinguish between the execution of operating-system code and user defined code.
- A bit, called the mode bit, is added to the hardware of the computer to indicate the current mode: kernel (0) or user (1).
- dual mode of operation provides protection of the operating system from errant users
- this protection is provided by designating some of the machine instructions that may cause harm as privileged instructions that are executed only in kernel mode.

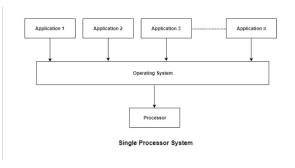


Transition from user to kernel mode.

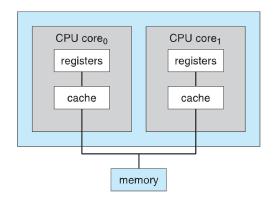
# Operating Systems OS Architecture

### **Operating System Architecture**

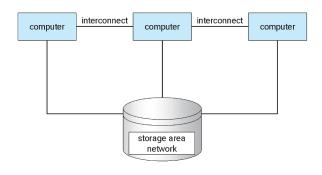
### **Single-Processor Systems**



### **Multiprocessor Systems**



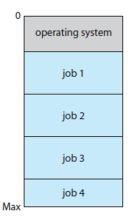
### **Clustered Systems**



### **Operating System Architecture**

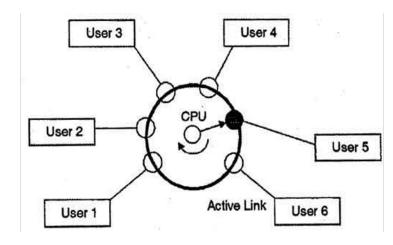
### Multiprogramming

- Job Scheduling
- CPU Scheduling



Memory layout for a multiprogramming system.

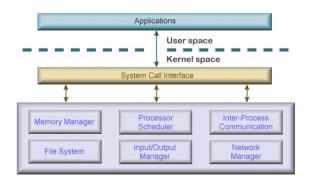
### **Time Sharing System**



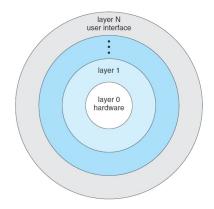
# Operating Systems OS Structure

## **Operating System Structure**

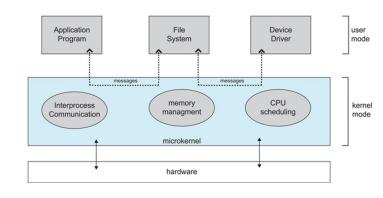
### Simple/Monolithic structure



### Layered structure



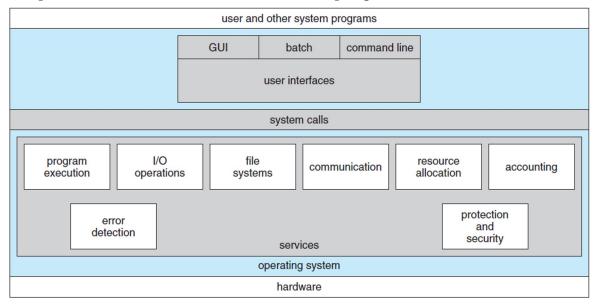
### Microkernel structure



# Operating Systems OS Services

# Operating System Services

- OS provides an environment for the execution of programs.
- Specific services provided, differ from one operating system to another, but there are some common classes
- Services are provided for the convenience of the programmer

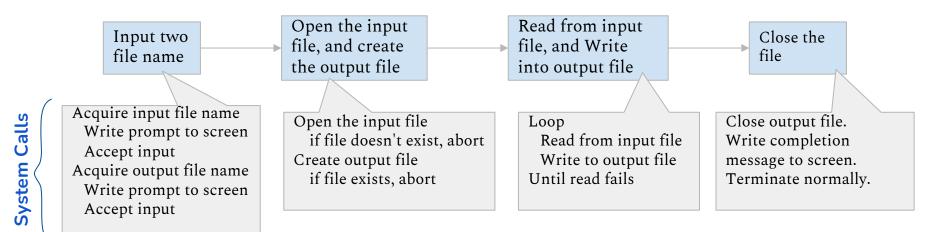


Operating Systems
System Call,
System Program,
System Boot

# System Call

- System calls provide an interface to the services made available by an operating system.
- These calls are generally available as routines.
- Routines are written in C or C++. Some low level tasks are written in assembly language.

A program to copy the contents of a file to another file!



# System Call Interface

- Serves as the link to system calls made available by the operating system.
- A number is associated with each system call, and the system-call interface maintains a table indexed according to these numbers.
- Invokes the intended system call in the operating-system kernel and returns the status of the system call and any return values.

