Concepts of Operating Systems

Objective

To introduce Operating System concepts with Linux environment, and to learn Shell Programming.

Course Content

https://www.cdac.in/index.aspx?id=DAC&courseid=0#

- Introduction to OS
- Introduction to Linux
- Shell Programming
- Process management
- Signals
- Threads
- Memory management
- Virtual Memory
- Deadlock
- Semaphore
- Inter process communication

Text Books

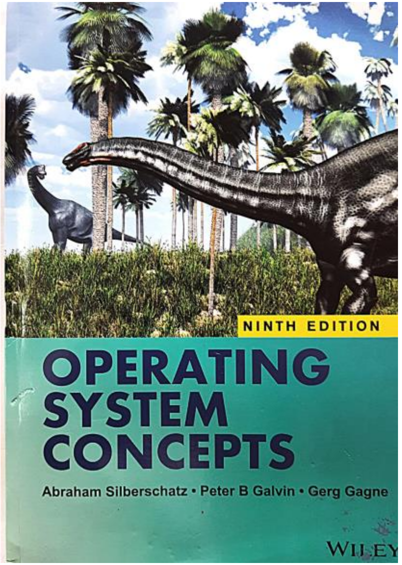
1. **Operating Systems Principles by Abraham Silberschatz, Peter Galvin & Greg Gagne / Wiley**

[Operating System Concepts - 9th edition]

1. **Unix Concepts and Applications by Sumitabha Das / Mc Graw Hill**

References:

- Modern operating Systems by Andrew Tanenbaum & Herbert Bos/ Pearson
- Principles of Operating Systems by Naresh Chauhan / Oxford University Press
- Beginning Linux Programming by Neil Matthew & Richard Stones / Wrox
- Operating System : A Design-Oriented Approach by Charles Crowley / McGraw Hill


NINTH EDITION
OPERATING SYSTEM CONCEPTS
Abraham Silberschatz • Peter B Galvin • Gerg Gagne
WILEY

Introduction to OS

(Session1)

Introduction to OS

- What is OS ?
- How is it different from other application software ?
- Why is it hardware dependent ?
- Different components of OS ?
- Basic computer organization required for OS
- Examples of well known OS including mobile OS, embedded system OS, Real Time OS, desktop OS server machine OS etc.
- How are these different from each other and why ?
- Functions of OS
- User and Kernel space and mode
- Interrupts and system calls

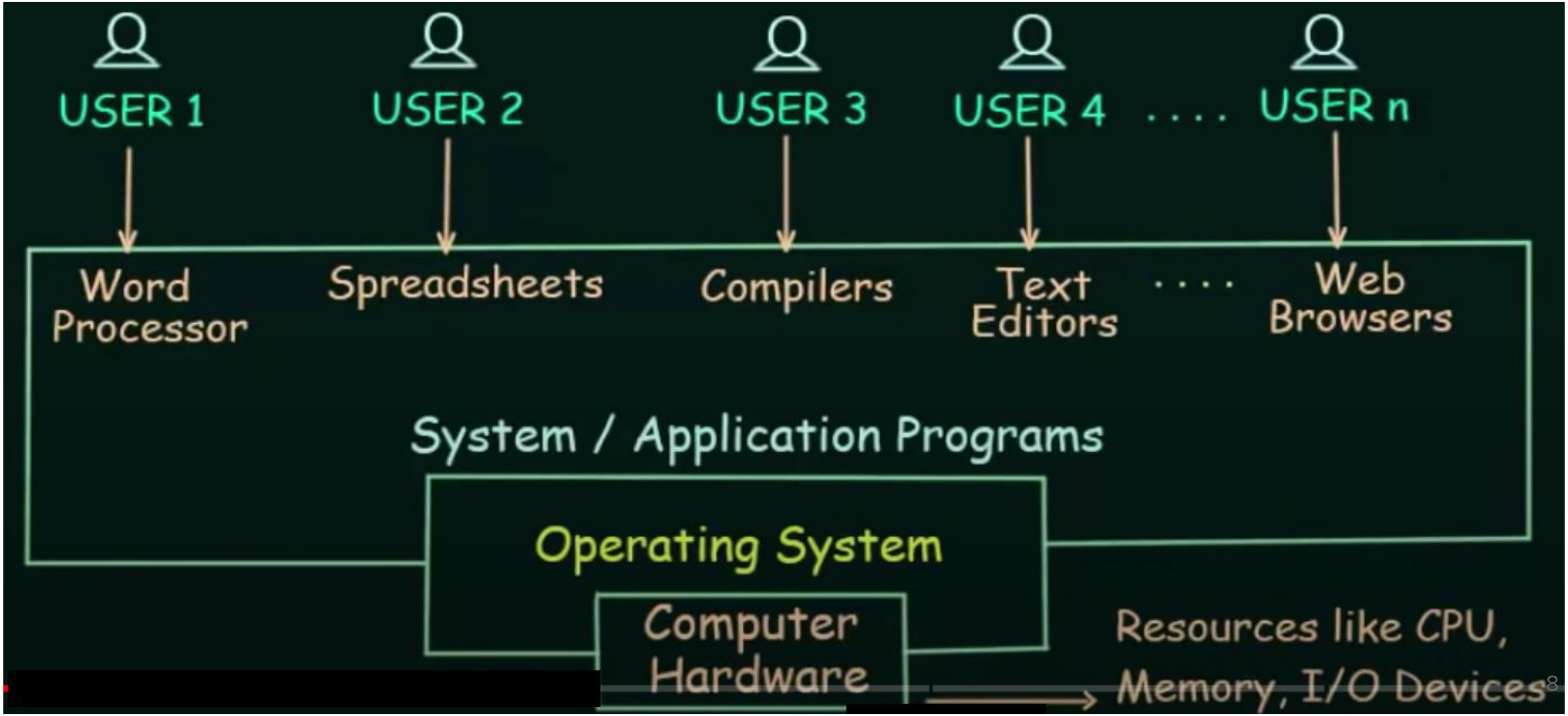
What is Operating system

An operating system is software that manages a computer's hardware.

It also provides a basis for application programs and acts as an intermediary between the computer user and the computer hardware.

Windows
Linux
Ubuntu
Mac OS X
iOS
Android

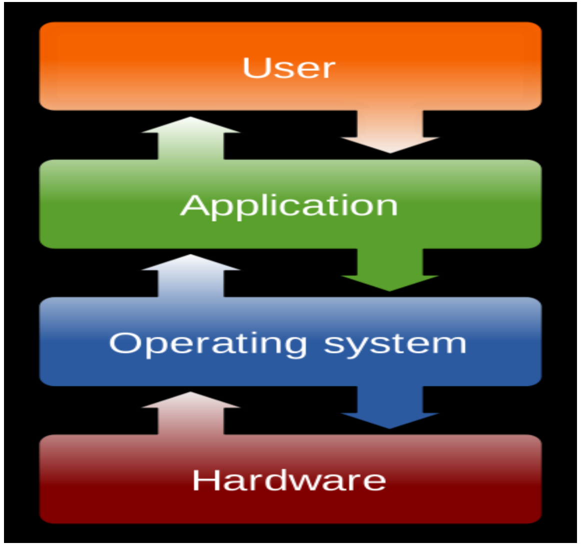
Computer System

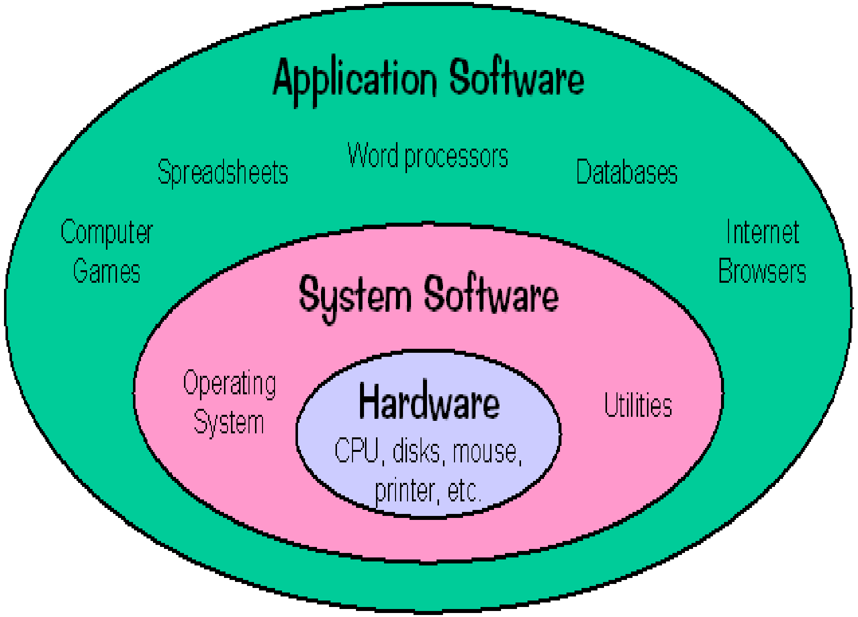


[redacted]

Computer System Components

A computer system can be divided roughly into **four components** – the **hardware,** the **operating system,** the **application programs, and a user**





Computer System Components

- The **hardware** — the central processing unit (the memory, and the input/output (I/O) devices provides the basic computing resources for the system
- The **application programs** — a software or program that is designed to complete a specific task of users
    - Eg. word processors, spreadsheets, and web browsers
- The **operating system** is a *system software* which provides a platform for running and operating other software, namely application programs. It controls the hardware and coordinates its use among the various application programs for the various users.

Facilities provided by OS

- Easy interaction
- Loading & scheduling user programs
- To allocate resources to processes
- Controlling I/O
- Controlling pgm execution
- Scheduling processes
- Managing main memory
- Giving security to users job & files

How is it different from other application software ?

Application software	Os
A computer program which is intended to perform some task classified along.	A system computer program that manages hardware and software resources and provides common services for computer program
It runs when the user desires to run the application.	it boots up when the user wants and run until the user switches off the machine.
It’s examples are Photoshop, VLC player etc.	It’s examples are Microsoft Windows, Linux, MAC
Users interact with application software while using specific applications.	Users never interact with system software as it functions in the background.
It is developed by using c++, c, assembly languages.	It is developed by using virtual basic, c++, c, java.

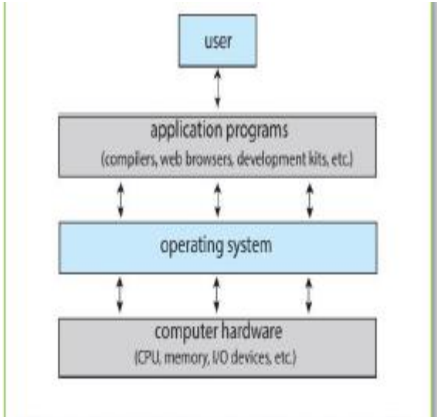
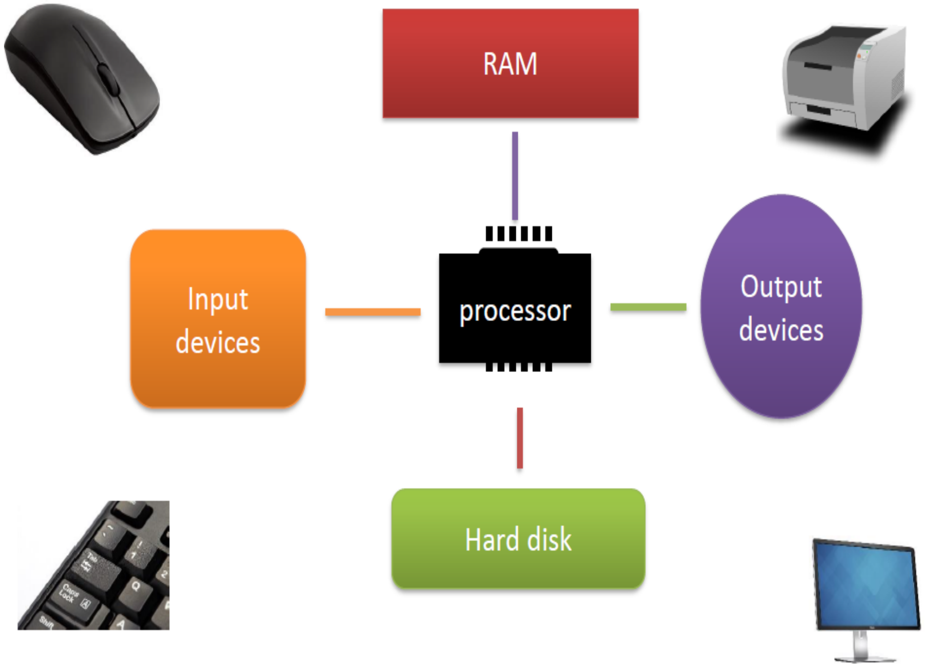

user
application programs
(compilers, web browsers, development kits, etc.)
operating system
computer hardware
(CPU, memory, I/O devices, etc.)

Figure 1.1 Abstract view of the components of a computer system.

Why is OS hardware dependant?

- An OS is "just a program" which needs to run on the CPU, so, it must be written in the CPUs native language and so it is dependent on the CPU (hardware).

Basic computer organization required for OS

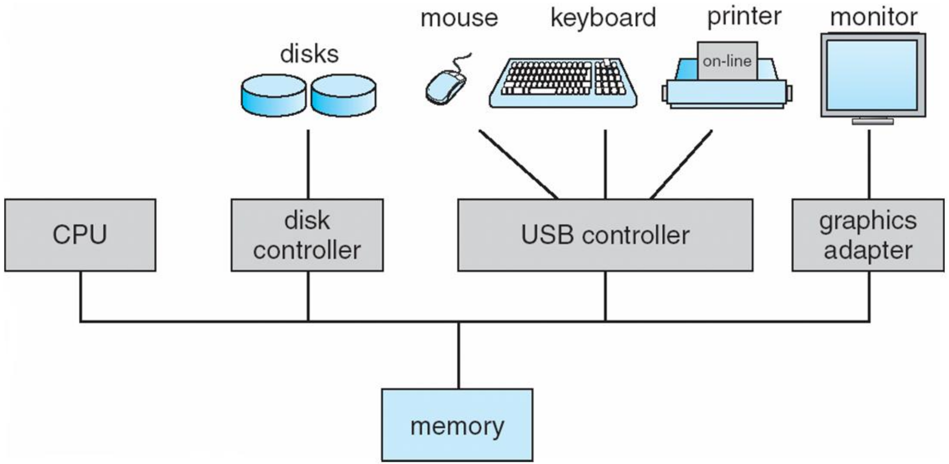


Computer organization is concerned with the way the hardware components operate and the way they are connected together to form the computer system.

The **central processing unit (CPU)** that processes data
The **memory** that holds the programs and data to be processed, and
I/O (input/output) devices as peripherals that communicate with the outside world.

Computer-system operation

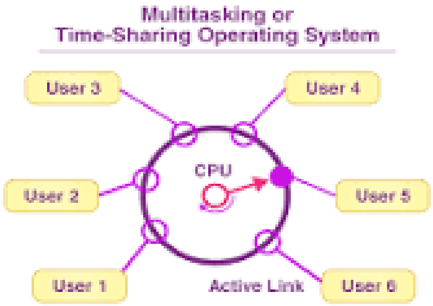
- One or more CPUs, device controllers connect through common bus providing access to shared memory
- Concurrent execution of CPUs and devices competing for memory cycles



Types Of OS

- **Multi tasking OS**
    - Multi tasking refers to execution of multiple tasks by the OS

- **Time-Sharing Operating Systems**
    - Timesharing operating system enables people located at different terminal (shell) to use a single computer system at the same time.
    - The process or time (CPU) which is shared among multiple users is termed as timesharing.



- **Mobile OS**
    - Mobile operating systems are those OS which is especially that are designed to power smartphones, tablets etc.

Types Of OS

- **Real Time OS**
  - A real- time operating system (RTOS) is an operating system with two key features: **predictability** and **determinism**. In an RTOS, repeated tasks are performed within **a tight time boundary**. Real- time systems are used when there are very strict time requirements -- like missile systems, air traffic control systems, robots, etc.
  - **Hard RTOS**
    - These operating systems guarantee that critical tasks be completed within a range of time.
  - **Soft RTOS**
    - This operating system provides some relaxation in the time limit.
    - For example, Multimedia systems, digital audio systems etc.

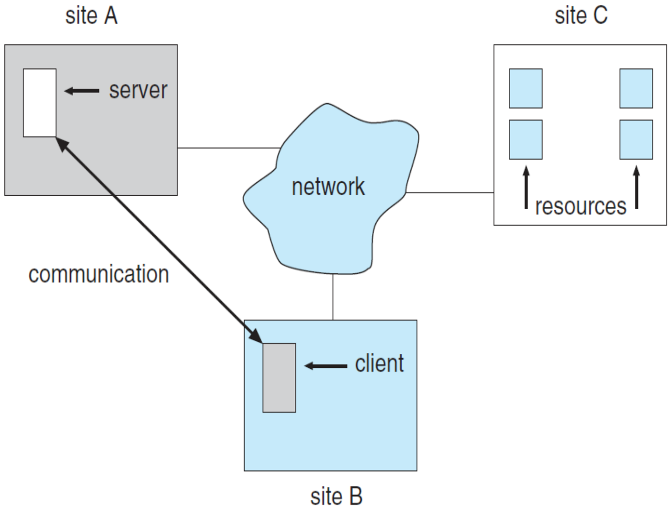
Types Of OS

- **Embedded OS**
  - An embedded system can be thought of as a computer hardware system having software embedded in it.
  - An embedded system can be an independent system or it can be a part of a large system.
  - An embedded system is a microcontroller or microprocessor based system which is designed to perform a specific task. microwave ovens, washing machines and dishwashers, television sets, Video Game consoles, set-top boxes, ATMs, Smart Tv etc.

- **Desktop OS**
  - The desktop OS is the environment where the user controls a personal computer .Ubuntu ,Windows 10
  - It is an operating system designed for usage on servers. It is utilized to give services to a large number of clients.
  - It is a very advanced operating system that can serve several clients simultaneously.

Types Of OS

- **Distributed Systems**
  - A distributed system is a collection of loosely coupled nodes interconnected by a communication network
  - From the point of view of a specific node in a distributed system, the rest of the nodes and their respective resources are remote, whereas its own resources are local.
  - nodes vary in size and function—may include small microprocessors, personal computers, and large general-purpose computer systems.
  - Generally, one node at one site, the server, has a resource that another node at another site, the client (or user), would like to use.



Important functions of Operating System

- Process Management
- Memory Management
- File Management
- Device Management
- Protection and Security

Process Management

- A **process** is **a program in execution**. It is **a unit of work** within the system. Program is a passive entity, process is an active entity.
- In multiprogramming environment, the OS decides which process gets the processor when and for how much time.
- This function is called ***process scheduling***.
- An Operating System does the following activities for processor management
  - Keeps tracks of processor and status of process. The program responsible for this task is known as traffic controller.
  - Allocates the processor (CPU) to a process.
  - De-allocates processor when a process is no longer required.
  - Providing mechanisms for process communication

Memory Management

- **To execute a program all (or part) of the instructions must be in memory**

- The operating system handles the responsibility of storing any data, system programs, and user programs in memory. This function of the operating system is called memory management.

- Memory management activities
  - Keeping ***track*** of ***which parts of memory are currently being used*** and ***by whom***
  - Deciding which ***processes to move into and out of memory***
  - ***Allocating and deallocating memory space*** as needed

File Management

- A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directories.
- An Operating System does the following activities for file management
    - Creating and deleting files and directories
    - Primitives to manipulate files and directories
    - Mapping files onto secondary storage
    - Backup files onto stable (non-volatile) storage media
    - Keeps track of information, location, users, status etc.
    - Decides who gets the resources.
    - Allocates the resources.
    - De-allocates the resources.

Device Management

- An Operating System manages device communication via their respective drivers
- It does the following activities for device management
  - Keeps track of all the devices – Program responsible for this task is known as the I/O controller
  - Decides which process gets the device & for how much time
  - Allocates the device in an efficient way
  - De-allocates the device

Protection and Security

- The operating system uses **password protection** to protect user data and similar other techniques. It also prevents unauthorized access to programs and user data.
- Monitors overall system health to help improve performance. It records the response time between service requests and system response to having a complete view of the system health. This can help improve performance by providing important information needed to troubleshoot problems
- The operating system uses password protection to protect user data and similar other techniques. it also prevents unauthorized access to programs and user data.

Protection and Security

- Protection – any mechanism for controlling access of processes or users to resources defined by the OS
- Security – defense of the system against internal and external attacks—
  - Huge range, including denial-of-service, worms, viruses, identity theft, theft of service
- Systems generally first distinguish among users, to determine who can do what
  - User identities (user IDs, security IDs) include name and associated number, one per user
  - User ID then associated with all files, processes of that user to determine access control
  - Group identifier (group ID) allows set of users to be defined and controls managed, then also associated with each process, file
  - Privilege escalation allows user to change to effective ID with more rights

Dual mode of operation

- **Dual-mode** operation allows OS to protect itself and other system components
    - **User mode** and **kernel mode**
    - **Mode bit** provided by hardware
        - Provides ability to distinguish when system is running user code or kernel code
        - Some instructions designated as **privileged**, only executable in kernel mode
        - System call changes mode to kernel, return from call resets it to user

User space and Kernel space

- When the computer system is run by user applications like creating a text document or using any application program, then the system is in **User space**

- The system starts in **kernel space** when it boots and after the operating system is loaded, it executes applications in user mode.

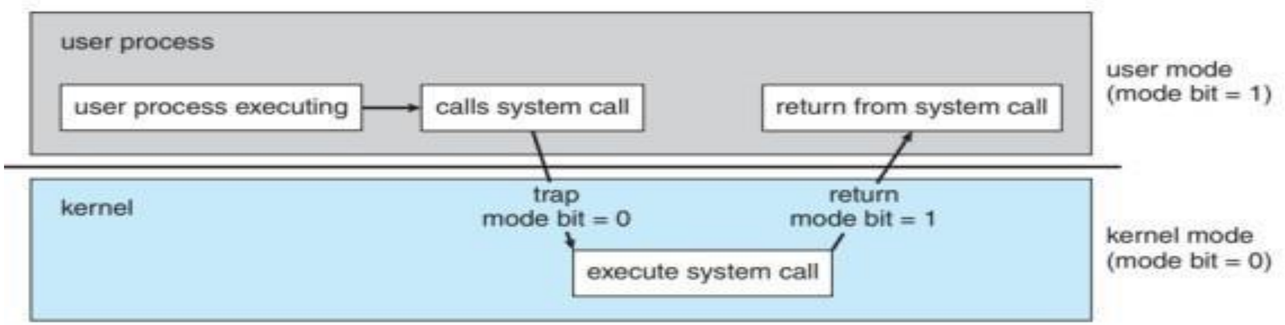


Figure 1.10 Transition from user to kernel mode.

User mode Vs Kernel mode

- The operating system puts the CPU in user mode when a user program is in execution
- When system runs a user application like creating a text document or using any application program, then the system is in user mode.
- When CPU is in user mode, the programs don't have direct access to memory and hardware resources.
- system will be in a safe state even if a program in user mode crashes.

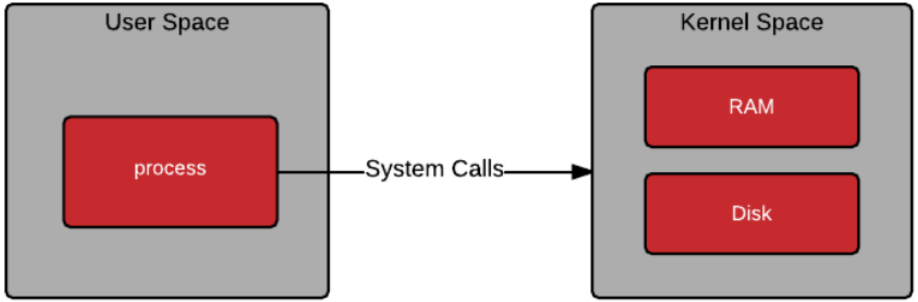
- The operating system puts the CPU in kernel mode when it is executing in the kernel
- At system boot time, the hardware starts in kernel mode
- The operating system is then loaded and starts user applications in user mode.
- The CPU can execute certain instruction only when it is in the kernel mode. These instruction are called **privilege instruction**
- To provide protection to the hardware, we have privileged instructions which execute only in kernel mode
- If a program crashes in kernel mode, the entire system will be halted

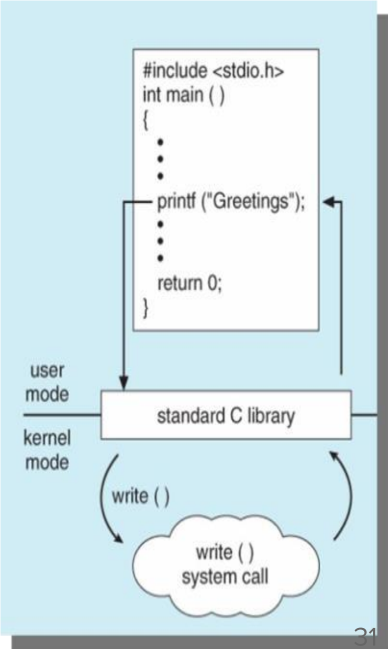
Interrupts

- The occurrence of an event is usually signalled by an interrupt
- Interrupts are of 2 types
  - Hardware interrupts
    - H/w may trigger an interrupt anytime by sending a signal to the CPU usually through the s/m bus
  - Software interrupts
    - S/w may trigger an interrupt by executing a special operation called system call

System Calls

- A **system call** is a method for a computer program to request a service from the kernel of the operating system on which it is running.
- When a program in user mode requires access to RAM or a hardware resource, it must ask the kernel to provide access to that resource. This is done via a system call.
- System calls are the only entry points into the kernel system.

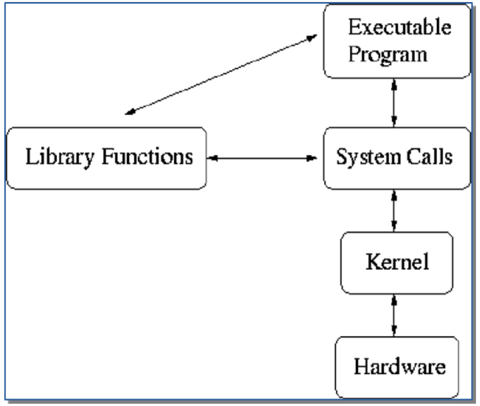




```
#include <stdio.h>
int main ( )
{
  •
  •
  •
  printf ("Greetings");
  •
  •
  •
  return 0;
}
```

System Calls

- System calls are made by the user level programs in the following situations:
  - Creating , opening , closing and deleting files in the file system.
  - Creating and managing new processes.
  - Creating a connection in the network, sending and receiving packets.
  - Requesting access to a hardware device, like a mouse or a printer.


Executable Program
Library Functions
System Calls
Kernel
Hardware