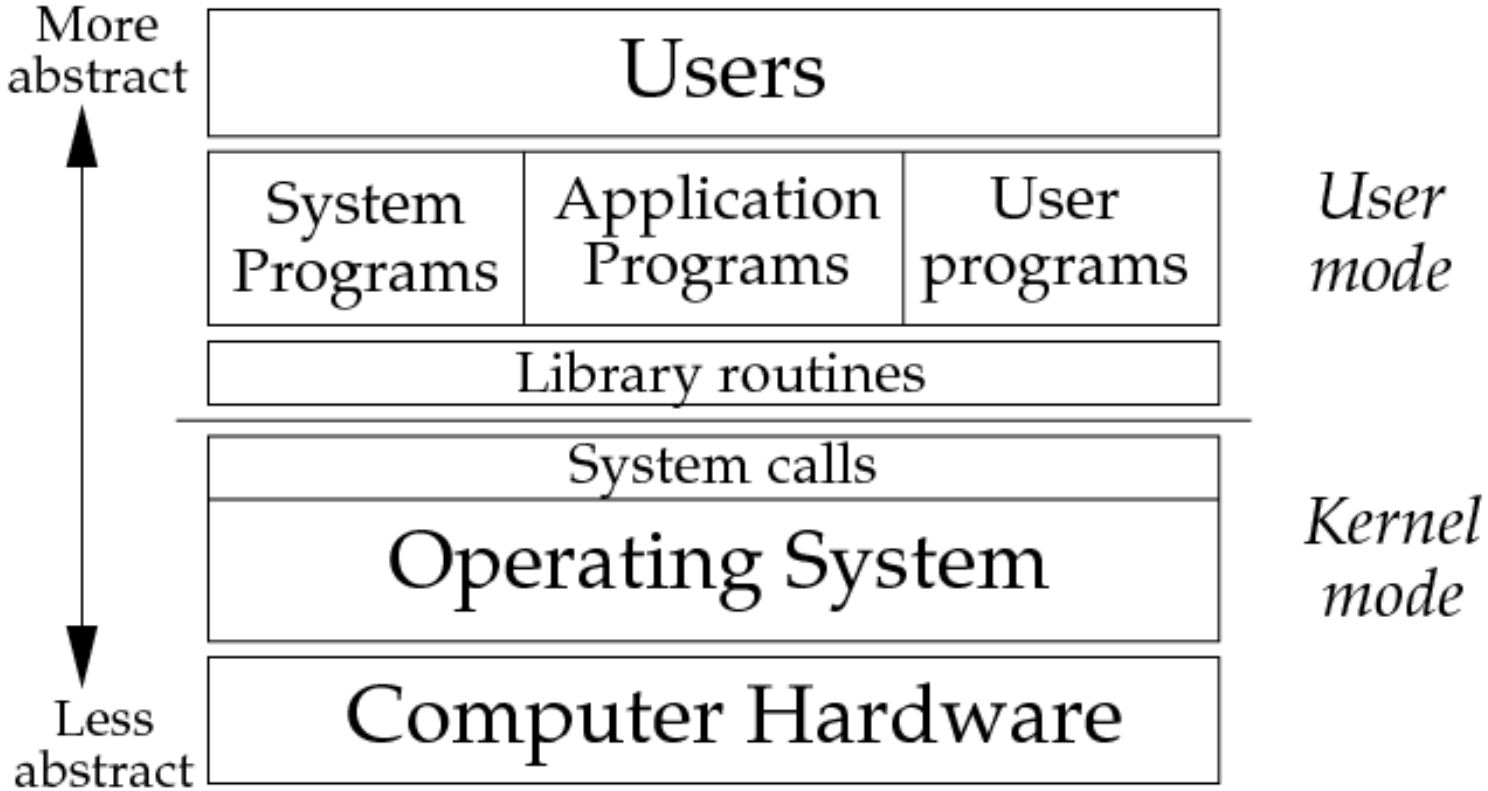
# Introduction

Prof. Tasneem Mirza

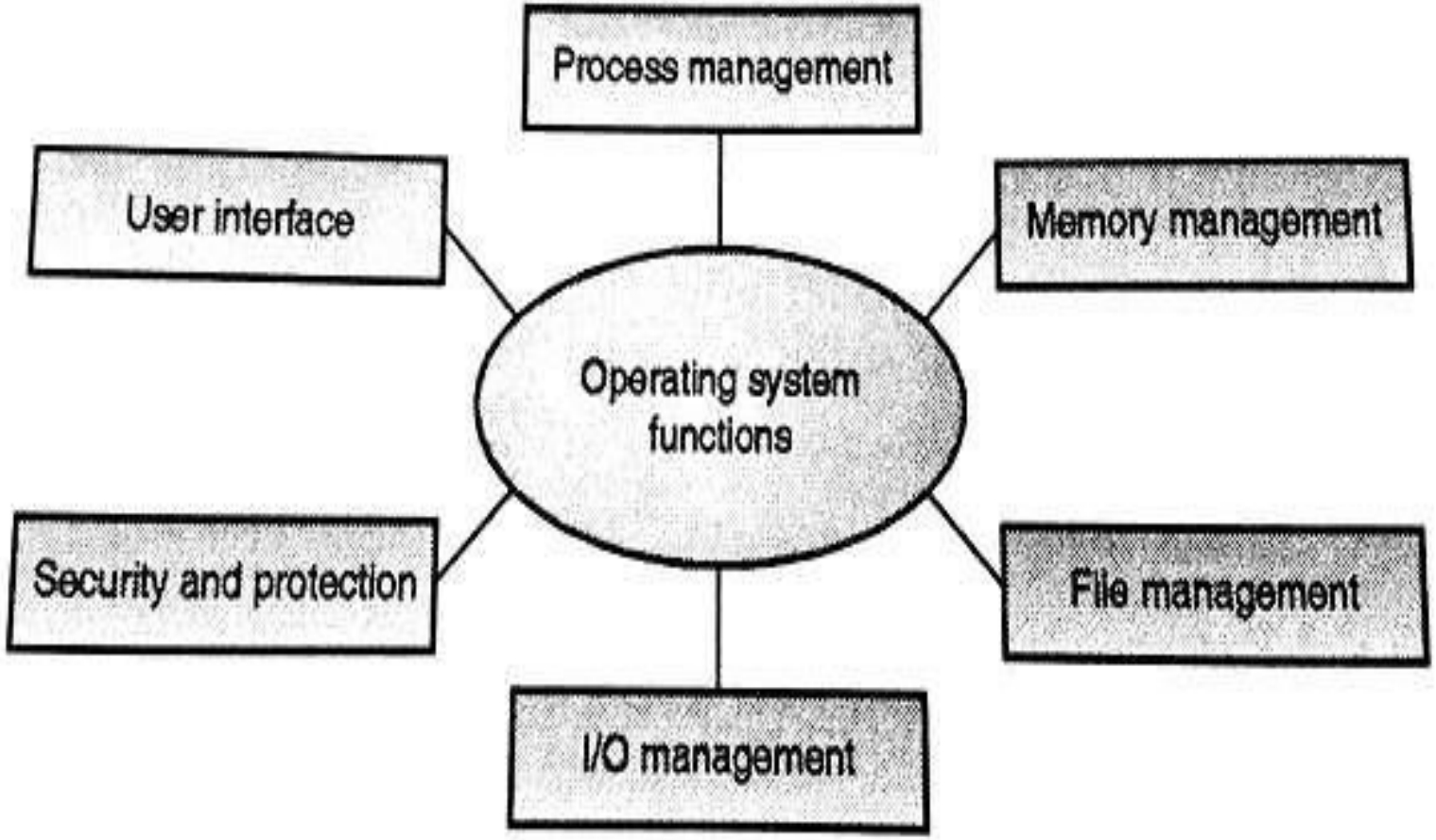
# What is an Operating System ?

- An Operating System is a **program** that acts as an intermediary/interface between a user of a computer and the computer hardware.

- **OS goals:**
  - Control/execute user/application programs.
  - Make the computer system convenient to use.
  - Ease the solving of user problems.
  - Use the computer hardware in an efficient manner.


More abstract
Users
System Programs
Application Programs
User programs
User mode
Library routines
System calls
Operating System
Kernel mode
Computer Hardware
Less abstract

# Functions of Operating System



# 1. Process Management

- **A *process* is a program in execution.**
- A process needs certain resources, including CPU time, memory, files, and I/O devices to accomplish its task.
- Simultaneous execution leads to multiple processes. Hence creation, execution and termination of a process are the most basic functionality of an OS
- If processes are dependent, than they may try to share same resources. thus task of process synchronization comes to the picture.
- If processes are independent, than a due care needs to be taken to avoid their overlapping in memory area.
- Based on priority, it is important to allow more important processes to execute first than others.

# 2. Memory management

- Memory is a large array of words or bytes, each with its own address.
- Main memory is a **volatile** storage device. When the computer made turn off everything stored in RAM will be erased automatically.
- In addition to the physical RAM installed in your computer, most modern operating systems allow your computer to use a *virtual memory system*. *Virtual memory allows your computer to use part of a permanent storage device (such as a hard disk) as extra memory.*
- The operating system is responsible for the following activities in connections with memory management:
  - Keep track of which parts of memory are currently being used and by whom.
  - Decide which processes to load when memory space becomes available.
  - Allocate and de-allocate memory space as needed.

# 3. File Management

- A file is a collection of related information defined by its creator.
- *File systems provide the conventions for the encoding, storage and management of data on a storage device such as a hard disk.*
- The operating system is responsible for the following activities in connections with file management:
  - ✦ File creation and deletion.
  - ✦ Directory creation and deletion.
  - ✦ Support of primitives for manipulating files and directories.
  - ✦ Mapping files onto secondary storage.
  - ✦ File backup on stable (nonvolatile) storage media.

# 4. Device Management or I/O Management

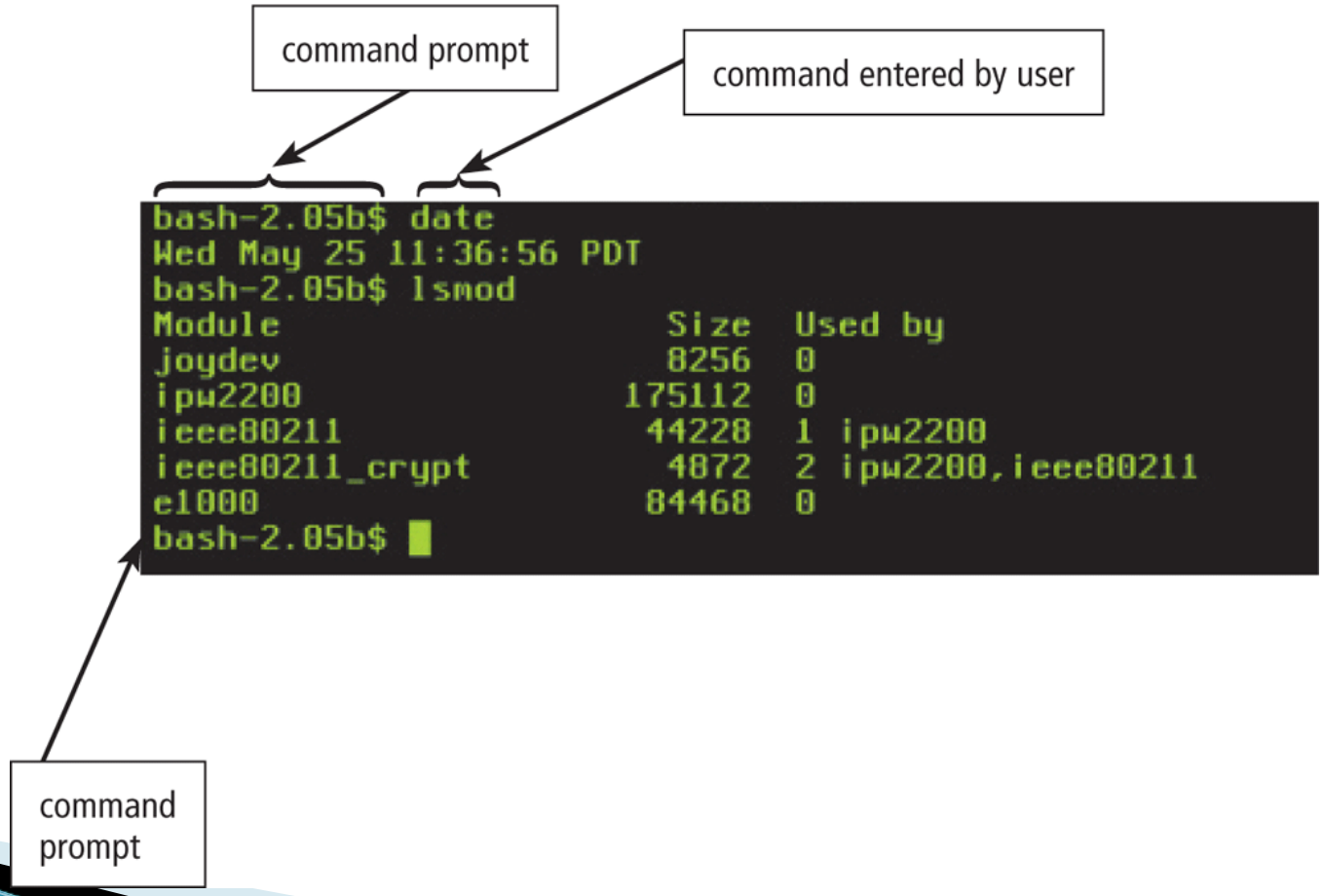
- *Device controllers are components on the motherboard (or on expansion cards) that act as an interface between the CPU and the actual device.*
- *Device drivers, which are the operating system software components that interact with the devices controllers.*
- OS performs the following activities for device management.
  - Keeps tracks of all devices connected to system.
  - Designates a program responsible for every device known as Input/output controller.
  - Decides which process gets access to a certain device and for how long.
  - Allocates devices in an effective and efficient way.
  - Deallocates devices when they are no longer required.

# 6. User Interface Mechanism

- A **user interface** (**UI**) controls how you enter data and instructions and how information is displayed on the screen
- There are two types of user interfaces
  1. Command Line Interface
  2. Graphical user Interface

# 1. Command-line interface

- In a command-line interface, a user types commands represented by short keywords or abbreviations or presses special keys on the keyboard to enter data and instructions



# 2. Graphical User Interface

- With a graphical user interface (GUI), you interact with menus and visual images

# Kernel

- Kernel is a software code that reside in central core of OS. It has complete control over system.
- When operation system boots, kernel is first part of OS to load in main memory.
- Kernel remains in main memory for entire duration of computer session. The kernel code is usually loaded in to protected area of memory.
- Kernel performs it's task like executing processes and handling interrupts in kernel space.
- User performs it's task in user area of memory.
- This memory separation is made in order to prevent user data and kernel data from interfering with each other.
- Kernel does not interact directly with user, but it interacts using SHELL and other programs and hardware.

# Kernel cont…

- Kernel includes:-

  1. Scheduler: It allocates the Kernel's processing time to various processes.

  2. Supervisor: It grants permission to use computer system resources to each process.

  3. Interrupt handler : It handles all requests from the various hardware devices which compete for kernel services.

  4. Memory manager : allocates space in memory for all users of kernel service.

- kernel provides services for process management, file management, I/O management, memory management.
- System calls are used to provide these type of services.

# System Call

- **System call** is the programmatic way in which a computer program/user application requests a service from the kernel of the operating system on which it is executed.
- Application program is just a user-process. Due to security reasons , user applications are not given access to privileged resources(the ones controlled by OS).
- When they need to **do any I/O** or have **some more memory** or **spawn a process** or wait for **signal/interrupt,** it requests operating system to facilitate all these. This **request is made through System Call.**
- System calls are also called **software-interrupts.**

# Types of Kernels Monolithic kernel Micro kernel (Modular kernel)

1. Monolithic kernel
2. Micro kernel (Modular kernel)

**Monolithic Kernel:**

All operating system services run along the main kernel thread in a monolithic kernel, which also resides in the same memory area, thereby providing powerful and rich hardware access.

**Micro Kernel:**

Define a simple abstraction over hardware that use primitives or system calls to implement minimum OS services such as multitasking, memory management and interprocess communication.

# Shell

- What is a Shell.
- A shell is a program that acts as the interface between the user and UNIX system.
- It acts as an interpreter or translator
- It allows user to enter commands for the operation system to execute