OPERATING SYSTEMS

Operating System Framework

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

OPERATING SYSTEMS

Course Syllabus - Unit 1

UNIT 1: Introduction and Process Management

Operating-System Structure & Operations, Kernel Data Structures,Computing Environments, Operating-System Services, OperatingSystem Design and Implementation. Process concept: Process in memory, Process State, Process Control Block, Process Creation and Termination, CPU Scheduling and Scheduling Algorithms, IPC - Shared Memory & Message Passing, Pipes - Named and Ordinary. Case Study: Linux/Windows Scheduling Policies.

OPERATING SYSTEMS

Course Outline

Class No.	Chapter Title / Reference Literature	Topics to be covered	% of Portions covered	
			Reference chapter	Cumulative
1	1.1-1.2	What Operating Systems Do, Computer-System Organization?	1	21.4
2	1.3,1.4,1.5	Computer-System Architecture, Operating-System Structure & Operations	1	
3	1.10,1.11	Kernel Data Structures, Computing Environments	1	
4	2.1,2.6	Operating-System Services, Operating System Design and Implementation	2	
5	3.1-3.3	Process concept: Process in memory, Process State, Process Control Block, Process Creation and Termination	3	
6	5.1-5.2	CPU Scheduling: Basic Concepts, Scheduling Criteria	5	
7	5.3	Scheduling Algorithms: First-Come, First-Served Scheduling, Shortest-Job-First Scheduling	5	
8	5.3	Scheduling Algorithms: Shortest-Job-First Scheduling (Pre-emptive), Priority Scheduling	5	
9	5.3	Round-Robin Scheduling, Multi-level Queue, Multi-Level Feedback Queue Scheduling	5	
10	5.5,5.6	Multiple-Processor Scheduling, Real-Time CPU Scheduling	5	
11	5.7	Case Study: Linux/Windows Scheduling Policies	5	
12	3.4,3.6.3	IPC – Shared Memory & Message Passing, Pipes – Named and Ordinary	3,6	

Topics Outline

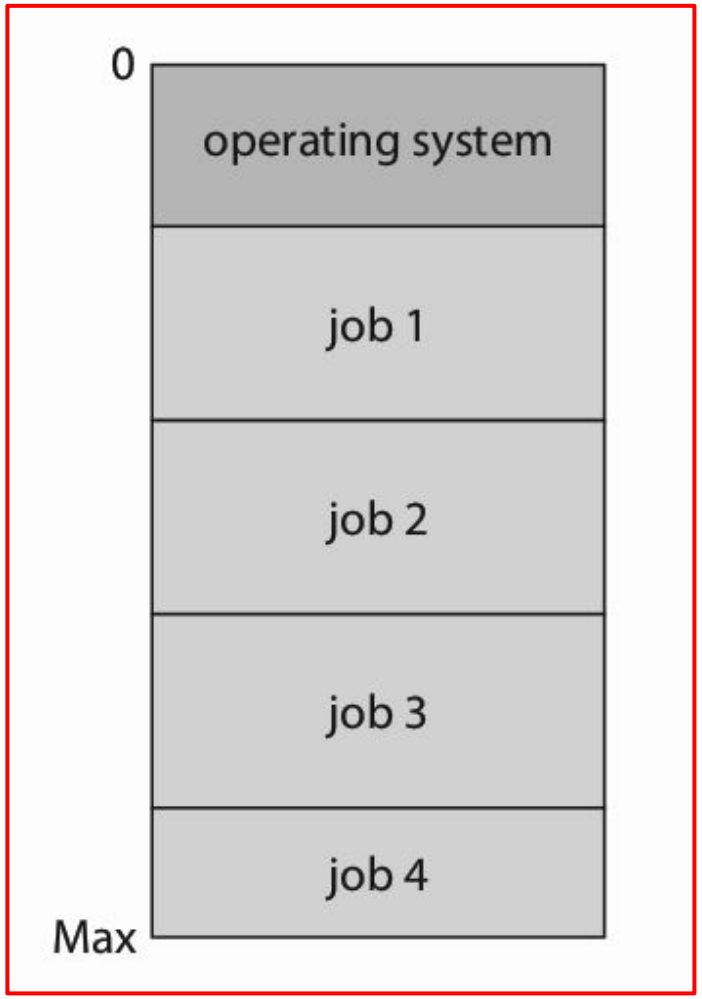
- Operating System structure and Operations
- Kernel Data Structures
- Computing Environment
- Operating System Services
- Operating System Design
- Typical Q and As
- Operating System Implementation

Operating System Structure

- An Operating System provides the **environment** within which programs are executed.
- Operating Systems are generally organized along many different lines.
- One of the most important aspects of Operating system is its ability to **handle multiple programs.**
- A single program cannot, in general, keep either the CPU or the I/O devices busy at all times because both CPU and I/O happen inherently concurrent
- Single users frequently have multiple programs running during the same time interval(s)
- Multiprogramming increases **CPU utilization** by organizing jobs (code and data) so that the CPU always has one to execute.

Operating System Structure

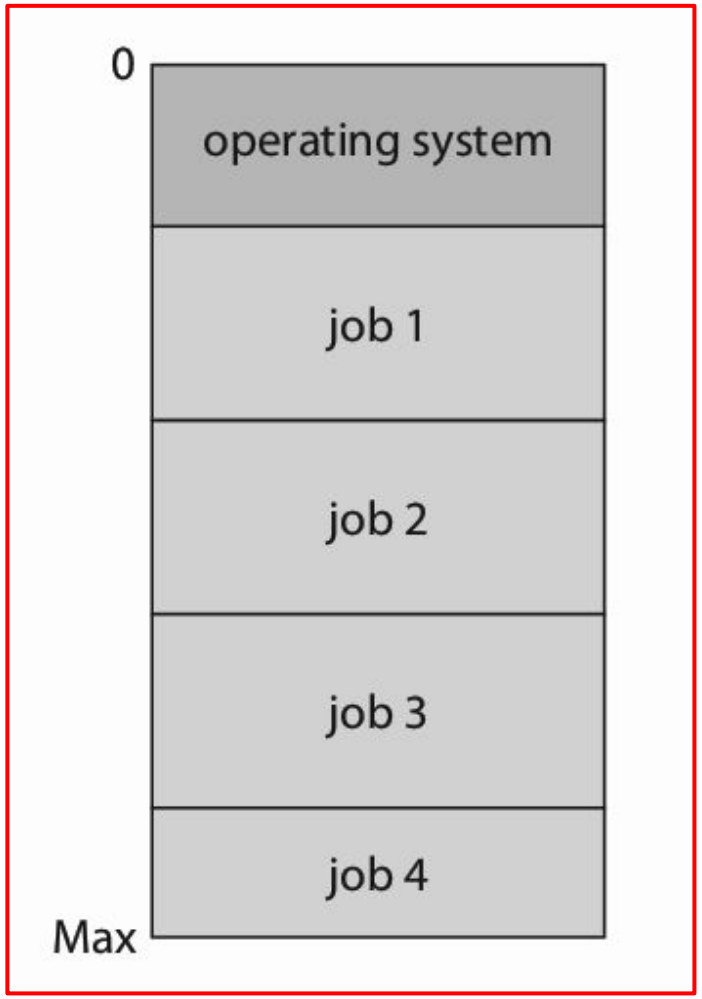
- The operating system has the capability to maintain several jobs in memory simultaneously
- Main memory too small is handled by accommodating all jobs, by initially keeping these jobs on the disk in the job pool.
- The set of jobs in memory can be a subset of the jobs kept in the job pool.
- The operating system picks and begins to execute one of the jobs in memory.
- Multiprogrammed systems provide an environment in which the various system resources (for example, CPU , memory, and peripheral devices) are utilized effectively, but they do not provide for user interaction with the computer system.



Memory layout for a multiprogramming system

Operating System Structure

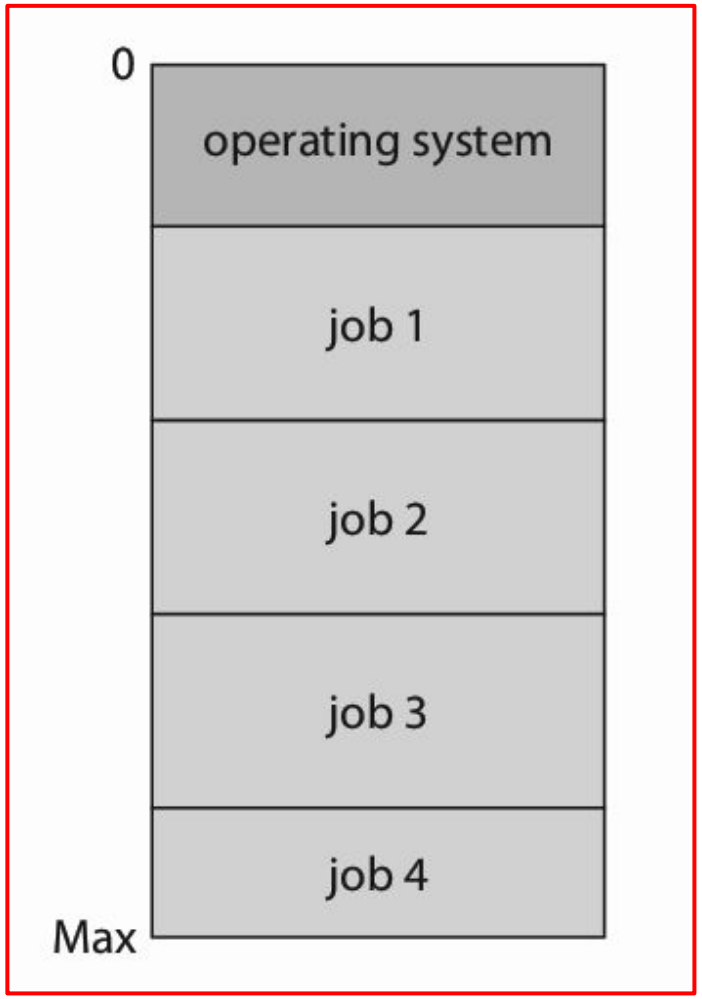
- **Time sharing (or multitasking)** is a logical extension of multiprogramming.
- In time-sharing systems, the CPU executes multiple jobs by switching among them, but the switches occur so frequently that the users can interact with each program while it is running
- The response time in such systems should be short, typically less than one second
- Each user has at least one separate program in memory.
- A program loaded into memory and executing is called a process.
- More about the concept of Process will be covered in the next Chapter



Memory layout for a multiprogramming system

Operating System Structure

- To achieve faster response time swapping is used whereby processes are swapped in and out of main memory to the disk.
- A more common method for ensuring reasonable response time is virtual memory
- More on memory management will be studied in later chapter(s) on Physical and Virtual Management



Memory layout for a multiprogramming system

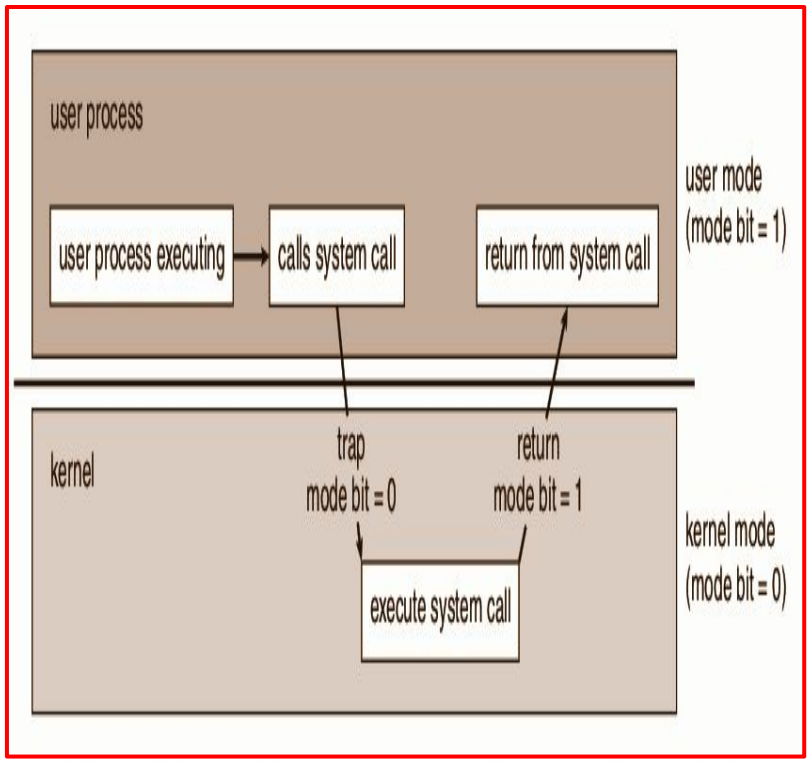
Operating System Operations

- Almost all modern operating systems are interrupt driven

- Events are typically signaled by the occurrence of an **interrupt** or a **trap**.

Operating System Operations - Dual Mode

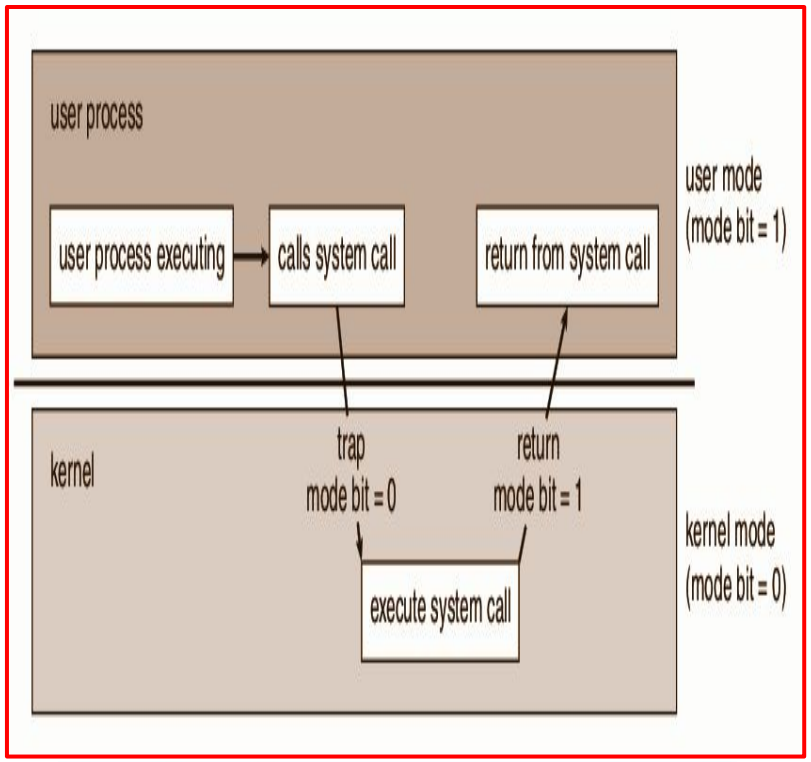
- Most computer systems takes the approach of providing hardware support that allows us to differentiate among various modes of execution, to ensure proper execution of the OS
- At the very least, one needs two separate modes of operation: **user mode and kernel mode (also called supervisor mode, system mode, or privileged mode).**
- A bit, called the mode bit, is added to the hardware of the computer to indicate the current mode: **kernel (0) or user (1).**
- The dual mode of operation provides us with the means for protecting the operating system from errant users and errant users from one another.



Transition from user to kernel mode

Operating System Operations - Dual Mode

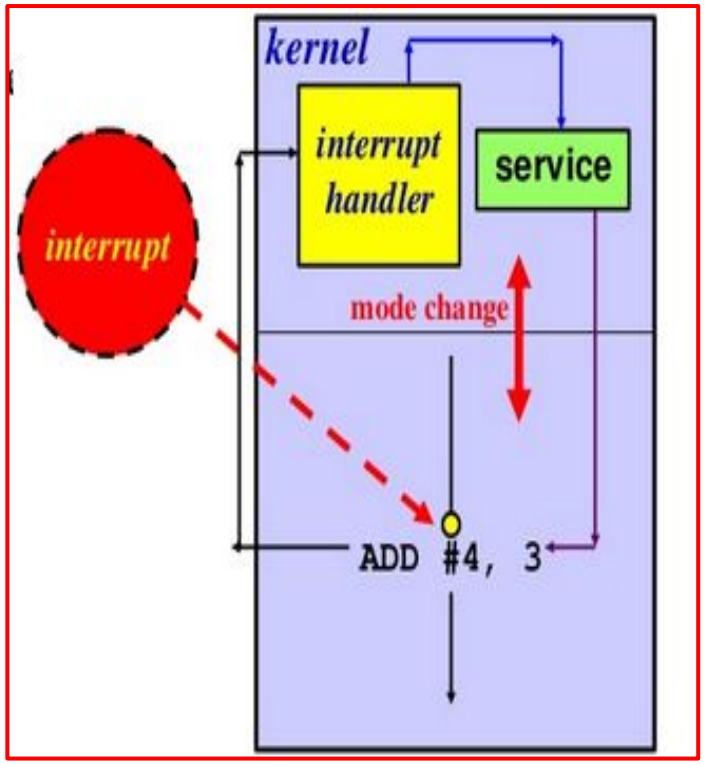
- This protection is achieved by designating some of the machine instructions that may cause harm as privileged instructions.
- The hardware allows privileged instructions to be executed only in kernel mode.
- If an attempt is made to execute a privileged instruction in user mode, the hardware does not execute the instruction but rather treats it as illegal and traps it to the operating system.



Transition from user to kernel mode

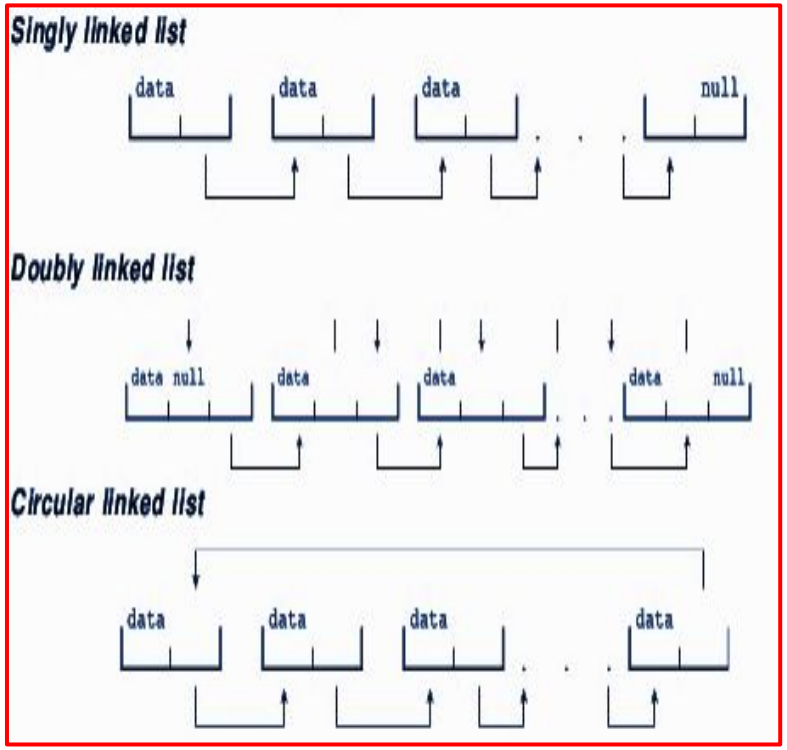
Operating System Operations - Timer

- The operating system maintaining **control** over the CPU must be ensured
- A **timer** can be set to interrupt the computer after a specified period.
- The period may be fixed (for example, 1/60 second) or variable (for example, from 1 millisecond to 1 second).
- A variable timer is generally implemented by a fixed-rate clock and a counter.



Kernel Data Structures

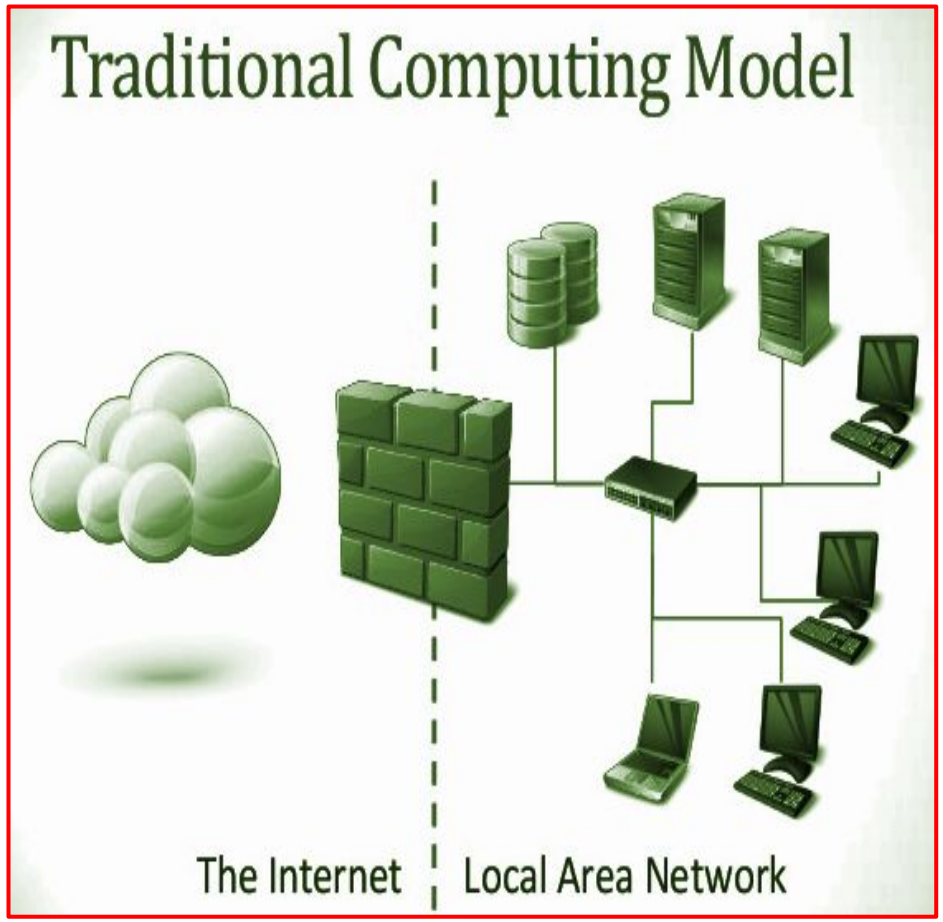
- Lists, Stacks, and Queues
- Trees
- Hash Functions and Maps
- Bitmaps



Computing Environments

Traditional Computing

- Portals
- Network computers
- Wireless Networks
- Firewalls



Computing Environments

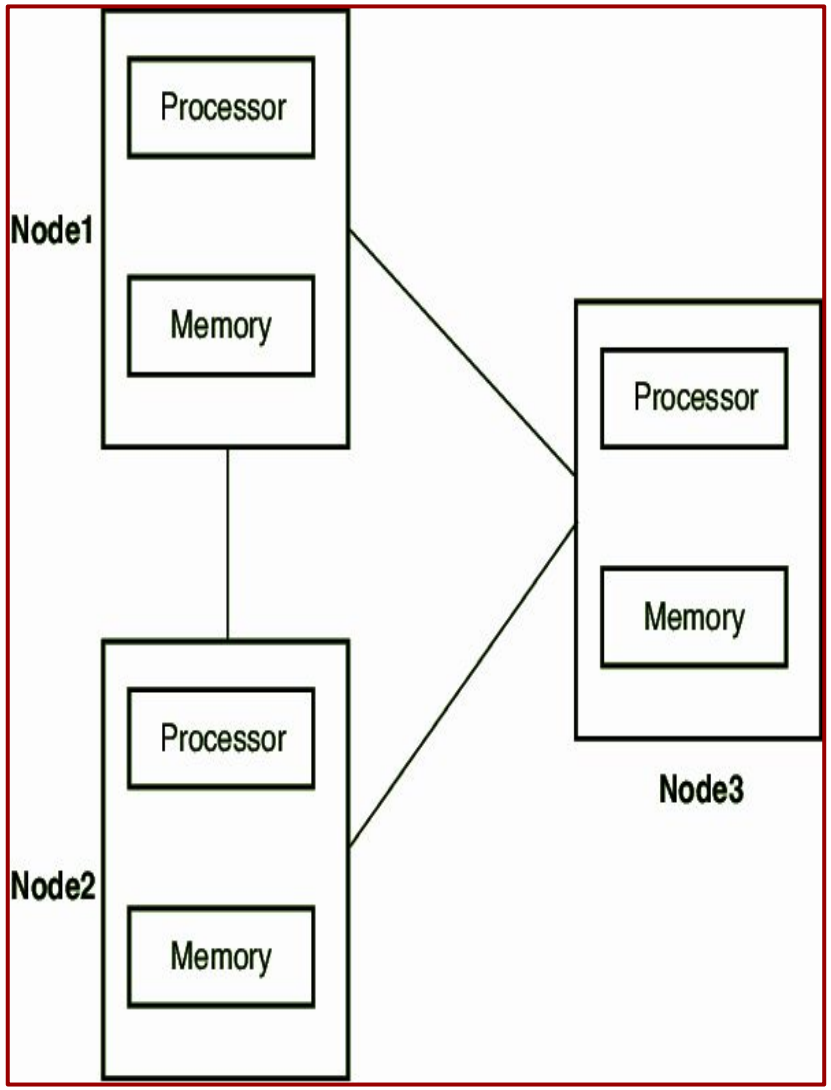
Mobile Computing

- Mobile computing refers to **computing** on handheld **smartphones** and **tablet** computers.
- Two operating systems currently dominate mobile computing: **Apple iOS and Google Android.**
- iOS was designed to run on Apple iPhone and iPad mobile devices.
- Android powers smartphones and tablet computers available from many manufacturers.

Computing Environments

Distributed Systems

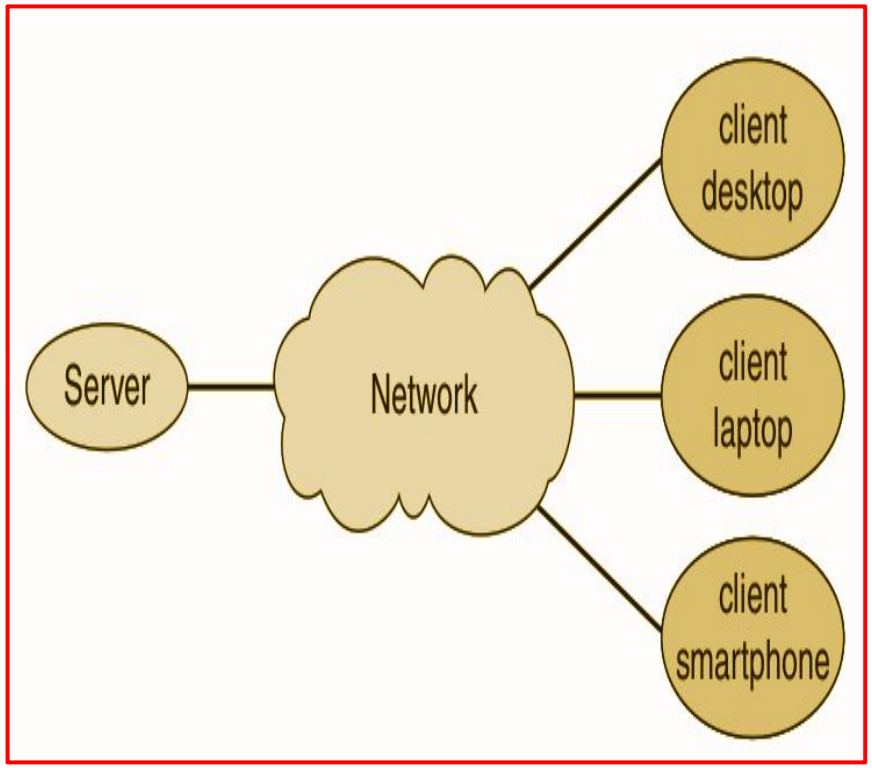
- Network
- TCP/IP - Transmission Control Protocol / Internet Protocol
- Local Area Network - LAN
- Wide Area Network - WAN
- Metropolitan Area Network - MAN
- Personal Area Network - PAN
- A **network operating system** is an operating system that provides features such as file sharing across the network, along with a communication scheme that allows different processes on different computers to exchange messages.



Computing Environments

Client Server Computing

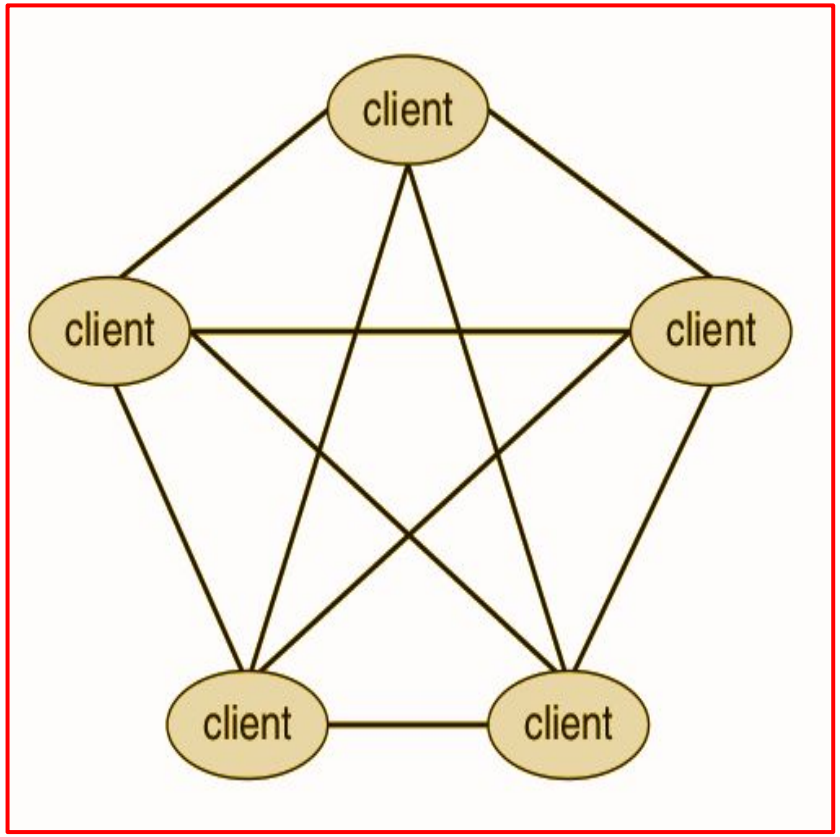
- Compute -Server System
- File-Server System



Computing Environments

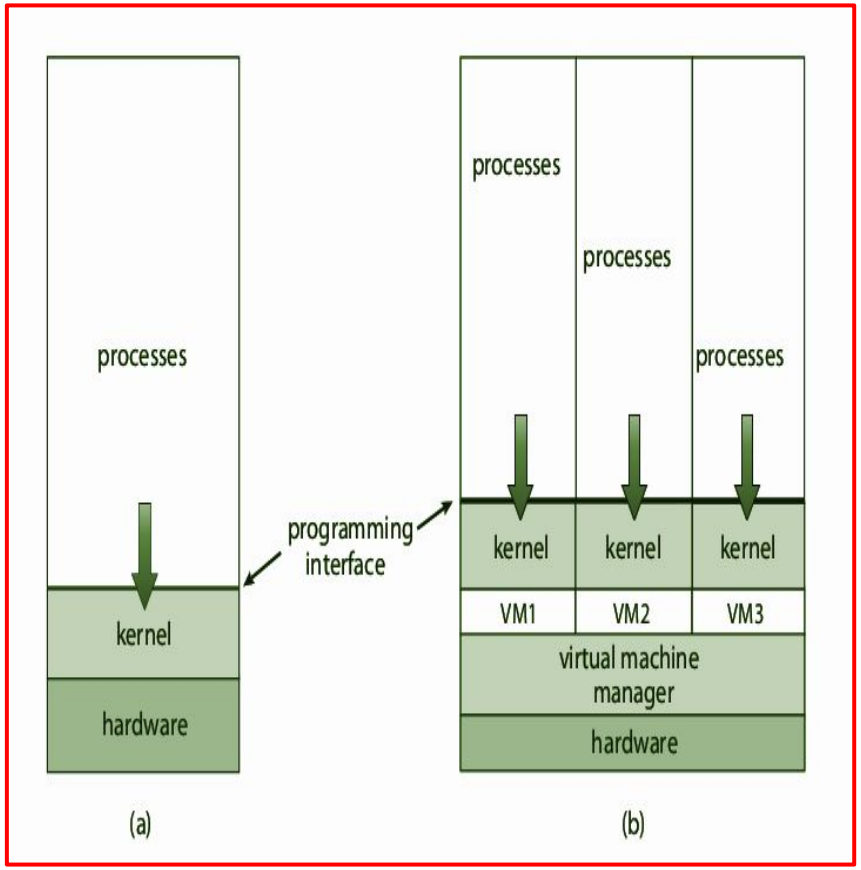
Peer - Peer Computing

- AKA P2P system



Virtualization

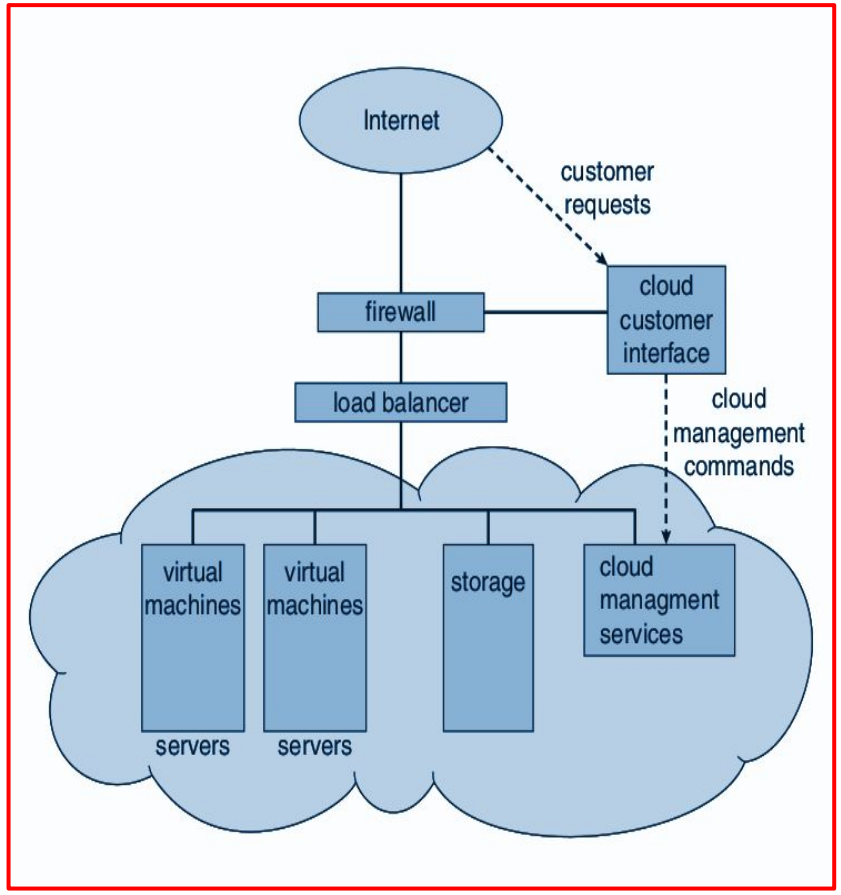
- Virtualization is a technology that allows operating systems to run as applications within other operating systems.
- Broadly speaking, virtualization is one member of a class of software that also includes emulation.
- Emulation is used when the source CPU type is different from the target CPU type.



Computing Environments

Cloud Computing

- **Cloud computing** is a type of computing that delivers computing, storage, and even applications as a service across a network.
- **Public cloud** => a cloud available via the Internet to anyone willing to pay for the services
- **Private cloud**=> a cloud run by a company for that company's own use
- **Hybrid cloud** => a cloud that includes both public and private cloud components



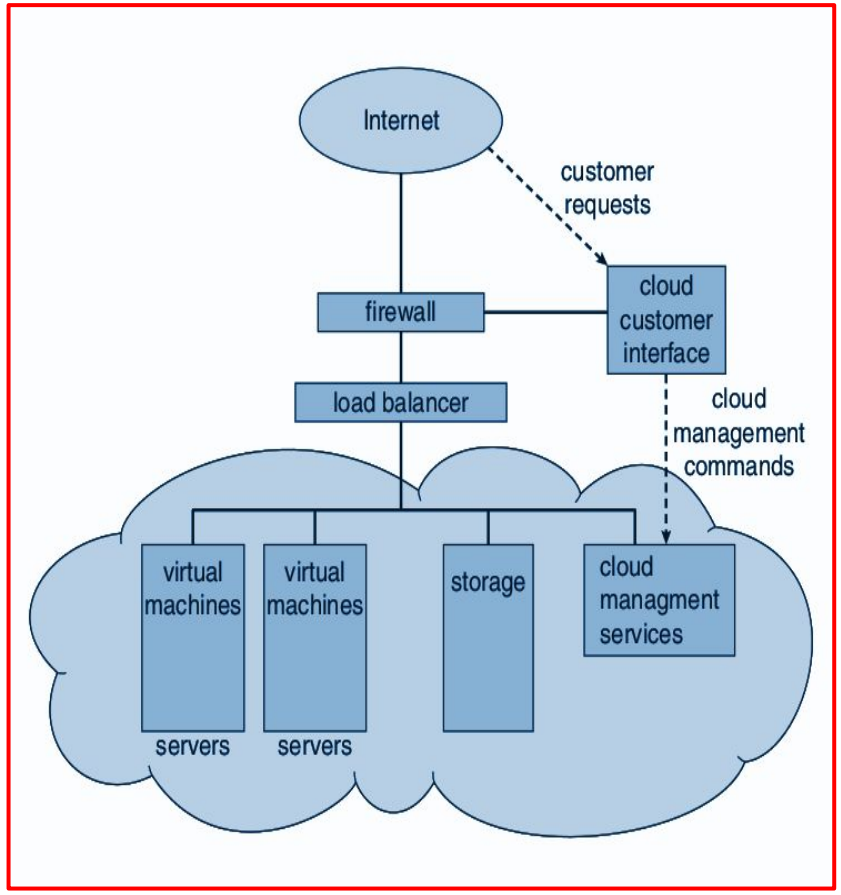
Computing Environments

Cloud Computing

- **Software as a service (SaaS)** => one or more applications (such as word processors or spreadsheets) available via the Internet

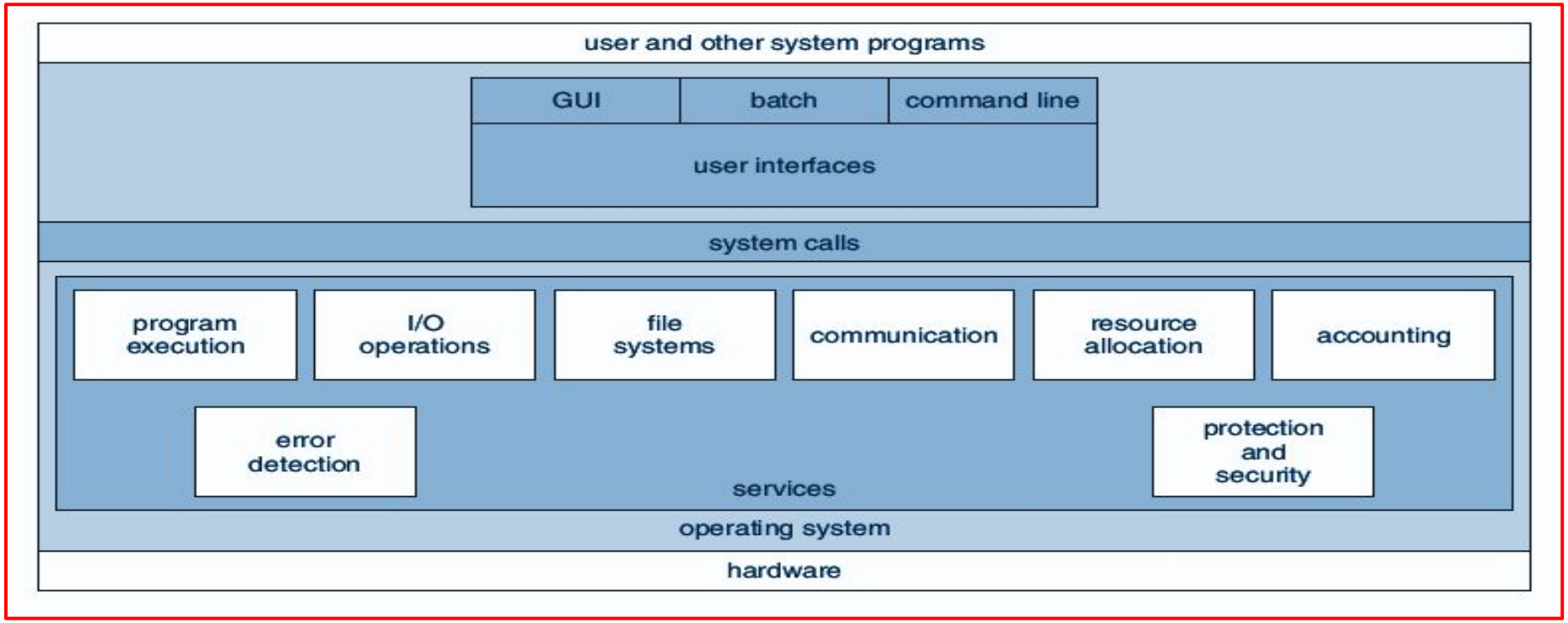
- **Platform as a service (PaaS)** => a software stack ready for application use via the Internet (for example, a database server)

- **Infrastructure as a service (IaaS)**=> servers or storage available over the Internet for example, storage available for making backup copies of production data



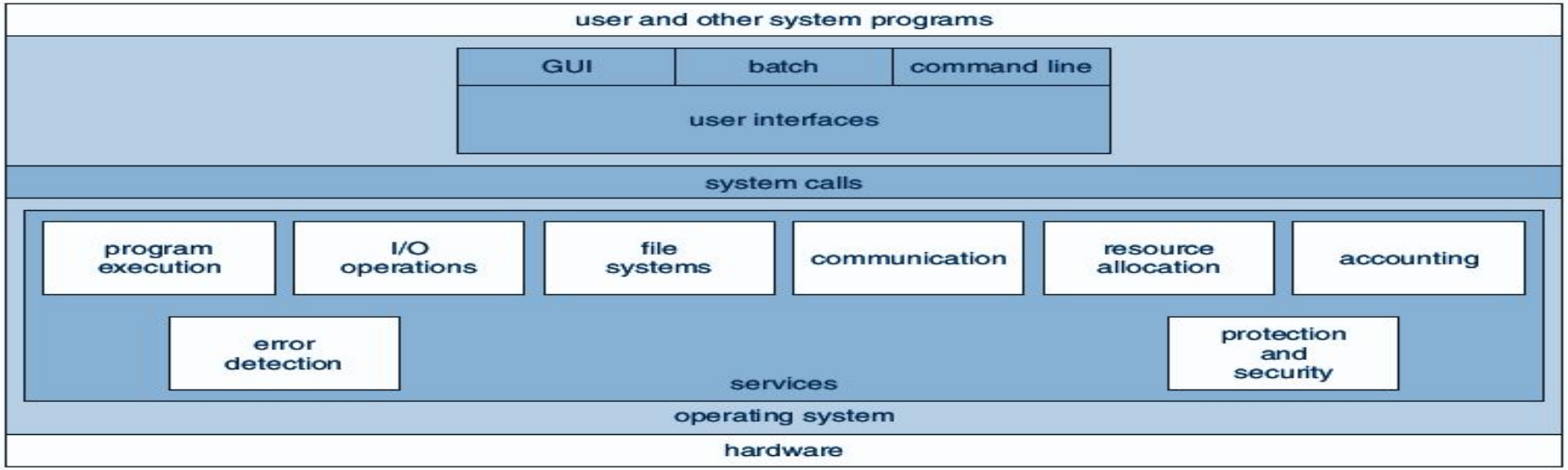
Operating System Services

- An operating system provides an **environment** for the **execution** of programs.
- It provides certain services to programs and to the users of those programs.
- The specific services provided, of course, differ from one operating system to another, but we can identify common classes.



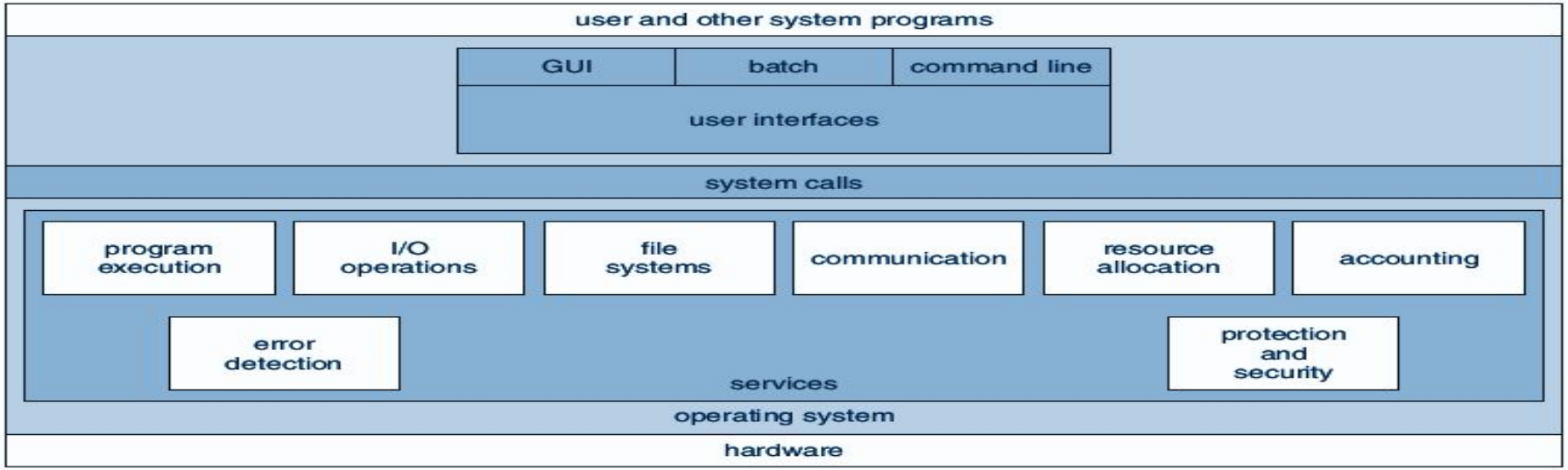
Operating System Services

- **User interface**
  - Almost all operating systems have a user interface (UI).
  - This interface can take several forms.
    - **One is a command-line interface(CLI)**, which uses text commands and a method for entering them
    - **Batch interface,** in which commands and directives to control those commands are entered into files, and those files are executed.
    - **Graphical user interface (GUI)** is used in almost all current OS. Here, the interface is a window system with a pointing device to direct I/O , choose from menus, and make selections and a keyboard to enter text.
  - Some systems provide two or all three of these variations.



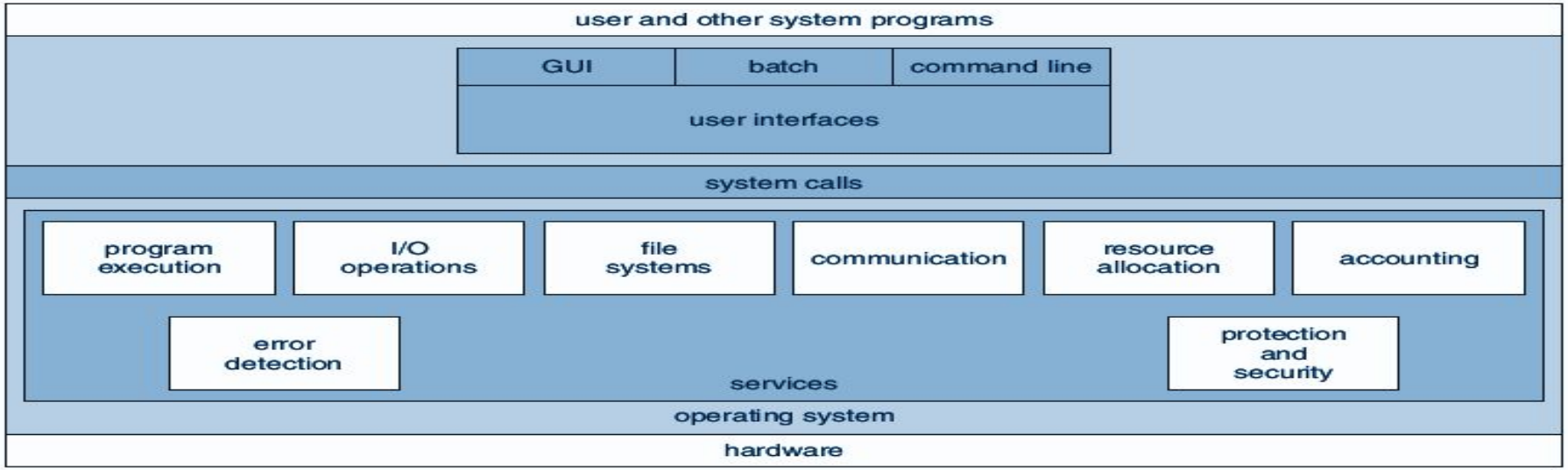
Operating System Services

- **Program execution**
  - The **system** must be able to **load** a program into memory and to **run** that program.
  - The program must be able to end its execution, either **normally** or **abnormally** (indicating error)



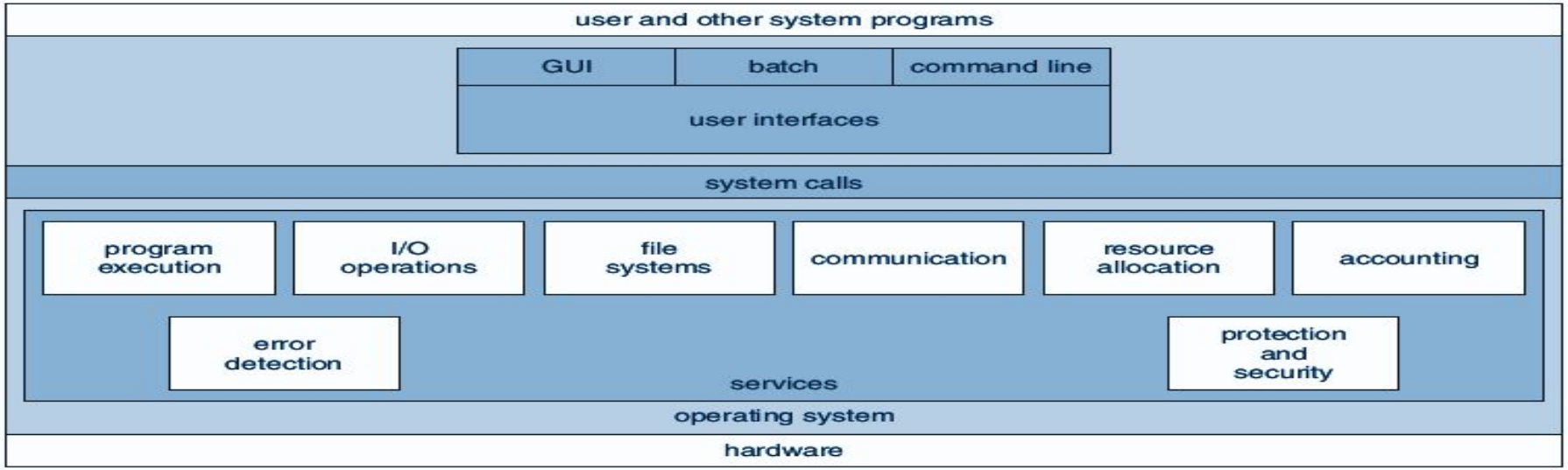
Operating System Services

- **I/O operations**
  - A running program may require I/O , which may involve a file or an I/O device. For specific devices, special functions may be desired
  - For efficiency and protection, users usually cannot control I/O devices directly. Therefore, the operating system must provide a means to do I/O .



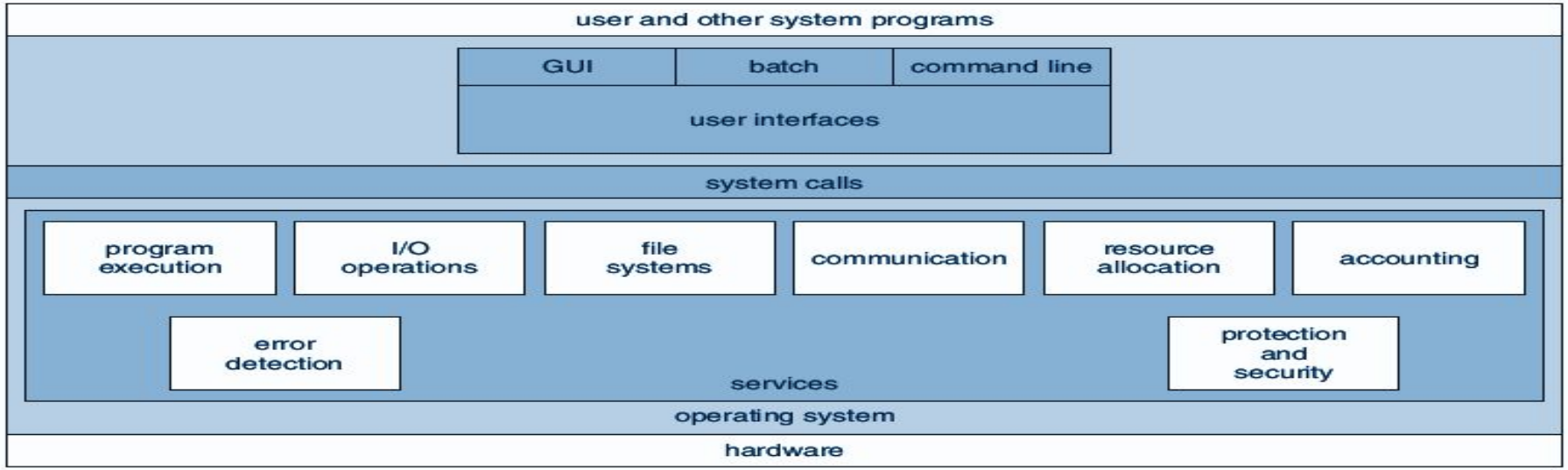
Operating System Services

- **File-system manipulation**
  - The **file system** is of particular interest. Programs need to read and write files and directories.
  - They also need to **create** and **delete** files by name, search for a given file, and list file information.
  - Some operating systems include **permissions** management to allow or deny access to files or directories based on file ownership.
  - Many operating systems provide a variety of file systems, sometimes to allow personal choice and sometimes to provide specific features or **performance** characteristics.



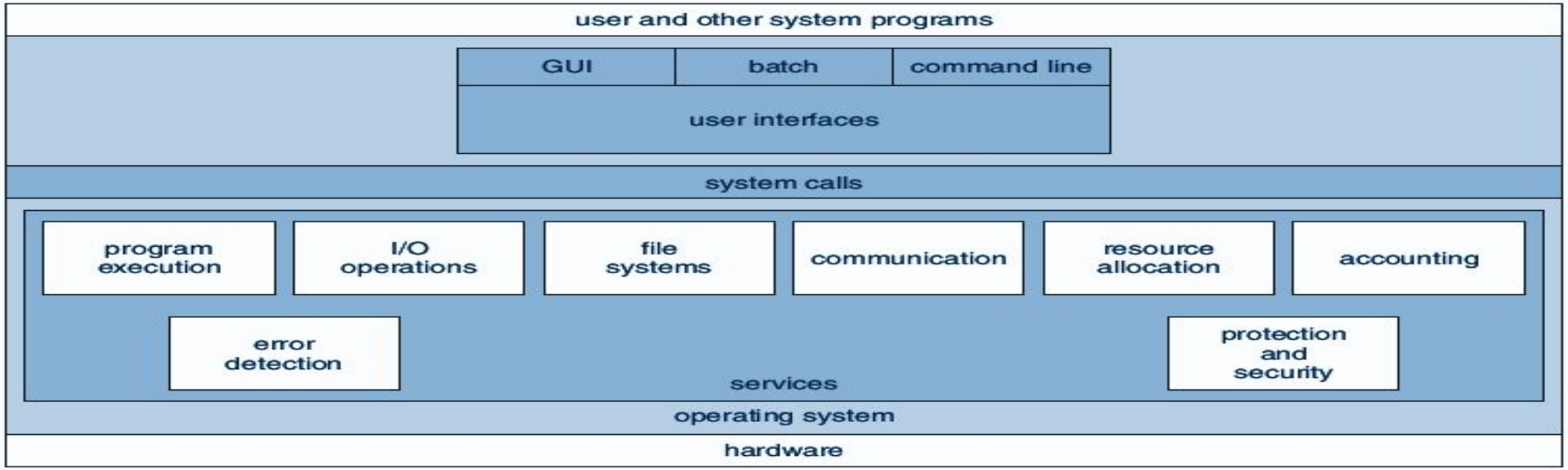
Operating System Services

- **Communications**
  - There are many circumstances in which one process needs to exchange information with another process.
  - Communication may occur between processes that are executing on the same computer or between processes that are executing on different computer systems tied together by a computer network.
  - Communications may be implemented via **shared memory**, in which two or more processes read and write to a shared section of memory, or **message passing**, in which packets of information in predefined formats are moved between processes by the operating system



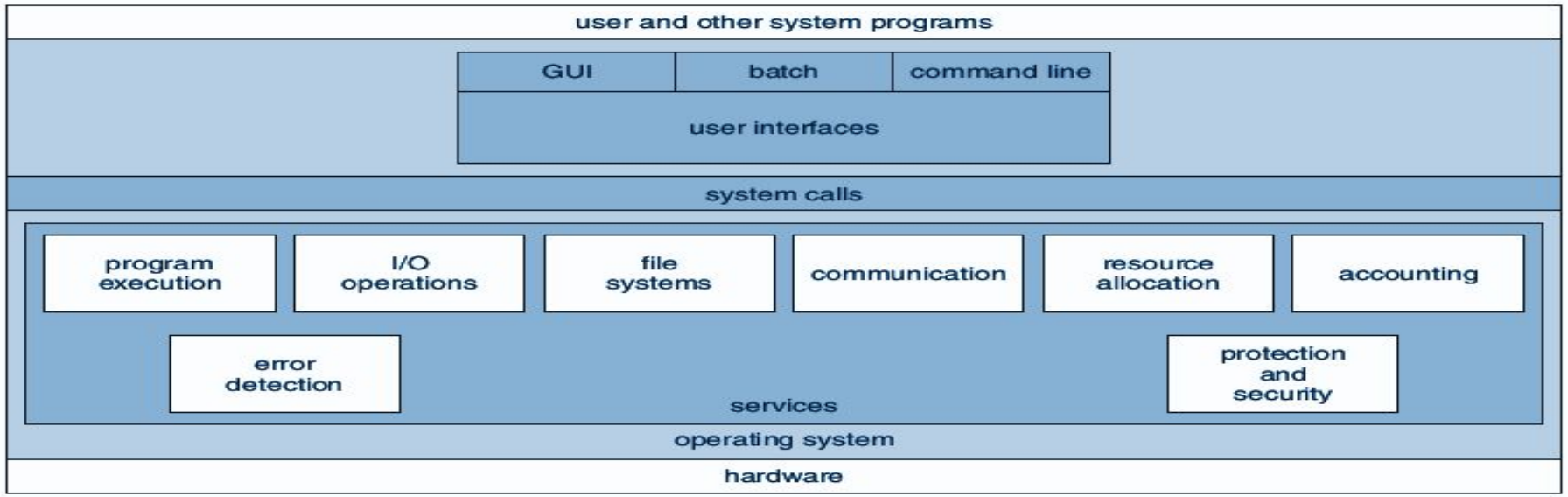
Operating System Services

- **Error detection.**
  - The operating system needs to be detecting and correcting errors constantly.
  - **Errors** may occur in the **CPU** and memory hardware (such as a memory error or a power failure), in I/O devices (such as a parity error on disk, a connection failure on a network, or lack of paper in the printer), and in the user program (such as an arithmetic overflow, an attempt to access an illegal memory location, or a too-great use of CPU time).
  - For each type of error, the operating system should take the appropriate action to ensure correct and consistent computing. Sometimes, **it has no choice but to halt the system**.
  - At other times, it might **terminate an error-causing process** or return an error code to a process for the process to detect and possibly correct.



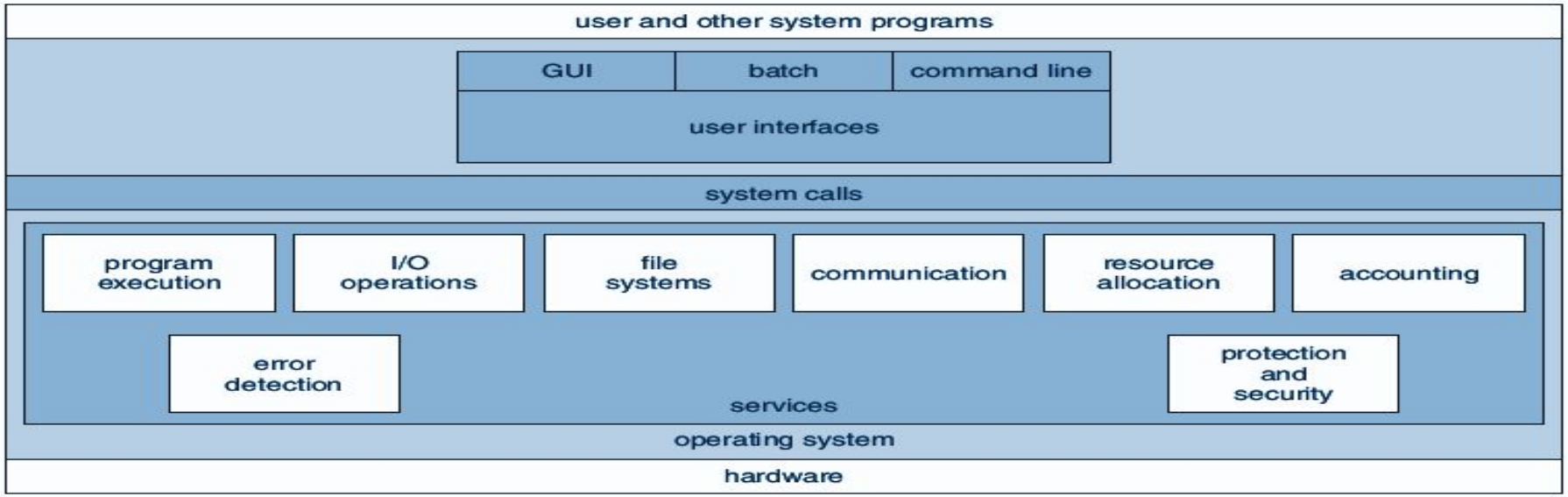
Operating System Services

- **Resource allocation.**
  - When there are multiple users or multiple jobs running at the same time, resources must be allocated to each of them.
  - The operating system manages many different types of resources.
  - Some (such as CPU cycles, main memory, and file storage) may have special allocation code, whereas others (such as I/O devices) may have much more general request and release code.
  - There may also be routines to allocate printers, USB storage drives, and other peripheral devices.



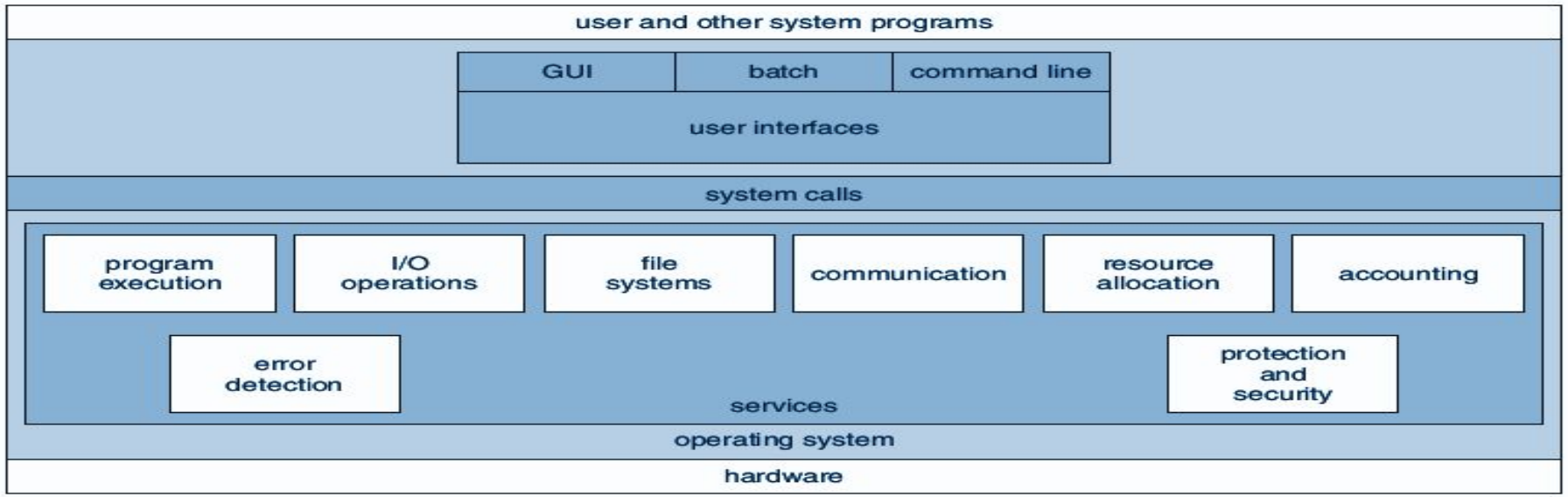
Operating System Services

- **Accounting.**
  - We want to keep track of which users use how much and what kinds of computer resources.
  - This record keeping may be used for accounting (so that users can be billed) or simply for accumulating usage statistics.
  - Usage statistics may be a valuable tool for researchers who wish to reconfigure the system to improve computing services.



Operating System Services

- **Protection and security.**
  - The owners of information stored in a multiuser or networked computer system may want to control use of that information.
  - When several separate processes execute concurrently, it should not be possible for one process to interfere with the others or with the operating system itself.
  - **Protection** involves ensuring that all access to system resources is controlled.
  - **Security** of the system from outsiders is also important.



Operating System Design and Implementation

- The first problem in designing a system is to define **goals** and **specifications**.

- At the highest level, the design of the system will be affected by the choice of hardware and the type of system: batch, time sharing, single user, multi-user, distributed, real time, or general purpose.

- The requirements can, however, be divided into two basic groups:
  - **User goals**
    - Users want certain obvious properties in a system. The system should be convenient to use, easy to learn and to use, reliable, safe, and fast.

  - **System Goals**
    - A similar set of requirements can be defined by those people who must design, create, maintain, and operate the system.
    - The system should be easy to design, implement, and maintain; and it should be flexible, reliable, error free, and efficient.

Operating System Design and Implementation - Mechanisms and Policies

- One important principle is the separation of policy from mechanism.
- **Mechanisms** determine how to do something; **policies** determine what will be done.
- A general mechanism insensitive to changes in policy would be **more desirable**. A change in policy would then require redefinition of only certain parameters of the system.

OPERATING SYSTEMS

Typical Q and As

Referred to as supervisor mode, system mode, or privileged mode. Kernel Mode: mode bit =0; User Mode: mode bit=1.	Dual Mode Operation
One of the most important mechanisms in every OS. Used to prevent a user program from getting stuck in an infinite loop or never returning control to OS.	Timer
File-system management, Mass-storage management, Cache.	Types of Storage Management
User interface, program execution, I/O operations, File system manipulation, file system manipulation, communications, error detection.	OS System Services from User Perspective

OPERATING SYSTEMS

Typical Q and As

Resource allocation, accounting, protection and security.	OS System Services from System Perspective
done though a command interpreter or graphical user interface (most common).	User Interfacing with the Operating System
also known as shells. The commands are received by the interpreter and can be carried out by the interpreter itself or the OS loading an exe.	Command Interpreter

Typical Q and As

Graphical User Interface (GUI)	A graphical user interface, or GUI, is a method by which a person communicates with a computer using graphical images, icons, and methods other than text. GUIs allow a user to use a mouse, touchpad, or another mechanism (in addition to the keyboard) to interact with the computer to issue commands.

OPERATING SYSTEMS

Typical Q and As

server operating system	Server operating systems allow for the sharing of information and applications by workstation operating systems. The operating system is optimized for background processes used for access by clients, as opposed to the foreground processes, such as the GUI.
Workstation Operating System	Workstation operating systems allow end users to access information and applications on server operating systems as well as independently run applications locally. The operating system is optimized for foreground processes, such as the GUI. Background sharing is usually limited to a specific number of users; workstation versions of Microsoft Windows are limited to 10 concurrent users.
Mobile Operating System	Mobile operating systems are found on mobile devices such as phones and tablets. The mobile operating system is generally optimized for touch-based devices where one program is used at a time.
Cloud Based Operating System	Cloud-based operating systems are a new breed of operating systems that have emerged from the mobile computing era. Cloud-based operating systems are also considered state-less operating systems, since personal data is not primarily stored on the device but in the cloud. Mobile operating systems can be considered cloud-based operating systems. So this category takes on a dual role of defining a category and describing where the data is stored. The Chrome operating system is not a mobile operating system and fits into this category.

OPERATING SYSTEMS

Typical Q and As

Service Pack	Although the term service pack is generally associated with updates, it is also used to describe a milestone in the life cycle of an operating system. When a service pack is released for an operating system, it adds major features as well as patches for both security and functionality. Microsoft has replaced this term with the introduction of Windows 8; it is now just called an update. However, many other operating systems still use "service pack."
Mainstream Support	During the mainstream support of an operating system, all hotfixes, security updates, feature updates, and general support for the operating system is supported. An operating system developer will usually have a predetermined mainstream support date when the operating system is released. When the mainstream support ends for an operating system, it is either no longer supported or enters into an extended support period.
Extended Support	The extended support period, sometimes called the long-term support (LTS) period, for an operating system is the final state of an operating system's life cycle. For example, during this time, hotfixes and features will no longer be supported, but critical security updates will be supported with Microsoft operating systems. Additional contract purchases may be required if hotfixes are required during this period. An administrator should have all systems upgraded to the current operating system by the end of the extended support date.

Topics Outline

- Operating System structure and Operations
- Kernel Data Structures
- Computing Environment
- Operating System Services
- Operating System Design
- Typical Q and As
- Operating System Implementation

THANK YOU

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

nitin.pujari@pes.edu

For Course Deliverables by the Anchor Faculty click on www.pesuacademy.com