Introduction to Operating System

Outline

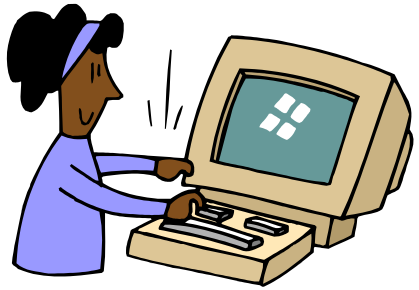
- Computer system overview
- Computer system architecture
- What is Operating System (OS)
- Roles of Operating System (OS)
- Objectives / Goals of Operating System (OS)
- Generations of Operating Systems (OS)
- Operating Systems (OS) services
- Types of Operating Systems (OS)
- System calls
- Operating Systems (OS) structure
- Multiprogramming v/s Multiprocessing v/s Multitasking
- Time Sharing Operating System
- Parallel Processing Operating System
- Distributed Operating System

Computer system overview

Basic elements of computer

Processor

I/O Modules



Memory

System Bus

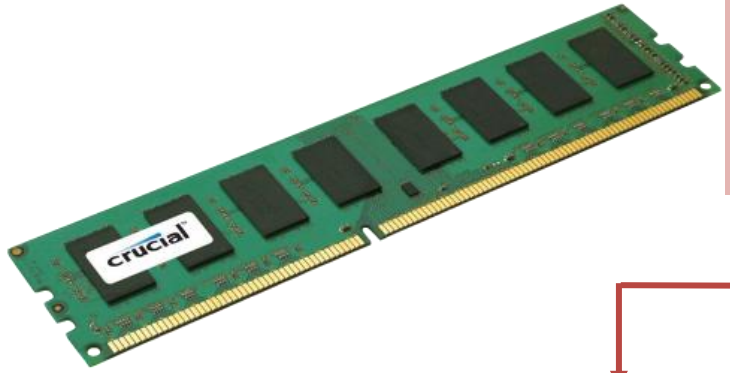
Processor

Referred to as the ***Central Processing Unit*** **(CPU)**

Arithmetic & Logic Unit	Control Unit
Performs the **data processing** functions	**Controls the operation** of the computer

Exercise Which two companies are famous for manufacturing computer processors?

Memory



Memory is device that is used to store data/information

Primary Memory	Secondary Memory
• Volatile • RAM & ROM	• Non-Volatile • HDD, CD & DVD

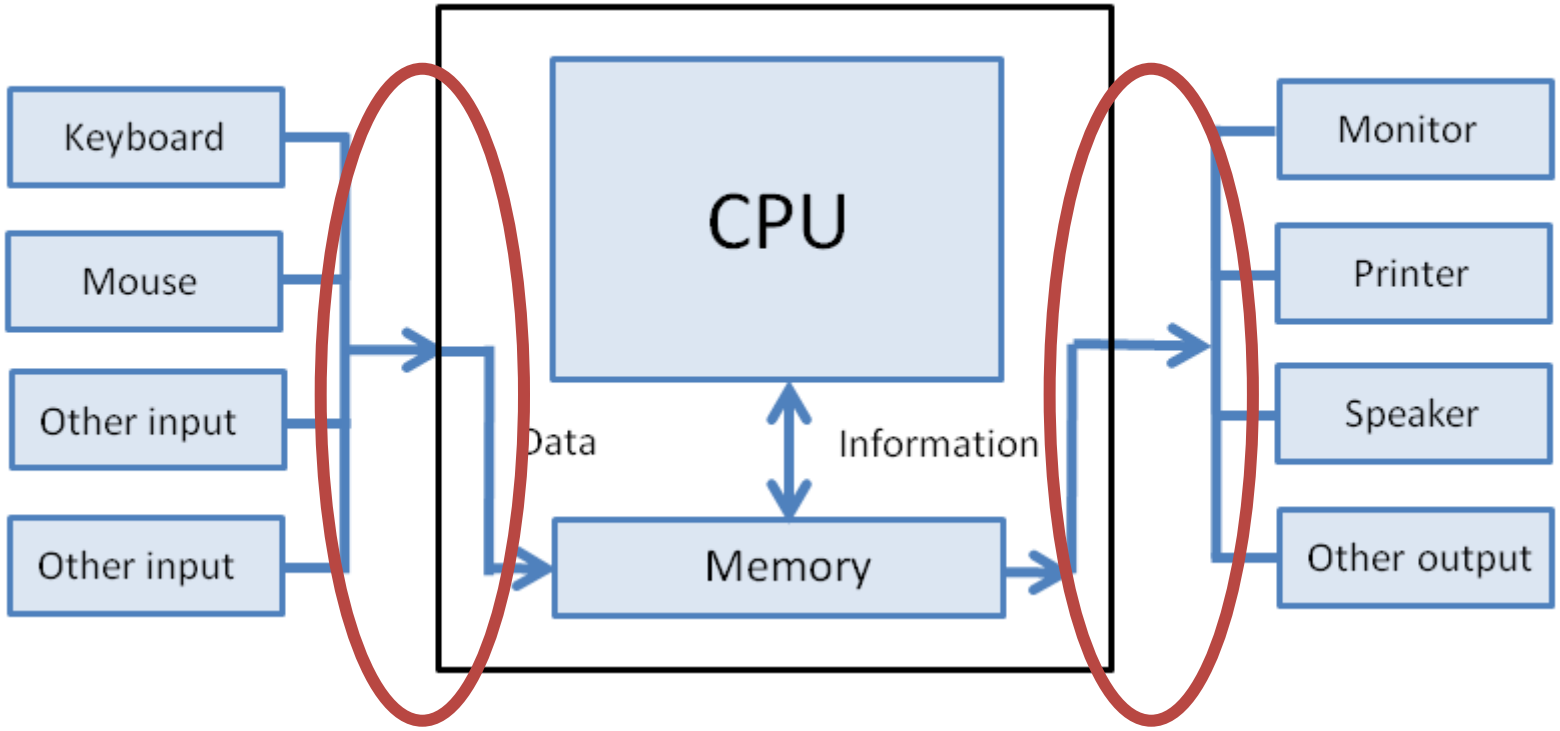
Exercise Give the difference between primary memory and secondary memory.

I/O Module (Input/Output Module)

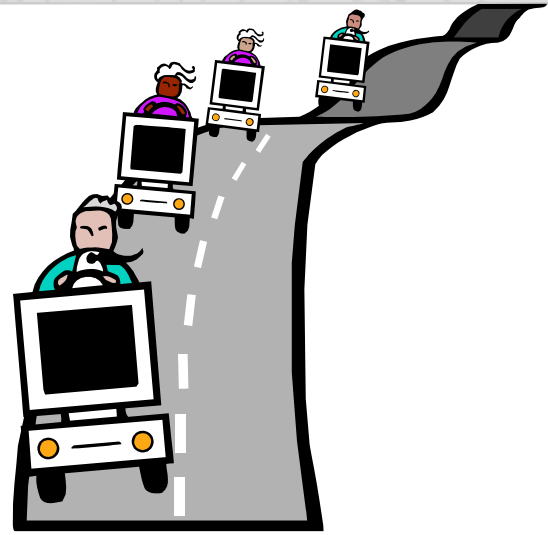
- Input/output module is a **device that acts as the connective bridge between a computer system** at one end **and an I/O or peripheral device** at the other, such as a printer, webcam or scanner.

- An I/O module is a **mediator between the processor/memory and an I/O devices**.

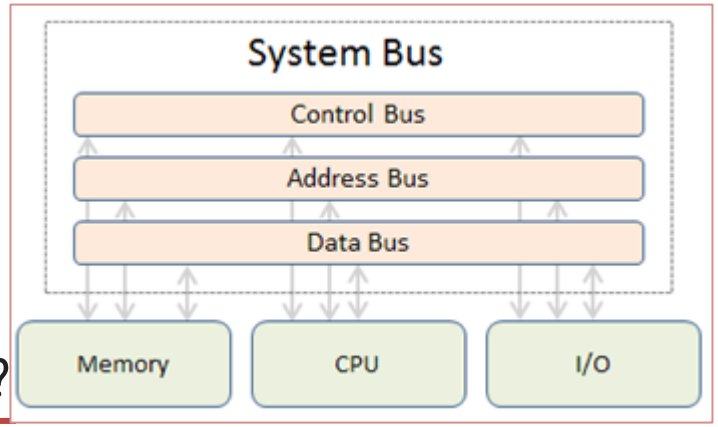
- It **controls the data exchange** between the **external devices and main memory** or **external devices and CPU registers**.



System Bus



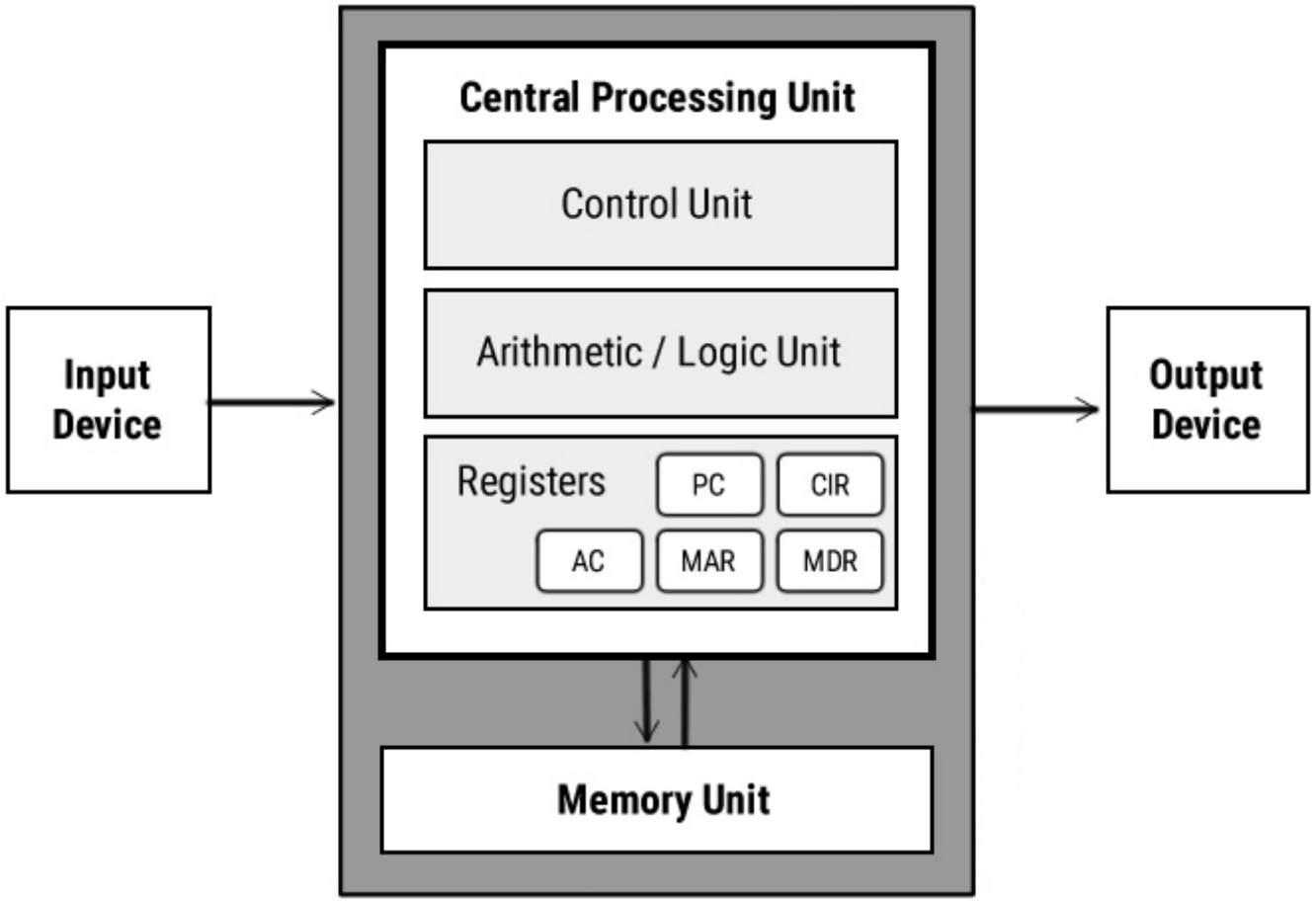
- **Provides communication among processors, main memory, and I/O devices**.
- The system bus is a pathway **composed of cables and connectors used to carry data** between a computer microprocessor and the main memory.
- Types of buses
  - Address bus - **carries memory addresses** from the processor to other components such as primary storage and input/output devices.
  - Data bus - **carries the data** between the processor and other components.
  - Control bus - **carries control signals** from the processor to other components.



Exercise What are the 3 types of buses?

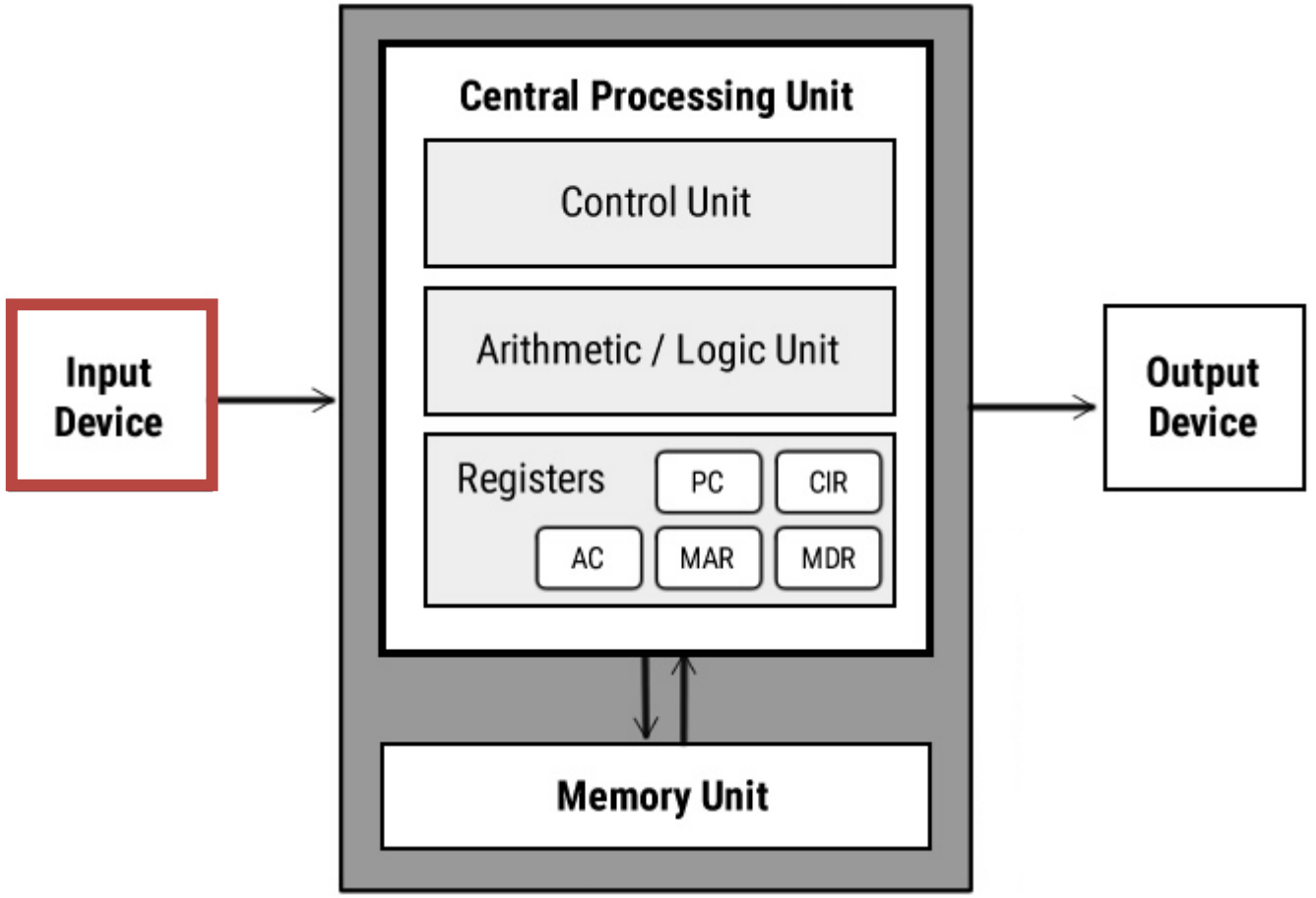
Computer system architecture

Computer system architecture



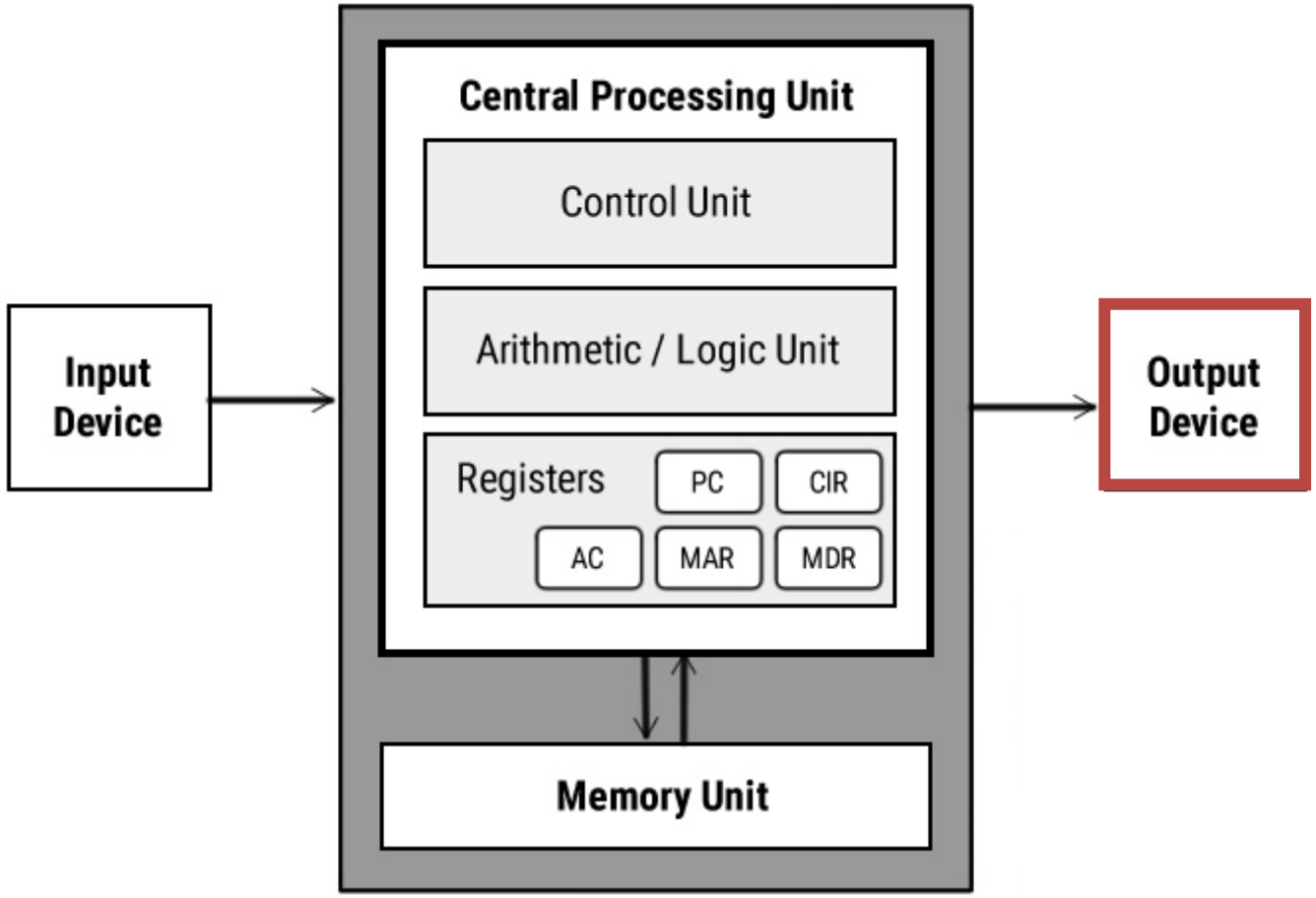
Input unit

- It **provides data** and **instructions** to the computer system.
- Commonly used input devices are keyboard, mouse, magnetic tape etc.
- Input unit performs following tasks:
  - Accept the data and instructions from the outside environment.
  - Convert it into machine language.
  - Supply the converted data to computer system.



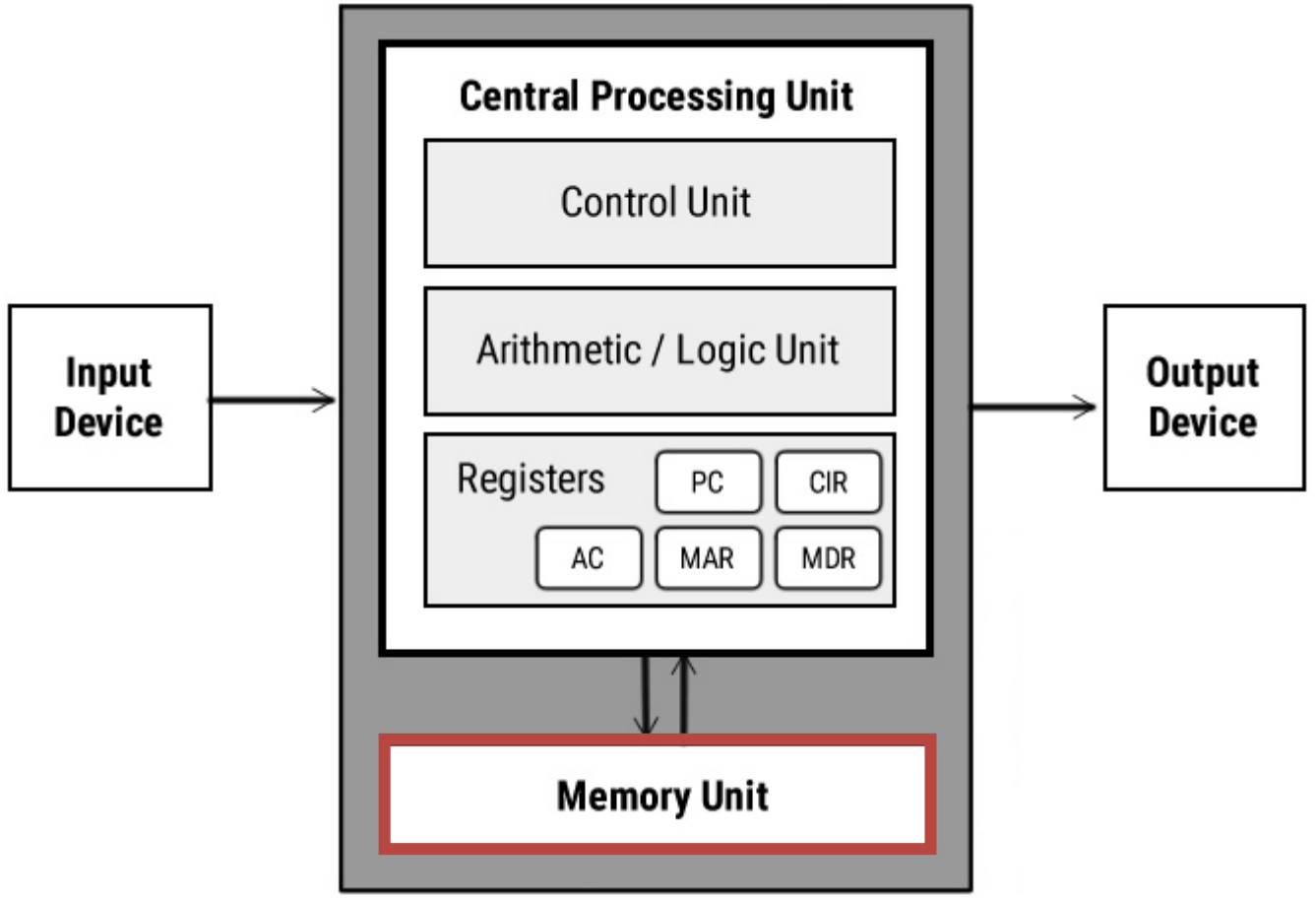
Output unit

- It **connects the internal system** of a computer **to the external environment**.
- It **provides the results** of any computation, or instructions to the outside world.
- Some output devices are printers, monitor etc.



Storage unit

- This unit **holds the data** and **instructions**.
- It also **stores the intermediate results** before these are sent to the output devices.
- It also **stores the data for later use**.
- The storage unit of a computer system can be divided into two categories:
  - Primary Storage
  - Secondary Storage

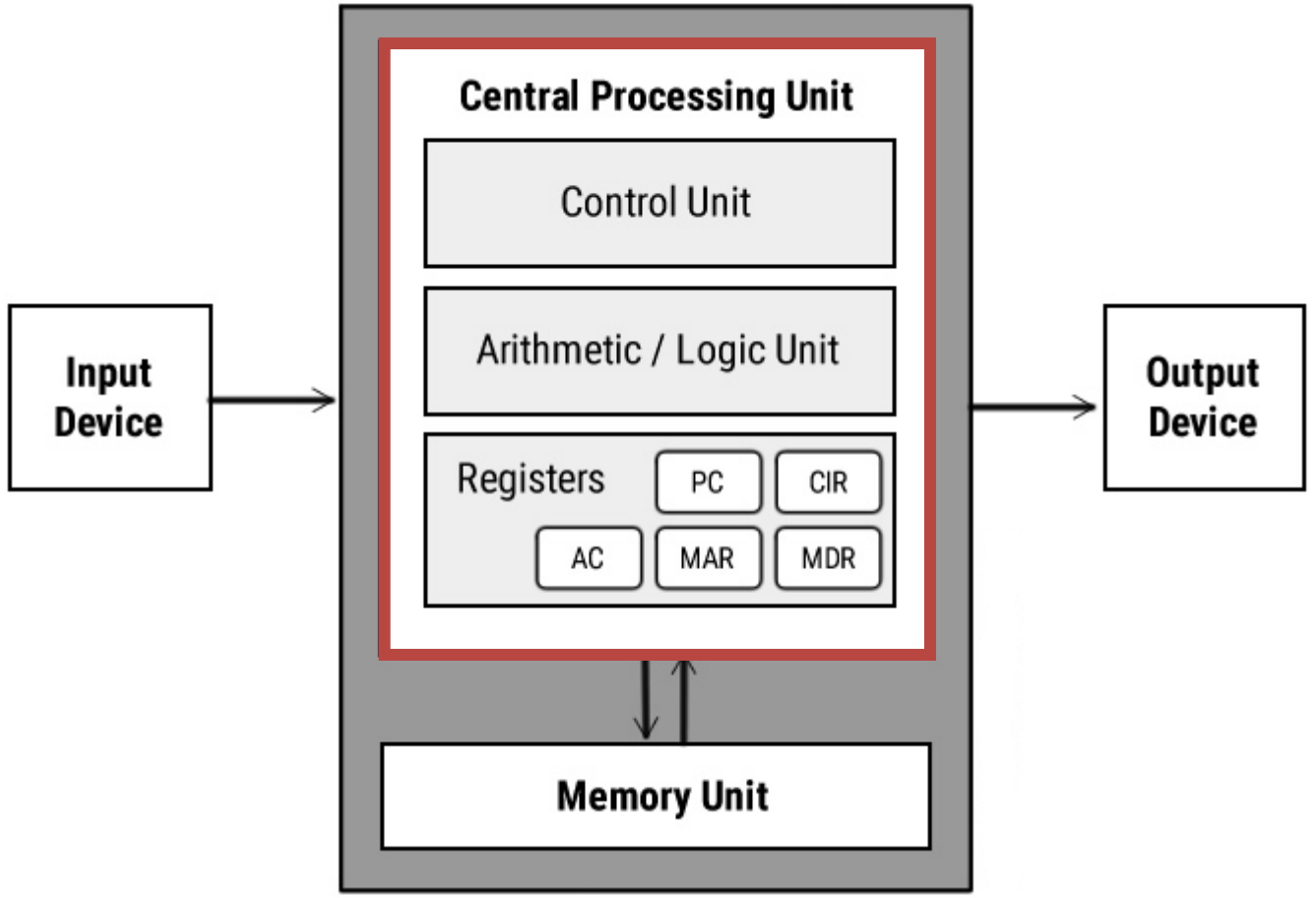


Primary storage (memory) vs Secondary storage (memory)

Primary storage	Secondary storage
Examples: RAM, ROM, Cache memory, PROM, EPROM, Registers, etc.	Examples: Hard Disk, Floppy Disk, Magnetic Tapes, etc.
It is temporary and volatile.	It is permanent and Non-volatile.
Primary memory is directly accessible by Processor/CPU.	Secondary memory is not directly accessible by the CPU.
Primary memory devices are expensive.	Secondary memory devices are cheaper.
The memory devices used for primary memory are semiconductor memories.	The secondary memory devices are magnetic and optical memories.
Primary memory is also known as Main memory or Internal memory.	Secondary memory is also known as External memory or Auxiliary memory.

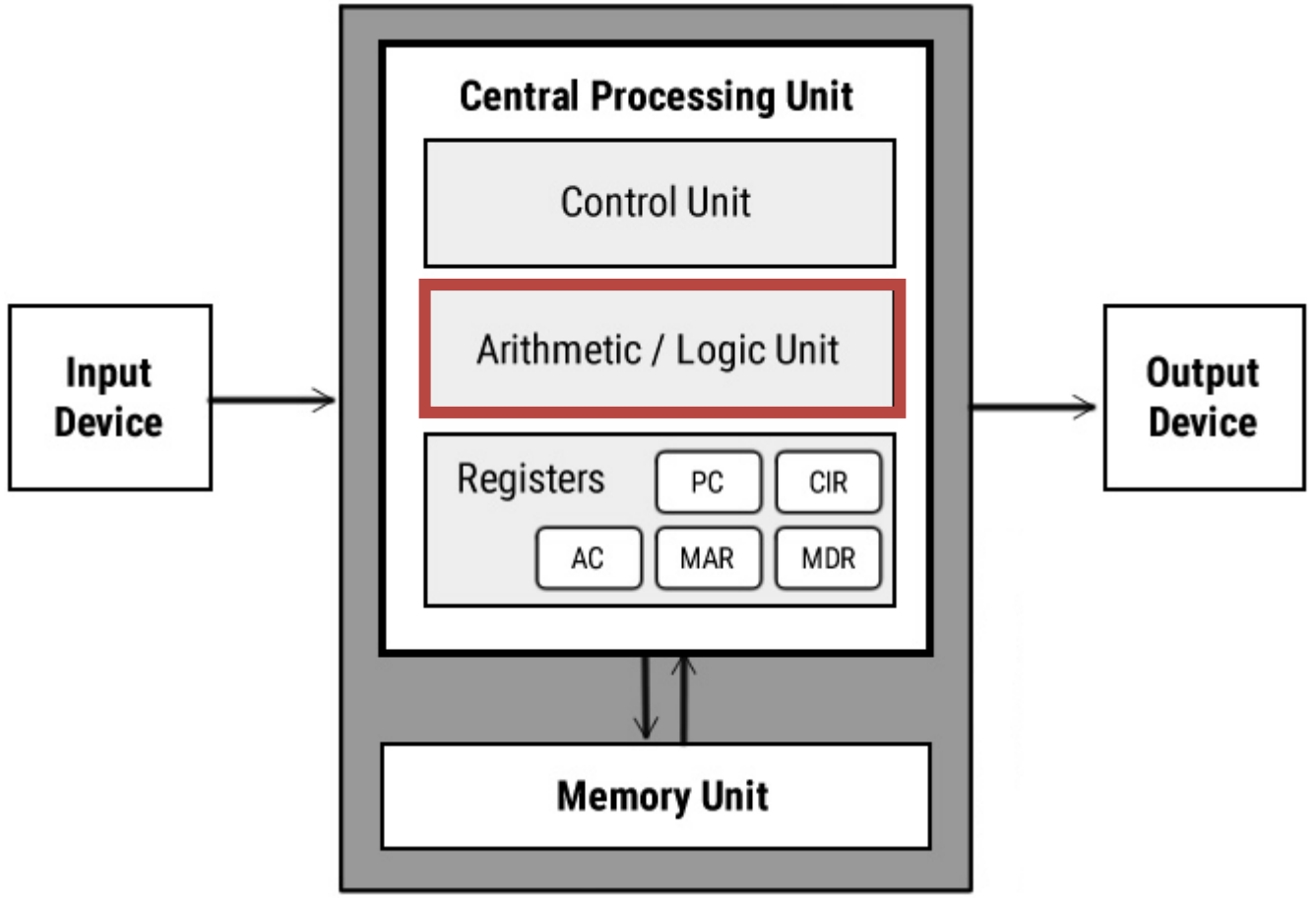
CPU (Central Processing Unit)

- The **Arithmetic Logical Unit and Control Unit are together known as CPU**.
- CPU is the **brain** of computer system.
- It performs following tasks:
  - performs all operations.
  - takes all decisions.
  - controls all the units of computer.



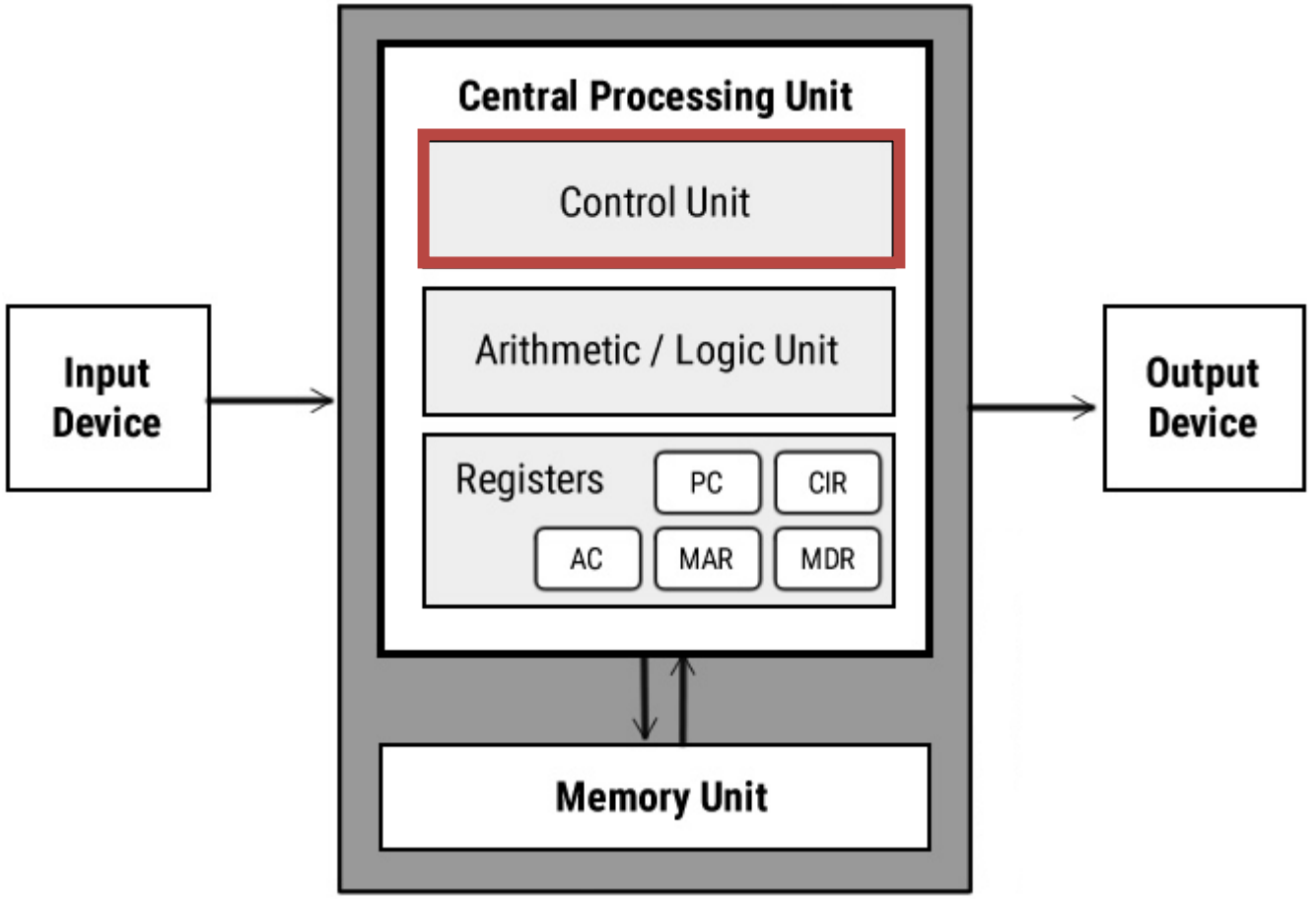
ALU (Arithmetic Logical Unit)

- All the **calculations** are performed in ALU of the computer system.
- The ALU can **perform basic operations** such as addition, subtraction, division, multiplication etc.
- Whenever calculations are required, the **control unit transfers the data from storage unit to ALU**.
- When the operations are done, the result is transferred back to the storage unit.



CU (Control Unit)

- It **controls all other units** of the computer.
- It **controls the flow of data and instructions** to and from the storage unit to ALU.
- Thus, it is also known as central nervous system of the computer.



What is Operating System (OS)

What is Operating System (OS)?

- A Computer System consists of various hardware's such as

Processor

RAM

Keyboard & Mouse

Hard Disk

Monitor

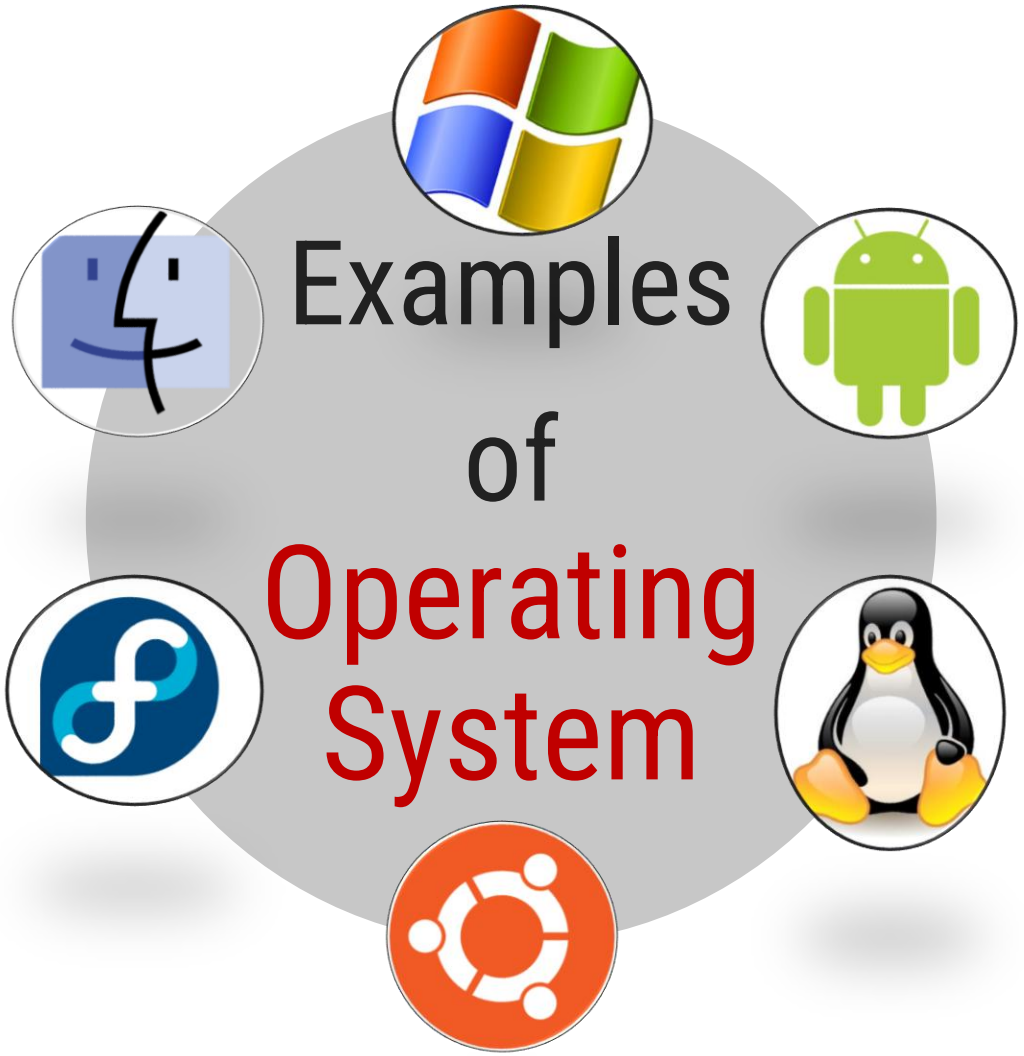
Printer

Who manages (controls) these hardwares??? **Operating System**

Definition of Operating System (OS)

- An Operating System (OS) is a **collection of software** that
  - **manages hardware resources**
  - **provides various service** to the users

Examples of Operating System (OS)

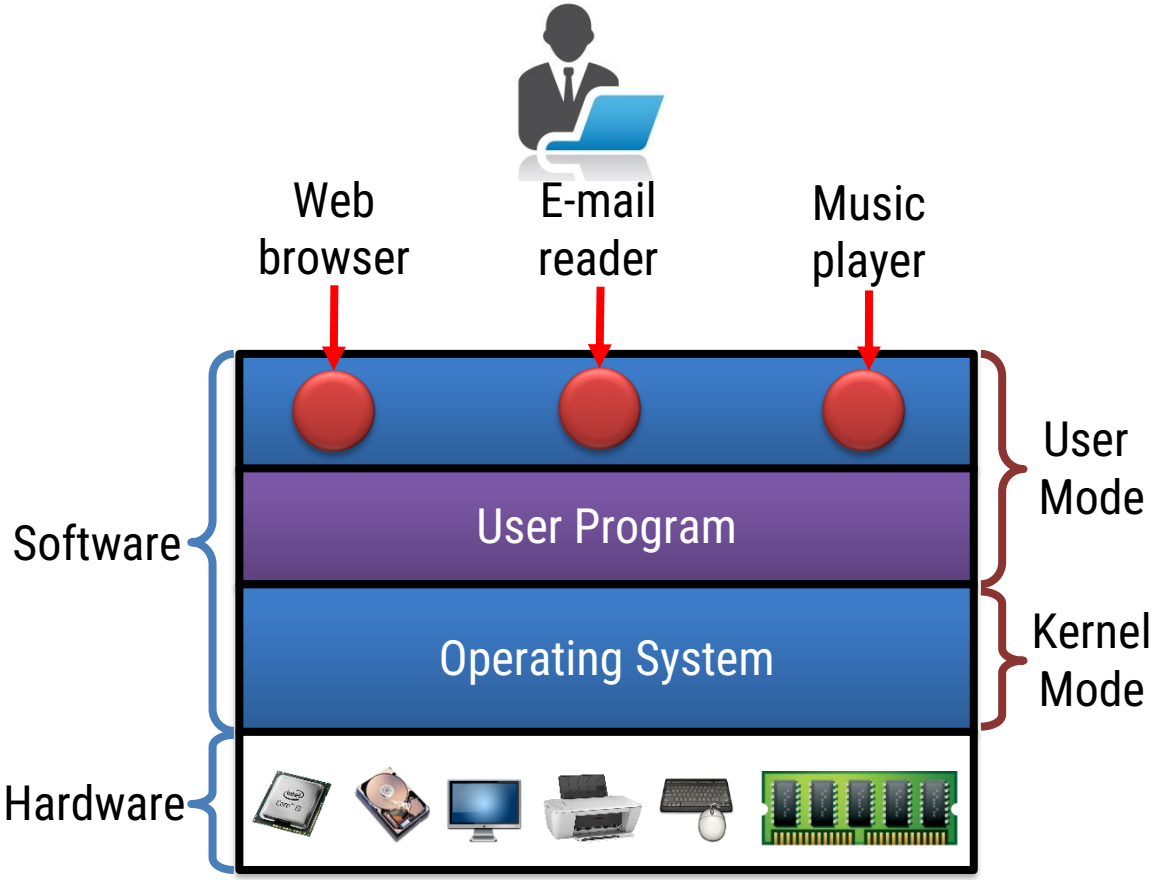


Apple Microsoft Mac OS Windows Server Linux MS DOS ubuntu

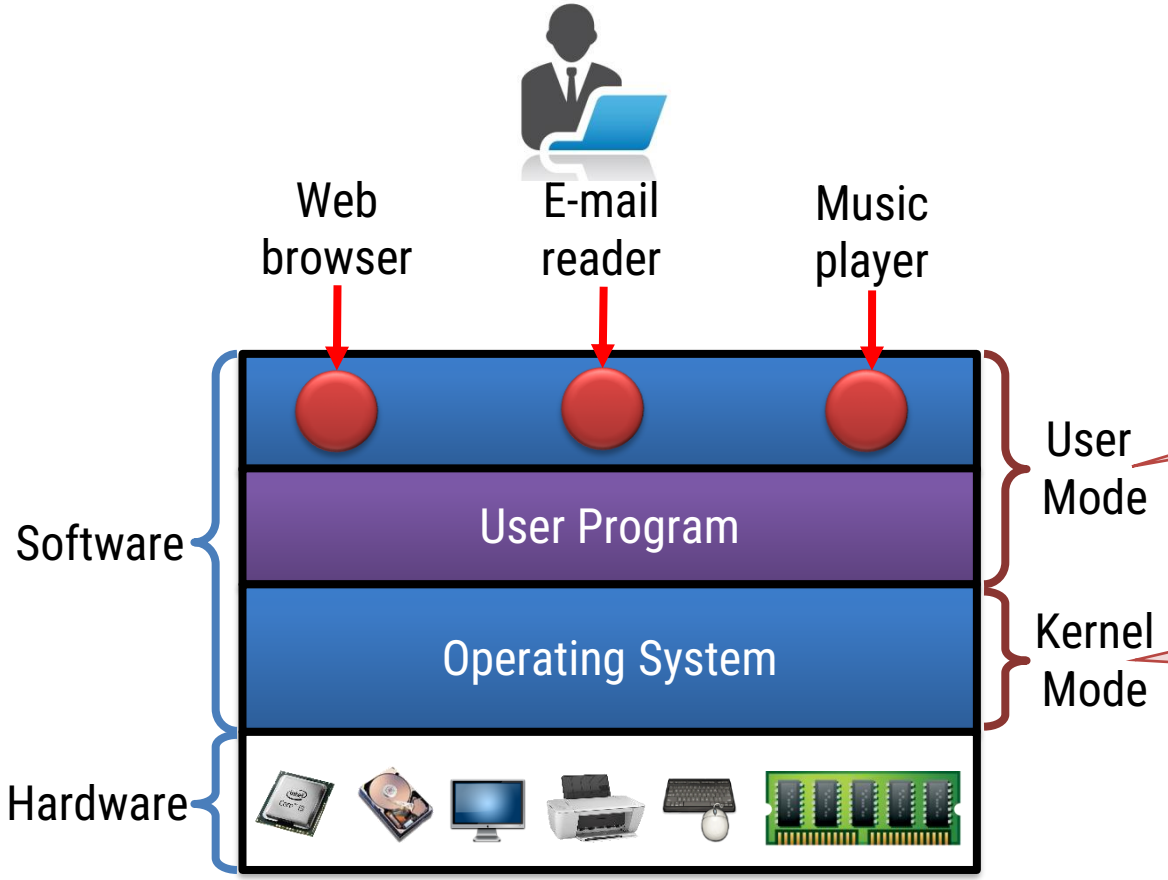
Where OS lies? (Interaction of OS & Hardware)

- OS **lies between hardware and user program**.
- It **acts as an intermediary** between the user and the hardware.
- Modes of operation of computer

1. Kernel Mode
   - has **complete access** to all the hardware
   - can **execute any instruction** that a machine is capable of executing
   - has **high privileged** (rights)
2. User Mode
   - has **limited access** to limited hardware
   - can **execute only subset (few)** of the machine instructions
   - has **less privileged** (rights)



Why and How switch occur?



- User is writing program in C.
- Once finish writing will execute it.

user-mode program performs trap instruction

- To execute this program
  1. Load program into RAM
  2. Program is executed by processor

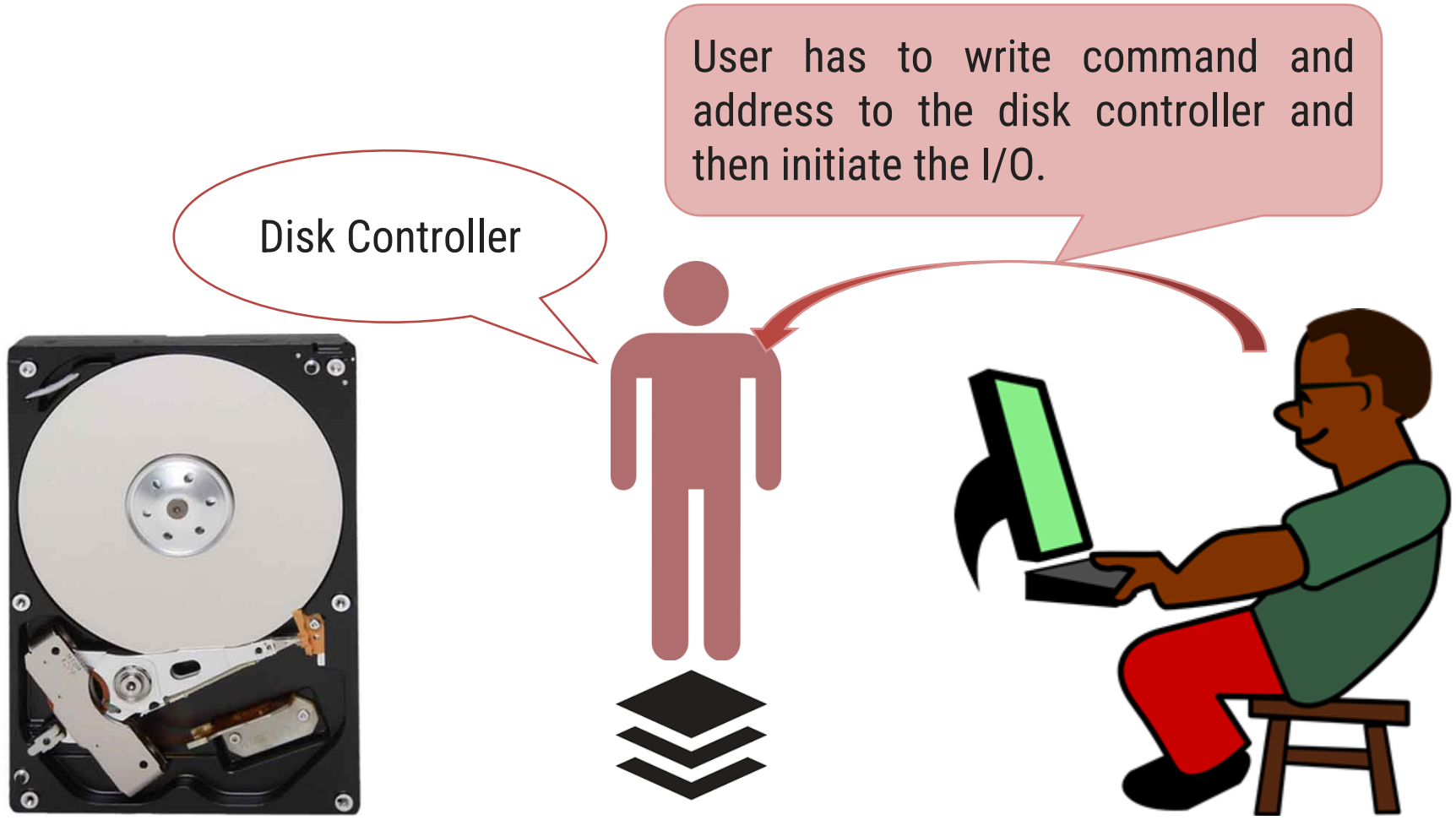
Roles of Operating System (OS)

OS as Extended Machine

- The architecture of a computer is difficult to program
  - Architecture (instruction set, memory organization, I/O, bus structure) of most of computer at the machine level language is primitive and awkward to program.
  - Example: If **user want to read from floppy or hard disk**:

OS as Extended Machine

- Example: If **user want to read from floppy or hard disk**:



OS as Extended Machine

The disk controller will find the requested data in the disk and fetch it from disk to disk controller buffer.

User has to check the status of disk controller operation where it has finished or not.

OS as Extended Machine

If success, the data from disk controller buffer should be moved to main memory (to the application buffer).

OS as Extended Machine

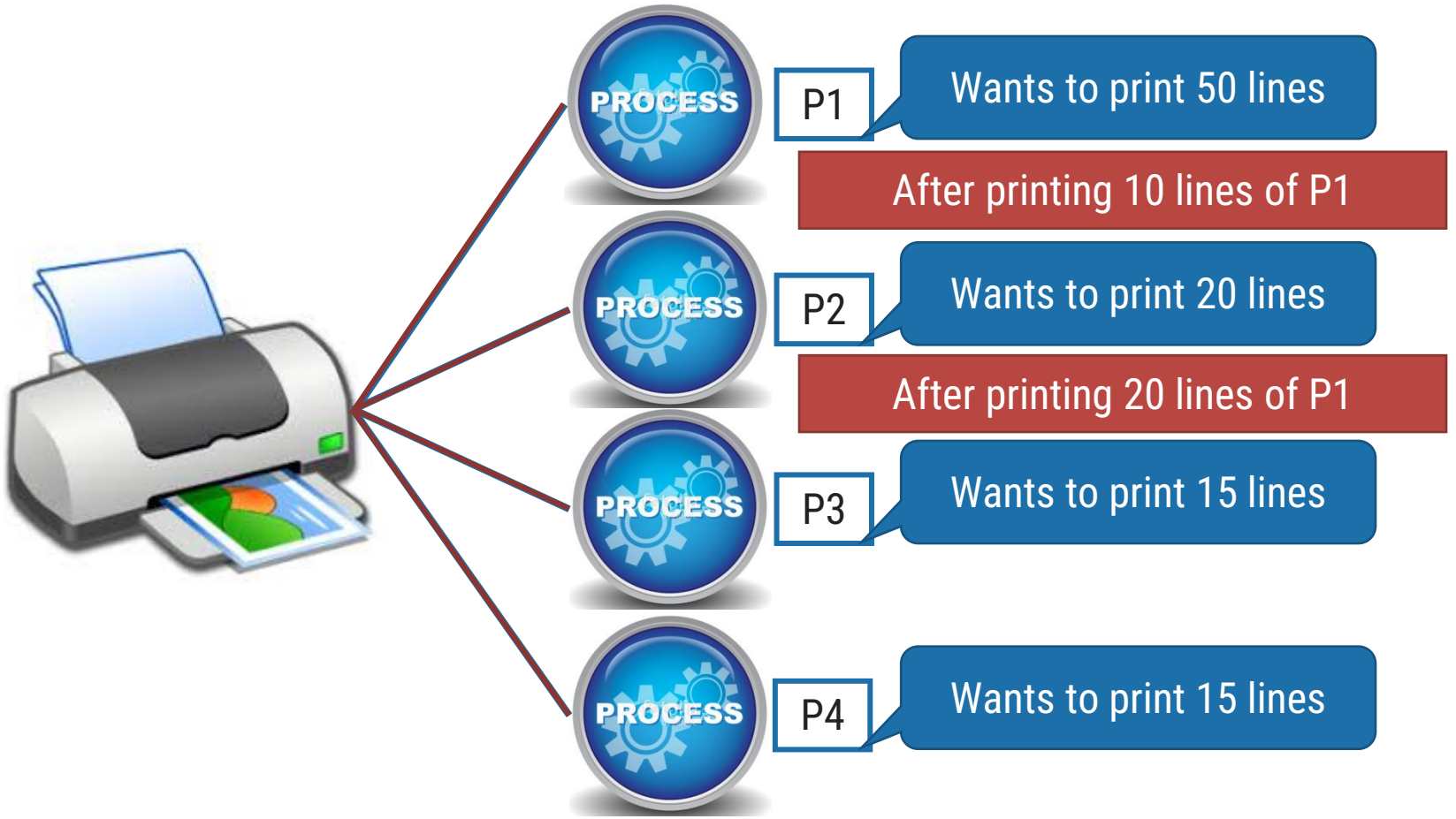
- If all the users will have to do these messy details:
  - The **program will be very difficult to write and quite long**.
  - The **program will be hardware dependent**.
- User **don't want to be involved in programming** of storage devices.
- Therefore, an **OS provides a set of basic commands** or instructions to perform various operations such as read, write, modify, save or close.
- Dealing with these command is easier than directly dealing with hardware.
- Operating system **hides the complexity of hardware** and present a **beautiful interface** to the users.

OS as Resource Manager

- There are lots of resources in computer system
  - CPU (Processor)
  - Memory
  - I/O devices such as hard disk, mouse, keyboard, printer, scanner etc.
- If a computer system is used by **multiple applications (or users)**, then they will **compete for these resources**.

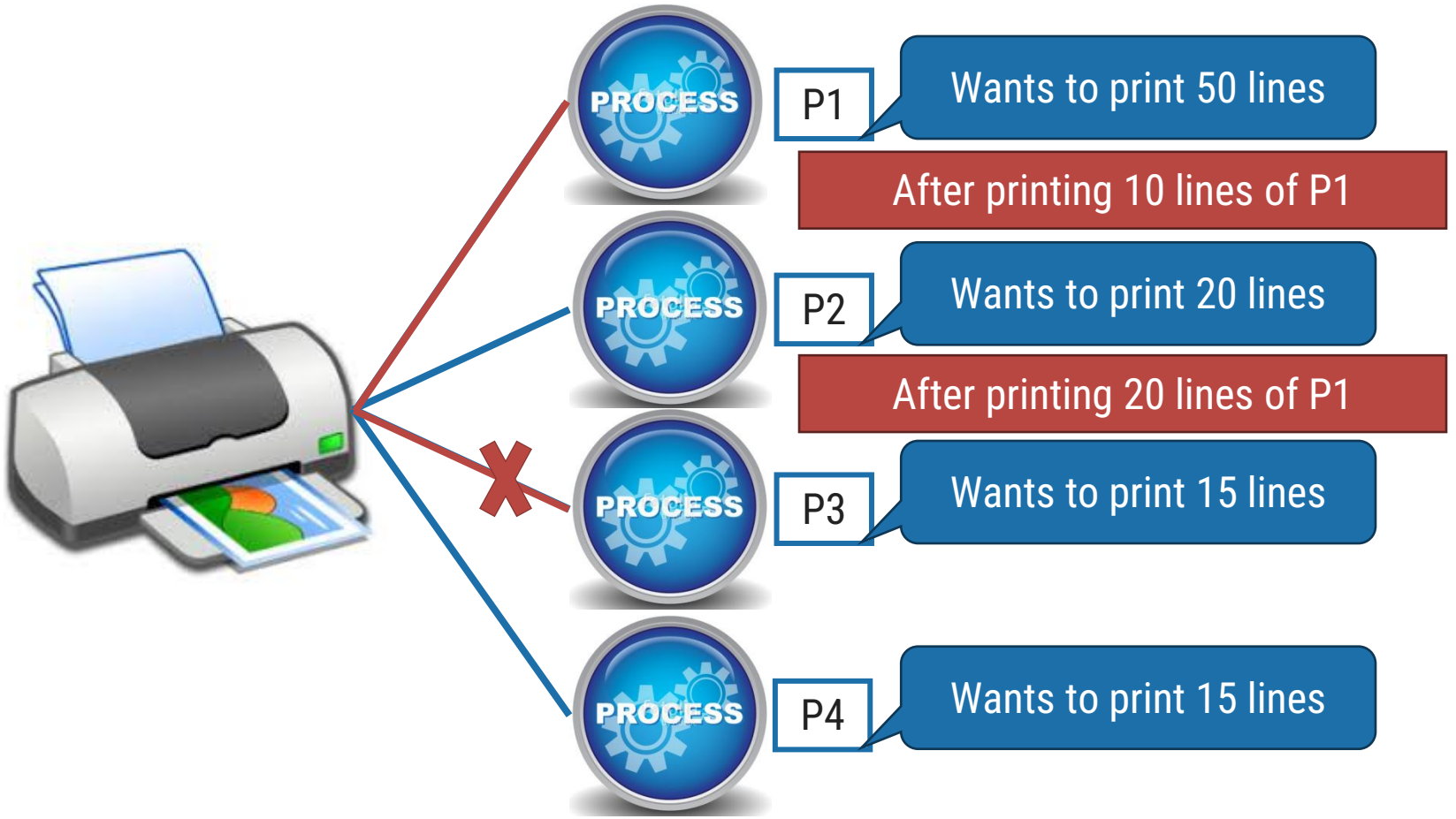
OS as Resource Manager

- It is the job of OS to allocate these resources to the various applications so that:
  - The **resources are allocated fairly** (equally)



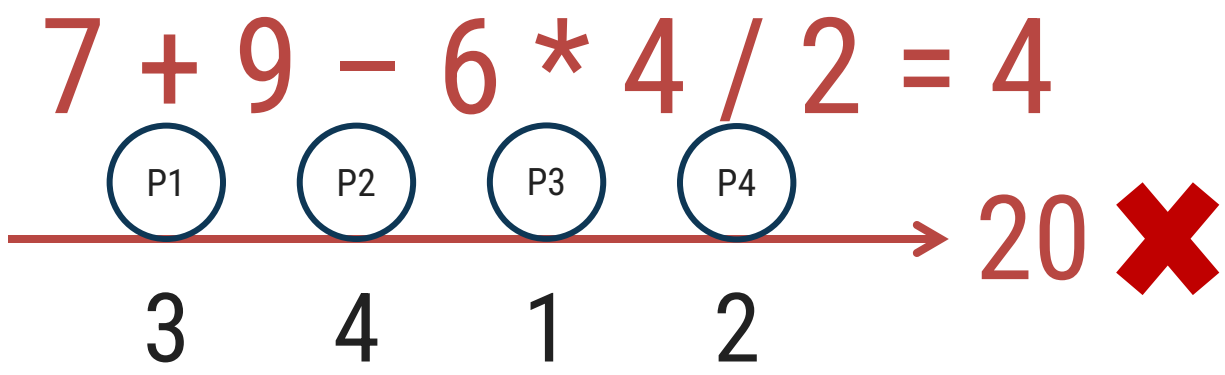
OS as Resource Manager

- It is the job of OS to allocate these resources to the various applications so that:
  - The **resources are protected from cross-access**.



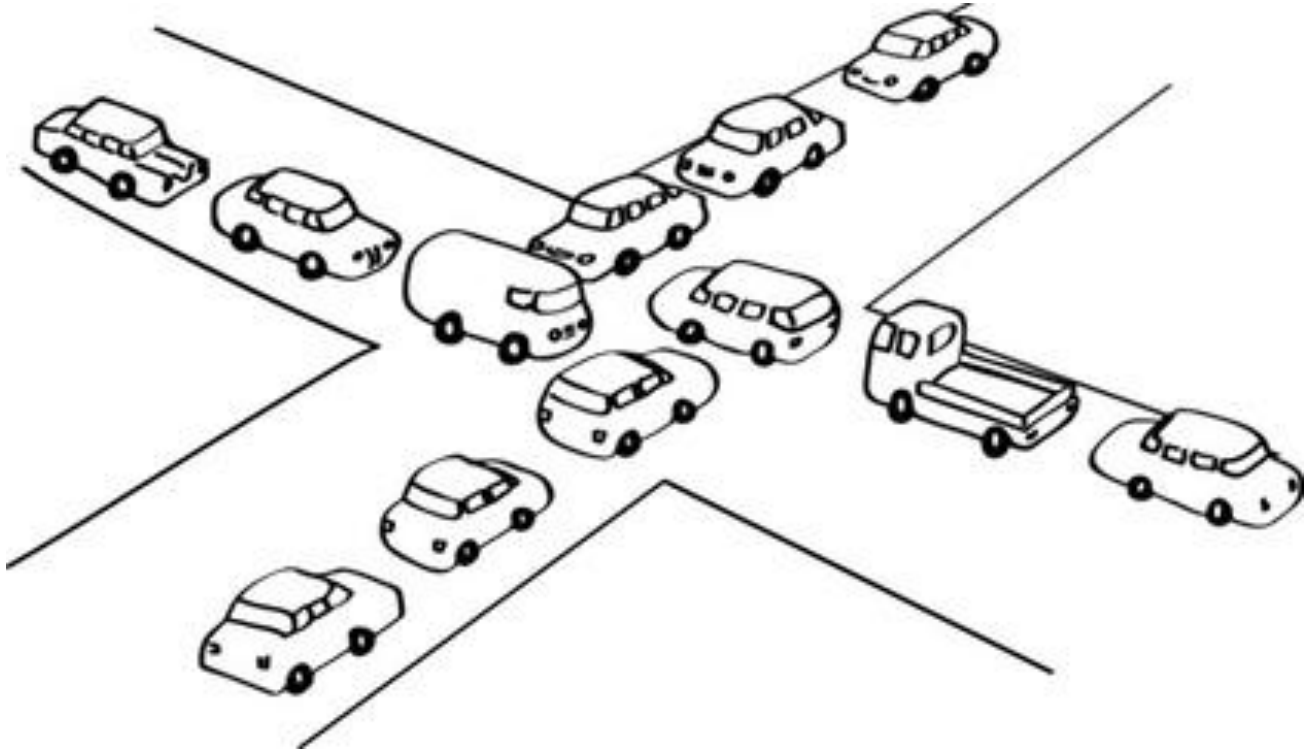
OS as Resource Manager

- It is the job of OS to allocate these resources to the various applications so that:
  - **Access to the resources is synchronized** so that operations are correct and consistent
  - Example: If we write a program to calculate below in C language



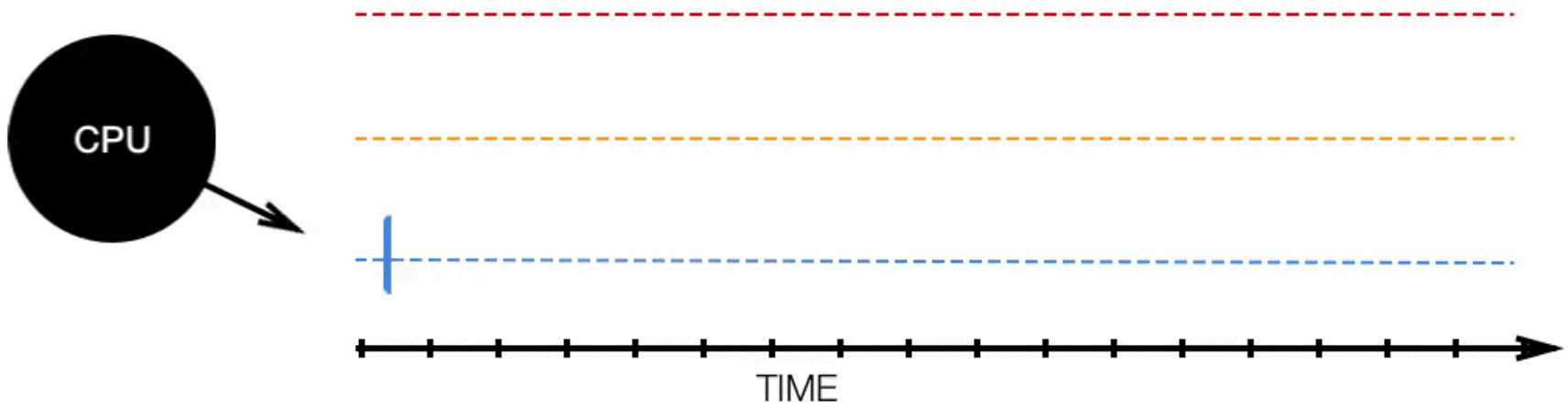
OS as Resource Manager

- It is the job of OS to **proper allocate these resources to the various applications** so that:
  - Deadlock are detected, resolved and avoided.



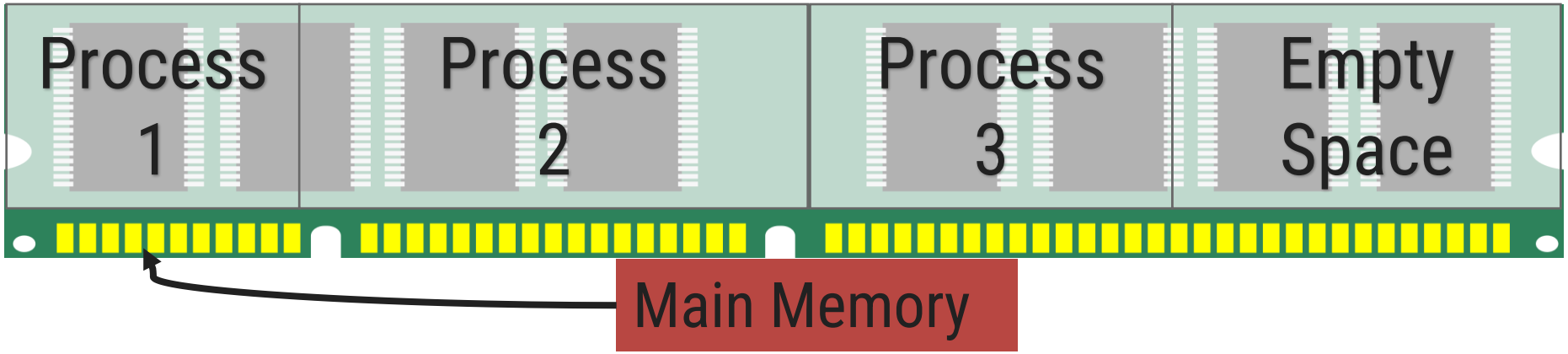
OS as Resource Manager

- Resource manager – sharing resources in two different ways:
  1. In time sharing/multiplexing (i.e **CPU**)



OS as Resource Manager

- Resource manager – sharing resources in two different ways
  2. In space sharing/multiplexing. (i.e Memory)



Objectives / Goals of Operating System (OS)

Objectives / Goals of Operating System (OS)

- Make the computer system **convenient to use in an efficient manner**.
- **Hide the details of the hardware** resources from the users.
- **Provide** users a **convenient interface** to use the computer system.
- **Act as an intermediary** between the **hardware and its users**, making it easier for the users to access and use other resources.

- **Manage the resources** of a computer system.
- **Keep track** of who is using which resource, granting resource requests, and mediating conflicting requests from different programs and users.
- **Provide efficient** and **fair sharing of resources** among users and programs.

Note: Overall duties of the OS are to user to handle all resources and improve efficiency/utilization of resources.

Generations of Operating Systems (OS)

Generations / History of OS (First generation)

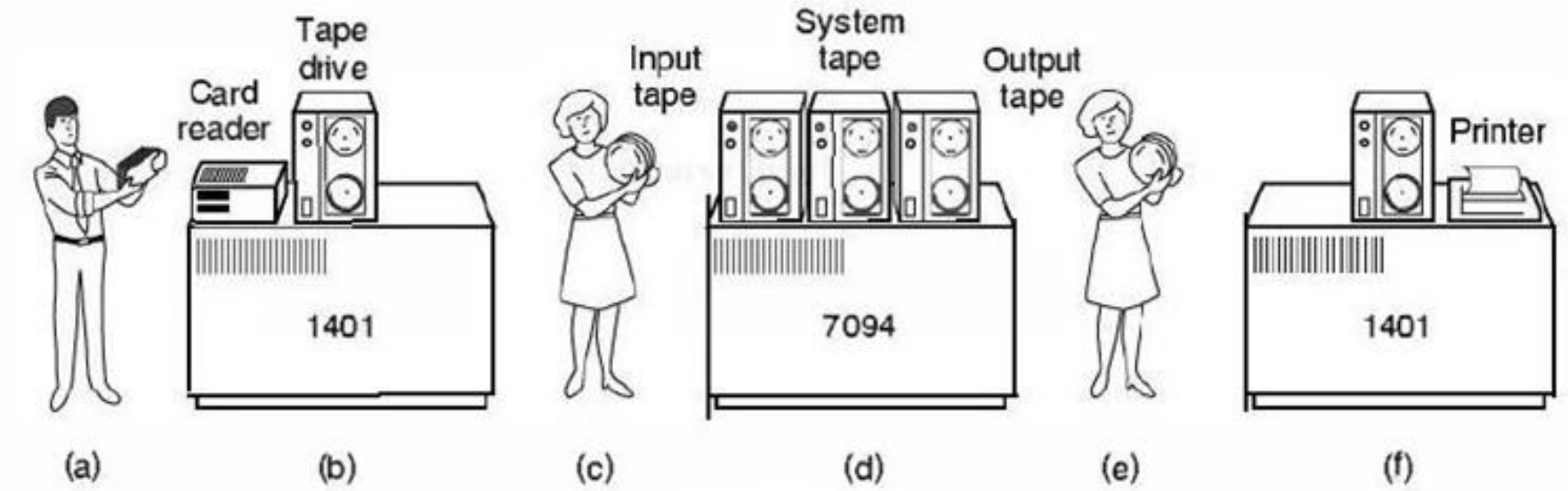
- First generation (1945-1955)
  - ↪ Vacuum tubes and plug-boards are used in these systems.

Vacuum tubes

Plug board

Generations / History of OS (Second generation)

- Second generation (1955-1965)
  - Transistors are used in these systems
  - The machine that are produced are called mainframes.
  - Batch systems was used for processing.



Generations / History of OS (Third generation)

- Third generation (1965-1980)
  - Integrated circuits (IC's) are used in place of transistors in these computers.
  - It provides multiprogramming (the ability to have several programs in memory at once, each in its own memory partition).

Generations / History of OS (Forth generation)

- Fourth generation (1980-present)
  - Personal Computers (PC)
  - LSI (Large Scale Integration) circuits, chips containing thousands of transistors are used in these systems.

Operating Systems (OS) services

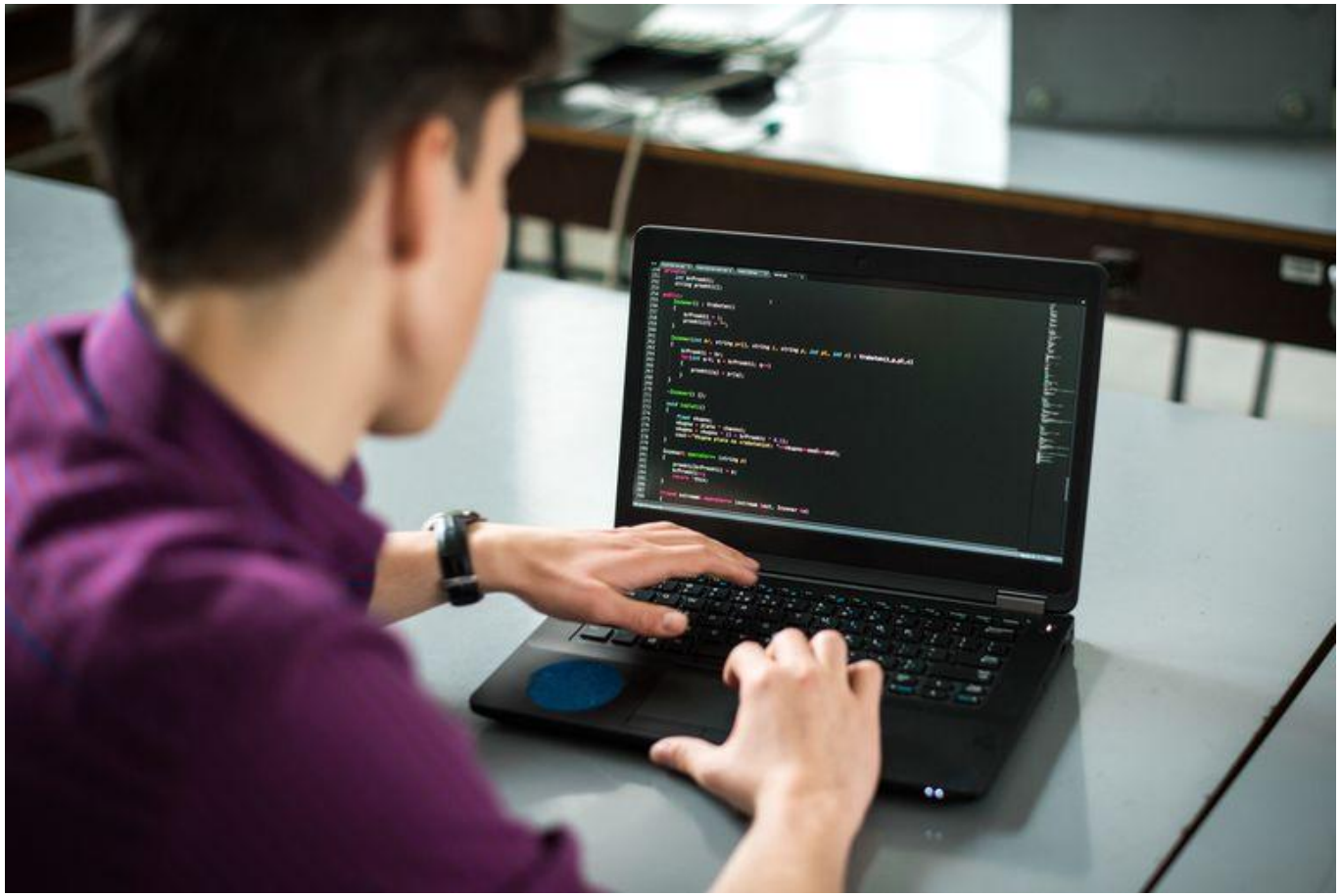
Section - 7

Definition of Operating System

- An Operating System (OS) is a collection of software that
  - manages hardware resources
  - provides various service to the users

Services / Functions / Tasks of Operating System (OS)

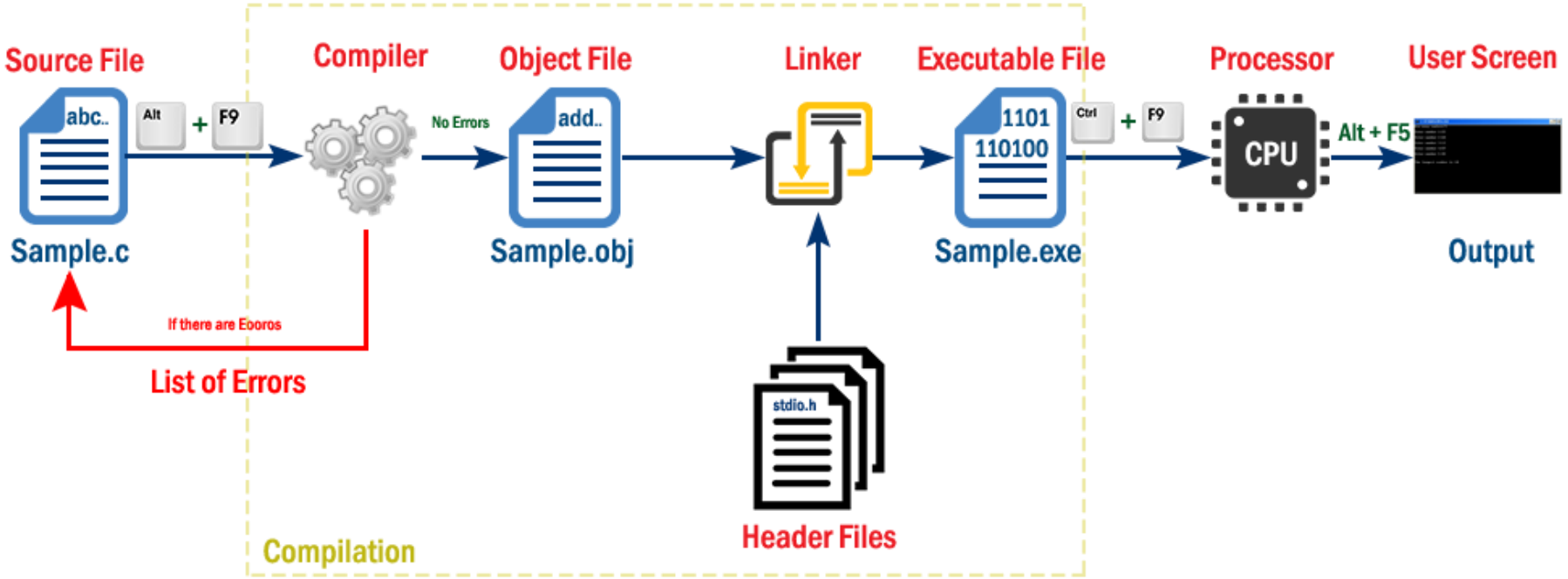
1. Program development
   - ➥ It provides editors and debuggers to assist (help) the programmer in creating programs.



Exercise Give the name of any five code editor for windows?

Services / Functions / Tasks of Operating System (OS)

2. Program execution
   - Following tasks need to be perform to execute a program:
     - Instructions and data must be loaded into main memory.
     - I/O devices and files must be initialized.
   - The OS handles all these duties for the user.

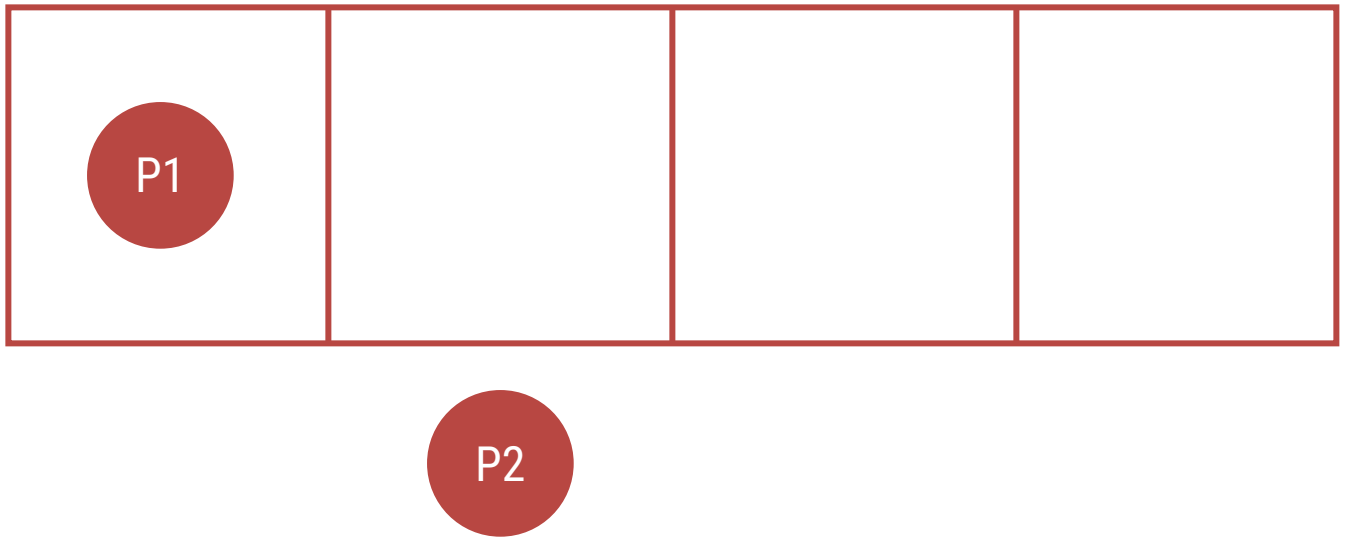


Services / Functions / Tasks of Operating System (OS)

3. Access to I/O devices (Resource allocation)
   - A running program may require I/O, which may involve file or an I/O device.
   - For efficiency and protection, users cannot control I/O devices directly.
   - Therefore, the OS controls these I/O devices and provides to program as per requirement.

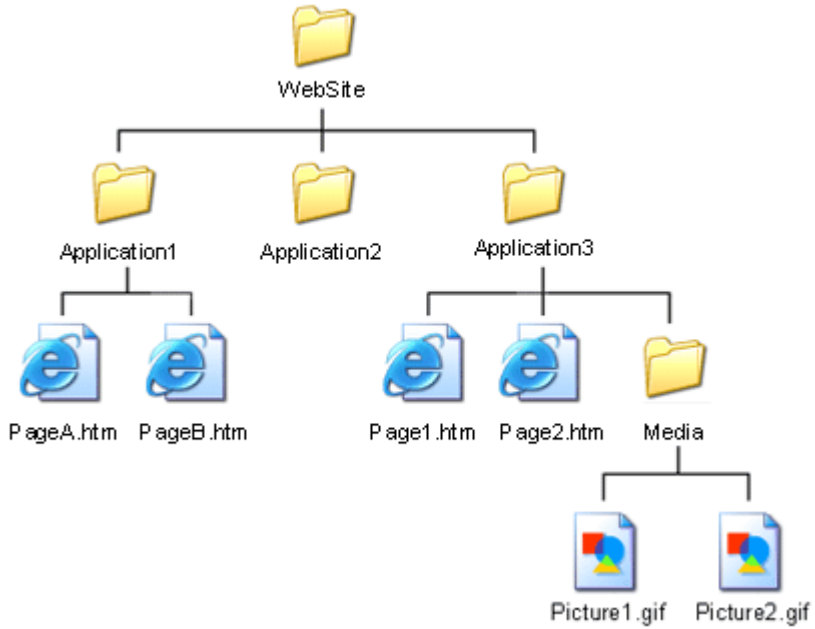
Services / Functions / Tasks of Operating System (OS)

4. Memory management
   - OS manages memory hierarchy.
   - OS keeps the track of which part of memory area in use and free memory.
   - It allocates memory to program when they need it.
   - It de-allocate the memory when the program finish execution.



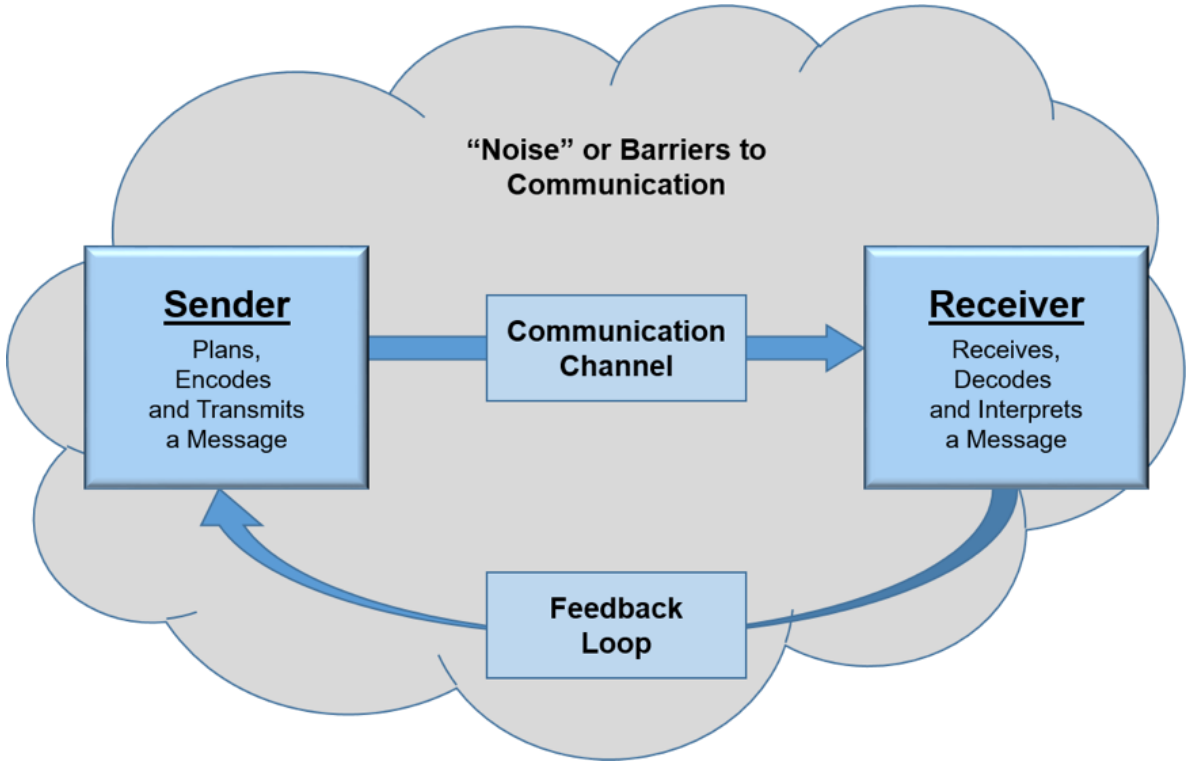
Services / Functions / Tasks of Operating System (OS)

5. Controlled access to file
   - In case of file access, OS provides a directory hierarchy for easy access and management of file.
   - OS provides various file handling commands using which user can easily read, write and modify file.



Services / Functions / Tasks of Operating System (OS)

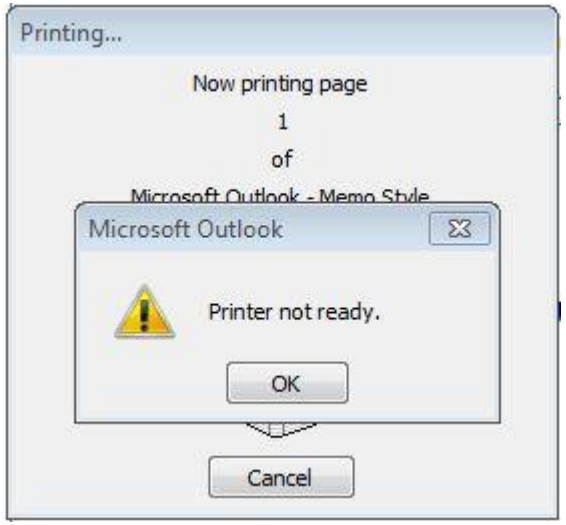
6. Communication
   - In multitasking environment, the processes need to communicate with each other and to exchange their information.
   - Operating system performs the communication among various types of processes in the form of shared memory.



Services / Functions / Tasks of Operating System (OS)

7. Error detection and response
   - An error may occur in CPU, in I/O devices or in the memory hardware.

   - Following are the major activities of an operating system with respect to error handling –
     - The OS constantly checks for possible errors.
     - The OS takes an appropriate action to ensure correct and consistent computing.



Services / Functions / Tasks of Operating System (OS)

8. Accounting
   - Keeping a track of which users are using how much and what kinds of computer resources can be used for accounting or simply for accumulating usage statistics.
   - Usage statistics is used to reconfigure the system to improve computing services.

Services / Functions / Tasks of Operating System (OS)

9. Protection & Security
   - Protection involves ensuring that all accesses to system resources is controlled.
   - To make a system secure, the user needs to authenticate himself or herself to the system.

Types of Operating Systems (OS)

Types of Operating Systems (OS)

1. Mainframe operating systems

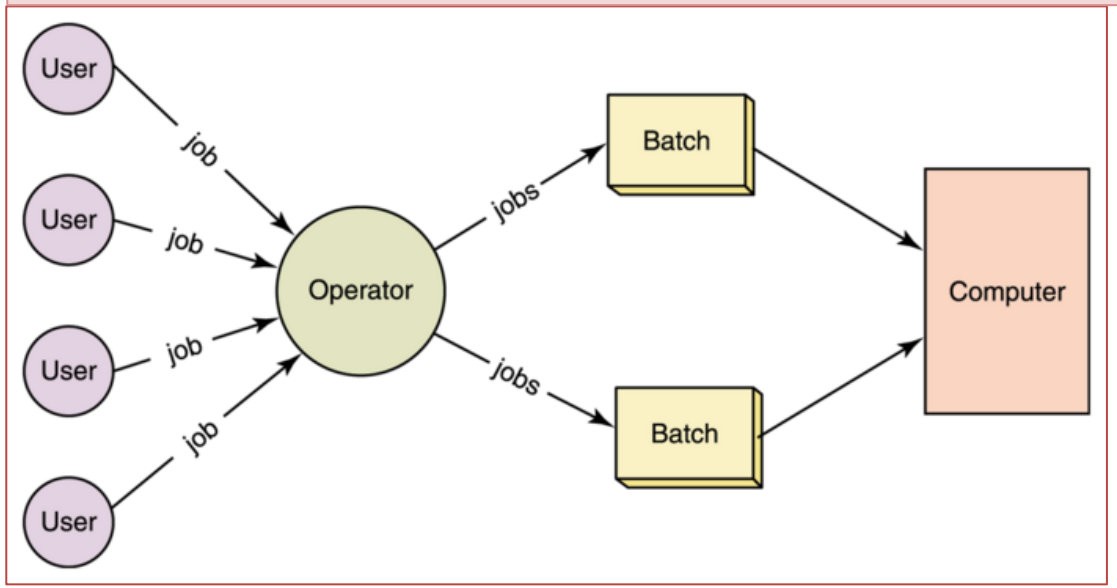
   - OS found in room sized computers which are still found in major corporate data centers.

   - They offer three kinds of services:
     1. Batch OS
     2. Transaction processing
     3. Timesharing

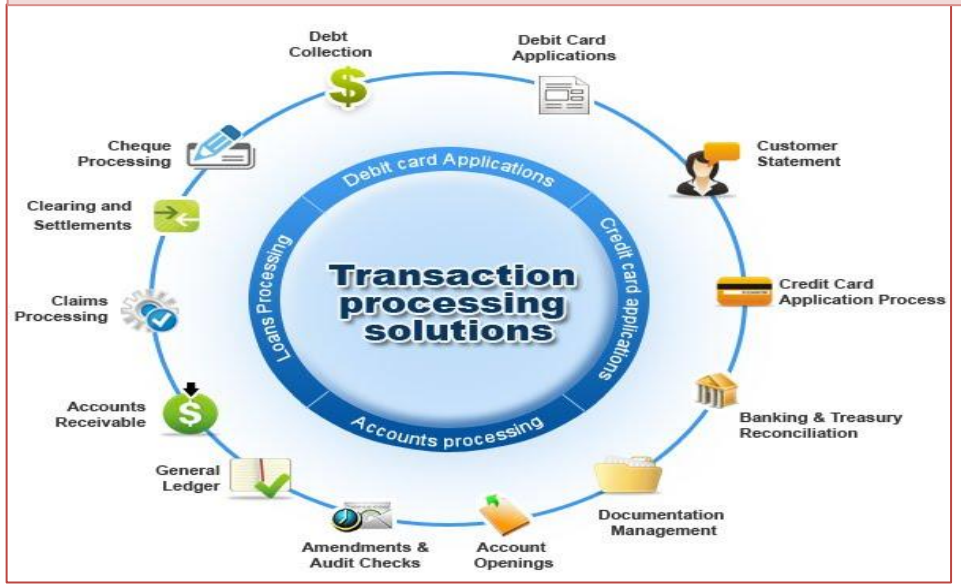
   - **Examples:** OS/390, OS/360.

Mainframe Operating Systems services

Batch OS – processes routine jobs without any interactive user presents
i.e. claim processing in insurance



Transaction processing – handles large numbers of small processes
i.e. cheque processing at banks

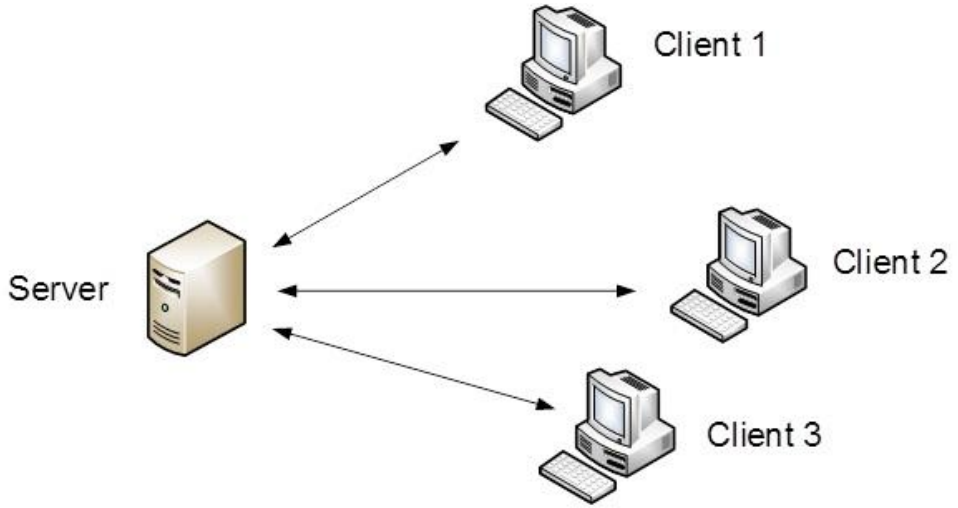


Timesharing – allows multiple remote users to run their jobs at once
i.e. querying a database, airline booking system

Types of Operating Systems (OS)

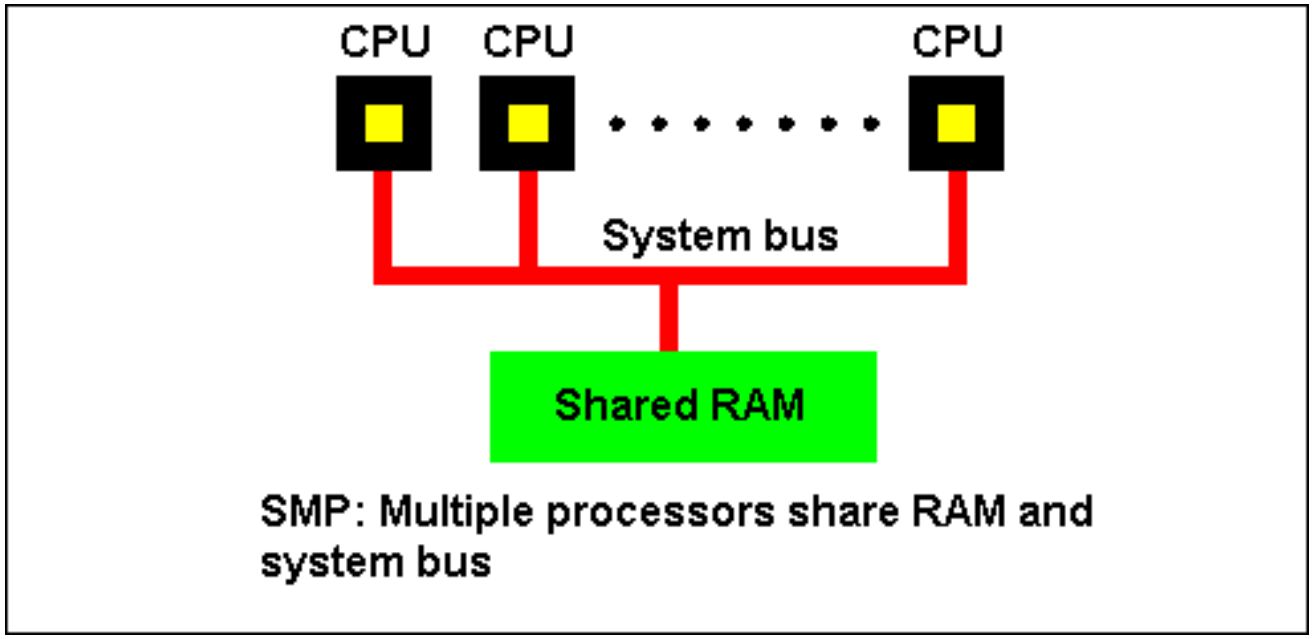
2. Server operating systems
   - This OS runs on servers which are very large PC, workstations or even mainframes.
   - They serve multiple users at once over a network and allow the users to share hardware & software resources.

   - It provides print services, file service or web service.
   - It handles the incoming requests from clients.
   - Examples: Solaris, FreeBSD, and Linux and Windows Server 200x.



Types of Operating Systems (OS)

3. Multiprocessor operating systems
   - A computer system consist two or more CPUs is called multiprocessor.
   - It is also called parallel computers, multicomputer or multiprocessor.
   - They need special OS or some variations on server OS with special features for communication, connectivity and consistency.
   - **Examples:** Windows and Linux.



SMP: Multiple processors share RAM and system bus

Types of Operating Systems (OS)

4. Personal computer operating systems
   - The operating systems installed on our personal computer and laptops are personal OS.
   - Job of this OS is to provide good support to single user.

   - This OS is widely used for word processing, spreadsheet and internet access.
   - Examples: Linux, Windows vista and Macintosh.

Types of Operating Systems (OS)

5. Handhelds computer operating systems
   - A handheld computer or PDA (Personal Digital Assistant) is small computer that fit in a Pocket and perform small number of functions such as electronic address book, memo pad.

   - The OS runs on these devices are handheld OS.
   - These OS also provides ability to handle telephony, digital photography and other functions.
   - Examples: Symbian OS, Palm OS.

Types of Operating Systems (OS)

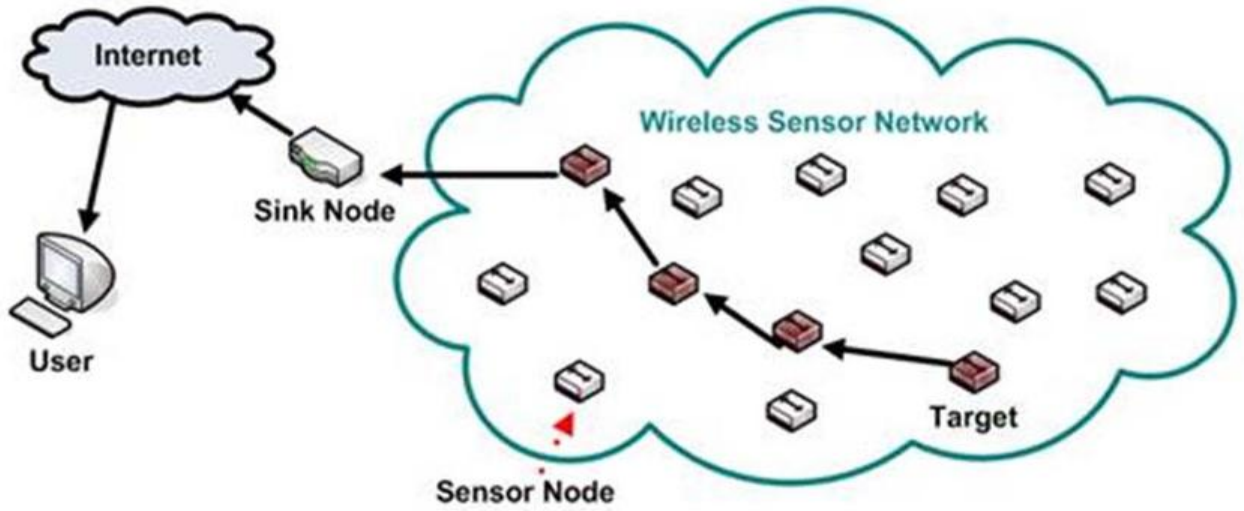
6. Embedded operating systems
   - This OS is installed in ATMs, printers, calculators and washing machine.
   - It runs on the computer that control devices.

   - It neither allow to download new software nor accept user installed software. So there is no need for protection.
   - **Examples:** QNX, VxWorks.

Types of Operating Systems (OS)

7. Sensor node operating systems
   - Network of tiny sensor nodes are being developed for numerous purpose.
   - Each nodes are tiny computers with a CPU, RAM, ROM and one or more environmental sensors.
   - The OS installed in these nodes are sensor node OS.
   - They communicate with each other and with base station using wireless communication.
   - These sensor network are used to protect area of building, detect fires in forest, measure temperature.

   - **Examples:** TinyOS.



Types of Operating Systems (OS)

8. Real time operating systems
   - These systems having time as a key parameter.
   - Real time OS has well defined fixed time constraints.
   - Processing must be done within defined time constraints otherwise system fails.

   - Two types of real time OS:
     - Hard real time – missing an occasional deadline can cause any permanent damage. Many of these are found in industrial process control, car engine control system.
     - Soft real time – missing an occasional deadline does not cause any permanent damage. Used in digital audio, multimedia system.
   - **Examples**: e-Cos.

Types of Operating Systems (OS)

9. Smart card operating systems
   - Smallest OS run on smart cards which are credit card sized devices containing CPU chip.
   - These OS are installed on electronic payments cards such as debit card, credit card etc.

   - They have limited processing power.
   - Some smart cards are Java oriented. ROM on smart card holds an interpreter for the JVM – small program.

System calls

What is System calls?

- A system call is a **way for programs to interact with the operating system**.
- A system call is a **mechanism that provides the interface between a process and the operating system**.
- A computer program makes a system call **when it makes a request to the operating system's kernel**.

- It is a **programmatic method** in which a **computer program requests a service from the kernel** of the OS.
- System call **provides the services of the operating system** to the user programs via **Application Program Interface(API)**.

- System calls are the **only entry points for the kernel system**.

Types of system calls

- **Process Control**: This system calls perform the task of **process creation, process termination**, etc.
  - Functions:
    - End and abort
    - Load and execute
    - Create process and terminate process
    - Wait and signed event
    - Allocate and free memory
- **File Management**: File management system calls handle **file manipulation jobs like creating a file, reading, and writing**, etc.
  - Functions:
    - Create a file
    - Delete file
    - Open and close file
    - Read, write and reposition
    - Get and set file attributes

Types of system calls

- **Device Management**: Device management does the job of **device manipulation like reading from device buffers, writing into device buffers**, etc.
  - Functions
    - Request and release device
    - Logically attach/ detach devices
    - Get and Set device attributes

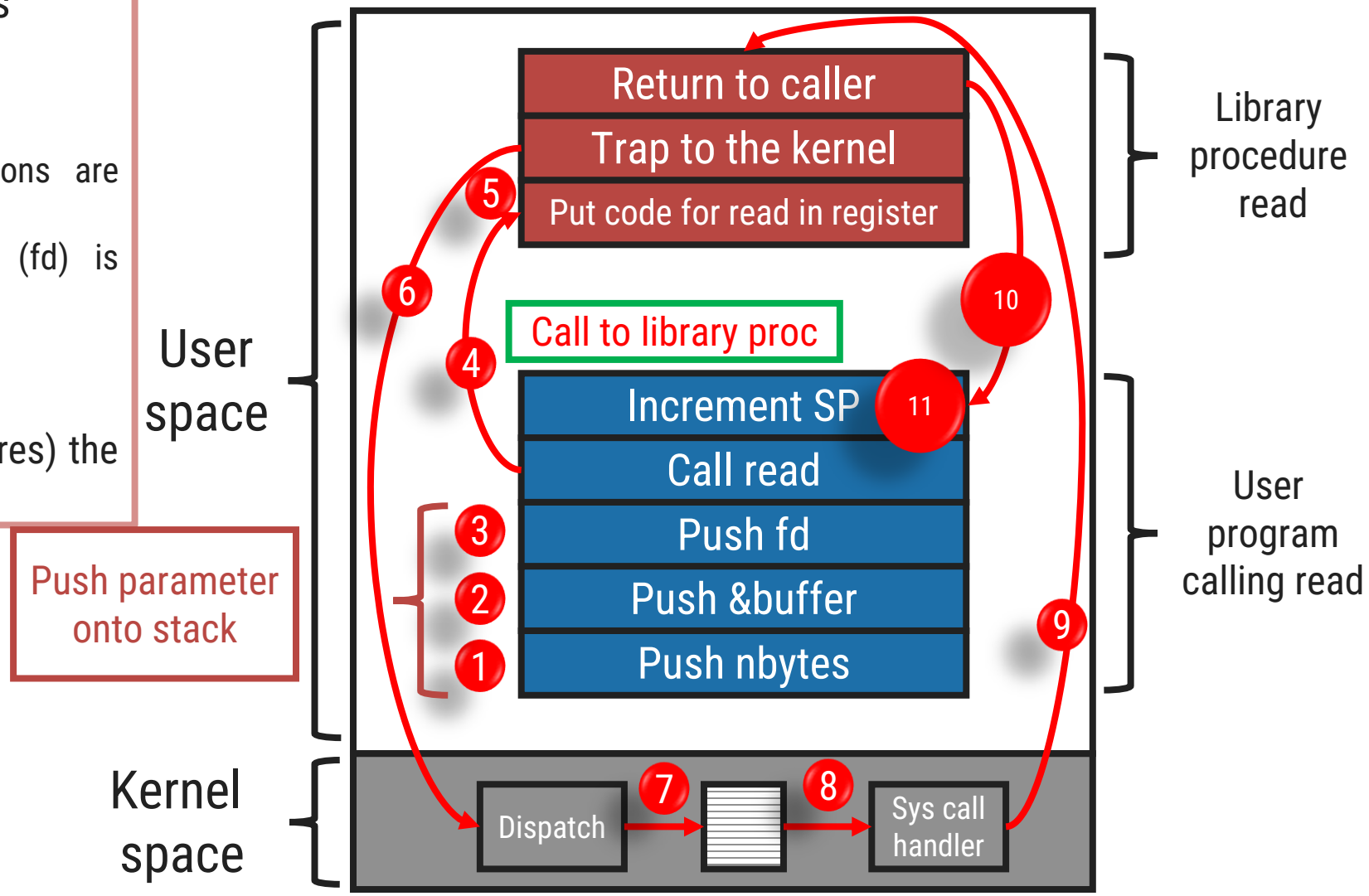
- **Information Maintenance**: It **handles information** and its **transfer between the OS and user program**.
  - Functions:
    - Get or set time and date
    - Get process and device attributes

Types of system calls

- **Communication**: These types of system calls are specially used for **interprocess communications (IPC)**.
  - Functions:
    - Create, delete communications connections
    - Send, receive message
    - Help OS to transfer status information
    - Attach or detach remote devices

Example of system calls (Read system call)

- Example: In Unix Read system call is
  - count = read(fd, buffer, nbytes)
    - fd is a file descriptor.
      - When a file is opened, permissions are checked.
      - If access is allowed, a number (fd) is returned.
      - Then file can be read/written.
    - nbytes is number of bytes to read
    - buffer is where read deposits (stores) the data



Steps involved in making a system call (Read system call)

- Steps involved in making a system call
  - In steps 1-3, the calling program pushes the parameters onto the stack. The first and third parameters are called by value, but the second one is called by its address as denoted by the & symbol.
  - In step 4, the actual call to the library procedure is made. This instruction is the normal procedure call instruction used to call all procedures.
  - In step 5, the library procedure places the system call number in a place where the operating system expects it, such as a register.
  - In step 6, the library procedure executes a TRAP instruction to switch from user mode to kernel mode and start execution at a fixed address within the kernel.
  - In step 7, the kernel examines the system call number and then dispatches it to the correct system call handler. This correct number is given in the table of system call handlers by pointers referenced at the system call number.
  - In step 8, the system call handler runs.
  - In step 9, the operation is completed, and the user is given back control once the TRAP instruction is set.
  - In step 10, this procedure returns to the user program, like how all normal library procedures do.
  - In step 11, the operating system has to clear the stack, so it increments it enough so that it is empty.

System calls

Director and file system management	
Call	Description
s = mkdir(name,mode)	Create a new directory
s = rmdir(name)	Remove an empty directory
s = link(name1, name2)	Create a new entry, name2, pointing to name1
s = unlink(name)	Remove a directory entry
s = mount(special, name, flag)	Mount a file system
s = umount(special)	Unmount a file system
Miscellaneous	
Call	Description
s = chdir(dir name)	Change the working directory
s = chmod(name,mode)	Change a file's protection bits
s = kill(pid, signal)	Send a signal to a process
seconds = time(&seconds)	Get the elapsed time since Jan. 1, 1970

System calls

Process management	
Call	Description
pid = fork( )	Create a child process identical to the parent
pid = waitpid(pid, &statloc, options)	Wait for a child to terminate
s = execve(name, argv, environp)	Replace a process' core image
exit(status)	Terminate process execution and return status
File management	
Call	Description
fd = open(file, how, ...)	Open a file for reading, writing, or both
s = close(fd)	Close an open file
n = read(fd, buffer, nbytes)	Read data from a file into a buffer
n = write(fd, buffer, nbytes)	Write data from a buffer into a file
position = lseek(fd, offset, whence)	Move the file pointer
s = stat(name, &buf)	Get a file's status information

Operating Systems (OS) structure

Operating Systems (OS) structure

1. Monolithic systems
2. Layered systems
3. Microkernel
4. Client-server model
5. Virtual machines
    I. VM/370
    II. Virtual machines rediscovered
    III. The java virtual machine
6. Exokernels

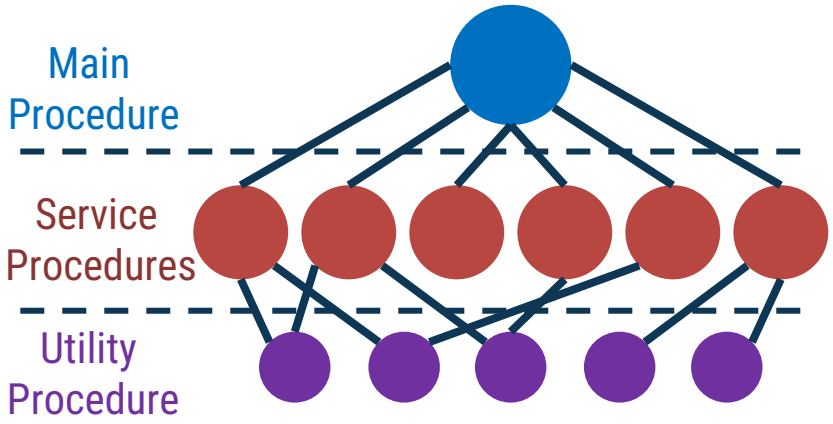
Monolithic systems

- The entire OS **runs as a single program in kernel mode**.
- OS is written as a **collection of procedures, linked together into a single large executable binary program**.
- Each procedure has well defined interface in terms of parameter and results, and each one is free to call any other one.

✓ A main program that invoke the requested service procedure.

✓ A set of service procedures that carry out the system calls.

✓ A set of utility procedures that help the service procedure.



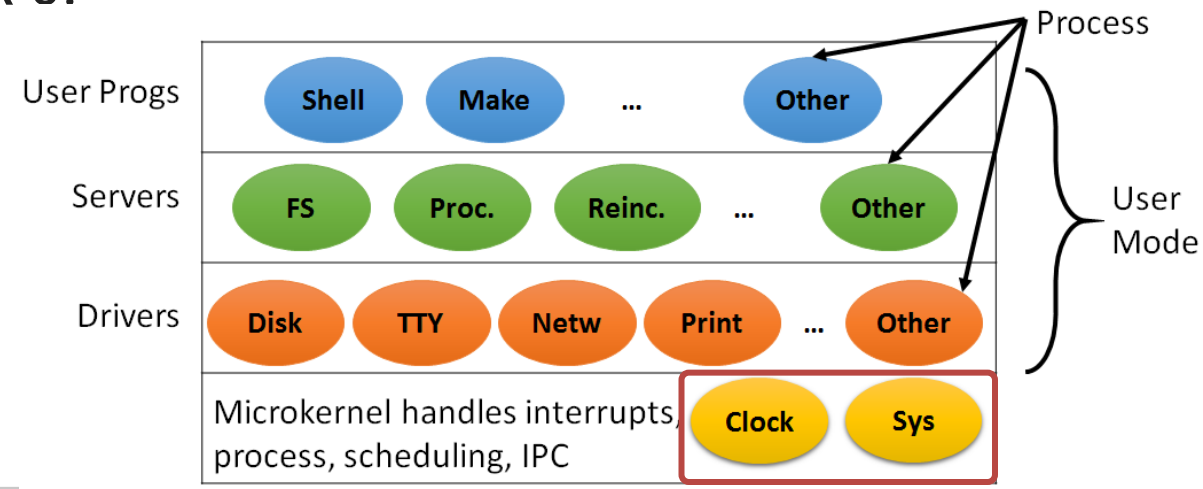
Layered systems

- In this system, the OS is organized as a **hierarchy of layers**.
- The first system constructed in this way was **THE system**.

Layer	Function	Description
5	Operator	**Operator** was located.
4	User programs	**User programs** were found.
3	Input / Output management	Takes care of **managing the I/O devices**. Buffering the information.
2	Operator-process communication	**Handles communication** between each process and the operator console (i.e. user).
1	Memory and drum management	Did the **memory management**. Allocated space for process in main memory and on a 512K word drum used for holding parts of processes for which there was no room in memory.
0	Processor allocation and multi-programming	Provided the **basic multiprogramming** of the CPU. Dealt with allocation of the processor, switching between processes when interrupts occurred or timers expired.

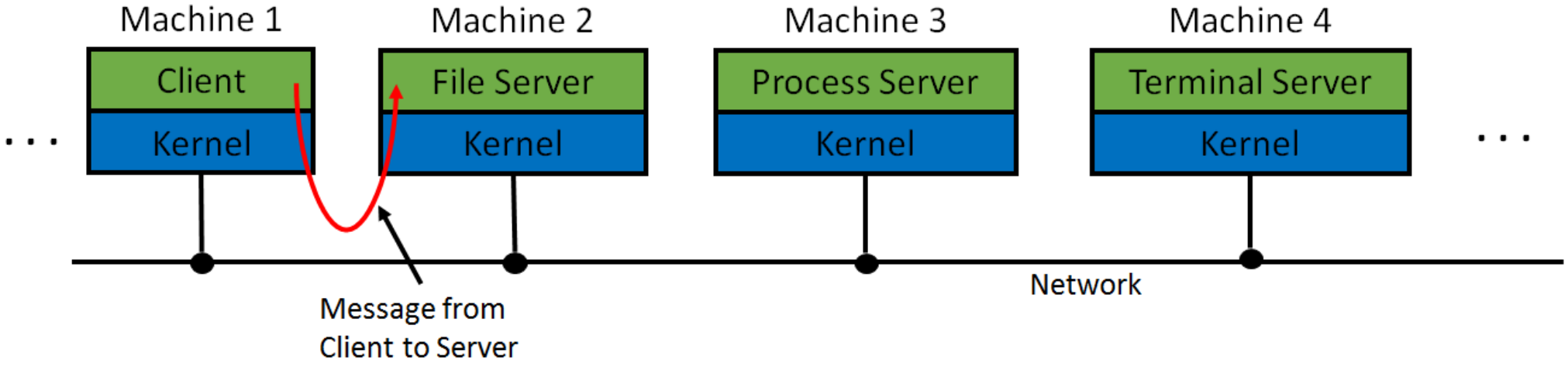
Microkernel

- In layered approach, the **designer have choice where to draw the kernel and user mode boundary**.
- It is **better to put as little as possible in kernel mode** because bugs in the kernel can bring down the system instantly.
- The microkernel design provides **high reliability by splitting OS up into small well defined modules, only one module run in kernel and rest of all run in user mode**.
- As each device driver runs as a user process, a bug in audio driver will cause the sound to be stop, but not crash the computer.
- Examples: Integrity, K42, QNX, Symbian and MINIX 3.
- Kernel contains only
  - Sys (Kernel call handler)
  - Clock (because scheduler interact with it)



Client-Server model

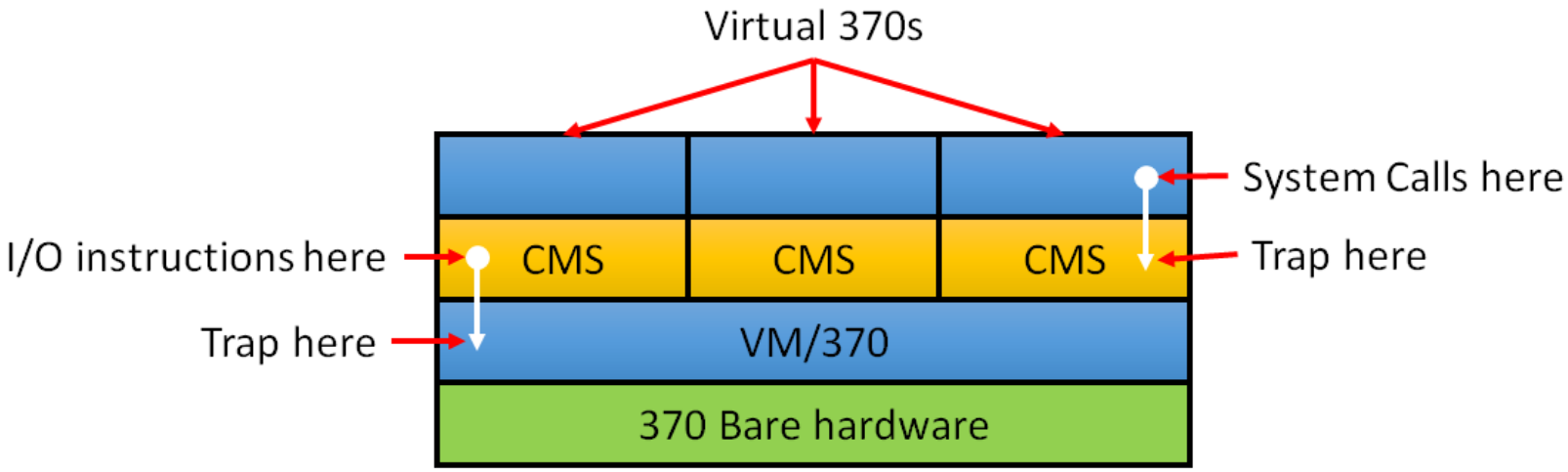
- Processes are divided into two categories
  - **Servers**: provide services
  - **Clients**: uses services
- Client and server **run on different computers, connected by LAN or WAN and communicate via message passing**.
- To obtain a service, **a client construct a message** saying **what it wants and send it to server**.
- The **server then does the work and send back the answer**.



Virtual machines

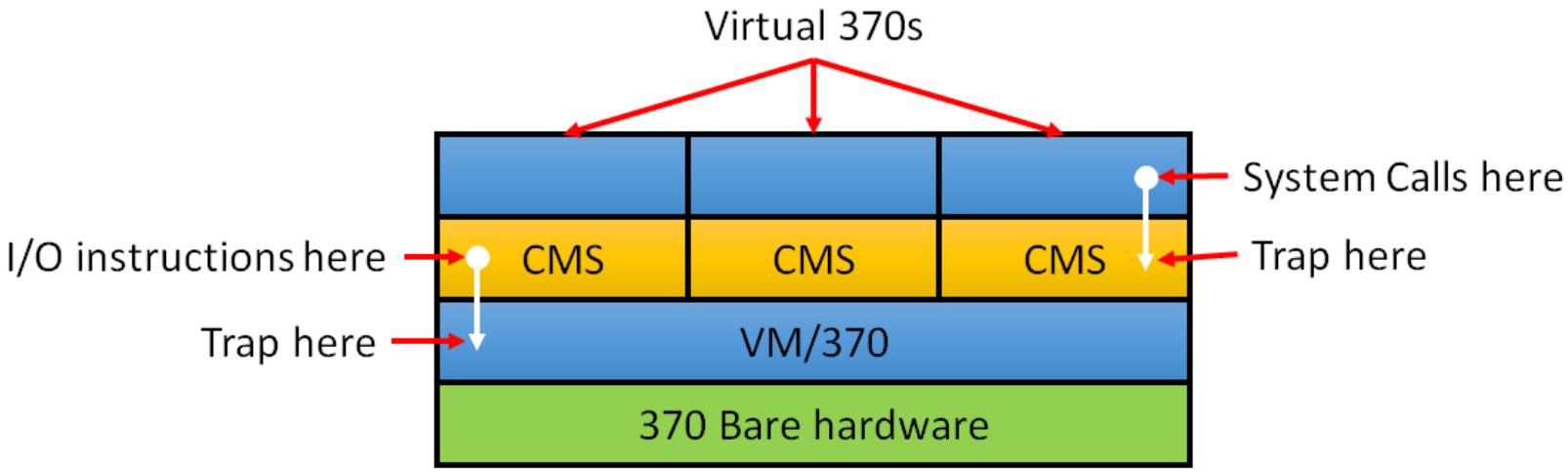
- A virtual machine (VM) is a **virtual environment that functions as a virtual computer system with its own CPU, memory, network interface, and storage, created on a physical hardware system**.
- The **initial releases of OS/360 were strictly batch systems**.
- But **many users wanted to be able to work interactively** at a terminal, so OS designers decided to write timesharing systems for it.
- Types of Virtual machines are:
  - VM/370
  - Virtual Machines Rediscovered
  - The Java Virtual Machine

VM/370



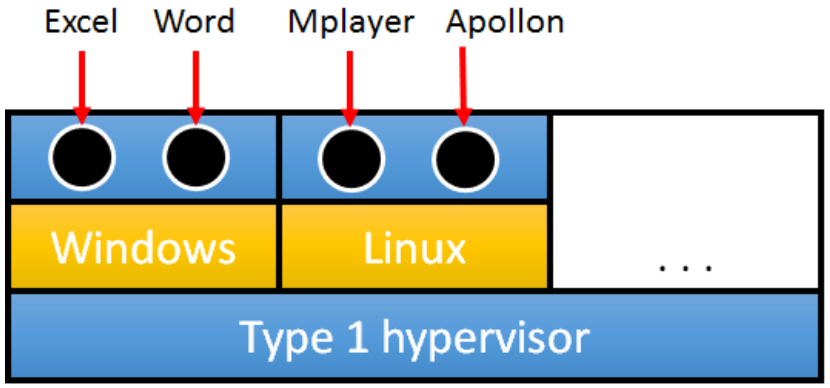
- Virtual machine monitor **run on the bare hardware and does the multiprogramming**.
- Each virtual machine is **identical to the true hardware**; each one can run any OS (may be different) that will run directly on the bare hardware.
- On VM/370, some run OS/360 while the others run single user interactive system called CMS (Conversational Monitor System) for interactive time sharing users.

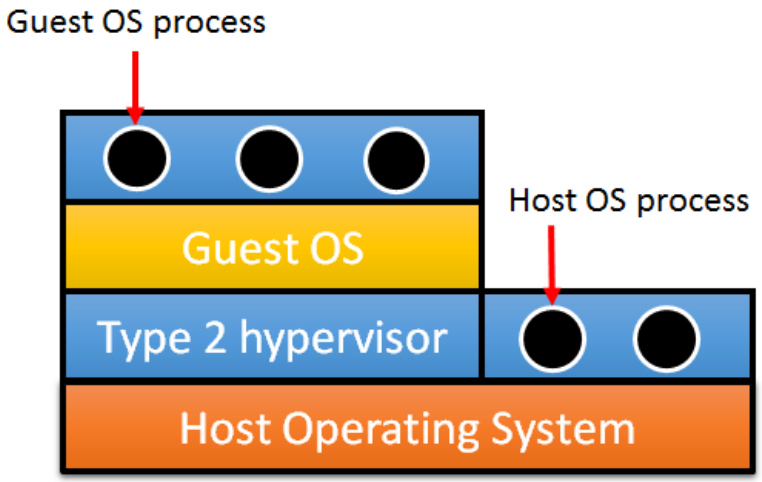
VM/370



- When CMS program executed a system call, a call was trapped to the operating system in its own virtual machine, not on VM/370.
- CMS then issued the normal hardware I/O instruction for reading its virtual disk or whatever was needed to carry out the call.
- These I/O instructions were trapped by VM/370 which then performs them.

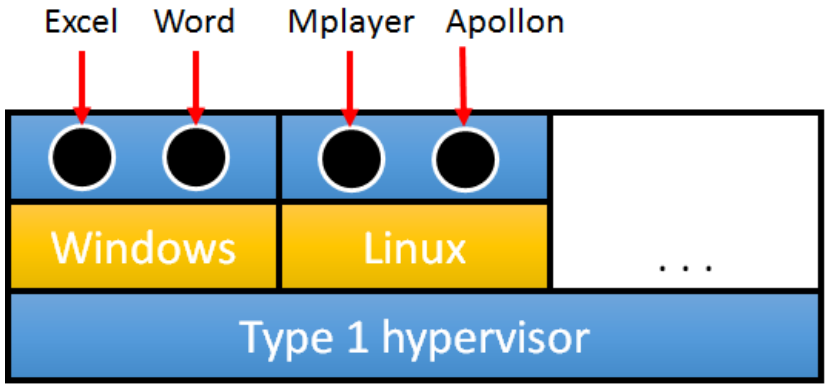
Virtual Machines Rediscovered

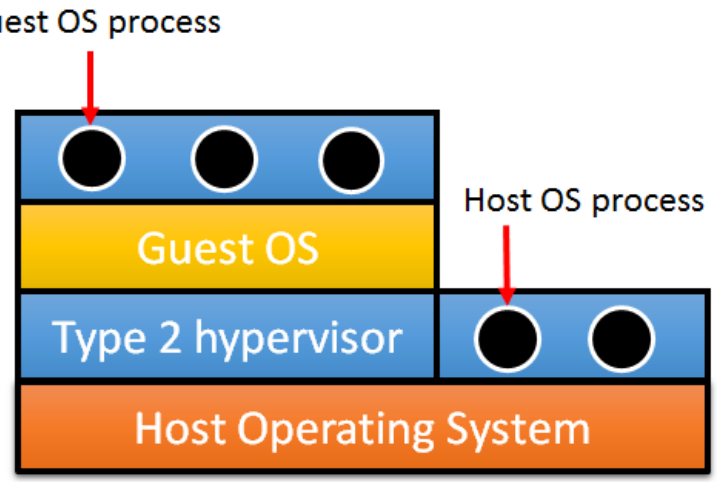




- Companies can **run their mail servers, web servers, FTP servers and other servers on the same machine without having a crash of one server bring down the rest**.
- **Web hosting company offers virtual machines for rent**, where a single physical machine can run many virtual machines; each one appears to be a complete machine.
- Customers who rent a virtual machine can run any OS or software.

Virtual Machines Rediscovered

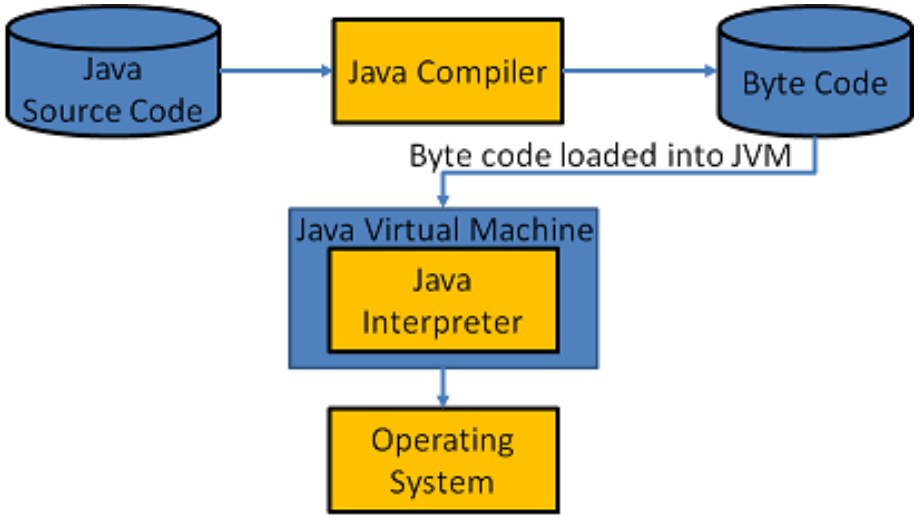




- Another use of virtualization is for **end users who want to be able to run two or more operating systems at the same time, say Windows and Linux**.
- Type 1 hypervisors **runs on the bare hardware**.
- Type 2 hypervisors **run as application programs on top of Windows, Linux, or some other operating system, known as the host operating system**.

Java Virtual Machine

- When Sun Microsystems invented the Java programming language, it also invented a virtual machine called the JVM (Java Virtual Machine).
- The Java compiler produces code for JVM, which then typically is executed by a software JVM interpreter.
- The advantage is that the JVM code can be shipped over the internet to any computer that has a JVM interpreter and run there.

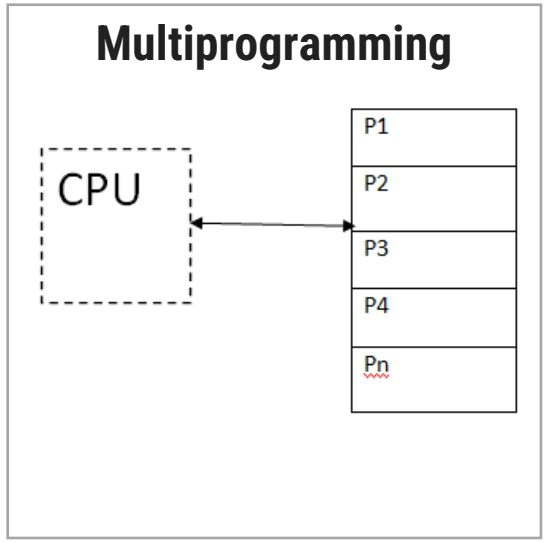


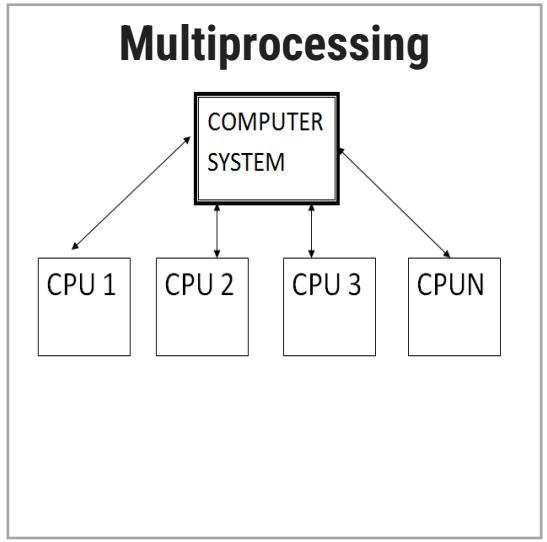
Exokernels

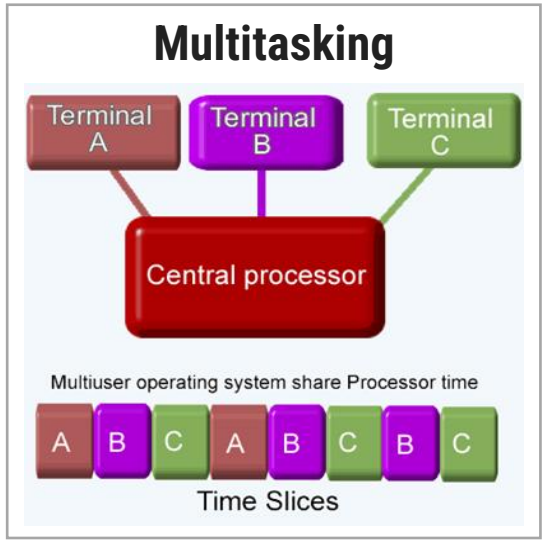
- Rather than cloning (copying) the actual machine, **another strategy is partitioning it** (giving each user a subset of the resource).
- For example one virtual machine might get disk blocks 0 to 1023, the next one might get block 1024 to 2047, and so on.
- **Program running at the bottom layer (kernel mode) is called the exokernel**.
- Its job is to **allocate resources to virtual machines and then continuously check to make sure no machine is trying to use somebody else's resources**.
- The advantage of the exokernel scheme is that it **saves a layer of mapping**.

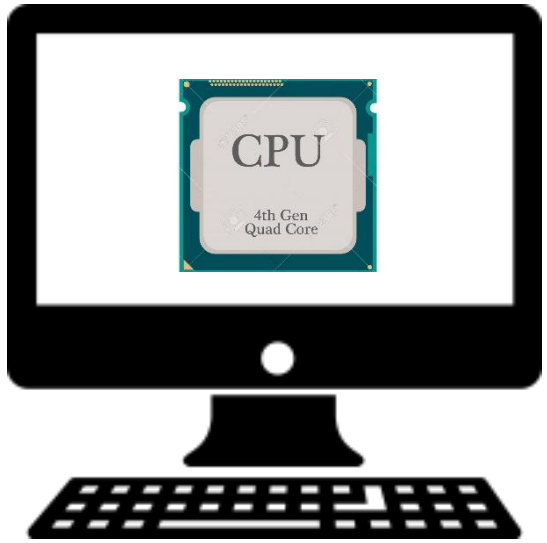
Multiprogramming v/s Multiprocessing v/s Multitasking

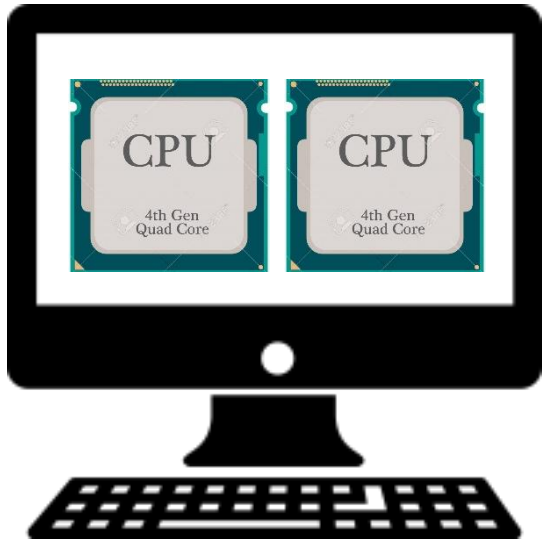
Multiprogramming v/s Multiprocessing v/s Multitasking

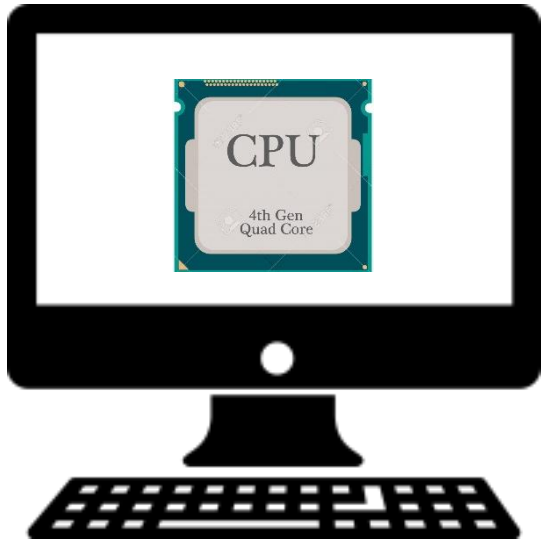












Multiprogramming v/s Multiprocessing v/s Multitasking

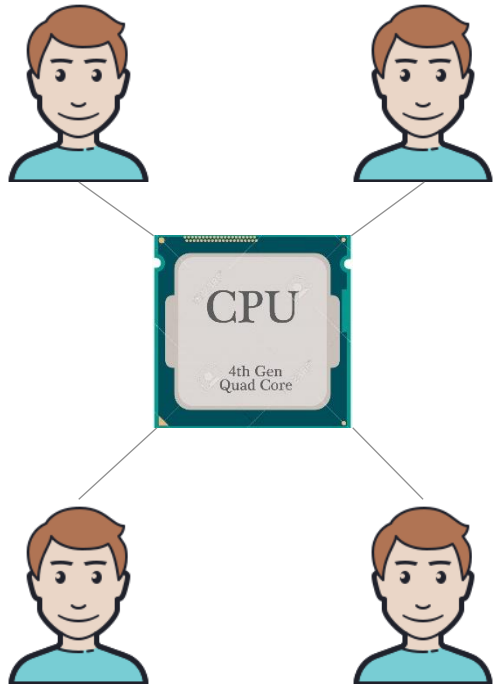
Multiprogramming	Multiprocessing	Multitasking
The concurrent residency of more than one program in the main memory is called as multiprogramming.	The availability of more than one processor per system, which can execute several set of instructions in parallel is called as multiprocessing	The execution of more than one task simultaneously is called as multitasking.
Number of processor: one	Number of processor: more than one	Number of processor: one
One process is executed at a time.	More than one process can be executed at a time.	One by one job is being executed at a time.

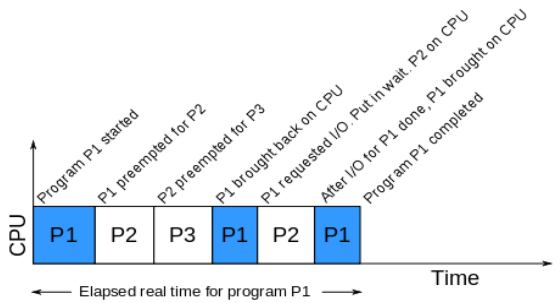
Multitasking is a logical extension of multi programming. The major way in which multitasking differs from multi programming is that **multi programming works solely on the concept of context switching whereas multitasking is based on time sharing alongside the concept of context switching**.

Time Sharing Operating System

Time Sharing Operating System

- A time sharing operating system **allows many users to share the computer resources simultaneously**.
- In other words, time sharing refers to the **allocation of computer resources in time slots to several programs simultaneously**.
- For example a **mainframe computer that has many users logged on to it**.
- Each user uses the resources of the mainframe i.e. memory, CPU etc.

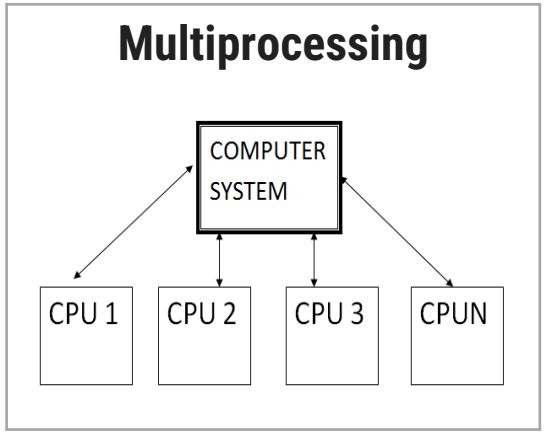


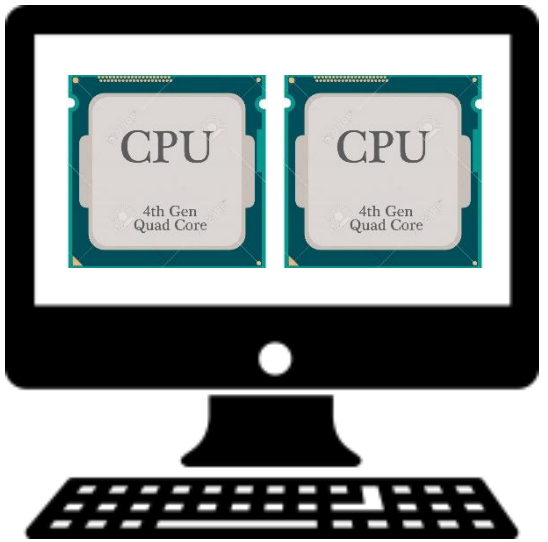


Parallel Processing Operating System

Parallel Processing Operating System

- Parallel Processing Operating Systems are **designed to speed up the execution of programs by dividing the program into multiple fragments and processing these fragments simultaneously**.
- Such systems are multiprocessor systems.
- Parallel systems deal with the **simultaneous use of multiple computer resources that can include a single computer with multiple processors**.

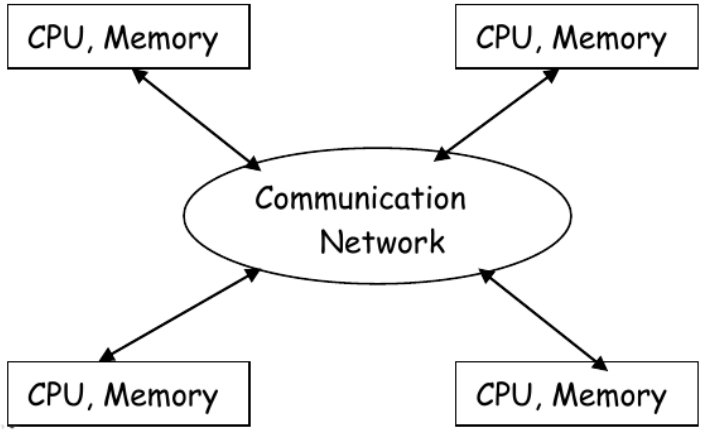




Distributed Operating System

Distributed Operating System

- Distributed Operating System is a **model where distributed applications are running on multiple computers linked by communications**.
- A distributed operating system is an **extension of the network operating system** that **supports higher levels of communication and integration of the machines on the network**.



Practice Questions

1. What is Kernel? Differentiate between Monolithic Kernel and Micro Kernel.
2. Explain different service/functions provided by operating system.
3. Discuss role of OS as a resource manager.
4. Explain the features of Time sharing system.
5. What is operating system? Give the view of OS as a resource manager.
6. What is system call? Explain steps for system call execution.
7. Write different types of system call.
8. List out types of operating system and explain batch OS and time sharing OS in brief.

Thank You