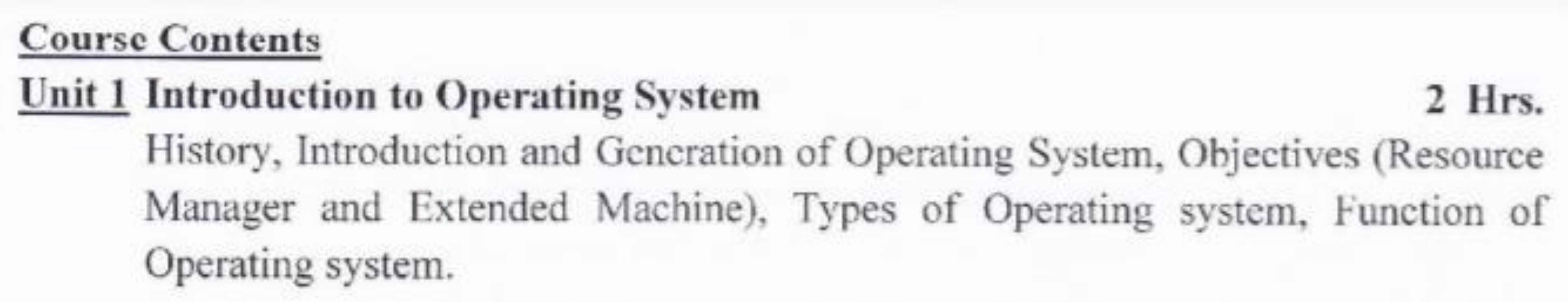
Operating system (OS)

kashiram pokharel

Chapter 1:Operating system type and structure



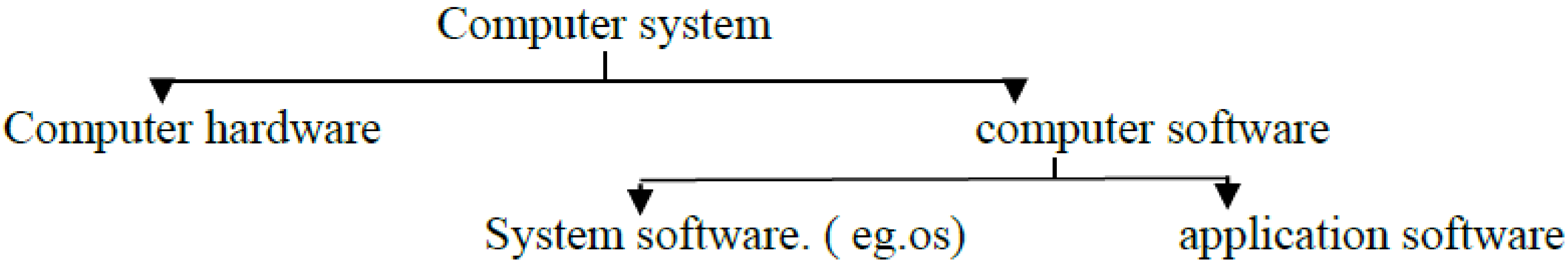
Course Contents

Unit 1 Introduction to Operating System **2 Hrs.**

History, Introduction and Generation of Operating System, Objectives (Resource Manager and Extended Machine), Types of Operating system, Function of Operating system.

Introduction

- computer system consists of **hardware and software**. hardware is those objects which can touch. software is a term for set of instruction and data which makes the computer to perform task. let us take an overview of computer software.



Computer system

Computer hardware

computer software

System software. (eg.os)

application software

Operating system

- An **operating system** is a system software that **manages** the computer hardware. **OS** is the **intermediate system program** between users and hardware which are very essential for the computer system to work and which **creates user friendly interaction between user and computer system** by **hiding all low level (hardware) detail.**
- **OS** is the most fundamental of all the system program because it **controls** all the system resources (memory, input/output devices ,program, files etc) and **provides a base** upon which all other application program are written.
- The operating system **controls and coordinates** the use of hardware among various application programs for the various users.
- An **operating system** is a program that acts as an **intermediary** between user of computer and computer hardware. i.e. it **enables the computer hardware** to communicate and operate with computer software. the purpose of an OS is to **provide an environment** in which a user can execute programs in a convenient and efficient manner.

Pros and cons of Operating system

Advantage of using Operating System

- Allows you to **hide details of hardware by creating an abstraction**
- Easy to use with a **GUI**
- Offers an **environment** in which a user may execute programs/applications
- The operating system must make sure that the computer system **convenient to use**
- Operating System acts as an intermediary among applications and the hardware components
- It provides the computer system resources with easy to use format
- Acts as an **intermediator** between all hardware's and software's of the system

Disadvantages of using Operating System

- **If any issue occurs in OS**, you may lose all the contents which have been stored in your system
- Operating system's software is **quite expensive** for small size organization which adds burden on them. Example Windows
- **It is never entirely secure as a threat can occur at any time**

Features of Operating System (OS)

- Protected and supervisor mode
- Allows disk access and file systems Device drivers Networking Security
- Program Execution
- Memory management Virtual Memory Multitasking
- Handling I/O operations
- Manipulation of the file system
- Error Detection and handling
- Resource allocation
- Information and Resource Protection

Function of OS:

- **Process management**:- Process management helps OS to create and delete processes. It also provides mechanisms for synchronization and communication among processes.
- **Memory management:-** Memory management module performs the task of allocation and de-allocation of memory space to programs in need of this resources.
- **File management**:- It manages all the file-related activities such as organization storage, retrieval, naming, sharing, and protection of files.
- **Device Management**: Device management keeps tracks of all devices. This module also responsible for this task is known as the I/O controller. It also performs the task of allocation and de-allocation of the devices.
- **I/O System Management:** One of the main objects of any OS is to hide the peculiarities of that hardware devices from the user.
- **Job accounting**: Keeping track of time & resource used by various job and users.

Contd..

- **Secondary-Storage Management**: Systems have several levels of storage which includes primary storage, secondary storage, and cache storage. Instructions and data must be stored in primary storage or cache so that a running program can reference it.
- **Security**:- Security module protects the data and information of a computer system against malware threat and authorized access.
- **Command interpretation**: This module is interpreting commands given by the and acting system resources to process that commands.
- **Networking:** A distributed system is a group of processors which do not share memory, hardware devices, or a clock. The processors communicate with one another through the network.
- **Communication management**: Coordination and assignment of compilers, interpreters, and another software resource of the various users of the computer systems.

Two different views of an OS

1. EXTENDED MACHINE VIEW
2. RESOURCE MANAGER VIEW

OPERATING SYSTEM AS AN EXTENDED MACHINE

- **User view** – the OS as an extended machine.
    - virtual machine that is easier to understand and program
    - tool to make programmer’s job easy
- Use the computer hardware in an efficient manner (converting hardware into useful form), and **“hide” the complexity** of the underlying hardware.
    - ➢operating systems **turn ugly hardware into beautiful abstractions**.
    - ➢OS provides **simple**, **high level abstraction** to the application programs
- OS offers several services to the application programs through availability of **system calls**
- Because of the **nice, clean, elegant and consistent,** abstractions that the OS provides, a programmer does not need to get too intimate with the hardware
- Top-down (user/application centered) view.
- **“Beautification principle”,**
    - ``hide'' the complexity of the underlying hardware
    - OS hides the implementation details

CONTD…

- **OS provides stable, portable, reliable, sage and well behave environment to the user.**
  - **stable**: doesn't crash
  - **portable**: can run code on more than one type of machine
  - **reliable**: always reacts in the same way
  - **safe**: doesn't do something dangerous.
  - **well-behaved**: acts in a proper manner
- **Features**:
  - threads, processes, files, communication channels.
- The OS in such cases is designed to maximize resource utilization to assure that all available CPU time, memory, and I/O are used efficiently and that no individual user takes more than fair share

Os as a resource manager

- resource management includes multiplexing resources in two different ways: **time and memory**
- when a resource is **time multiplexed**, different programs or users take turns using it .eg: cpu sharing
- instead of taking turns, each programs gets part of the resource **for example**: main memory sharing
- **The resources like memory space, CPU time, file storage space are properly managed**

with the help of the operating system. It support many devices simultaneously– e.g. keyboard, mouse, printer, speakers, microphone

- **Fairly, safely and efficiently share resources among** users and programs

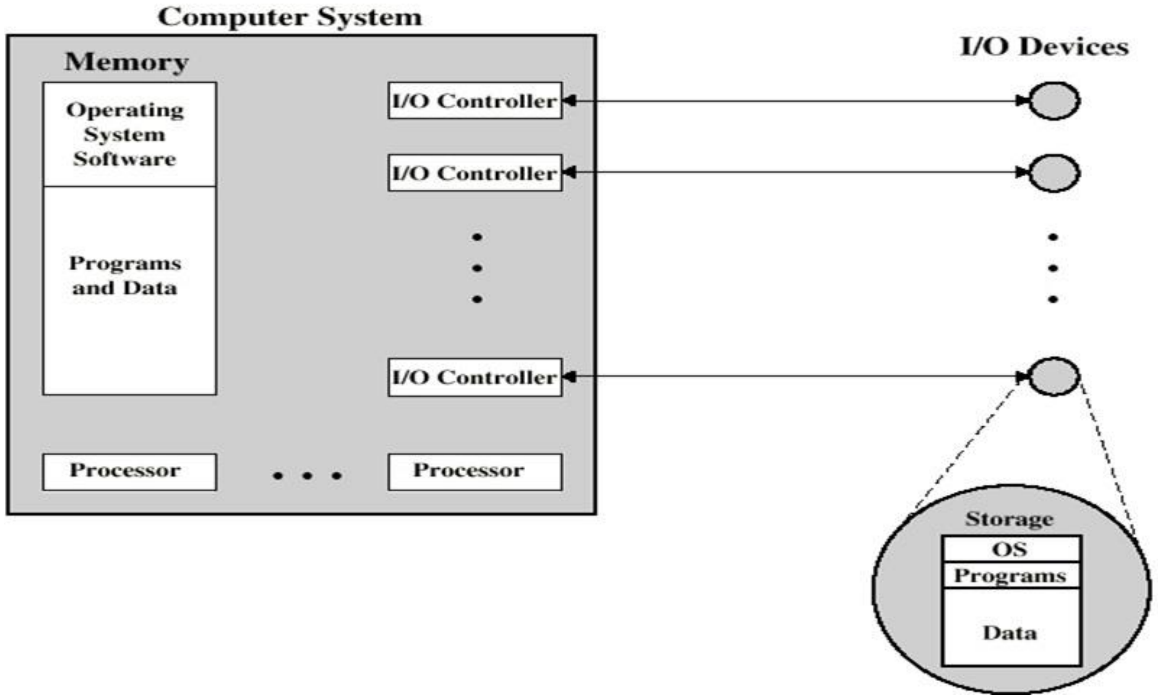
  fairly: each program gets a chance to execute.

  safely: protects against corruption

  efficiently: using the available resources to provides the best possible service

- viewing os as a resource manager, each manager must do following:
- **To keep track of the resources:**
  - **for memory**, it keep track which parts are in use and which are free, and also who are the user etc
  - **for process**, it keep track of the processors being allocated to which process and also keep track of the status of the processes whether it is in running, ready or waiting state.
  - **for device**: it keep track of which device are in used and which are free and who are the users etc
  - **for files or information**: it keep track of the location of the files ,it’s users and status of file(opened or closed)
- **enforce policy** that determines who get what, when and how much. this is known as scheduling.
- **Allocate** the resources to process and **de-allocated** the resources when free. i.e. Allocates resources to users such as disks, memory, network interfaces, timers, terminals/displays, laser printers, etc
- **Keep information** about who is using what? And how is it shared?

- maintains a “**multi-user**” illusion to several users
- most systems allow several user “programs” (even if you think you are only running one program, there are several internal programs running)
- Os manages
  - processors: 1, 2 or several
  - memories: RAM, cache
  - timers: internal, external
  - disks
  - mouse
  - network interfaces: intranet, LAN, wan
  - laser printers: several printers, priority
  - other devices.



History of os

Generation and devices	Year	Electronic devices used	Types of OS and device
First	1945 – 55	Vacuum tubes	Plug boards
Second	1955 - 1965	Transistors	Batch system,
Third	1965 - 1980	Integrated Circuit (IC)	Multiprogramming
Fourth	Since 1980	Large scale integration	Client/server, Distributed

First Generation (Early 1945's to about 1955)

- When the first electronic computer was developed in 1940, it was created **without any operating system**.
- In early times, **users have full access to the computer machine and write a program for each task in absolute machine language**. All **programming was done in absolute machine language**, often by wiring up plug boards to control the machine's basic functions.
- The programmer can perform and solve only simple mathematical calculations during the computer generation, and this calculation does not require an operating system.
- First form of **automatic scheduling**: users gives deck to operator who reads them in via card reader. The jobs are executed (in FIFO order) and listed on printer. User has no control over job during processing. Processor utilization goes up to 55 percent.
- User becomes disassociated from machine - **any machine handling is done by operators.**
- During this generation computers were generally used to solve simple math calculations, operating systems were not necessarily needed.

Second Generation (1955 - 1965)

- For the first time, there was a **clear separation between designers, builders, operators, programmers, and maintenance personnel**.
- The first operating system (OS) was created in the early 1950s and was known as **GMOS. General Motors** has developed OS for the **IBM** computer.
- The second-generation operating system was based on a single stream **batch processing** system because it collects all similar jobs in groups or batches and then submits the jobs to the operating system using a punch card to complete all jobs in a machine. At each completion of jobs (either normally or abnormally), control transfer to the operating system that is cleaned after completing one job and then continues to read and initiates the next job in a punch card
- So operating systems **designers developed the concept of multiprogramming** in which several jobs are in main memory at once; a processor is switched from job to job as needed to keep several jobs advancing while keeping the peripheral devices in use.
- To run a job (i.e., a program or set of programs), a programmer would first write the program on paper (in FORTRAN or assembler), then punch it on cards. He would then bring the card deck down to the input room and hand it to one of the operators

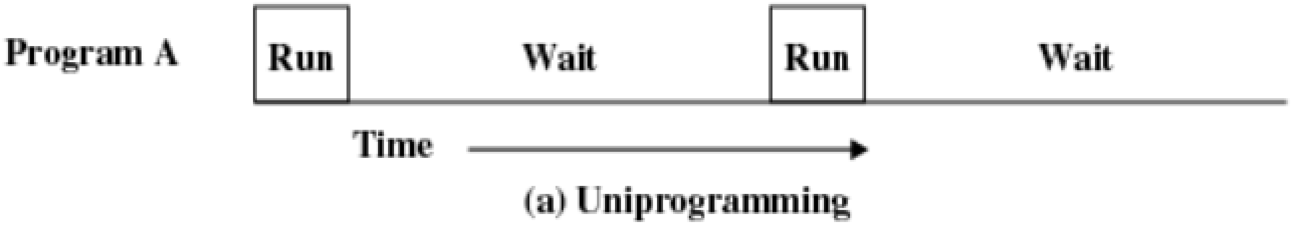
Third Generation (1965 - Mid 1980's)

- **Standard I/O routines (device drivers) begin to appear** - unburdening the user with the responsibility of direct peripheral device communication.
- **Multiprogramming (multiplexing)** develops. **Multiprocessor** configurations begin to appear.
- (Data) **Channels (Autonomous Peripheral Devices)** allow **concurrent** CPU - I/O processing. Synchronization of the CPU and I/O device provided by an **interrupt**, or timing signal set by peripheral in a register which is examined by CPU after execution of each instruction (contrast with polling).
- High-speed direct-access secondary storage devices become available. **Direct Memory Access** (**DMA**) allows for large amounts of data to be directly transferred from such devices to memory without intervention on the part of the CPU.
- **Timesharing** (interactive systems) and **real-time** systems emerge (mainly as a result of multiprogramming).
- **Age of mammoth operating systems**. Characterized by **IBM's OS/360**.
- **Systems became multimode** - allowed for mixing of batch, timesharing and real-time processing.
- System complexity introduces need for controlling mechanism; **job control languages** are introduced.
- **Virtual storage** appears.

Fourth Generation (Mid 1980's to Present)

- **Kernel-based systems** becomes the way to go.
- **Personal computers** begin to have their own OS needs.
- Networks and PC's introduce a new category of user.
- Through **virtual machines** user can now ignore physical machine and work with machine presented by the operating system.
- GUI interfaces begin replacing / supplementing command-line interfaces.
- Client server architecture and distributed system supported.

Type of os



(a) Uniprogramming

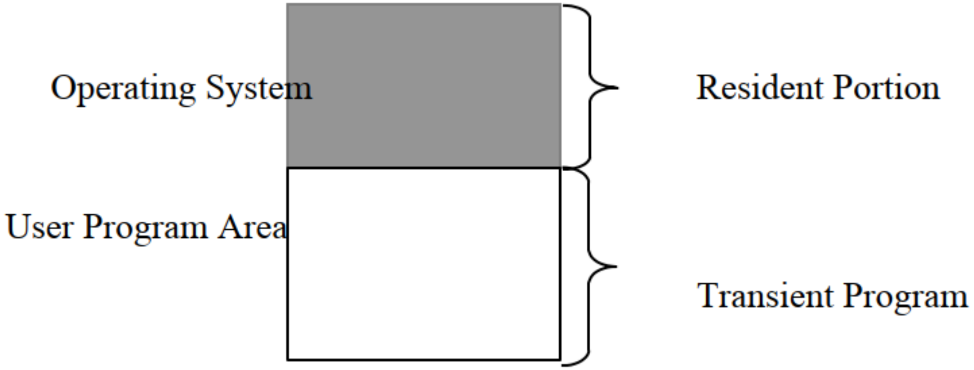
1. batch OS

- Uni-programming system i.e. systems only did one thing at a time.
- Batch operating system is one where programs and data are collected together in a batch before processing starts. A job is predefined sequence of commands, programs and data that are combined in to a single unit called job.
- The **users of a batch operating system do not interact with the computer directly.** each user prepares his job on an off-line device like punch cards and submits it to the computer operator
- Programmer submits job (program or a set of program), which are collected forming a batch of jobs
- The batch is then written to a magnetic tape using inexpensive machines like IBM 1401
- Operator loads a special program (ancestor of today's OS) into the machine (like 7094) which reads first job from the tape and runs it. Then the output is written in a separate tape

- After a job is finished, the os automatically reads the next job in the batch.
  - After whole batch is completed, the operator removes the input and output tape, replace the input tape with the next batch.
  - Output tape is brought for printing
  - example: FMS (fortran monitor system), ibsys etc.
- **Memory management** in batch system is very simple. Memory is usually divided into two areas: **Operating system and user program area**.
- **Scheduling** is also simple in batch system.

  Jobs are processed in the order of submission i.e.

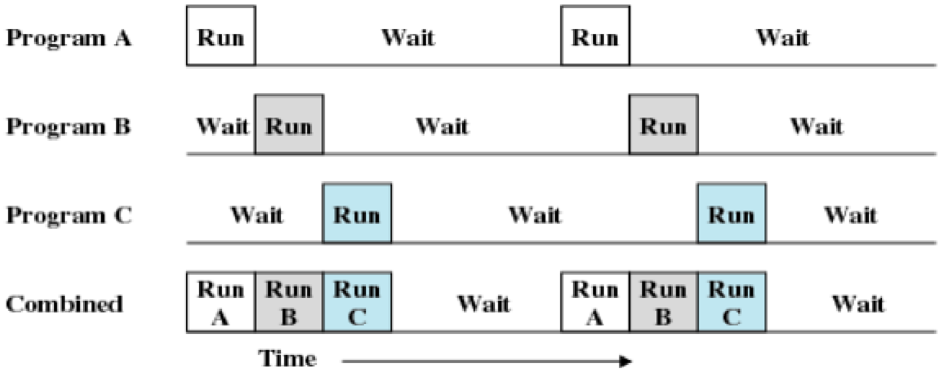
  first come first served fashion.



The problems with Batch Systems are as follows –

- lack of interaction between the user and the job.
- CPU is often idle, because the speed of the mechanical I/O devices is slower than the CPU .
- difficult to provide the desired priority.
- Turnaround time can be large from user standpoint. Difficult to debug program.
- A job could enter an infinite loop.

2. Time Sharing system



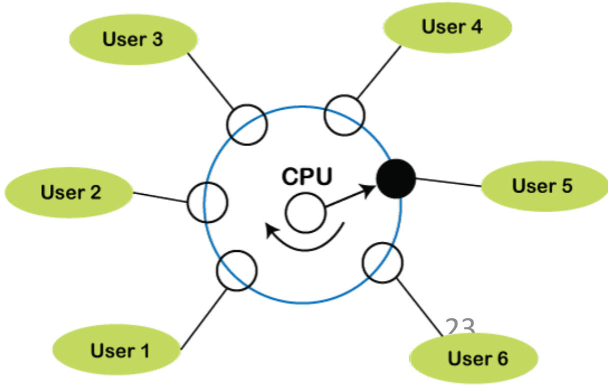
(c) Multiprogramming with three programs

- In a multiprogramming system there are one or more programs loaded in main memory which are ready to execute.
- Only one program at a time is able to get the CPU for executing its instructions (i.e., there is at most one process running on the system) while all the others are waiting their turn.
- Time sharing, or multitasking, is a logical extension of multiprogramming. In this approach, memory was partitioned into several pieces with a different job in each partition.
- Multiple jobs are executed by the CPU switching between them, but the switches

  occur so frequently that the users may interact with each program while it is running
- While one job is paused to wait for an i/o operation, another job could be started.

TIME SHARING (MULTITASKING) OS

- It is a variant of multiprogramming, in which processor time is shared among multiple users. it provides each user has his terminal for input or output that impacts the program or processor currently running on the system
  - In a time sharing system, multiple users simultaneously access the system through terminals.
  - The system switches so rapidly from one user to another, each user is given the impression that he has his own computer.
  - Multiple jobs are executed by the CPU switching between them, but the switches occur so frequently that the users may interact with the system while it is executing.
  - Emphasizes response time over processor use.



Distributed system

- The two dominating factors as important needs in the field of computing / information access
    1. The need for physically distributed hardware
        - Distributed system
    2. The need for logically centralized software
        - Distributed operating system
- **Distributed system must have the following properties**
    1. Data are distributed
    2. Processing is distributed
    3. User are distributed.

DISTRIBUTED OS

- **A distributed system is a collection of independent computers that appears to its users as a single coherent system** – andrew tanenbaum
- **A distributed system is the one that prevents you from working because of the failure of a machine that you had never heard of** – verismo & rodrigues
- **A distributed system is a collection of independent computers that are used jointly to perform a single task or to provide a single service**.
  - Manages distributed systems, which contains several autonomous computers.
  - Jobs are distributed among processors accordingly to which computer can perform the job most effectively.
  - OS ensures fault tolerance and effective processor communication.
  - OS also manages resources sharing facilities
- Distributed system must have the following properties
  - User are distributed Processing are distributed Data are distributed

Advantage and disadvantage:

Advantage:

- **Data and resource sharing**: With resource sharing facility user at one site may be able to use the resources available at another.
- **Speed**: Speedup the exchange of data with one another via electronic mail. Reduction of delays in data processing
- **fault tolerant** If one site fails in a distributed system, the remaining sites can potentially continue operating.
- **Reliable and Economic**: Better service to the customers.
- Reduction of the load on the host computer.

Disadvantage are:

- **Software**: Needs complex software to manage the resources and processing.
- **Networking** : fast , reliable networking is needed every time and everywhere.
- **Security**: if sharing violation is occur, there is a chance of security britches.

REAL TIME OS

- Real-time systems are defined as those systems in which the correctness of the system depends not only on the **logical result** of computation, but also on the **time** at which the results are produced.
- It is an operating system **that supports real-time applications** by providing logically correct result within the deadline set by the user.
- Real time system are the systems that react instantly to the input without any time delay.
- The most critical thing in a real time system is the response time.
- Real time OS are designed to meet strict deadlines in such systems
    - **hard real time system (e.g. industrial process control, avionics, military)**
        - A hard real-time system has time-critical deadlines that must be met; otherwise a catastrophic system failure can occur
    - **soft real time system (multimedia systems, digital audio systems)**
        - A Soft real time system is one in which the performance of the system is only degraded but, not destroyed if the timing deadlines are not meet

MULTIPROCESSOR OS

- It is the type of operating system that refers to using two or more central processing units (CPU) in a single computer system.
- These multiprocessor systems or parallel operating systems are used to increase the computer system's efficiency.
- With the use of a multiprocessor system, they share computer bus, clock, memory and input or output device for concurrent execution of process or program and resource management in the CPU.
- connecting multiple CPU into a single system increases the computing power of the system
- special type of OS is required to handle these systems with special features for communication, connectivity and consistency

Parallel system:

- Parallel operating systems are a type of computer processing platform that **breaks large tasks into smaller pieces** that are done at the same time in different places and by different mechanisms. They are sometimes also described as “**multi-core” processors.**
- Parallel Processing 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 also known as tightly coupled systems.
- Parallel systems deal with the **simultaneous use of multiple computer resources** that can include a single computer with multiple processors, a number of computers connected by a network to form a parallel processing cluster or a combination of both.
- Parallel systems enables the system to achieve simultaneous data-processing tasks to increase the computational speed of a computer system, these **are more difficult to program than computers** with a single processor because the architecture of parallel computers varies accordingly and the processes of multiple CPUs must be coordinated and synchronized. Several models for connecting processors and memory modules exist, and each topology requires a different programming model

Thank you