Competency 5

Uses operating systems to manage the functionality of computers

5.1 Defines the term computer operating system (OS) and Investigates its need in computer systems.

Time: 4 periods

Learning Outcomes

- Defines the computer operating system
- Briefly describes the evolution of OS
- Identifies the main functions and the abstractions (directories, files and data) provided by the operating system to the user
- Describes how the operating system manages the resources of a computer.
- Classifies the operating systems based on their users and tasks

Why We Need OS?

- Not all computer systems need an Operating System.
- Example: Computer System in a washing machine, microwave oven (Embedded Systems) does not need an OS. Because these computer systems are not typical computers (Inside the machine there is no mouse, key board, and etc..).
  - For an example, the washing machine does one set of tasks (Get user inputs, get water, get detergent, drive motor, drain gray water)
  - Simple user interaction through a few buttons and indicator lights (on/off, fabric type-cotton, silk, nylon, solid level – heavy, moderate, light)
  - Never changing hardware/software (no change in functionality, no change of buttons and indicator lights)
  - Therefore we no need an OS and application programs can directly work with hardware.
- But, for an example, consider a desktop computer.
- If multiple different sets of tasks (programs)to perform, they have to be managed (order, allocation of resource, protection of tasks from other tasks)
- If users interact with more complicated ways, they should be provided with an interface that support complicated interactions (multiple key

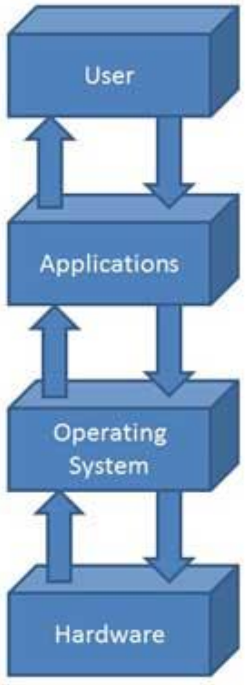
combinations, different mouse inputs such as click, double click, drag, different outputs such as numbers, letters, graphics)

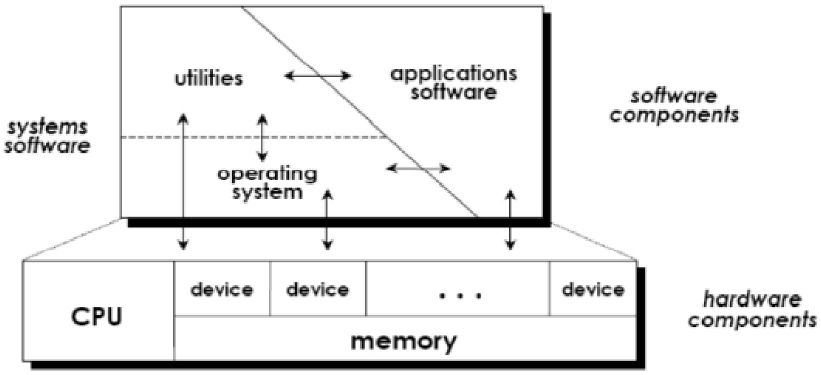
- If hardware / software change over time (update), changes have to be dealt with little or no user involvement
    - different devices (pen drive → potable hard disk),
    - network connections (Ethernet →Wi-Fi),
    - updated software (Office 2013 → Office 2016)

Computer operating system

- An OS is a program that controls the execution of application programs and acts as an interface between applications and the computer hardware.

- An operating system (OS) is a system software that provides a virtual machine (hides hardware details, provides an interface to applications and end users), manages computing resources (keeps track of resource usage, grants/revokes permissions for resources), and executes application software. (Teacher's Guide Definition)





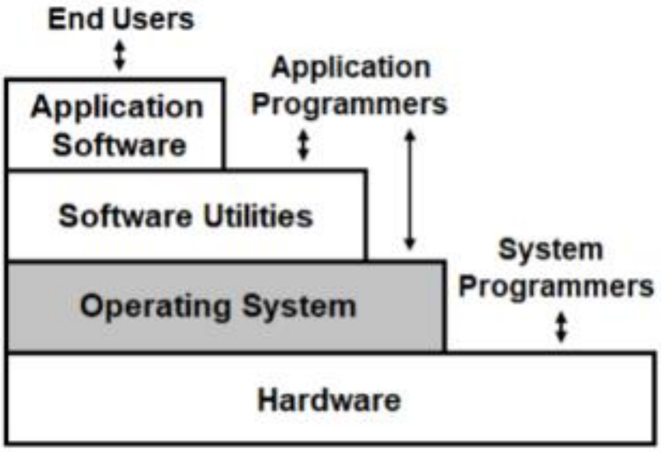
- Any OS **must play 3 essential** roles. Otherwise you can't say it is an OS.
    1. OS provides a **virtual machine,**
        - That **hides hardware details** .For example, we just do the drag and drop when we copy a file and no need to know the details such as how many bytes we need, what is the block size, data transfer rate etc. We can just see flying papers from here to there.

- It **provides a nice interface to application software and end users**. So we can easily do the job. For example, if you want to connect to the internet/network, no need to worry about the things such as turning on wireless adapters, data packets handling, IP address transformation etc.

2. OS **manages computing resources** (software and hardware)
   - Keeps track of resource usage
   - Grants/ revokes permissions for resources (protection of program from other programs) - For example, suppose that there is a particular region of memory that we won't allow access to program "X", because it already allowed to access program "Y". When program "Y" is over, then give permission to access program "X" to access that particular memory area.(granting revoking permission)

3. OS allow to **execute applications** (Application software runs on top of the OS).For example double click on the .exe file means that instruct the operating system.

How we interact with OS?



Evolution of OS (History of OS)

1. No OS (late 1940s – mid 1950s)
2. Simple batch system (mid 1950s)
3. Multi-programmed batch systems (1960s)
4. Time sharing systems (1960s)
5. Multi-programmed time sharing systems

1. **No OS (late 1940s – mid 1950s)**

- Used in the **1st generation of computers**.

- **Programmers (users) directly interacted with hardware** - very primitive computers were used, programs were loaded one bit at time using rows of mechanical switched called plug board, no programming languages (no even assembly language and only 0s and 1s used)

- **Serial Processing** (processed programs one after another) – program loading time is high. After loading the program then run the program. After finished it then load another.

- **Uniprogramming** (executed one program at a time)
- **Manual program scheduling** – We have to schedule manually which program execute first and next. Then Load that program (by switches on and off).
- One of the main problem is **the processor sat idle when loading programs and doing I/O**

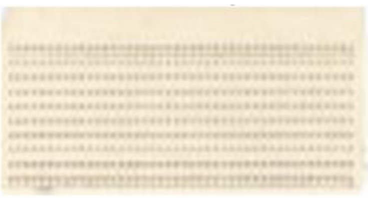
- **Followings are the few computers with NO-OS**

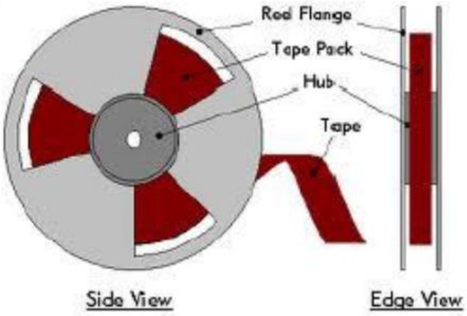
    - 1945: ENIAC, Moore School of Engineering, University of Pennsylvania.
    - 1949: EDSAC and EDVAC
    - 1949: BINAC - a successor to the ENIAC
    - 1951: UNIVAC by Remington
    - 1952: IBM 701
    - 1956: The interrupt
    - 1954-1957: FORTRAN was developed

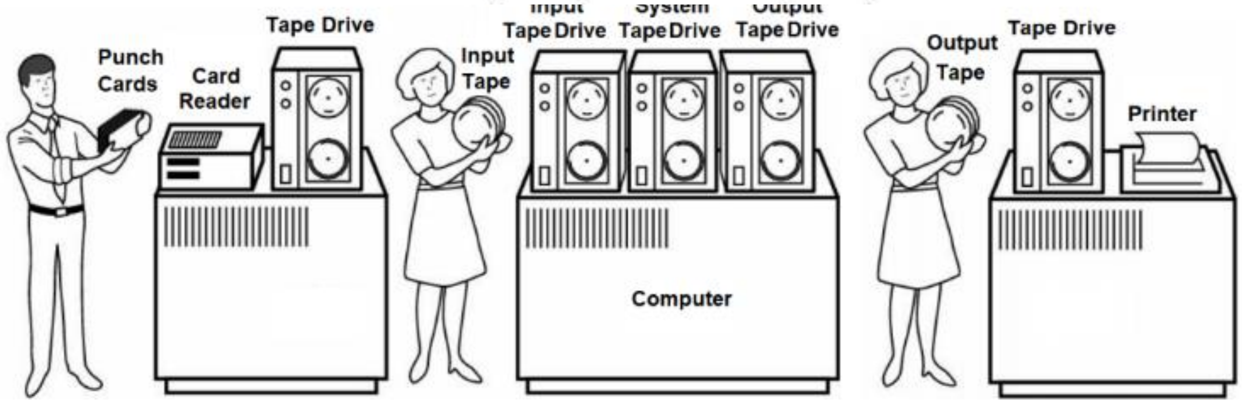
- The computer architects wanted to solve above problem and they wanted to maximize the processor utilization. To do that they introduced simple batch system.

2. Simple Batch System

- Used in **the 2nd generation of computers**.
- Introduced to maximize the processor utilization
- The first batch OS (and the first OS of any kind) was developed in the mid-1950s by General Motors for use on an IBM 701 mainframe computer.
- OS loaded and executed a new program from a batch of programs once the current program ended (completed or crashed) its execution.
- A batch of Programs were recorded in a magnetic tape with an inexpensive machine.
- When current program ended execution, its output was written to another tape and OS loaded next program.
- At the end of entire batch of programs, output tape was printed with an inexpensive machine.







Features of simple batch system

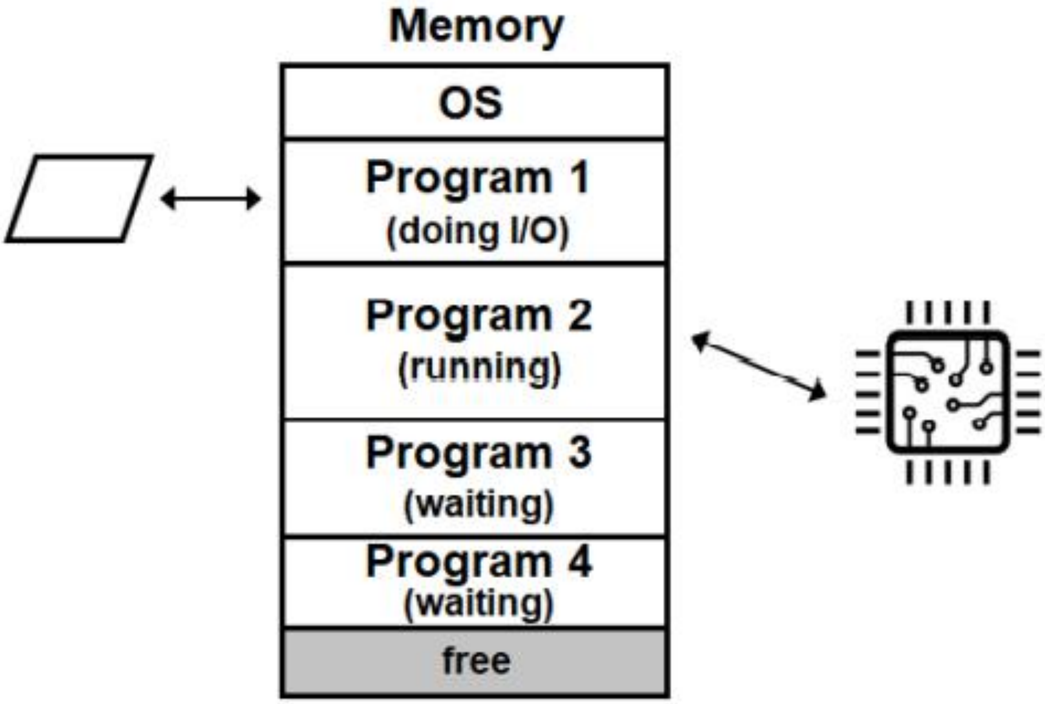
- No direct access to computer
- Serial processing (Processed program one after another)
- Uniprogramming (execute one program at a time)
- High response time (time between submission of request and receive of respond)
- Processor sat idle during I/O

Loading time problem is **solved**. Therefore the processor utilization is relatively high and much efficient.

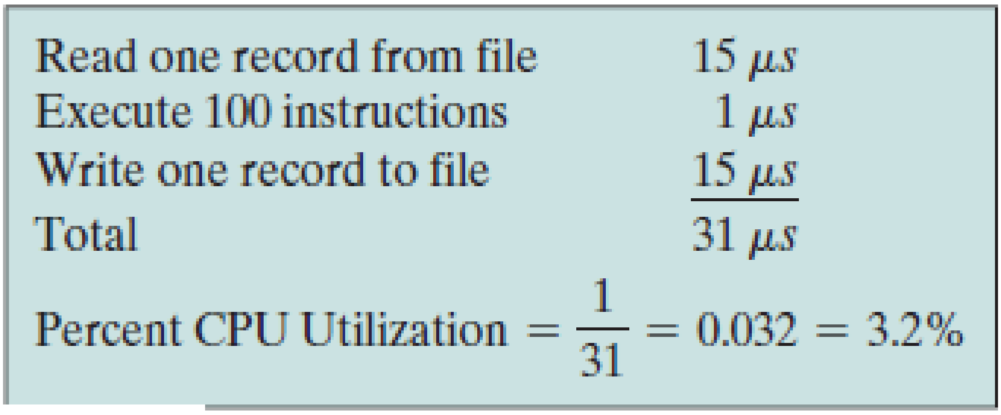
Now still remain the **problem is processor sat idle when doing I/O.** As a solution introduced **Multi-Programmed batch Systems.**

3. Multi-Programmed batch Systems

- Central theme of modern OS introduce in 3rd generation to minimize the processor idle time during I/O.
- Memory is partitioned to hold multiple programs.
- When current program waiting for I/O, OS switches processor to execute another program in memory.



- If memory is large enough to hold more programs, processor could keep 100% busy and can maximize the processor utilization.



Read one record from file	15 μs
Execute 100 instructions	1 μs
Write one record to file	15 μs
Total	31 μs

$$\text{Percent CPU Utilization} = \frac{1}{31} = 0.032 = 3.2\%$$

System Utilization Example

- Still there is a problem with high response time and introduced time sharing system as a solution.

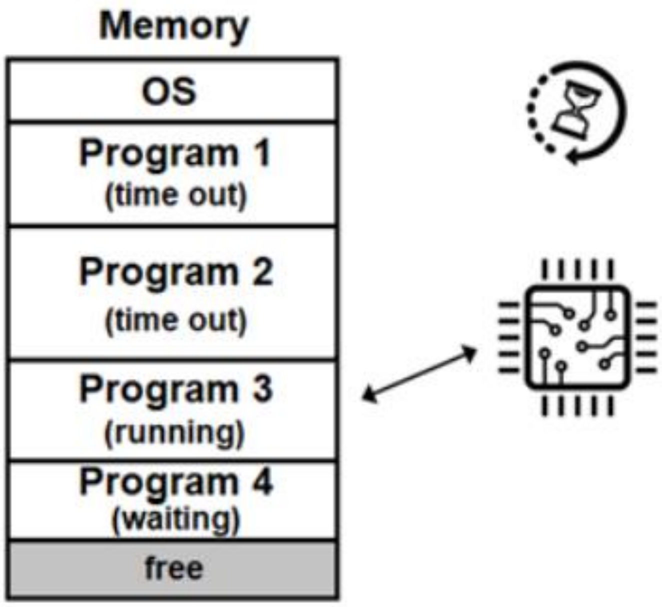
4. Time Sharing System

- Introduced to **minimize the response time and maximize the user interaction during program execution** in **3rd generation of computers**

- OS switches processor from one program to another after a certain time quantum (**context switching**)

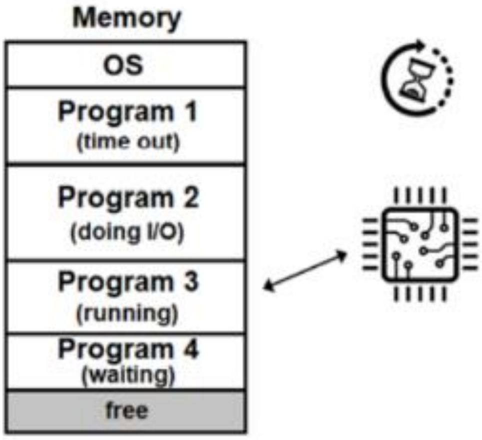
- Enables to share the processor time among multiple programs.

- By rapidly switching processor among programs, creates illusion of concurrent execution of multiple programs. But also add overhead to the system. OS has to do some works. It consume some resources such as computer time, CPU and other resources. It is an extra overhead for the CPU. This is the idea behind the **multi-tasking.**



5. Multi Programming Time Sharing Systems

- Used in **Modern OS.**
- Memory is partitioned to hold multiple programs.



- OS switches processor from one program to another after a certain time quantum.
- When current program wants to do I/O, OS switches processor to execute another program in memory.
- Maximize processor utilization as well as user interaction.

Main functions of OS?

- Operating systems are among the most complex pieces of software ever developed.
- This reflects the challenge of trying to meet the difficult and in some cases competing objectives of convenience, efficiency, and ability to evolve.
- Proposes that there have been five major theoretical advances in the development of operating systems:

1. File Management
2. Process Management
3. Resource Management (Memory, I/O device, Storage)
4. Providing User Interfaces
5. Security and Protection

Classification of Operating Systems

There are different types of Operating Systems based on the users and tasks.

Different types of Operating Systems (Based on the users)

1. Single user - Facilitates single user to use the system at a time

2. Multi User-Facilitates multiple users to use the system at a time

Different types of Operating Systems (Based on Number of tasks)

1. Single Task – Executes only one program at a time

2. Multi Task - Executes multiple programs at a time

Single user-single task – A single task is performed by one user at a time.

Eg.

- For Palm handheld computers (Personal Digital Assistant - PDAs)

  - Pocket PC

  - Palm OS

  - Windows Mobile

- Ms-DOS

Single user-Multi task - Several programs are run at the same time by a single user. For an example Windows user to be writing a note in a word processor while downloading a file from the Internet while printing the text of an e-mail message.

Eg:

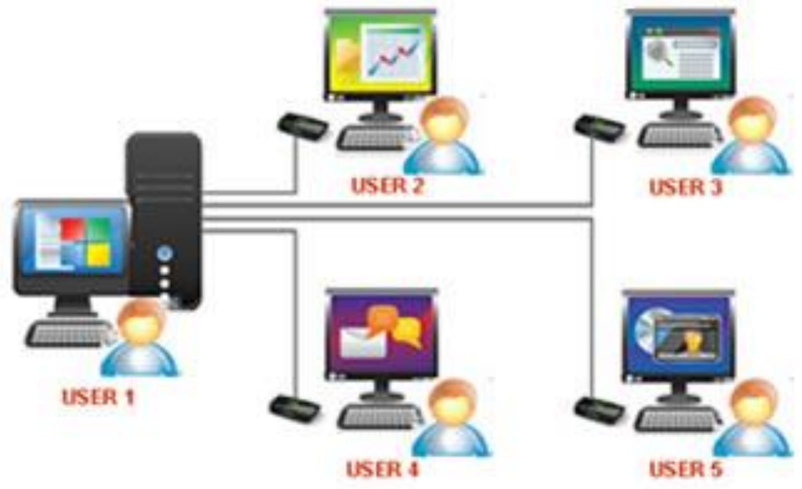
- Microsoft's Windows

- Apple's MacOS platforms

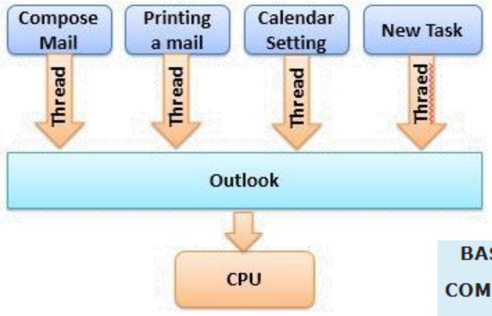
Multi user-Multi task – A multi-user operating system has been designed for more than one user to access the computer at the same or different time.

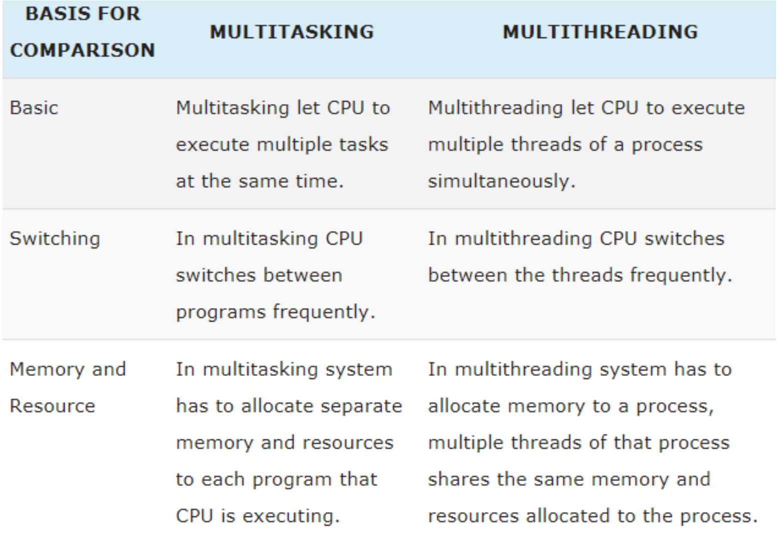
Eg:

- Unix server



Multi-threading – A thread is also called a sub process. Threads provide a way to improve application performance through the parallel execution of sub process.





BASIS FOR COMPARISON	MULTITASKING	MULTITHREADING
Basic	Multitasking let CPU to execute multiple tasks at the same time.	Multithreading let CPU to execute multiple threads of a process simultaneously.
Switching	In multitasking CPU switches between programs frequently.	In multithreading CPU switches between the threads frequently.
Memory and Resource	In multitasking system has to allocate separate memory and resources to each program that CPU is executing.	In multithreading system has to allocate memory to a process, multiple threads of that process shares the same memory and resources allocated to the process.

Real Time –

- OS is designed to run applications with very precise timing and with a high degree of reliability.

- The main objective of real-time operating systems is their quick and predictable response to events.

- These types of OS are needed in situations where downtime is costly or a program delay could cause a safety hazard.

- Eg: used when there are time requirements are very strict like missile systems, air traffic control systems, robots, Air bag control in cars, anti-lock brake, engine control system, scientific experiments, medical imaging systems, industrial control systems, weapon systems etc…

- A general purpose operating system was designed to handle multiple tasks with no time limit, we cannot say in a certain time a task will get happened.

- An example is form your PC if you copy some data from one device to another device it may take several minutes or more, we cannot predict responsiveness of the system of course, due to the tasks running parallel at that time.

- But, a real time embedded system can give an accurate output at right time that means, it is time critical no delay is encouraged for real time systems and if any delay occurs it may lead to catastrophic effects.

- For example Airbag control system in a car. If you drive a car at a high speed accidents may happen, in such case airbag opens and saves your life.

Time Sharing Systems – Processor's time is shared among multiple users/applications

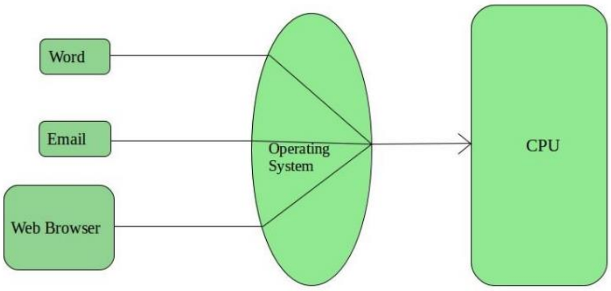
E.g: Multics, Unix etc.

Features of Timesharing operating systems are as follows:

- Provides quick response
- Reduces CPU idle time

Advantages of Time-Sharing OS:

- Each task gets an equal opportunity
- Less chances of duplication of software
- CPU idle time can be reduced



Answer the following Questions

1. What is an Operating System?
2. What is the relationship between operating systems and computer hardware?
3. What inconveniences that a user can face while interacting with a computer system, which is without an operating system?
4. Briefly explain the evolution of computer OS?
5. What are the main functions of OS?
6. What are the types of OS? Explain with examples.
7. What are the advantages of multiprogramming**?**

References

Teachers Guide – 2017

Chapter 3 and 9, W.Stallings, Operating System: Internals and Design Principles, 9th Edition, Pearson, 2017