Day 1

> Date ⇒ 27 aug 2024.

💡 OS Content Discussed Day-1

- What is OS;
- How is it different from other application software;
- Why is it hardware dependent?
- Different components of OS
- Basic computer organization required for OS.
- Examples of well-known OS including mobile OS, embedded system OS, Real Time OS, desktop OS, server machine OS etc. ;
- How are these different from each other and why
- Briefly discussed Functions of OS
- Introduction to User and Kernel space and mode
- Linux OS and its features
- Working basics of file system of Linux
- Commands associated with files/directories
- other basic commands
- What are file permissions and how to set them
- Giving and revoking permissions using chmod for Owner, Group and Other users

Lecture One :

- Which manages all the hardware resources of computer ⇒ **Resource manager**
- Computer which manages and supervise all the processes task ⇒ **Process manager.**
- OS is code which has eyes on execution unit , what will be done , what needs to be done , OS will manage that , it process all the execution happening in computer machine .
- How memory or hardware is interacting with each other like keypress , mouse , etc ⇒
- All the process going on the computer is being monitored by OS.
- Generally it is called as system software , now a days called as interface between user and hardware .
- **Os acts like interface between user and hardware , it takes input from user and give the desired output .**
- OS manage all the input and process it and give output , OS always take help of hardware .
- Before operating system , there used to be no controller for hardware or pre installed software to work on data , computation was done mathematics and punch cards were used ,
- That time we have to give the code for the data to get process at the same time we have to write the steps to execute that , example we have to write hello world program as well as JVM code also .

- Punch card consist of whole from where the electricity was passed , if passed it coded as 1 and did not then 0 , according to the combination the processing starts .
- The output in these punching machine may take one hour to one month .
- Process and hardware were managed by user only , everything was done manually .
- Now the operating does this , the efficiency has improved which means throughput has increased.
- Hardware which still convert input and output like Ram and storage unit hard drive ⇒ Hardware is still the BOSS , hardware still responsible for executing things .
- **OS is a software which is totally dependent on hardware (Because code has to be install somewhere i.e on hardware)**
- Now everything is done in nano secs , millisecs from months to nano secs .
- **RAM is volatile memory which has higher speed than hard drive , so that task can be executed in very high speed .**
- **Intermediate result when in execution gets stored in Register and also cache memory , which is attached to motherboard .**
- OS is installed on hardware and application software are installed on OS .
- The user interacts with application software through an abstraction layer on the application's interface. The application then gives instructions to the OS, which in turn communicates with the hardware.
- Hardware doesn't work directly with the operating system. Instead, there's a crucial component: the kernel. The hardware listens to and follows instructions from the kernel.

▼ **Why OS are hardware dependent ?**

Ans ⇒ It has to be stored in somewhere in physical memory

It requires some physical memory for execution of process and execute itself also as OS is the first program which get loaded in physical memory .

Os is processing it may need some speed memory to store the intermediate result(memory addressed or future task , calculation , anything which may be needed in future) .

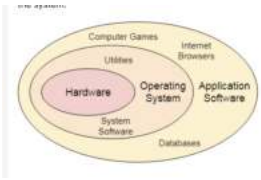
The ouput generated will also need an hardware to display the output like monitor , speaker ,etc.

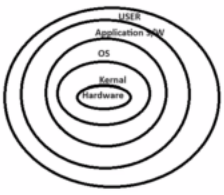
Register and cache is a physical memory , they are faster than cache but does not have memory just 32 bits .

Cache memory are more expensive and has 2 mb or in mb for 8K. It is designed to increase the speed.

In mobile cache is dumb data which can be removed when required , not same as the computer as a memory .

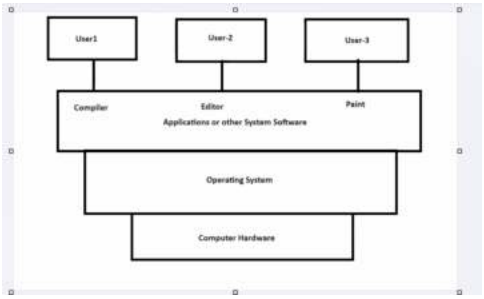
```
### OS Notes Day-1 Date: 27-08-2024
#### Session-1: Introduction to Operating System
- What is OS?
- How is it different from other application software?
- Why is it hardware dependent?
- Different components of OS
- Basic computer organization required for OS.
- Examples of well-known OS
    1. Mobile OS
    2. Embedded System OS
    3. Real Time OS
    4. Desktop OS
    5. Server machine OS etc.
- How are these different from each other and why?
- Functions of OS
- User and Kernel space and mode; Interrupts and system calls
#### Session-2: Introduction to Linux
- Working with basics file system of Linux
- Commands associated with files/directories
- Other basic commands.
- Ref: https://ubuntu.com/tutorials/command-line-for-beginners#1-overview
- Operators like Redirection (>), Pipe (|)
- What are file permissions and how to set them?
```



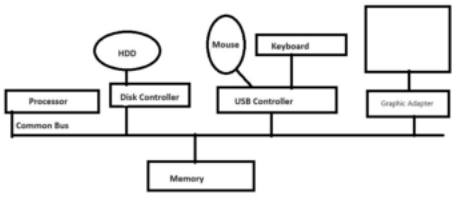


> What is kernel ⇒ Kernel is the main person , the units will listen to only the kernel . OS always instructs the kernel and then kernel which do that in hardware . In the realm of computing, the kernel is the core, most fundamental part of an operating system. Think of it as the bridge between your computer's hardware and the software applications you run. Kernel stays in RAM , but sometimes it depend 80% present or accordance to memory available .

Components of OS



- Basic computer organization required for OS.



- Bios/UEFI/firmware is a Rom structure which is attached to computer .
- Bios is Basic input and output system . It has the actual of bootloader .
- Bootloader loads the OS from hardware to the main memory .
- **Sector zero** , when bios test that every device is ready now os can be loaded , it trugger sector and it ask bootloader to load os , then bootloader loads into the memory .

Booting :

1. Cold booting : When we load pour OS into Ram first time that is cold booting .
2. Hot Booting : When we restart the computer , OS is already loaded , just we need to reload it again .

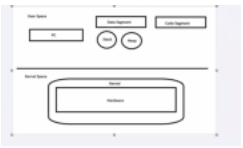
```
- Examples of well-known OS
    1. Mobile OS: Android, iOS, Windows
    2. Embedded System OS:
    3. Real Time OS: HRT, SRT
    4. Desktop OS: Personal Computer
    5. Server machine OS etc.
```

Function Of OS :

1. **Process management** ⇒ Process creator , Process execution Process Termination and process algos .
2. **Memory management(RAM)**
3. **Device management** ⇒ HDD , printer , monitor , speaker , webcam , mouse .
4. **Context Switching** ⇒ When the process switched to other processes when the one process is over , it switches immediately to last running process .
5. **Disc management** ⇒ Disc scheduling Algos .
6. **Network management** ⇒ NIC n all .
7. **File management**
8. **Security management ⇒ Firewall , anti virus , Anti spyware .**

User Space and kernel Space : Disrupt and System Call

- Hardware is being processed by kernel .
  - User Space : Every process is taking which memory stack heap , etc
  - Kernel Space : Every machine which need IO devices it give control to kernel .



▼ **User Space vs. Kernel Space: The Fundamental Divide**

In modern operating systems, memory is logically divided into two distinct areas:

- **User Space:**
  - This is where most applications, user processes, and libraries reside.
  - Processes running in user space have limited access to system resources.
  - They cannot directly interact with hardware or critical system data structures. This restriction is vital for system stability and security.
- **Kernel Space:**
  - This is the privileged area where the core of the operating system, the kernel, resides.
  - The kernel has full access to all system resources, including hardware, memory, and CPU.
  - It handles critical tasks such as process scheduling, memory management, device drivers, and system calls.

Why the Separation?

The separation between user space and kernel space serves a few crucial purposes:

- **Protection & Stability:**
  - Prevents user-level processes from accidentally or maliciously corrupting the kernel or other processes.
  - Ensures that a misbehaving application doesn't bring down the entire system.
- **Security:**
  - Restricts access to sensitive hardware and data structures to privileged kernel code.
  - Makes it harder for malware to gain control of the system.
- **Abstraction:**
  - Provides a clean interface (system calls) for user-level processes to interact with the kernel.
  - Shields applications from the complexities of the underlying hardware.

How They Interact: System Calls

When a user-level process needs to perform a privileged operation (like accessing a file on disk or sending data over the network), it makes a **system call**.

1. The process triggers a software interrupt.
2. The CPU switches from user mode to kernel mode.
3. The kernel executes the requested system call on behalf of the process.
4. The CPU switches back to user mode and returns control to the user process.

This controlled transition between user space and kernel space ensures that the system remains secure and stable.

Key Points to Remember:

- User space and kernel space are distinct memory regions with different levels of privilege.
- User processes run in user space, while the kernel runs in kernel space.
- System calls provide a controlled way for user processes to request services from the kernel.
- This separation is critical for system stability, security, and abstraction.

Feel free to ask if you would like to delve deeper into any particular aspect!

Types of Memory :

1. **Primary Memory :**
   a. **RAM :**
      i. SRAM
      ii. DRAM
   b. **ROM** :
      i. EROM
      ii. EPROM : used today
      iii. EEPROM: used today
2. **Secondary Memory :**
   a. Hard Drive Disc
   b. Magnetic Disc
   c. USB Pen Drive
   d. Optical disc
   e. Floppy
   f. CD's

Preference in order of Speed and Cost

Register (Max)
Cache
RAM
HDD
External Devices (Min)

Preference in order of Space

Register (Min)
Cache
RAM
HDD
External Devices (Max)

Volatile Memory

Register (A, B,C,D,E,F)
Cache
RAM

Types of Operating System :

1. **Batch OS ⇒**
    a. Executes batches of jobs with similar type of output Once a batch of tasks is submitted, the OS processes them one by one.
    b. It's not modern day operating system .
2. **Multiprogramming / multitasking OS**
    a. Which can run multi-program at the same time by context switching with one processing element ⇒ they have only one processor .
3. **Multiprocessing OS ⇒**
    a. In this , we have more than one processor means CPU's but they are working on common physical memory , when we don't want context switching(it depends on OS ⇒ Hard Real Time and soft real time), when we want high performance.
    b. It is very aware of managing load .
    c. Multiple process will execute together .
    d. Throughput : in one min how many processes are being executed completely.
    e. **Symmetric** ⇒ multiple process with one memory and **Asymmetric** ⇒ multiple process with 2 to 3 memory .
4. **Distributed OS :** Distributed over a network .
5. **Desktop OS :**
6. **Server OS**

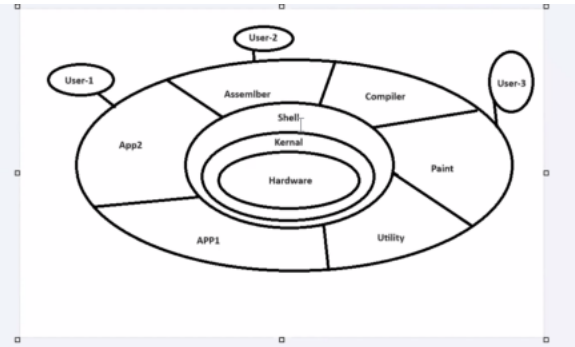
💡 **Context switching** ⇒ when context is switched , then it have to same the current context somewhere it stores in RAM if space is available but if not it takes space in HDD as a swap area .

Linux

1. Linus Torvalds Founder since 1991 .
2. Linux is a family of open-source Unix-like operating systems based on the Linux kernel. It is one of the most popular and widely used operating systems, especially in server environments, cloud computing, embedded systems, and increasingly in personal computing.
3. Hardware ⇒ Kernel ⇒ shell ⇒ user .
4. There is OS , we are just dont interacting with shell directly to kernel .
5. Open source ⇒ free ad user can modify it .
6. No Cost
7. Multi Tasking
8. Security
9. Scalable
10. Networking = http only
11. Shell CLI as well as GUI .

Better file system ⇒ Easy to maintain

```
- Working with basics file system of Linux
- / is root directory
    1. /bin: User Bineries
    2. /sbin: System Bineries
    3. /etc: Configuration Files
    4. /dev: Device Files
    5. /proc: Process Information
    6. /var: Variables Files
    7. /tmp: Temporary Files
    8. /usr: User Programs
    9. /home: Parent directory of user friendly directory
    10. /boot: Boot Loader Files
    11. /opt:
    12. /lib
```




```
- - -- - - - - - - - - - - - - - - - - - - - - - - - - - - - -
there are permission shown including 10 dashes = > -* - r -* - w -* - e -
- the first  3 dashes group from  and leave first , are for the owner permission and created the file  :   1. r : read
                                                                                                          2 . w : write
```

3 . e : execute

- the second 3 dashes group from and leave first , are for the owner permission : User in the same operating system
- the third 3 dashes group from and leave first , are for the owner permission : Remnaining other users/sources any location
Username : visible =>
Password : non-visible =>

- - -- -

1. pwd command : Present working directory => current working directory
/home/folder_name

2. nano cmd => editor cmd and open it
nano abc.txt
create or open a file .

3. ls : list out all of the file and directory of current working directory
H.W =>   a . ls -l =>

4. touch cmd => create new file only
touch file1.txt

5. mkdir => create a new directory
mkdir Day-1

6. whoami => to know the name

u => owner user , g => group , a => all others

7. chmod => To give permission
         chmod a+wx
         chmod o+wx
8. rm > delete file
9 . cd Day-1/ => move one step ahead in directory
10. cd .. => move one step backward
11. cd shortform of direct like aditi = ad , directly moves into the directory

12. sub directory has to be removed first when want to dlete the folder

13. rm -r directory_name => delete the directory with sub-directory .
14. rmdir => delete the single directory .

15. sudo adduser user_name
=> sudo means now the owner is doing the stuff so all the sudo permission given , it is greatest priviledge .
=> adduser cant be run alone , sude permission are required

16. chown cmd => Change owner cmd
chown user2_name file_name OR
sudo chown user2_name file_name

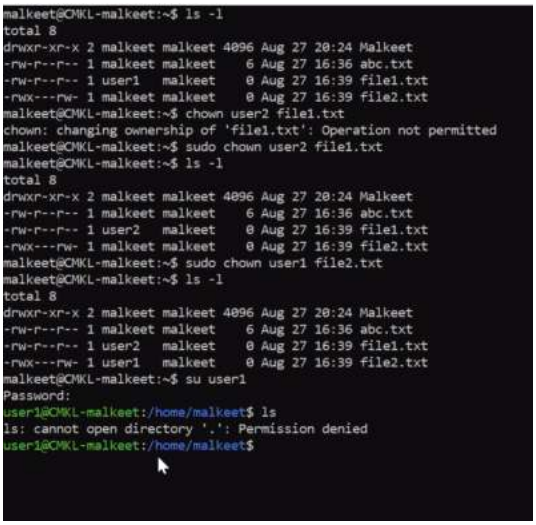
17. To give sudo permissions to any user we have to do confg, changes we have to go to etc
cd/etc/
inside which sudo ers was the file . and inside that we can give permissions .
we have to read file using nano

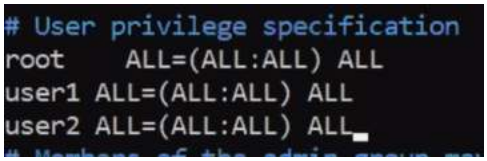
18 . su cmd is used ofr switching user ,
=> su user_name , then type the password .

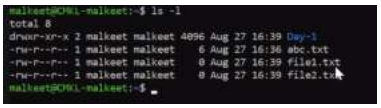
****
 Very More imp cmds :
 1 . To dislay the content of the files , we use cat cmd OR we want to create a new file and write in console itself => cat file
 => cat file_name
 2 . only display first 5 line
 => head -5 file_name
 3 . Beneath 5 line
 => tail -5 file_name
 4 . head file_name = prints only uptill 10 line by default , we can change it also . (ASKED IN MCQ)

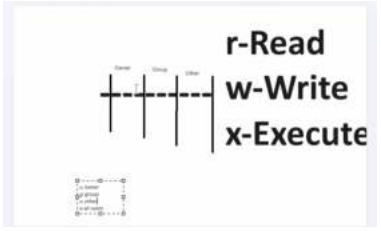
```
=> whole content
5 . tail cmd will take last 10 lines
```

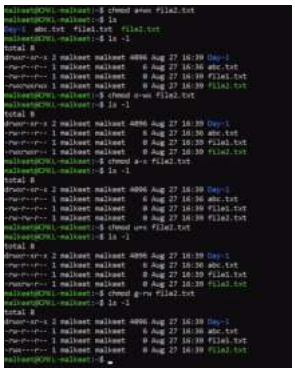
- Ref: https://help.ubuntu.com/community/FilePermissions
- chown: to change owner of the file
- su: to switch the user from current to any specified user
- cat: to display content of the file on console
- head: to display top n lines of the file on console. By default it will print first 10 lines
- tail: to display bottom n lines of the file on console. By default it will print last 10 lines
- adduser: to add new user into the system
- sudo: to give some specific privilages to the user's other than root.











- They can't be any root directory only root users .
- We can check logs to see who has accessed the sudo permission . Refer to log folder
- Beware of sudo command and rm cmd , they can be dangerous and dnt delete system files.

💡 Asked in interview ⇒ What directory contains which files etc (IMP).

1. Refer to recordings for various func of cat cmd .

Redirection :

> I want to redirect the output of one cmd is feeded as a input for the other cmd .

1 . First we want the output of one cmd and redirection .

2. We want to append that , meaning ⇒ we want to modify the file not overwrite .
3. Try out the variations for this redirections .
4. This is Operator used ⇒ (>)

Piping :

Cmd Which we should know :

Topic: Recording
Module: COS
Date: 27/08/24
Session: Afternoon

https://us02web.zoom.us/rec/share/tUrNjxd8CKvFdedy0-MLidoxALTA-U3PxekPWcqul59Ix8kpFNAkZdSf_eidv0yK.5c3owdOvZYOfESbi