

Linux Workshop

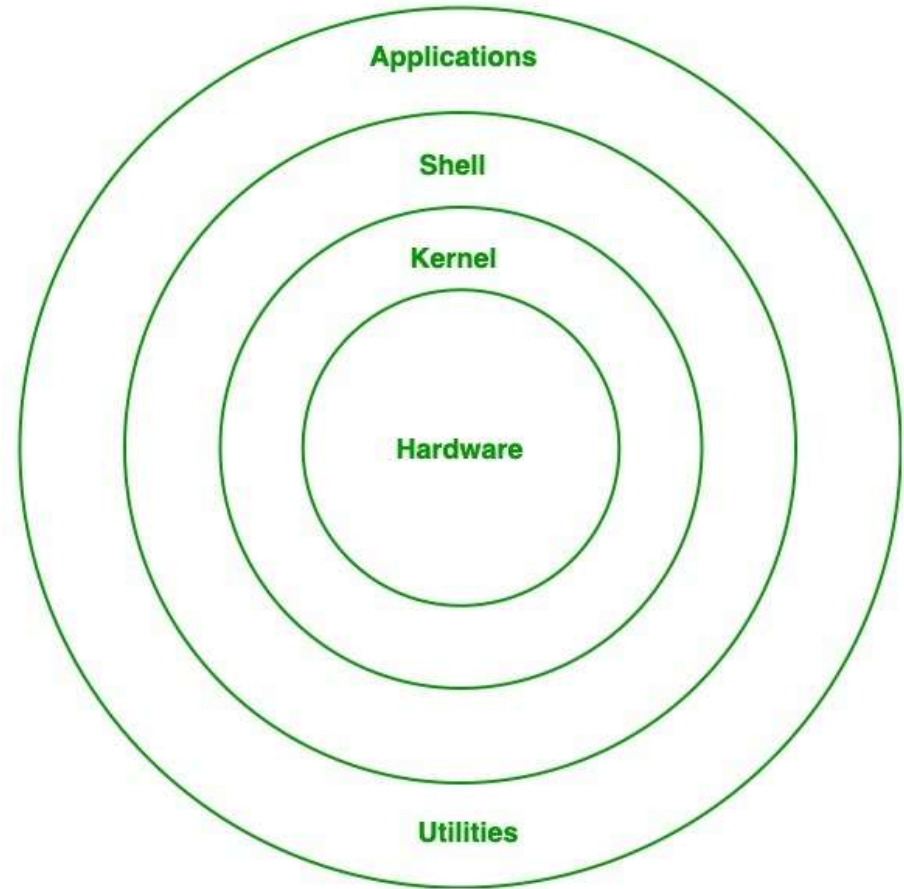
CSI

What is Linux

- A family of Free Operating Systems
- Available in various flavors, each suitable for a particular type of application
- (Kali for Penetration Testing, Red Hat for Enterprise Workload, Ubuntu for Beginners)
- Ubuntu is a Debian-Based OS
- Some Linux-Specific Features-
 - ❑ Live CD/USB: Almost all Linux distros provide live CD/USB so that users can run/try it without installing it.
 - ❑ Open Source: Linux code is freely available to all and is a community based development project.

Linux Architecture

- **Hardware Layer:** This layer consists all peripheral devices like RAM/ HDD/ CPU etc.
- **Kernel:** Kernel is the core of the Linux based operating system. It virtualizes the common hardware resources of the computer to provide each process with its virtual resources. This makes the process seem as if it is the sole process running on the machine. The kernel is also responsible for preventing and mitigating conflicts between different processes.
- **System Library:** Is the special types of functions that are used to implement the functionality of the operating system.
- **Shell:** It is an interface to the kernel which hides the complexity of the kernel's functions from the users. It takes commands from the user and executes the kernel's functions.
- **System Utility:** It provides the functionalities of an operating system to the user.



Linux File System

- In Linux, the file system creates a tree structure.
- Linux treats everything as a file. Including directories.
- Features:
 - Linux does not use the backslash (\) to separate the components; it uses forward slash (/) as an alternative.
 - In Linux, a file may have the extension '.txt,' but it is not necessary that a file should have a file extension.
 - Hidden files in Linux are represented by a dot (.) before the file name (ex: .bash_history).

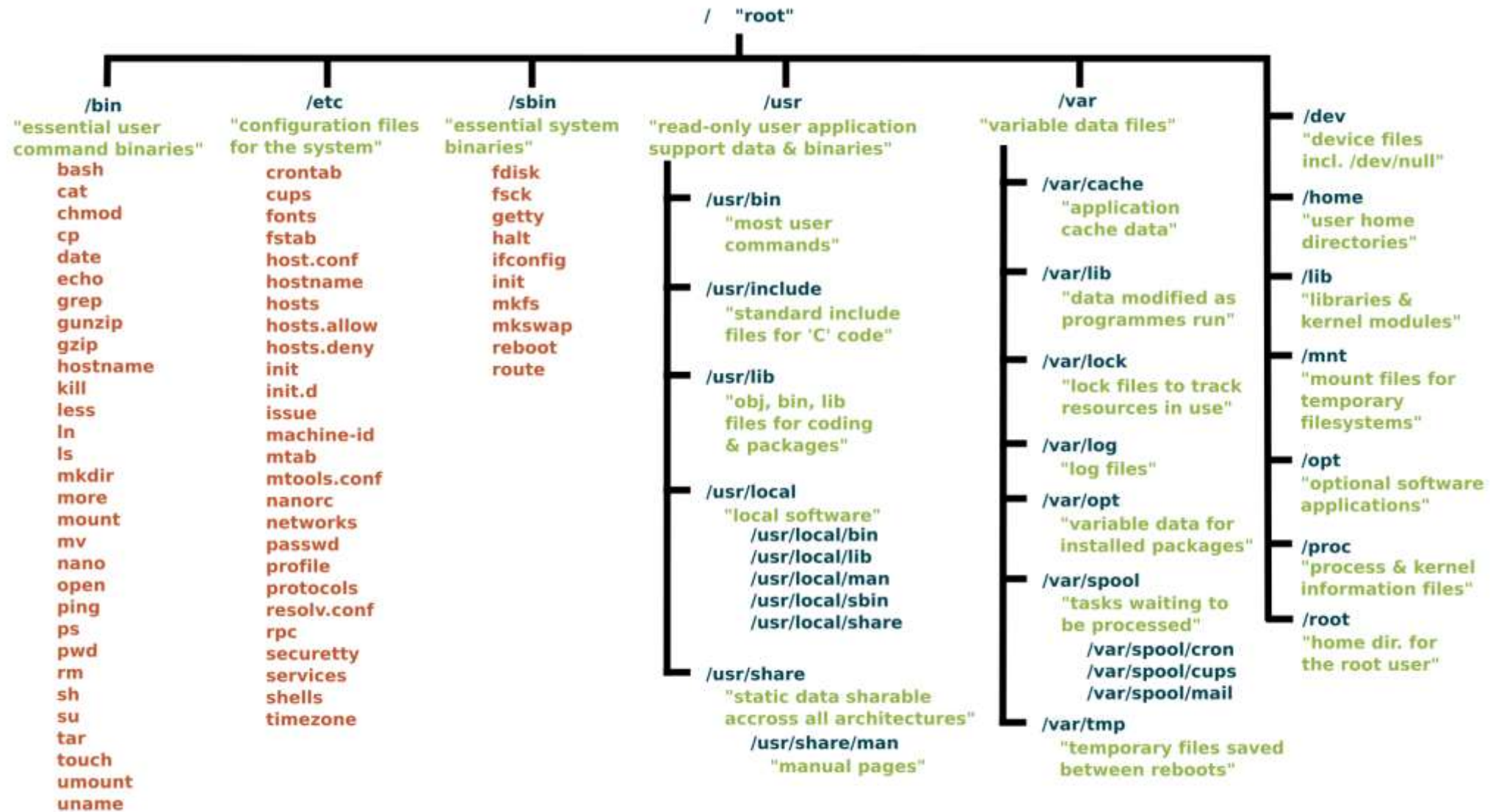
Special Directories

- The entire Linux directory structure starting at the top (/) root directory.
- The root filesystem is the top-level directory of the filesystem.
- It must contain all of the files required to boot the Linux system before other filesystems are mounted.
- It must include all of the required executables and libraries required to boot the remaining filesystems.
- The /root/ folder (aforementioned root “directory” is just /) acts as the /home/ for the superuser (su)
- /boot: Contains the static bootloader and kernel executable and configuration files required to boot a Linux computer.

(continued)

- /tmp Temporary directory. Used by the operating system and many programs to store temporary files.
- /var Variable data files are stored here. This can include things like log files, MySQL, and other database files, web server data files, email inboxes, and much more.
- /usr These are shareable, read-only files, including executable binaries and libraries, man files, and other types of documentation.
- /etc Contains the local system configuration files for the host computer.

```
(kali㉿kali)-[~]  
$ tree / -d -L 1  
/  
├── bin → usr/bin  
├── boot  
├── dev  
├── etc  
├── home  
├── lib → usr/lib  
├── lib32 → usr/lib32  
├── lib64 → usr/lib64  
├── libx32 → usr/libx32  
├── lost+found  
├── media  
├── mnt  
├── opt  
├── proc  
├── root  
├── run  
├── sbin → usr/sbin  
├── srv  
├── sys  
├── tmp  
├── usr  
└── var  
  
22 directories
```



iNodes (not-really-from-Apple)

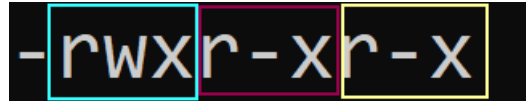
- Every Linux file or directory has an inode, and this inode contains all of the file's metadata.
- Every Node on the File tree has an unique inode number
- Inode number is also known as index number.
- An inode is a unique number assigned to files and directories while it is created. The inode number will be unique to entire filesystem.

```
(kali@kali)-[~]
$ stat a.txt
File: a.txt
Size: 0          Blocks: 0          IO Block: 4096   regular empty file
Device: 801h/2049d Inode: 2105422   Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/   kali)   Gid: ( 1000/   kali)
Access: 2022-02-19 11:57:52.465731828 -0500
Modify: 2022-02-19 11:57:52.465731828 -0500
Change: 2022-02-19 11:57:52.465731828 -0500
Birth: 2022-02-19 11:57:52.465731828 -0500
```

```
(kali@kali)-[~]
$ df -i
Filesystem      Inodes   IUsed   IFree IUse% Mounted on
udev            242057    394    241663    1% /dev
tmpfs           252696    684    252012    1% /run
/dev/sda1       5185536  372602  4812934    8% /
tmpfs           252696     1    252695    1% /dev/shm
tmpfs           252696     3    252693    1% /run/lock
tmpfs           50539     78    50461    1% /run/user/1000
```

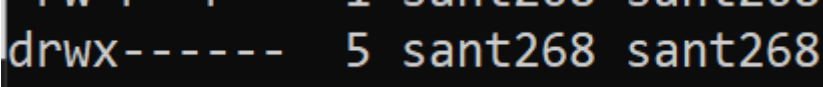

File Permissions

- There are three characters: x for Executable, R for Read and W for Write
- These permissions are part of the file metadata (Even directories have permissions)



-rwxr-xr-x

{file permissions} {number of hardlinks} owner group



```
drwx----- 5 sant268 sant268
```

(ls -l will display this)

First character is reserved for Special bits (here d is for directory)
(similarly “l” represents File Links, and “t”/“u” are SUID/SGID/Sticky bits)

- The Cyan Box has permissions for the Owner of that particular file
- The Red Box has permissions for the Group that owner-user is a part of
- The Yellow Box is the permissions given to everyone

Using Binary to Set Permissions

- The first number represents the Owner permission; the second represents the Group permissions; and the last number represents the permissions for all other users. The numbers are a binary representation of the rwx string.
- $r = 4$
- $w = 2$
- $x = 1$
- Add the Numbers and assign it using chmod
- So- `chmod 740 <file>` would give RWX (4+2+1) to the Owner, only Read Permissions to the group and 0 (No) Permission for the other users on system