

# 23MCAC103- ADVANCED OPERATING SYSTEMS

by  
Dr.Gobi Natesan

# Course Outcome –Unit 2

---

- **Apply** the Linux commands for manage the files to perform the basic operations of File Systems.

# Syllabus- Unit 2

---

## **Open Source Operating System:**

Linux System: GUIs-Command Line-Virtual Machines-Unix and Linux-Types of Users-Linux Commands- Man Pages-Bash Features- Interpreters- File Specifications File System Commands-File System Structure.

# Definition & Naming of Linux Operating System

- The **Linux** open source **operating system**, or **Linux OS**, is a freely distributable, cross-platform **operating system** based on Unix that can be installed on PCs, laptops, notebooks, mobile and tablet devices, video game consoles, servers, supercomputers and more.
- It was developed by Linus Torvalds.

# History of Linux Operating System

- UNIX: 1969 Thompson & Ritchie AT&T Bell Labs.
- Commercial Vendors: Sun, HP, IBM, SGI, DEC.
- GNU: 1984 Richard Stallman, FSF.

Open Source: GPL.

# Components of Linux System

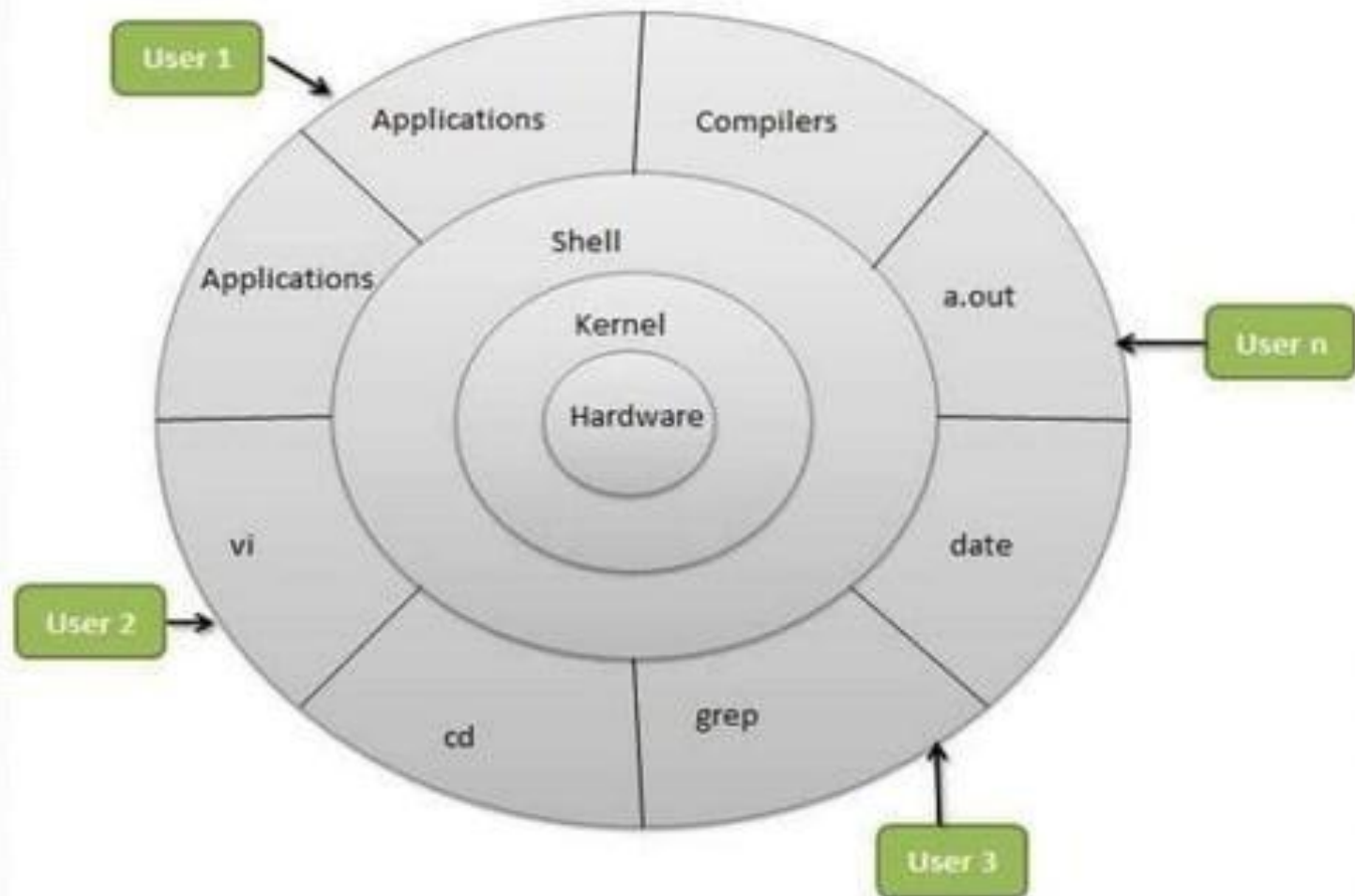
- a) **Kernel** – Kernel is the core part of Linux. It is responsible for all major activities of this operating system.
- b) **System Library** – System libraries are special functions or programs using which application programs or system utilities accesses Kernel's features
- c) **System Utility** – System Utility programs are responsible to do specialized, individual level tasks.

# Basic Features of Linux

- a) **Portable** – Portability means software can work on different types of hardware in the same way.
- b) **Open Source** – Linux source code is freely available and it is a community based development project.
- c) **Multiprogramming** – Linux is a multiprogramming system means multiple applications can run at the same time.
- d) **Security** – Linux provides user security using authentication features like password protection/ controlled access to specific files/ encryption of data.



# Architecture of Linux Operating System





The architecture of Linux contains some layers :

- a) **Hardware layer** – Hardware consists of all peripheral devices (RAM/ HDD/ CPU etc).
- b) **Kernel** – It is the core component of Operating System, interacts directly with hardware, provides low level services to upper layer components.
- c) **Shell** – It takes commands from the user and executes kernel's functions.
- d) **Utilities** – Utility programs that provide the user most of the functionalities of an operating systems.

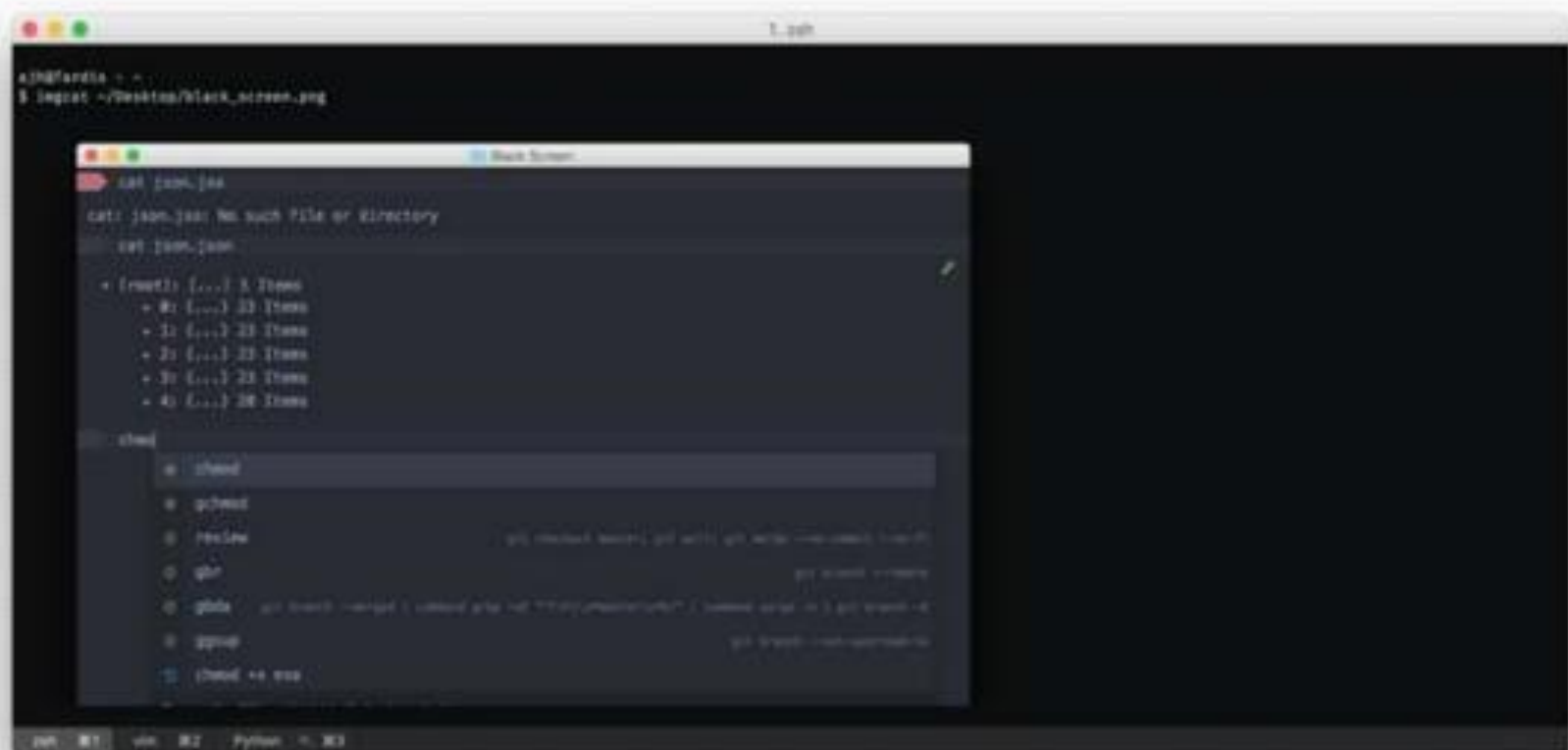
# About Kernel

- It interacts with the actual hardware in machine language.
- It is the monarch who has overall control of everything.
- It has various functions such as file management , data transfer between file system and hardware, memory management, scheduling of various programs in the memory, interrupts issues.

# User-Interface of Linux

- The user interface is either a command line interface (CLI), a graphical user interface (GUI), or through controls are associated with hardware.
- CLI shells are text based user interfaces, which use text for both input and output.
- On desktop systems, the most popular user interfaces are the GUI shells.
- Most popular user interfaces are based on the X Window System, often simply called "X".

## CLI view of Linux





# Desktop View of Linux



# Inside view of Linux





# Distribution of Linux

- Corel Linux
- Debian GNU/Linux
- OpenLinux (Caldera)
- Red Hat
- Ubuntu
- TurboLinux

# Software application for Linux

- OpenOffice: word processing, spreadsheets, drawing
- Adobe Acrobat Reader
- Konqueror: The KDE File Manager and Web Browser
- TV, Video, Radio, and Webcam

# Editors of Linux

- There are some editors in Linux
  - a) Vi/Vm editor
  - b) Gedit editor
  - c) Nano editor
  - d) GNU Emacs editor
  - e) Kate/Kwrite editor
  - f) Lime Text editorand many more.

# Why we use Linux?

- Costless
  - Stable
  - Reliable
- Extremely powerful
  - Highly Secure

# Merits and Demerits of Linux

- It can be easily accessible to the old computers .
- It cannot be made for gaming purpose.
- It is not easy to understand for those who are new to Linux.
- It is mostly used by the programmers.
- It is used for both commercial and personal but for home purpose, for this *Windows* is mostly preferred.



# Use of Linux in various fields



***Android App Development***



***Operating System for Routers/Transmitting Devices.***



***Game Designing***

***It is used also used in the department of Defence, Education.  
It is also popular in the field of Banking or Government Sector.***



# VIRTUAL MACHINES

# Virtual Machine

22

- A **Virtual Machine** is a software that creates a virtualized environment between the computer platform and the end user in which the end user can operate software.

# Description

23

- A virtual machine provides an interface identical to the underlying bare hardware.
- The operating system creates the illusion of multiple processes, each executing on its own processor with its own (virtual) memory.

# Virtualization

24

- Virtualization is an abstraction layer that decouples the physical hardware from the operating system to deliver greater IT resource utilization and flexibility.
- It allows multiple virtual machines, with heterogeneous operating systems to run in isolation, side-by-side on the same physical machine.

# Virtualization contd..

25

- Each virtual machine has its own set of virtual hardware (e.g., RAM, CPU, NIC, etc.) upon which an operating system and applications are loaded.
- The operating system creates the illusion of multiple processes, each executing on its own processor with its own (virtual) memory.

# History

26

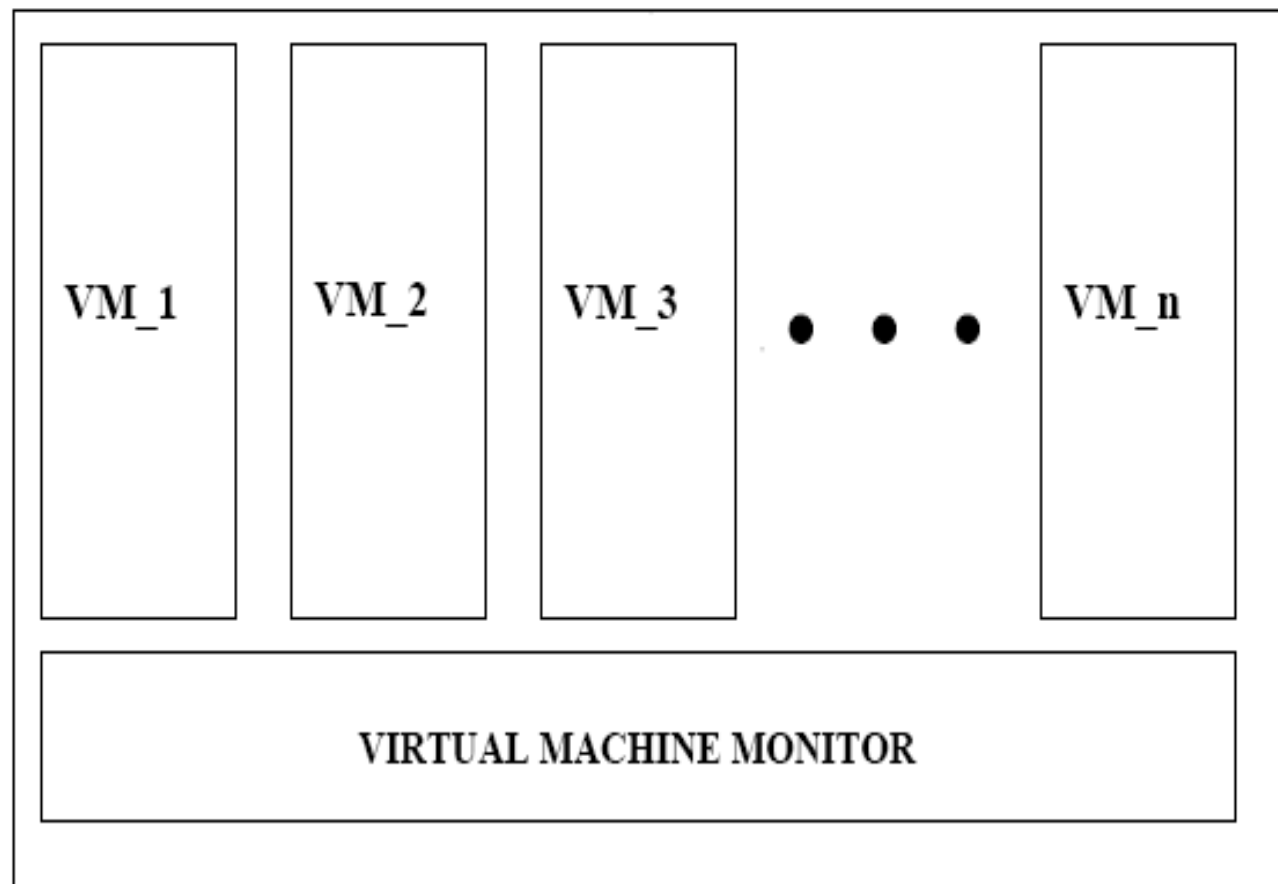
- ❑ Virtualization was first introduced in the 1960s to allow partitioning of large, mainframe hardware.
- ❑ In the 1990s, researchers began to see how virtualization could solve some of the problems associated with the proliferation of less expensive hardware, including underutilization, escalating management costs and vulnerability.



# Virtual Machine Monitor

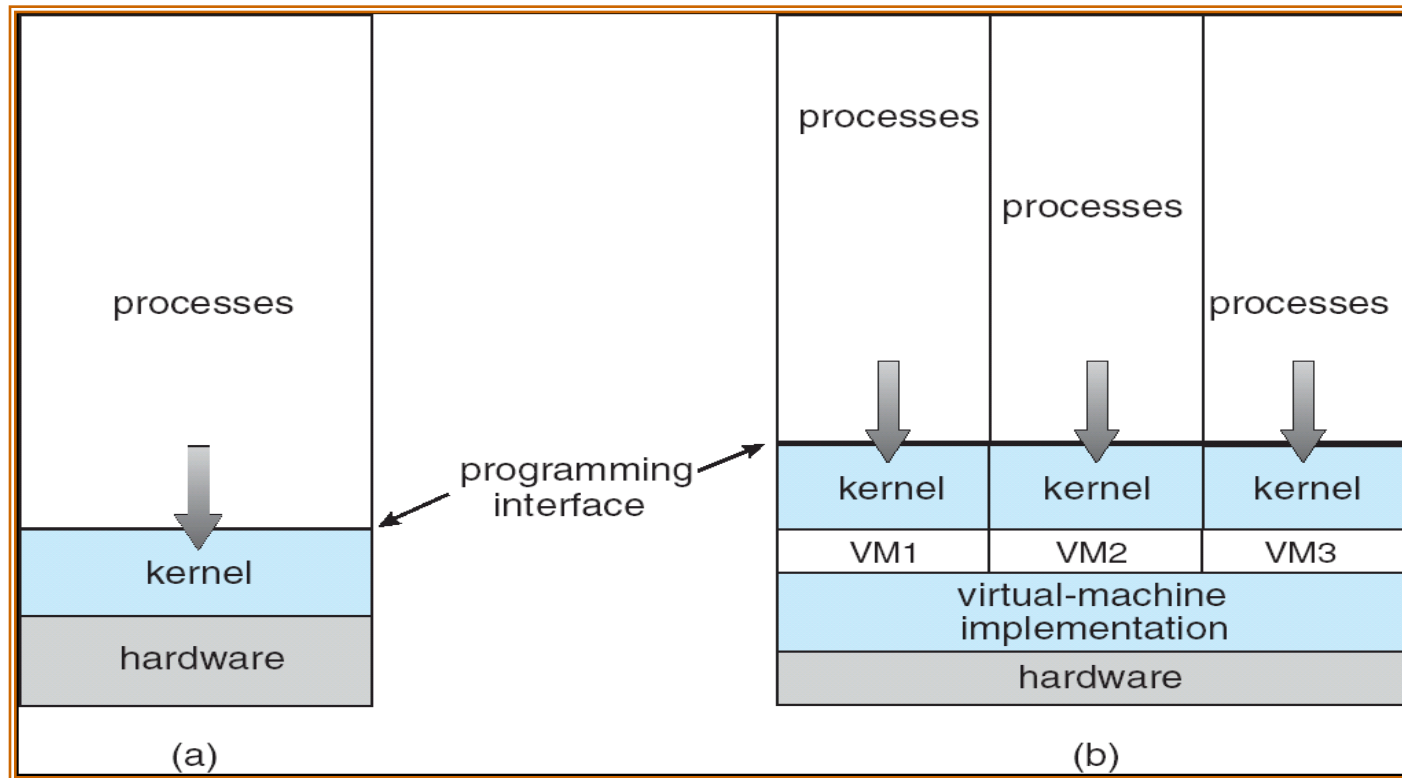
27

- ❑ The host software that provides virtualization is often referred to as a **virtual machine monitor (VMM)** or hypervisor.
- ❑ The VMM gives each virtual machine an illusion of a complete computer to itself.



# Architecture

29



# Features

- Each virtual machine has its own set of virtual hardware (e.g., RAM, CPU, NIC, etc.) upon which an operating system and applications are loaded.
- The operating system sees a consistent, normalized set of hardware regardless of the actual physical hardware components.

# Benefits

31

## 1. Partitioning

- Multiple applications and operating systems can be supported within a single physical system.
- There is no overlap amongst memory as each Virtual Memory has its own memory space.

# Benefits

32

## 2.Isolation

- Virtual machines are completely isolated from the host machine and other virtual machines. If a virtual machine crashes, all others are unaffected.

- Data does not leak across virtual machines.



# Summary

33

- **Virtual machines are a number of discrete identical execution environments on a single computer, each of which runs an operating system.**
- **This can allow applications written for one OS to be executed on a machine which runs a different OS which provide a greater level of isolation between processes than is achieved when running multiple processes on the same instance of an OS.**

Differences	Linux	Unix
Origins	Linux was developed in the 1990s by Linus Torvalds as a free and open-source alternative to Unix.	Unix was developed in the 1970s at Bell Labs
Introduction	<u>Linux</u> is Open Source, and a large number of programmers work together online and contribute to its development.	<u>Unix</u> was developed by AT&T Labs, different commercial vendors, and non-profit organizations.
Licensing	Linux, on the other hand, is open-source software and can be used freely without any licensing fees.	Unix is a proprietary any operating system, meaning that it requires a license to use.
Kernels	both have a similar design but are less complex than the Unixhold-upthat kernel.	both have a similar design but larger and more complex than the Linux kernel.
Availability	On the other hand, Linux is widely used on both enterprise and personal computers.	Unix is typically found on enterprise-level servers and workstations and is less commonly used on personal computers.

Accessibility	It is an open-source operating system which is freely accessible to everyone.	It is an operating system which can only be utilized by its copywriters.
<b>Source Code Availability</b>	The source is accessible to the general public.	The source is not accessible to the general public.
<b>Operating System Versions</b>	Some Linux versions are <a href="#">Ubuntu</a> , <a href="#">Debian</a> GNU, <a href="#">Arch Linux</a> , etc.	Some Unix versions are SunOS, <a href="#">Solaris</a> , SCO UNIX, <a href="#">AIX</a> , <a href="#">HP/UX</a> , ULTRIX, etc.
Community Support:	Linux has a large and active community of developers and users who contribute to its development and provide support.	While Unix also has a community, it is generally smaller and more focused on enterprise-level users.

## ROOT ACCOUNT



This is also called **superuser** and would have complete control of the Linux system. A superuser can run any commands without any restriction. This user should therefore not be used to access the system directly, it should be assumed as a system administrator.

## SYSTEM ACCOUNTS



- System accounts are those needed for the operation of system-specific components, for example mail accounts and the **sshd** accounts. These accounts are usually needed for some specific function on your Linux system, and any modifications to them could adversely affect the system.



# USER ACCOUNTS



User accounts provide interactive access to the system for users and groups. General users are typically assigned to these accounts and usually have limited access to critical system files and directories.

Linux uses *Group Account* which logically groups a number of accounts. Every account would be a part of another group account. A group plays important role in handling file permissions and process management.

# MANAGING USERS AND GROUPS IN LINUX



There are four main user administration files –

1. **/etc/passwd** – Keeps the user account and password information. This file holds the majority of information about accounts on the Unix system.
2. **/etc/shadow** – Holds the encrypted password of the corresponding account. Not all the systems support this file.
3. **/etc/group** – This file contains the group information for each account.
4. **/etc/gshadow** – This file contains secure group account information.

# BASIC USER ADMINISTRATION COMMANDS



- 1      **Useradd:** - Adds accounts to the system
- 2      **Usermod:** - Modifies account attributes
- 3      **Userdel:** - Deletes accounts from the system
- 4      **Groupadd:** - Adds groups to the system
- 5      **Groupmod:** - Modifies group attributes
- 6      **Groupdel:** - Removes groups from the system



# Commands of Linux

- There are some commands in Linux which give direct access to the files by using terminal.

Some of them are:

**ls-** (List Command)

**mv-** (Move Command)

**mkdir-** (Make Directories)

**rmdir-** (Remove Directory)

**locate-** (Locate Directory) etc.,

# Linux Commands

42

## Simple Linux Commands

Command	Meaning	Usage
cd	Change working directory.	cd .. (go up one level) cd /home/foxr (move to foxr's home directory)
echo	Output a message, including values stored in variables.	echo Hello world! echo Hello \$USER
hostname	Display the computer's host name.	
ls	List the items in the current working (or specified) directory(ies).	ls (list current working directory) ls /home/foxr (list contents of foxr's home directory)
pwd	Print the current working directory.	
vi, emacs	Run the vi or emacs text editor.	may be followed by a filename as in vi newfile1.txt
who	Print all logged in users.	
whoami	Print your username.	

# Linux Commands

43

Command	Meaning
arch	Output the computer's architecture (processor type).
bash	Start a new bash session.
exit	Exit the current bash session and if this is the "outermost" session, exit the current window.
passwd	Change your password; you are prompted to input the current password followed by a new password twice; if the password entered is not strong enough, you are warned (and depending on settings your new password may be rejected).
su	Switch user to the specified user; if no username is provided, switch to root; unless you are currently root you are queried to enter the user's password.
uname	Output information about your operating system.

# More About Linux Commands

<https://www.javatpoint.com/linux-ls>

# Man Pages in Linux

- ❑ Man command expects the name of a Linux command as its argument.
- ❑ Response of the man command is to display the *manual page* for that command, called a *man page for short*.
- ❑ Man page is displayed within the vi text editor.
- ❑ Movement and search commands available in vi can be used to move through the man page.
- ❑ It provides a detailed view of the command which includes NAME, SYNOPSIS, DESCRIPTION, OPTIONS, EXIT STATUS, RETURN VALUES, ERRORS, FILES, VERSIONS, EXAMPLES, AUTHORS and SEE ALSO.

# Man Commands Syntax

```
$man [OPTION]... [COMMAND NAME]...
```

```
$ man printf
```

```
$ man [SECTION-NUM] [COMMAND NAME]
```

# BASH FEATURES

47

- ❑ Linux Bash is also known as '**Bourne-again Shell.**'
- ❑ It is a **command language interpreter** for the Linux based system.
- ❑ It is a replacement of Bourne shell (sh).
- ❑ Linux/Unix shell allows us to interact with the Linux system through the commands.
- ❑ Bash is a **command language interpreter** as well as a **programming language**. It supports **variables, functions, and flow control**, like other programming languages. It can also read and execute the commands from a file, which is called a **shell script**.
- ❑ To open the terminal, press **CTRL+ALT+T** keys. Perform some basic operations such as **date, cal, ls, and pwd** to take a tour with it.

# BASH FEATURES

48

- ❑ **Shell Syntax:** The shell syntax contains **shell operations, quoting, and comments.**
- ❑ **Shell commands:** Shell commands are the types of commands that you can execute.
- ❑ **Shell Functions:** Shell functions are used to group commands by name.
- ❑ **Redirections:** It is a way to manage and control the input and output.
- ❑ **Command execution:** It decides how the system will react when we execute a command.
- ❑ **Shell Scripts:** It is a text file that has shell commands and executes them when it is used. Bash reads and executes the commands then exits.



# INTERPRETERS

49

- An interpreter is a program whose task is to execute instructions. To execute an instruction, the instruction must be converted into an executable form.
- parsing the instruction entered on the command line into its component parts and then executing the program that the command is implemented by on the arguments of the rest of the instruction.

# INTERPRETERS

50

- *Compiler* is a program which performs this translation process, converting an entire program into an executable program.
  
- Interpreted approach to programming utilizes an environment in which the programmer enters instructions one at a time.
- Each instruction, upon being entered, is parsed, converted into executable code and executed.
- Parsing, converting and executing steps are all handled by the *interpreter*
  
- ❖ ls – listing files
- ❖ ls –l display a detailed list

# FILENAME SPECIFICATION

51

- Focus on specifying paths and files as parameters in Linux instructions.
- *Path* is a description of how to reach a particular location in the file space.
- Path is required if the specified item is not in the current working directory.
- For executable programs, we can omit the path if the directory containing the program is stored in our PATH variable.
- Paths can be expressed as absolute and relative.

/home/student/

└─ Documents/

│ └─ file1.txt

│ └─ file2.txt

└─ Pictures/

│ └─ image1.jpg

│ └─ image2.jpg

└─ Projects/

└─ project1/

│ └─ code.c

└─ project2/

└─ script.sh

# Absolute Path:

53

- ❑ An absolute path specifies the exact location of a file or directory from the root directory. It always starts with a forward slash (/).
- ❑ For example:
  - Absolute path to file1.txt: /home/student/Documents/file1.txt
  - Absolute path to script.sh: /home/student/Projects/project2/script.sh
- ❑ An absolute path always begins at the root level of the file space (/) and specifies each directory from the root level down to the location of interest, separating each directory in the path by a forward slash (/).
- ❑ Length of an absolute path is determined by the number of directories/subdirectories that exist between the root level and the target file or directory.

# Relative path

54

- A relative path specifies the location of a file or directory relative to the current working directory.
- Here are some examples based on different scenarios:
  - ❖ If the current working directory is `/home/student/Documents/`, the relative path to `file2.txt` is: `file2.txt`
  - ❖ If the current working directory is `/home/student/Projects/project1/`, the relative path to `code.c` is: `code.c`
  - ❖ If the current working directory is `/home/student/`, the relative path to `image1.jpg` is: `Pictures/image1.jpg`

# INTERPRETERS

55

```
my_documents/  
|-- documents/  
|   |-- file1.txt  
|   |-- file2.txt  
|   |-- file3.txt  
|-- pictures/  
|   |-- pic1.jpg  
|   |-- pic2.jpg  
|-- notes.txt
```

# INTERPRETERS

56

- *ls command is used to list files and directories in a directory.*

- *\$ ls my\_documents*

*documents pictures notes.txt*

<https://bellard.org/jslinux/vm.html?url=alpine-x86.cfg&mem=192>




## 1. ls (List)

The `ls` command is used to list files and directories in a directory.

Example:

shell

 Copy code


```
$ ls  
file1.txt file2.txt directory1 directory2
```

## 2. cd (Change Directory)

The `cd` command is used to change the current directory.

Example:

shell

 Copy code


```
$ cd directory1
```

## 3. pwd (Print Working Directory)

The `pwd` command displays the current working directory path.

Example:

bash

 Copy code

```
$ pwd  
/home/user/directory1
```

## 4. mkdir (Make Directory)

The `mkdir` command is used to create a new directory.

Example:

shell

 Copy code

```
$ mkdir new_directory
```

## 5. rmdir (Remove Directory)

The `rmdir` command is used to remove a directory (which should be empty).

Example:

shell

 Copy code


```
$ rmdir directory2
```

## 6. cp (Copy)

The `cp` command is used to copy files or directories.

Example:

shell

 Copy code


```
$ cp file1.txt /path/to/destination
```

## 7. mv (Move)

The `mv` command is used to move or rename files and directories.

Example:

shell

 Copy code

```
$ mv file1.txt new_location/file1.txt  
$ mv old_filename.txt new_filename.txt
```

## 8. rm (Remove)

The `rm` command is used to remove files or directories.

Example:

shell

 Copy code

```
$ rm file1.txt  
$ rm -r directory1
```

## 9. cat (Concatenate and Display)

The `cat` command is used to display the content of a file.

Example:

shell

 Copy code


```
$ cat file1.txt
```

## 10. chmod (Change Mode)

The `chmod` command is used to change the permissions of a file or directory.

Example:

shell

 Copy code


```
$ chmod 755 file1.txt
```

## 11. chown (Change Owner)

The `chown` command is used to change the owner of a file or directory.

Example:

shell

 Copy code


```
$ chown user1 file1.txt
```

## 12. chgrp (Change Group)

The `chgrp` command is used to change the group ownership of a file or directory.

Example:

shell

 Copy code

```
$ chgrp staff file1.txt
```

# INTERPRETERS

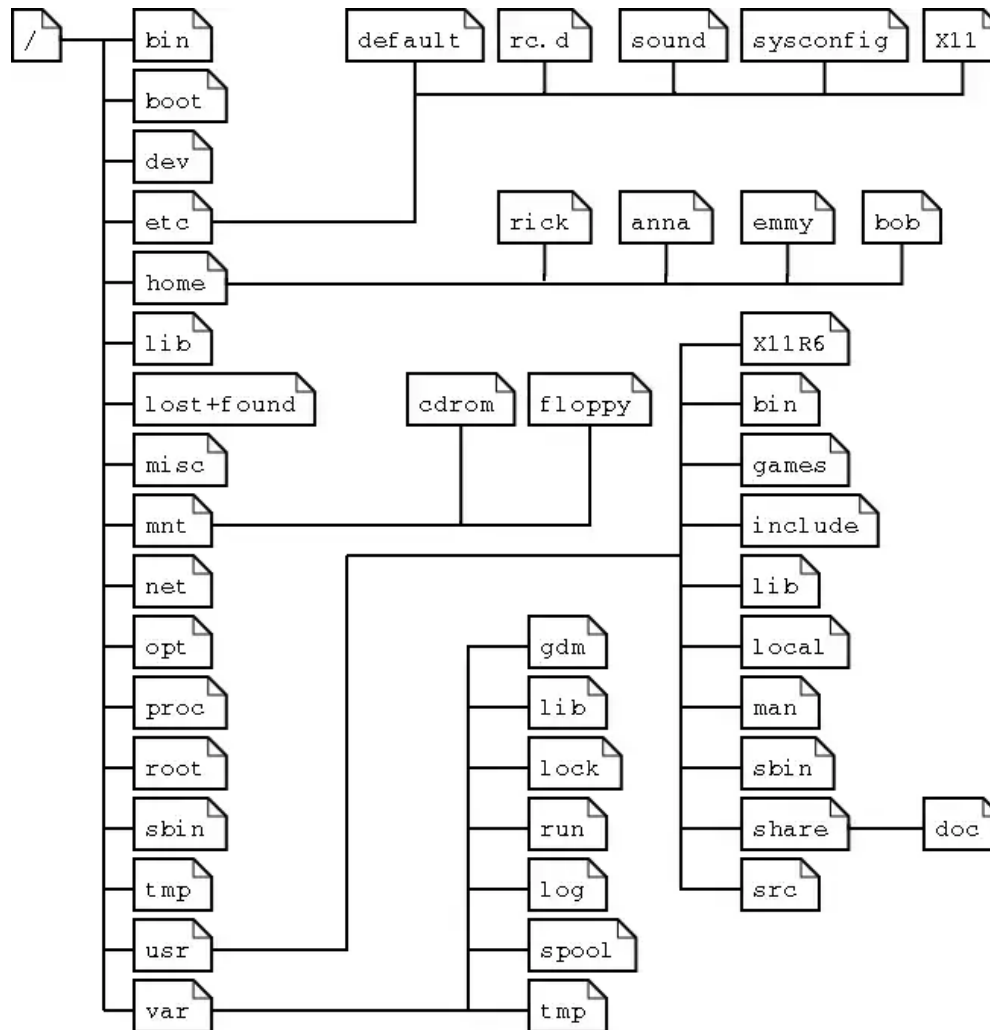
62

- *ls command is used to list files and directories in a directory.*

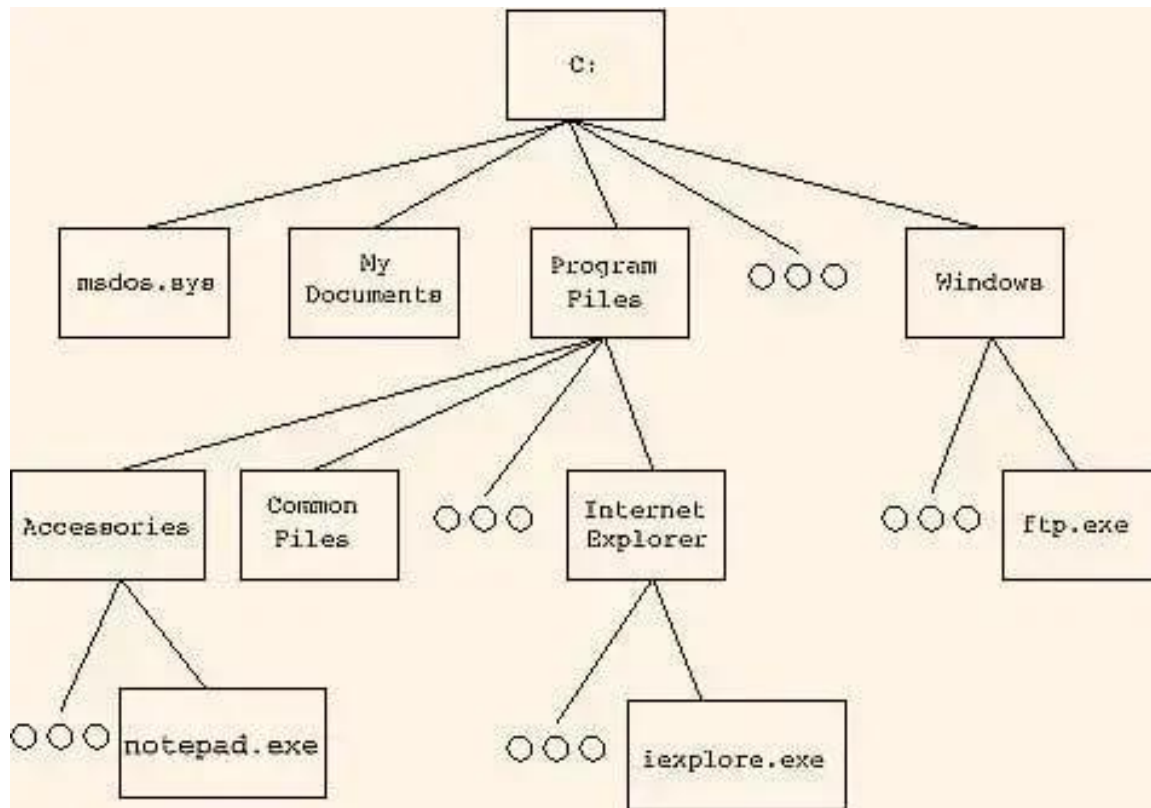
- *\$ ls my\_documents*

*documents pictures notes.txt*

# Linux System Structure



# Windows System Structure





# Linux System Structure


65

- **/** – this is known as “root”, the logical beginning of the Linux file system structure.
- **/bin** –most of your binary files are stored, typically for the Linux terminal commands and core utilities, such as `cd` (change directory), `pwd` (print working directory), `mv` (move), and so on.
- **/boot** – This is where all the needed files for Linux to boot are kept.
- **/dev** – This is where your physical devices are mounted, such as your hard drives, USB drives, optical drives, and so on.
- **/etc** –configuration files are stored.

# Linux System Structure

66

- ❑ **/home** – This is where you'll spend the vast majority of your time because it has all of your personal data.
- ❑ **/lib** – This is the location of libraries.
- ❑ **/media** – Another place where external devices such as optical drives and USB drives can be mounted.
- ❑ **/mnt** – This is basically a placeholder folder used for mounting other folders or drives. Typically this is used for Network locations.
- ❑ **/proc** – The “processes” folder where a lot of system information is represented as files.
- ❑ **/root** – This is the equivalent to the /home folder specifically for the root user, also called the superuser.

- 
- ❑ **/sbin** – Similar to /bin, except that it's dedicated to certain commands that can only be run by the root user, or the superuser.
  - ❑ **/tmp** – This is where temporary files are stored, and they are usually deleted upon shutdown, which saves you from having to manually delete them like is required in Windows.
  - ❑ **/usr** – Contains files and utilities that are shared between users.
  - ❑ **/var** – This is where variable data is kept, usually system logs but can also include other types of data as well.

Thank You