

Installation of Linux

<https://ubuntu.com/download/desktop>

Ubuntu 20.04.2.0 LTS

- **Dual Boot**

<https://www.youtube.com/watch?v=-iSAyiicyQY&t=276s>

- **Ubuntu as VM**

- **VMWARE:** *<https://www.youtube.com/watch?v=9rUhGWijf9U>*
- **VIRTUAL BOX:** *<https://www.youtube.com/watch?v=x5MhydiJWmc>*

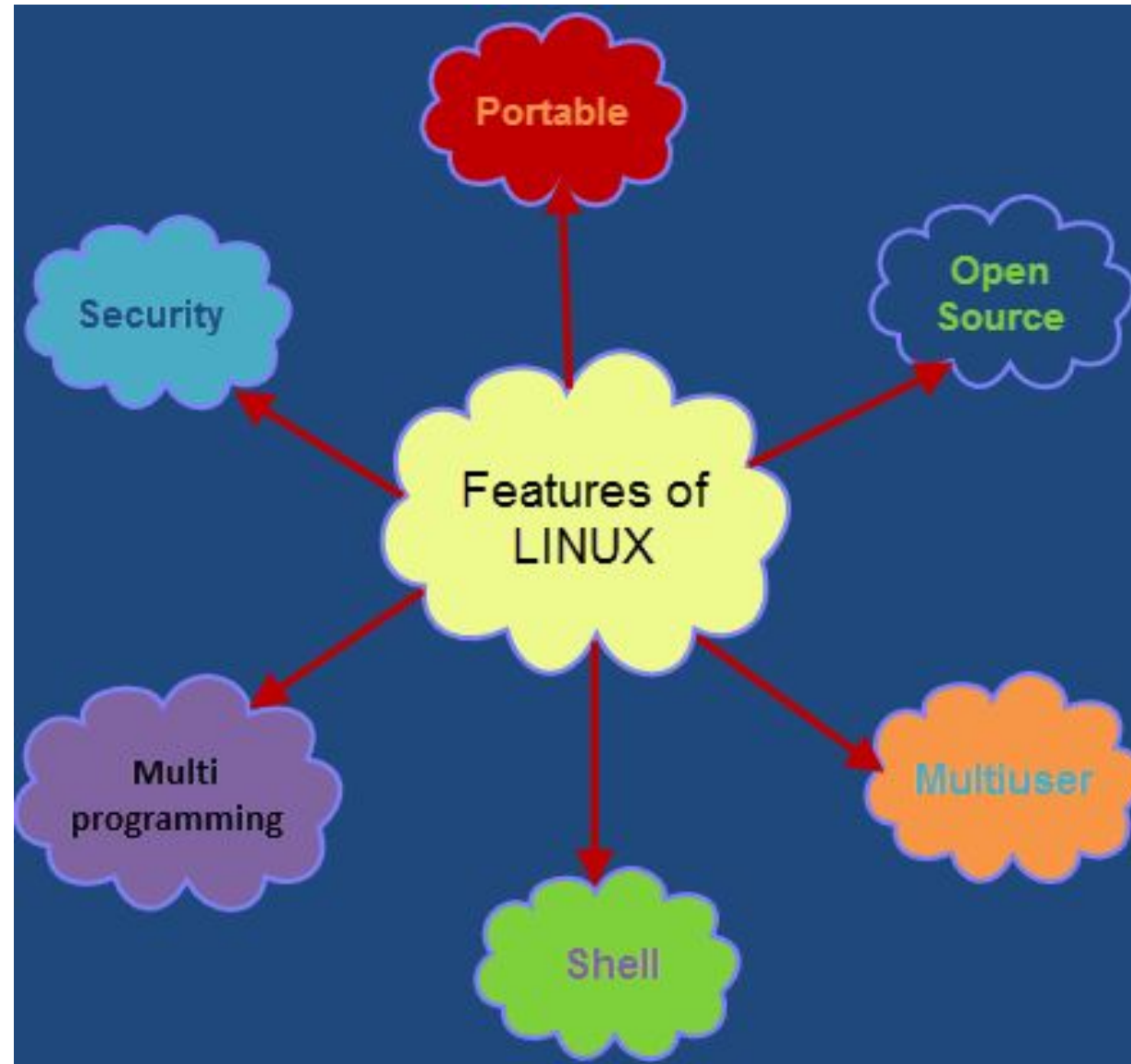
Operating System

- An operating system acts as an *intermediary* between the user of a computer and computer hardware. The purpose of an operating system is to provide an environment in which a user can execute programs *conveniently* and *efficiently*.
- Resource Management
- Process Management
- Storage Management
- Memory Management
- Security & Privacy

History of Linux

- *Linux OS is one of the famous versions of the **UNIX OS**. It is developed to provide a **low-cost or free** OS for several personal computer system users.*
- *In 1991, the Linux history started with the starting of a particular project by the Finland student **Linus Torvalds** for creating a new free **OS kernel**.*
- *Torvalds started developing Linux to create a system similar to **MINIX**, a UNIX operating system.*
 - *Linux was proposed by the Finland student Linus Torvalds in **1991**.*
 - *Hewlett Packard 9.0 version was published in 1992.*
 - *Red Hat Linux was proposed in 1994.*
 - *The first publication of **Ubuntu** was published in 2004.*
 - ***Oracle** published its Red Hat distribution in 2006.*

Features of Linux OS



Why Use Linux

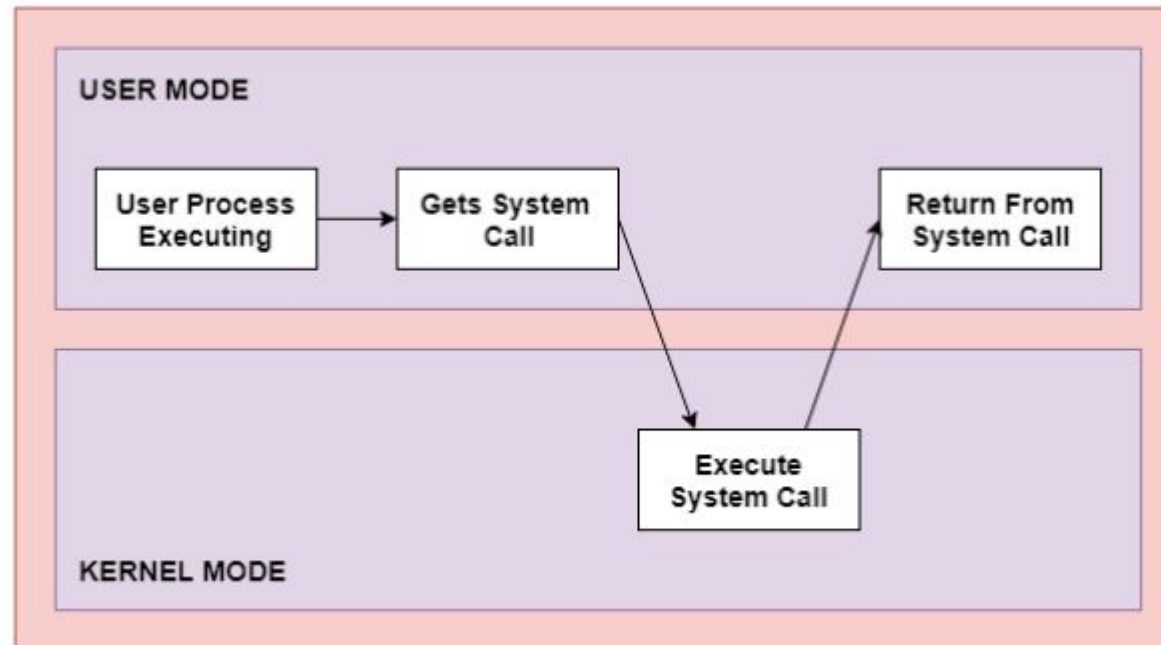
- It is an **open source** OS which gives a great advantage to the programmers as they can design their own custom operating systems.
- It gives you a lot of option of programs having some different features so you can choose according to your need.
- A global development community look at different ways to enhance its security, hence it is highly secured and robust so we don't need an anti virus to scan it regularly. Companies like Google, Amazon and Facebook use linux in order to protect their servers as it is highly reliable and stable.
- Above all you don't have to pay for software and server licensing to install Linux, its absolutely free and you can install it on as many computers as you want.
- Its completely trouble free operating system and don't have an issue with viruses, malware and slowing down your computer.

Linux Distributions List

- *Ubuntu*
- *Linux mint*
- *Red Hat Enterprise / CentOS*
- *Fedora*
- *Debian*

SYSTEM CALL

- System call is the programmatic way in which computer program requests a services from the kernel of the operating system.
- System calls are usually made when a process in user mode requires access to a resource. Then it requests the kernel to provide the resource via a system call.



SYSTEM CALL

Types of System Calls	Windows	Linux
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
File Management	CreateFile() ReadFile() WriteFile() CloseHandle()	open() read() write() close()
Device Management	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()
Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	pipe() shmget() mmap()

Command Structure

A **command** is a program that tells the Unix system to do something. It has the form:

command **[options] [arguments]**

where an argument indicates **on what the command is to perform its action**, usually a file or series of files. An option **modifies the command, changing the way it performs**.

- *Commands are case sensitive.*
- *Options are generally preceded by a hyphen (-), and for most commands, more than one option can be strung together, in the form*

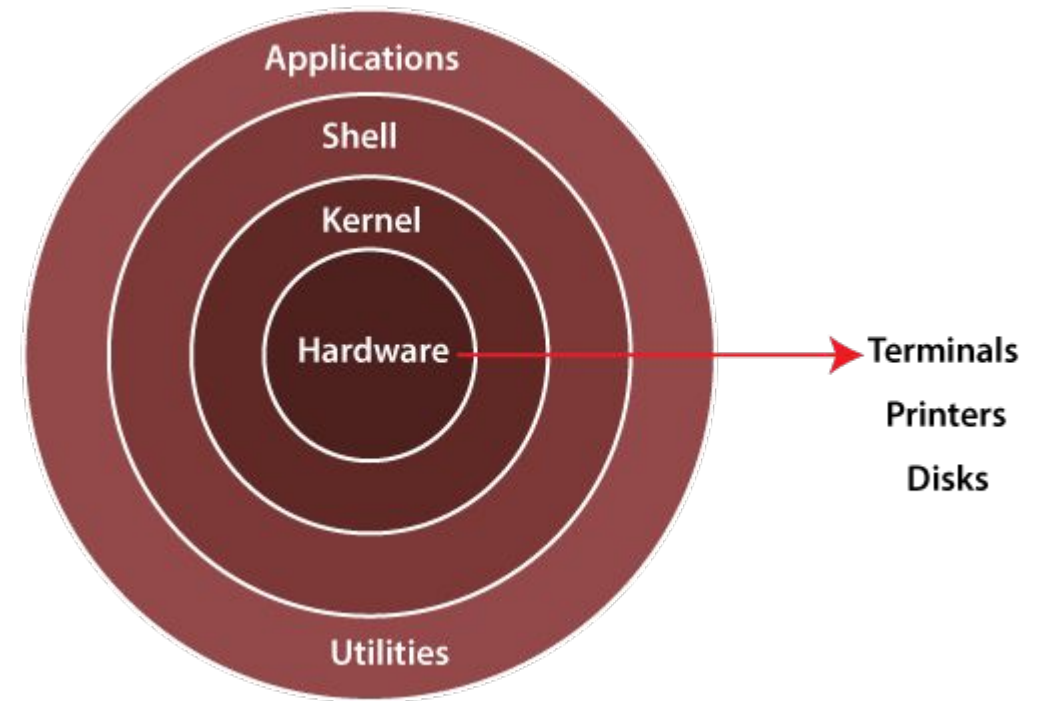
command -[option][option][option]

For most commands you can separate the options, preceding each with a hyphen, e.g.:

command -option1 -option2 -option3

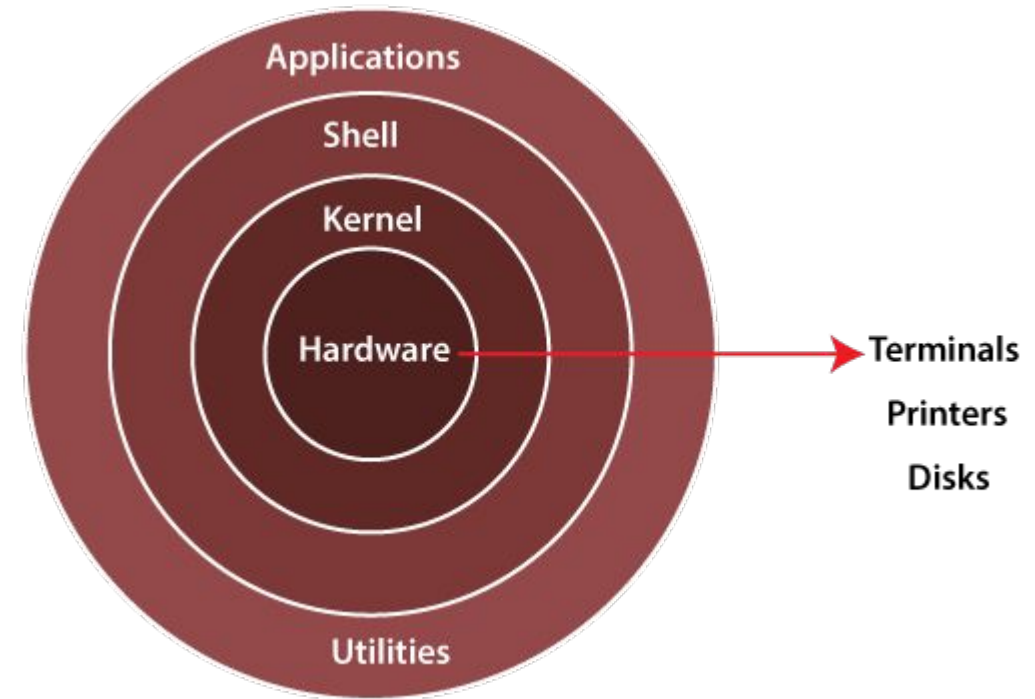
Architecture of Linux

- The kernel is a computer program that is the *core of a computer's operating system*, with complete control over everything in the system. It manages following resources of the Linux system –
 - *File management*
 - *Process management*
 - *I/O management*
 - *Memory management*
 - *Device management etc.*



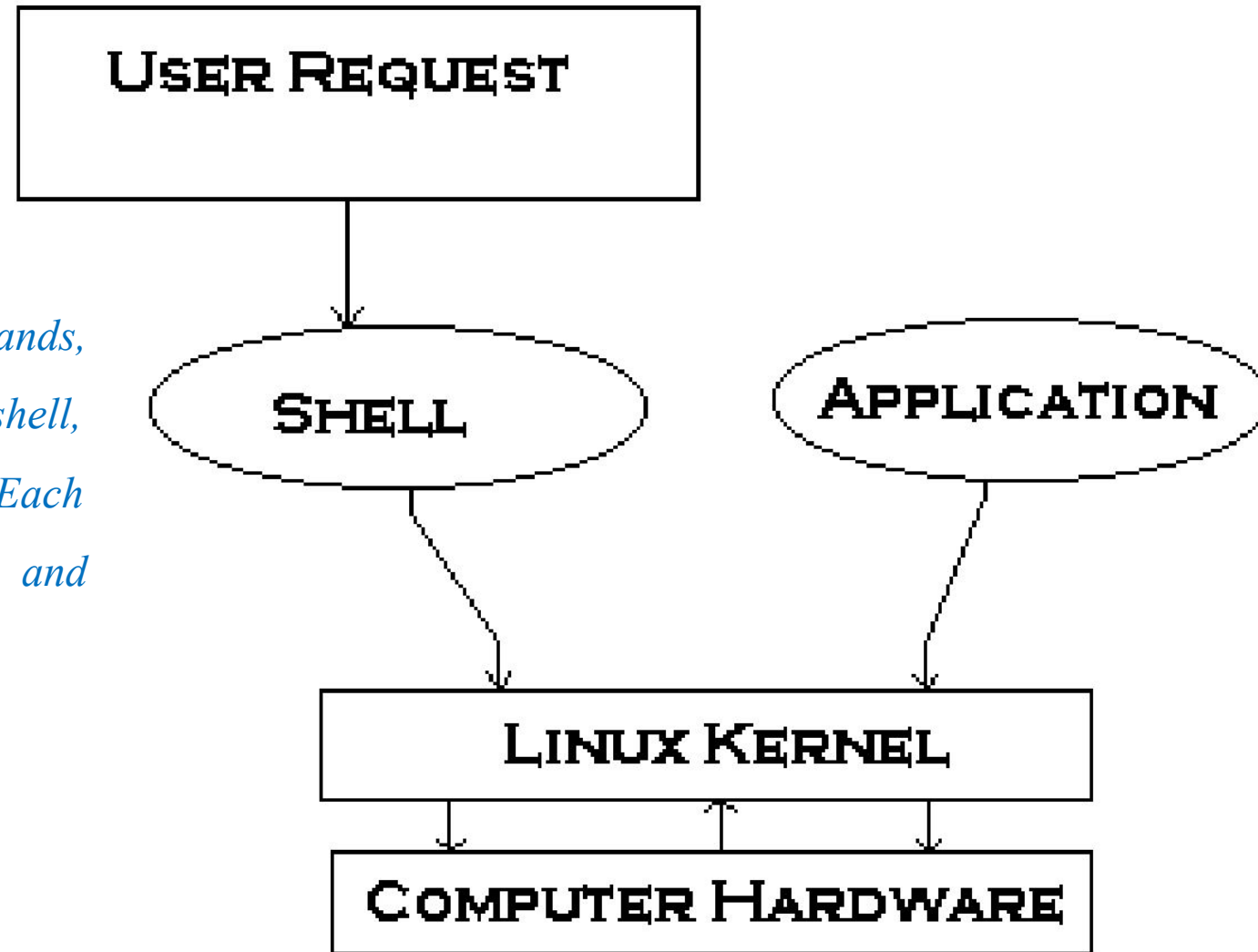
What is Shell

- A shell is special user program which provide an **interface** to user to use operating system services.
- Shell accept human readable commands from user and convert them into something which kernel can understand.
- It is a **command language interpreter** that execute commands.
- The shell gets started when the user logs in or start the terminal.



What is Shell

Shell is an environment in which we can run our commands, programs, and shell scripts. There are different flavors of a shell, just as there are different flavors of operating systems. Each flavor of shell has its own set of recognized commands and functions.

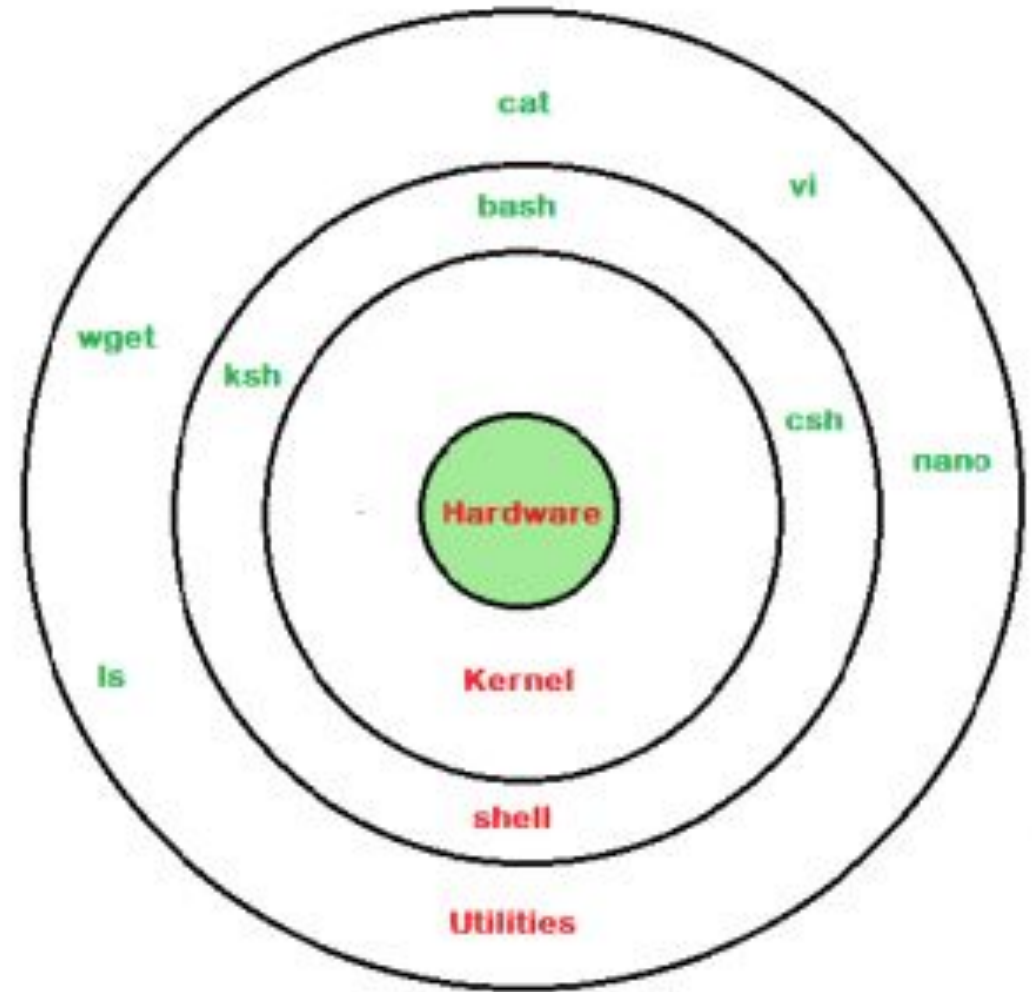


What is Shell

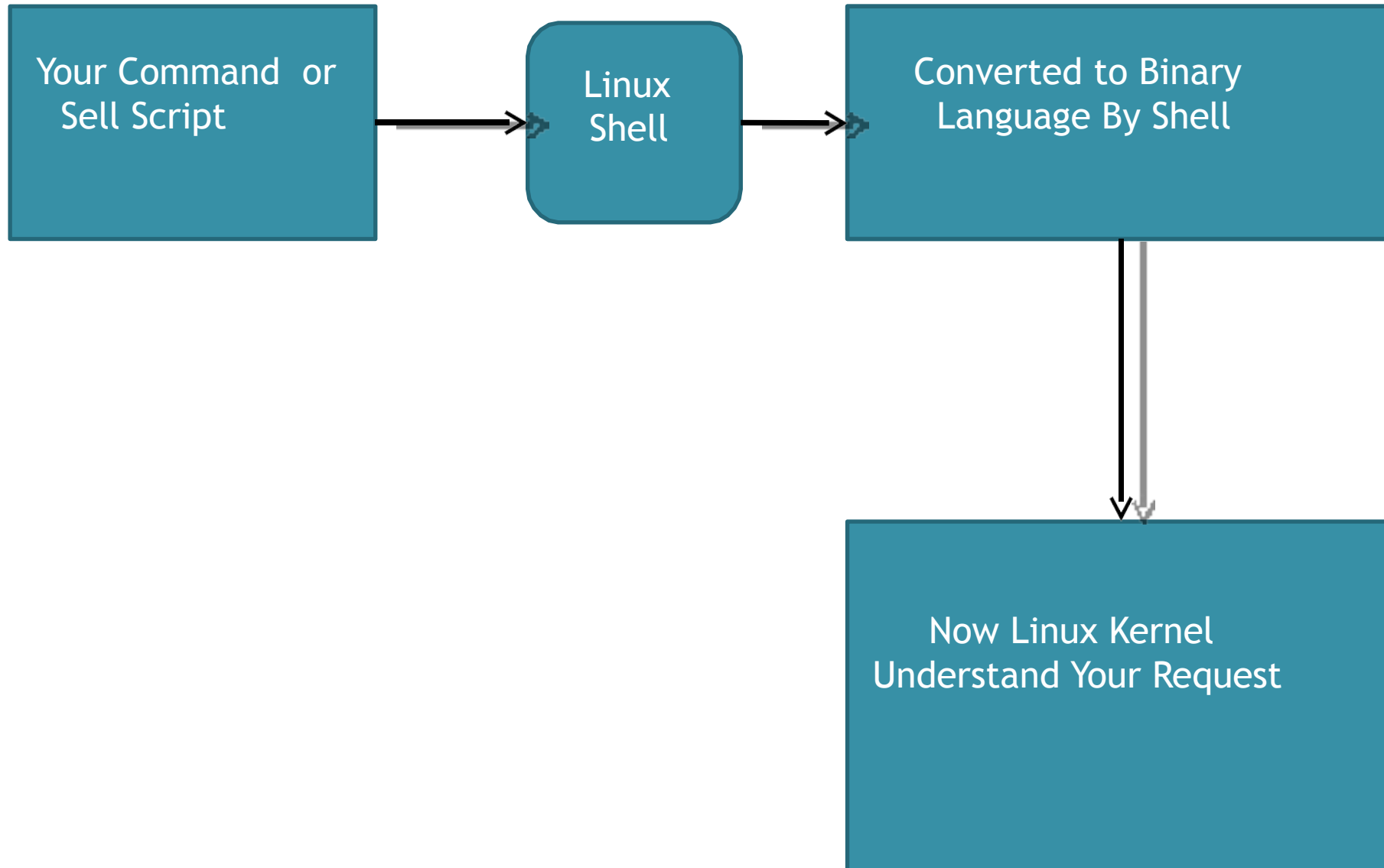
Shell is broadly classified into two categories –

- ***Command Line Shell/ Command Line Interface***
- ***Graphical shell/ Graphical user Interface***

CLI: When you run the **terminal**, the Shell issues a command prompt (usually \$), where you can type your input, which is then executed when you hit the Enter key.



What is Shell



Types of SHELL

- ***The Bourne Shell (sh):***

- Compact in nature and provide high speeds of operation.
- It doesn't have in-built functionality to handle logical and arithmetic operations.
- The Bourne shell cannot recall previously used commands.

Command full-path name is */bin/sh* and */sbin/sh*,

Non-root user default prompt is *\$*,

Root user default prompt is *#*

- ***The GNU Bourne-Again Shell (bash):***

- The default shell on many Linux distributions today. Compatible with the Bourne shell.
- Incorporates useful features from different types of shells in Linux such as Korn shell and C shell.
- It allows to automatically recall previously used commands and edit them with help of arrow keys, unlike the Bourne shell.

The full path-name for the GNU Bourne-Again shell is */bin/bash*.

Root user default prompt is *bash-VersionNumber#*

Non-Root user default prompt is bash-VersionNumber\$

Types of SHELL

- **The C Shell (csh):**

- It was developed to include useful programming features like in-built support for arithmetic operations and a syntax similar to the C programming language.
- Further, it incorporated command history which was missing in different types of shells in Linux like the Bourne shell.
- Another prominent feature of a C shell is “aliases”.

The complete path-name for the C shell is /bin/csh.

By default, it uses the prompt *hostname#* for the root user

By default, it uses the prompt *hostname%* for the non-root users.

Types of SHELL

- **The Korn Shell (ksh):**

- The Korn shell is essentially a superset of the Bourne shell
- It allows in-built support for arithmetic operations while offering interactive features which are similar to the C shell.
- It includes features like built-in arithmetic and C-like arrays, functions, and string-manipulation facilities.
- The Korn shell runs scripts made for the Bourne shell, while offering string, array and function manipulation similar to the C programming language.
- It also supports scripts which were written for the C shell. Further, it is faster than most different types of shells in Linux, including the C shell.

The complete path-name for the Korn shell is /bin/ksh.

By default, it uses the prompt # for the root user

By default, it uses the prompt \$ for the non-root users.

- **The Z Shell (zsh):**

- The Z Shell or zsh is a sh shell extension with tons of improvements for customization. If you want a modern shell that has all the features a much more, the zsh shell is what you're looking for.

Types of SHELL

Shell	Complete path-name	Prompt for root user	Prompt for non root user
Bourne shell (sh)	/bin/sh and /sbin/sh	#	\$
GNU Bourne-Again shell (bash)	/bin/bash	bash-VersionNumber#	bash-VersionNumber\$
C shell (csh)	/bin/csh	#	%
Korn shell (ksh)	/bin/ksh	#	\$
Z Shell (zsh)	/bin/zsh	<hostname>#	<hostname>%

Basic commands for SHELL

1. ***cat /etc/shells*** : List the Installed Shells
2. ***sudo apt-get install SHELL-NAME*** : Installing Another Shell
3. ***sudo -i***: change to root user
4. ***echo \$SHELL***: The \$SHELL environment variable holds the name of current shell.
5. ***Shell-name***: start new shell
6. ***chsh***: allows you to change your default shell.
7. ***chsh -s /usr/bin/shell-name***: To change your interactive shell

LINUX COMMAND – Directory

- **sudo (SuperUser DO):** Linux command allows you to run programs or other commands with administrative privileges, just like “Run as administrator” in Windows.
- **apt-get:** It is used to install, update, upgrade and remove any package.
- **sudo apt-get update:** This command updates the database and let your system know if there are newer packages available or not.
- **sudo apt-get upgrade:** After updating the package database, next step is to to upgrade the installed packages.
- **sudo apt-get install:** If you know the name of the package, then you can easily install a program using this command:

sudo apt-get install <package-name>

- **sudo apt-get remove:** When it comes to removing the installed program apt-get remove command suits your need.

sudo apt-get remove <package-name>

LINUX COMMAND – Directory

- ***sudo apt-get purge:*** apt-get purge command is used when you want to remove a software completely from your system with its configuration or data files so that no longer personalized settings will be available during reinstallation.

sudo apt-get purge <package-name>

LINUX COMMAND – File Listing (ls)

*The **ls** is the list command in Linux. It will show the full list or content of your directory.*

- ***ls** List the contents of the folder from which it runs.*
- ***ls -a** The (ls -a) command will enlist the whole list of the current directory including the hidden files. Hidden files name begins with “.”*
- ***ls -l** List all attributes of all files in current directory.*
- ***ls -lh** This command will show you the file sizes in human readable format (kb, gb, tb).*
- ***ls -R** Shows all the files not only in directories but also subdirectories.*
- ***ls -t** Place files in order of their modification. The last modified will be placed first.*
- ***ls -s** Biggest file in size will be placed first.*
- ***ls ../** It give the contents of parent directory.*
- ***ls -lX** It will group the files with same extensions together in the list.*
- ***ls -d*/** It is used to display only subdirectories.*

LINUX COMMAND – File Listing (ls)

- ***ls -g or ls -lG*** *With this you can exclude column of group information and owner.*
- ***ls -n*** *It is used to print group ID and owner ID instead of their names.*
- ***ls -lX*** *It will group the files with same extensions together in the list.*
- ***ls --color=[never/auto]*** *This command is used to print list as colored or discolored.*
- ***ls -l - -block-size=[SIZE]*** *It is used to display the files in a specific size format. Here, in [SIZE] you can assign size according to your requirement.*
- ***ls -p*** *It is used to identify the directory easily by marking the directories with a slash (/) line sign.*
- ***ls -g or ls -lG*** *With this you can exclude column of group information and owner.*

> Directing the output of a command

*To use bash redirection, you run a command, specify the **>** or **>>** operator, and then provide the **path of a file** you want the output redirected to.*

- ***>** redirects the output of a command to a file, replacing the existing contents of the file.*
- ***>>** redirects the output of a command to a file, appending the output to the existing contents of the file.*

ls > /path/to/file

ls >> /path/to/file

LINUX COMMAND – cat

- **cat:** *cat(concatenate) command is very frequently used in Linux. It reads data from the file and gives their content as output. It helps us to **create, view, concatenate** files.*

- *To view a single file*

\$cat filename

- *To view multiple files*

\$cat file1 file2

- *To view contents of a file preceding with line numbers.*

\$cat -n filename

- *Create a file*

\$ cat >newfile

- *Copy the contents of one file to another file.*

\$cat [filename-whose-contents-is-to-be-copied] > [destination-filename]

LINUX COMMAND – cat

- *Cat command can suppress repeated empty lines in output*

\$cat -s file name

- *Cat command can append the contents of one file to the end of another file.*

\$cat file1 >> file2

- *Cat command can display content in reverse order using tac command.*

\$tac filename

- *Cat command can highlight the end of line.*

\$cat -E "filename"

- *Cat command if the file has a lot of content and can't fit in the terminal.*

\$cat "filename" | more

LINUX COMMAND – cat

- *Cat command to merge the contents of multiple files.*

\$cat "filename1" "filename2" "filename3" > "merged_filename"

- *Cat command to display the content of all text files in the folder.*

\$cat *.txt

- *Cat command to write in an already existing file*

\$cat >> geeks.txt

The newly added text.

Feeding output of one command to another command

- The | *(pipe)* symbol connects two commands to create a pipeline

\$ ls | wc

The PATH

- *The sequence of directories that the shell searches to look for a command is specified in its `PATH` variable,*
- *Use `echo` to evaluate this variable and you will see directory list separated by colons:*

`echo $PATH`

Locating Command

- *Commands are essentially files containing programs written in c. Files are stored in directories.*
- *The easiest way of knowing the location of an executable program is to use **type** command.*

type cat

- *Type looks up only for the directories specified in PATH variable.*

Combining the Commands

- The **semicolon (;)** operator allows you to execute multiple commands in succession, regardless of whether each previous command succeeds. For example, open a Terminal window. Then, type the following three commands on one line, separated by semicolons, and press Enter. This will give you a listing of the current directory (`ls`), find out which directory you're currently in (`pwd`), and display your login name (`whoami`) all at once.

\$ ls ; pwd ; whoami

echo and echo -e

- **echo** command in linux is used to display line of text/string that are passed as an argument . This is a built in command that is mostly used in shell scripts and batch files to output status text to the screen or a file.

echo [option] [string]

- **echo** — The "**echo**" command helps us move some data, usually text into a file. For example, if you want to create a new text file or add to an already made text file, you just need to type i

\$ echo hello, my name is xyz >> new.txt.

echo and echo -e

- The **'-e'** option in Linux acts as interpretation of escaped characters that are back slashed.
- **\b** : it removes all the spaces in between the text

echo -e "abc \bklm \bxyz"

- **\c** : suppress trailing new line with backspace interpreter '-e' to continue without emitting new line.

echo -e "abc \cabc abc"

- **\n** : this option creates new line from where it is used.

echo -e "abc \nklm \nxyz"

- **\t** : this option is used to create horizontal tab spaces.

- **\r** : carriage return with backspace interpreter '-e' to have specified carriage return in output.

echo and echo -e

- **\v** : *this option is used to create vertical tab spaces.*

echo -e "abc \vklm \vxyz"

- **\a** : *alert return with backspace interpretor '-e' to have sound alert.*

echo -e "\aabc xyz"

echo and echo -e

- ***echo *** : this command will print all files/folders, similar to ls command .*

Example : echo *

LINUX COMMAND – Directory

- ***pwd:*** *To know which directory you are in, you can use the “pwd (print working directory)” command.*
- ***cd dir:*** *Use the "cd" command to go to a directory. With the help of this command, we can move all over our directories in our system.*
- ***cd dir_1/dir_2/dir_3:*** *This command is used to move inside a directory from a directory using absolute path.*
- ***cd /:*** *this command is used to change directory to the root directory.*
- ***cd ~ :*** *this command is used to change directory to the home directory.*
- ***cd .. :*** *this command is used to move to the parent directory of current directory, or the directory one level up from the current directory.*
- ***cd –*** – *Takes you to the previous directory.*
- ***cd ~username:*** *We can change the directory from the current working directory to a user's home directory*
- ***history:*** *history command displays all of your previous commands up to the history limit.*
- ***shutdown/poweroff:***

LINUX COMMAND – Directory/Files

- ***mkdir***— Use the **mkdir** command when you need to create a folder or a directory
- ***mkdir <dirname1> <dirname2> <dirname3>***: To make multiple directories.
- ***rmdir***: Use **rmdir** to delete a directory. But **rmdir** can only be used to delete an empty directory.
- ***rmdir -p***: This command will delete a directory including its sub-directories all at once.
- ***rm***: Use the **rm** command to delete files and directories. Use "**rm -r**" to delete just the directory. It deletes both the folder and the files it contains when using only the **rm** command.
- ***touch*** — The **touch** command is used to create a file. It can be anything, from an empty txt file to an empty zip file. For example, “**touch new.txt**”.
- ***cp*** — Use the **cp** command to copy files through the command line. It takes two arguments: The first is the location of the file to be copied, the second is where to copy.

LINUX COMMAND – Directory/Files

- ***mv*** — Use the **mv** command to move files through the command line. We can also use the **mv** command to rename a file. For example, if we want to rename the file “**file1**” to “**file2**”, we can use “**mv file1 file2**”.
- ***locate*** — The **locate** command is used to locate a file in a Linux system, just like the search command in Windows. Using the ***-i*** argument with the command helps to ignore the case (it doesn't matter if it is uppercase or lowercase). So, if you want a file that has the word “hello”, it gives the list of all the files in your Linux system containing the word "hello" when you type in “**locate -i hello**”. If you remember two words, you can separate them using an asterisk (*). For example, to locate a file containing the words "hello" and "this", you can use the command “**locate -i *hello*this**”.

LINUX COMMAND

- **zip, unzip:** Use **zip** to compress files into a zip archive, and **unzip** to extract files from a zip archive.
- **ping:** Use **ping** to check your connection to a server.
- **clear:** This command clears the screen.
- **head:** Displays the first few lines of a file. By default, the 'head' command displays the first 10 lines of a file.

But with -n option, the number of lines to be viewed can be specified.

- **tail:** Similar to 'head'; the 'tail' command shows the last 10 lines by default, and -n option is available as well.

cal: The Calendar

- *cal* is a handy tool that you can invoke any time to see the calendar of any specific month or a complete year.

\$ cal 7 2006

cal: The Calendar

- *cal* is a handy tool that you can invoke any time to see the calendar of any specific month or a complete year.

\$ cal 7 2006

who: Who are the users?

- *Linux is a system that can be concurrently used by multiple users, and you might be interested in knowing the people who are also using the system like you.*

\$ who

WC

- ***wc***: This command counts lines, words and letters of the input given to it.

\$ wc file-name

Internal and External commands

- *The UNIX system is command-based i.e things happen because of the commands that you key in.*
- They are grouped into two categories:*
- ***Internal Commands*** : *Commands which are built into the shell. For all the shell built-in commands, execution of the same is fast in the sense that the shell doesn't have to search the given path for them in the PATH variable. Examples: source, cd, fg, etc.*
 - ***External Commands*** : *Commands which aren't built into the shell. When an external command has to be executed, the shell looks for its path given in the PATH variable. They are usually located in **/bin or /usr/bin**. For example, when you execute the “cat” command, which usually is at /usr/bin, the executable /usr/bin/cat gets executed.*
 - *Examples: ls, cat etc.*

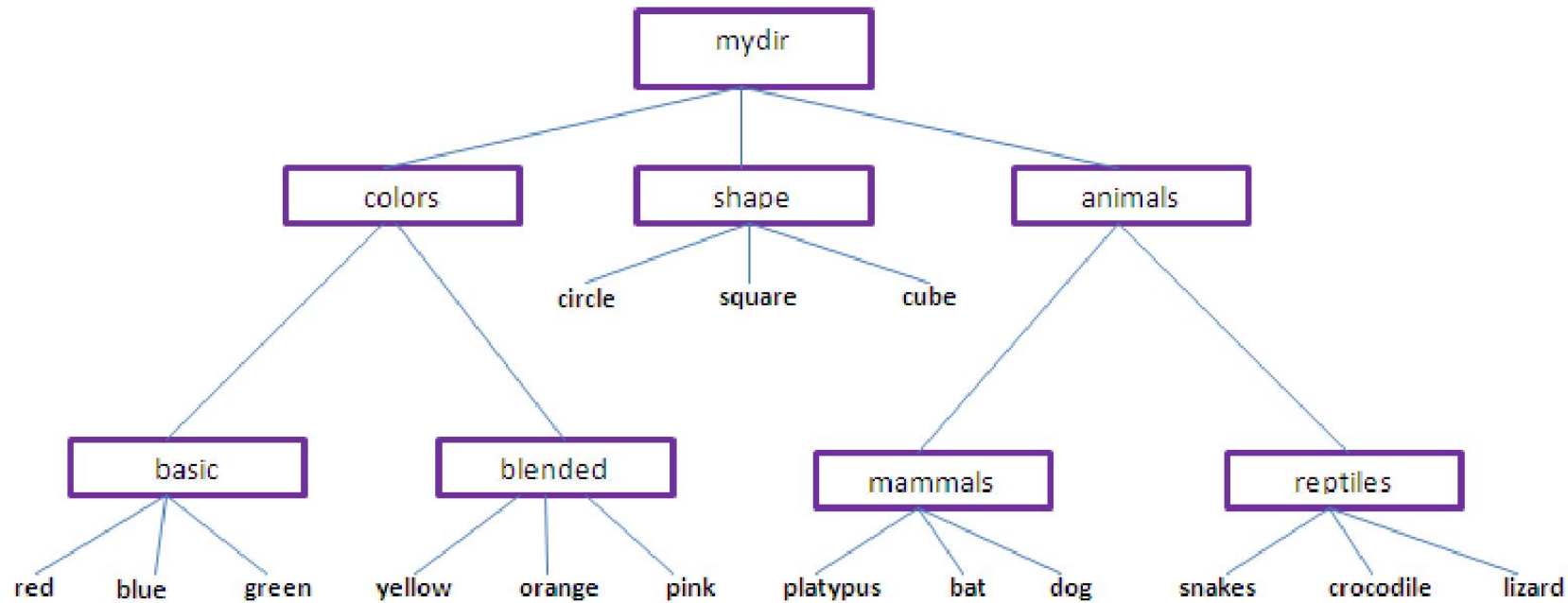
Internal and External commands

- *Most commands are external in nature, but there are some which are not really found anywhere, and some which are normally not executed even if they are in one of the directories specified by PATH. For instance, take echo command:*

\$ type echo

echo is a shell builtin

- *echo isn't an external command in the sense that, when you type echo, the shell won't look in its PATH to locate it(even if it is there in /bin). Rather, it will execute it from its own set of built-in commands that are not stored as separate files. These built-in commands, of which echo is a member, are known as internal commands.*



1. Copy red.txt to reptiles,
2. Delete Shape directory,
3. copy orange.txt to bat.txt,
4. Rename the yellow.txt to some other name and append it to snakes.txt,
- 5) Print first 5 lines for the file lizard.txt.
6. Print the number of bytes, words, and lines in the pink.txt.