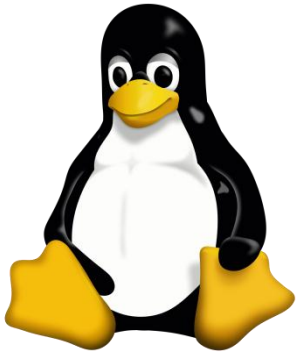
Linux Fundamentals

VERSION 4
April-2023



Part 2

Eng Ali Mohammad. Bani Bakkar

What is the shell?

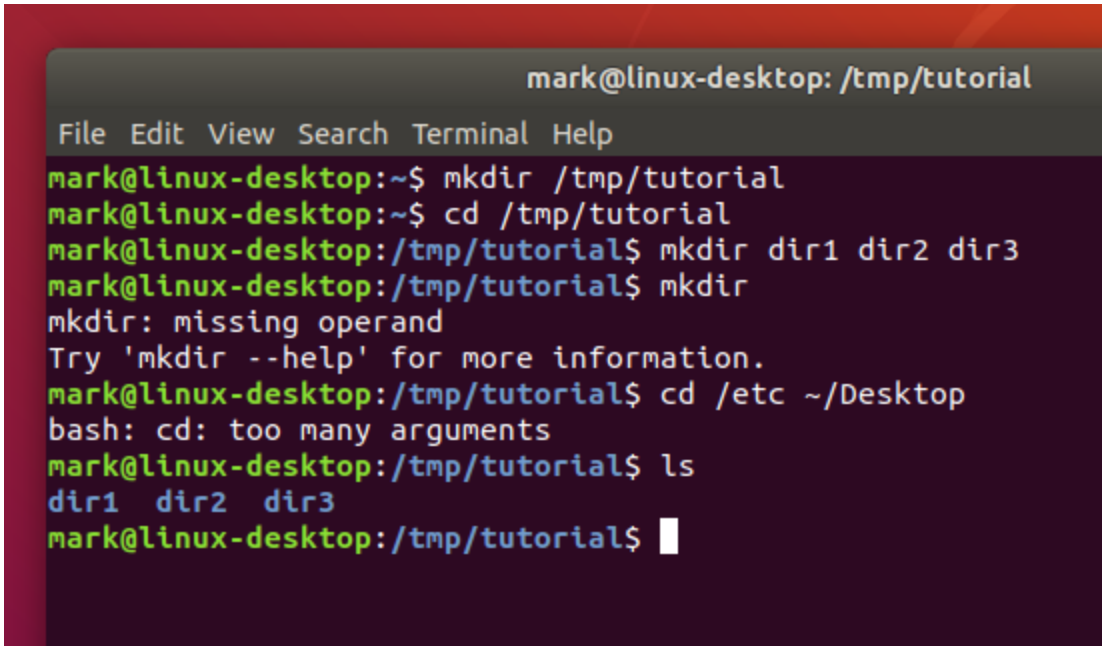
- The shell is a program that takes commands from user's keyboard and passes them to the operating system to execute.
- Many shell programs are available for Linux:
- bash, sh, csh, tcsh, ksh, zsh, …
- we will use bash, but other shells are conceptually similar

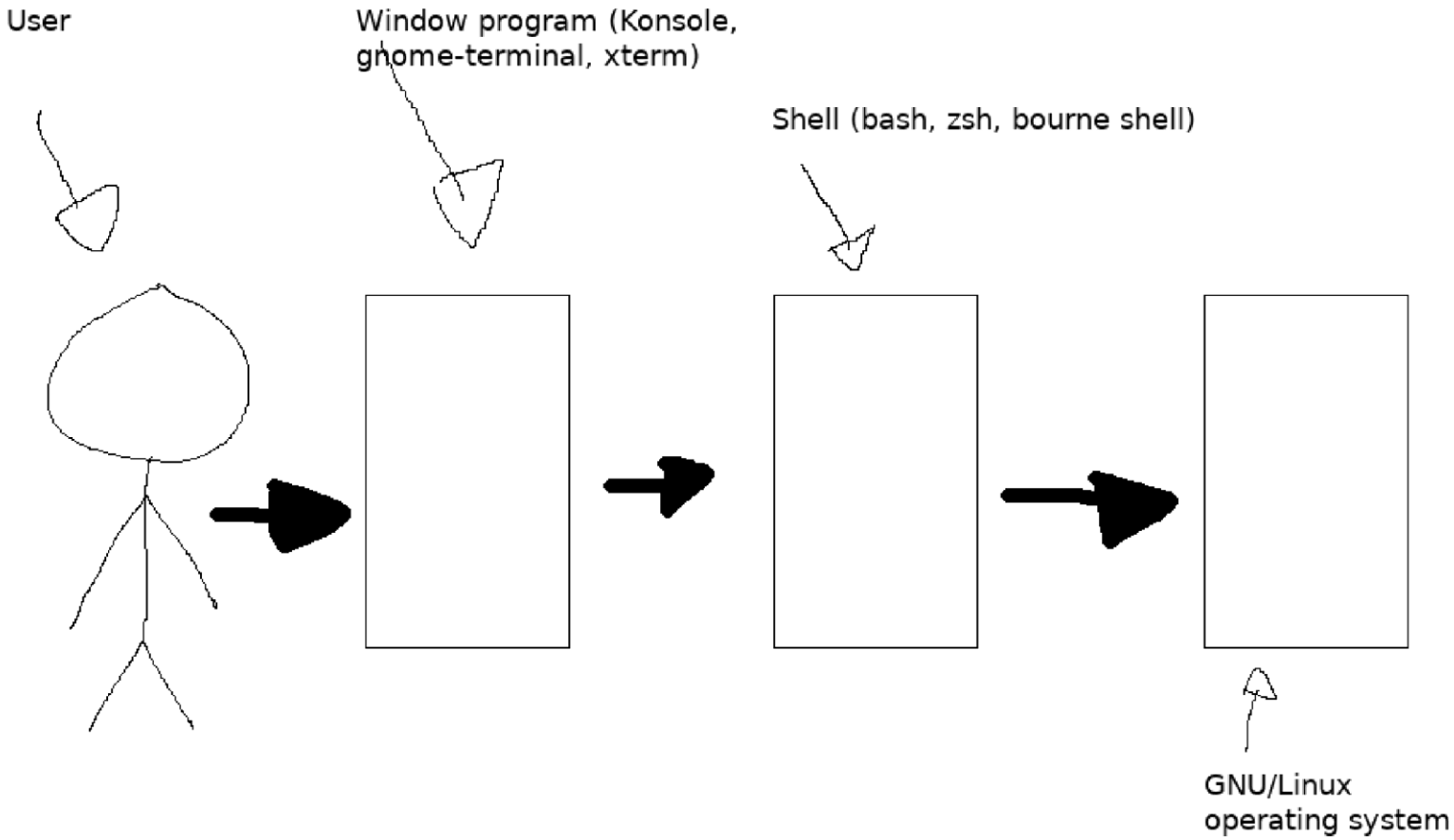
What is the shell?

```
metalx1000@fort ~ % date -d "1/2/2016"
Sat Jan  2 00:00:00 EST 2016
metalx1000@fort ~ % date -d "1/2/2016 1am"
Sat Jan  2 01:00:00 EST 2016
metalx1000@fort ~ % date -d "1/2/2016 1am" +%s
1451714400
metalx1000@fort ~ % date -d "1/2/2016 1am" +%F
2016-01-02
metalx1000@fort ~ % date -d "1/2/2016 1am" +%s
1451714400
metalx1000@fort ~ % date -d "1/2/2016 1:12:11am" +%s
1451715131
metalx1000@fort ~ % date -d "1/2/2016 1:12:11am"
Sat Jan  2 01:12:11 EST 2016
metalx1000@fort ~ % date -d "Monday Jan 2016 1:12:11am"
date: invalid date ‘Monday Jan 2016 1:12:11am’
1 metalx1000@fort ~ % date -d "Jan 2nd 2016 1:12:11am"                    :(
date: invalid date ‘Jan 2nd 2016 1:12:11am’
1 metalx1000@fort ~ % date -d "Jan 2nd 2016 1:12:11am"                    :(
0:zsh*Z                                                          10/16  10:14
```

What is Terminal?

- Terminal is a program that opens in a window and lets users interact with the shell.




User
Window program (Konsole, gnome-terminal, xterm)
Shell (bash, zsh, bourne shell)
GNU/Linux operating system

Ls Command

- **Ls :**You can list the contents of a directory with **ls**.
- **ls –a :** A frequently used option with ls is **-a** to show all files (including **hidden files**)
- the **hidden files**. When a file name on a Linux file system starts with a dot.
- **ls –l :**Typing just **ls** gives you a
- list of files in the directory. Typing **ls -l** gives you a

long listing.

ls –lh : It shows the numbers (file sizes) in a more human readable format.

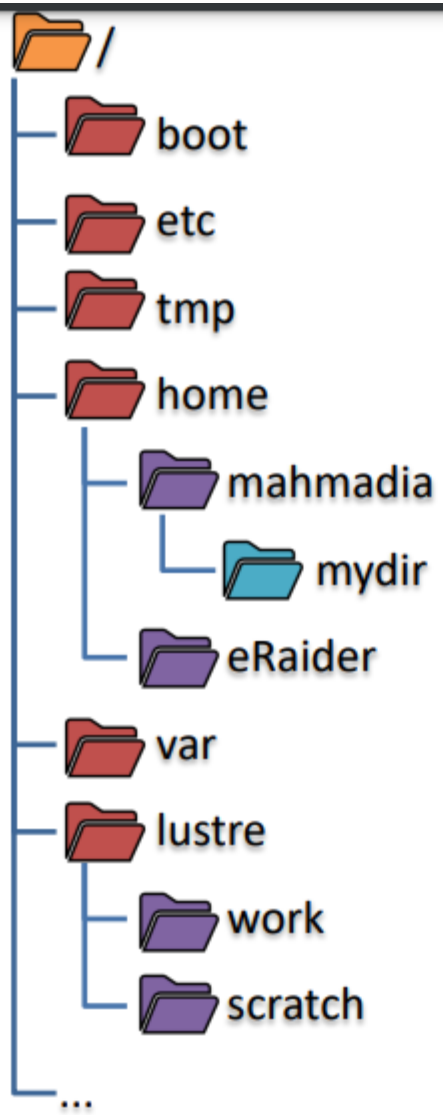
Ls –lr :

Ls –lt :

Ls ~

Ls .

Working With Directories



/
/boot
/etc
/tmp
/home
/home/mahmadia
/home/mahmadia/mydir
/home/eraider
/var
/lustre
/lustre/work
/lustre/scratch

Working With Directories (1)

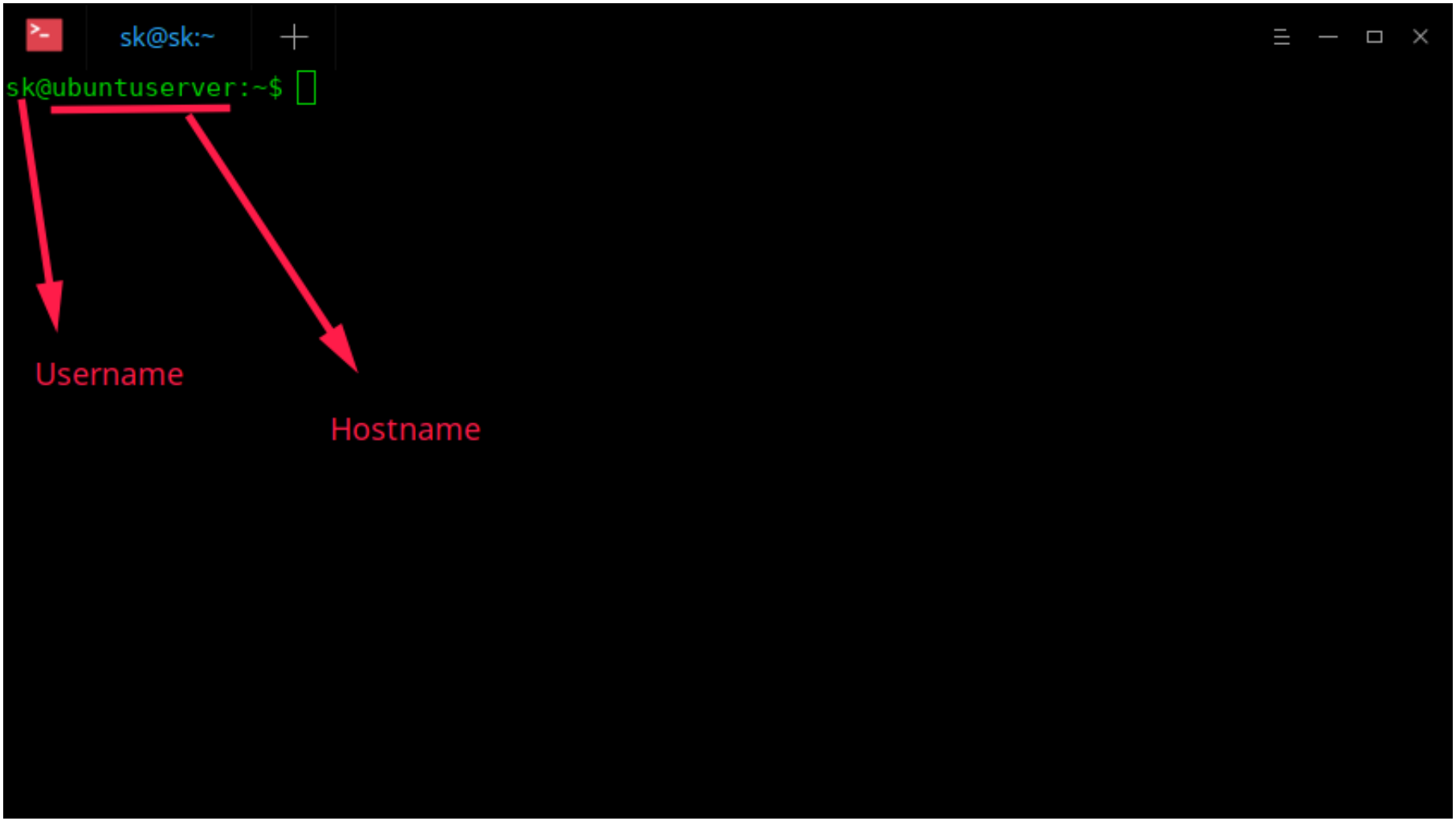
- **Pwd** (Print Working Directory):The tool displays your **current directory**.
- **Cd :** You can change your current directory with the **cd** command (Change Directory).
- **cd ~:**The **cd** is also a shortcut to get back into your home directory. Typing **cd ~** has the same effect.

Working With Directories (2)

- **cd .. :** To go to the **parent directory** (the one just above your current directory in the directory tree), type **cd ..** .
- cd ../../
- *To stay in the current directory, type **cd .***
- **cd - :** to go to the previous directory.

Try

- Ls --help
- Init – help??
- Cal –help
- Date –help
- su
- Passwd
- Whoami
- Which
- History [!number of line]
- History –c (clear)
- Cat .bash_history
- cd ~ali
- Dir --help
- Nautilus :Nautilus file manager for the GENOME desktop system. similar to windows explorer.
- hostname


sk@sk:~
sk@ubuntuserver:~$
Username
Hostname

apt update vs. apt upgrade: A Comparison

- **apt update** command might seem like the obvious go-to option to update your packages on Linux, it’s not entirely the case. The update command gives you an idea about the available updates, but it does not download or install the updates within your distro.
- **apt upgrade** command downloads and installs available updates on your machine in one go. Your Linux system has an available cache of software (packages), which contains the necessary metadata related to those packages. The metadata includes information pertaining to the version, repository, dependency, and other relevant package details.
- If you don’t use the update command, you won’t refresh the cache, which would not give you a clue about the available package updates.

```
[linuxconfig.org]$ sudo apt update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
26 packages can be upgraded. Run 'apt list --upgradable' to see them.
[linuxconfig.org]$
[linuxconfig.org]$ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  alsa-ucm-conf apport apport-gtk busybox-initramfs busybox-static gir1.2-snapd-1 grub-common grub-pc
  grub-pc-bin grub2-common libasound2 libasound2-data libatopology2 libefiboot1 libefivar1 libsnapd-glib1
  python3-apport python3-distupgrade python3-problem-report snapd ubuntu-release-upgrader-core
  ubuntu-release-upgrader-gtk yaru-theme-gnome-shell yaru-theme-gtk yaru-theme-icon yaru-theme-sound
26 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 48.9 MB of archives.
After this operation, 632 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

- Apt update
- Apt upgrade
- Apt install package_name
- Apt remove package_name
- Apt remove –purge package_name //configuration
- Dpkg –I package_name

- Example

Apt-cache search apache2
Apt show apache2
Apt install apache2

```
sofija@sofija-VirtualBox:~$ sudo apt-get install nano
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  hunspell
The following NEW packages will be installed:
  nano
0 upgraded, 1 newly installed, 0 to remove and 241 not upgraded.
Need to get 269 kB of archives.
```

help

- Man ls
- Ls -- help
- Whatis ls
- Apropos : The **apropos** command helps users find any command using its man pages.
- Which ls
- Locate passwd //updated
- Find location –name filename

Absolute and Relative Paths

- An **absolute path** is defined as specifying the location of a file or directory from the root directory(/).
- **Relative path** is defined as the **path** related to the present working directly(pwd).

Absolute and Relative Paths Example

- current directory **/home/ali**. From within this
- directory, you have to type **cd /home** instead of **cd home** to go to the **/home** directory.
- $ **pwd**
- /home/ali
- $ **cd home**
- bash: cd: home: No such file or directory
- $ **cd /home**
- $ **pwd**
- /home
- When inside **/home**, you have to type **cd ali** instead of **cd /ali** to enter the subdirectory
- **ali** of the current directory **/home**.
- $ **pwd**
- /home
- $ **cd /ali**
- bash: cd: /ali: No such file or directory
- $ **cd ali**
- $ **pwd**
- /home/ali

Path Completion

- The **tab key** can help you in typing a path without errors.
- Typing **cd /et** followed by the **tab**
- **key** will expand the command line to **cd /etc/**. When typing **cd /Et** followed by the **tab key**,
- nothing will happen because you typed the wrong **path** (upper case E).

Mkdir Command

- **Mkdir :** to create a directory.
- **mkdir –p : create the directory and, if required, all parent directories**.
- mkdir -p fruits/apples
- mkdir fruits cars

Rmdir Command

- **Rmdir :** When a directory is empty, you can use **rmdir** to remove the directory.
- **rmdir –p :**remove directory and its ancestors
- **rm –r** : you can use **rmdir** to recursively remove directories.
- **rm –rf**: you can also use **rmdir** to recursively force remove directories with out interaction.
- To delete folders with files in them, we'll use the more

generic rm command which deletes files and folders,

using the -rf options:

Working with Files

- **File**
- **touch**
- **rm**
- **cp**
- **mv**
- **rename**.

Working with Files Notes (1)

- **all files are case sensitive (**This means that **FILE1** is different from **file1**, and **/etc/hosts** is different from **/etc/Hosts) .**
- **everything is a file (**A **directory** is a special kind of **file**, but it is still a (case sensitive!) **file**.
- Each terminal window (for example **/dev/pts/4**), any hard disk or partition (for example **/dev/sdb1**) and any process are all represented somewhere in the **file system** as a **file**.

Working with Files Notes (2)

- **File :** The **file** utility determines the file type. Linux does not use extensions to determine the file type.

$file filename

- The command line does not care whether a file ends in .txt or .pdf. As a system administrator, you should use the **file c**ommand to determine the file type.

Touch Command

- **Touch** Touch command is a **Linux command** is mainly used to create empty files, and change timestamps of files or folders. The timestamp information of files consists of three attributes – access time, modification time, and change time.
- **touch –t :** set access and modification time of a file to a particular date by using **t** option followed by datetime.
- Touch file1 file2 file3
- Touch file 5 (does it work?)(the solution)
- touch file_name{1..3}.txt
- touch -t 201903081047.30 file_name.txt

Rm command

- **Rm : remove forever :** The rm command in Linux OS is used to remove files and directories from the command line.
- **rm -i** will ask before deleting each file.
- **rm -r** will recursively delete a directory and all its contents .
- **rm -f** will forcibly delete files without askingrm testfile
- rm ~/Documents/testfile
- rm testfile1 testfile2 testfile3
- rm -v -i testfile
- rm -v -f testfile
- rm -v -r test_directory
- sudo rm -v *
- rm -v *.txt
- sudo rm -v -d test_dircetory1 / **Remove Empty Directories**
- rm -v -r /
- rm -v -r --no-preserve-root /
- rm -v user*
- rm -v sample[1234].list

Cp Command

- **copy one file :**To copy a file, use **cp** with a source and a target argument.

cp file1.txt file2.txt

cp file42 file42.copy

- **copy to another directory** : the source files are copied to that target directory.

cp a.txt dir42

- **cp –r:**To copy complete directories, use **cp -r** (the **-r** option forces **recursive** copying of all files in all subdirectories).

cp -r dir42/ dir33

- **copy multiple files to directory** You can also use cp to copy multiple files into a directory. In this case, the last argument must be a directory.

cp a b.txt cc dir42/

- **cp –I** To prevent **cp** from overwriting existing files, use the **-i** (for interactive) option.

cp -i a.txt file42

cp -r logs1 logs2

Mv Command

- **rename files with mv :** Use **mv** to rename a file or to move the file to another directory.
- **rename directories with mv :**The same **mv** command can be used to rename directories.
- **mv –i** The **mv** also has a **-i** switch similar to **cp** and **rm**.

Useful commands

- Reboot
- Exit
- shutdown

date command

- displays and sets the system date and time. This command also allows users to print the time in different formats and calculate future and past dates.
- date [option]... [+format]
- Date
- date -d "2022-11-22 09:10:15“
- date +"Year: %Y, Month: %m, Day: %d“
- date "+DATE: %D%nTIME: %T“ (try this format)

- **Set or Change Date in Linux**
- date --set="20100513 05:30“
- date --date="4 day“
- **Display Last Modified Timestamp of a Date File**
- date -r /etc/hosts

Which command

- The **which** command allows users to search the list of paths in the **$PATH** environment variable and outputs the full path of the command specified as an argument.
- which -a touch
- which ls mount sort

Cal command

- **cal** command is a calendar command in Linux which is used to see the calendar of a specific month or a whole year.
- cal [[month] year]
- **Cal**
- **cal –y**
- **cal 08 2000**
- **cal 2018 | more**

Install updates via apt-get

1.apt-get update : First, you use the update option to resynchronize the package index files from their sources on Ubuntu Linux via the Internet.

-sudo apt-get update

-or sudo apt update

2.apt-get upgrade : Second, you use the upgrade option to install the newest versions of all packages currently installed on the Ubuntu system. In other words, get security updates for your machine.

- sudo apt-get upgrade
- or sudo apt upgrade

3.sudo apt-get install package-name : Install is followed by one or more packages desired for installation. If package is already installed it will try to update to latest version.

- sudo apt-get install foo

Man command

- ***man*** command in Linux is used to display the user manual of any command that we can run on the terminal.
- 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 [OPTION]... [COMMAND NAME]...
- man printf

Man command cont.

- $man date
- /(search for word in the man pages)(n/N to transfer among words,g/G first and last page)
- Man is divided in 9 chapters (1: program or shell commands,5:configuration ,8:command belongs to root)

$man 1 passwd

man 5 passwd

$man 9 passwd

$man cat

Man command cont.

- Man print (no command found)
- Man –k “print files”
- Man –K “print files” all command man page
- Man –K time|grep 1
- Note : you can try **info** command , what is the difference?

Working with File Contents

- **Head**
- **Tail**
- **cat**
- **tac**
- **more**
- **less**

Head

- **Head :** You can use **head** to display the first ten lines of a file.
- **Head -**n :The **head** command can also display the first **n** lines of a file.
- **Tail** :display the last ten lines of a file.
- **tail -**n :The **tail** command can also display the last **n** lines of a file.

Cat Command

- **Cat :** use cat to display a file on the screen.
- **Concatenate :cat** is short for **concatenate**. One of the basic uses of **cat** is to concatenate files into a bigger (or complete) file.
- **create files :**You can use **cat** to create flat text files. Type the **cat > winter.txt** Then type one or more lines, finishing each line with the enter key. After the last line, type and hold the Control (Ctrl) key and press d.
- **custom end marker** You can choose an end marker for **cat** with **<<** .
- example s:

$ **cat part1 part2 part3 >all**

$ **cat all**

cat > hot.txt <<stop

> It is hot today!

> Yes it is summer.

> stop

cat hot.txt

It is hot today!

Yes it is summer.

Copy Files.

cat winter.txt

It is very cold today!

cat winter.txt > cold.txt

cat cold.txt

It is very cold today!

tac

The 'tac' command is **the reverse of the 'cat' command**.

$ **cat count**

one

two

three

four

$ **tac count**

four

three

two

one

more and less

- The **more** command is useful for displaying files that take up more than one screen. More will allow you to see the contents of the file page by page.
- Use the space bar to see the next page, or **q** to quit. Some people prefer the **less** command to **more**.

Try: ls –lR \

What is the sloution?

Arguments :
Echo Command

- The **echo :** command is very simple: it echoes the input that it receives.
- Example :

```
$ echo Hello World
Hello World
$ echo   Hello     World
Hello World
$ echo 'A line        with single        quotes'
A line with      single      quotes
```

echo and quotes

- Quoted lines can include special escaped characters recognised by the **echo** command
- The example below shows how to use **\n** for a newline and **\t** for a tab (usually eight white spaces).

```
echo -e "A line with \na newline"
A line with
a newline
$ echo -e 'A line with \na newline'
A line with
a newline
$ echo -e "A line with \ta tab"
A line with	a tab
$ echo -e 'A line with \ta tab'
A line with	a tab
```

commands

external or builtin commands ?

- Not all commands are external to the shell, some are **builtin**. **External commands** are programs that have their own binary and reside somewhere in the file system.
- Many external commands are located in **/bin** or **/sbin**. **Builtin commands** are an integral part of the shell program itself.

Linux Command Types (Internal and External)

- Internal commands are the shell built-in commands.
- External commands are files present in the $PATH.
- Commands pwd,cd, echo comes under the category of the internal commands.
- ls, cp external commands
- To check whether is internal or external use **type**

Type command

- To find out whether a command given to the shell will be executed as an **external command** or as a **builtin command**, use the **type** command.
- Example :

type cd

cd is a shell builtin

type cat

cat is /bin/cat

As you can see, the **cd** command is **builtin** and the **cat** command is **external**.

running external commands

- Some commands have both builtin and external versions. When one of these commands is executed, the builtin version takes priority. To run the external version, you must enter the full path to the command.

$ type -a echo

echo is a shell builtin

echo is /bin/echo

$ /bin/echo Running the external

Which Command

- The **which** command will search for binaries in the **$PATH** environment variable (variables will be explained later). In the following, it is determined that **cd** is **builtin**, and **ls,**

cp, rm, mv, mkdir, pwd, and **which** are external commands.

which cp ls cd mkdir pwd

/bin/cp

/bin/ls

/usr/bin/which: no cd in (/usr/kerberos/sbin:/usr/kerberos/bin:...

/bin/mkdir

/bin/pwd

aliases

- In Linux, **an alias is a shortcut** that references a command. An alias replaces a string that invokes a command in the Linux shell with another user-defined string.
- Aliases are mostly used to replace long commands, improving efficiency and avoiding potential spelling errors. Aliases can also replace commands with additional options, making them easier to use.

alias c='clear‘

alias move='mv -i'

alias frename='Example/Test/file_rename.sh'

aliases

- **abbreviate commands**
- An **alias** can also be useful to abbreviate an existing command.

$ alias ll='ls -a'

$ alias c='clear'

alias :default options

- Aliases can be used to supply commands with default options. The example below shows how to set the **-i** option default when typing **rm**.

$ alias rm='rm -i'

$ rm winter.txt

rm: remove regular empty file `winter.txt'? No

unalias

- You can undo an alias with the **unalias** command.

```
$ which rm
/bin/rm
$ alias rm='rm -i'
$ which rm
alias rm='rm -i'
/bin/rm
$ unalias rm
$ which rm
/bin/rm
```