| -> use to consecutive commands

/dev/null -> black hole in Linux

‘’

System Directory Structure

![system directory structure](<https://thesagediary.files.wordpress.com/2018/09/linuxfile.png?w=1080>)

/ – The Root Directory

/bin – Essential User Binaries

/boot – Static Boot Files

/cdrom – Historical Mount Point for CD-ROMs

/dev – Device Files

/etc – Configuration Files

/home – Home Folders

/lib – Essential Shared Libraries

/lost+found – Recovered Files

/media – Removable Media

/mnt – Temporary Mount Points

/opt – Optional Packages

/proc – Kernel & Process Files

/root – Root Home Directory

/run – Application State Files

/sbin – System Administration Binaries

/selinux – SELinux Virtual File System

/srv – Service Data

/tmp – Temporary Files

/usr – User Binaries & Read-Only Data

/var – Variable Data Files

[more detail](<https://www.howtogeek.com/117435/htg-explains-the-linux-directory-structure-explained/>)

echo -e “<text>” > <file name>

echo -e “<text>” >> <file name>

ls -> list directory contents

-l – permissions owner and size (long listing format)

-d – list directories

-a – list hidden files

\*.< extension> - list entries with <extension>

<name> - list entries with <name>

-lt – the most recent is on top

-ltr – the oldest is on top

Permissions

Reference https://www.guru99.com/file-permissions.html

Type- 1 |Owner – 3 |group-3 |others-3

**r** = read permission |**w** = write permission |**x** = execute permission |**-** = no permission

Type – d directory | - file | l link

[Permissions](<https://www.guru99.com/images/permission(1).png>)

User Denotations

user/owner……u

group…………….g

other.……………o

all………………….a

Operator Description

+ Adds a permission to a file or directory

- Removes the permission

= Sets the permission and overrides the permissions set earlier.

chmod <permissions> <target>

chmod u=rwx,g=rwx,o= <file>

chmod 770 <file>

* Minimum require to access a directory is read and executable otherwise will throw a error
* chown – change file owner and group – chown <root> <owner> -> owner to root
* chgrp – change the group of each <FILE> to <GROUP>. Example chgrp <GROUP> <FILE>
* use ls -la to see the modifications

ln – link

ln -s <path/destination – target> <name shortcut>

ln -s /bin executables

-s – symbolic link, do a symbolic links instead of hard links

readlink – pathname of link value

cat – concatenate files and print in standard output

cat – n <file> -> enumerate each line

cat -n = nl (alias)

-v -> show special signals

nl -s ‘<string>’ <FILE> -> count how many times appear that word

head -<N> <FILE>

head -1 <FILE> -> First line

N -> number de lines to display

wc - > count words, lines and characters in files

wc <OPTIONS> <FILE>

wc <FILE> -> display counter of words, lines and characters

tree – recursive list the content of directory

tree <DIRECTORY> -> display a useful and friendly structure of content

mkdir-> create a directory

-m <PERMISSION> - create a directory with default permission

-p – create the parents too

cd – change directory

cd <DIR>

.. back in directory

~ -> home

$Home -> home

/ -> root

cd - ->previous location

pwd – print full filename of current working directory

* It’s very useful in shell scripting

mktemp – create temporary file and return the full path

mktemp <OPTIONS>

-d – create a directory

rm -> remove

-r – recursive

-rf – recursive and force without asking

mv – move a file or rename

mv <TARGET> <DESTINATION>

cp – copy a file

cp <TARGET> <NAME> <DIRECTORY>

cp test.txt new.txt ./dir/

touch – important to change date/time of files

touch -t 201911271000.00 <FILE> -> create or change a date

touch `date “+%Y%m%d”` <FILE>

date “+%Y%m%d” -> date in this format

tee <FILE> ->create a file and show the result

Channing commands => | (pipe)

Example:

* ls -la | tee test.txt
* ls -ld |wc -l

grep – search for a pattern in each file

grep <OPTIONS> <VALUE> <LOCATION>

-c – repeat times

-I – inactive case-sensitive

-v – opposite of search

-l – name of file where the string was found

-ln – prints the line number where the string was found

grep test ./\*

grep -v test \*

ls -la | grep “<REGEX>”

sort – sort lines of a text file

-u – remove duplicates

-r – reverts

-f – sort is case insensitive

-t’<DELIMITER>’ – separator

-k<column1>,<column2>

Find – search for a file

Find <FOLDER> -name ”<REGEX>”

-mtime +/- <NUMBER>

-type f/d

-prune

-perm <PERMISSIONS>

-quit – stop in first match

-print – display a result in standard output

-exec <command>

find . -iname “sample\*” -exec programa.sh {}\;

find . -iname “sample\*” -exec `ls -la` {}\;

{} -> arguments

\; -> each one – slowest

\+ -> all -> faster

find . -name “\*” -type f -mtime +360 -exec `ls -l` {}\+ -print

time -> spend time to execute the task

xargs – build and execute command lines from standard input

time find . -name “\*.exe\*” | xargs ls -l

cut – remove sections of file

-d<DELIMITER> - separator

-f<COLUMN1…>

cut –d‘ ’ -f2,3,4,5,6

paste -d’;’ <FILE> <FILE> |more

diff -> diference between files

-w – ignore whites spaces

-y – display differences with a new representation

gzip

-d – unzip

-v Resume of compress

gzip -d = gunzip

gunzip

zcat -> allow see content of zip file

gzip -c file > name.gz

tar

-cf

-tvf

-xvf

df – shows the free spaces

du – shows used space

-p user friendly

-H/h -> display in GB/Mb, default Kb