```
# becoming root temporarily in the terminal (available only if root has a password set)
su        # => enter the root password
```

Linux Paths

Paths:
- absolute
- relative

Any absolute path starts with /

```
.           # => the current working directory
..          # => the parent directory
~           # => the user's home directory

cd          # => changing the current directory to user's home directory
cd ~        # => changing the current directory to user's home directory
cd -        # => changing the current directory to the last directory
cd /path_to_dir         # => changing the current directory to path_to_dir
pwd                     # => printing the current working directory

# installing tree
sudo apt install tree

tree directory/         # => Example: tree .
tree -d .               # => print only directories
tree -f .               # => print absolute paths
```

The ls Command

```
ls [OPTIONS] [FILES]
# ~ => user's home directory
# . => current directory
# .. => parent directory

# listing the current directory
ls
ls .

# listing more directories
ls ~ /var /
```

-l => long listing
ls -l ~

-a => listing all files and directories including hidden ones
ls -la ~

-1 => listing on a single column
ls -1 /etc

-d => displaying information about the directory, not about its contents
ls -ld /etc

-h => displaying the size in human readable format
ls -h /etc

-S => displaying sorting by size
ls -Sh /var/log

Note: ls does not display the size of a directory and all its contents. Use du instead
du -sh ~

-X => displaying sorting by extension
ls -lX /etc

--hide => hiding some files
ls --hide=*.log /var/log

-R => displaying a directory recursively
ls -lR ~

-i => displaying the inode number
ls -li /etc

File Timestamps and Date

---

displaying atime
ls -lu

displaying mtime
ls -l
ls -lt

displaying ctime

```
ls -lc

# displaying all timestamps
stat file.txt

# displaying the full timestamp
ls -l --full-time /etc/

# creating an empty file if it does not exist, update the timestamps if the file exists
touch file.txt

# changing only the access time to current time
touch -a file

# changing only the modification time to current time
touch -m file

# changing the modification time to a specific date and time
touch -m -t 201812301530.45 a.txt

# changing both atime and mtime to a specific date and time
touch -d "2010-10-31 15:45:30" a.txt

# changing the timestamp of a.txt to those of b.txt
touch a.txt -r b.txt

# displaying the date and time
date

# showing this month's calendar
cal

# showing the calendar of a specific year
cal 2021

# showing the calendar of a specific month and year
cal 7 2021

# showing the calendar of previous, current and next month
cal -3

# setting the date and time
date --set="2 OCT 2020 18:00:00"

# displaying the modification time and sorting the output by name.
```

```
ls -l

# displaying the output sorted by modification time, newest files first
ls -lt

# displaying and sorting by atime
ls -ltu

# reversing the sorting order
ls -ltu --reverse
```

Viewing files (cat, less, more, head, tail, watch)

```
# displaying the contents of a file
cat filename

# displaying more files
cat filename1 filename2

# displaying the line numbers
can -n filename

# concatenating 2 files
cat filename1 filename2 > filename3

# viewing a file using less
less filename
```

less shortcuts:

```
h          => getting help
q          => quit
enter      => show next line
space       => show next screen
/string    => search forward for a string
?string    => search backwards for a string
n / N      => next/previous appearance

# showing the last 10 lines of a file
tail filename

# showing the last 15 lines of a file
tail -n 15 filename
```

showing the last lines of a file starting with line number 15
tail -n +5 filename

showing the last 10 lines of the file in real-time
tail -f filename

showing the first 10 lines of a file
head filename

showing the first 15 lines of a file
head -n 15 filename

running repeatedly a command with refresh of 3 seconds
watch -n 3 ls -l

Working with files and directory (touch, mkdir, cp, mv, rm, shred)

---

creating a new file or updating the timestamps if the file already exists
touch filename

creating a new directory
mkdir dir1

creating a directory and its parents as well
mkdir -p mydir1/mydir2/mydir3

The cp command

copying file1 to file2 in the current directory
cp file1 file2

copying file1 to dir1 as another name (file2)
cp file1 dir1/file2

copying a file prompting the user if it overwrites the destination
cp -i file1 file2

preserving the file permissions, group and ownership when copying
cp -p file1 file2

being verbose
cp -v file1 file2

recursively copying dir1 to dir2 in the current directory
cp -r dir1 dir2/

copy more source files and directories to a destination directory
cp -r file1 file2 dir1 dir2 destination_directory/

The mv command

renaming file1 to file2
mv file1 file2

moving file1 to dir1
mv file1 dir1/

moving a file prompting the user if it overwrites the destination file
mv -i file1 dir1/

preventing a existing file from being overwritten
mv -n file1 dir1/

moving only if the source file is newer than the destination file or when the destination file is missing
mv -u file1 dir1/

moving file1 to dir1 as file2
mv file1 dir1/file2

moving more source files and directories to a destination directory
mv file1 file2 dir1/ dir2/ destination_directory/

The rm command

removing a file
rm file1

being verbose when removing a file
rm -v file1

removing a directory
rm -r dir1/

```
# removing a directory without prompting
rm -rf dir1/

# removing a file and a directory prompting the user for confirmation
rm -ri fil1 dir1/

# secure removal of a file (verbose with 100 rounds of overwriting)
shred -vu -n 100 file1
```

Piping and Command Redirection

Piping Examples:

```
ls -lSh /etc/ | head                  # see the first 10 files by size
ps -ef | grep sshd                    # checking if sshd is running
ps aux --sort=-%mem | head -n 3    # showing the first 3 process by memory consumption
```

Command Redirection

```
# output redirection
ps aux > running_processes.txt
who -H > loggedin_users.txt

# appending to a file
id >> loggedin_users.txt

# output and error redirection
tail -n 10 /var/log/*.log > output.txt 2> errors.txt

# redirecting both the output and errors to the same file
tail -n 2 /etc/passwd /etc/shadow > output_errors.txt 2>&1

cat -n /var/log/auth.log | grep -ai "authentication failure" | wc -l
cat -n /var/log/auth.log | grep -ai "authentication failure" > auth.txt        # => piping and redirection
```

Finding Files (find, locate)

---

locate

```
# installing plocate
sudo apt install plocate

# updating the locate db
sudo updatedb

# displaying statistics
locate -S

# finding file by name
locate filename          # => filename is expanded to *filename*
locate -i filename       # => the filename is case insensitive
locate -b '\filename'    # => finding by exact name

# finding using the basename
locate -b filename

# finding using regular expressions
locate -r 'regex'

# checking that the file exists
locate -e filename

# showing command path
which command
which -a command
```

find

```
find PATH OPTIONS
```

Example: `find ~ -type f -size +1M` `# => finding all files in ~ bigger than 1 MB`

Options:

```
 -type f, d, l, s, p
 -name filename
-iname filename     # => case-insensitive
 -size n, +n, -n
 -perm permissions
 -links n, +n, -n
 -atime n, -mtime n, ctime n
```