

# Linux commands

- `whoami` - print the user name currently logged in to the terminal session
- `man` - get the manual of the command
- `pwd` - print current working directory
- `ls` - list all the files inside a folder. You can add a path as an option.
  - `-l` gives a lot of other info about files (their size, permissions, etc.)
  - `-a` does not ignore files starting with ".".
- `help` - provides help for a command. Much like `man` but its for commands build into the shell like `cd`.
- `cd` - change directory
- `mkdir` - creates new directory. Can take multiple names.
  - `-p` - makes it possible to create nested directories. ex.: `mkdir -p fruits/apples`. With `-p` `mkdir` doesn't throw an error when the directory already exists.
- `touch` - create an empty file, or updates the modification time of a file if it already exists
  - `-d <date>` - selects another date
  - `--reference <file>` - copies date from another file
  - `-a` - changes access time instead.
  - `-c` - changes change time instead.
  - `-m` - changes modification time instead.
- `rmdir` - delete a directory, only works if the folders are empty. Can take multiple names.
- `rm` - removes files or optionally directories. Can take multiple names.
  - `-r` - to remove a directory, empty or not

- `-i` - for interactive mode. Asks us if we want to delete each and every file separately
- `-f` - to force file removal. It works even if a file does not have write permissions.
- `mv <old_location> <new_location>` - move a file, or rename a file. Can take multiple files, as long as the last parameter is the destination folder. Can also move and rename folders.
- `cp <file> <copy_name>` - copies a file. Can copy multiple files, if the last argument is a directory.
  - `-r` - copies a directory recursively.
  - `-a` - preserves the dates of modifications, permissions and copies directories recursively.
- `head` - outputs the first 10 lines of a file.
  - `-n <number>` - specify the number of lines.
  - `-c <number>` - specify the number of bytes.
- `tail` - outputs the last 10 lines of a file
  - `-n <number>` - specify the number of lines.
  - `-c <number>` - specify the number of bytes.
  - `-f` - keep printing new lines appended to a file as its being modified
- `date` - print the current date and time.
- `cat` - prints file's content or concatenates multiple files.
  - `-n` - output line numbers
  - `-v` - output non-printable characters.
- `less` - read a file in a nice interactive UI.
- `echo` - prints something on the screen, ends with `\n`.
  - `-n` - omits the newline
  - `-e` turns backslash escapes into characters.

- `wc` - word count, returns the lines, words and bytes in that order.
- `sort` - sorts lines in a file. It does not modify its contents, it only outputs the sorted list alphabetically (not numerically). Can also take a std output.
  - `-n` - sorts it numerically.
  - `-r` - reverse order.
  - `-u` - only get the unique values.
  - `-h` - sorts by numbers in human readable format, i.e. with units and decimals.
  - `-k<columns>` - sorts by columns. When `columns` is a single number it sorts by `columns+1` , `columns+2` , `columns+3` ...When it is two comma separated numbers it sorts by the first column to another.
  - `-s` - stable sort
- `uniq` - omits repeated lines. Only removes adjacent duplicates.
  - `-d` - only print out the lines that have duplicates.
  - `-u` - only print the lines that have no duplicates.
  - `-c` - count.
- `diff <file1> <file2>` - show differences between files
  - `-y` - show side by side comparison
  - `-u` - show git like output
- `find .` - finds all matching files in the current directory and deeper. Substitute `'/'` for a path if you want to search somewhere else
  - `-name <name>` - looks by the name of the file. `<name>` could be a regular expression.
  - `-type <name>` - looks by the type of the file. d — directory, f — files
  - `-iname` - looks by the name, case insensitive.
  - `-or` - no need for explanation. The order of switches matters in this context.
  - `-not` - again, self explanatory.

- `-size` - looks by the size of the file. `-size +100k` looks for files more than 100Kb and `-size -1M` looks for files less than a megabyte.
- `-mtime` - searches by date of modification. `-mtime +3` searches for files modified more than 3 days ago.
- `grep <text> <file>` - finds text inside a file
  - `-n` - get line numbers.
  - `-C <number>` - get `number` of lines of context before a match and `number` of lines after.
  - `-r` - recursive, looks inside the current directory instead of a single file. The second parameter should be a directory name instead of a file, or when omitted, it defaults to the current directory.
  - `-i` - case insensitive.
  - `-v` - find non-matching lines instead.
- `du` - find the sizes of files in a directory.
  - `-m` - sizes in megabytes.
  - `-h` - human readable size. Gives size in proper units
- `df` - gives disc usage.
- `history` - get history of typed commands. It will give you line numbers and commands. To run a command again type `!<line number>`.
- `ps` - inspect running processes started by currently logged in user. Use `ps ax` to see every process.
  - `-e` shows every process
  - `l|-f|-F` chooses long, full or extra full output format
  - `L|-T` includes the threads in results
- `pstree` - shows a tree of processes.
  - `-p` - expands by pid.
  - `-u` - expands by user name.
  - `-a` - expands by arguments.

- `top` - show top most processor intensive processes.
- `pgrep` - matches process name with a regex.
  - `-f` - matching regex with name and arguments
- `kill <PID>` - sends signals to processes.
  - `-l` - list all signals
  - `-TERM` - terminate. The default.
  - `-<number>` - where `number` is the number of the signal
- `pkill <regex>` - works just like `pgrep` but will also send a signal when matched.
  - The same as for `kill`.
- `killall <name>` - kill a process by a name. If multiple processes match, it kills them all.
  - All the switches from `kill`
- `jobs` - see current jobs in this window.
- `bg <job number>` - resume a job in background. To run a command in background add a `&` at the end.
- `fg <job number>` - resume a job in foreground.
- `ln <original> <link>` - to create a hard link.
  - `-s` - create a softlink.
- `who` - who is logged in.
- `su <user>` - switch users.
- `sudo <command>` - run commands as root user.
- `passwd` - change someone's password. By default changes your password.
- `chown <new owner> <file>` - change ownership of a file or directory.
  - `-R` - change ownership recursively.
- `chmod <mode> <file>` - change permissions of a file.
  - `-R` - change permissions recursively

- `--reference <file>` - to copy permissions from a reference file.
- `ssh user@host` - to login into a remote shell.
- `whatis <pagename>` - to see which sections contain given page name. To then display a manual for a specific page use `man <section> <pagename>`.
- `groups <name>` - lists all groups a user belongs to.
- `stat file` - to see details about a file including permissions.
- `umask [octal num]` - display and change default permissions for a file.
- `paste <file>...` - outputs files like zip in python, separated by a TAB, until the longest one ends.
- `printf <format> <args>` - works almost like `printf` in C.
- `seq [from [seq]] to` - prints numbers from `from` until it reaches `to`, and increments by `step`.
  - `-w` - makes numbers equal width.
- `cut` - output only selected characters.
- `tr` - replaces or deletes characters.
  - `-d LIST` - deletes all characters from the `LIST`.
  - `FROM TO` - translates characters from list `FROM` to list `TO`.
- `locate` - searches files from database.
- `alias word=value` - to set an alias. Without arguments it lists aliases.
- `unalias word` - to unset an alias.