
Linux Commands - Complete Notes

When we type something in the terminal in Linux, it will try to search for that command. If it's not found, it will show an error.

Reference: [The Linux Commands Handbook]

You can use `help` before a command to get basic information about it.

List of Common Commands

1. whoami
2. man
3. clear / Ctrl+L
4. pwd
5. ls
6. cd
7. mkdir
8. touch
9. rmdir
10. rm
11. open
12. mv
13. cp
14. head
15. tail
16. date
17. >
18. >>
19. cat
20. less
21. echo
22. wc
23. wc
24. | (pipe)
25. sort
26. uniq
27. expansions
28. diff
29. find
30. grep
31. du
32. df
33. history
34. ps
35. top
36. kill
37. file permissions
38. chmod
39. groups
40. -
41. -
42. -
43. -
44. -
45. killall
46. jobs, bg, fg
47. gzip
48. unzip
49. tar
50. nano
51. sleep
52. alias
53. xargs
54. ln
55. who
56. su
57. sudo
58. apt install
59. passwd
60. chown

Detailed Notes

man

- Manual for a command.
- Use: `man <command>`
- Use **space** to scroll. Square brackets in synopsis (`[]`) indicate optional arguments.

clear

- Clears the screen, but you can't scroll back.
- `clear -x` clears the screen but allows scrolling back.

pwd

- Shows the present working directory.

ls

- Lists files and directories.
- Use `/` after directory names (e.g., `ls Folder/`).
- Use full paths if needed.

Options:

- `-l` : Long listing format.
- `-a` : Show hidden files (starting with `.`).
- Combined: `ls -la`

cd

- Changes directories.
- `~` represents `/home/username`
- Example: `cd ~/Downloads`

mkdir

- Creates directories.

Use `-p` for nested directory creation.

```
mkdir -p winter/seeds/lettuce
```

-

touch

- Creates empty files or updates timestamps.
- Can create multiple files: `touch file1 file2`

rmdir

- Deletes empty directories only.

rm

- Deletes files and directories.
- No recycle bin!

Options:

- `-v`: Verbose.
- `-r`: Recursive (for non-empty folders).
- `-i`: Interactive (asks confirmation).
- `-ri`: Both recursive and interactive.

open / xdg-open

- Opens a file with the default program.

`open` for macOS, `xdg-open` for Ubuntu:

```
xdg-open .
```

-

mv
Moves or renames files.

```
 mv old_name new_name
mv file1 file2 folder/
```

-

cp
Copies files.

```
 cp file1 file2
cp -r folder1 folder2
```

-

head / tail
`head` shows the first 10 lines. Use `-n` to specify:

```
 head -n 20 file.txt
```

-

`tail` shows the last 10 lines. Use `-f` for real-time:

```
 tail -f log.txt
```

-

date

- Displays current date and time.

> / >>

- `>` overwrites content in a file.
- `>>` appends content.

cat

- Read file content.

Can concatenate files.

```
 cat file1 file2 > combined.txt
```

-

less

- View file content page-wise.

echo
Prints text to terminal or writes to a file.

```
echo "Hello" > file.txt
```

-

wc

- Word count:
    - `-l`: lines
    - `-w`: words
    - `-m`: characters

Piping (|)
Sends output of one command to another.

```
ls | wc -l
```

-

sort

- Alphabetical or numerical sort.
- Use `-n` for number sort, `-u` for unique, `-r` for reverse.

uniq

- Removes adjacent duplicates.
- Often used with `sort`.
- Options:
    - `-d`: Duplicates only.

    - `-u`: Unique only.
    - `-c`: Count occurrences.

Expansions

- `~`: Home directory
- `$PATH`, `$USER`: Environment variables
- `*`, `*.txt`, `*.??`: Wildcards

`{}`: Combinations

touch file{1..5}.txt

-

diff

- Compares files line by line.
- Output uses:
    - `a`: Added
    - `d`: Deleted
    - `<` or `>` to show changes

find
Search for files/directories.

find . -name '*.py'

-

Options:

- `-type f`: Files
- `-type d`: Directories

- `-iname`: Case-insensitive
- Combine with `-or`, `-not`, etc.

grep
Searches for text patterns in files.

grep "green" song.txt

-
- `-n`: Shows line numbers.
- `-C 2`: Shows 2 lines before and after match.
- `-r`: Recursive search.

du

- Disk usage of files/directories.

`-h`: Human-readable

du -h | sort -h | tail

-

df

- Shows disk space info.
- Use `df -h`

history

- Shows terminal command history.
- Re-run command with `!<number>`

ps / top

- `ps ax`: Process list

- `top`: Real-time system usage

Use `grep` to filter:

```
ps ax | grep "chrome"
```

-

kill / killall

- Kill processes by PID or name.

`SIGTERM(15)` for soft kill, `SIGKILL(9)` for force kill.

```
kill -9 <pid>
killall <program>
```

-

jobs, fg, bg

- `jobs`: Lists background jobs.
- `fg`, `bg`: Bring jobs to foreground/background.

gzip / unzip

- Compress with `gzip`, decompress with `gzip -d`
- `gzip -k file` keeps original file

tar

Archive multiple files:

```
tar -cf archive.tar file1 file2
tar -czf compressed.tar.gz file1 file2
tar -xf archive.tar
```

-

nano

- Terminal text editor.

    - `Ctrl+O`: Save
    - `Ctrl+X`: Exit
    - `Ctrl+W`: Search
    - `Ctrl+K`: Cut
    - `Ctrl+U`: Paste

sleep
Delays the execution.

sleep 5

-

alias
Create shortcuts for commands.

alias ll='ls -al'

-
- Make permanent in `~/.bashrc` and run `source ~/.bashrc`.

xargs
Converts standard input into command arguments.

cat files.txt | xargs rm

-

ln

- Creates links (shortcuts).
    - Hard link: `ln original.txt link.txt`
    - Soft link: `ln -s original.txt link.txt`

who

- Lists users currently logged in.

su
Switch user.

```
su <username>
```

-
- Exit with `exit`.

sudo

- Run command as root (admin privileges).

apt install

- Install packages using APT (Debian/Ubuntu).

passwd

- Change password.
- Admins can change others' passwords.

chown
Change file ownership.

```
sudo chown user file.txt
sudo chown -R user folder/
```

-

groups

- Lists groups a user belongs to.

Change group ownership:

```
sudo chown user:group file
```

-

File Permissions

- File types:

    - `-`: file
    - `d`: directory
- Permissions:
    - 3 sets: owner, group, others
    - `rwx`: read, write, execute

chmod

- Change permissions.

Symbols:

- `u`: user
- `g`: group
- `o`: others
- `a`: all

Operators:

- `+`: add
- `-`: remove
- `=`: exact

Examples:

```
chmod g+w file.txt
chmod a-rw file.txt
chmod 755 file.txt  # rwxr-xr-x
```
