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

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
```

Some more notes

Linux Commands Review

1. head

Displays the **first 10 lines** of a file by default.

- **Syntax:**
`head [options] [file]`
- **Common options:**
 - `-n <number>` : Show the first *n* lines

Examples:

```
head filename.txt
head -n 5 filename.txt
```

-

2. tail

Displays the **last 10 lines** of a file by default.

- **Syntax:**
`tail [options] [file]`
- **Common options:**
 - `-n <number>` : Show the last *n* lines
 - `-f` : Continuously monitor the file (useful for logs)

Examples:

```
tail filename.txt
tail -n 20 filename.txt
tail -f /var/log/syslog
```

-

3. **cat**

Used to **view** file contents, **concatenate** files, and **create** files.

- **Syntax:**
`cat [options] [file(s)]`
- **Common uses:**
 - Display file: `cat file.txt`
 - Combine files: `cat file1.txt file2.txt > merged.txt`
 - Create file: `cat > newfile.txt` (type content then **Ctrl+D**)

Example:

```
cat sample.txt
```

-

4. **wc** (Word Count)

Counts **lines**, **words**, and **characters** in a file.

- **Syntax:**
`wc [options] [file]`
- **Options:**
 - `-l` : lines
 - `-w` : words
 - `-c` : bytes

Example:

```
wc -l file.txt  
wc -w file.txt  
wc file.txt
```

-

5. Piping (|)

Sends the output of one command as the input to another.

- **Syntax:**
`command1 | command2`

Example:

```
cat file.txt | wc -l  
ps aux | grep apache
```

-

6. sort

Sorts the content of a file **alphabetically** or **numerically**.

- **Syntax:**
`sort [options] [file]`
- **Options:**
 - `-n` : numerical sort
 - `-r` : reverse
 - `-k` : sort by column

Examples:

```
sort names.txt  
sort -n numbers.txt  
sort -rk 2 file.txt
```

-

7. uniq

Removes **duplicate** lines from sorted data.

- **Syntax:**
`uniq [options] [file]`
- **Options:**

- `-c` : counts occurrences
- `-d` : show only duplicate lines

Example:

```
sort file.txt | uniq  
sort file.txt | uniq -c
```

-
-

8. Expansions

Used to shorten commands or generate patterns.

- **Types:**

Brace expansion: `{a,b,c}` → `a b c`

```
echo file{1..3}.txt # file1.txt file2.txt file3.txt
```

-

Tilde expansion: `~` = home directory

```
cd ~
```

-

Command substitution: `$(command)`

```
echo "Today is $(date)"
```

-
-

9. `diff`

Compares two files line by line.

- **Syntax:**
`diff [options] file1 file2`

Example:

```
diff old.txt new.txt
```

-
-

10. **find**

Searches for files and directories in a directory hierarchy.

- **Syntax:**
`find [path] [options]`
- **Common options:**
 - `-name` : search by name
 - `-type` : `f` for file, `d` for directory
 - `-size`, `-mtime`, `-exec`

Examples:

```
find . -name "*.txt"
find /home/user -type f -size +10M
find . -name "test.txt" -exec rm {} \;
```

-
-

11. **jobs, bg, fg**

Manage background and suspended processes.

- **jobs**: Lists jobs in the shell.
- **bg**: Resumes a suspended job in background.
- **fg**: Brings a background job to the foreground.

Example:

```
sleep 100 &
jobs
fg %1
```

-
-

12. **gzip**

Compresses files using **GNU zip**.

- **Syntax:**
`gzip [file]`
- **To decompress:**
`gunzip file.gz`

Example:

```
gzip report.txt  
gunzip report.txt.gz
```

-

13. **tar**

Used to **archive** multiple files/directories.

- **Syntax:**
 - Create: `tar -cvf archive.tar dir/`
 - Extract: `tar -xvf archive.tar`
 - With gzip: `tar -czvf archive.tar.gz dir/`
- **Options:**
 - `-c` : create
 - `-x` : extract
 - `-v` : verbose
 - `-f` : filename
 - `-z` : gzip compression

Examples:

```
tar -cvf files.tar mydir/  
tar -xvzf backup.tar.gz
```

-

14. xargs

Builds and executes command lines from input.

- **Syntax:**
`command | xargs command2`
- **Use cases:**

With `find`:

```
find . -name "*.log" | xargs rm
```

○

With `echo`:

```
echo "file1 file2" | xargs rm
```

○

15. ln

Creates **links** between files.

- **Syntax:**
`ln [options] target link_name`
- **Types:**
 - Hard link: `ln file1 link1`
 - Symbolic (soft) link: `ln -s file1 link1`

Example:

```
ln -s /var/log/syslog mylog
```

-

16. chown

Changes **ownership** of a file or directory.

- **Syntax:**
`chown [user]:[group] file`

Example:

```
sudo chown gouri:dev file.txt
```

-
-

17. **chmod**

Changes **permissions** of a file or directory.

- **Syntax:**
`chmod [options] mode file`
- **Permission modes:**
 - Read = 4, Write = 2, Execute = 1
 - `chmod 755 file: rwxr-xr-x`
- **Symbolic form:**
 - `chmod u+x script.sh` (adds execute permission to user)

Examples:

```
chmod 644 file.txt  
chmod +x script.sh
```

-
-