Files & Directories (Core)

Linux Commands Course · Section 2

Everything is a File

In Linux, almost everything is treated as a **file** — whether it's a document, folder, device, or socket.

- Regular files → data you create (.txt, .py, .jpg)
 Directories → special files that store file lists
- Devices → /dev/sda, /dev/null
- Processes → /proc/<pid>
- Links → alternate names or shortcuts to files

Creating Files — touch

touch creates an empty file if it doesn't exist.

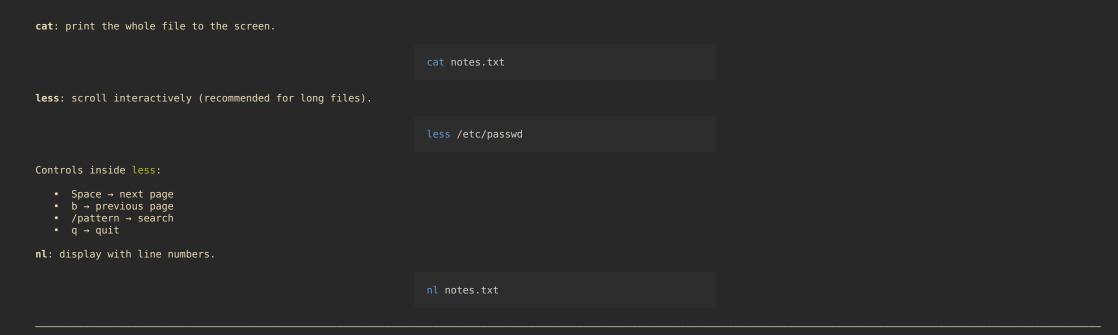
touch notes.txt

If the file exists, touch updates its modification timestamp.

You can create multiple files at once:

touch a.txt b.txt c.txt

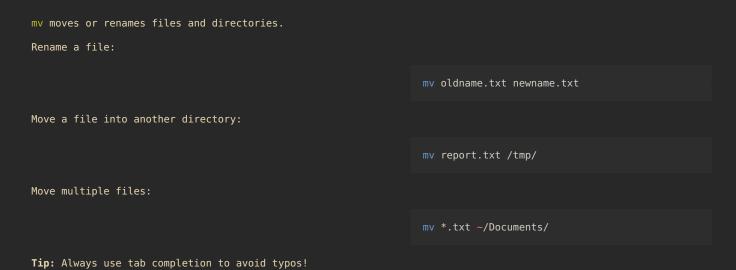
Reading Files — cat, less, nl



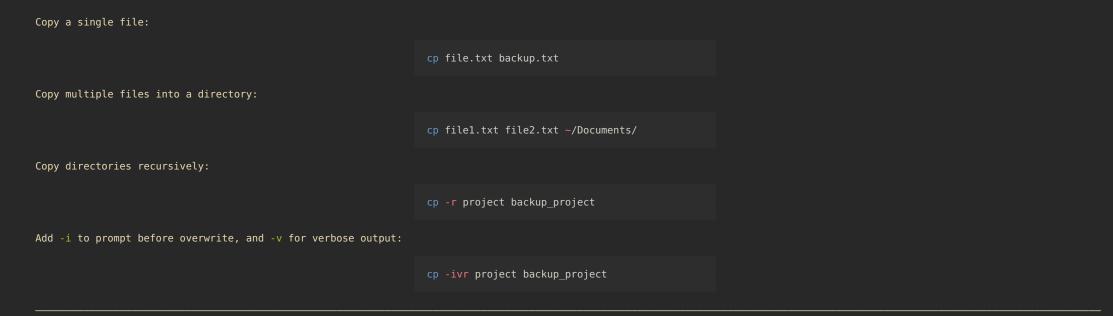
Previewing Files — head and tail



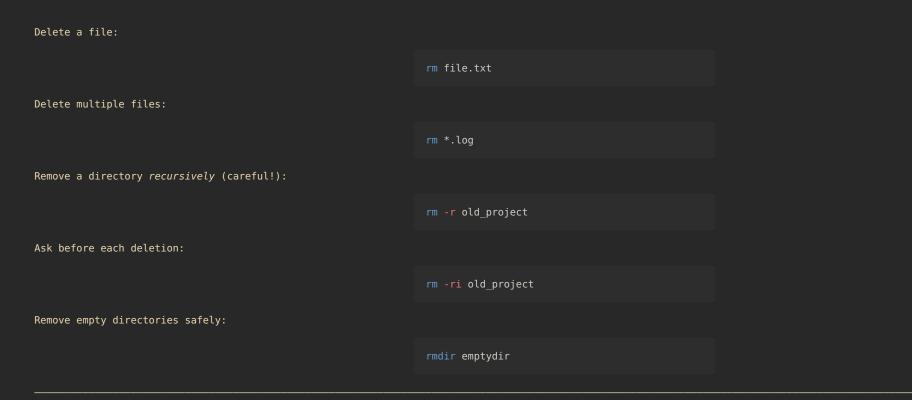
Renaming & Moving — mv



Copying Files — cp



Deleting Files & Folders — rm, rmdir



Creating Directories — mkdir

Create one directory:

mkdir projects

Create nested directories in one go:

mkdir -p projects/python/scripts

-p ensures parent folders are created if missing.

Inspecting File Metadata - stat

stat displays detailed information about a file.

stat notes.txt

Example output:

File: notes.txt
Size: 4096 Blocks: 8 IO Block: 4096 regular file
Device: 802h/2050d Inode: 1234567 Links: 1
Access: (0644/-rw-r--r--) Uid: (1000/student) Gid: (1000/student)
Access, Modify, Change times...

Shows size, type, permissions, timestamps, and inode (unique identifier).

Detecting File Type - file

Check what kind of data a file contains.

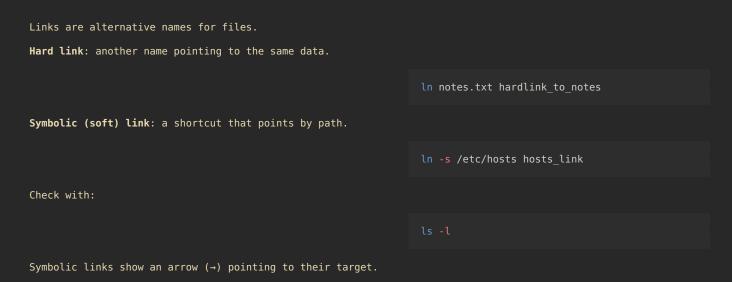
```
file /bin/bash
file photo.jpg
file script.sh
```

Output examples:

- ELF 64-bit executable (for programs)
- JPEG image dataASCII text

It's a quick way to understand what a file really is, regardless of its extension.

Links — Hard vs Symbolic



Differences Between Link Types

Feature	Hard Link	Symbolic Link
Points to Works across filesystems Affected if original deleted Shown in ls -l	file's inode (real data) X stays (until inode reused) same inode number	file path (name) ✓ breaks (dangling link) with → target path

Use symbolic links for convenience and hard links for redundancy.

Safety Tips

- Use -i (interactive) with cp, mv, and rm while learning.
 Always double-check paths before using rm -r.
 Use less instead of cat for large files.
 For log monitoring, combine tail -f with grep.

Example:

tail -f /var/log/syslog | grep "error"

Recap

```
    Create files → touch
    Read → cat, less, nl, head, tail -f
    Modify / Move → cp, mv, rm
    Directories → mkdir -p, rmdir
    Inspect → stat, file
    Links → ln, ln -s
```

These are your daily drivers for file management in Linux.

15 / 1