

\$ mv

Master mv — Interactively

Why the terminal is the fastest way to move & rename files

GO

Instant feedback

Press Enter, see it happen — no compiling.

RE

Learn by tweaking

Change one flag, rerun, and watch it change.

IO

Real files, real power

Move and rename files instantly, no copy.

OK

Zero setup

Built into every Linux system — just open a shell.

```
user@linux: ~  
$ mv draft.txt \  
  final.txt
```

Renamed!

~10 sec

01

Rename A File

Give a file a new name in one command

- `mv old new` renames it in the same folder
- No actual copying happens — it's instant
- Works on files and folders alike

SYNTAX

```
mv draft.txt final.txt
```



```
user@linux: ~  
  
$ mv draft.txt \  
  final.txt  
  
Renamed!
```

02

Move A File To A Folder

Relocate a file without leaving a copy behind

- `mv file path/` moves it into that folder
- The original location no longer has it
- Faster than `cp + rm` for the same result

SYNTAX

```
mv notes.txt archive/
```



user@linux: ~

```
$ mv notes.txt \  
archive/
```

Moved!



03

Move Without Overwriting

Avoid accidentally replacing an existing file

- `-n` skips the move if dest already exists
- Protects files you didn't mean to overwrite
- Combine with `-v` to confirm what happened

SYNTAX

```
mv -n data.csv backup/
```



user@linux: ~

```
$ mv -n data.csv \  
backup/
```

Skipped (exists)



04

Move Multiple Files At Once

Relocate several files into one folder in one go

- List several sources before the destination
- The last argument must be the target folder
- Great for batch-organizing files quickly

SYNTAX

```
mv a.txt b.txt archive/
```

user@linux: ~

```
$ mv a.txt b.txt \  
archive/
```

2 files moved



```
$ echo "Thanks for watching!"
```

Thank You For Watching

You just learned 4 real mv moves — now go run them yourself.

▶ SUBSCRIBE

@MyWebUniversity

on YouTube — for more Linux & command-line tutorials