CYPHERPUNK SCHOOL — SOVEREIGNTY SERIES

The Command Line, Part 3

Linux CLI 103 — A Faster Terminal

Once the fundamentals click, a handful of modern tools make the terminal genuinely pleasant — faster versions of what you know, and lookups so you never memorize a flag again

A practical guide to digital sovereignty

Cypherpunk School

cypherpunkschool.com

Sovereignty Series • Foundation: Builds on CLI 101 & 102 • June 2026

Contents

1. Sugar on Top of the Fundamentals

You've got the basics (*Part 1*) and the stream-and-pipe model (*Part 2*). Everything from here is optional — quality-of-life tools that make the terminal faster and friendlier. None of it replaces what you've learned; it sits *on top* of it. The plain commands are always there on any machine, and that matters when you're on a stripped-down server. Treat this chapter as a buffet: install what appeals, skip what doesn't.

The tools come in two families:

- **Better versions of what you know** — drop-in upgrades to `ls`, `cat`, `find`, `grep`, and `cd`.
- **Never memorize a flag again** — lookups that hand you the example you need instead of a 40-page manual.

> **The originals still matter.** Learn these as *conveniences*, not crutches. When you SSH into a fresh box, `ls`, `cat`, `grep`, and `man` are there; `eza` and `bat` are not. Knowing the plain tools is the skill — these just make your daily machine nicer.

2. Installing Them

Most live in the standard repositories. On Debian/Ubuntu:

```
sudo apt update
sudo apt install ripgrep bat fd-find fzf
```

Newer tools (`eza`, `zoxide`, `tealdeer`) may need a recent distro or Rust's `cargo install`; each project's README has the current method. On macOS, `brew install` covers all of them.

> **The Debian renaming trap.** To avoid name clashes, Debian/Ubuntu ship two of these under different command names: `bat` runs as `batcat`, and `fd` runs as `fdfind`. If `bat` "isn't found" right after installing, that's why. Make a friendly alias once:

```
mkdir -p ~/.local/bin
ln -s $(which batcat) ~/.local/bin/bat
ln -s $(which fdfind) ~/.local/bin/fd
# ensure ~/.local/bin is on your PATH (most distros already do this)
```

3. Better Versions of What You Know

eza — a better `ls`

Colored, git-aware listings, and a real tree view without the separate `tree` tool.

```
eza -la           # like ls -la, but readable and colored
eza --tree        # show the directory as a tree
eza -la --git     # add a column showing git status per file
```

bat — a better `cat`

`cat` with syntax highlighting and line numbers. When you pipe it somewhere, it quietly behaves like plain `cat`, so it won't break your pipelines.

```
bat config.yml          # view a file with highlighting + line numbers
bat -pp config.yml      # plain (no decorations) - good for copy/paste
```

fd — a better `find`

The same job as `find`, with sane defaults and a syntax you can actually remember.

```
fd report               # find files/dirs matching "report"
fd -e md                # only .md files
fd -H secret            # include hidden files in the search
```

ripgrep (rg) — a better `grep`

Searches file *contents* recursively, fast, and skips junk in `.gitignore` by default.

```
rg "TODO"               # find "TODO" in every file under here
rg -n "api_key"         # -n shows line numbers
rg -i "error"           # -i = case-insensitive
```

zoxide — a smarter `cd`

It remembers the directories you visit and lets you jump by a fragment of the name — no more long chains of `cd ..`. It needs one line in your shell config to hook in:

```
# add to ~/.bashrc (or ~/.zshrc), then open a new terminal:
eval "$(zoxide init bash)"

# afterwards, anywhere:
z school        # jump to the dir you visit most that matches "school"
zi school       # pick interactively if there are several
```

fzf — the fuzzy finder

A general-purpose "type a few letters, pick from a live-filtered list" tool. Its best trick is upgrading your shell's history search: press `Ctrl+R` and fuzzy-find any command you've ever run.

```
# pipe anything into fzf to pick interactively:
fd -e md | fzf          # fuzzy-pick a markdown file
nano "$(fd -e conf | fzf)"   # find a config, pick it, open it

# after enabling fzf's shell keybindings:
#   Ctrl+R  - fuzzy search command history
#   Ctrl+T  - fuzzy-pick a file into the current command
```

> **`fzf` keybindings need enabling.** The `Ctrl+R`/`Ctrl+T` widgets come from fzf's shell integration. On recent versions add `eval "$(fzf --bash)"` to `~/.bashrc`; on Debian you can instead `source /usr/share/doc/fzf/examples/key-bindings.bash`. Until then, `fzf` still works as a pipe target — you just don't get the hotkeys.

4. Never Memorize a Flag Again

You learned the *right* way to get help in Part 1: `man` and `--help`. These three are the fast way — reach for them once you understand what a command *does* and just need the incantation.

tldr — example-first help

Where `man tar` is a wall of text, `tldr tar` is the five commands you actually use. Community-maintained, example-driven.

```
tldr tar          # the common tar invocations, with examples
tldr --update     # refresh the local page cache (tealdeer)
```

cheat.sh — cheat sheets over curl

Nothing to install — if you have `curl`, you have it. Great on a bare server.

```
curl cht.sh/tar           # a concise tar cheat sheet
curl cht.sh/find/delete   # answers to a specific question
```

navi — interactive cheat sheets

A browsable, runnable cheat-sheet launcher: pick a command from a fuzzy list and navi fills in the blanks before running it.

```
navi              # browse and run from interactive cheat sheets
```

5. Practice (Pick Two, Try Them)

No capstone here — just install a couple and feel the difference on a throwaway folder.

```
mkdir cli3-reps && cd cli3-reps
printf 'TODO: ship it\nall good\nTODO: write tests\n' > notes.txt
rg -n TODO notes.txt          # find both TODO lines, with line numbers
bat notes.txt                 # view it highlighted (or: batcat notes.txt)
eza -la                       # a nicer listing of this folder
tldr rg                       # see ripgrep's own common examples
cd .. && rm -rf cli3-reps     # clean up
```

> **That's the whole idea.** These tools don't change *what* you do — they make the doing quicker and the looking-up painless. Add them one at a time as you feel the need; there's no prize for installing all nine at once. The fundamentals from Parts 1 and 2 are the real skill — this is just a sharper set of knives.

That's the track

You've finished the Command Line track. New here? Begin at **Part 1 — The Command Line, From Zero**. More guides on self-hosting, privacy, local AI, and digital sovereignty:

cypherpunkschool.com
