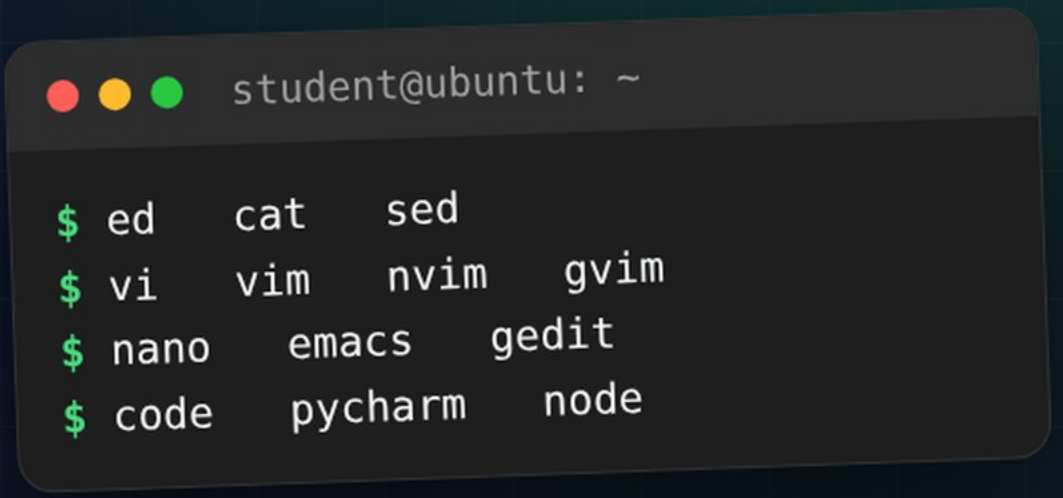
FULL COURSE

LINUX EDITORS & UTILITIES

LINUX EDITORS

Full Course · 6 Lessons



FULL COURSE

Linux Editors & Utilities

1. ed & cat
2. sed
3. vi & vim
4. Neovim (nvim) & gVim
5. nano & Emacs
6. gedit, VS Code, PyCharm & Node.js on Linux

LESSON 1

ed & cat: The Original Linux Text Tools

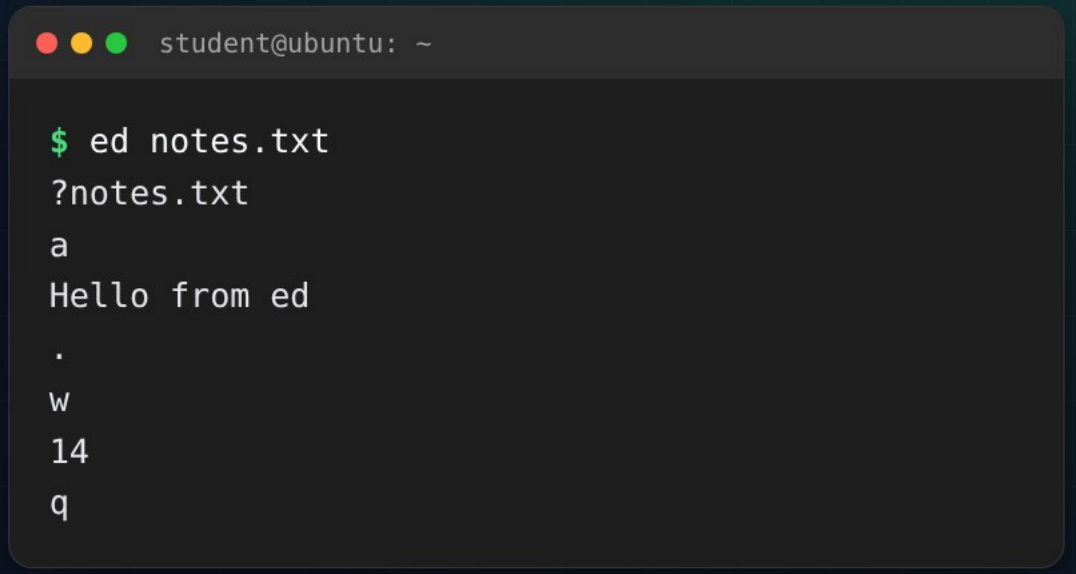
ed · cat

LINUX EDITORS & UTILITIES | Linux | FREE

PART 1 OF 5

ed: The First Unix Editor

1. Written by Ken Thompson in 1969
2. A line editor, no full screen
3. Still useful in scripts & rescue mode



PART 2 OF 5

ed Commands You Need

1. a … then a single dot: append text
2. ,p print all · s/old/new/ substitute
3. w write · q quit

```
student@ubuntu: ~

$ ed -p '> ' notes.txt
14
> ,p
Hello from ed
> 1s/Hello/Hi/
> w
11
> q
```

PART 3 OF 5

cat: View and Join Files

1. cat file: print a file
2. cat -n: number the lines
3. cat a b > all: join files

```
student@ubuntu: ~
$ cat -n notes.txt
     1  Hi from ed
$ cat part1.txt part2.txt > all.txt
$ wc -l all.txt
42 all.txt
```

PART 4 OF 5

Create Files with cat

1. cat > file … then Ctrl+D to save
2. >> appends instead of overwriting
3. Here-doc: cat << EOF in scripts

```
student@ubuntu: ~

$ cat > todo.txt
learn ed
learn cat
^D
$ cat >> todo.txt
learn vim
^D
$ cat << EOF > app.conf
port=8080
EOF
```

PART 5 OF 5

Helpers: less, head, tail

1. less: scroll & search, q to quit
2. head / tail: first or last lines
3. tail -f: watch a log live

```
student@ubuntu: ~

$ less /var/log/syslog
$ head -n 1 /etc/passwd
root:x:0:0:root:/root:/bin/bash
$ tail -n 2 todo.txt
learn cat
learn vim
$ tail -f /var/log/syslog
```

Keep Learning, Free

Slides (PDF): MyWebUniversity.com/presentations

YouTube: @MyWebUniversity

Dari: @MyWebUniversity-Dari

LESSON 2

sed: The Linux Stream Editor for Find & Replace

sed

LINUX EDITORS & UTILITIES | Linux | FREE

PART 1 OF 5

What Is sed?

1. Reads text line by line
2. Applies your commands to each line
3. Prints the result, no window

```
student@ubuntu: ~

$ cat fruits.txt
red apple
green apple
$ sed 's/apple/pear/' fruits.txt
red pear
green pear
```

PART 2 OF 5

Find & Replace: s/old/new/

1. s/old/new/: first match per line
2. g flag: every match
3. I flag: ignore case (GNU sed)

```
student@ubuntu: ~

$ echo 'cat cat cat' | sed 's/cat/dog/'
dog cat cat
$ echo 'cat cat cat' | sed 's/cat/dog/g'
dog dog dog
$ echo 'Linux' | sed 's/linux/Ubuntu/I'
Ubuntu
```

PART 3 OF 5

Edit Files in Place

1. -i: save changes into the file
2. -i.bak: keep a backup
3. Test without -i, then apply

```
student@ubuntu: ~
$ sed 's/8080/9090/' app.conf     # test
port=9090
$ sed -i.bak 's/8080/9090/' app.conf
$ ls app.conf*
app.conf  app.conf.bak
```

PART 4 OF 5

Delete and Print Lines

1. sed '3d': delete line 3
2. sed '/^#/d': drop comment lines
3. sed -n '5,10p': print lines 5–10

```
student@ubuntu: ~

$ sed '3d' list.txt
$ sed '/^#/d' /etc/ssh/sshd_config
Include /etc/ssh/sshd_config.d/*.conf
KbdInteractiveAuthentication no
UsePAM yes
$ sed -n '5,10p' big.log
```

PART 5 OF 5

sed in Real DevOps Work

1. Update config files in scripts
2. Clean data before analysis
3. Combine with grep, awk & pipes

student@ubuntu: ~

```
# set a value in a config file
$ sed -i 's/^debug=.*/debug=false/' app.conf
# CSV to tab-separated
$ sed 's/,/\t/g' sales.csv > sales.tsv
# keep only the error text
$ grep ERROR app.log | sed 's/.*ERROR: //'
```

Keep Learning, Free

Slides (PDF): MyWebUniversity.com/presentations

YouTube: @MyWebUniversity

Dari: @MyWebUniversity-Dari

LESSON 3

vi & vim: The Editor on Every Linux Server

vi · vim

LINUX EDITORS & UTILITIES | Linux | FREE

PART 1 OF 5

vi and vim

1. vi: on every Unix & Linux system
2. vim: colors, undo, splits, plugins
3. Open a file: vim app.py

```
student@ubuntu: ~

$ which vi vim
/usr/bin/vi
/usr/bin/vim
$ vim app.py
```

PART 2 OF 5

Modes: The Big Idea

1. Normal mode: move & give commands
2. Insert mode: press i, then type
3. Esc always takes you back

```
vim

-- NORMAL --   move & give commands
  i    ->  -- INSERT --   type text
  Esc ->  back to NORMAL
  :    ->  command line  (:w  :q)
```

PART 3 OF 5

Save and Quit

1. :w save · :q quit
2. :wq or ZZ: save and quit
3. :q! quit without saving

vim

```
:w      save
:q      quit
:wq     save and quit   (or ZZ)
:q!     quit, throw away changes
```

PART 4 OF 5

Move and Edit Fast

1. h j k l · w b · gg G
2. dd delete · yy copy · p paste
3. u undo · Ctrl+R redo

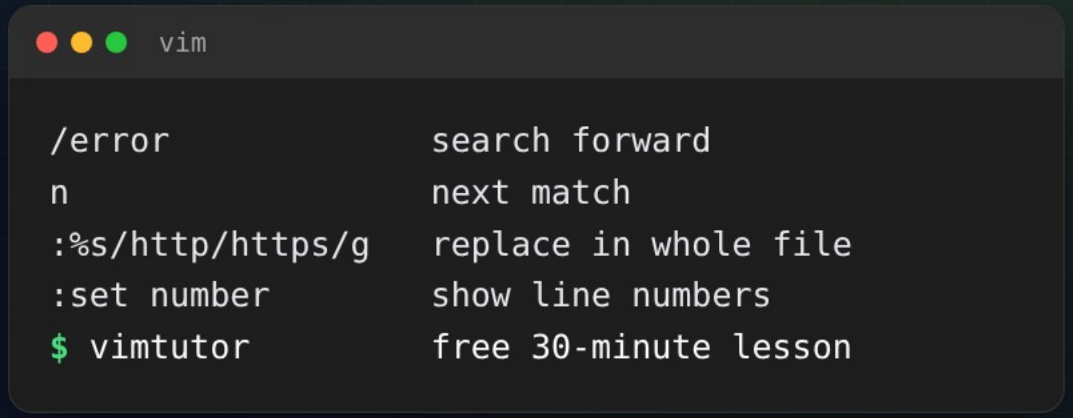
vim

```
h j k l   left down up right
w  b      next / previous word
gg  G     top / bottom of file
dd  yy  p delete, copy, paste a line
u  Ctrl+R undo / redo
```

PART 5 OF 5

Search and Replace

1. /word to search, n for next
2. :%s/old/new/g in the whole file
3. vimtutor: free built-in lesson



Keep Learning, Free

Slides (PDF): MyWebUniversity.com/presentations

YouTube: @MyWebUniversity

Dari: @MyWebUniversity-Dari

LESSON 4

Neovim (nvim) & gVim: Modern and Graphical Vim

`nvim · gvim`

LINUX EDITORS & UTILITIES | Linux | FREE

PART 1 OF 5

What Is Neovim?

1. Same keys and modes as vim
2. Command: nvim
3. Config in Lua: init.lua

```
student@ubuntu: ~

$ sudo apt install neovim
$ nvim app.py
$ nvim ~/.config/nvim/init.lua
```

PART 2 OF 5

Why Developers Choose Neovim

1. Built-in LSP: smart code help
2. Tree-sitter: smarter syntax colors
3. Huge plugin ecosystem

vim

```
:checkhealth          check your setup
:lua print('hi')     run Lua inside nvim
:Lazy                 plugin manager UI
                      (if you install lazy.nvim)
```

PART 3 OF 5

A First init.lua

1. vim.opt.number = true
2. tabstop & shiftwidth = 4
3. Save, then restart nvim

init.lua

```
-- ~/.config/nvim/init.lua
vim.opt.number = true
vim.opt.relativenumber = true
vim.opt.tabstop = 4
vim.opt.shiftwidth = 4
vim.opt.expandtab = true
```

PART 4 OF 5

gVim: Vim in a Window

1. Menus, mouse and toolbar
2. Ubuntu: sudo apt install vim-gtk3
3. Same commands, same .vimrc

```
student@ubuntu: ~

$ sudo apt install vim-gtk3
$ gvim notes.txt
$ vim --version | grep -o '+clipboard'
+clipboard
```

PART 5 OF 5

Which One Should You Use?

1. Servers: vi or vim, always there
2. Coding all day: try Neovim
3. Like menus & a mouse: gVim

```
student@ubuntu: ~

server$   vim /etc/hosts
laptop$   nvim main.go
desktop$  gvim report.txt
```

Keep Learning, Free

Slides (PDF): MyWebUniversity.com/presentations

YouTube: @MyWebUniversity

Dari: @MyWebUniversity-Dari

LESSON 5

nano & Emacs: Easy and Powerful Linux Terminal Editors

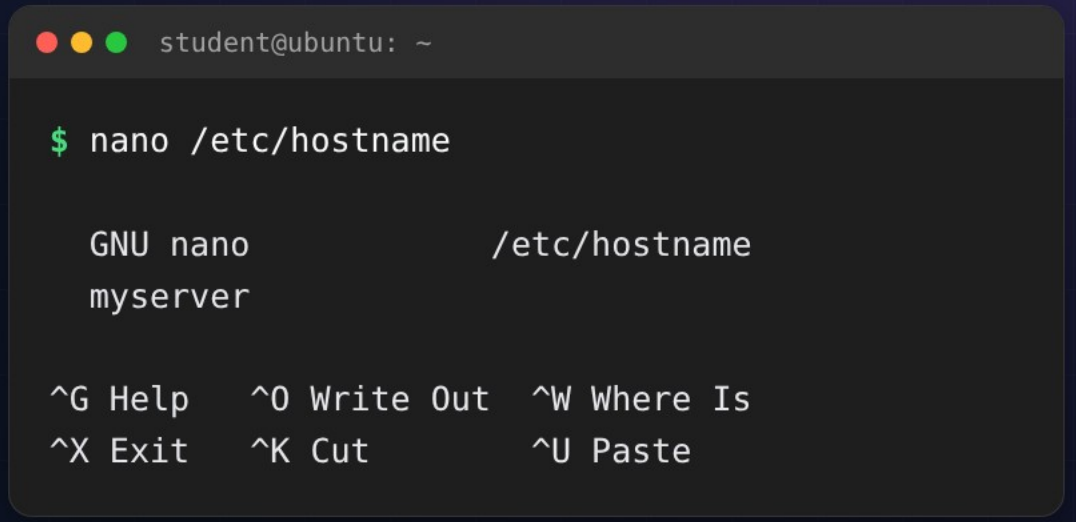
`nano · emacs`

LINUX EDITORS & UTILITIES | Linux | FREE

PART 1 OF 5

nano: The Easiest Start

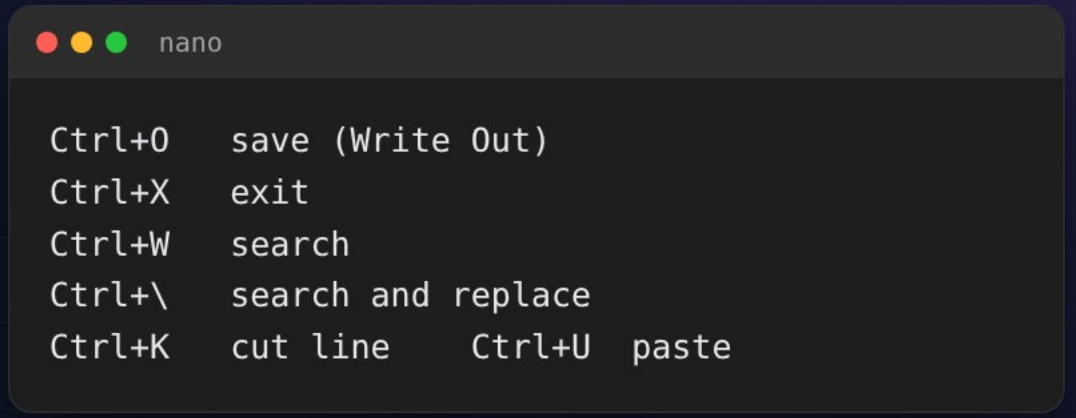
1. No modes: just type
2. Shortcuts listed at the bottom
3. Installed on Ubuntu by default



PART 2 OF 5

nano Keys You Need

1. Ctrl+O save · Ctrl+X exit
2. Ctrl+W search · Ctrl+\ replace
3. Ctrl+K cut line · Ctrl+U paste



PART 3 OF 5

Emacs: The Extensible Editor

1. Install: sudo apt install emacs
2. emacs -nw: inside the terminal
3. Modeless, built on Emacs Lisp

```
student@ubuntu: ~

$ sudo apt install emacs
$ emacs -nw notes.org      # in terminal
$ emacs report.txt &       # in a window
```

PART 4 OF 5

Emacs Key Basics

1. C-x C-f open · C-x C-s save
2. C-x C-c quit · C-g cancel
3. C-s search · M-% replace

emacs

```
C-x C-f   open a file
C-x C-s   save
C-x C-c   quit
C-g       cancel anything
C-s       search     M-%  replace
```

PART 5 OF 5

Why Emacs Is Special

1. Org mode: notes, tasks, agendas
2. Magit: a great Git interface
3. C-h t: built-in tutorial

emacs

```
* TODO Learn nano
* DONE Learn vim
* TODO Finish the Emacs tutorial
  SCHEDULED: <2026-11-02 Mon>

M-x magit-status
C-h t     built-in tutorial
```

Keep Learning, Free

Slides (PDF): MyWebUniversity.com/presentations

YouTube: @MyWebUniversity

Dari: @MyWebUniversity-Dari

LESSON 6

gedit, VS Code, PyCharm & Node.js on Linux

`code · node`

LINUX EDITORS & UTILITIES | Linux | FREE

PART 1 OF 5

gedit & GNOME Text Editor

1. Ubuntu default: GNOME Text Editor
2. Classic gedit: sudo apt install gedit
3. From the terminal: gedit file &

```
student@ubuntu: ~

$ gnome-text-editor notes.txt &
$ sudo apt install gedit
$ gedit ~/.bashrc &
```

PART 2 OF 5

VS Code on Linux

1. sudo snap install code --classic
2. code . opens the current folder
3. Extensions for every language

```
student@ubuntu: ~
$ sudo snap install code --classic
$ cd ~/projects/myapp
$ code .
$ code --install-extension ms-python.python
```

PART 3 OF 5

PyCharm for Python

1. One PyCharm: core features free
2. sudo snap install pycharm --classic
3. Debugger, virtualenvs, Jupyter

```
student@ubuntu: ~

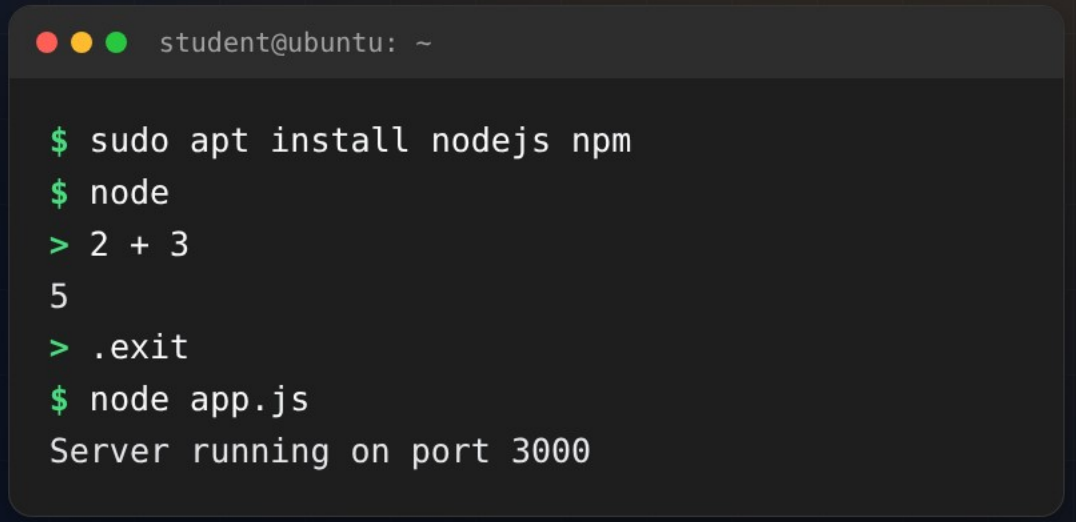
$ sudo snap install pycharm --classic
$ pycharm ~/projects/analysis &

  New Project > Interpreter:
  Project venv (Python 3)
```

PART 4 OF 5

Node.js: JavaScript Anywhere

1. sudo apt install nodejs npm
2. node: an interactive prompt
3. node app.js: run a script



PART 5 OF 5

Pick the Right Tool

1. Quick notes: gedit / Text Editor
2. Any language & DevOps: VS Code
3. Big Python & data projects: PyCharm

```
student@ubuntu: ~

notes.txt        ->  gedit / Text Editor
Dockerfile       ->  code
analysis.ipynb   ->  pycharm
server.js        ->  code + node
```

Keep Learning, Free

Slides (PDF): MyWebUniversity.com/presentations

YouTube: @MyWebUniversity

Dari: @MyWebUniversity-Dari