

CHAPTER 2

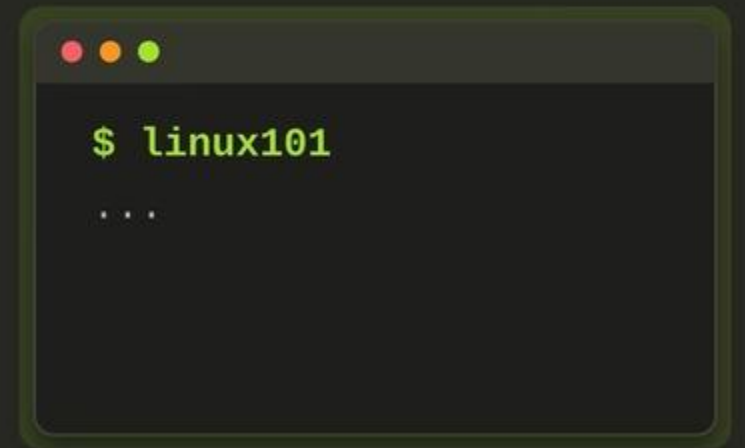
`$ linux101`

LINUX 101

30 essential commands, real examples, zero fluff

30 Commands • Full Course

Linux Basics — Full Course



\$ linux101

Linux Basics — Full Course

30 essential commands, real examples, zero fluff

GO

Learn By Doing

Every command shown live in a real terminal.

RE

Beginner Friendly

No prior experience needed to follow along.

IO

30 Real Commands

The exact basics every Linux user needs.

OK

One Video, Start To Finish

Bookmark it and work through at your pace.

Commands covered: banner · cal · cat · clear · date · echo
head · hostname · host · id · info · paste · ps · pwd · uname
less · ls · man · mkdir · more · pg · ping · dig · nslookup
atrm · atq · sort · touch · w · who

user@linux: ~

```
$ echo \  
"Let's learn Linux!"
```

**Let's learn
Linux!**

```
wahid@192.168.254.29:22 - Bitwise xterm - wahid@algorithm: ~  
wahid@algorithm:~$ echo \  
> "Let's Learn Linux!"  
Let's Learn Linux!  
wahid@algorithm:~$ echo "Let's Learn Linux!"  
Let's Learn Linux!
```

~35-40 min

\$ linux101

Linux Basics — Full Course

30 essential commands, real examples, zero fluff

GO

Learn By Doing

Every command shown live in a real terminal.

RE

Beginner Friendly

No prior experience needed to follow along.

IO

30 Real Commands

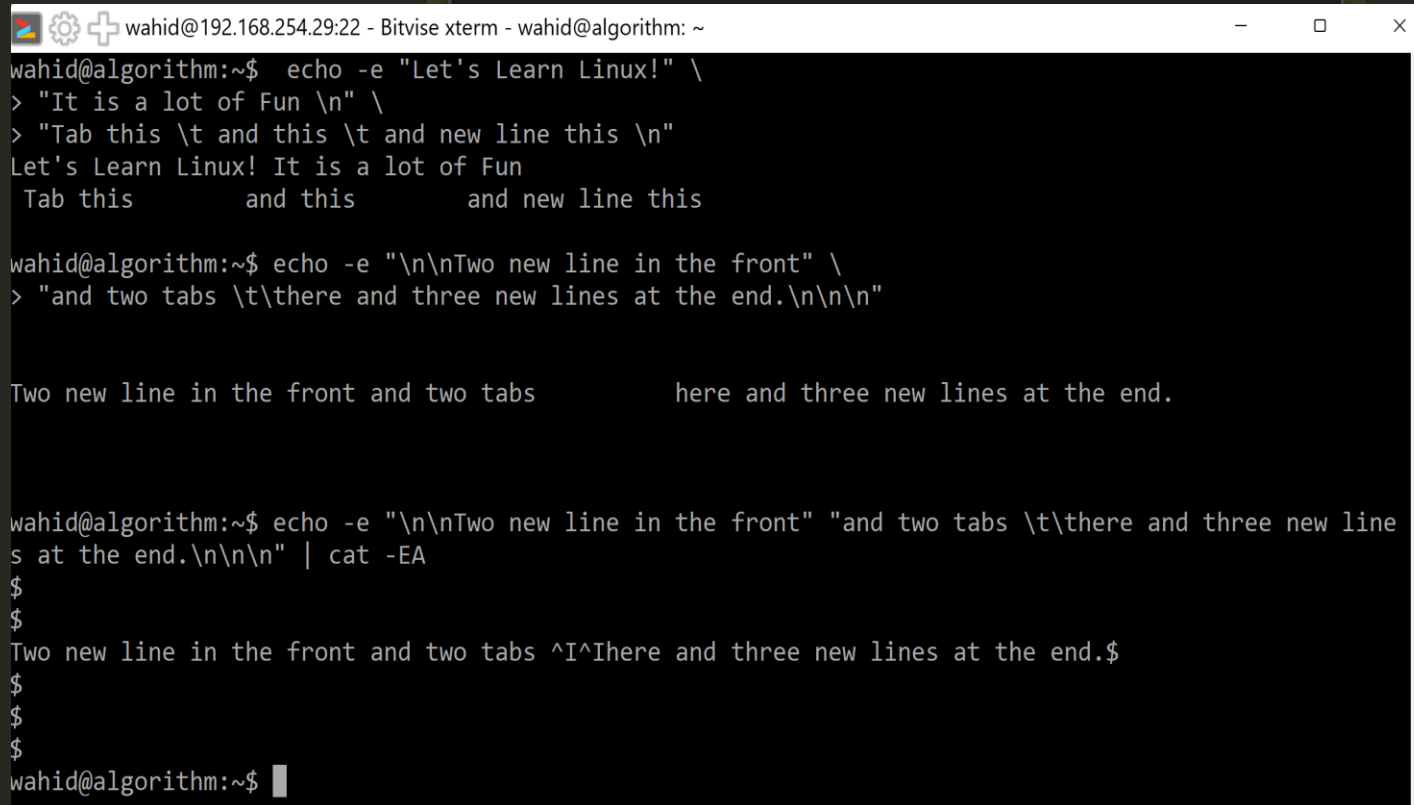
The exact basics every Linux user needs.

OK

One Video, Start To Finish

Bookmark it and work through at your pace.

Commands covered: banner · cal · cat · clear · date · echo
head · hostname · host · id · info · paste · ps · pwd · uname
less · ls · man · mkdir · more · pg · ping · dig · nslookup
atrm · atq · sort · touch · w · who



```
wahid@192.168.254.29:22 - Bitvise xterm - wahid@algorithm: ~
wahid@algorithm:~$ echo -e "Let's Learn Linux!" \
> "It is a lot of Fun \n" \
> "Tab this \t and this \t and new line this \n"
Let's Learn Linux! It is a lot of Fun
Tab this      and this      and new line this

wahid@algorithm:~$ echo -e "\n\nTwo new line in the front" \
> "and two tabs \t\there and three new lines at the end.\n\n\n"

Two new line in the front and two tabs      here and three new lines at the end.

wahid@algorithm:~$ echo -e "\n\nTwo new line in the front" "and two tabs \t\there and three new line
s at the end.\n\n\n" | cat -EA
$
$
Two new line in the front and two tabs ^I^Ihere and three new lines at the end.$
$
$
wahid@algorithm:~$
```

Linux Basics — Full Course

```
wahid@192.168.254.29:22 - Bitvise xterm - wahid@algorithm: ~
wahid@algorithm:~$ echo -e "\n\nTwo new line in the front" "and two tabs \t\tthere and three new lines
at the end.\n\n\n" | lolcat -F 0.001

Two new line in the front and two tabs          here and three new lines at the end.

wahid@algorithm:~$ echo -e "\n\nTwo new line in the front" "and two tabs \t\tthere and three new lines
at the end.\n\n\n" | cat -EA | lolcat -F 0.001
$
$
Two new line in the front and two tabs ^I^Ihere and three new lines at the end.$
$
$
$
$
wahid@algorithm:~$ echo -e "\n\nTwo new line in the front" "and two tabs \t\tthere and three new lines
at the end.\n\n\n" | cat -EA | lolcat
$
$
Two new line in the front and two tabs ^I^Ihere and three new lines at the end.$
$
$
$
$
wahid@algorithm:~$
```

```
$ linux101
```

```
user@linux: ~

$ echo \
    "Let's learn Linux!"
Let's learn Linux!
```

```
wahid@192.168.254.29:22 - Bitvise xterm - wahid@algorithm: ~
wahid@algorithm:~$ echo -e "\n\nTwo new line in the front" "and two tabs \t\there and three new lines
at the end.\n\n\n" | lolcat -F 0.001

Two new line in the front and two tabs          here and three new lines at the end.

wahid@algorithm:~$ echo -e "\n\nTwo new line in the front" "and two tabs \t\there and three new lines
at the end.\n\n\n" | cat -EA | lolcat -F 0.001
$
$
Two new line in the front and two tabs ^I^Ihere and three new lines at the end.$
$
$
$
wahid@algorithm:~$ echo -e "\n\nTwo new line in the front" "and two tabs \t\there and three new lines
at the end.\n\n\n" | cat -EA | lolcat
$
$
Two new line in the front and two tabs ^I^Ihere and three new lines at the end.$
$
$
$
wahid@algorithm:~$ █
```

01

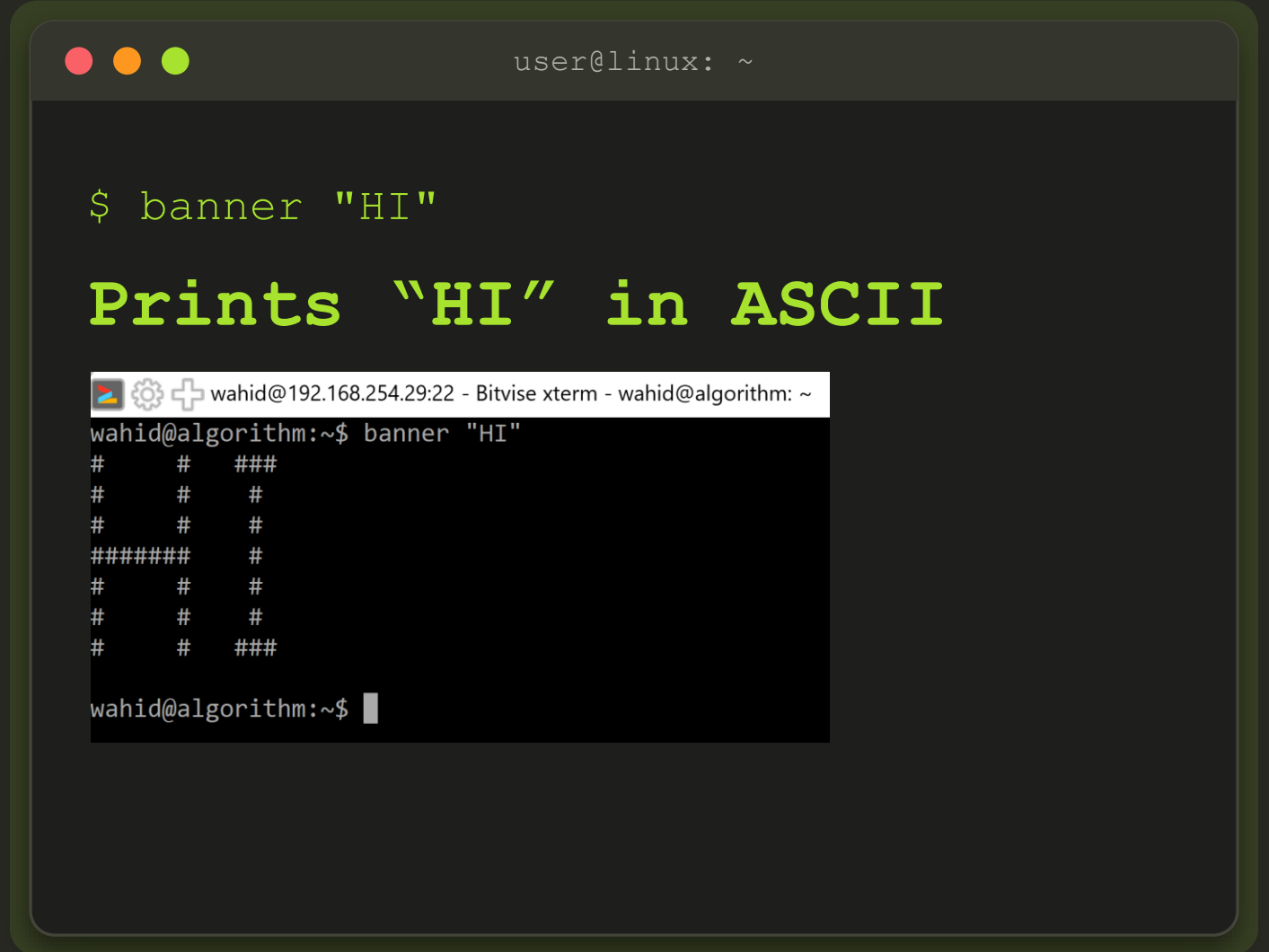
Print A Big ASCII Banner

Turn plain text into giant letters in the terminal

- banner "text" prints huge ASCII letters
- Great for headers in scripts or splash screens
- May need installing first: `sudo apt install sysvbanner`

SYNTAX

```
banner "HI"
```

A terminal window titled 'user@linux: ~' showing the command '\$ banner "HI"' and its output. The output is a large, stylized ASCII banner that reads 'Prints "HI" in ASCII'. Below this, there is an inset image showing a terminal window titled 'wahid@192.168.254.29:22 - Bitvise xterm - wahid@algorithm: ~'. This inset shows the command 'wahid@algorithm:~\$ banner "HI"' and its output, which is a smaller version of the same ASCII banner.

```
user@linux: ~  
  
$ banner "HI"  
  
Prints "HI" in ASCII  
  
wahid@192.168.254.29:22 - Bitvise xterm - wahid@algorithm: ~  
wahid@algorithm:~$ banner "HI"  
#   #   ###  
#   #   #  
#   #   #  
##### #  
#   #   #  
#   #   #  
#   #   ###  
  
wahid@algorithm:~$
```

02

Show A Calendar In The Terminal

See any month or year without leaving the shell

- `cal` alone shows the current month
- `cal 2026` prints the whole year
- `cal 8 2026` jumps to a specific month

SYNTAX

`cal`

```
user@linux: ~  
  
$ cal  
  
Su Mo Tu We Th Fr Sa  
1  2  3  4  5  6  
  
August 2026
```

03

Display A File's Contents

Print any text file straight to your screen

- `cat` file prints the whole file at once
- `cat a.txt b.txt` joins multiple files together
- The go-to command for quick file peeks

SYNTAX

```
cat notes.txt
```



```
user@linux: ~  
$ cat notes.txt  
  
hello world
```

04

Clear A Cluttered Terminal

Wipe the screen and start with a clean slate

- `clear` erases everything currently visible
- Your command history is still there (↑ arrow)
- Shortcut: `Ctrl+L` does the same thing instantly

SYNTAX

`clear`



```
user@linux: ~  
  
$ clear  
  
Screen cleared
```

05

Show The Current Date & Time

Check exactly what the system clock says

- `date` alone prints the full date and time
- `+FORMAT` customizes exactly how it looks
- Useful for timestamps inside shell scripts

SYNTAX

```
date "+%Y-%m-%d"
```



```
user@linux: ~  
  
$ date "+%Y-%m-%d"  
  
2026-08-05
```

06

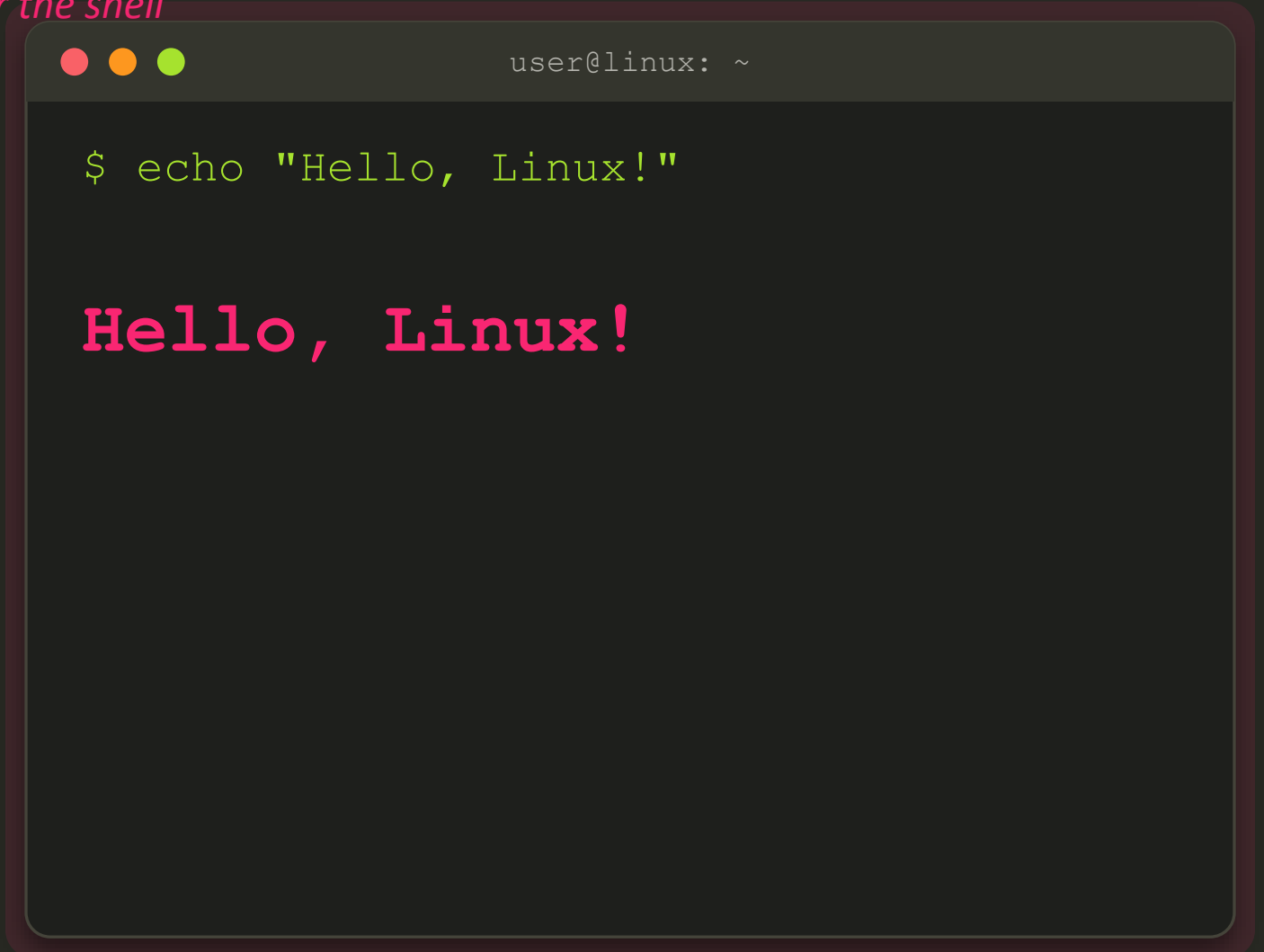
Print Text To The Screen

Output any message from a script or the shell

- `echo "text"` prints it right back to you
- Echoes variables too: `echo $HOME`
- The backbone of countless shell scripts

SYNTAX

```
echo "Hello, Linux!"
```

A terminal window with a dark background and a title bar showing 'user@linux: ~'. The prompt '\$' is followed by the command 'echo "Hello, Linux!"' in green. The output 'Hello, Linux!' is displayed in large, bold, pink letters.

```
user@linux: ~  
$ echo "Hello, Linux!"  
Hello, Linux!
```

07

Preview The Start Of A File

See just the first few lines, not the whole thing

- head file shows the first 10 lines by default
- -n 5 changes how many lines it shows
- Great for peeking at huge log files safely

SYNTAX

```
head -n 3 log.txt
```



user@linux: ~

```
$ head -n 3 log.txt
```

```
line 1
```

```
line 2
```

First 3 lines

08

Find Your Machine's Hostname

See the name your computer is known by on the network

- `hostname` alone prints the current machine name
- `-I` shows the machine's IP addresses instead
- Handy when you're SSH'd into several servers

SYNTAX

`hostname`



```
user@linux: ~  
  
$ hostname  
  
web-server-01
```

09

Look Up A Domain's IP Address

Turn a website name into its real IP address

- `host domain.com` resolves it through DNS
- Shows the A record (IPv4 address) by default
- A quick, simple alternative to `dig`

SYNTAX

```
host example.com
```

A terminal window with a dark background and light green text. The window title bar shows 'user@linux: ~' and three colored window control buttons (red, yellow, green). The terminal content shows the command '\$ host example.com' followed by the output 'example.com has address' and the IP address '93.184.216.34' in a larger, bold font.

```
user@linux: ~  
  
$ host example.com  
example.com has address  
  
93.184.216.34
```

10

Check Your User & Group ID

See exactly who you are to the system

- `id` alone shows your UID, GID, and groups
- `id username` checks another account
- Useful for confirming permissions issues

SYNTAX

`id`

```
user@linux: ~  
  
$ id  
uid=1000(ana) gid=1000  
  
You: ana
```

`$ progress --check`

10 Down, 20 To Go!

Liking this so far? Hit subscribe — it really helps the channel.



SUBSCRIBE

English

@MyWebUniversity



SUBSCRIBE

Dari / دری

@MyWebUniversity-Dari

on YouTube — for more Linux & command-line tutorials



11

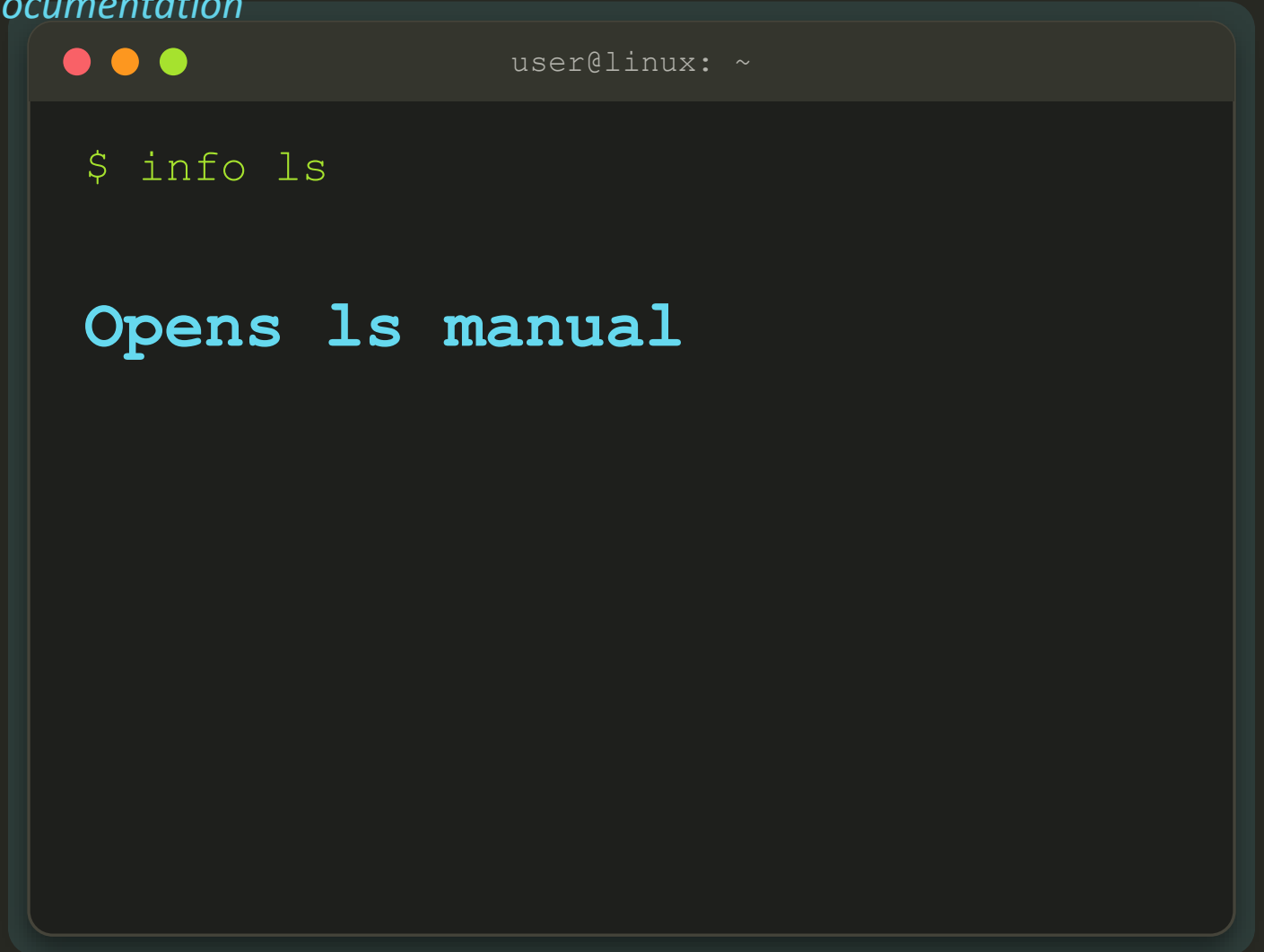
Read A Command's Full Manual

Browse GNU's deeper, hyperlinked documentation

- info command opens its full GNU manual
- Navigate with arrow keys, press q to quit
- Often more detailed than a man page

SYNTAX

`info ls`



```
user@linux: ~  
  
$ info ls  
  
Opens ls manual
```

12

Merge Files Side By Side

Combine matching lines from two files into columns

- `paste a.txt b.txt` joins lines side by side
- Uses a tab character between each column
- Handy for building quick two-column reports

SYNTAX

```
paste names.txt ages.txt
```



user@linux: ~

```
$ paste names.txt \  
ages.txt
```

```
Ana    29
```

Merged columns

13

See What's Running Right Now

List every active process on the system

- `ps` alone shows processes in your terminal
- `ps aux` shows every process for every user
- The first step for finding a stuck program

SYNTAX

`ps aux`

```
user@linux: ~  
  
$ ps aux  
USER    PID    COMMAND  
ana     812    bash  
  
1 process
```

14

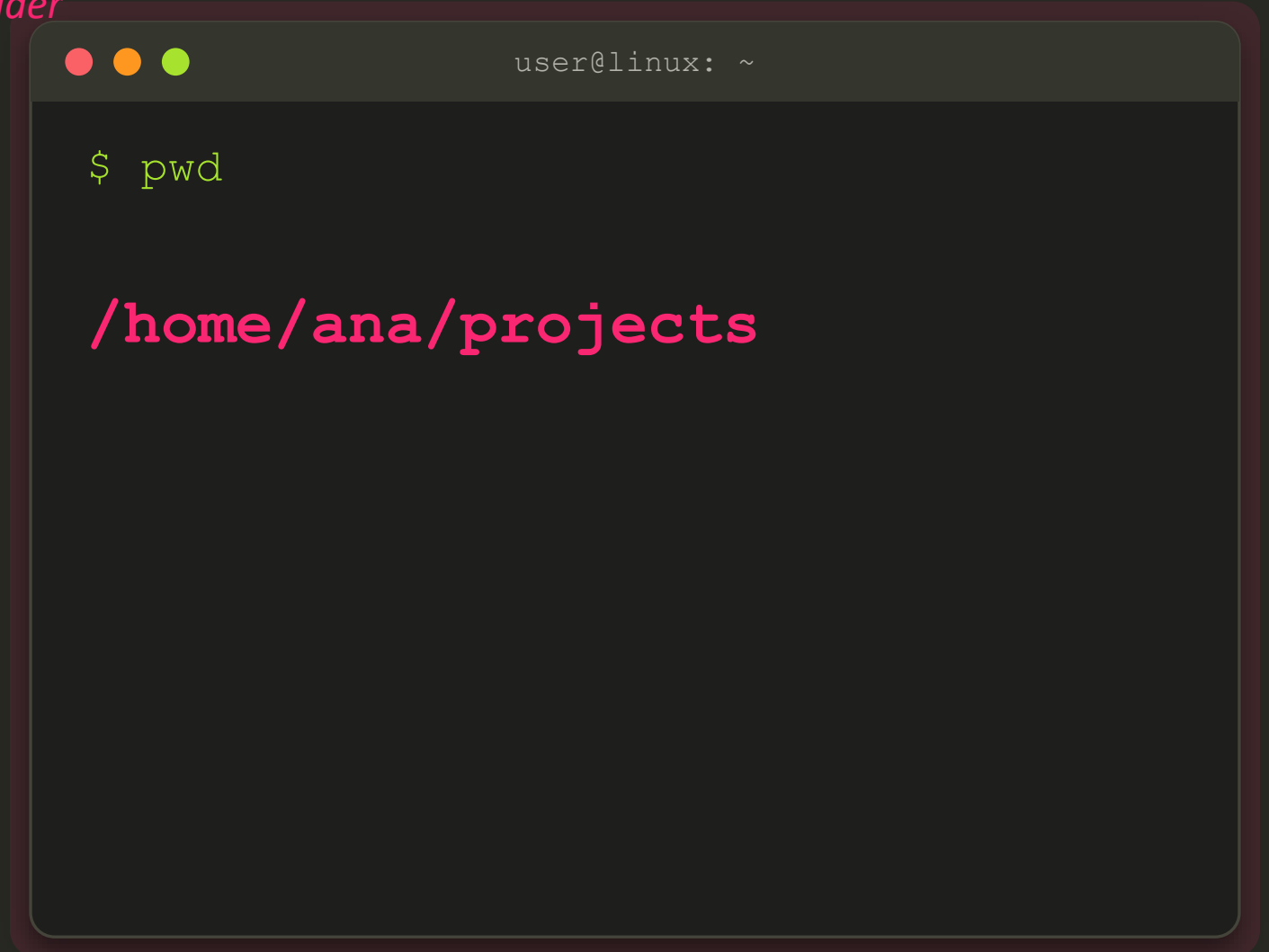
Find Out Where You Are

Print the full path of your current folder

- `pwd` prints your exact current location
- Essential once you've `cd`'d several folders deep
- Combine with `cd $(pwd)` inside scripts

SYNTAX

`pwd`



```
user@linux: ~  
  
$ pwd  
  
/home/ana/projects
```

15

Check Your Linux System Info

See the kernel, OS, and architecture you're on

- `uname` alone shows just the OS name
- `-a` prints everything: kernel, host, arch
- Handy when reporting bugs or checking compatibility

SYNTAX

`uname -a`



user@linux: ~

```
$ uname -a
```

```
Linux web-server-01
```

Linux 6.8.0



16

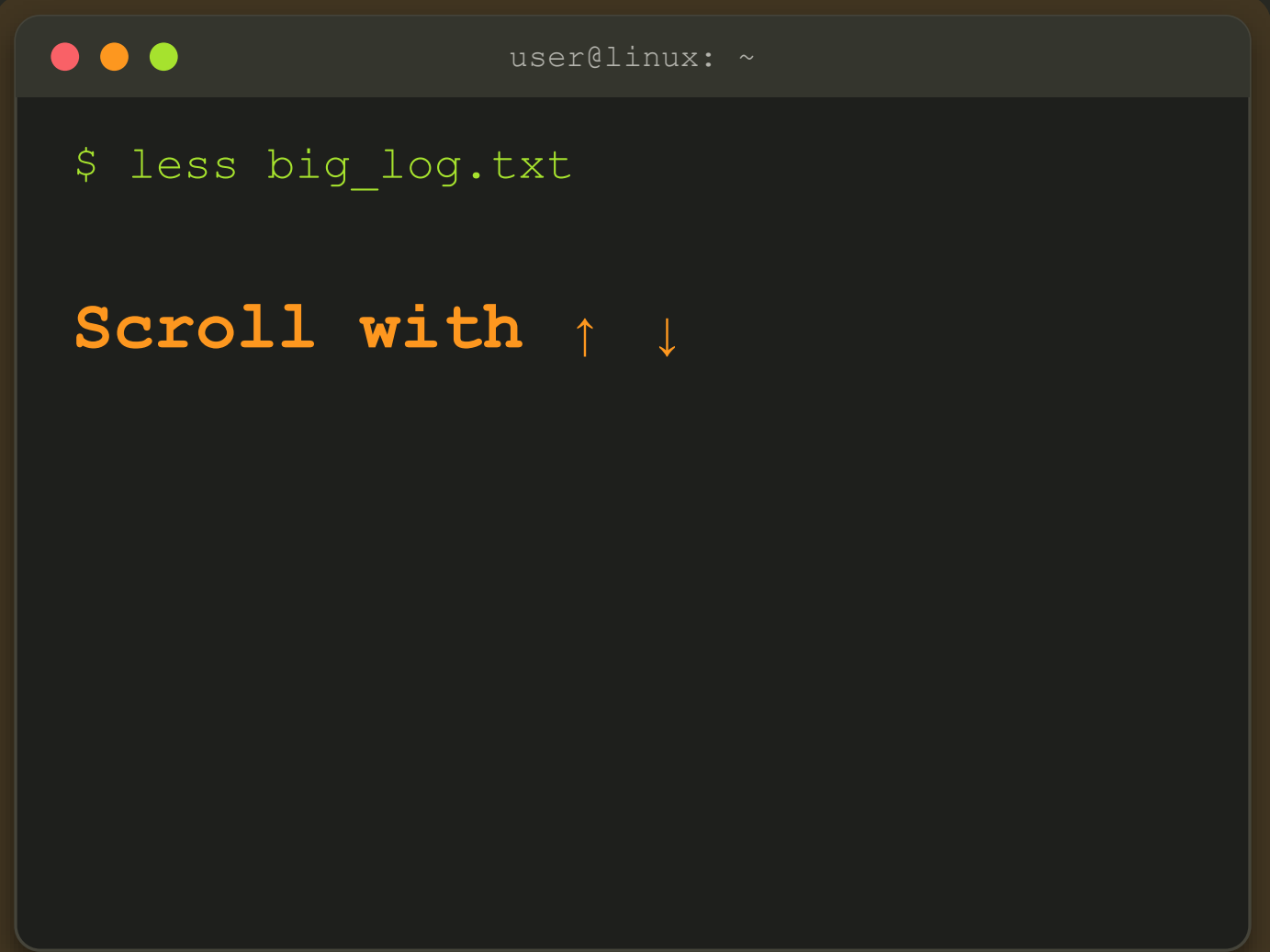
Scroll Through A Big File

View long files one screen at a time

- less file lets you scroll up and down freely
- / searches forward, n jumps to next match
- Press q anytime to quit back to the shell

SYNTAX

```
less big_log.txt
```



A terminal window with a dark background and a title bar showing 'user@linux: ~'. The terminal contains the command `$ less big_log.txt`. Below the command, the text **Scroll with** is displayed in orange, followed by an upward arrow `↑` and a downward arrow `↓`, indicating the keys used for scrolling.

17

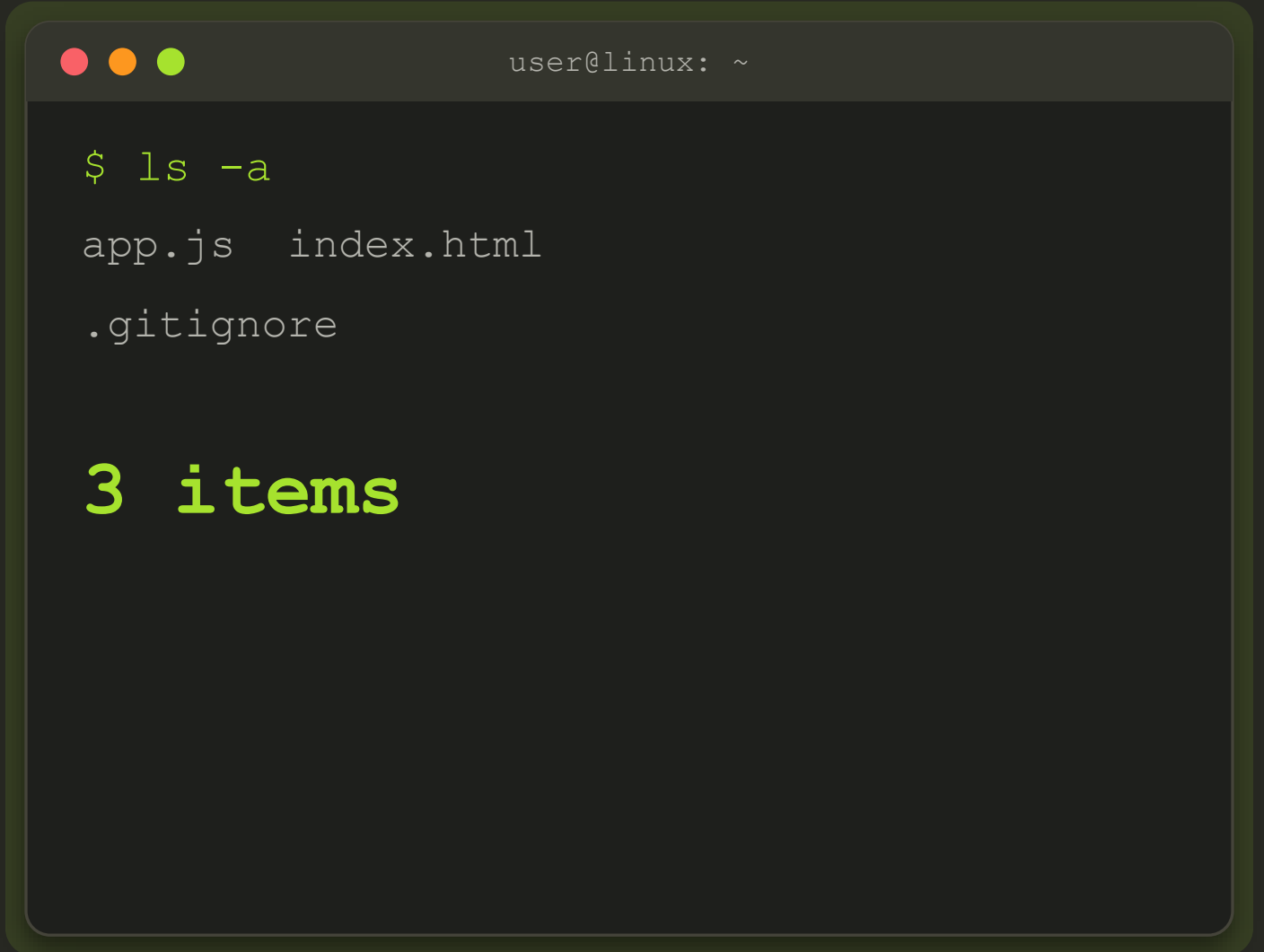
List What's Inside A Folder

See every file and folder at a glance

- ls alone lists the current folder's contents
- -l adds permissions, size, and date
- -a reveals hidden dotfiles too

SYNTAX

```
ls -la
```

A terminal window with a dark background and a title bar showing 'user@linux: ~'. The command '\$ ls -a' has been executed, resulting in the output: 'app.js index.html .gitignore'. Below the output, the text '3 items' is displayed in a large, bold, light blue font.

```
user@linux: ~  
  
$ ls -a  
app.js  index.html  
.gitignore  
  
3 items
```

18

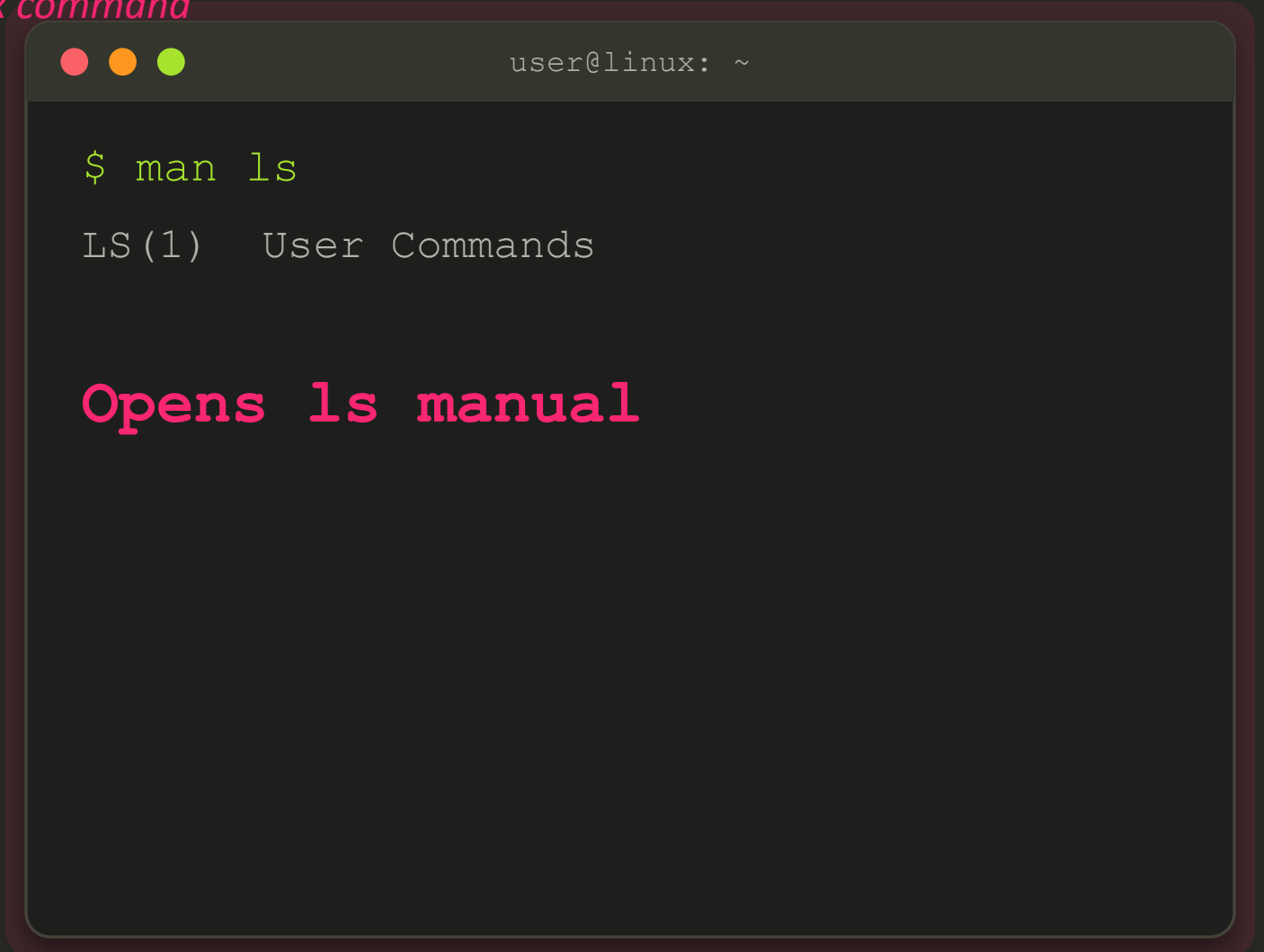
Read Any Command's Manual

The built-in reference for every Linux command

- `man` command opens its official manual
- Press `q` to quit, `/` to search inside it
- Works completely offline, no internet needed

SYNTAX

`man ls`



```
user@linux: ~  
  
$ man ls  
LS(1)  User Commands  
  
Opens ls manual
```

19

Create A New Folder

Build a directory in a single command

- `mkdir` name creates the folder instantly
- `-p` creates nested parent folders too
- The starting point for any new project

SYNTAX

```
mkdir project
```



```
user@linux: ~  
  
$ mkdir project  
  
Created!
```

20

Page Through Output, Old-School

The simpler ancestor of the less command

- more file shows one screen at a time
- Space bar advances to the next page
- less does everything more does, and more

SYNTAX

```
more log.txt
```



A terminal window with a dark background and a title bar showing 'user@linux: ~'. The terminal displays the command '\$ more log.txt' in green. Below it, the output '--More-- (20%)' is shown in white. A large orange text overlay '20% shown' is positioned in the center of the terminal window.

```
user@linux: ~  
  
$ more log.txt  
--More-- (20%)  
  
20% shown
```

`$ progress --check`

20 Down, 10 To Go!

Almost there! A quick subscribe keeps more tutorials coming.

▶ **SUBSCRIBE**
English

▶ **SUBSCRIBE**
Dari / دری

@MyWebUniversity

@MyWebUniversity-Dari

on YouTube — for more Linux & command-line tutorials

21

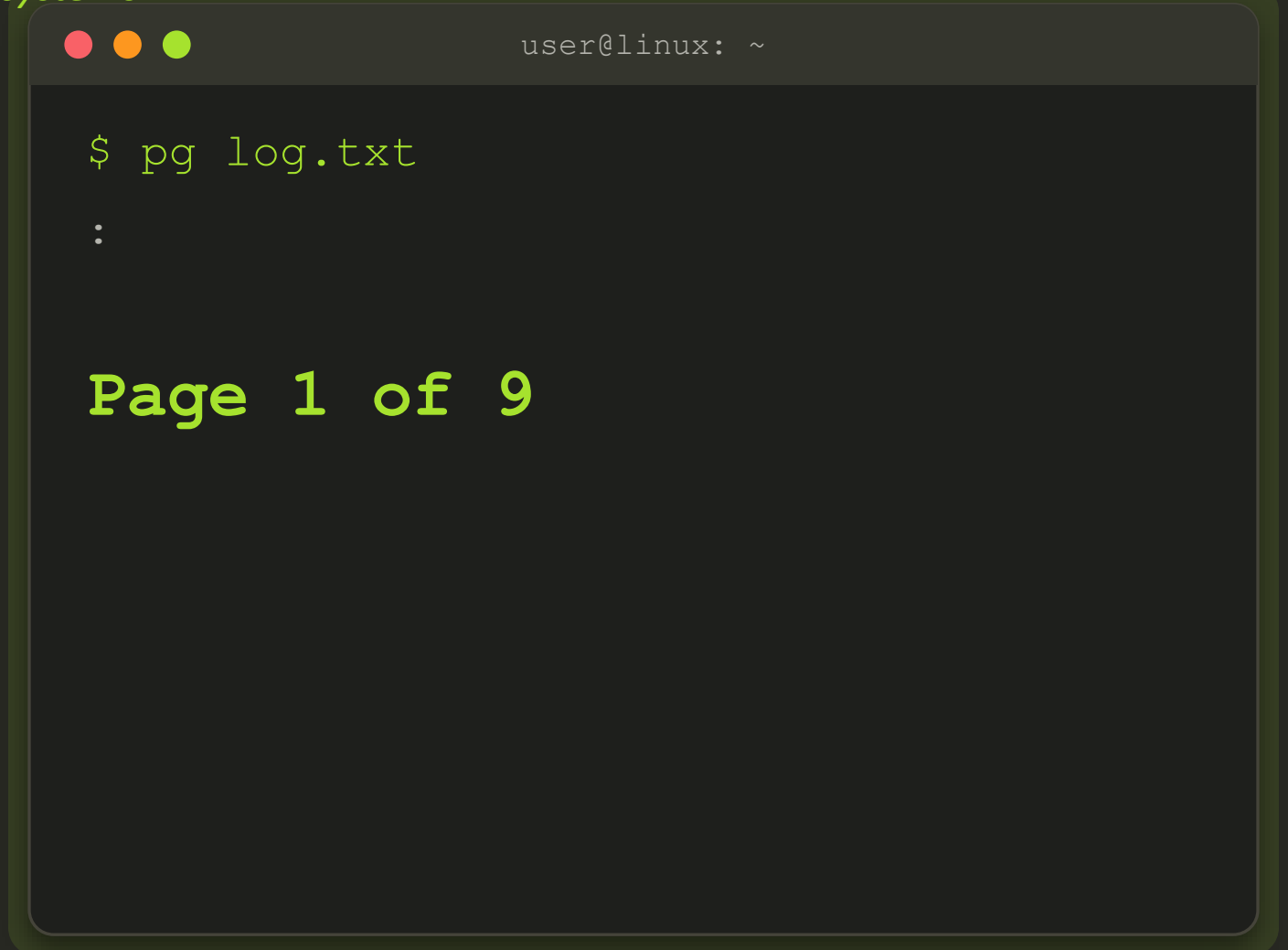
Page Through Files, Another Way

A classic pager found on many Unix systems

- `pg` file shows one page at a time
- Press Enter to move to the next page
- Largely replaced today by `less`

SYNTAX

```
pg log.txt
```



```
user@linux: ~  
  
$ pg log.txt  
:  
  
Page 1 of 9
```

22

Test If A Server Is Reachable

Send a quick network signal and wait for a reply

- ping host sends repeated test packets
- Reply time shows your connection speed
- Ctrl+C stops it whenever you're done

SYNTAX

```
ping example.com
```



```
user@linux: ~  
  
$ ping example.com  
64 bytes: time=12ms  
  
12ms reply
```

23

Dig Deep Into DNS Records

Get the full technical DNS lookup details

- `dig domain.com` queries DNS servers directly
- Shows every record type: A, MX, NS, and more
- The go-to tool for serious DNS debugging

SYNTAX

```
dig example.com
```



user@linux: ~

```
$ dig example.com
```

```
ANSWER SECTION:
```

```
93.184.216.34
```



24

Look Up DNS The Classic Way

A simple, interactive tool for DNS queries

- nslookup domain.com resolves it instantly
- Also works in interactive mode, no arguments
- An older tool, but still works everywhere

SYNTAX

```
nslookup example.com
```

A terminal window with a dark background and a title bar showing 'user@linux: ~'. The terminal displays the command '\$ nslookup example.com' in green, followed by the output 'Address: 93.184.216.34' in white. Below the output, the word 'Resolved!' is written in large, bold, orange letters.

```
user@linux: ~  
  
$ nslookup example.com  
Address: 93.184.216.34  
  
Resolved!
```

25

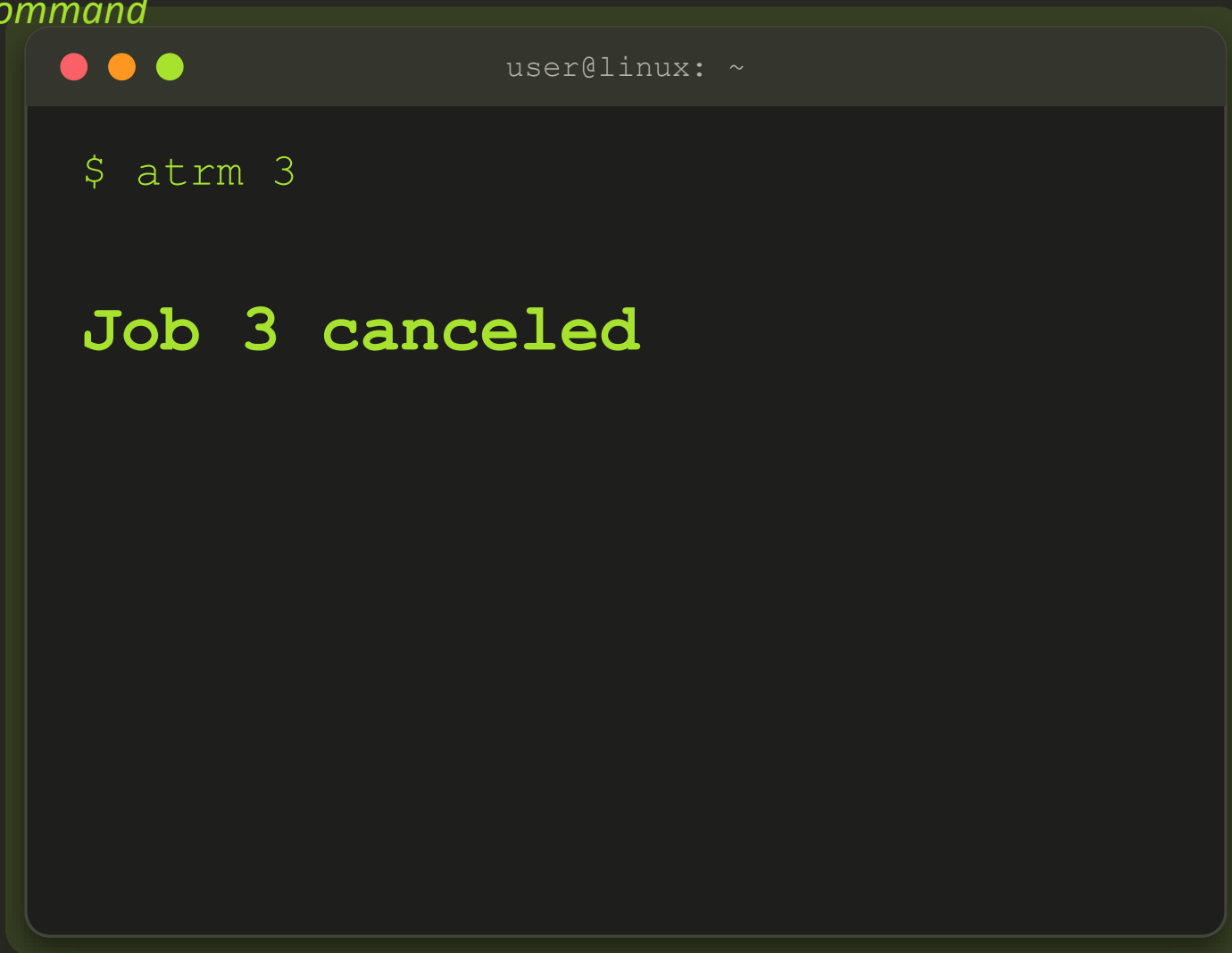
Cancel A Scheduled Job

Remove a task queued with the at command

- atrm JOBID cancels that scheduled job
- Find the job number first with atq
- Useful for undoing a one-time scheduled task

SYNTAX

```
atrm 3
```



```
user@linux: ~  
  
$ atrm 3  
  
Job 3 canceled
```

26

See Your Scheduled Jobs

List every one-time task waiting to run

- atq shows every job queued with at
- Displays the job number, date, and time
- Check this before canceling one with atrm

SYNTAX

atq

```
user@linux: ~  
  
$ atq  
3  Wed Aug 5 14:00  
  
1 job queued
```

27

Sort Lines Of Text Instantly

Put any list into alphabetical or numeric order

- sort file arranges lines alphabetically
- -n sorts numbers correctly, not as text
- -r reverses the order, largest/last first

SYNTAX

```
sort names.txt
```



user@linux: ~

```
$ sort names.txt
```

Ana

Ben

Sorted A → Z



28

Create An Empty File Instantly

Make a new file or refresh its timestamp

- touch name creates the file if it's missing
- On an existing file, it updates the timestamp
- The fastest way to scaffold new files

SYNTAX

```
touch notes.txt
```

A terminal window with a dark background and a title bar showing 'user@linux: ~'. The command '\$ touch notes.txt' is entered in green text. Below the command, the word 'Created!' is displayed in large, bold, orange text.

```
user@linux: ~  
  
$ touch notes.txt  
  
Created!
```

29

See Who's Logged In Right Now

Check every active user and what they're running

- `w` lists every logged-in user at a glance
- Shows their login time and current command
- More detailed than the simpler `who` command

SYNTAX

w

```
user@linux: ~  
  
$ w  
  
USER  TTY  WHAT  
ana   pts0 bash  
  
1 user online
```

30

See Who's Logged In, Simply

A quick list of every active login session

- who lists usernames and login times
- Simpler and faster than the w command
- Great for a quick check on a shared server

SYNTAX

who

```
user@linux: ~  
  
$ who  
ana pts/0 09:12  
  
1 session
```

`$ echo "Course complete!"`

You Did It!

You just learned 30 real Linux commands — go use them!

▶ **SUBSCRIBE**
English

▶ **SUBSCRIBE**
Dari / دری

@MyWebUniversity

@MyWebUniversity-Dari

on YouTube — for more Linux & command-line tutorials

