SIGPwny

General FA2025 • 2025-09-04

Intro to Terminal and Setup

Suchit Bapatla and Cameron Asher

Suchit Bapatla

- Helper
- MCS CS
- I am a black belt in TaeKwonDo

Cameron Asher

- President
- Stats & CS Major
- Joined SIGPwny without any security experience
- Can make macarons from scratch

Announcements

- Fall CTF registration is open!
  - sigpwny.com/fallctf
  - Beginner-friendly CTF on **Sunday, September 21st** from **12-6 PM**!
  - Free t-shirts and badges are first come, first serve!
    - We will try to prioritize those who registered

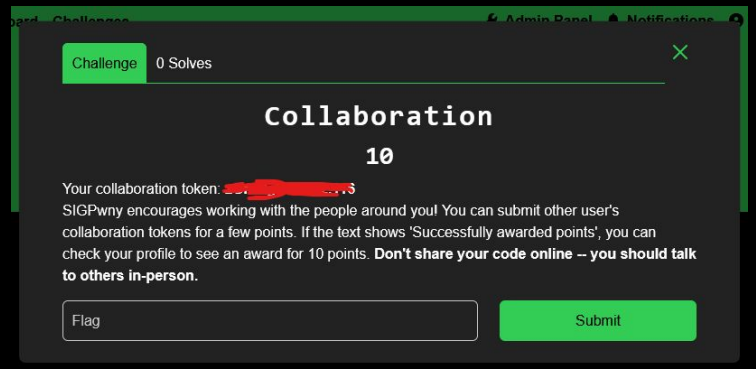
The "Don't Get Arrested" Slide

Computer Fraud and Abuse Act (CFAA)

- Attacking "protected" computers
- Anywhere between a fine and **TWENTY** years in jail.
- If you don't have **EXPLICIT** permission to break into it, **DON'T**

Pwny CTF (ctf.sigpwny.com)

- Create an account right now!
- Where we put our challenges for you to build hands on experience
- Solve challenges, find flags, submit flags on website
- Talk to your neighbors and solve the collaboration challenge!



ctf.sigpwny.com

sigpwny{lets_get_setup}

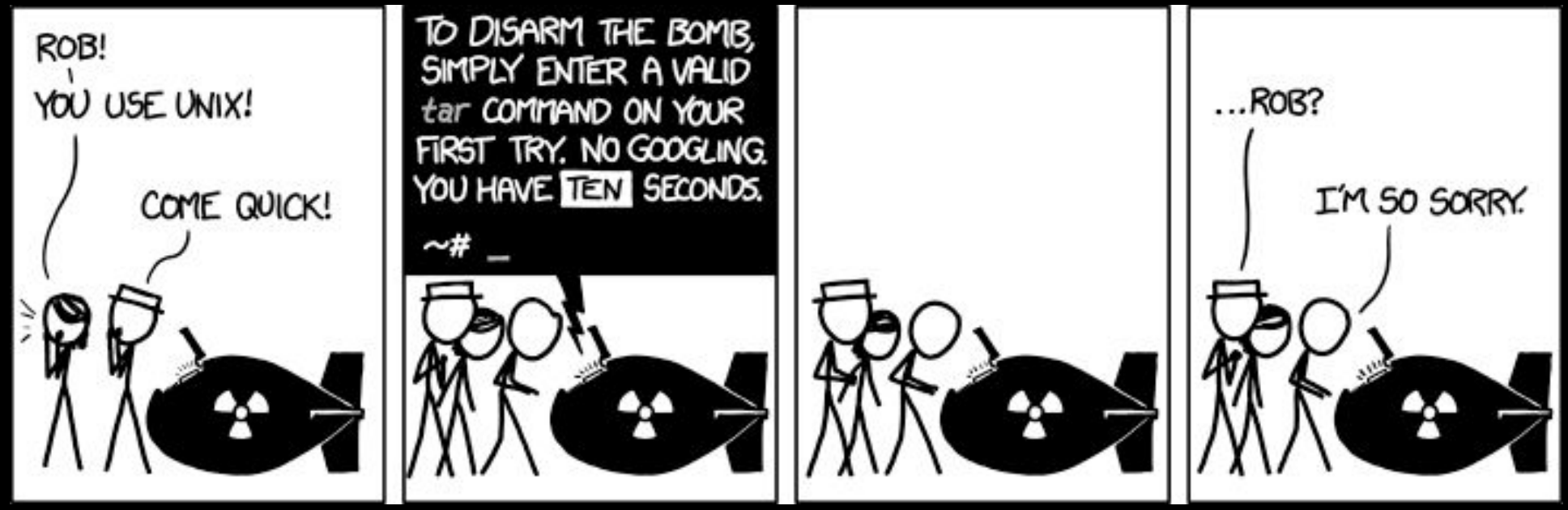


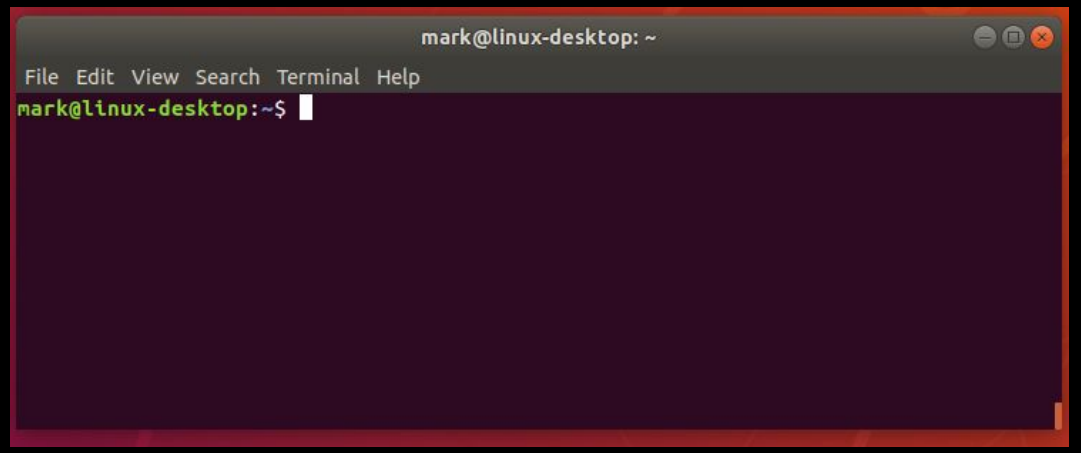
Table of Contents

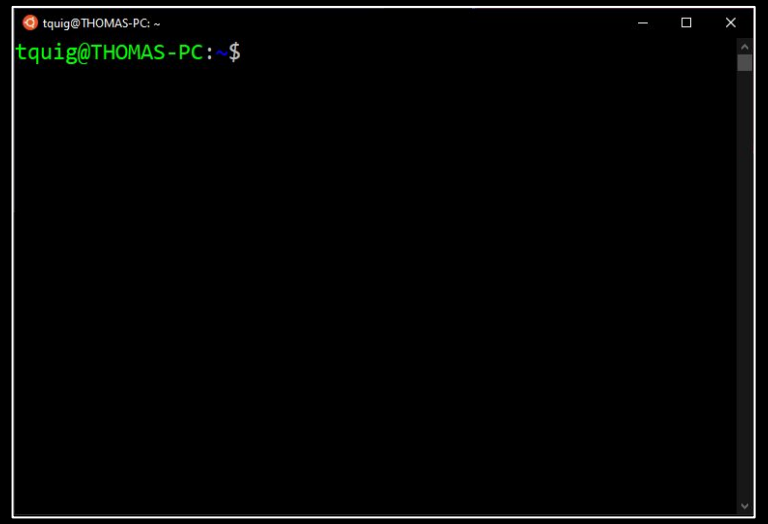
1. What is a shell
   - *I want one!*
   - Install a package manager, terminal emulator
   - WSL or a virtual machine?
2. Using the shell
   - Platform differences
   - Useful builtins and utilities
3. Setup for reverse-engineering (rev) and binary exploitation (pwn) meetings
   - Installing Ghidra, pwntools, and GDB

> The Terminal

"It’s where things happen" - Ravi

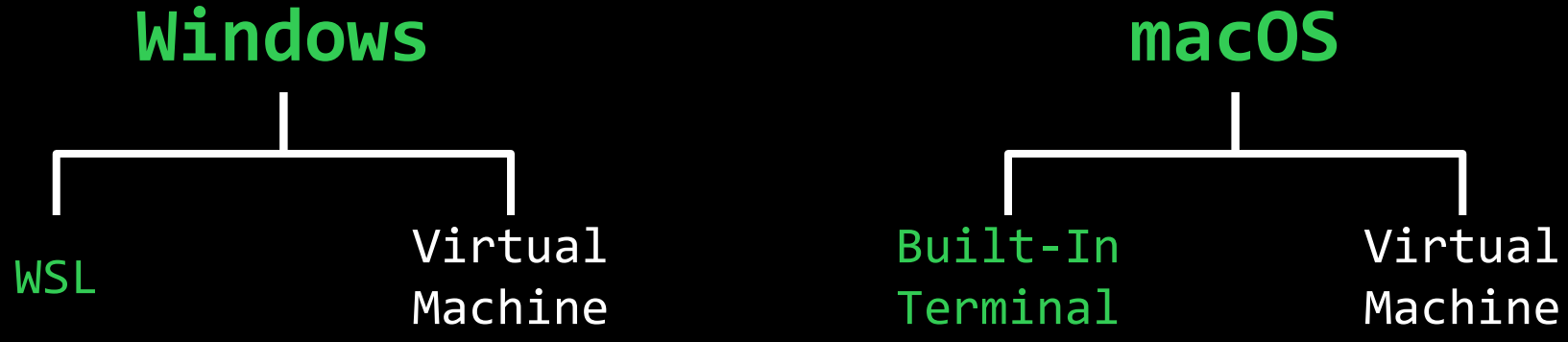
⌥⌘1
/dev/ttys000
→ CSAW2020 ls
bard grid kui_blox1_sol.png
bard.hop grid_solve.py libc-2.27.so
ezbreezy krakme.exe solve_ezbreezy.py
→ CSAW2020


mark@linux-desktop: ~
File Edit View Search Terminal Help
mark@linux-desktop:~$


tquig@THOMAS-PC: ~
tquig@THOMAS-PC:~$

Linux

You’re good to go!


Windows
WSL
Virtual Machine
macOS
Built-In Terminal
Virtual Machine

PowerShell? Command Prompt?

- Those are shells too!
- However, the Windows terminal is built differently than the Mac and Linux terminals (which are both UNIX based)
  - Different command structure/rules
  - Less support for CTF relevant applications

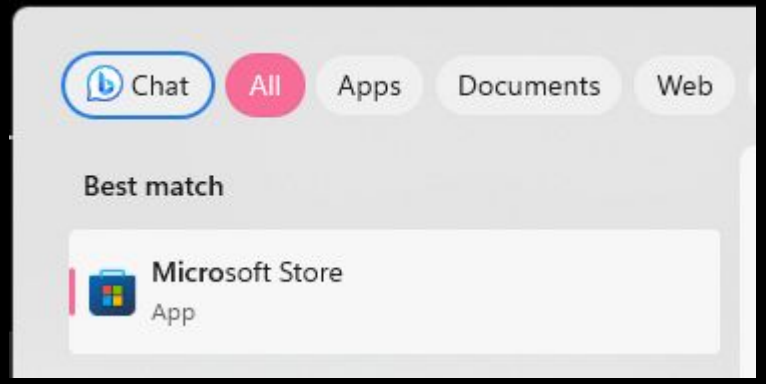
Windows Subsystem for Linux (WSL)

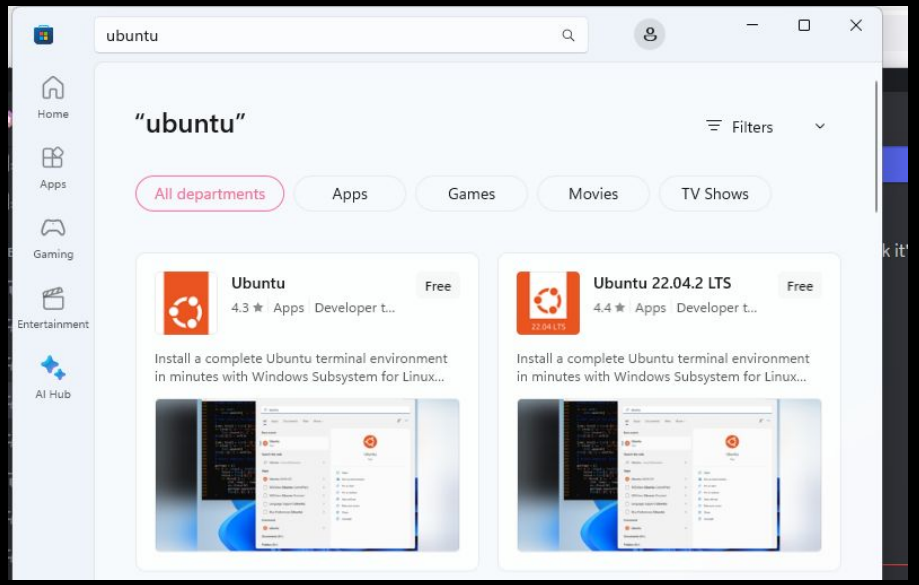
Mac users hold tight…

Linux users … I hope you know this stuff already ;)

Installation (Windows Store)

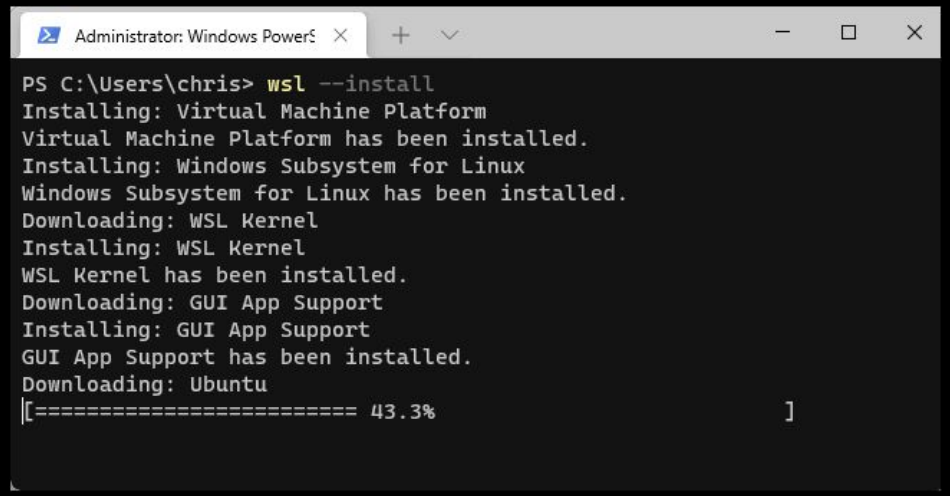
Open the Microsoft Store → Search "Ubuntu" → Install "Ubuntu" (use the one without a version number)





Installation

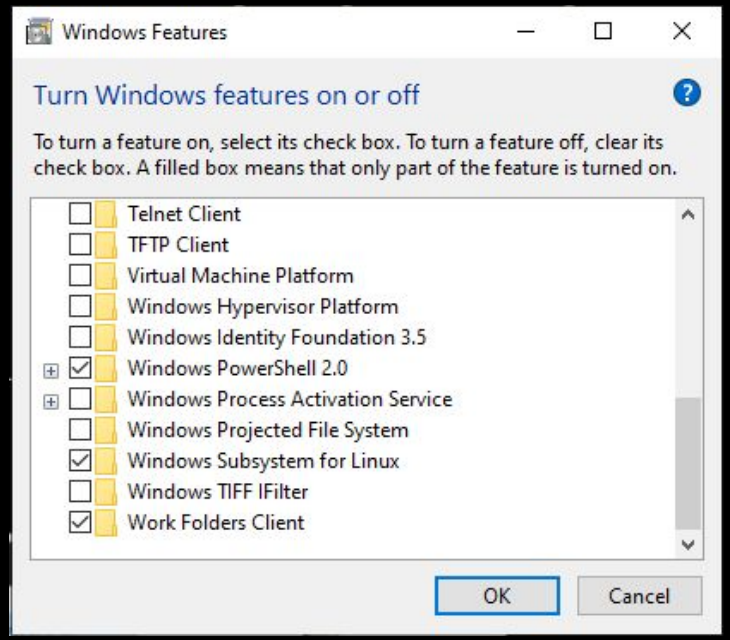
- Open command prompt as administrator
  - (Start button → type **`cmd`** → right click → "Run as Administrator")
- Type **`wsl --install [-d <distro>]`**
- Restart computer
- You should be able to launch Ubuntu from the start menu



Installation (older Windows version)

If you get a command not found error when trying to run `wsl --install`, try this

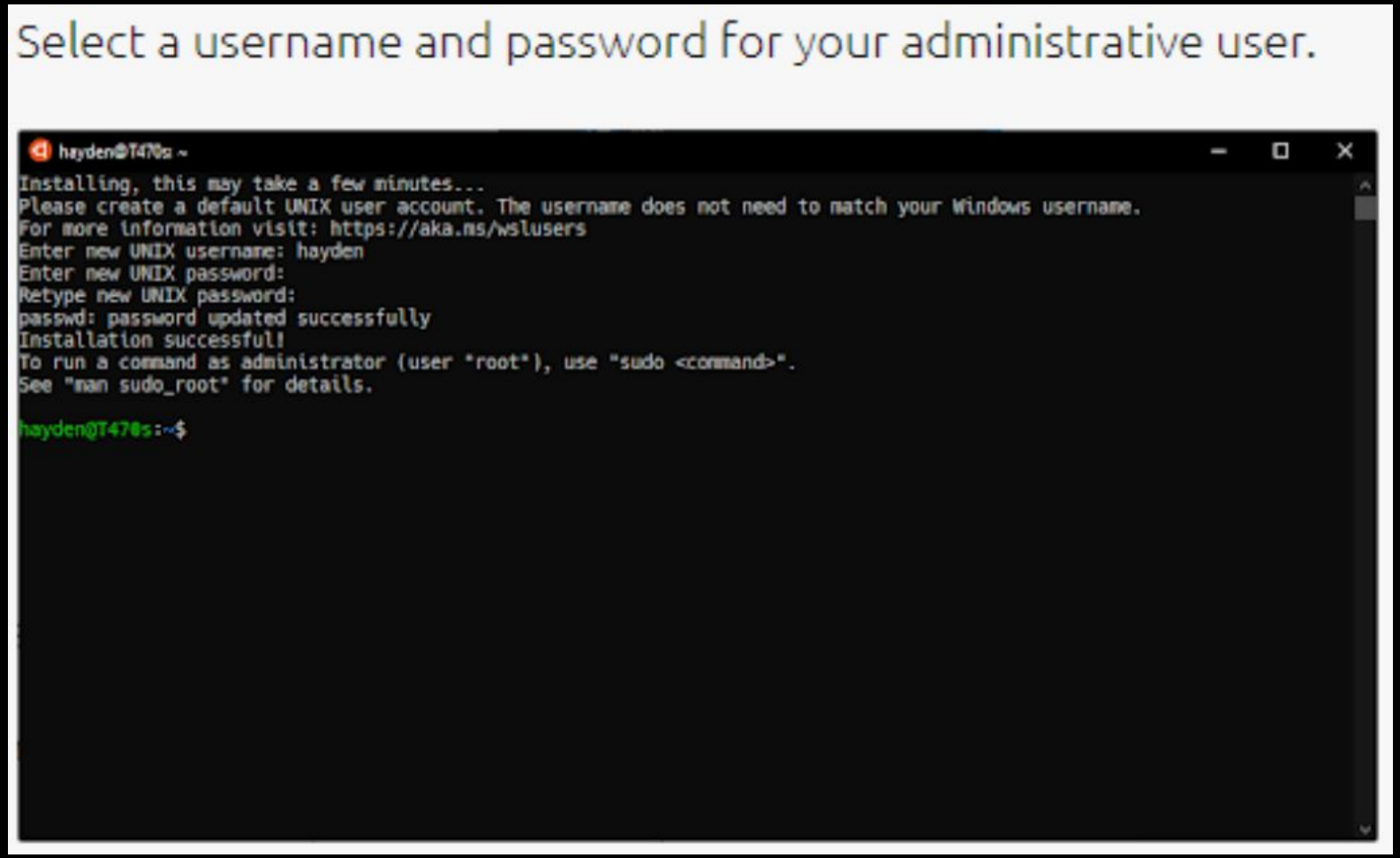
- Go to the Windows search bar
- Search "Turn Windows features on or off"
- Check "Virtual Machine Platform" and "Windows Subsystem for Linux"
- Restart computer



Do you have Docker for Windows?

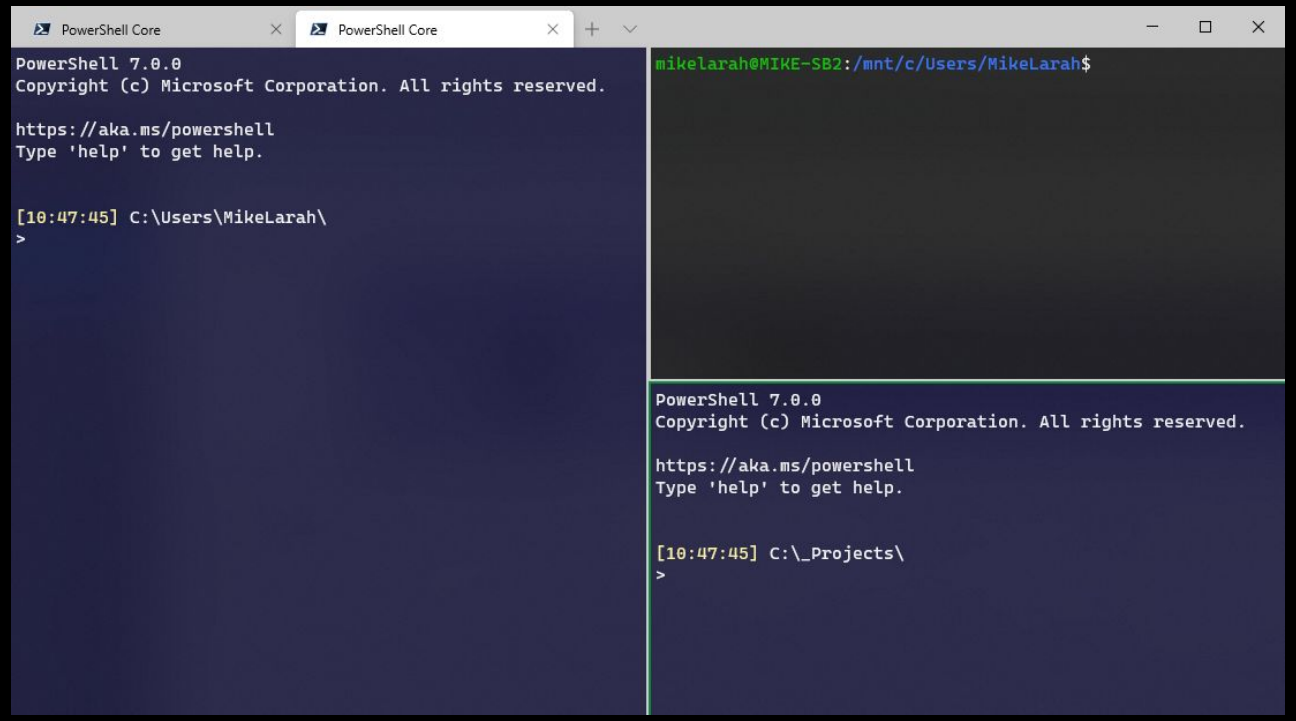
- If you are in CS 128 or have previously installed Docker for Windows, then you already have WSL installed.
- Run `wsl --list` to view your installed Linux distros.
- Your default should be set to “Ubuntu”
- If it is set to docker-desktop or docker-desktop-data (or anything else) run `wsl --set-default Ubuntu`

Set a "root" username and password!



Windows Terminal (Optional)

- Nice for managing multiple types of shells (e.g. tabs for PowerShell, Kali on WSL, Debian on WSL, all in one terminal)
- Download from the Microsoft Store



macOS Terminal

Command
+ Space → Search “Terminal” →

Homebrew

- AKA "brew"
- Popular package installation tool on macOS
- Install: https://brew.sh
- Search: https://formulae.brew.sh/
- To install tools with brew, use `brew install <package>`
- Example: `brew install wget`

iTerm2 (Optional)

iTerm2

iTerm2 is a terminal emulator for macOS that does amazing things.

- Modern replacement for the basic macOS Terminal
- https://iterm2.com
- See also: kitty, Alacritty, WezTerm, Hyper…

Installing Python

Installing Python

pyenv allows you to easily manage and switch between different Python versions (e.g. 3.12 and 3.8)

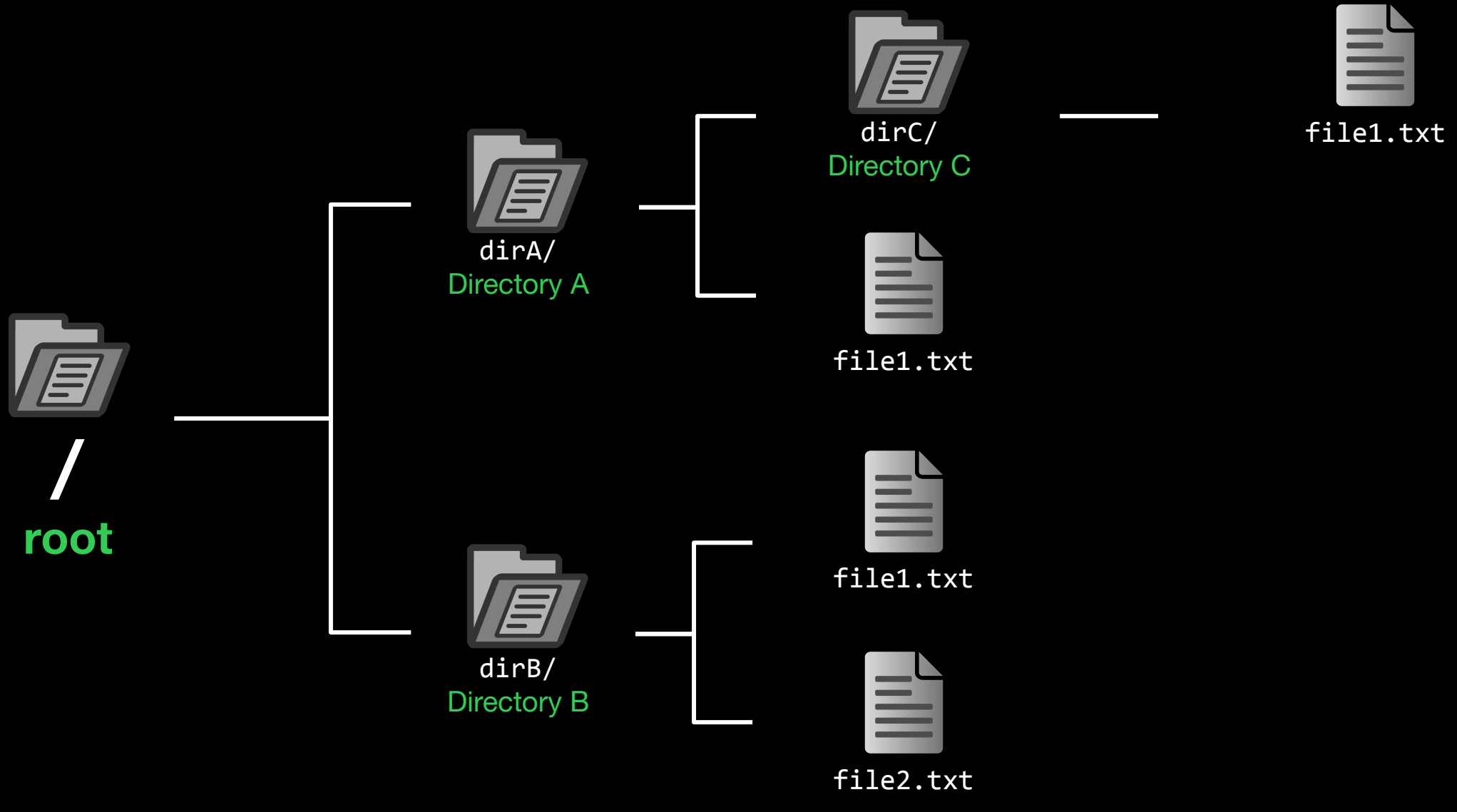
This is **preferred** over a system installation of Python - but both will work fine!

```
$ curl https://pyenv.run | bash
 - add the EXPORT ... snippet in output to the
   end of your ~/.bashrc OR ~/.zshrc
$ pyenv install 3.11
$ source ~/.bashrc / source ~/.zshrc
$ pyenv global 3.11
```

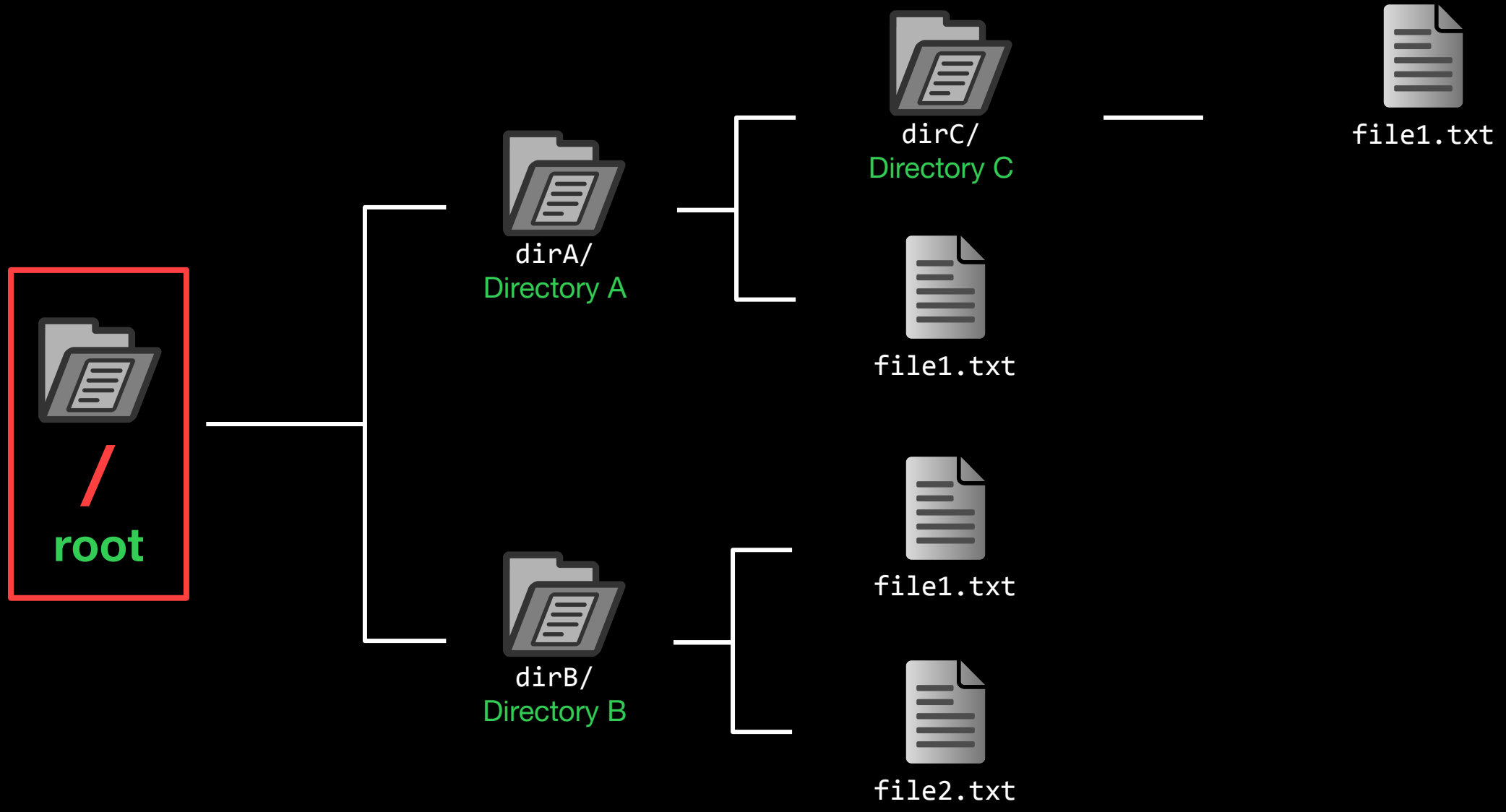
Unix Crash Course

Navigate a file system!

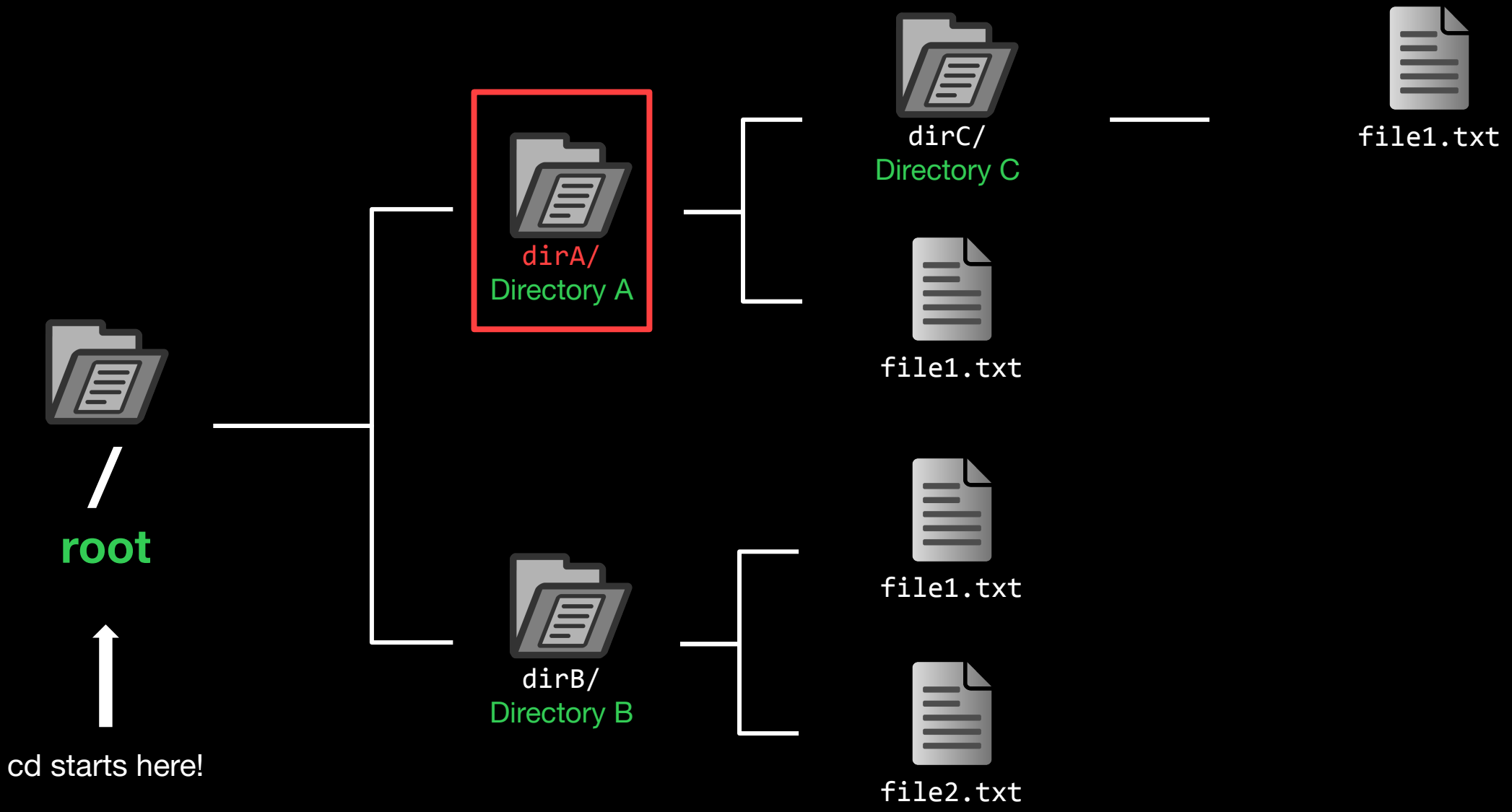
Filesystems



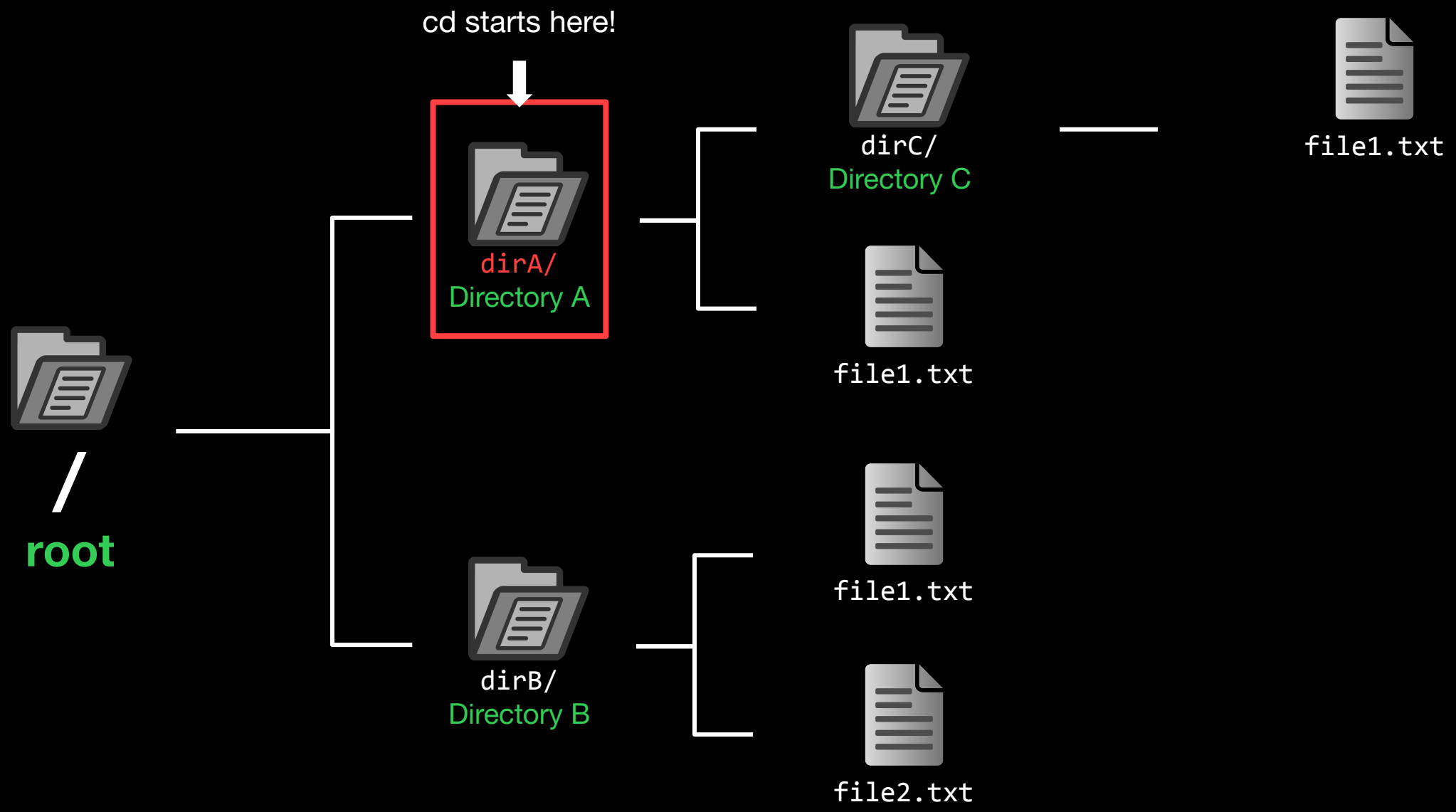
cd /

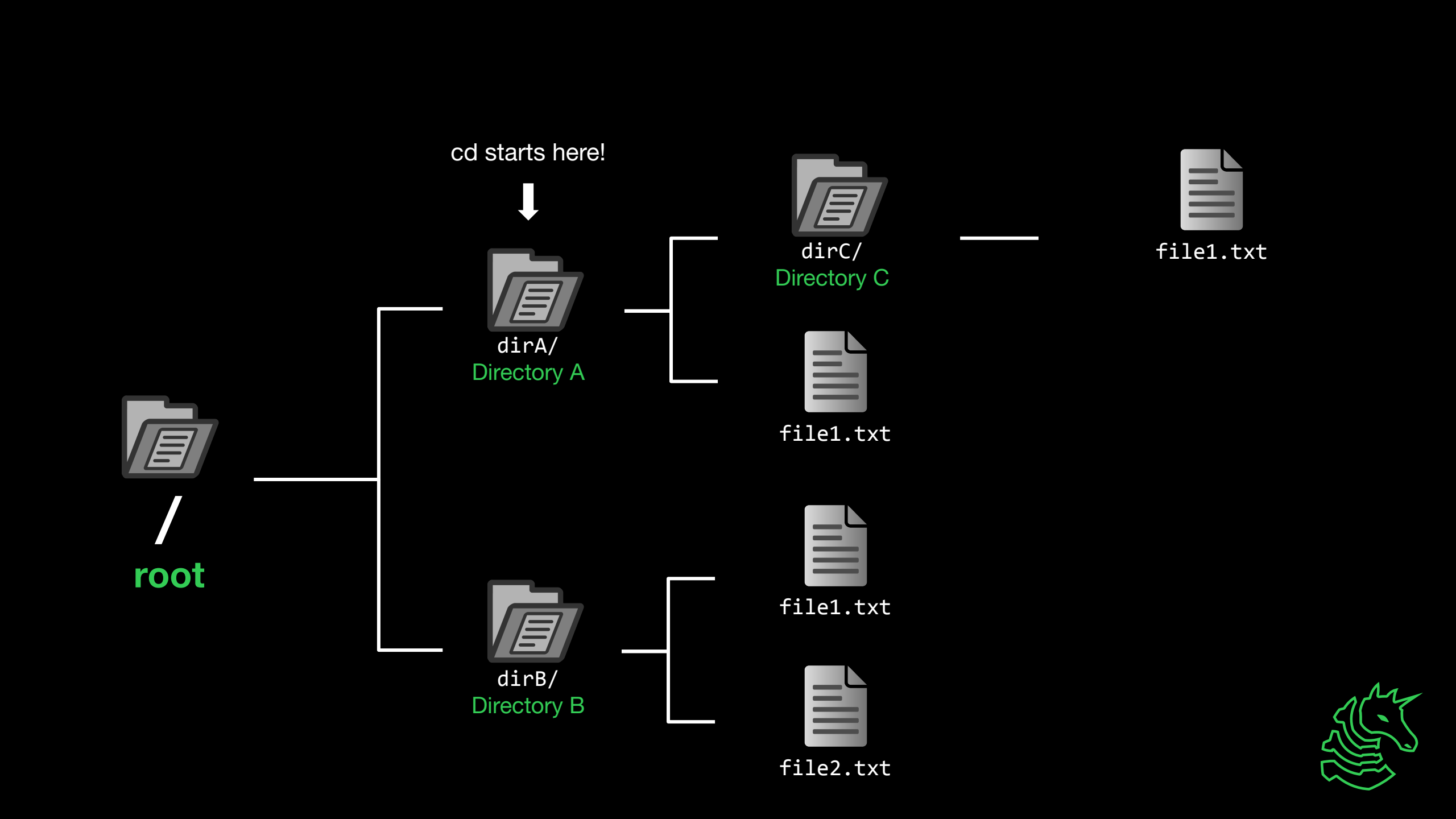

/
root
dirA/
Directory A
dirB/
Directory B
dirC/
Directory C
file1.txt
file1.txt
file1.txt
file2.txt
file1.txt

cd dirA

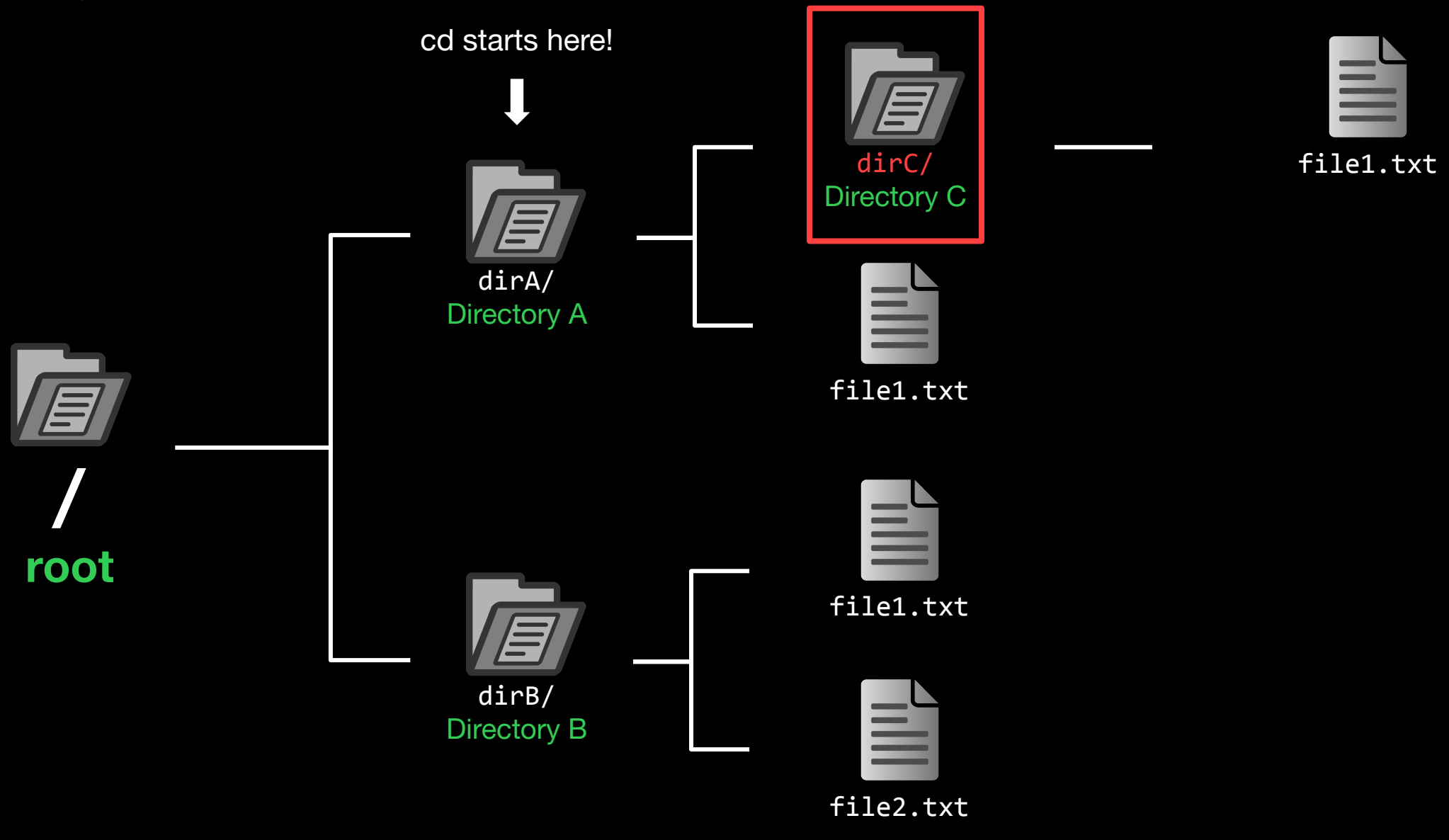

/
root
cd starts here!
dirA/
Directory A
dirB/
Directory B
dirC/
Directory C
file1.txt
file1.txt
file1.txt
file2.txt

cd dirA

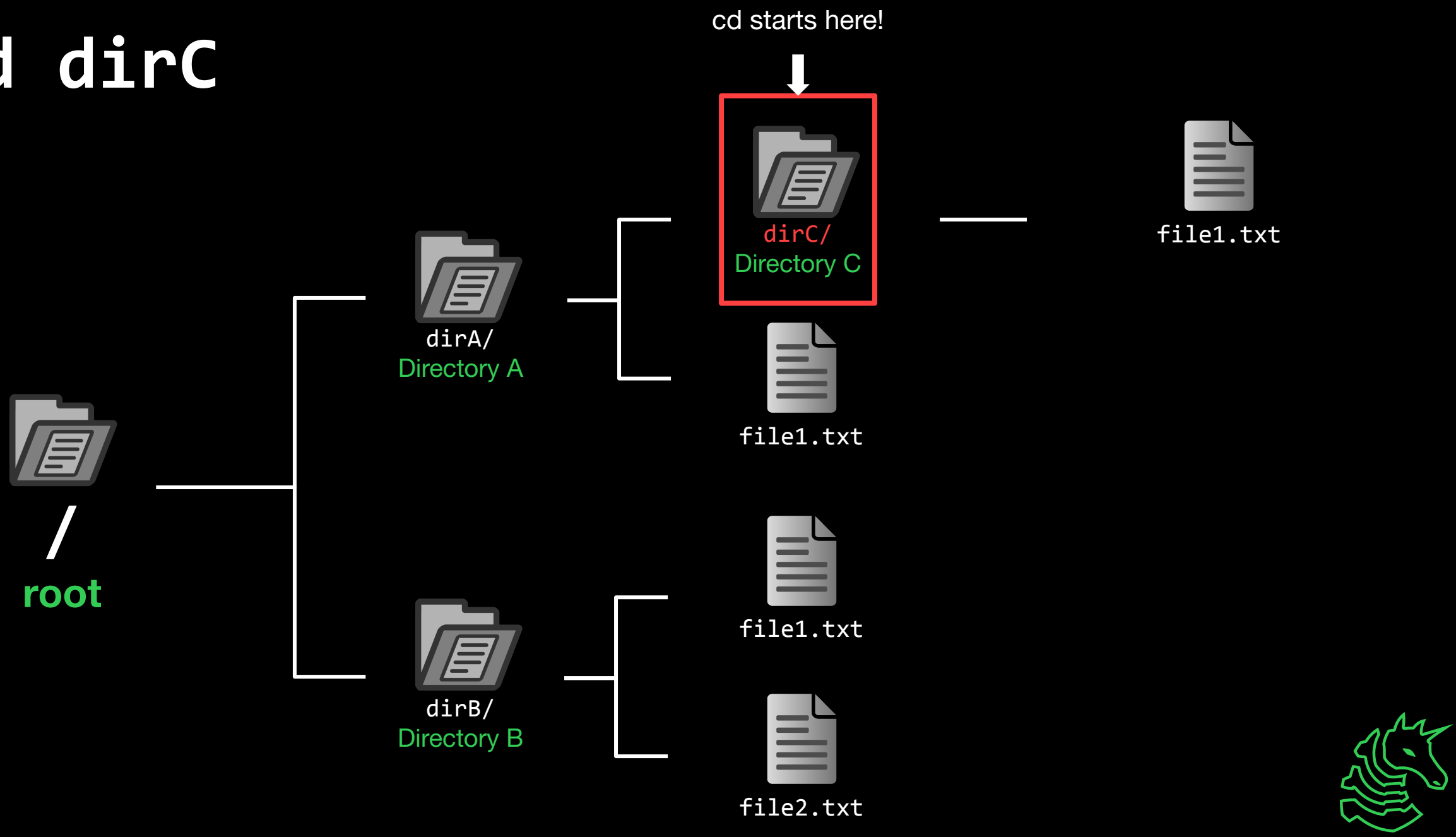

cd starts here!
/
root
dirA/
Directory A
dirB/
Directory B
dirC/
Directory C
file1.txt
file1.txt
file1.txt
file2.txt

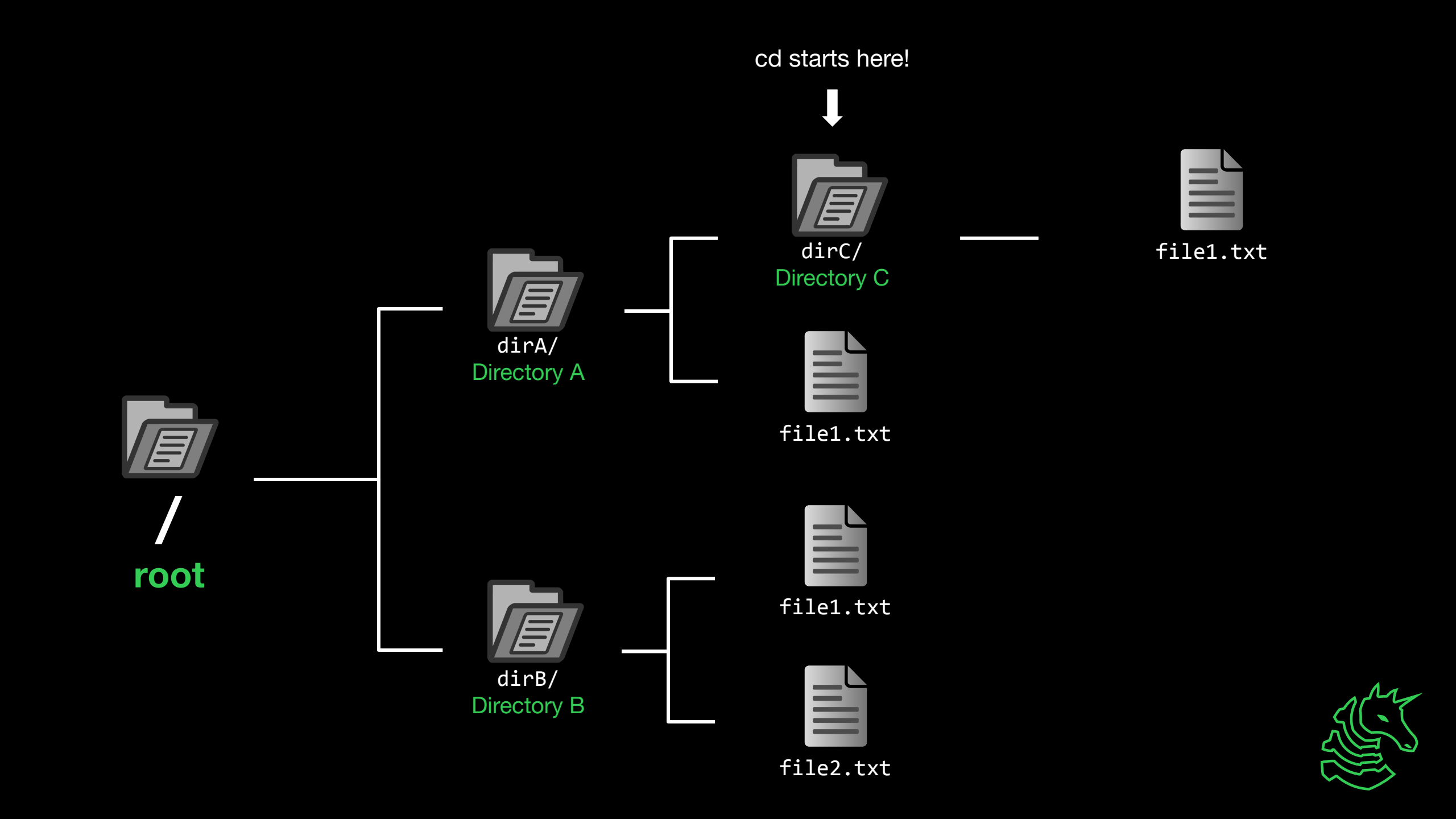

cd starts here!
/
root
dirA/
Directory A
dirB/
Directory B
dirC/
Directory C
file1.txt
file1.txt
file1.txt
file2.txt
file1.txt

cd dirC

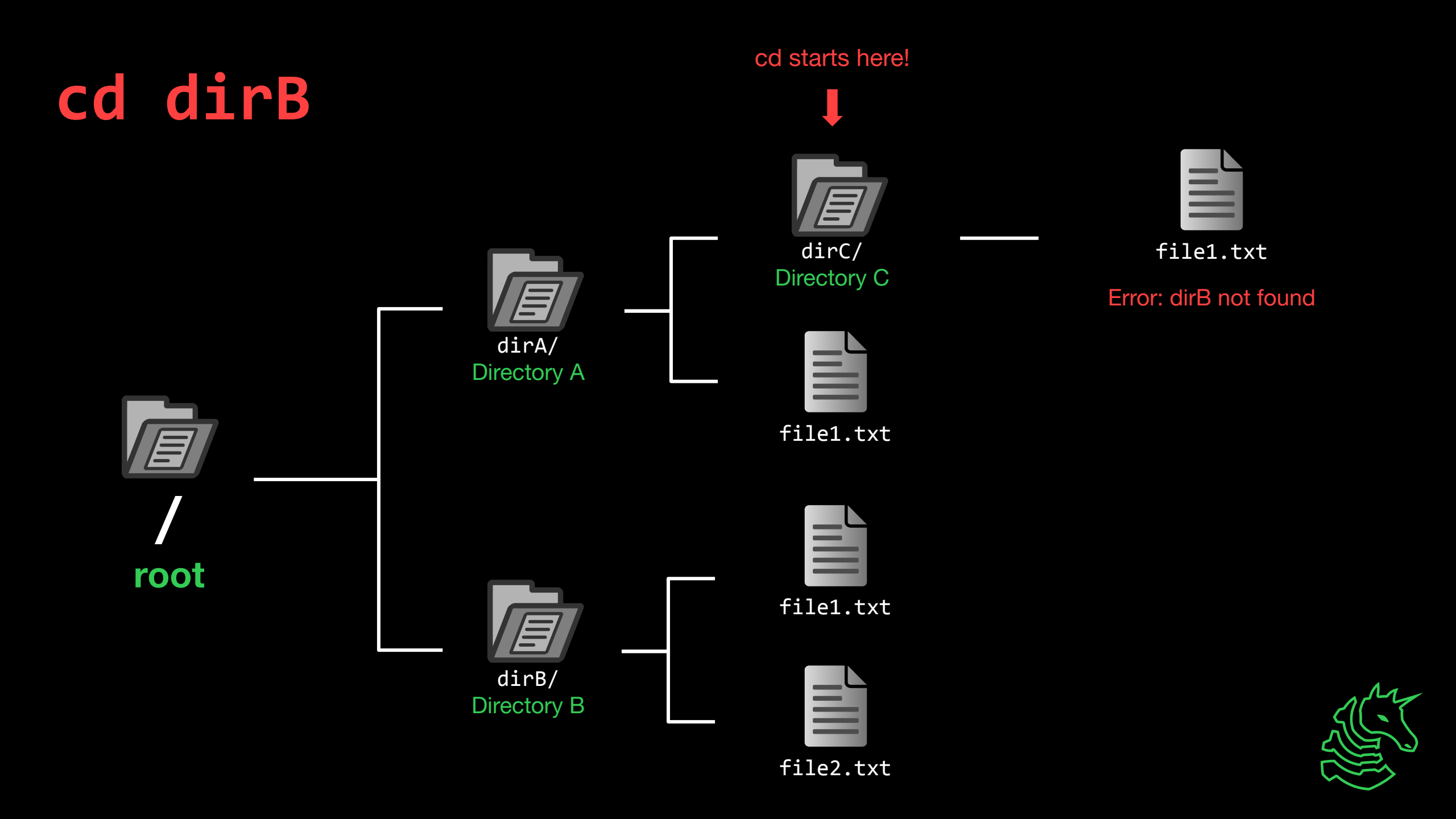

cd starts here!
/
root
dirA/
Directory A
dirB/
Directory B
dirC/
Directory C
file1.txt
file1.txt
file1.txt
file2.txt

cd dirC

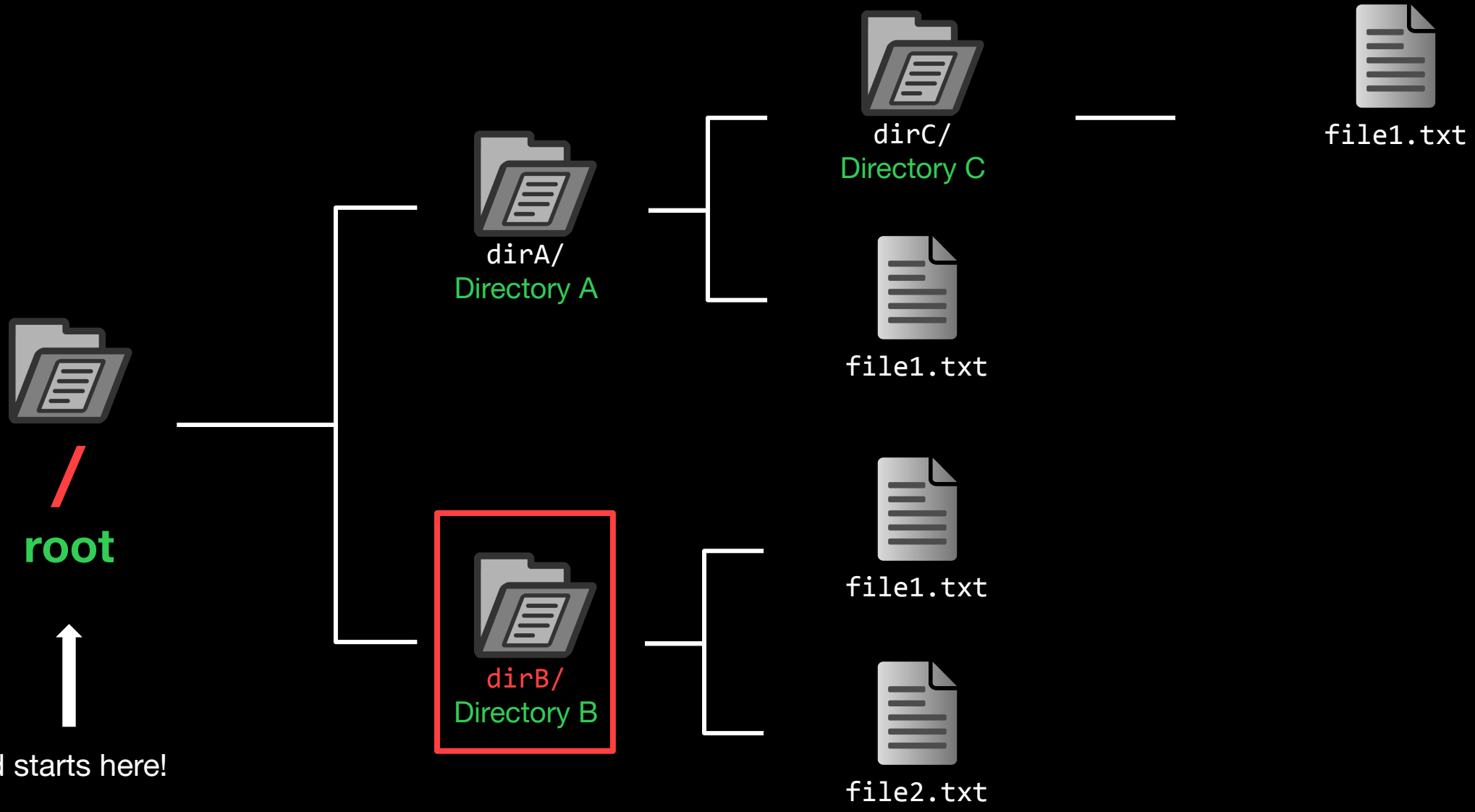

cd starts here!
dirC/
Directory C
file1.txt
dirA/
Directory A
file1.txt
/
root
file1.txt
dirB/
Directory B
file2.txt


cd starts here!
/
root
dirA/
Directory A
dirB/
Directory B
dirC/
Directory C
file1.txt
file1.txt
file1.txt
file2.txt
file1.txt

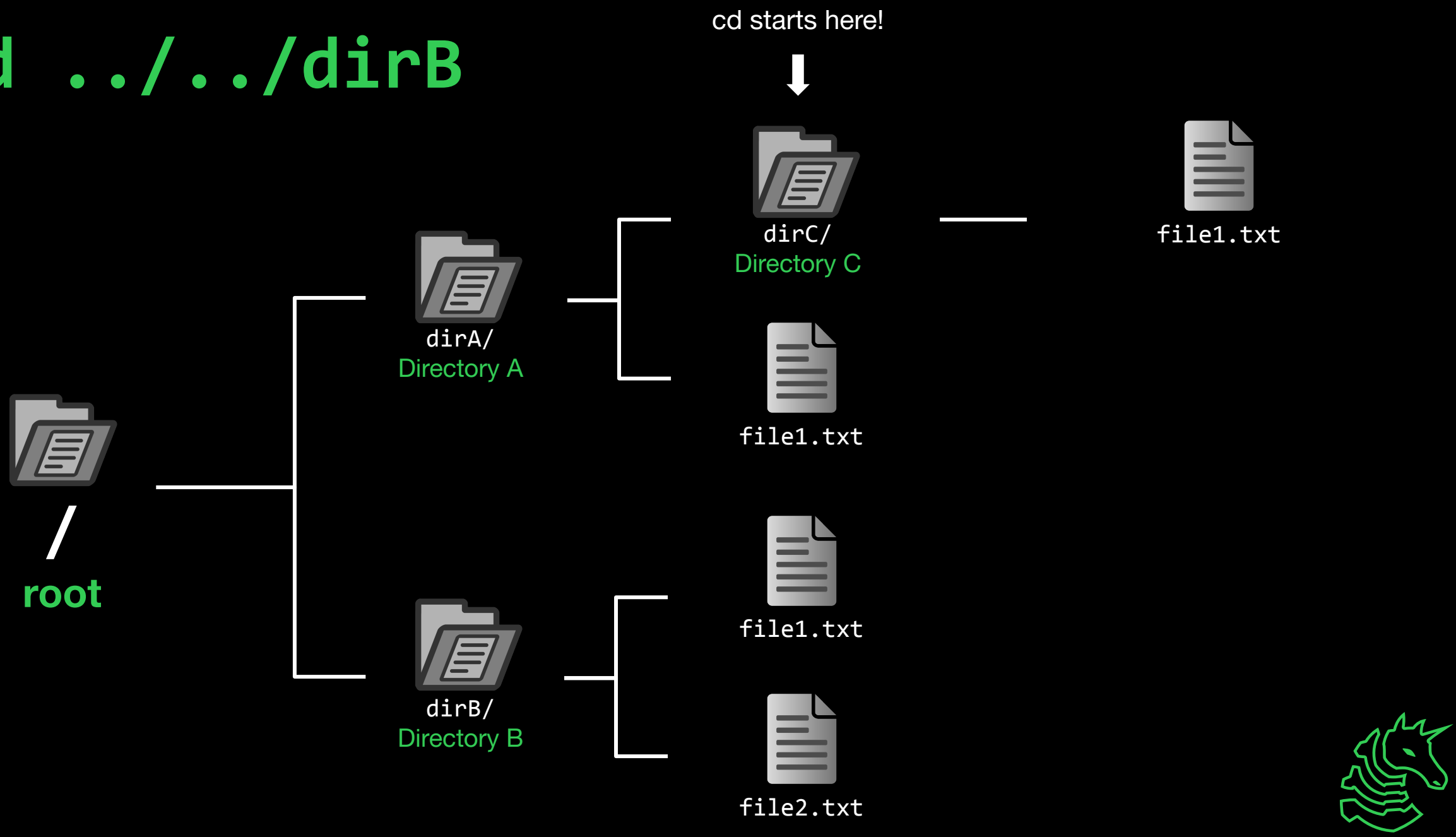
cd dirB


cd starts here!
dirC/
Directory C
file1.txt
Error: dirB not found
dirA/
Directory A
file1.txt
/
root
file1.txt
dirB/
Directory B
file2.txt

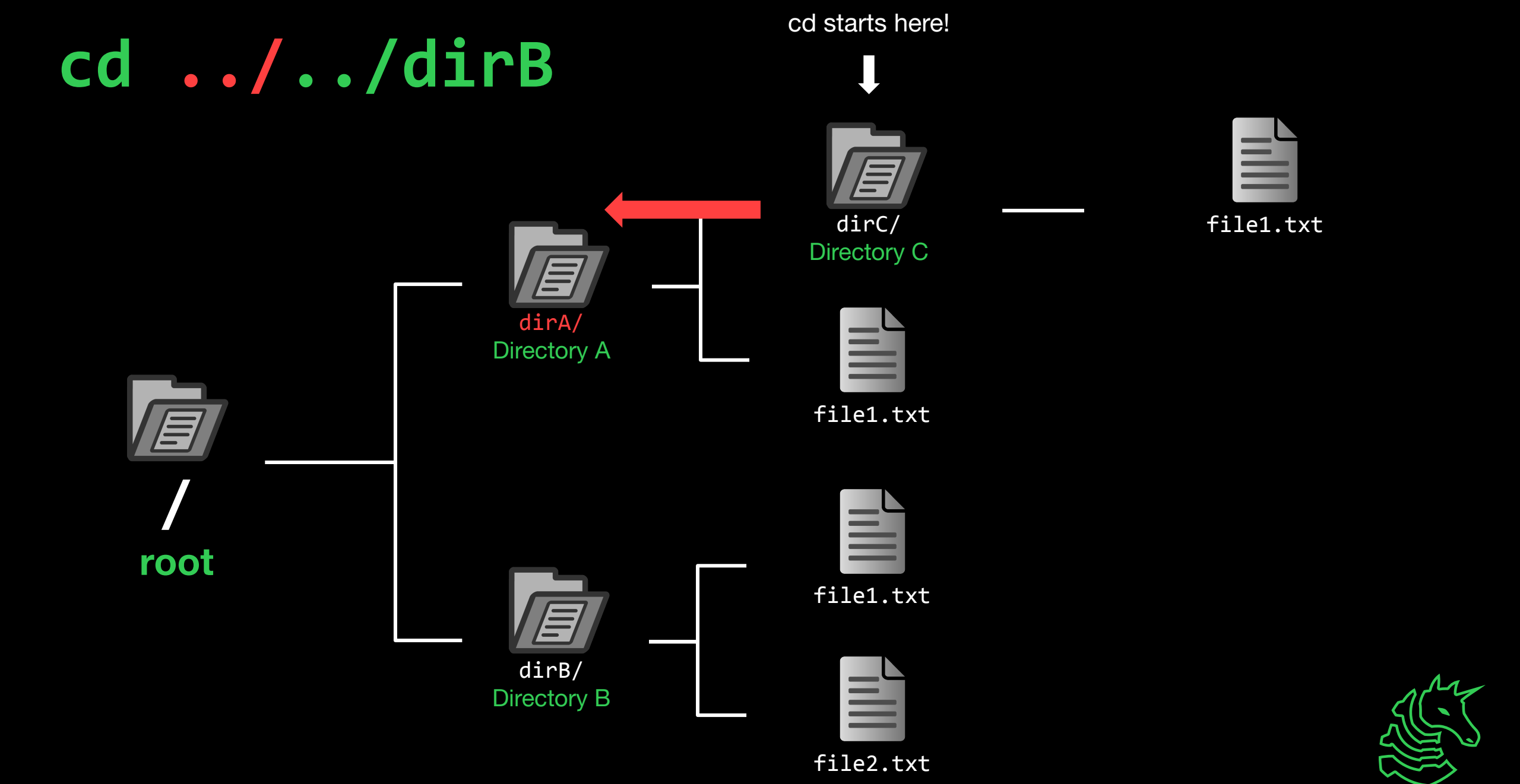
cd /dirB


/
root
cd starts here!
dirA/
Directory A
dirB/
Directory B
dirC/
Directory C
file1.txt
file1.txt
file1.txt
file2.txt
file1.txt

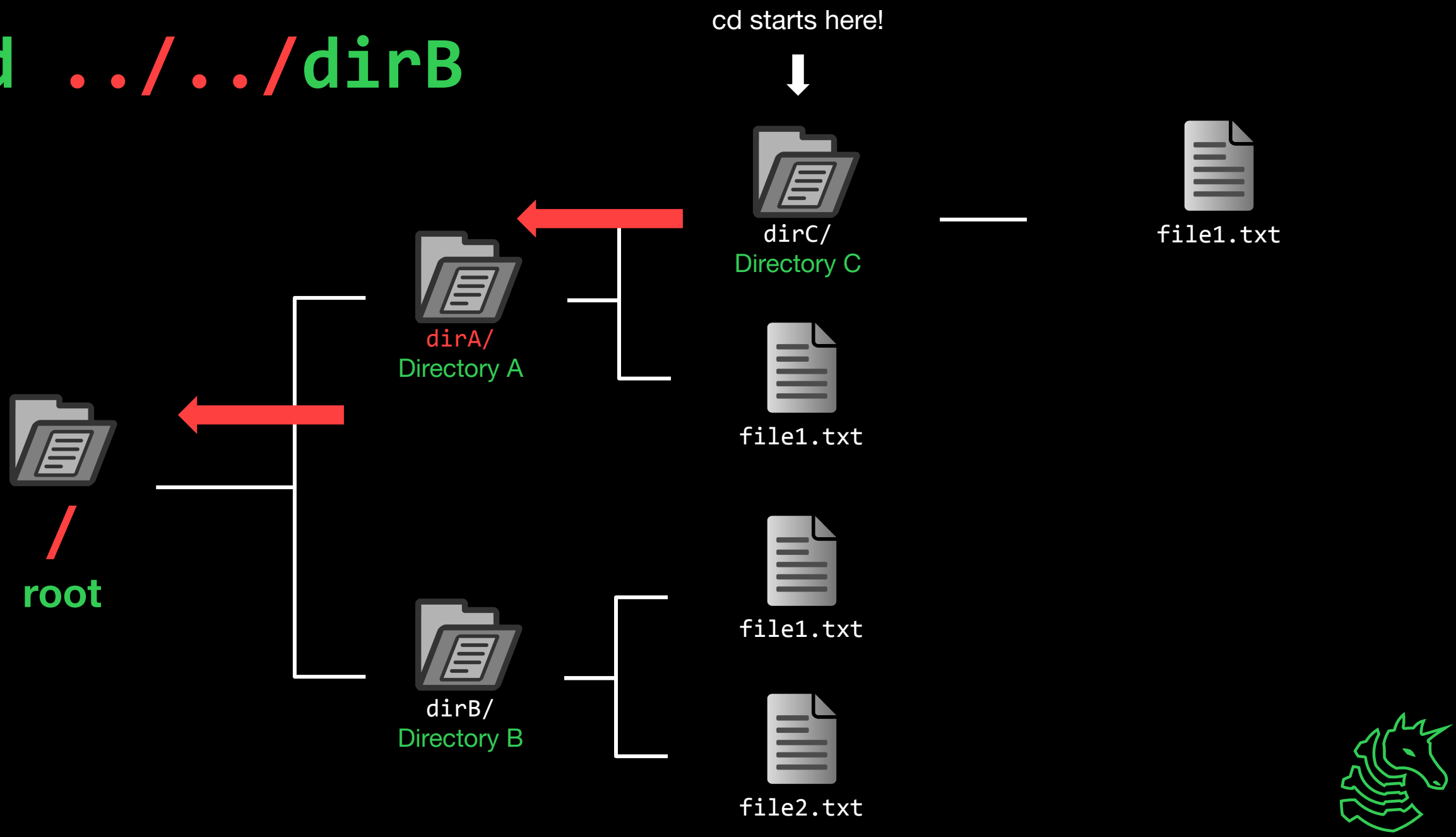
cd ../../dirB


cd starts here!
/
root
dirA/
Directory A
dirB/
Directory B
dirC/
Directory C
file1.txt
file1.txt
file1.txt
file2.txt
file1.txt

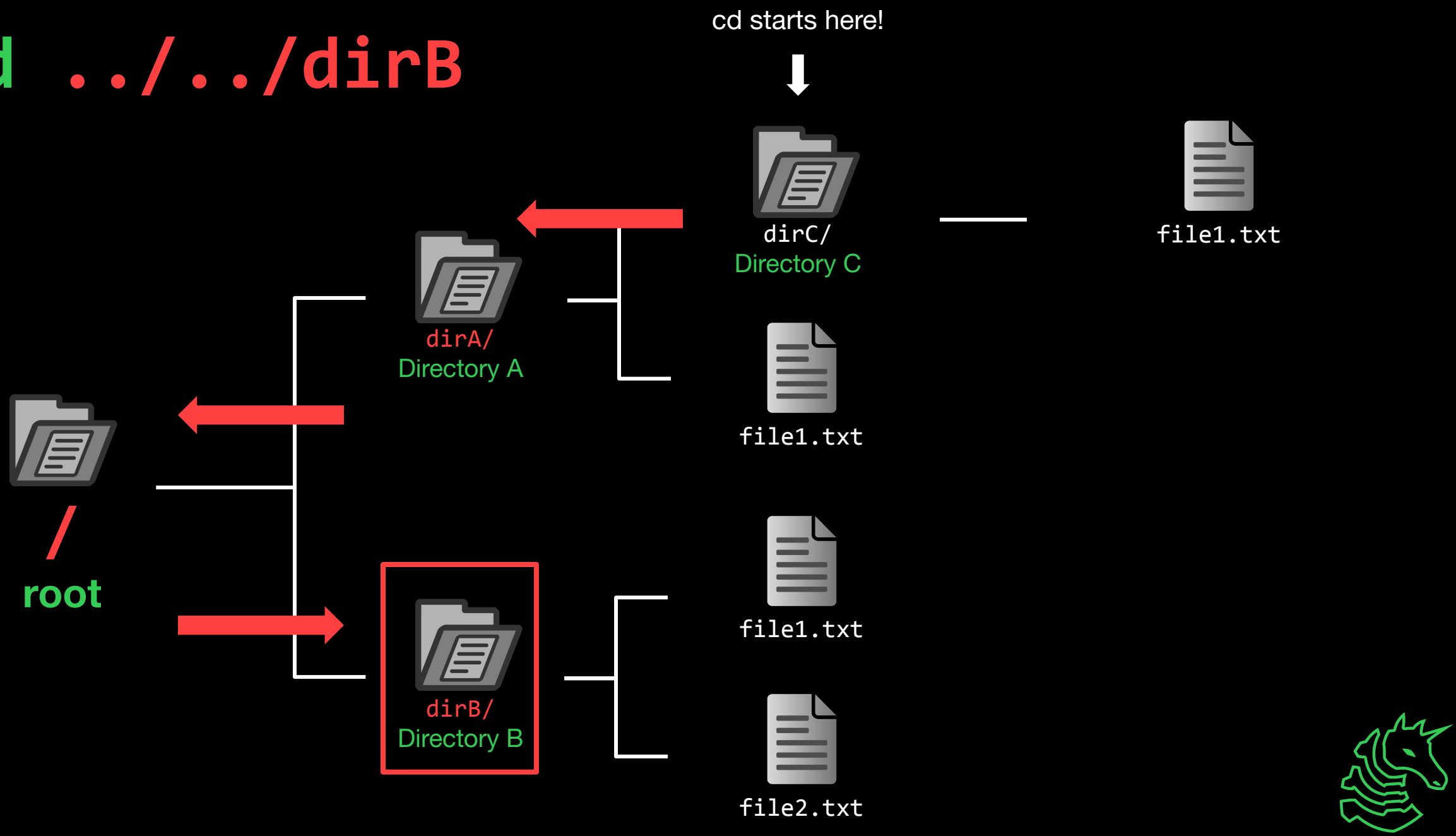
cd ../../dirB


cd starts here!
dirC/
Directory C
file1.txt
dirA/
Directory A
file1.txt
/
root
file1.txt
dirB/
Directory B
file2.txt

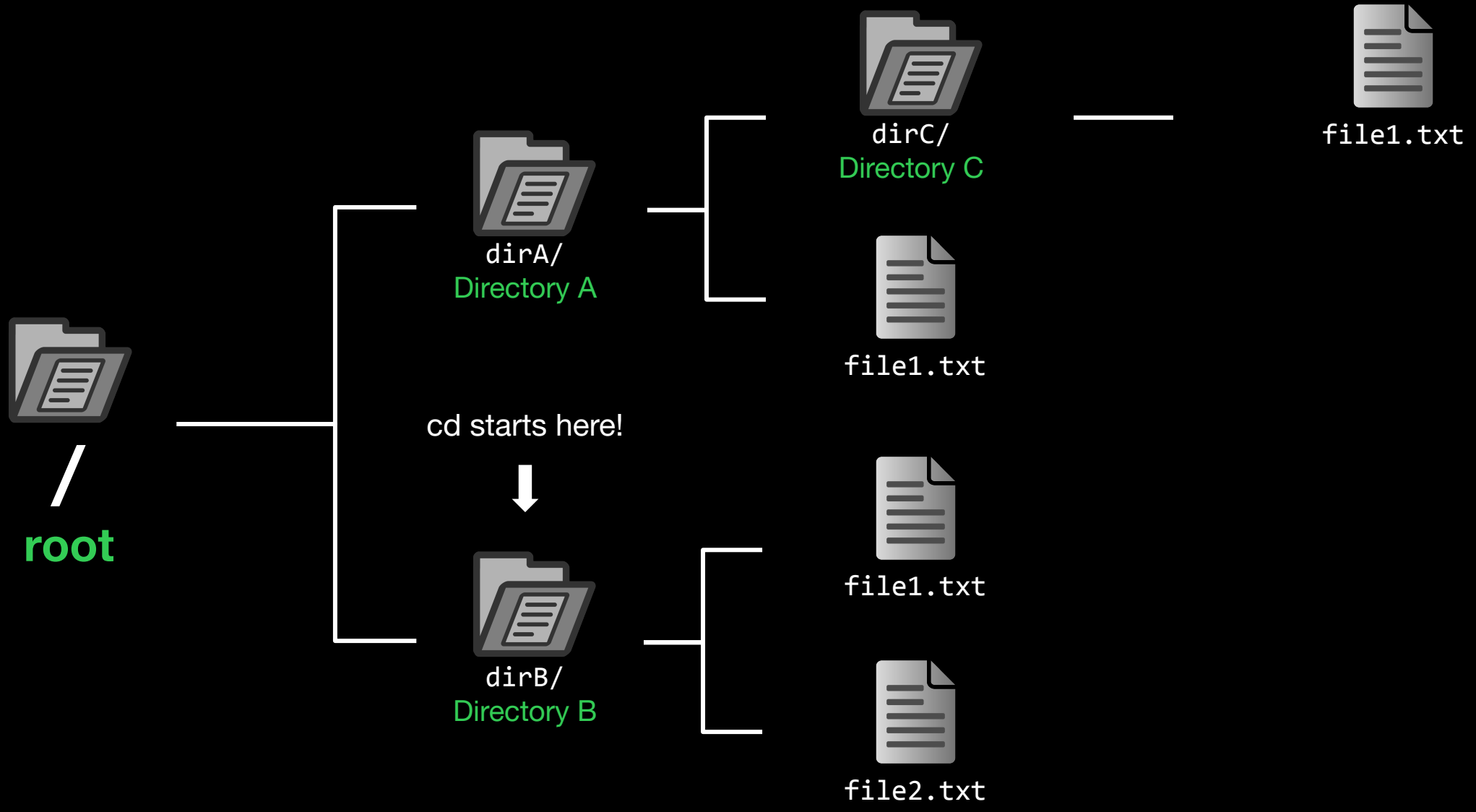
cd ../../dirB


cd starts here!
dirC/
Directory C
file1.txt
dirA/
Directory A
file1.txt
/
root
file1.txt
dirB/
Directory B
file2.txt

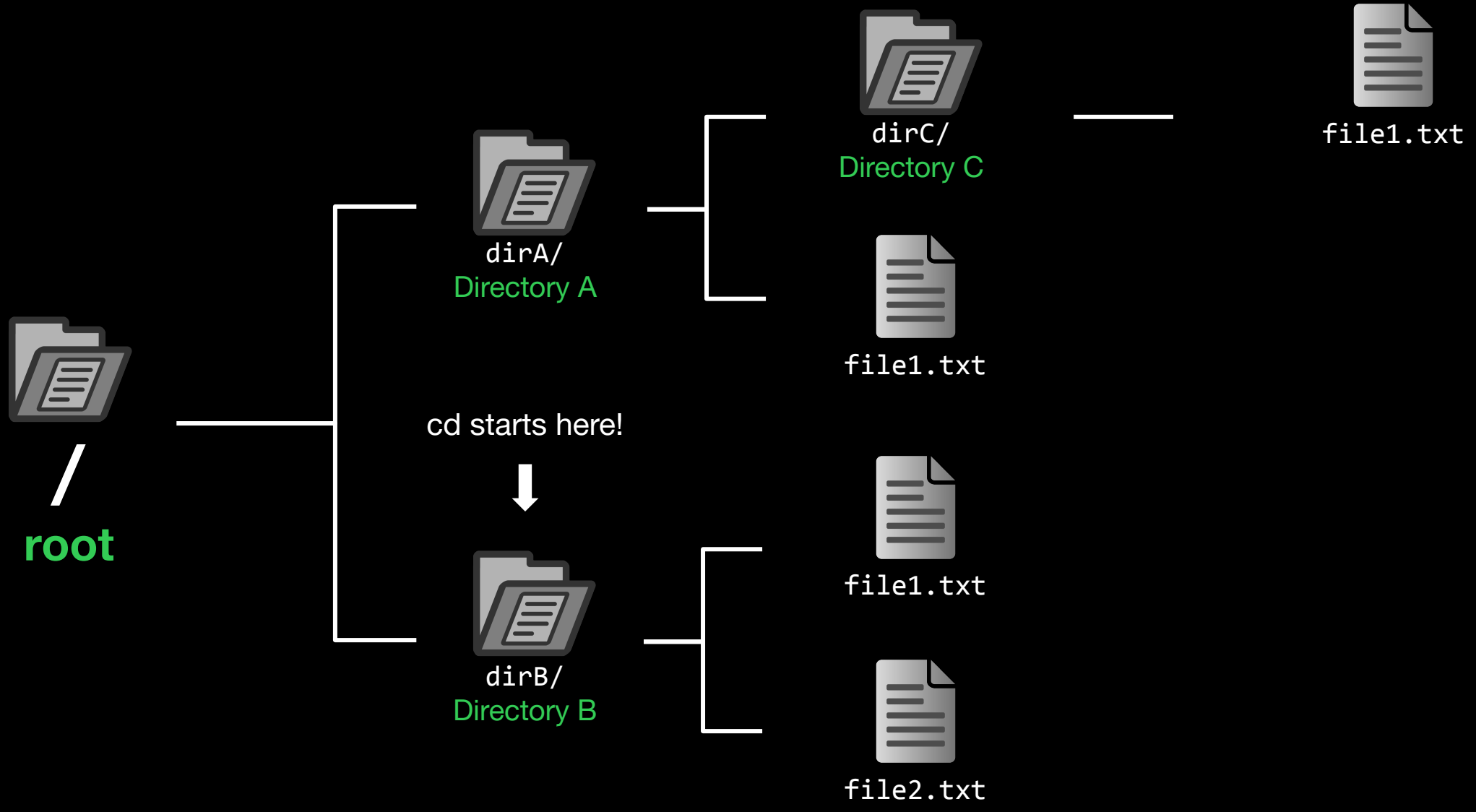
cd ../../dirB


cd starts here!
dirC/
Directory C
file1.txt
dirA/
Directory A
file1.txt
/
root
file1.txt
dirB/
Directory B
file2.txt

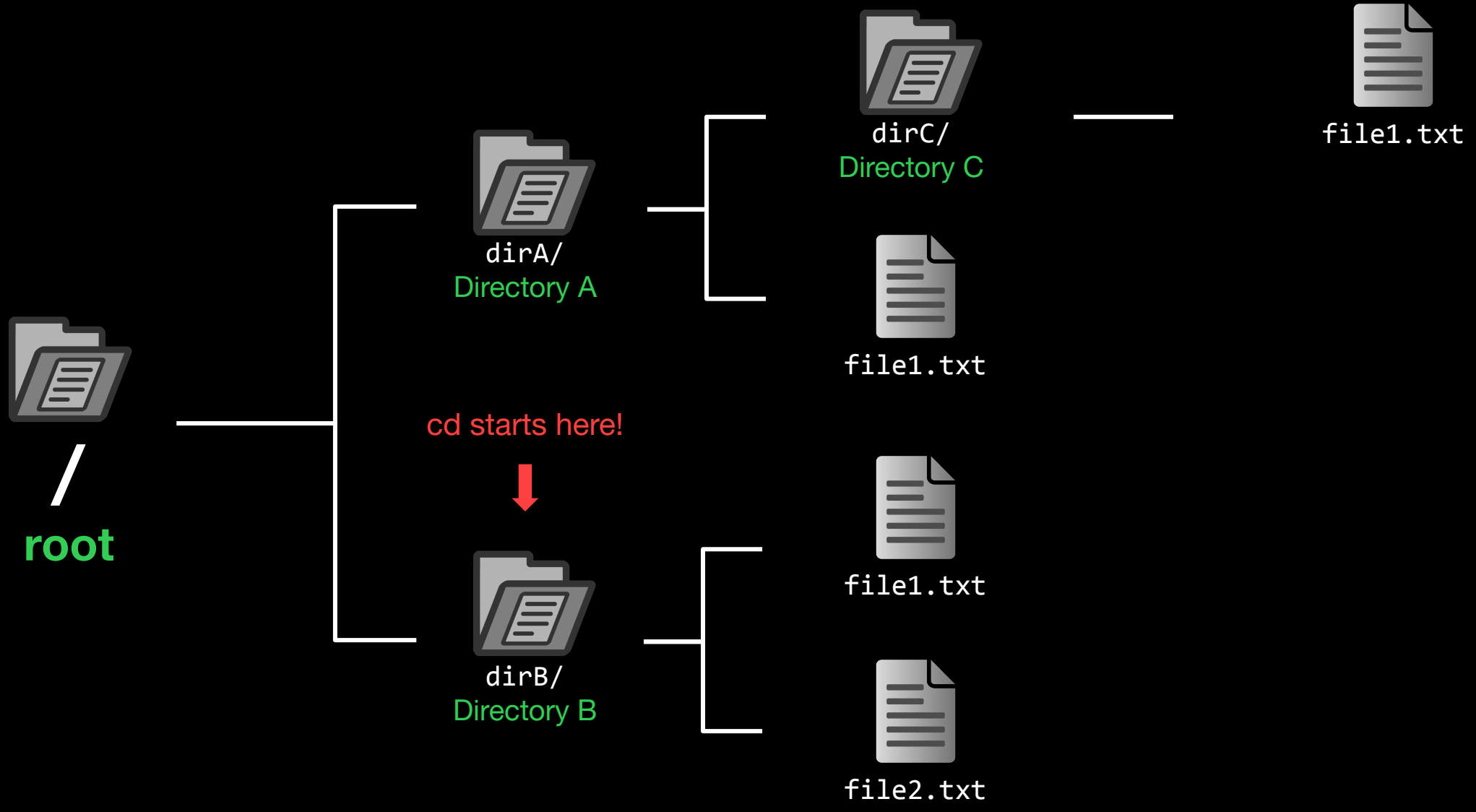
cd ../../dirB


/
root
dirA/
Directory A
dirC/
Directory C
file1.txt
file1.txt
cd starts here!
dirB/
Directory B
file1.txt
file2.txt

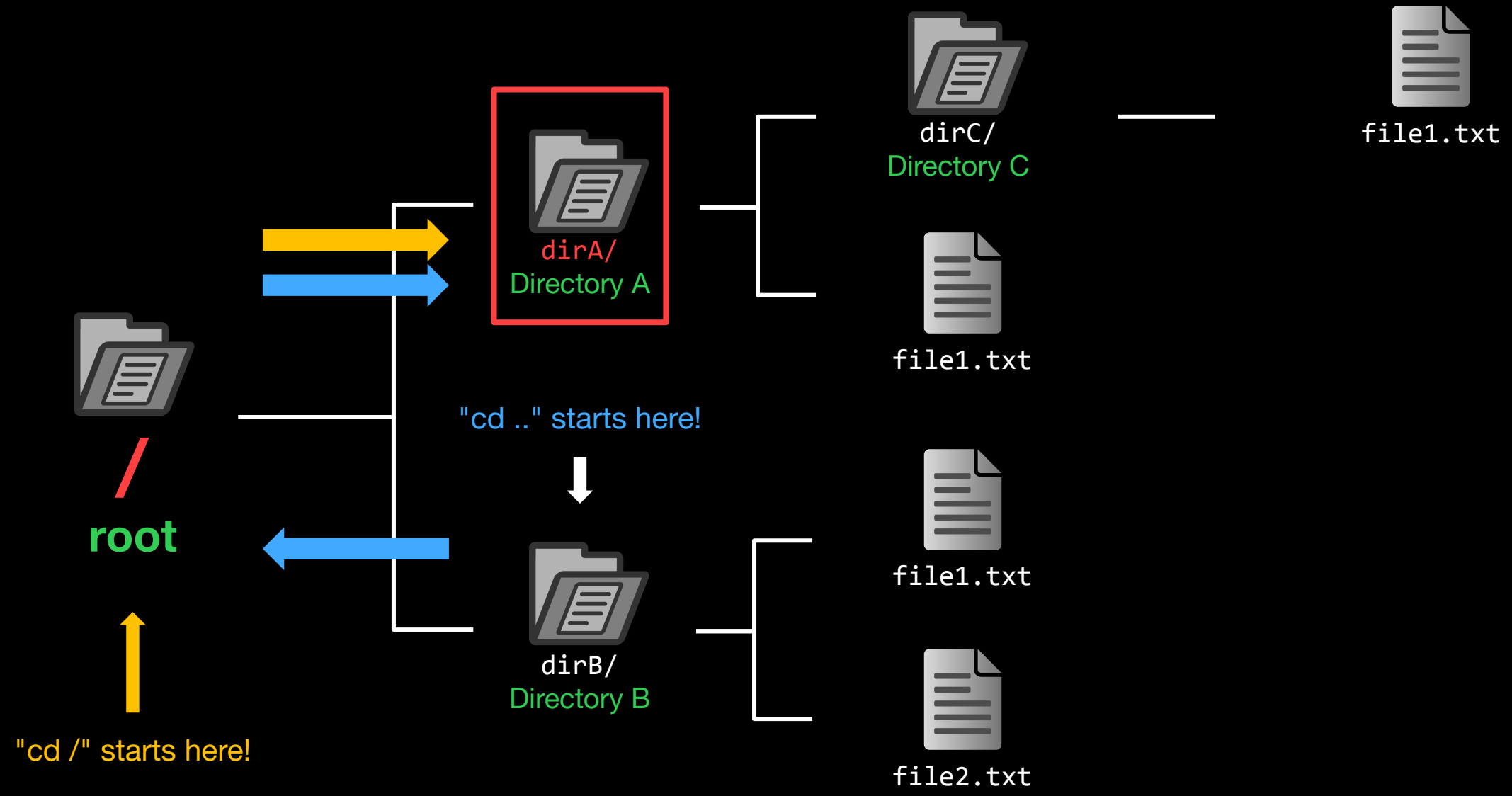
How to get to dirA?



How to get to dirA?



"cd /dirA" or "cd ../dirA"



Paths

Absolute Path

The full path that always starts at root (/)

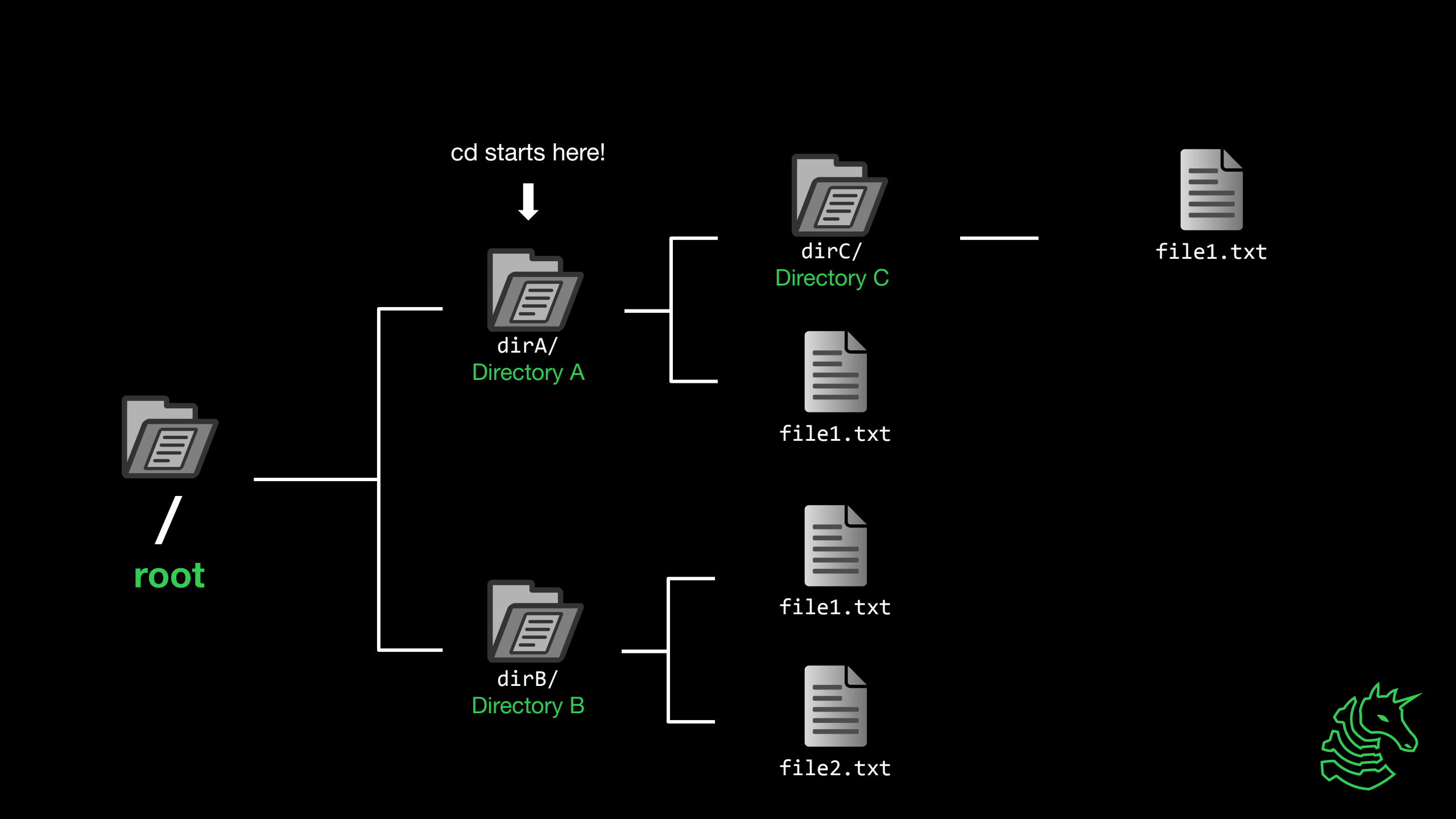
```
/dirA/file1.txt
/dirA/dirC/file1.txt
```

Relative Path

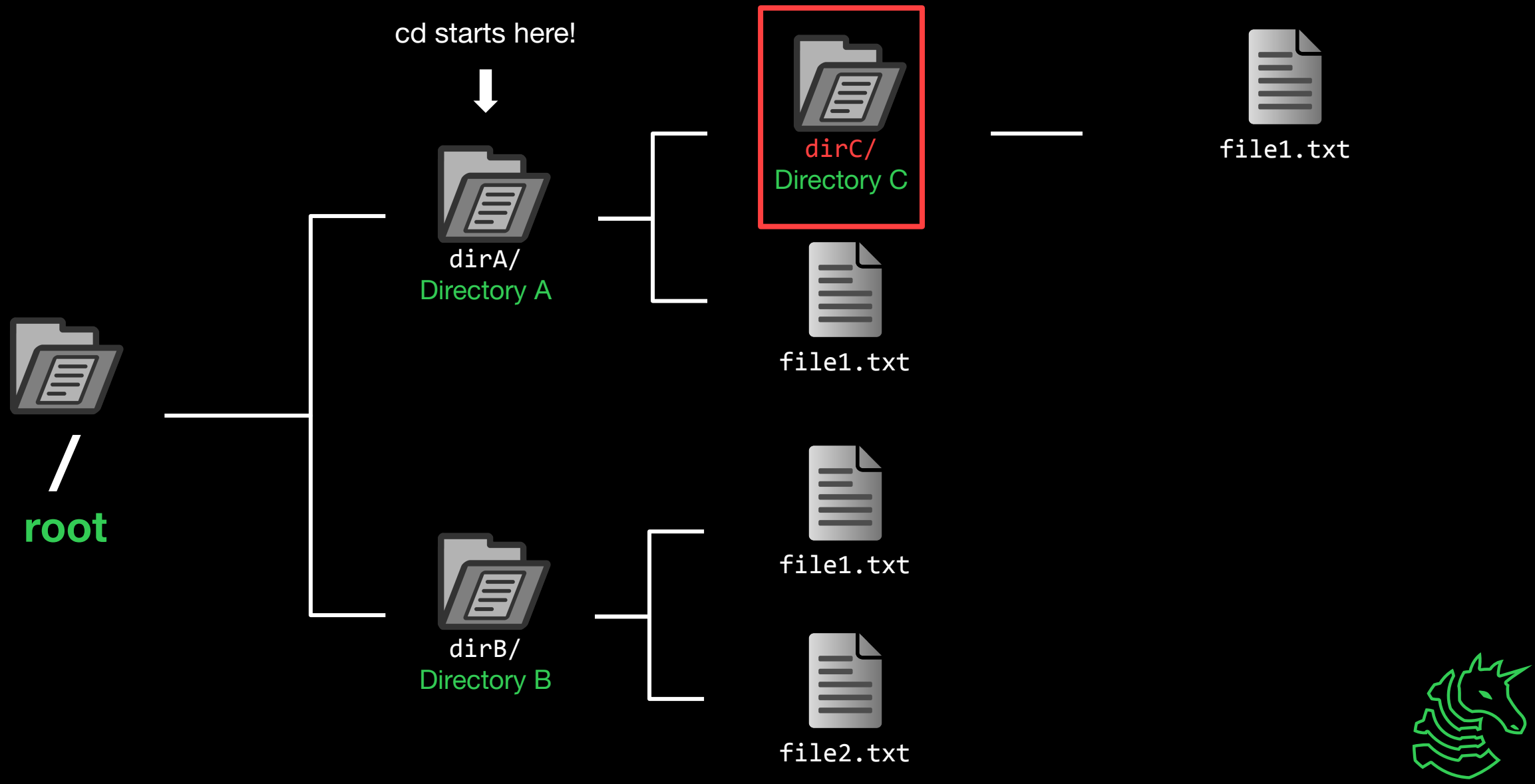
The partial path relative to where you are currently in the terminal

(Relative to dirA)

```
file1.txt
dirC/file1.txt
```


cd starts here!
/
root
dirA/
Directory A
dirB/
Directory B
dirC/
Directory C
file1.txt
file1.txt
file1.txt
file2.txt
file1.txt

"cd dirC" or "cd ./dirC" or "cd dirC/"



`./dirC == dirC == dirC/`

Also `././dirC` and `./././dirC` and `./././././dirC` and...

These are just conventions!

What the `.*?$`&>|~` is going on?

Every directory has special `.` and `..` files for the current directory and the parent directory

Piping (`|`), redirection (`><`), job control (`&`)… the shell is very powerful!

Wildcards (glob patterns) like `*` and `?` can match multiple files

`export ENV=VALUE` sets variables in your environment (add to `~/.bashrc` to make persistent)

Useful Commands - Filesystem

`ls [-la...] [directory]`: lists files in your current directory or specified directory

`cd <directory>`: changes your current directory to specified directory

`mv <sources> <dest>`: moves file(s) from source to dest (rename), if dest is a directory, move source

`rm [-r...] <sources>`: removes file(s) (**NOT REVERSIBLE**)

`cat <file>`: prints the contents of file (sometimes it prints gibberish: why might that happen?)

`./file`: executes whatever is at file (see also `$PATH`, How programs get run for a deep dive)

`man <command>`: lets you see info about a command and all of its parameters/options

`<parameter>` means it's a required parameter

`[parameter]` means it's an optional parameter

Useful Commands - Networking

`nc <ip> <port>`: Netcat, connect to ip (or hostname) on port port

`ssh <user@host> [-p port]`: Secure Shell, run a shell as user on host (SSH keys)

`ping <ip>`: see if an IP address is up using ICMP (sometimes blocked by firewalls)

`curl <url>`: versatile network access tool that is mainly used to access websites from the terminal

`wget <url>`: download the file at `url`

Networking Fundamentals

`nc <ip> <port>`: Netcat, connect to ip (or hostname) on port port
`nc -l <port>`: Open a network socket to listen on port

Ports: communication endpoints on your computer (1-65535)

- Ports numbers ≤ 1024 are reserved for other programs

Next Steps - Bandit

```
ssh bandit0@bandit.labs.overthewire.org -p 2220
```

Next Steps - Bandit

```
ssh bandit0@bandit.labs.overthewire.org -p 2220
```

command
user
IP
port

Next Steps - Terminal Challenges

- **netcat**
  - Refer back to the slides!
- **Shell Basics**
  - Learn the ins and outs of using the terminal
- **A Very Special Character**
  - Intro to the ASCII table and Netcat

More Resources

- **SIGPwny!**
  - Meeting archive
  - #ask-for-help in Discord
- The Missing Semester
- The Linux Command Line, GNU Coreutils manual
- Google

Questions/Issues?

- **We're here to help!**
  - Ask your friendly neighborhood helper
  - *#ask-for-help* in the SIGPwny Discord
- Consult Google/GitHub
  - Someone may have encountered and fixed your issue already
  - Writeups from past CTFs can be very informative (check CTFtime)

Next Meetings

2025-09-06 • **This Saturday 3-5 PM**

- Our first social!
- We'll meet on the main quad with chalk and board games!

2025-09-07 • **This Sunday**

- *Web Hacking I* with Cameron and 😁
- Learn introductory knowledge on web hacking

2025-09-21 • **Fall CTF 2025**

- Register at sigpwny.com/fallctf!

ctf.sigpwny.com

`sigpwny{lets_get_setup}`

Meeting content can be found at sigpwny.com/meetings.

SIGPwny