SIGPwny

FA2024 Week 01 • 2024-09-08

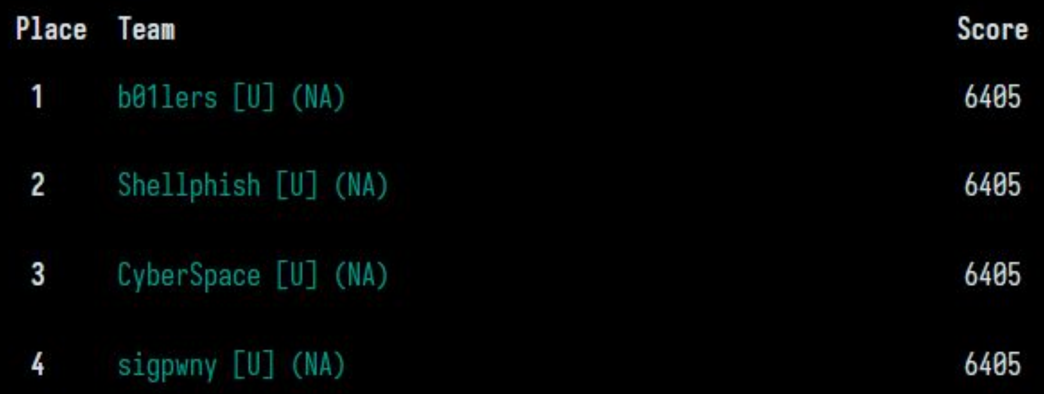
Intro to Terminal and Setup

George and Adarsh

Announcements

- Fall CTF registration is open!
  - sigpwny.com/register24
  - Beginner-friendly CTF on **Sunday**, September 22nd 12-6 PM!
  - Free t-shirts and badges are first come, first serve!

- We finished 4th place in CSAW CTF Quals!
  - We will be sending a team to New York for finals on November 6th-9th



Place	Team	Score
1	b01lers [U] (NA)	6405
2	Shellphish [U] (NA)	6405
3	CyberSpace [U] (NA)	6405
4	sigpwny [U] (NA)	6405

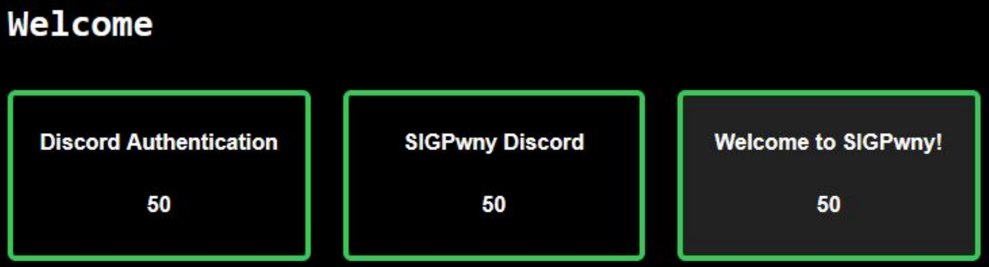
The "Don't Get Arrested" Slide

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!

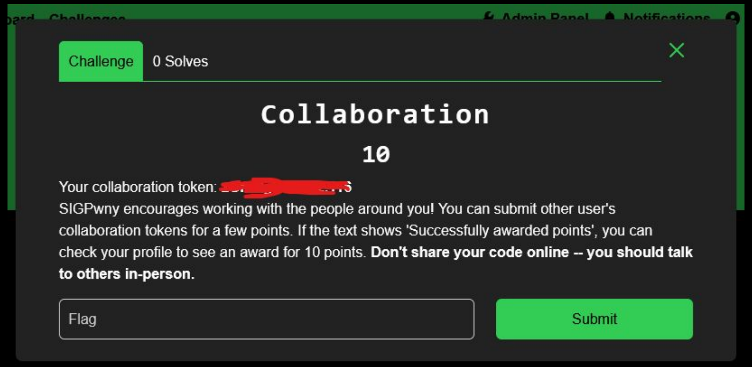


Welcome

Discord Authentication 50

SIGPwny Discord 50

Welcome to SIGPwny! 50



Collaboration

10

Your collaboration token: [redacted]

SIGPwny encourages working with the people around you! You can submit other user's collaboration tokens for a few points. If the text shows 'Successfully awarded points', you can check your profile to see an award for 10 points. **Don't share your code online -- you should talk to others in-person.**

ctf.sigpwny.com

sigpwny{starting_off_strong}

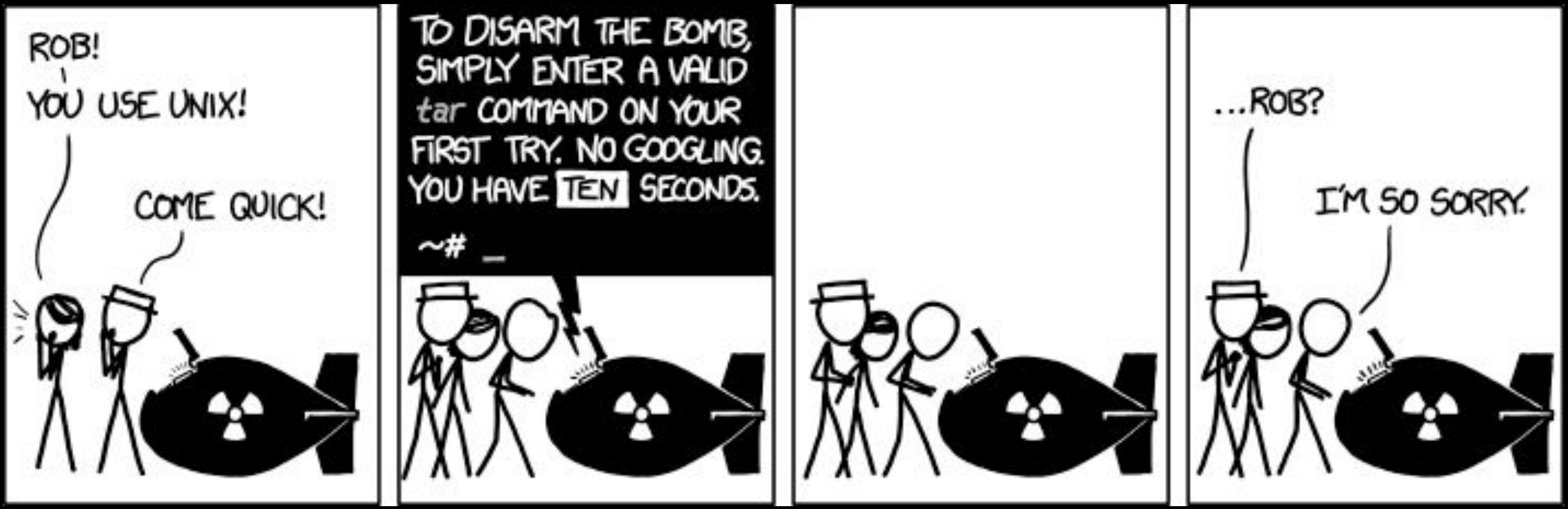
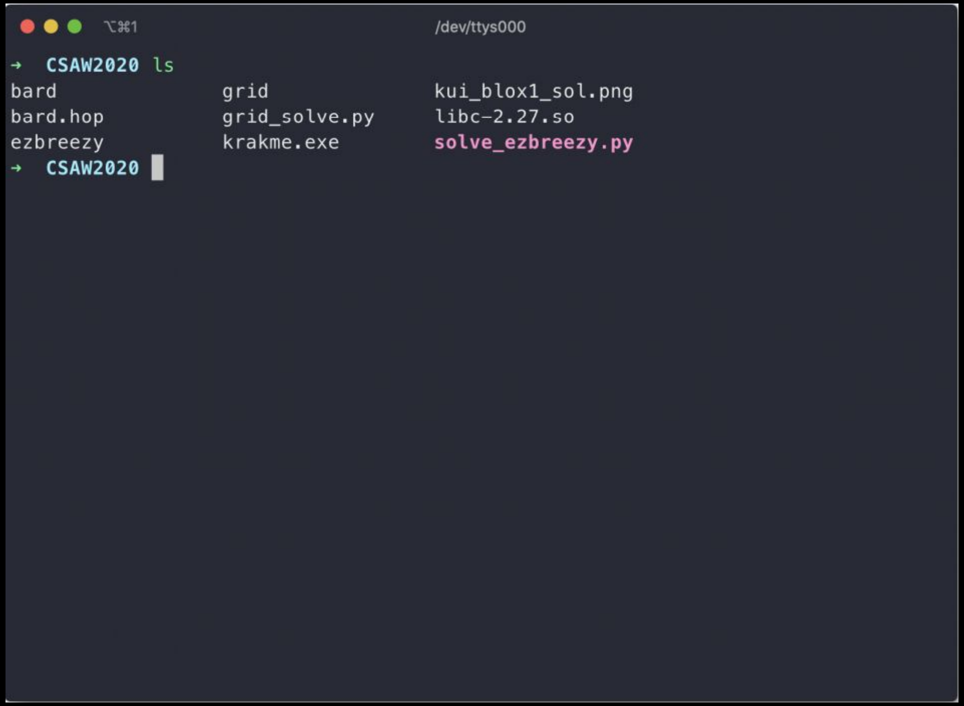

ROB!
YOU USE UNIX!
COME QUICK!
TO DISARM THE BOMB, SIMPLY ENTER A VALID tar COMMAND ON YOUR FIRST TRY. NO GOOGLING. YOU HAVE TEN SECONDS.
~# _
...ROB?
I'M SO SORRY.

Table of Contents

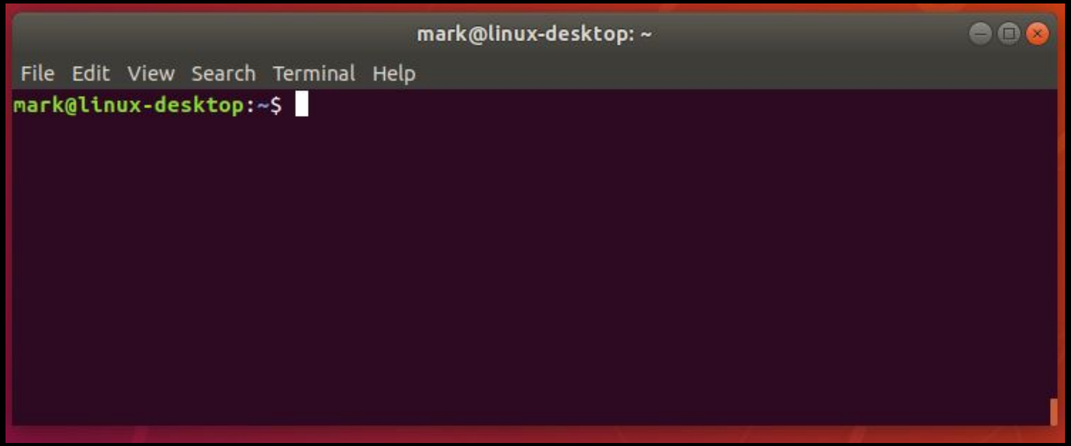
1. What is a shell
    - *I want one!*
    - Install a package manager, terminal emulator
    - WSL or a virtual machine?
2. Using the shell
    - Platform differences
    - Useful builtins and utilities
3. Setup for reverse-engineering (rev) and binary exploitation (pwn) meetings
    - Installing Ghidra,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

> The Terminal

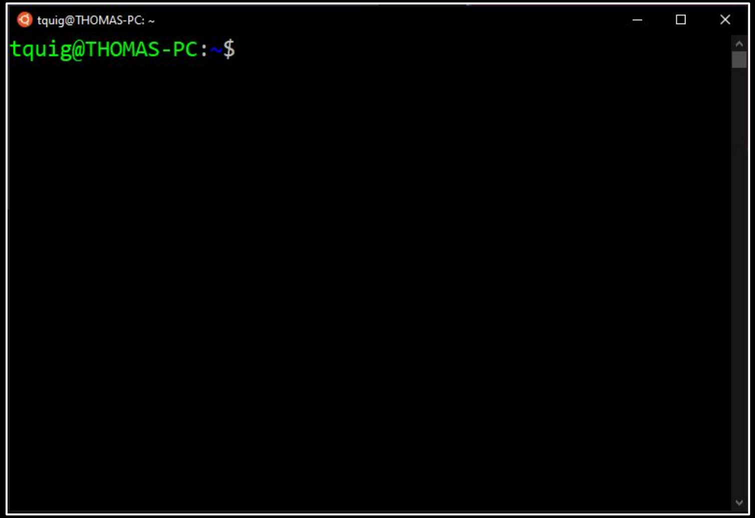
"It’s where things happen" - Ravi



```
➜  CSAW2020 ls
bard              grid              kui_blox1_sol.png
bard.hop          grid_solve.py     libc-2.27.so
ezbreezy          krakme.exe        solve_ezbreezy.py
➜  CSAW2020
```



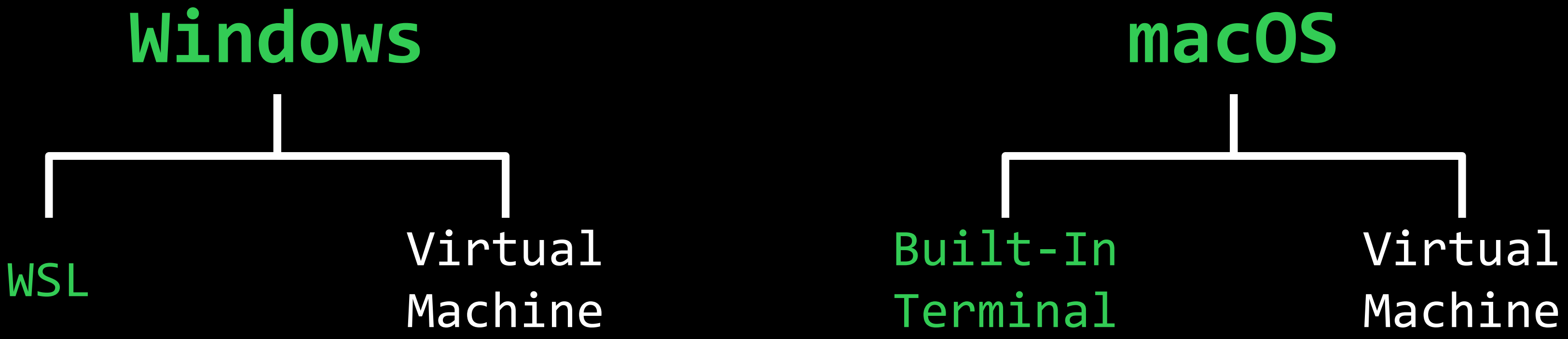
```
mark@linux-desktop:~$
```



```
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

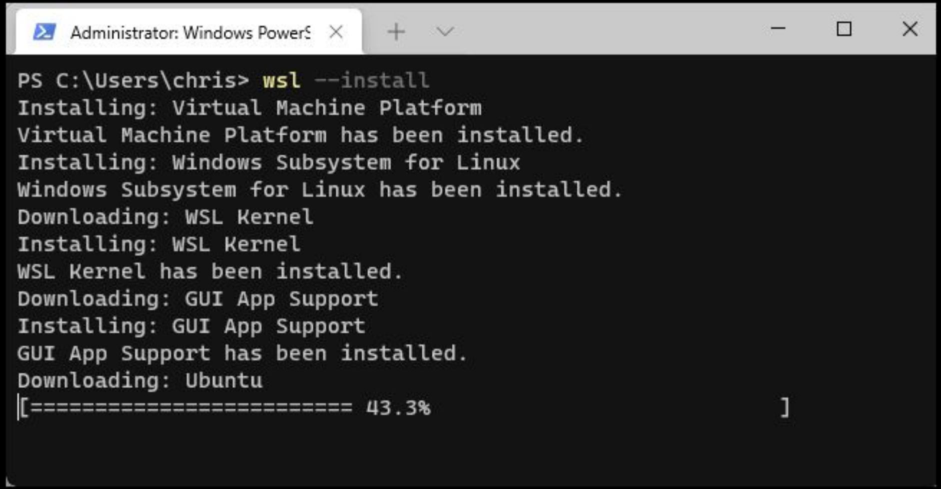
Windows Subsystem for Linux (WSL)

Mac users hold tight…

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

Installation

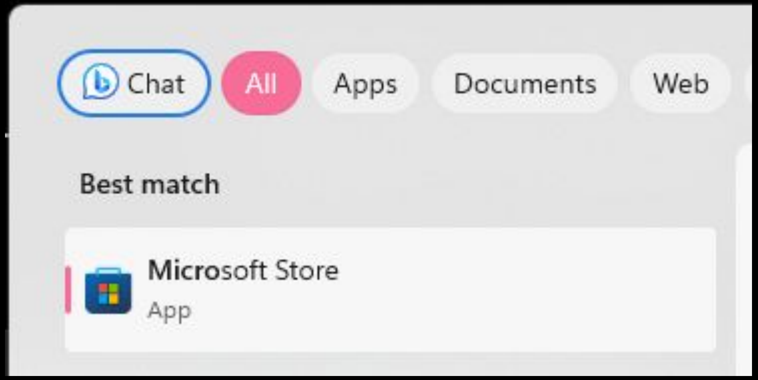
- Open command prompt as administrator
  - (Start button → type `cmd` → right click → "Run as Administrator")
- Type `wsl --install [-d <distro>]`
- Restart computer
- You should be able to launch Ubuntu from the start menu

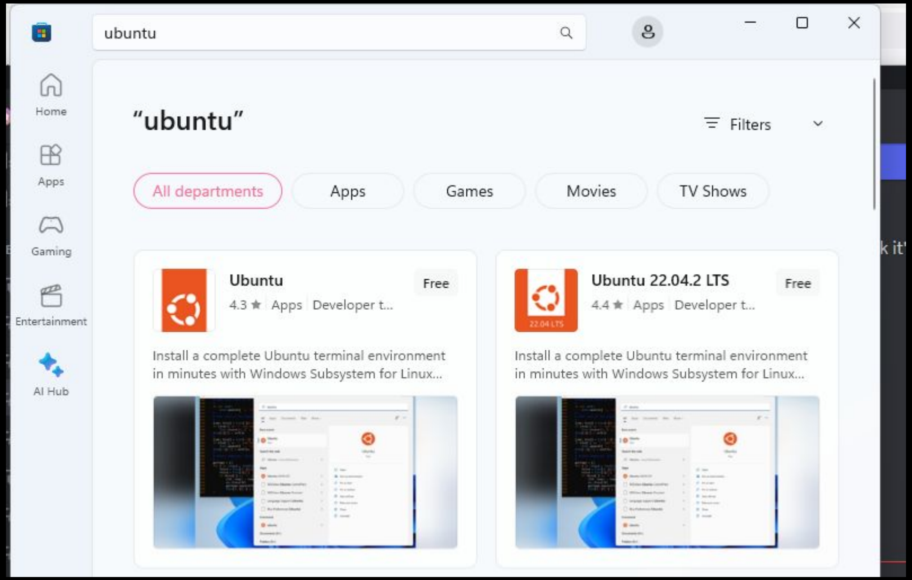


```
PS C:\Users\chris> wsl --install
Installing: Virtual Machine Platform
Virtual Machine Platform has been installed.
Installing: Windows Subsystem for Linux
Windows Subsystem for Linux has been installed.
Downloading: WSL Kernel
Installing: WSL Kernel
WSL Kernel has been installed.
Downloading: GUI App Support
Installing: GUI App Support
GUI App Support has been installed.
Downloading: Ubuntu
[========================= 43.3%                           ]
```

Installation (Windows Store)

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





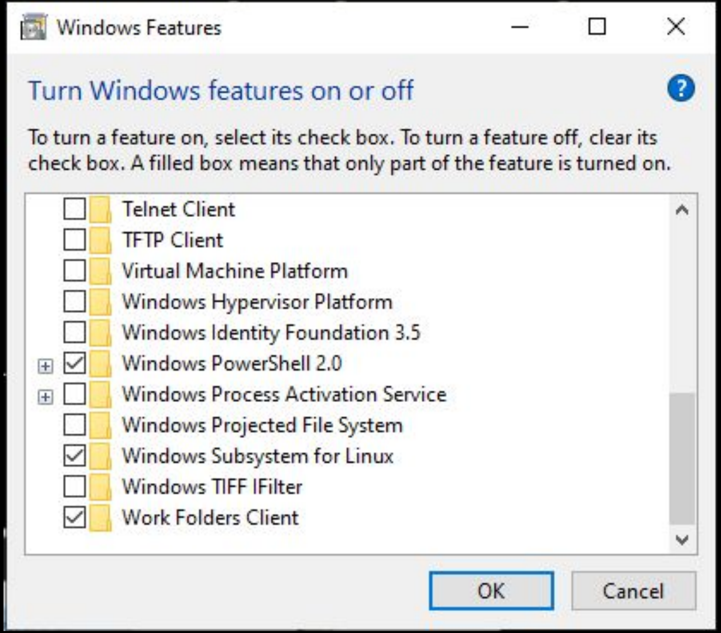
Install a complete Ubuntu terminal environment in minutes with Windows Subsystem for Linux...

Install a complete Ubuntu terminal environment in minutes with Windows Subsystem for Linux...

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



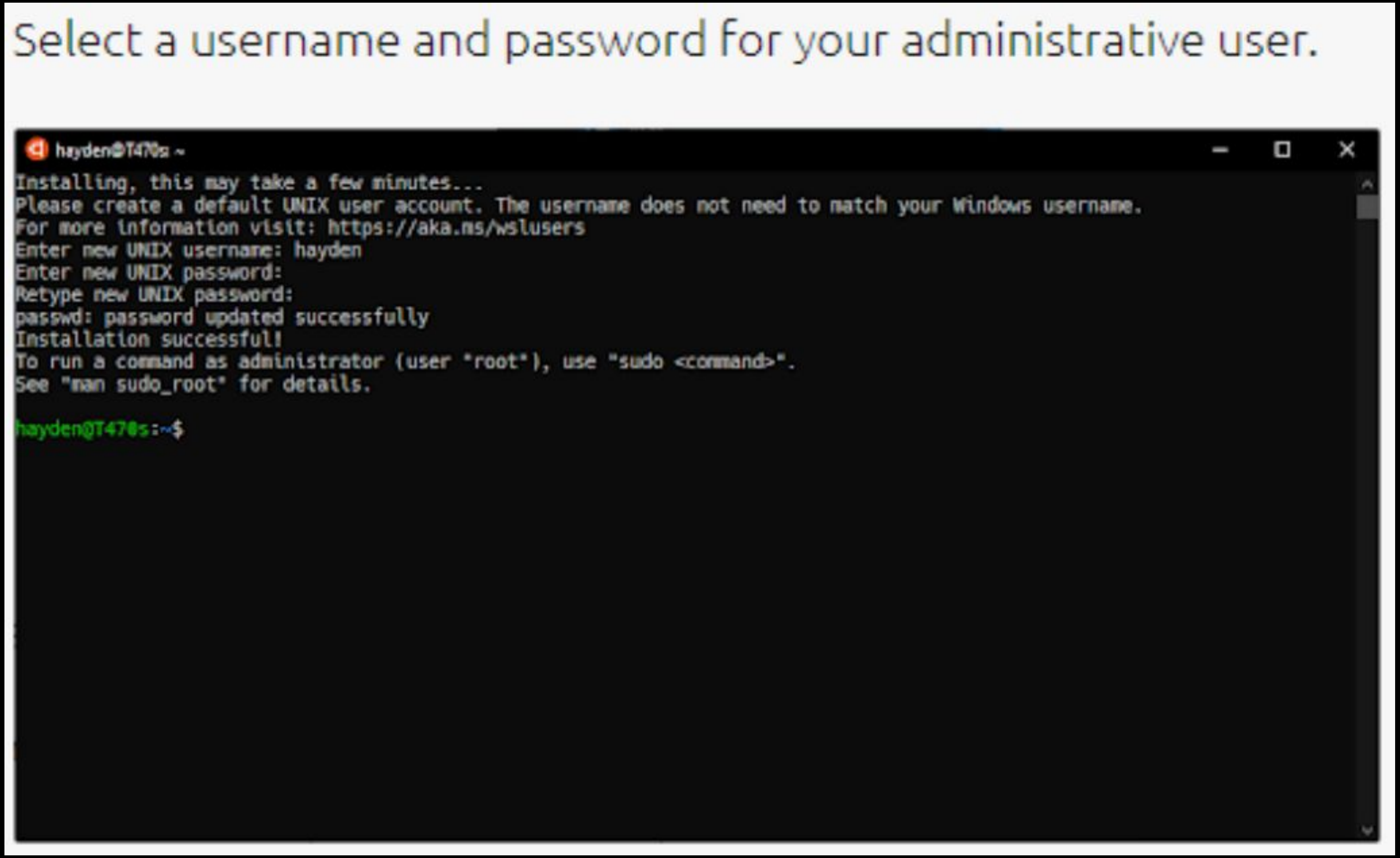
Windows Features

Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- ☐ Telnet Client
- ☐ TFTP Client
- ☐ Virtual Machine Platform
- ☐ Windows Hypervisor Platform
- ☐ Windows Identity Foundation 3.5
- ☑ Windows PowerShell 2.0
- ☐ Windows Process Activation Service
- ☐ Windows Projected File System
- ☑ Windows Subsystem for Linux
- ☐ Windows TIFF IFilter
- ☑ Work Folders Client

Set a "root" username and password!

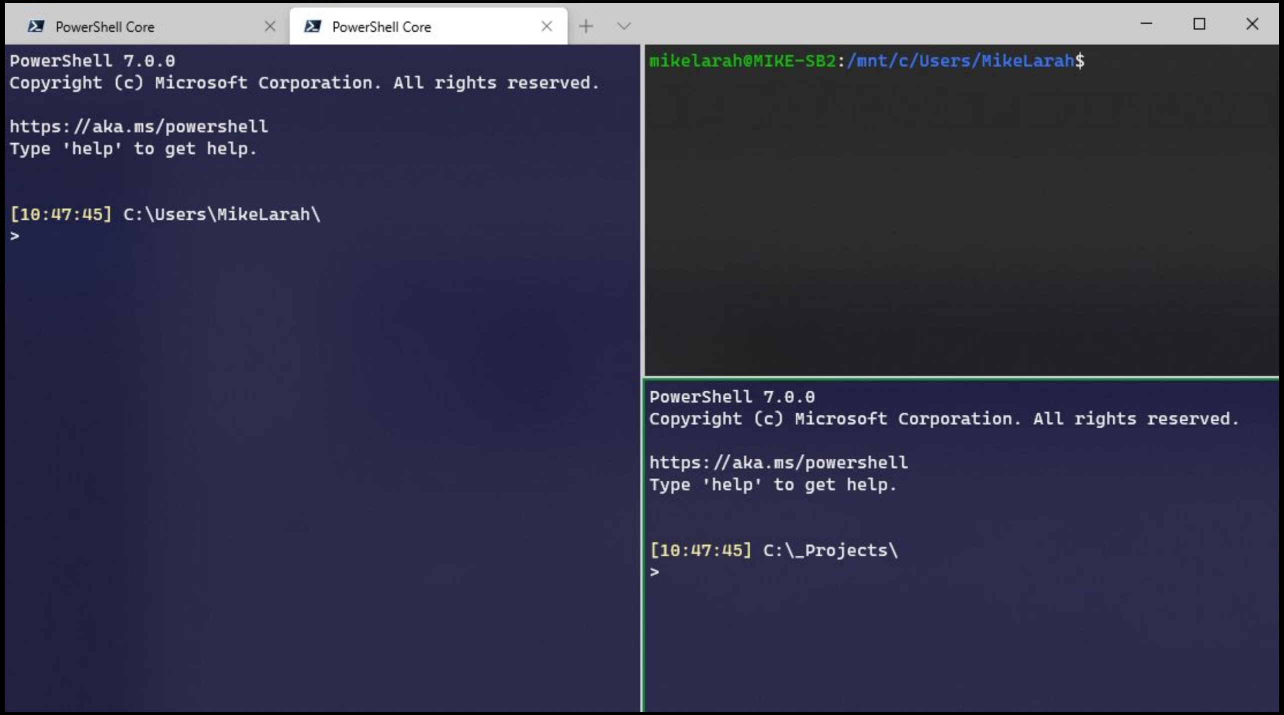


Select a username and password for your administrative user.

```
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: hayden
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

hayden@T470s:~$
```

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



```
PowerShell 7.0.0
Copyright (c) Microsoft Corporation. All rights reserved.

https://aka.ms/powershell
Type 'help' to get help.


[10:47:45] C:\Users\MikeLarah\
>
```

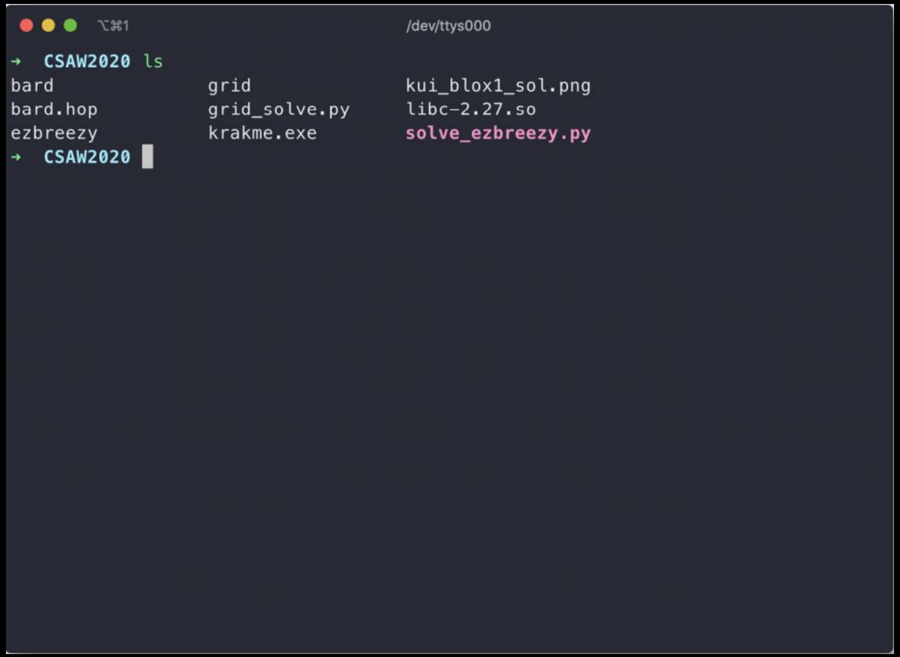
```
mikelarah@MIKE-SB2:/mnt/c/Users/MikeLarah$
```

```
PowerShell 7.0.0
Copyright (c) Microsoft Corporation. All rights reserved.

https://aka.ms/powershell
Type 'help' to get help.


[10:47:45] C:\_Projects\
>
```

macOS Terminal

Command
+ Space

Search “Terminal”



```
➜  CSAW2020 ls
bard                grid                kui_blox1_sol.png
bard.hop            grid_solve.py       libc-2.27.so
ezbreezy            krakme.exe          solve_ezbreezy.py
➜  CSAW2020 
```

Homebrew

Homebrew

The missing package manager for macOS (or Linux)

- 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…

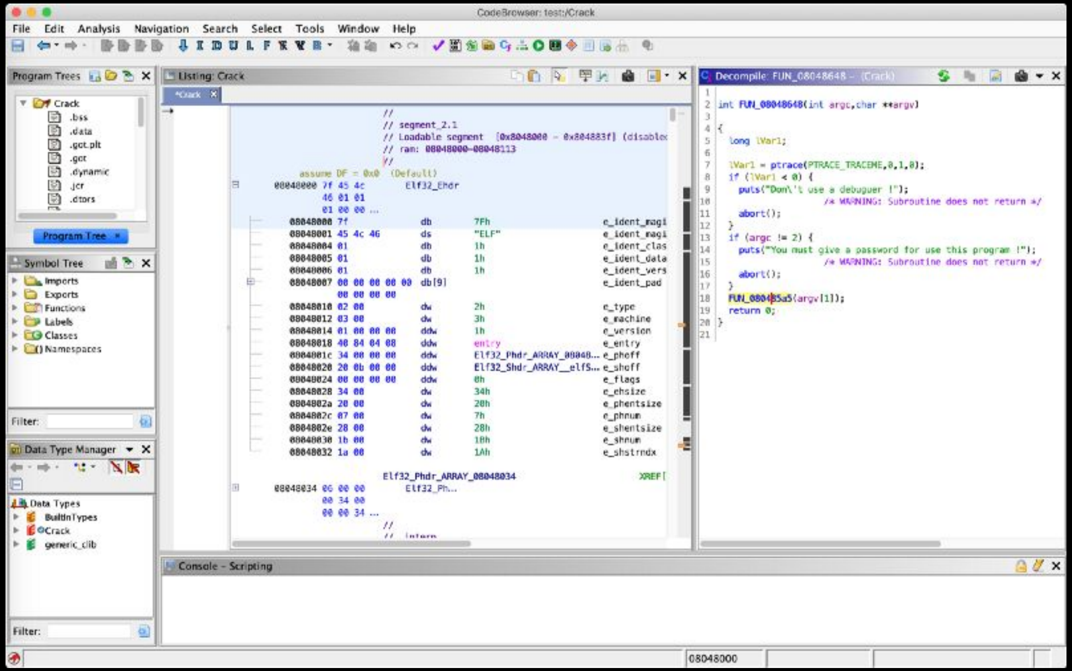
Reverse Engineering Setup

Ghidra

Decompiler go brrr

What is Ghidra?

- Ghidra is a reverse engineering toolkit developed by the NSA and made open source
- Allows you to disassemble applications - essentially turn an unreadable application into readable code

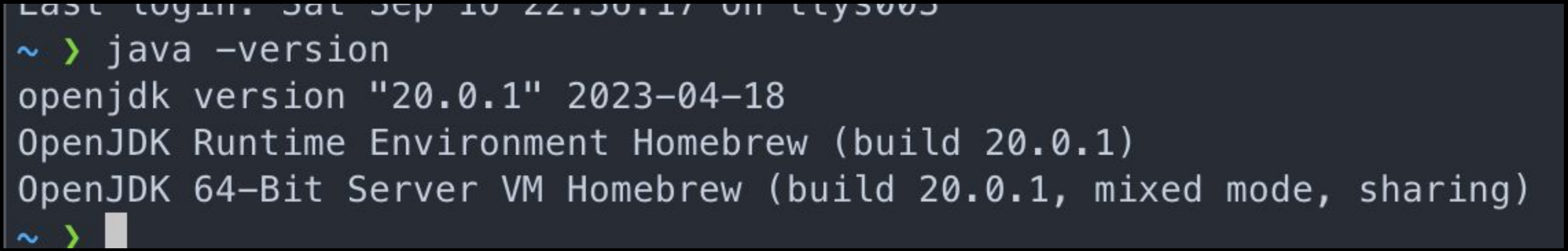


```
                           //
                           // segment_2.1
                           // Loadable segment  [0x8048000 - 0x804883f] (disable
                           // ram: 08048000-08048113
                           //
              assume DF = 0x0  (Default)
         08048000 7f 45 4c        Elf32_Ehdr
                  46 01 01
                  01 00 00 ...
           08048000 7f              db        7Fh                  e_ident_magi
           08048001 45 4c 46        ds        "ELF"                e_ident_magi
           08048004 01              db        1h                   e_ident_clas
           08048005 01              db        1h                   e_ident_data
           08048006 01              db        1h                   e_ident_vers
           08048007 00 00 00 00 00  db[9]                          e_ident_pad
                    00 00 00 00
           08048010 02 00           dw        2h                   e_type
           08048012 03 00           dw        3h                   e_machine
           08048014 01 00 00 00     ddw       1h                   e_version
           08048018 40 84 04 08     ddw       entry                e_entry
           0804801c 34 00 00 00     ddw       Elf32_Phdr_ARRAY_08048... e_phoff
           08048020 20 0b 00 00     ddw       Elf32_Shdr_ARRAY__elfS... e_shoff
           08048024 00 00 00 00     ddw       0h                   e_flags
           08048028 34 00           dw        34h                  e_ehsize
           0804802a 20 00           dw        20h                  e_phentsize
           0804802c 07 00           dw        7h                   e_phnum
           0804802e 28 00           dw        28h                  e_shentsize
           08048030 1b 00           dw        1Bh                  e_shnum
           08048032 1a 00           dw        1Ah                  e_shstrndx

                           Elf32_Phdr_ARRAY_08048034                    XREF[
         08048034 06 00 00        Elf32_Ph...
                  00 34 00
                  00 00 34 ...
                           //
                           // .intern
```

```c

int FUN_08048648(int argc,char **argv)

{
  long lVar1;

  lVar1 = ptrace(PTRACE_TRACEME,0,1,0);
  if (lVar1 < 0) {
    puts("Don\'t use a debuguer !");
                    /* WARNING: Subroutine does not return */
    abort();
  }
  if (argc != 2) {
    puts("You must give a password for use this program !");
                    /* WARNING: Subroutine does not return */
    abort();
  }
  FUN_080485a5(argv[1]);
  return 0;
}

```

Installing Java (Windows/macOS)

Check if you have Java, and if so what version; should be ≥17



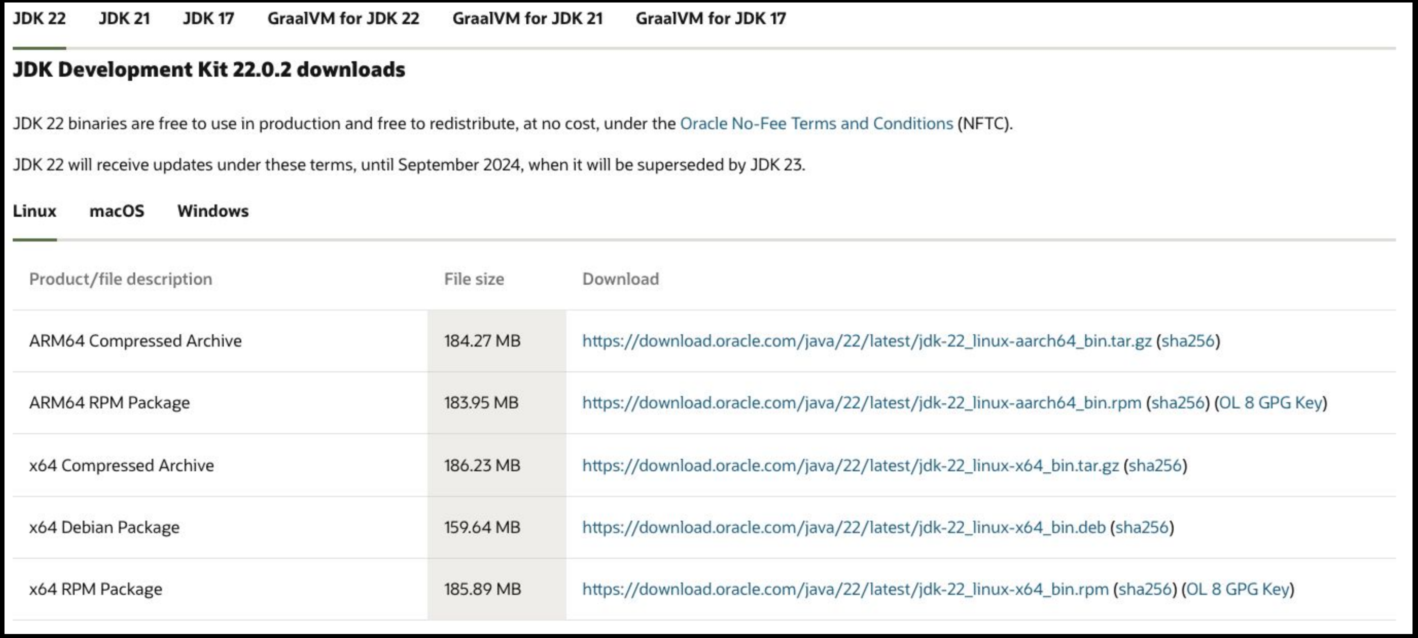
```
Last login: Sat Sep 16 22:30:17 on ttys005
~ ❯ java -version
openjdk version "20.0.1" 2023-04-18
OpenJDK Runtime Environment Homebrew (build 20.0.1)
OpenJDK 64-Bit Server VM Homebrew (build 20.0.1, mixed mode, sharing)
~ ❯ 
```

Note: we recommend installing JDK and Ghidra on Windows, not WSL

Installing Java (Windows/macOS)

Install JDK 17+ (not JRE!) from Oracle (or package manager, if applicable)

https://www.oracle.com/java/technologies/downloads/#java22



JDK Development Kit 22.0.2 downloads

JDK 22 binaries are free to use in production and free to redistribute, at no cost, under the Oracle No-Fee Terms and Conditions (NFTC).

JDK 22 will receive updates under these terms, until September 2024, when it will be superseded by JDK 23.

Product/file description	File size	Download
ARM64 Compressed Archive	184.27 MB	https://download.oracle.com/java/22/latest/jdk-22_linux-aarch64_bin.tar.gz (sha256)
ARM64 RPM Package	183.95 MB	https://download.oracle.com/java/22/latest/jdk-22_linux-aarch64_bin.rpm (sha256) (OL 8 GPG Key)
x64 Compressed Archive	186.23 MB	https://download.oracle.com/java/22/latest/jdk-22_linux-x64_bin.tar.gz (sha256)
x64 Debian Package	159.64 MB	https://download.oracle.com/java/22/latest/jdk-22_linux-x64_bin.deb (sha256)
x64 RPM Package	185.89 MB	https://download.oracle.com/java/22/latest/jdk-22_linux-x64_bin.rpm (sha256) (OL 8 GPG Key)

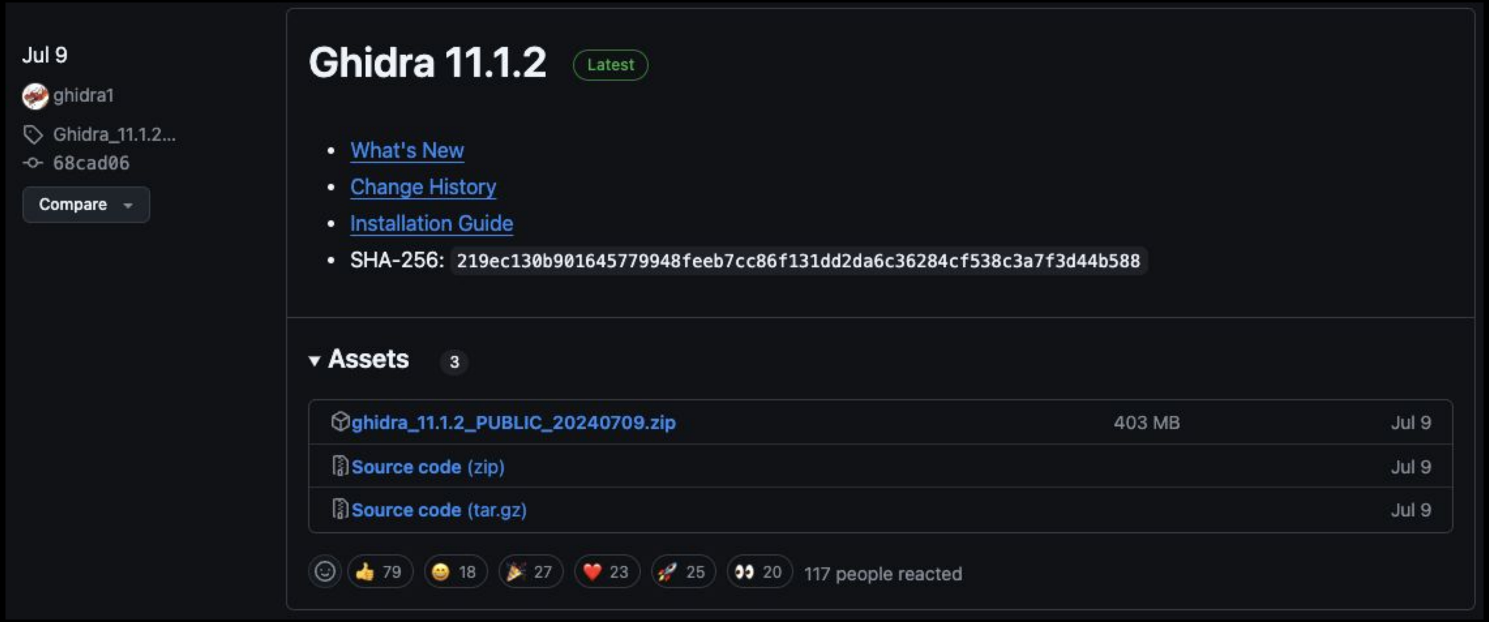
Installing JDK (Linux)

```
sudo apt update
sudo apt install openjdk-17-jdk
```

Downloading Ghidra (All Platforms)

https://github.com/NationalSecurityAgency/ghidra/releases

Download the public archive in assets for the latest release (`ghidra_X.X.X_PUBLIC_XXXXXXXX.zip`, not `Source code.zip`)



Jul 9

ghidra1

Ghidra_11.1.2...

68cad06

Ghidra 11.1.2 Latest

- What's New
- Change History
- Installation Guide
- SHA-256: `219ec130b901645779948feeb7cc86f131dd2da6c36284cf538c3a7f3d44b588`

▾ Assets 3

ghidra_11.1.2_PUBLIC_20240709.zip	403 MB	Jul 9
Source code (zip)		Jul 9
Source code (tar.gz)		Jul 9

117 people reacted

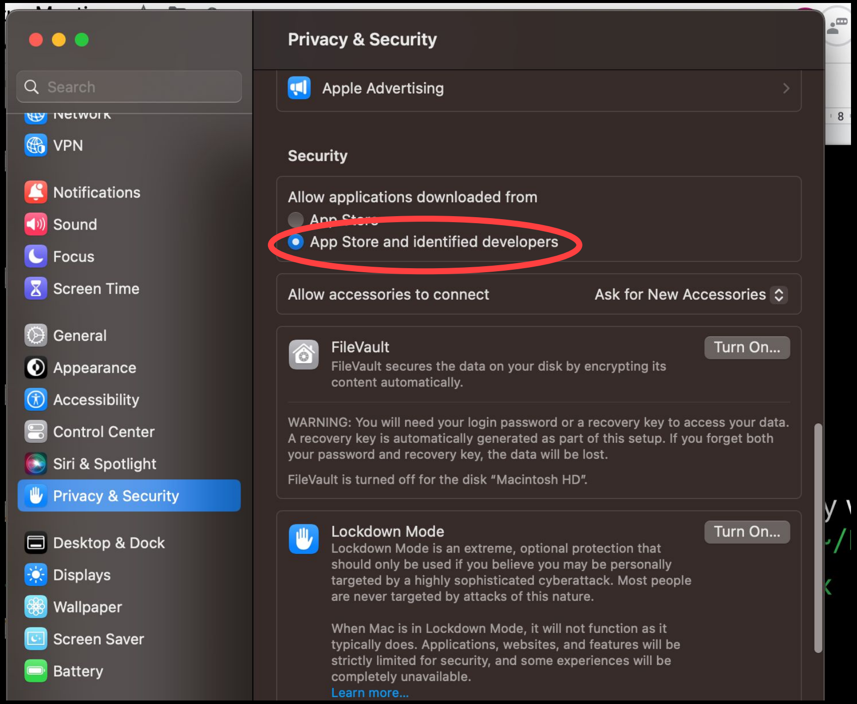
Running Ghidra

Windows:

Double click `ghidraRun.bat`

Mac/Linux:

```
$ cd ~/Downloads
$ unzip ghidra_??.?.?_PUBLIC_*.zip && cd
ghidra_??.?.?_PUBLIC
$ chmod +x ghidraRun && ./ghidraRun
```



Security

Allow applications downloaded from

○ App Store

● App Store and identified developers

Allow accessories to connect: Ask for New Accessories

FileVault

FileVault secures the data on your disk by encrypting its content automatically.

WARNING: You will need your login password or a recovery key to access your data. A recovery key is automatically generated as part of this setup. If you forget both your password and recovery key, the data will be lost.

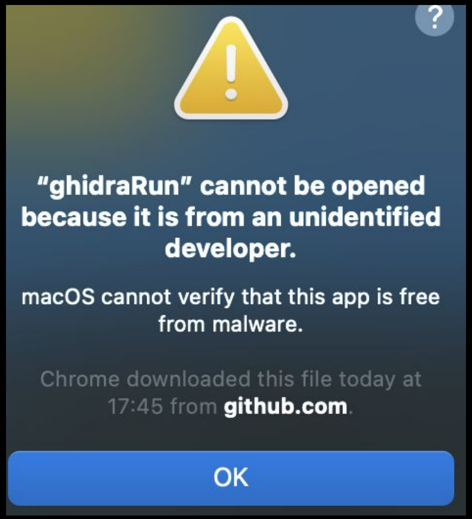
FileVault is turned off for the disk "Macintosh HD".

Lockdown Mode

Lockdown Mode is an extreme, optional protection that should only be used if you believe you may be personally targeted by a highly sophisticated cyberattack. Most people are never targeted by attacks of this nature.

When Mac is in Lockdown Mode, it will not function as it typically does. Applications, websites, and features will be strictly limited for security, and some experiences will be completely unavailable.

Running Ghidra (macOS)



“ghidraRun” cannot be opened because it is from an unidentified developer.

macOS cannot verify that this app is free from malware.

Chrome downloaded this file today at 17:45 from **github.com**.

The Ghidra distributable on GitHub is unsigned and needs permission to run the decompiler binaries

1. Open an x86 binary and run through the default decompiler
2. When you receive an error, go back to the "Privacy & Security" tab of settings, and hit "allow" on the binary that appears there
3. Repeat until you receive no errors when decompiling

OR run this one-liner to remove Ghidra from "quarantine":

```
sudo xattr -d -r com.apple.quarantine $GHIDRA_ROOT
```

`$GHIDRA_ROOT - where you downloaded ghidra to`

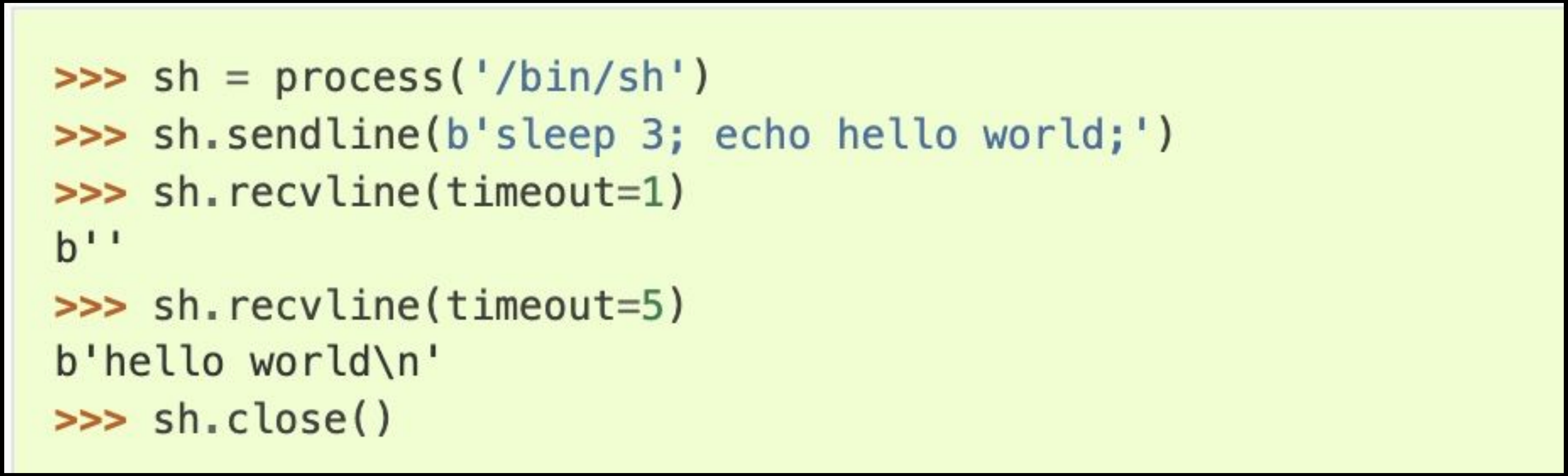
Python and pwntools

"Now is better than never." (*The Zen of Python*, aphorism 15)

What is pwntools?

pwntools is a CTF framework and exploit development library written in Python

It makes scripting exploits much simpler/less tedious



```
>>> sh = process('/bin/sh')
>>> sh.sendline(b'sleep 3; echo hello world;')
>>> sh.recvline(timeout=1)
b''
>>> sh.recvline(timeout=5)
b'hello world\n'
>>> sh.close()
```

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

```
$ 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
```

Installing pwntools

```
python3 -m pip install pwntools
```

If you get a "command not found", you may need to refresh the shell environment:

```
source ~/.bashrc
source ~/.zshrc # zsh is default on macOS
```

on Apple silicon (M1, etc.) run this first!

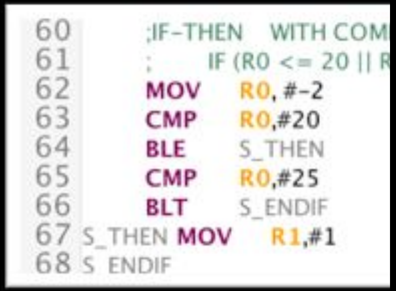
```
$ brew install cmake pkg-config qemu
```

GDB + pwndbg

For those times where `printf` doesn't cut it

Computer Architectures

M1 Macbook

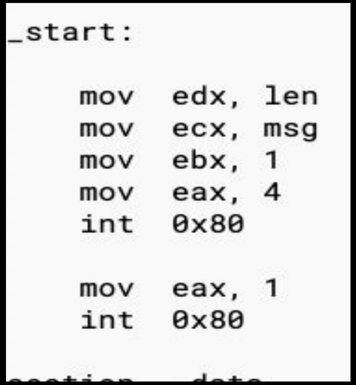


```
        ;IF-THEN   WITH COM
        ;     IF (R0 <= 20 || R
        MOV     R0, #-2
        CMP     R0,#20
        BLE     S_THEN
        CMP     R0,#25
        BLT     S_ENDIF
S_THEN  MOV     R1,#1
S_ENDIF
```

aarch64 / arm64
“arm, 64 bit”

You cannot run x86 programs normally* on arm64, or vice versa!

i9-morbillion
laptop



```
_start:

    mov   edx, len
    mov   ecx, msg
    mov   ebx, 1
    mov   eax, 4
    int   0x80

    mov   eax, 1
    int   0x80

section   .data
```

x86 / x86_64
“x86, 64 bit”

*We will talk about an exception on Macs called Rosetta

**Otherwise, you can use QEMU

What is GDB?

- The **G**NU **D**e**B**ugger allows you to inspect and modify execution of programs
- We will teach you how to debug **x86** binaries in **Rev II: x86 Reversing**!
- **pwndbg** is a "plugin" (`gdbinit`) for GDB that adds lots of nice features that are useful for binary exploitation and reverse-engineering

Installing GDB + pwndbg

macOS:

- GDB cannot debug **native** programs on Apple silicon (`aarch64-darwin`), *but can still debug binaries for other platforms (including x86)*
- Use our Docker container!

WSL/Linux:

```
$ sudo apt install gdb
$ git clone https://github.com/pwndbg/pwndbg && cd
pwndbg && ./setup.sh
```

pwn-docker

For debugging and running x86 applications on **arm64 macs**

- if you have e.g. a windows arm machine, talk to us after the meeting

Installation

You must be running macOS 13 or newer!

Enable Rosetta:

```
$ /usr/sbin/softwareupdate --install-rosetta --agree-to-license
```

Download the latest Docker Desktop and:

- Enable '**Use Virtualization Framework**' in 'Settings > General'
- Enable '**Use Rosetta for x86/amd64 on Apple Silicon**' in 'Settings > Features in Development'

Clone `pwn-docker`:

```
git clone https://github.com/sigpwny/pwn-docker.git
```

`pwn-docker` Usage

`./create.sh` - Run this to start your container. Type 'y' to initialize a permanent container, or 'n' for a temporary container. Don’t start in background – still WIP.

`./connect.sh` - Connect to your permanent container after it has been stopped

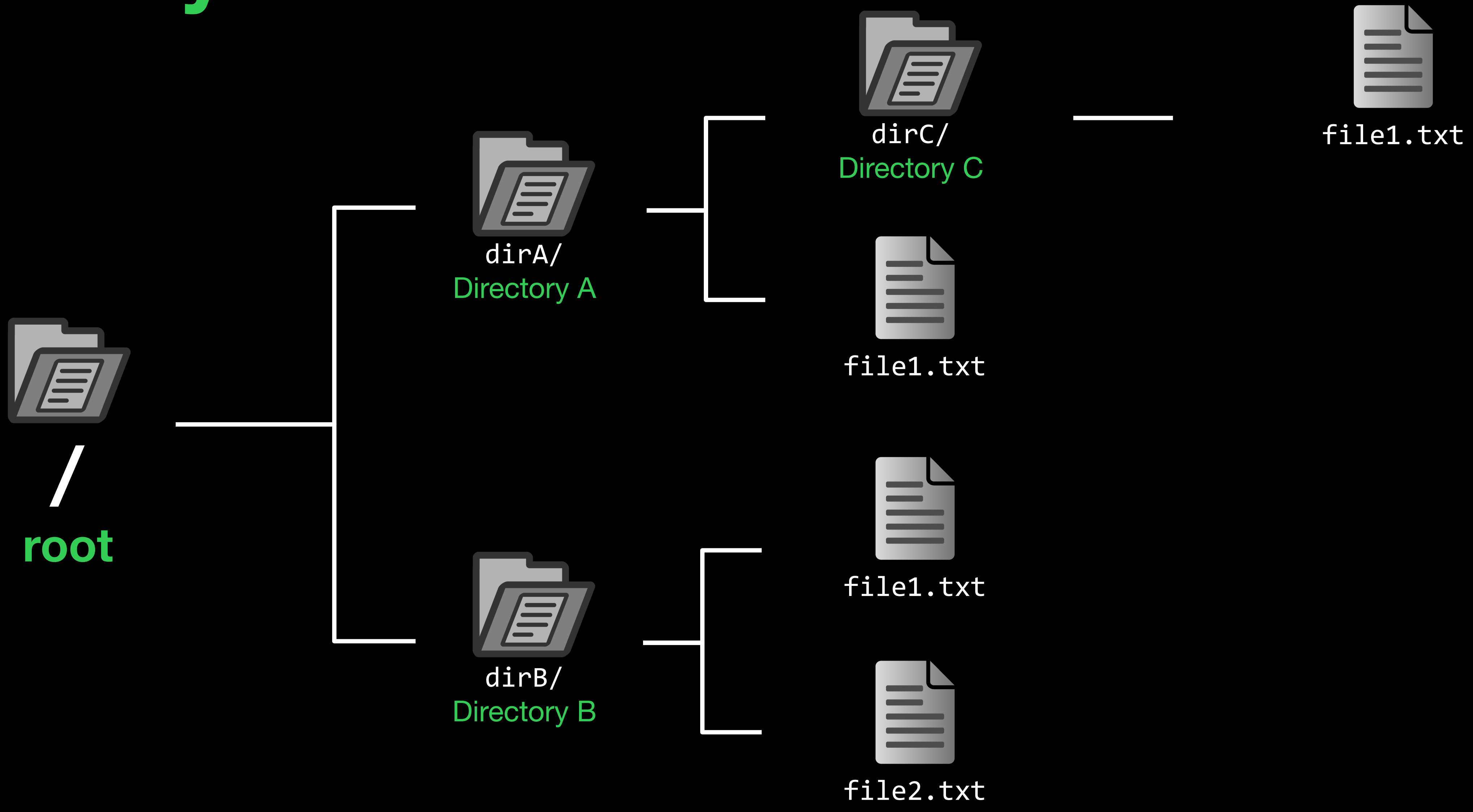
GDB *should* work, ask in Discord if you run into a problem

```
$ file ./challenge
challenge: ELF 64-bit LSB pie executable, x86-64, ...
$ ROSETTA_DEBUGSERVER_PORT=1234 ./challenge
$ gdb ./challenge -ex 'target remote localhost:1234'
```

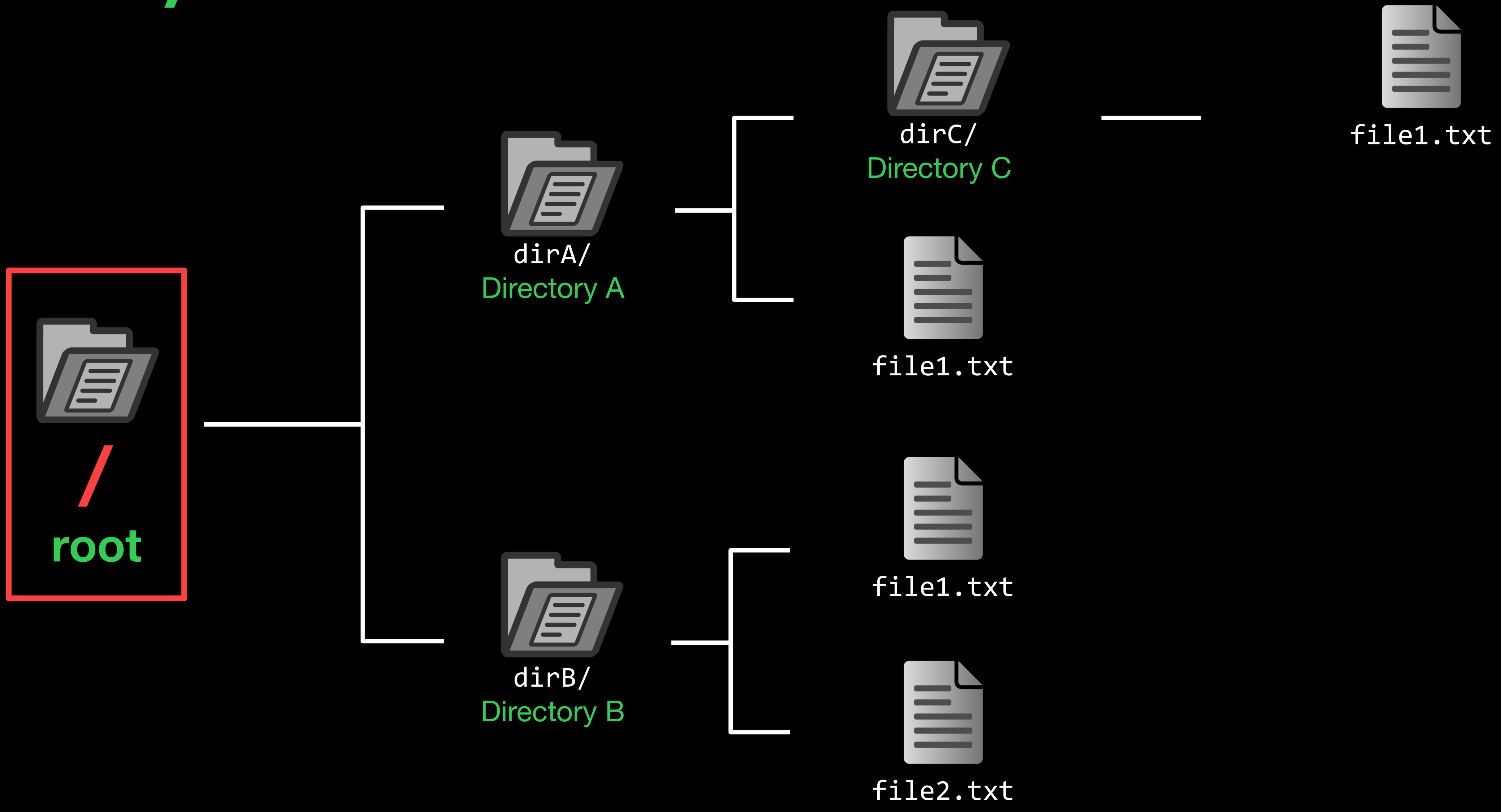
Unix Crash Course

Navigate a file system!

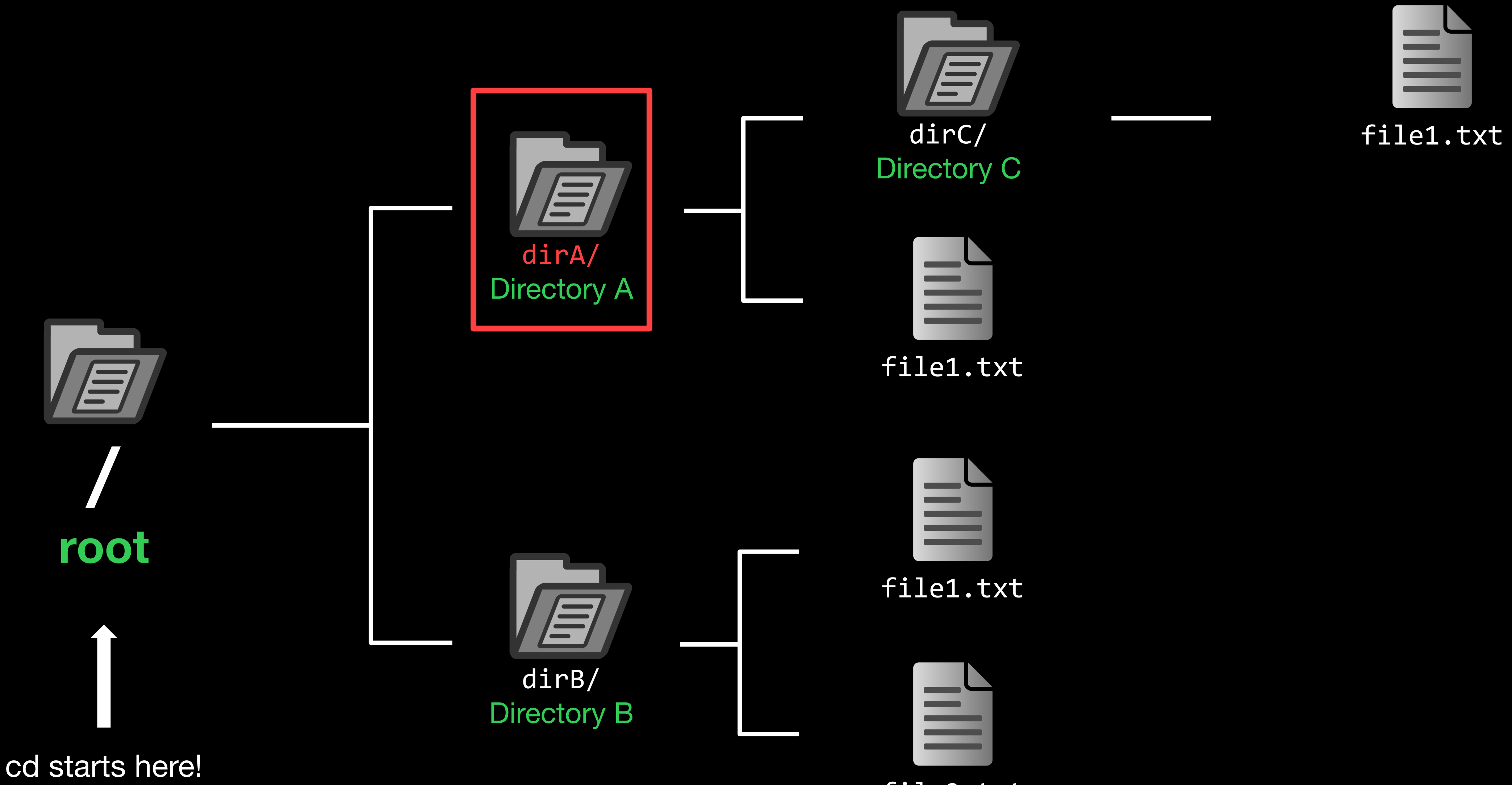
Filesystems



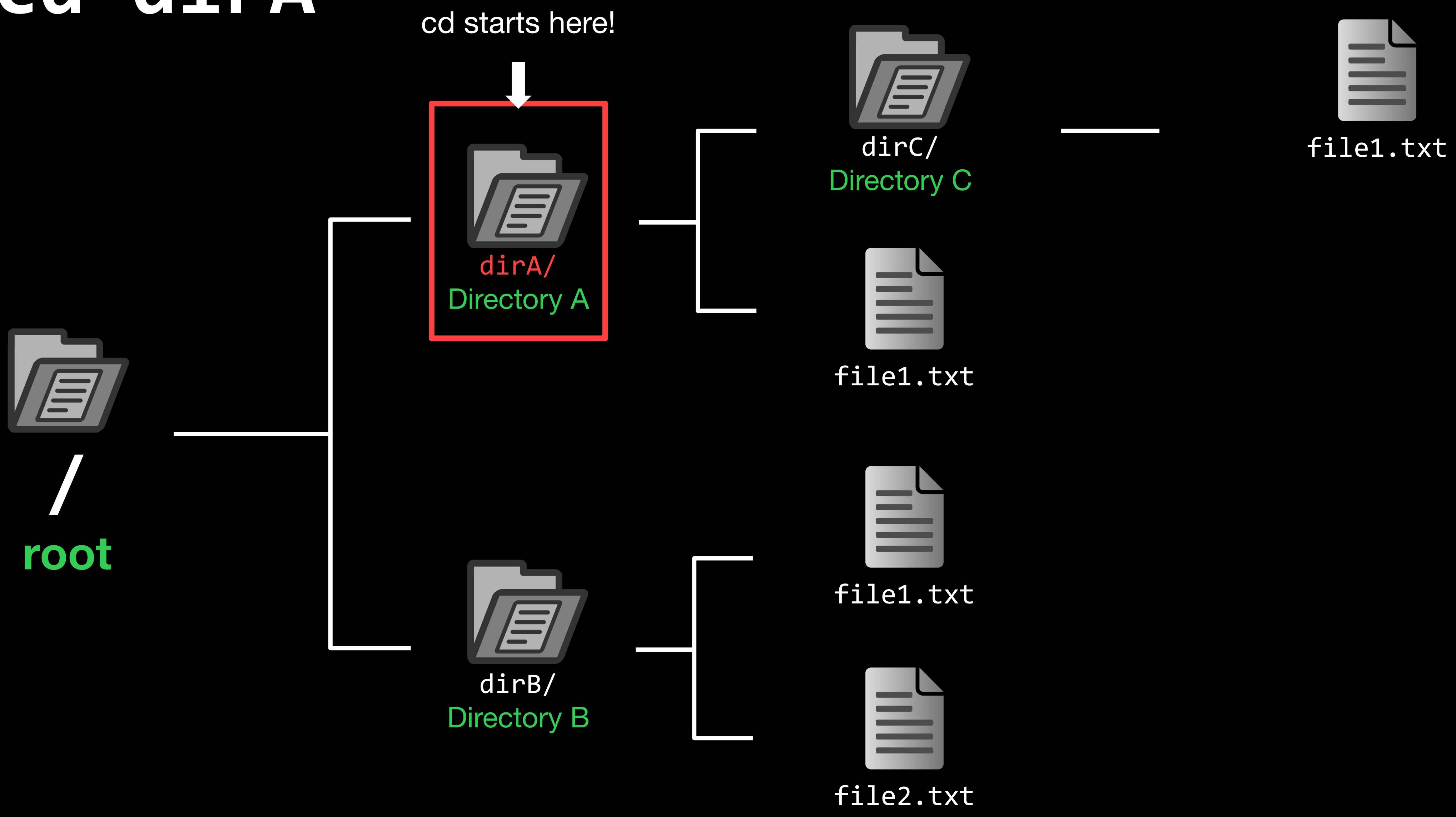
```
cd /
```

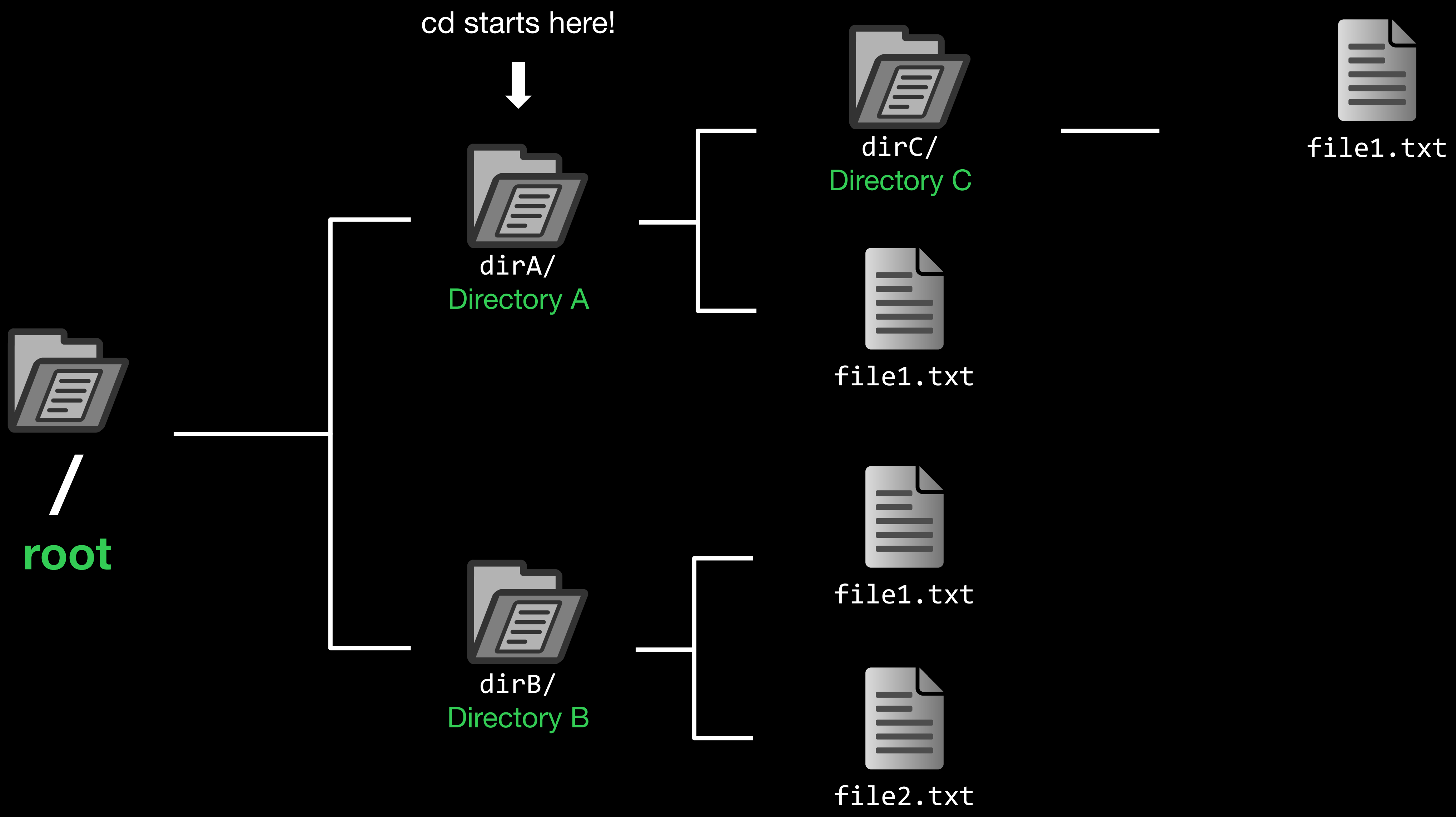

/
root
dirA/
Directory A
dirB/
Directory B
dirC/
Directory C
file1.txt
file1.txt
file1.txt
file2.txt

cd dirA

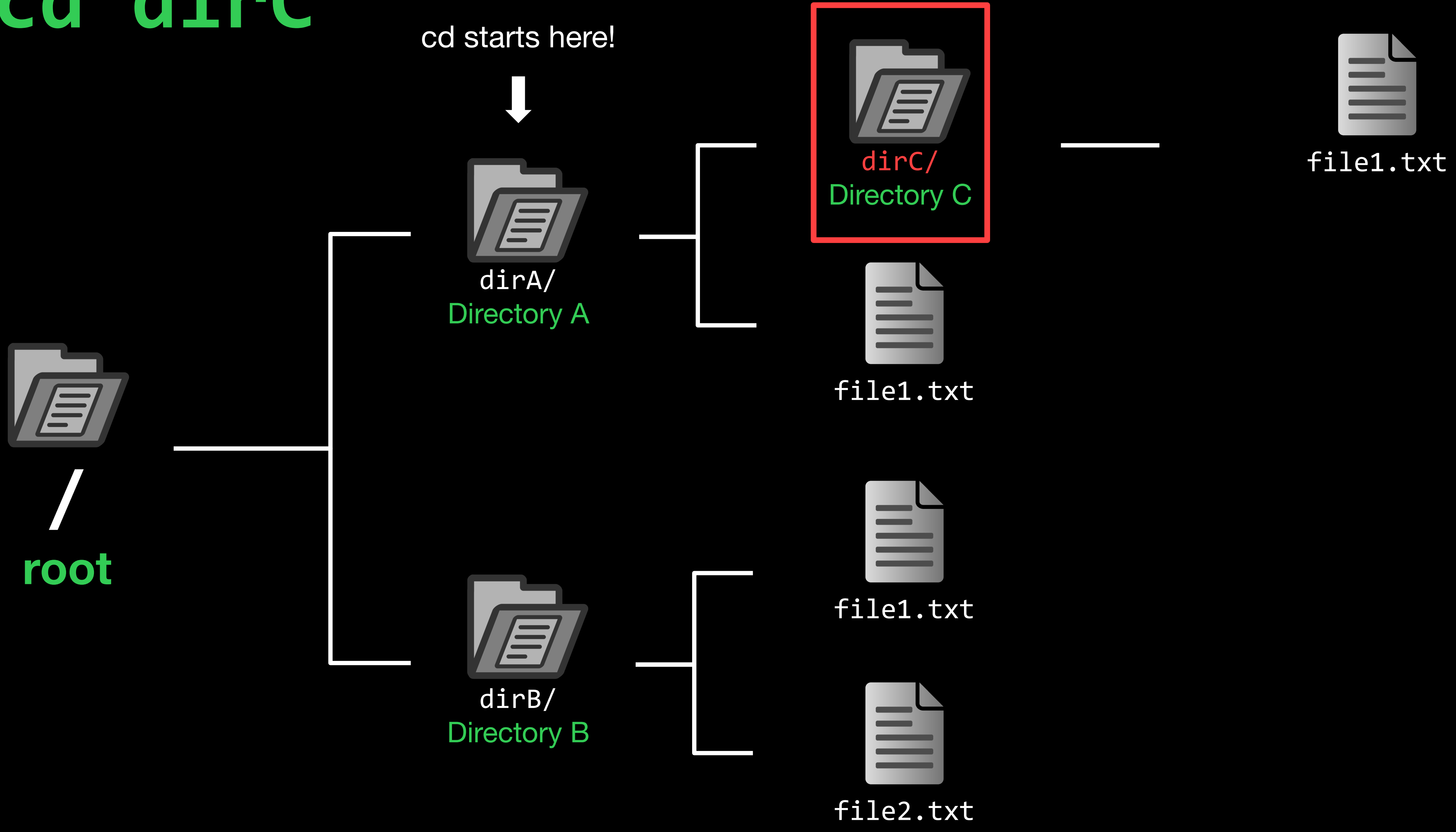

dirC/
Directory C
file1.txt
dirA/
Directory A
file1.txt
/
root
file1.txt
dirB/
Directory B
file2.txt
cd starts here!

cd dirA

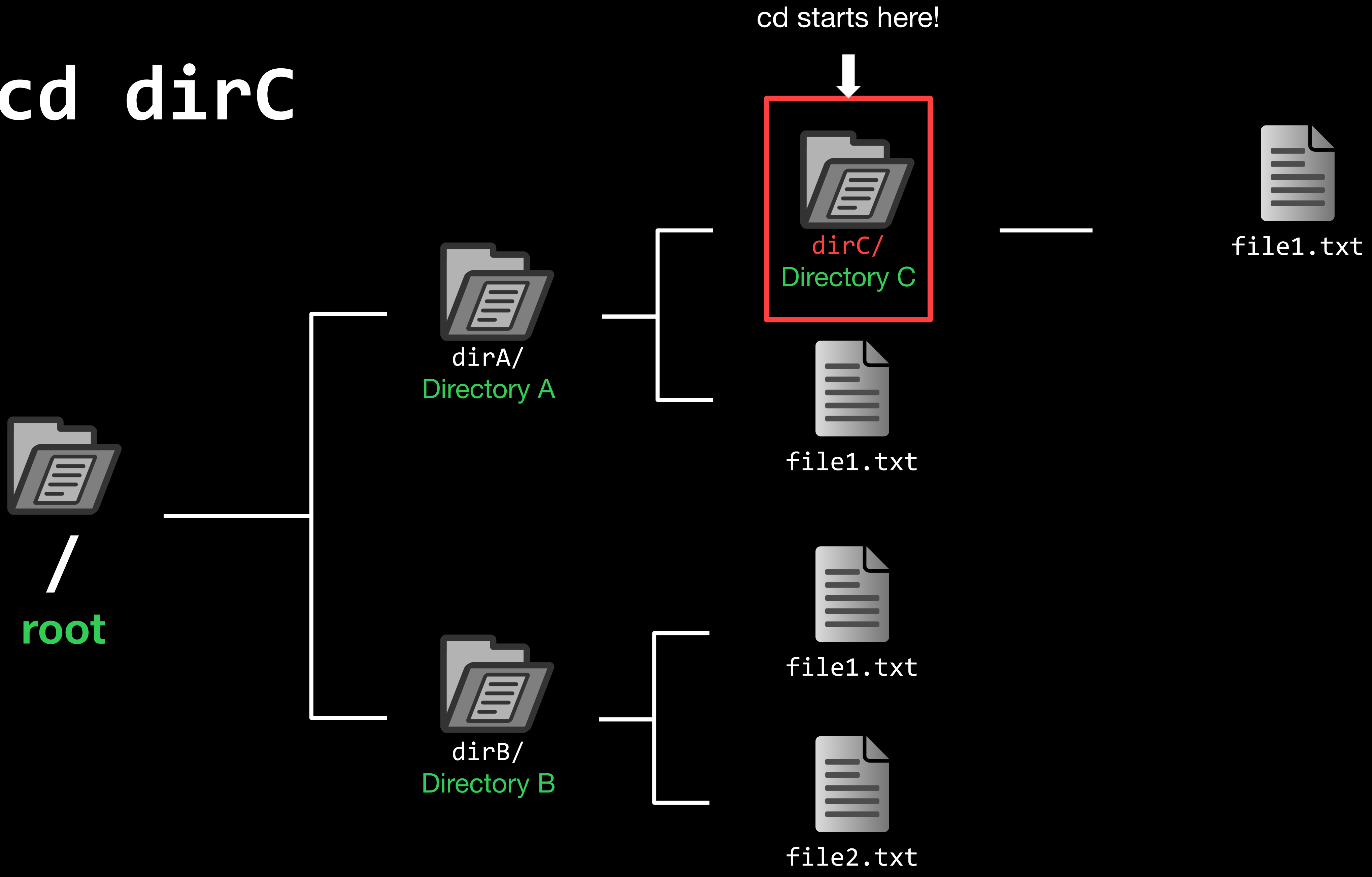

cd starts here!
dirA/
Directory A
dirC/
Directory C
file1.txt
file1.txt
/
root
file1.txt
dirB/
Directory B
file2.txt

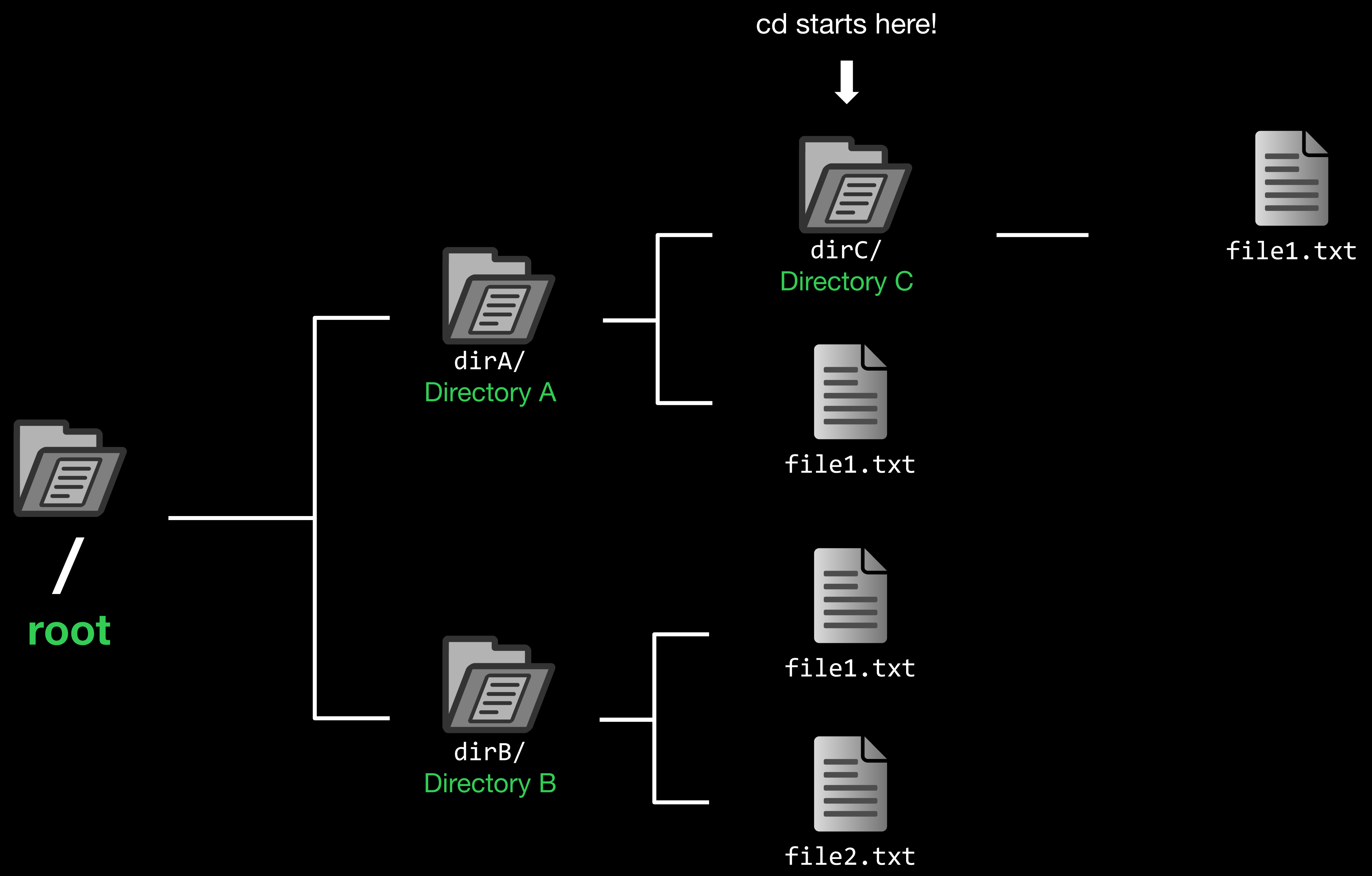

cd starts here!
/
root
dirA/
Directory A
dirB/
Directory B
dirC/
Directory C
file1.txt
file1.txt
file1.txt
file2.txt

cd dirC

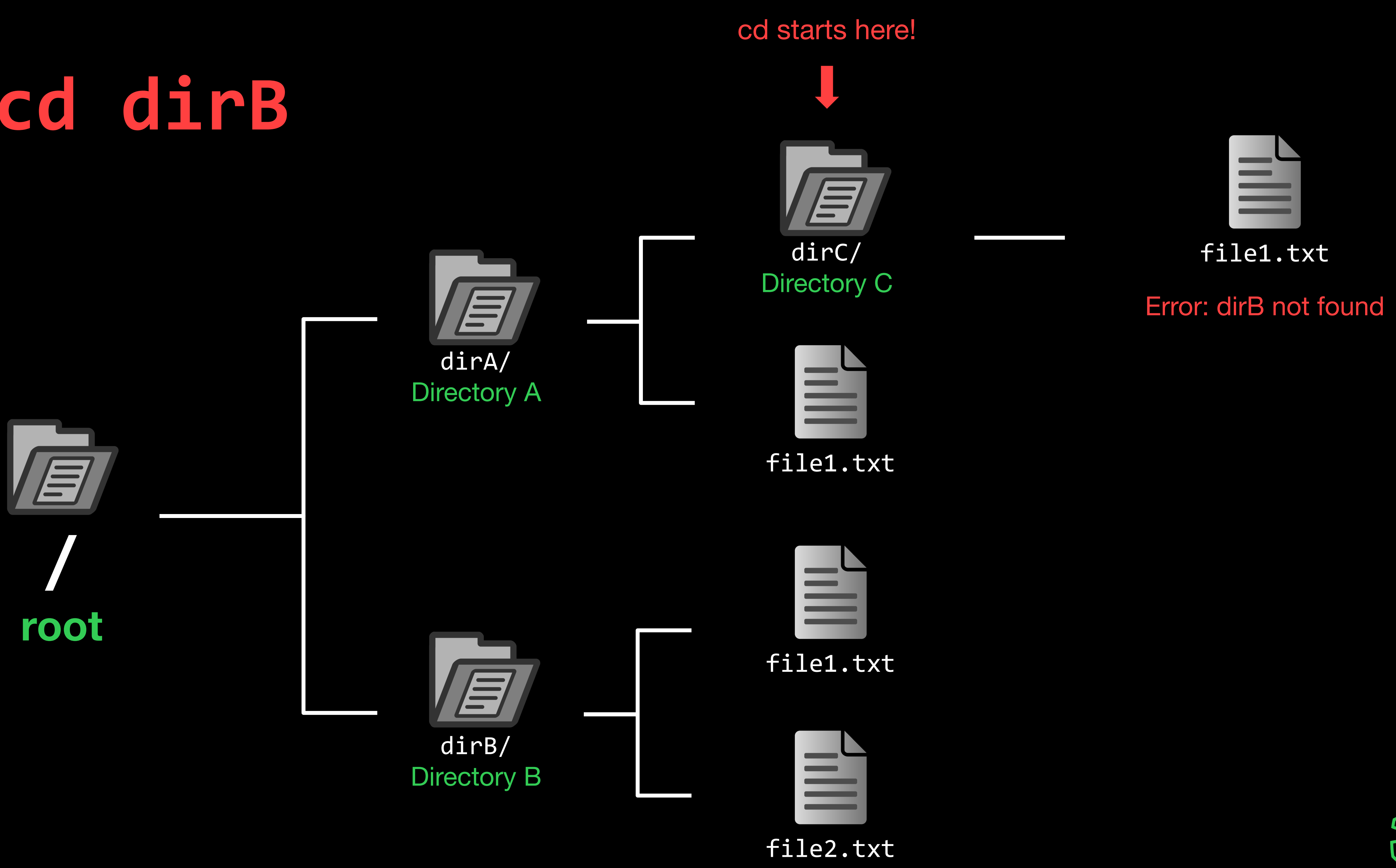

cd starts here!
dirC/
Directory C
file1.txt
dirA/
Directory A
file1.txt
/
root
file1.txt
dirB/
Directory B
file2.txt

cd dirC

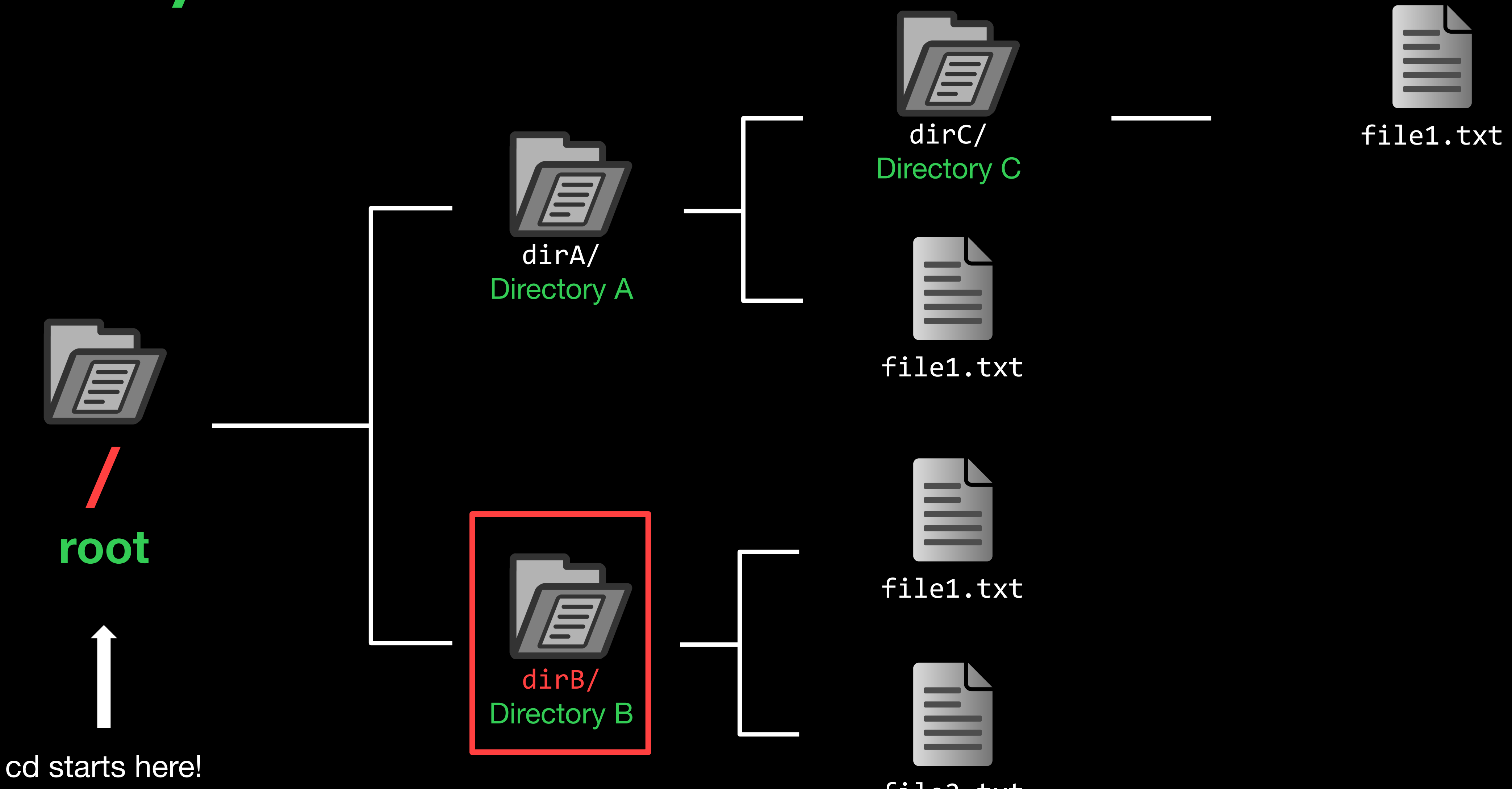

cd starts here!
dirC/
Directory C
file1.txt
dirA/
Directory A
file1.txt
/
root
file1.txt
dirB/
Directory B
file2.txt


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

cd dirB

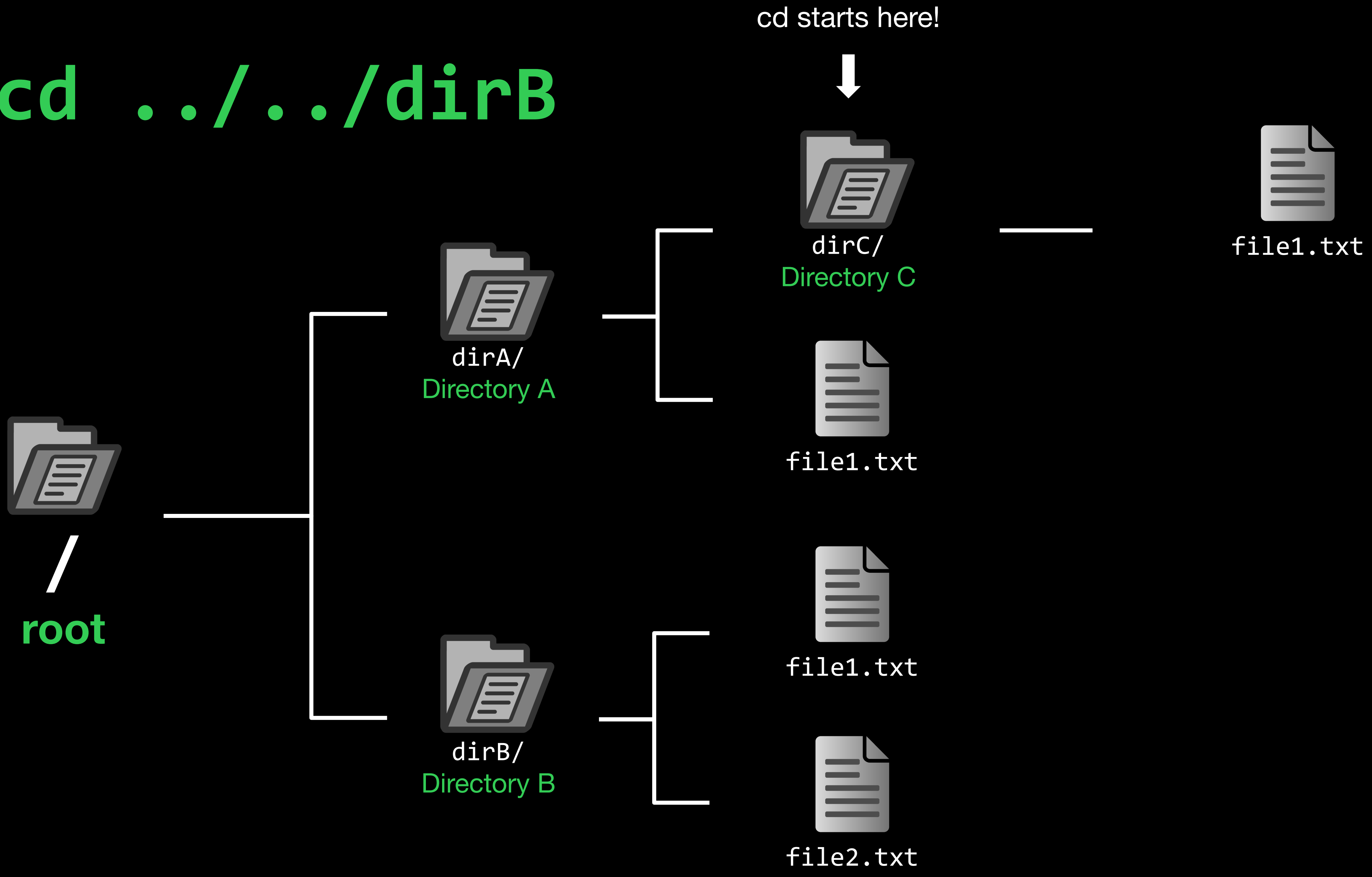

cd starts here!
/
root
dirA/
Directory A
dirB/
Directory B
dirC/
Directory C
file1.txt
file1.txt
file2.txt
file1.txt
Error: dirB not found

cd /dirB

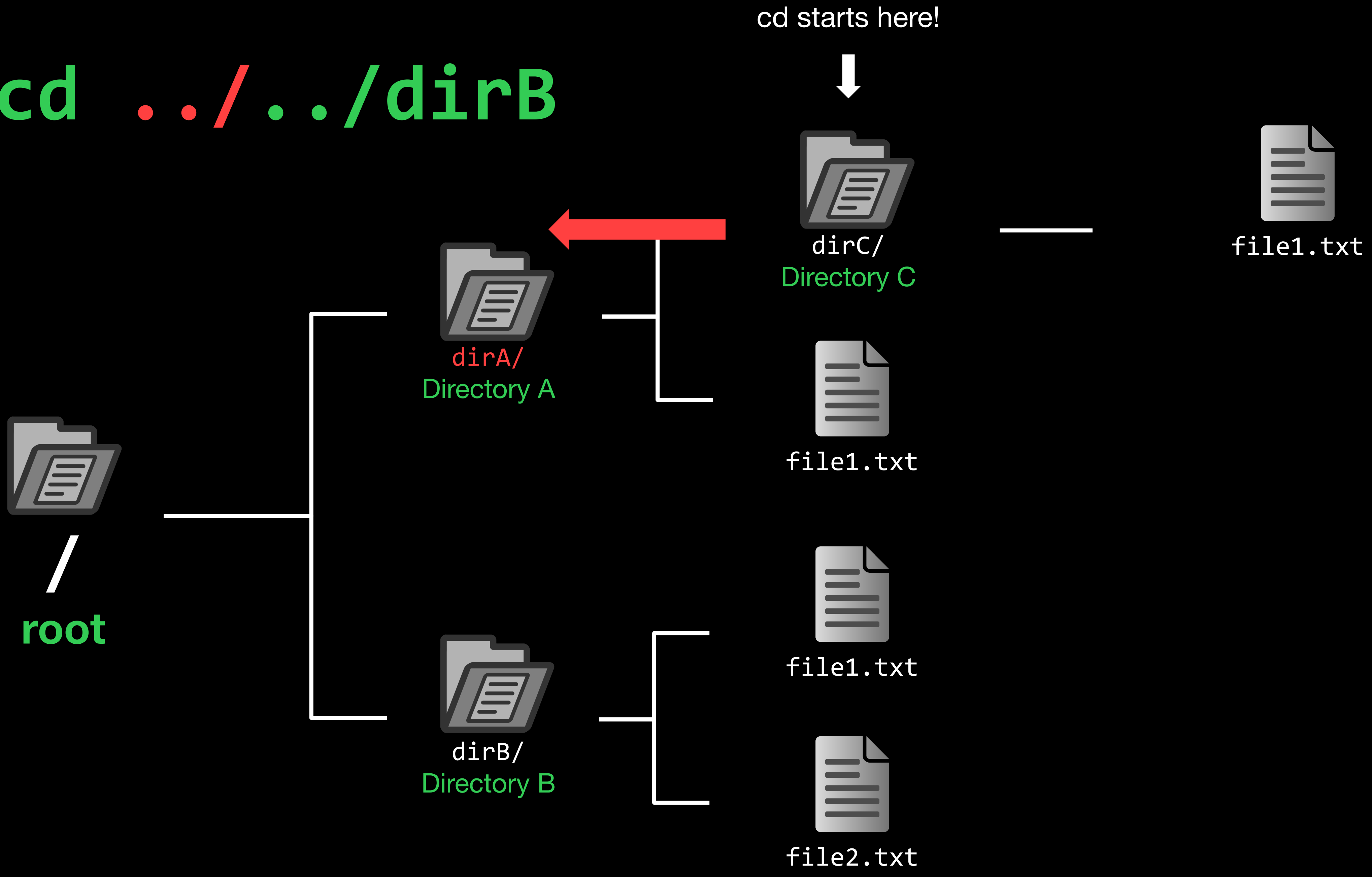


cd starts here!

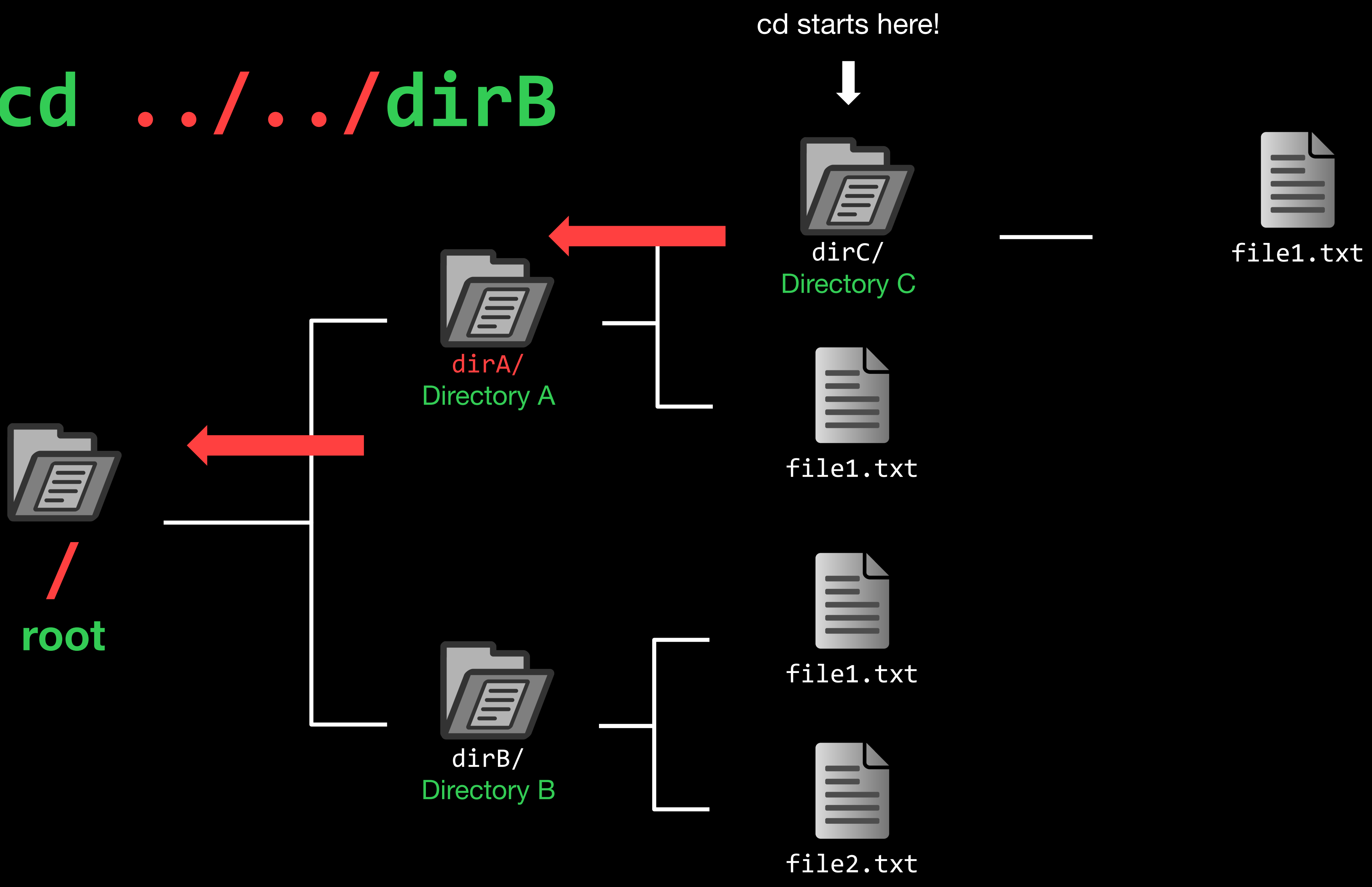
```
cd ../../dirB
```


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

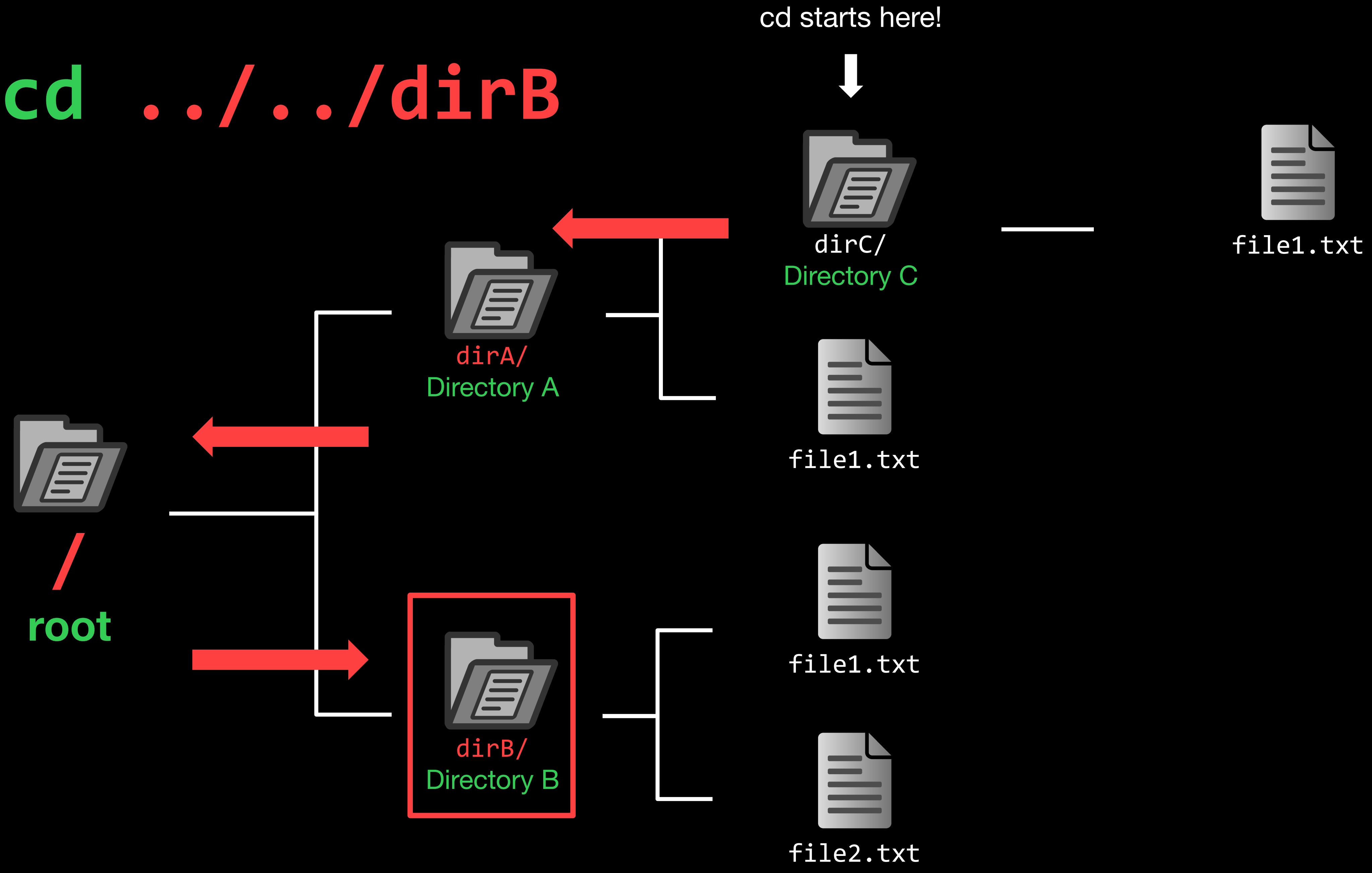
```
cd ../../dirB
```


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

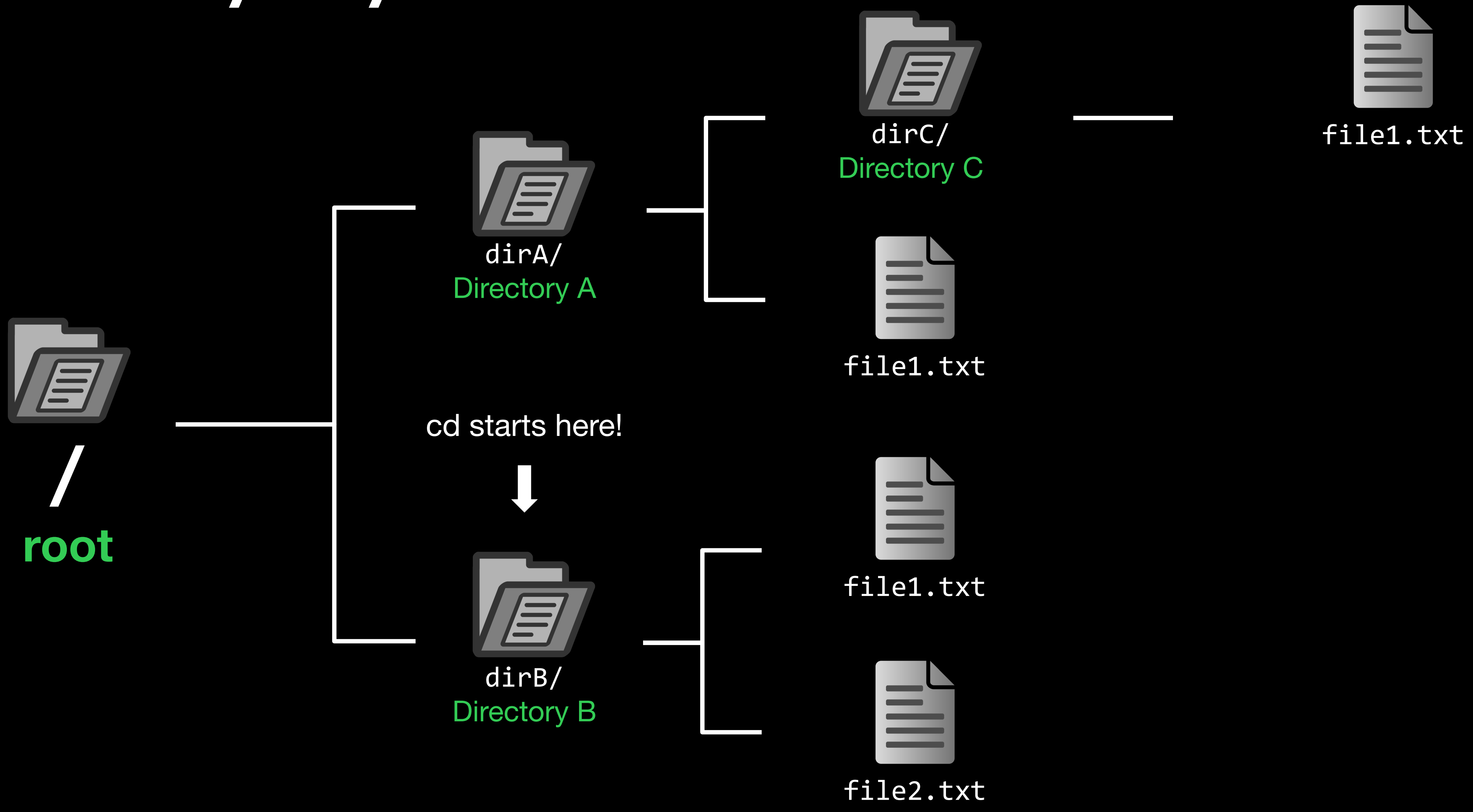
```
cd ../../dirB
```


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

```
cd ../../dirB
```


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

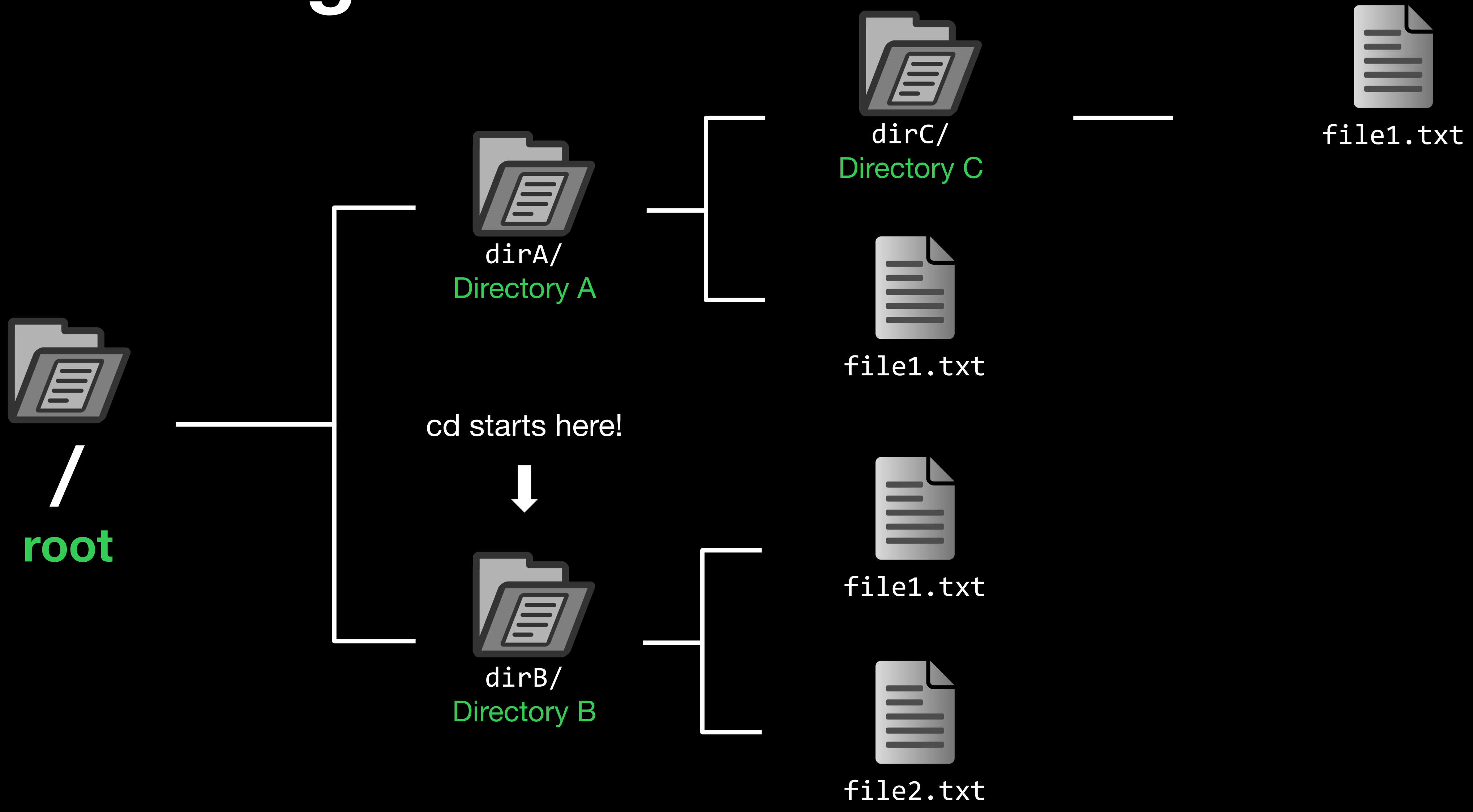
cd starts here!

dirB/
Directory B

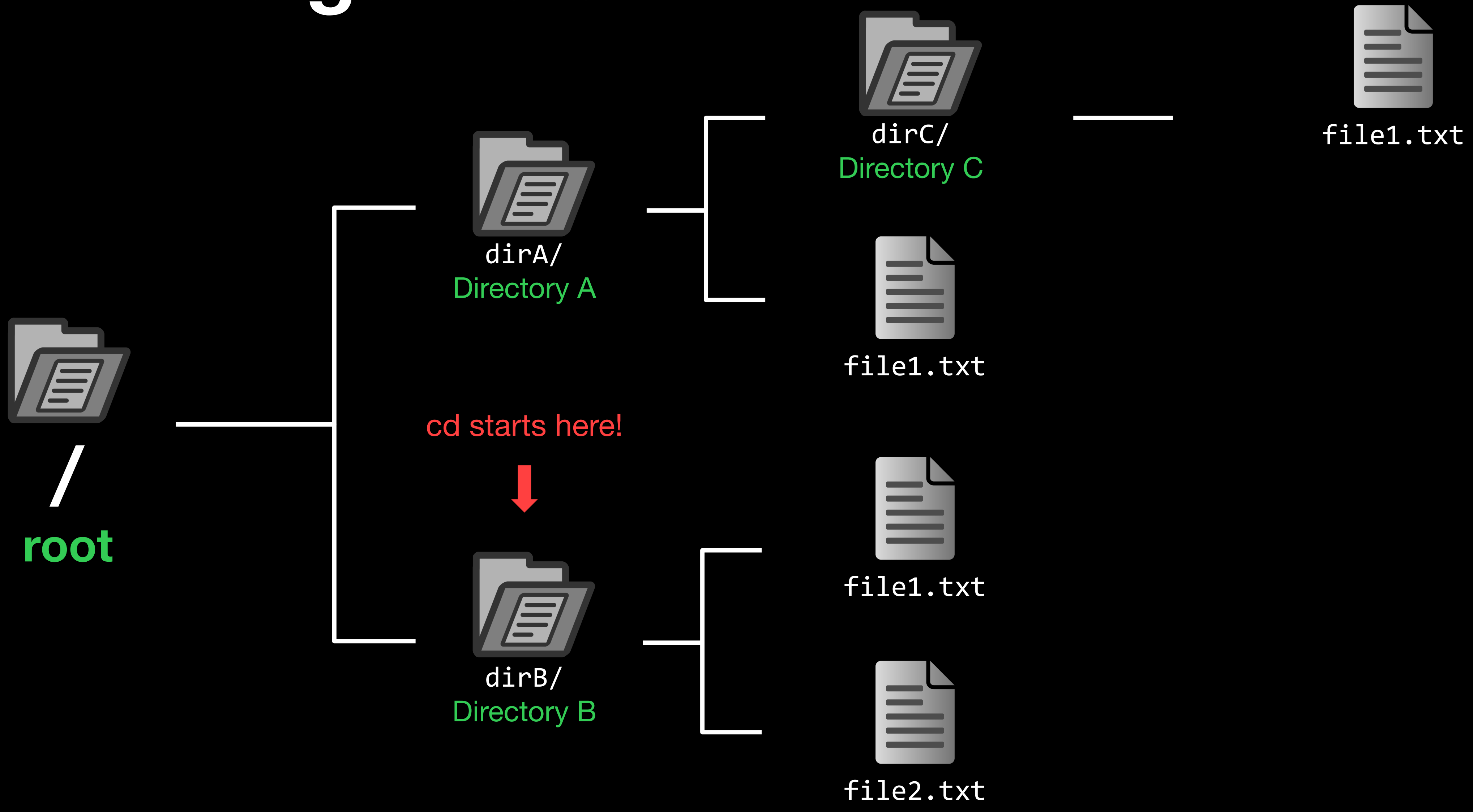
file1.txt

file2.txt

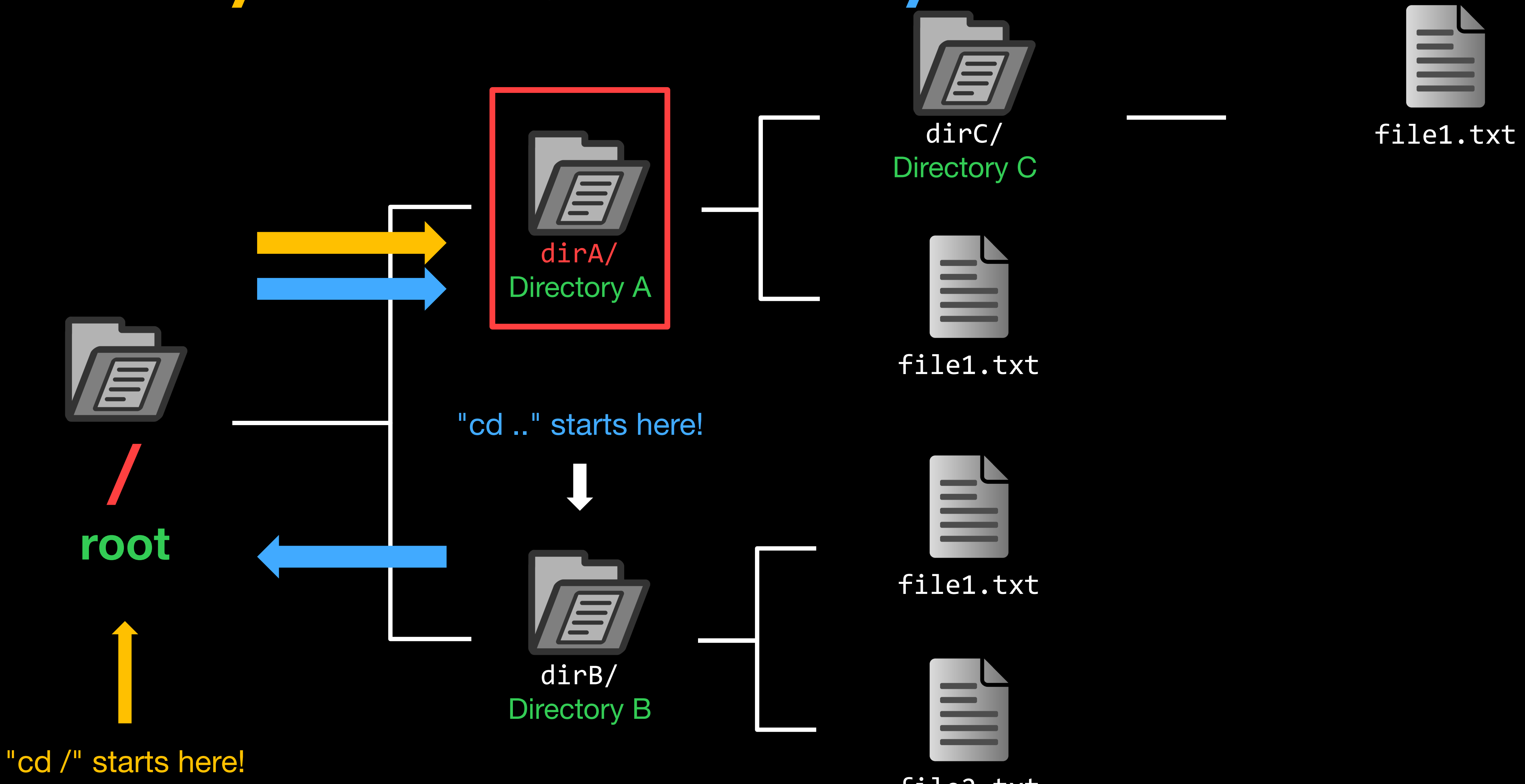
How to get to dirA?



How to get to dirA?



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



Paths

Absolute Path

The full path that always starts at root (/)

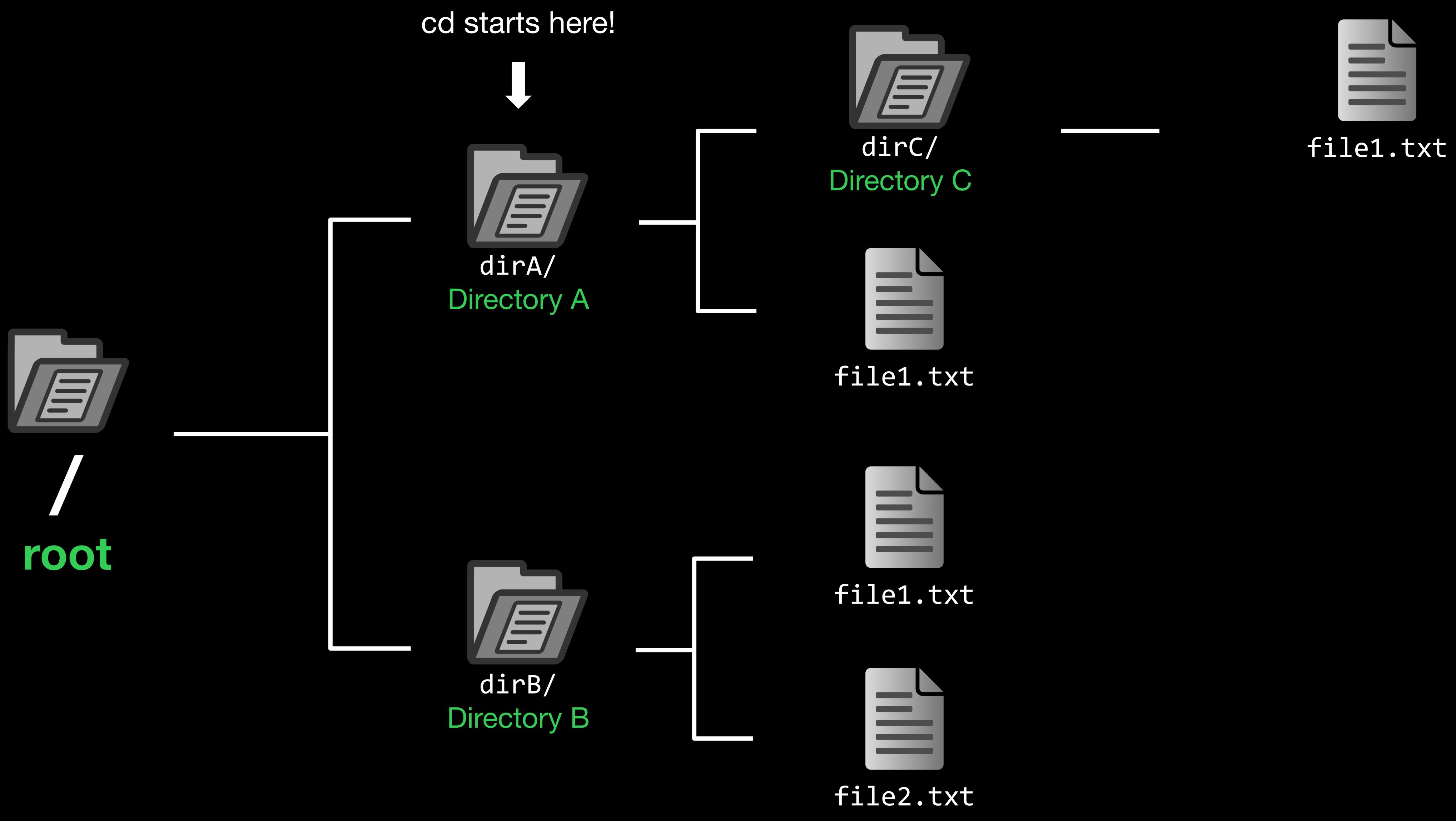
```
/dirA/file1.txt
/dirA/dirC/file1.txt
```

Relative Path

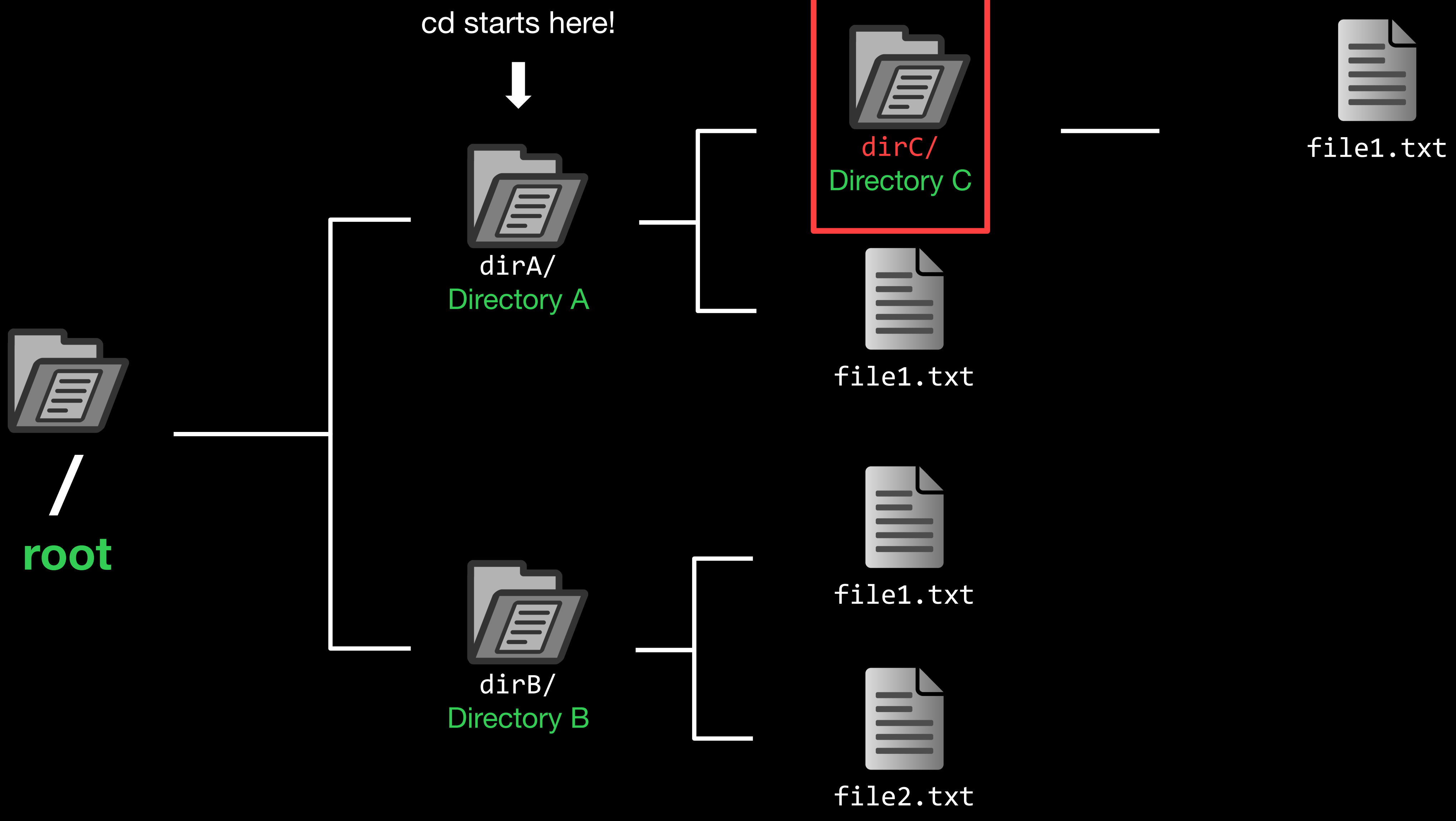
The partial path relative to where you are currently in the terminal

(Relative to dirA)

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


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

(bonus round: SIGPwny Editors meeting)

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

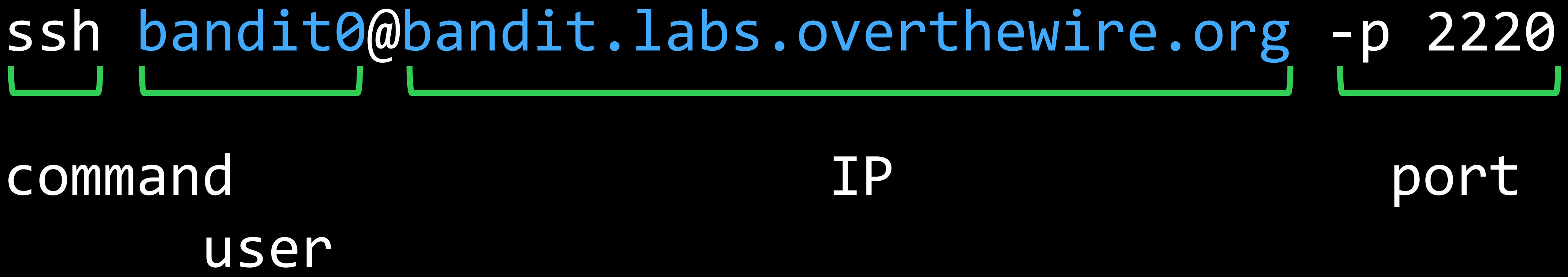
Ports: communication endpoints on your computer (1-65535)

- 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

2024-09-12 • This Thursday

- *Web Hacking I* with Jake and Emma
- Learn introductory knowledge on web hacking

2024-09-15 • Next Sunday

- *Web Hacking II* with Louis
- Learn more advanced web hacking, such as XSS and SSRF!

2024-09-22 • Fall CTF 2024

- Register at sigpwny.com/register24!

ctf.sigpwny.com

sigpwny{starting_off_strong}

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

SIGPwny