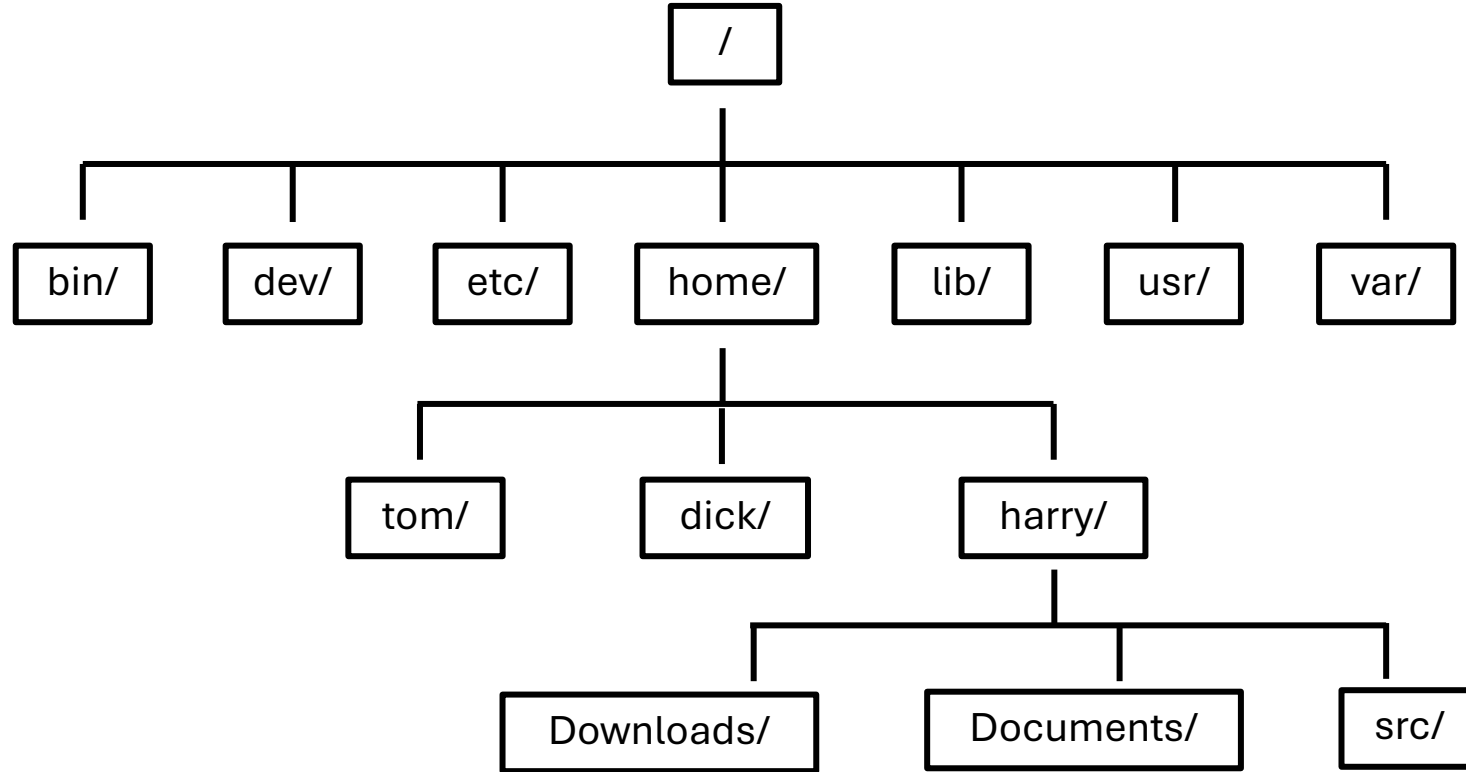# GEO 242

Lecture 1: Installation and the file system

Outline

Purpose of the course

Things to install for Windows and Mac

Background to Unix and Linux

The Unix file system

Purpose of the course

"Things have gone well since I learned how to use the computer!"

Chad Severson, MS, 2012

The purpose of this class is to get you to use computers like a scientist
- Use the command line, natively
- Quickly reformat and search through text files
- Write scripts to automate tasks and make plots
- Make professional, formatted documents with LaTeX
- Quickly visualize and analyze data
- Properly curate your codes

You will need (1)

Access to a computer you can use in class that has a POSIX (Unix or Linux) compatible command line interface

- Windows machines can emulate Ubuntu Linux through 'Windows subsystem for Linux v 2' (WSL2)
- MacOS is based on FreeBSD, a version of Unix
- If you are using a Linux operating system, you should be good!

You will need (2)

To get graphical output from the programs you run, you will need to install an X-Windows client for your system
- For Windows, the (free) VcXSrv package works well
- For Macs, you will need XQuartz (also free)

Installers

To install the packages you need for the course, you will need to run package installers

- aptitude for Linux or WSL (for Debian derivatives like Ubuntu) – already built in
- Homebrew (or MacPorts) for MacOS – you will have to install
- conda for Python on both (prefer miniforge as a conda installer over miniconda, but both will work) – you will have to install

Unix: a history

Unix ('UNICS') was originally developed at Bell Labs in the 1960s as an improvement over their existing mainframe operating system ('MULTICS')

The code (originally written in assembly code, later in C) was shared with anyone who requested it, as AT&T were not legally allowed to sell software at that time

Some of these licensees (e.g. UC Berkeley) started to make their own additions to the code, and rewrite portions of the originals; these versions gradually became more popular than the original Unix, after AT&T gained permission to commercialize the code

Components of Unix

Unix is divided into several different components with different roles:

The kernel: boot code, hardware and memory control, header information that can be incorporated in other programs

The development environment: specific programs and code (e.g. compilers) for generating the whole operating system from scratch, and for generating programs that work in it

Commands: programs fulfilling various system operations, e.g. the shell, utilities, graphics, text editors

Documentation: manuals and online help

The Unix file system (1)

Unix was responsible for popularizing the directory-based file system, now common in almost all operating systems (it is one of the few features that originated in MULTICS)

In Unix, files and directories have both local and absolute addresses

- a local address gives the location of a file relative to the current directory

- a global address gives the location of a file relative to the 'root' directory (/)
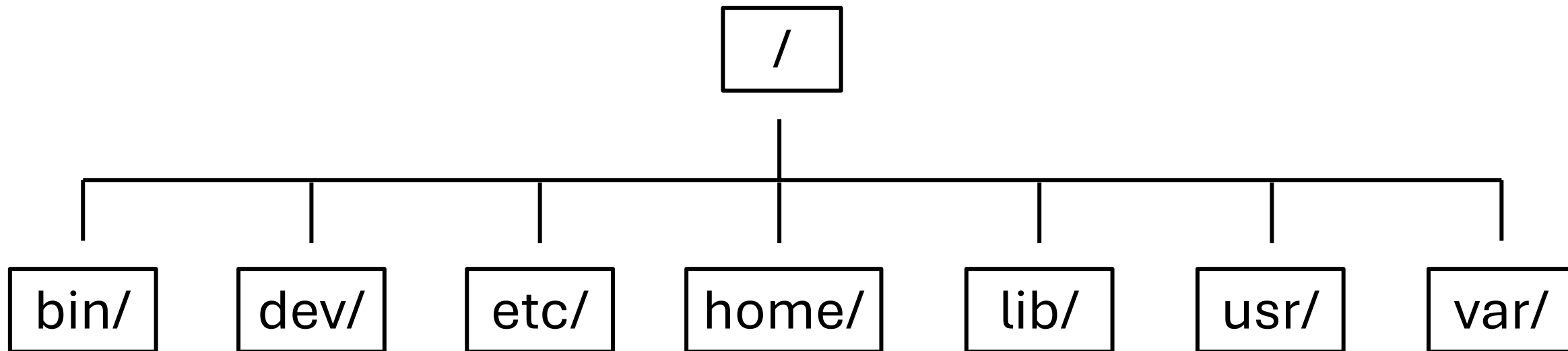
The Unix file system (2)

Each level of the hierarchy is separated by a '/' (forward slash) when writing out an address

Each directory contains two hidden links to directories – . (present directory) and .. (previous directory) – these are present as hidden files in every directory

root directory

```
┌─────┐
│  /  │
└─────┘
```
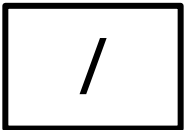
bin – binary files for system commands
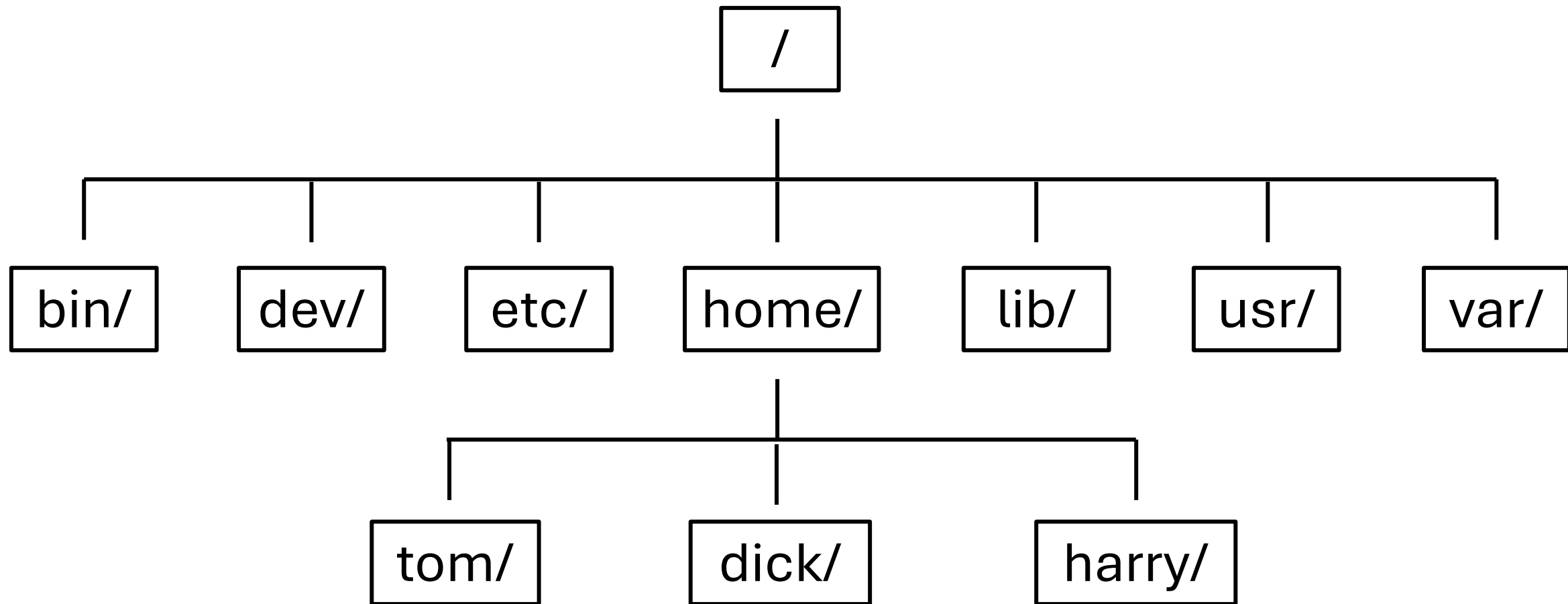dev – information on connected devices
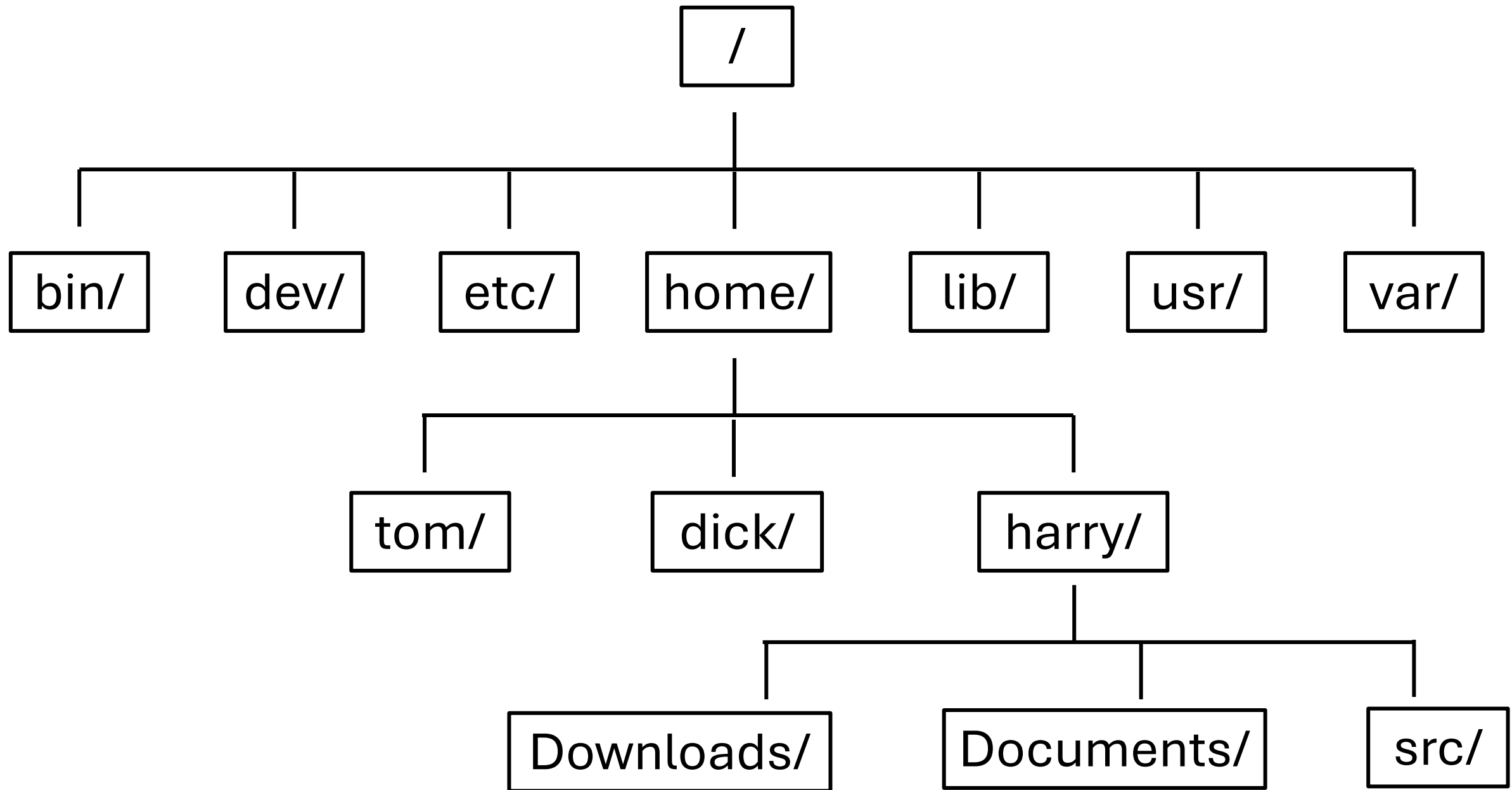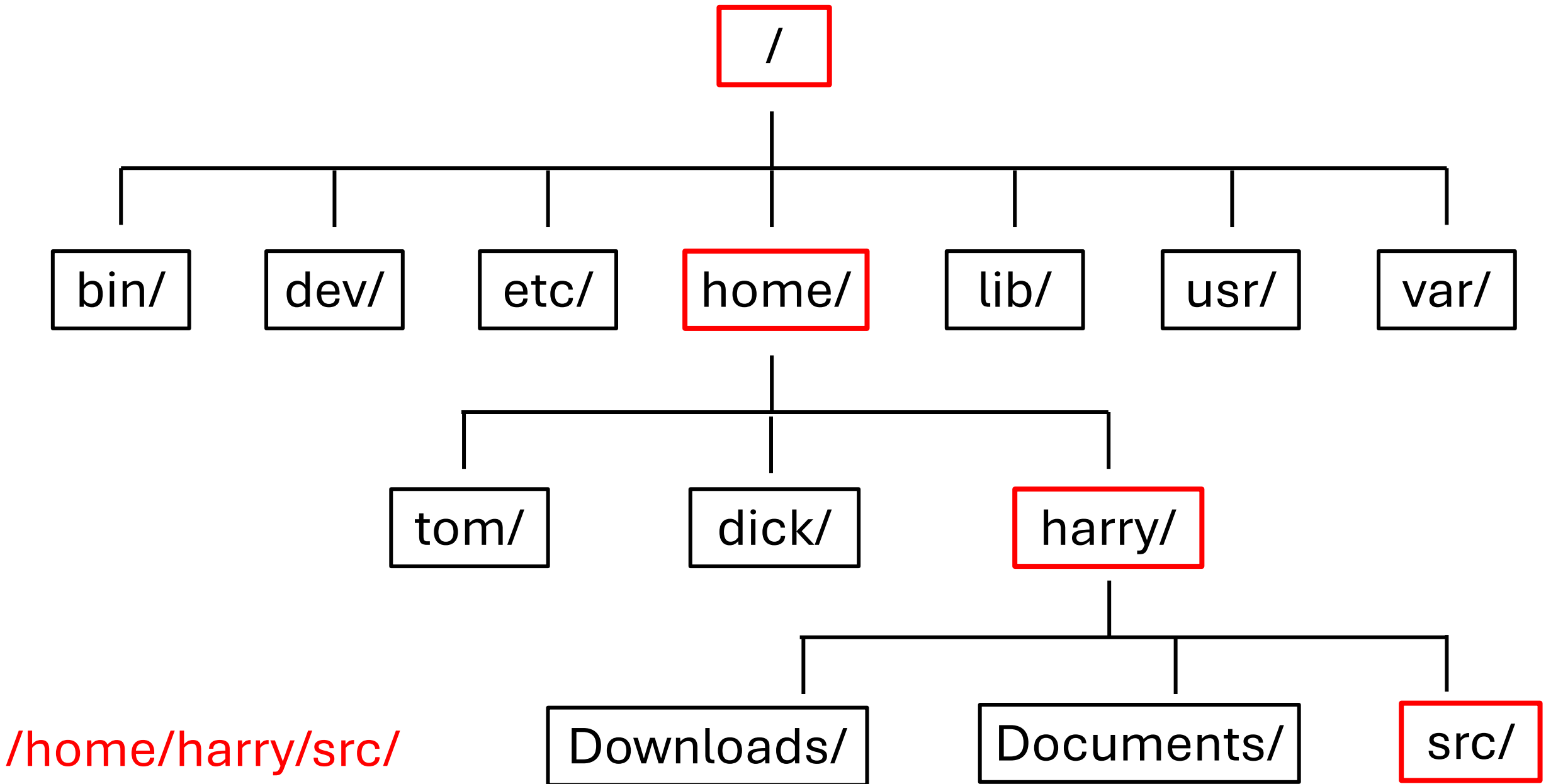etc – system configuration files
home – user files
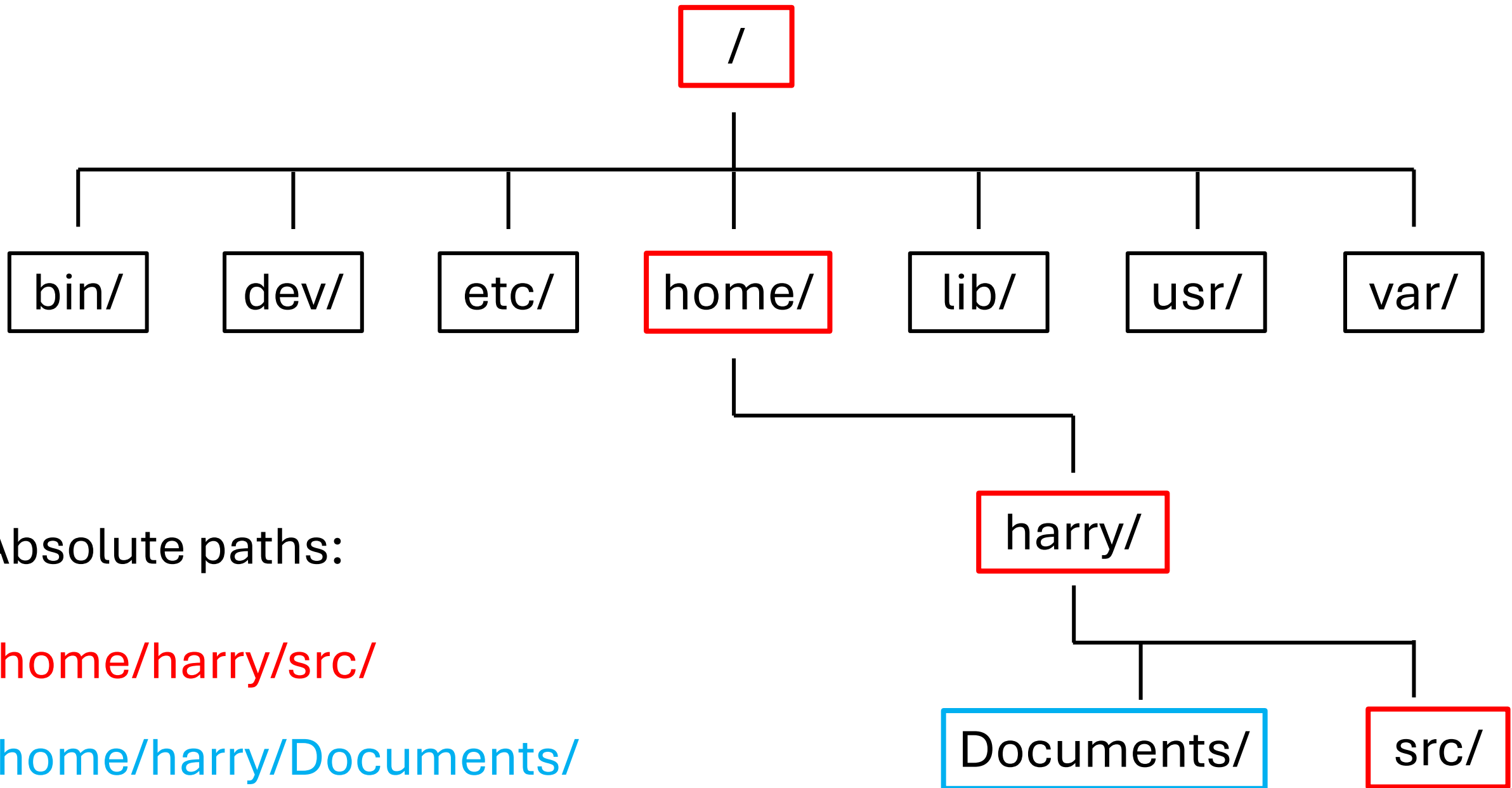lib – system libraries
usr – user-installed binaries and libraries
var – system log files and temporary files

```
                            ┌───────┐
                            │   /   │
                            └───────┘
                                │
      ┌──────────┬──────────┬───┴────┬──────────┬──────────┐
  ┌───────┐  ┌───────┐  ┌───────┐ ┌───────┐ ┌───────┐  ┌───────┐  ┌───────┐
  │ bin/  │  │ dev/  │  │ etc/  │ │ home/ │ │ lib/  │  │ usr/  │  │ var/  │
  └───────┘  └───────┘  └───────┘ └───────┘ └───────┘  └───────┘  └───────┘
                                      │
                         ┌────────────┼────────────┐
                     ┌───────┐    ┌───────┐    ┌───────┐
                     │ tom/  │    │ dick/ │    │ harry/│
                     └───────┘    └───────┘    └───────┘
```
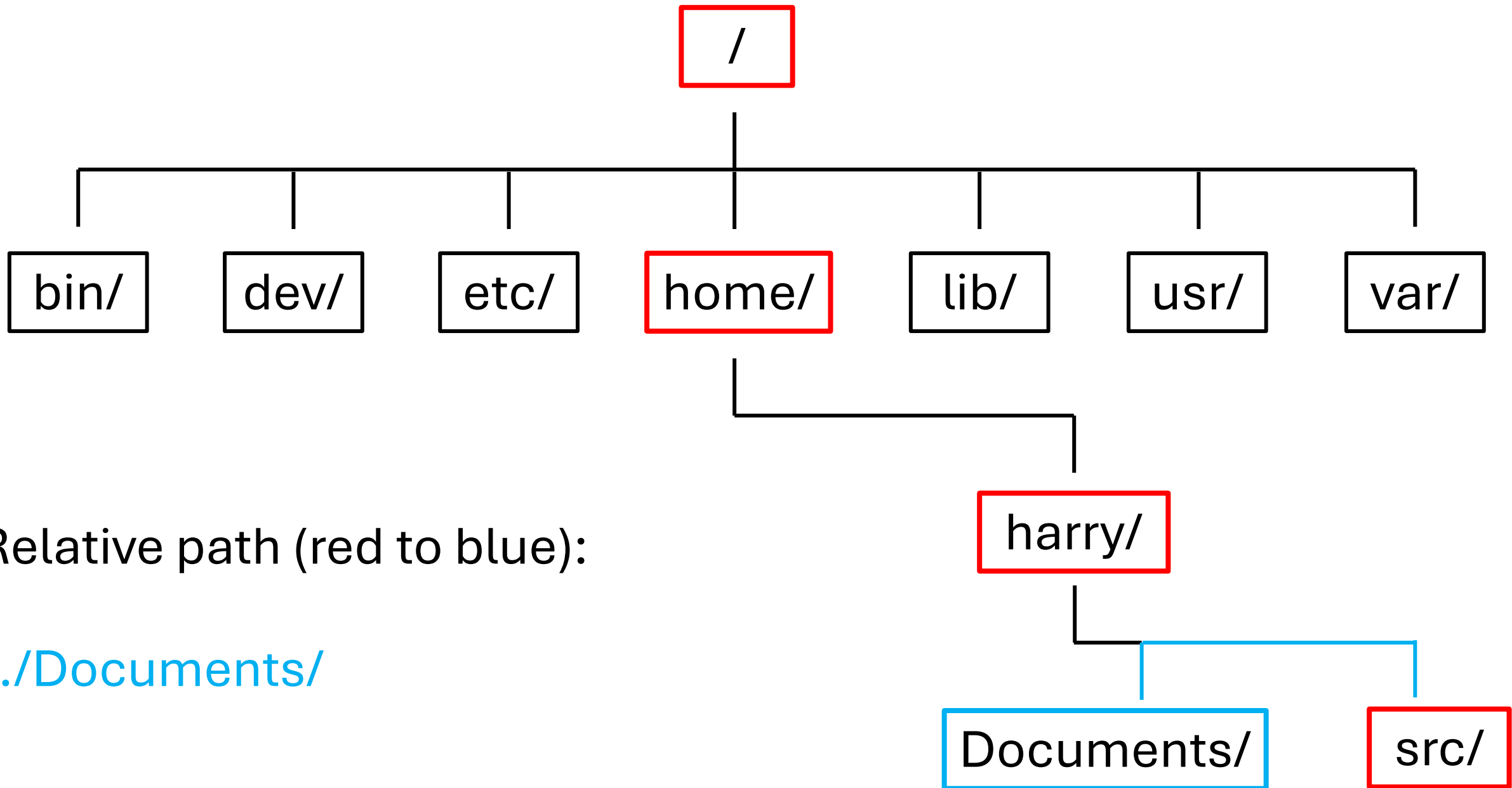
/home/harry/src/

Absolute paths:

/home/harry/src/

/home/harry/Documents/

Relative path (red to blue):

../Documents/

Absolute path:

/home/tom/Downloads/

Relative path (red to green):

../../tom/Downloads/
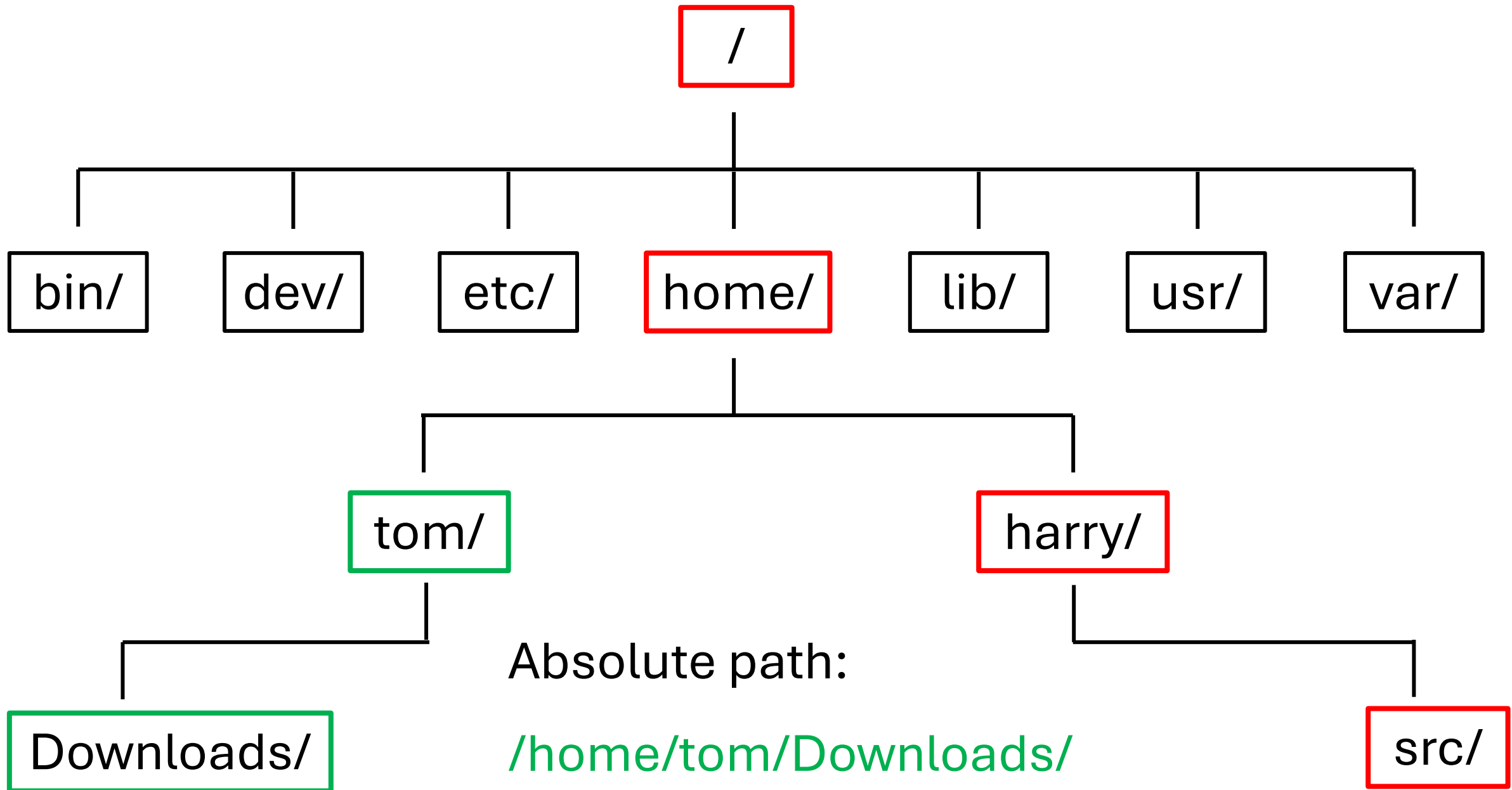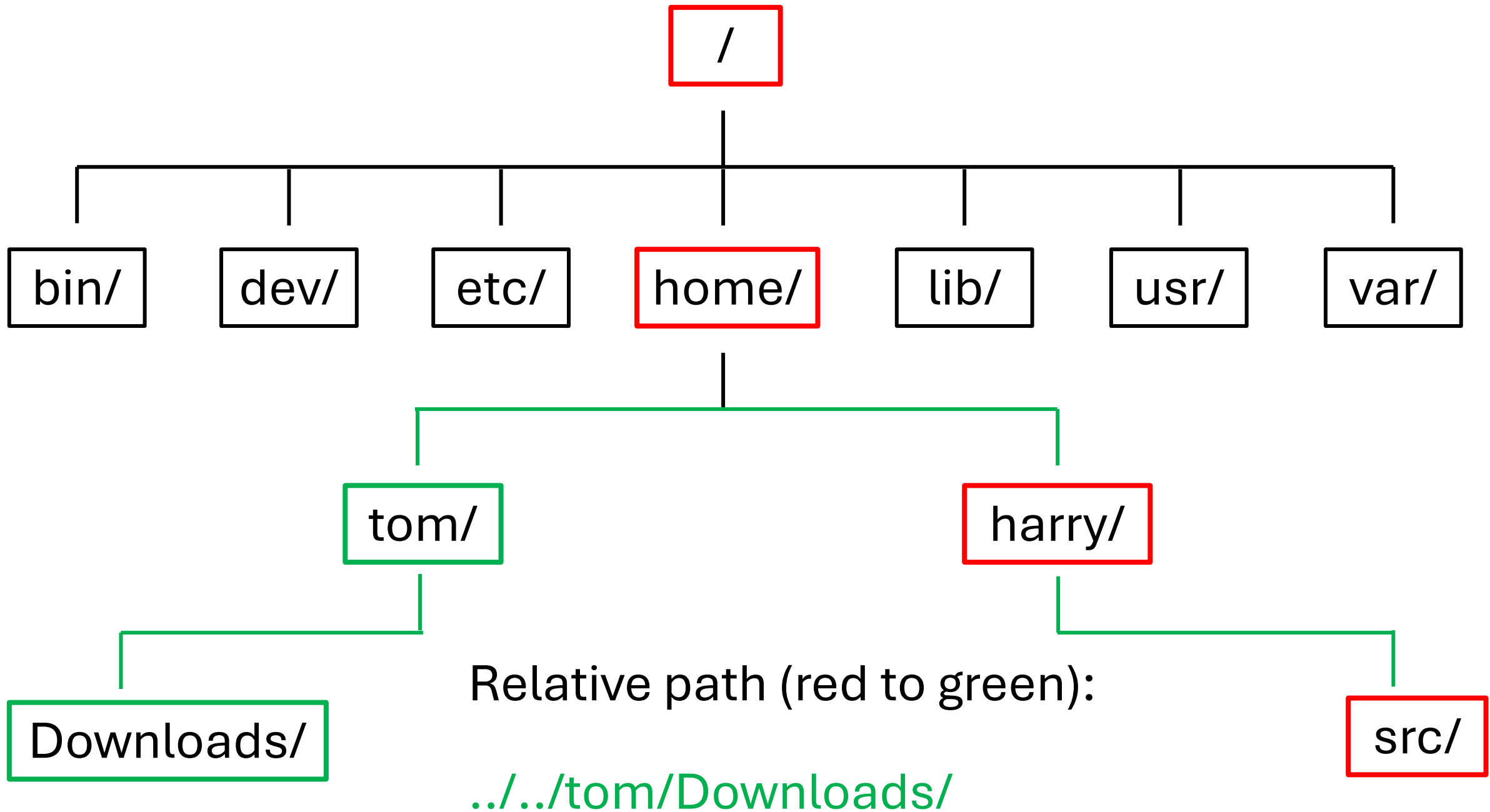
Variants and impersonators

The Berkeley Software Distribution (BSD) version of Unix was developed in parallel with the AT&T version, and was shared freely; now several open source variants exist

BSD was the foundation of SunOS (Solaris), the operating system for Sun workstations; FreeBSD, a freely-available open-source version, is the basis for MacOS (Darwin)

Linux, which has many similarities to Unix, started in the early 1990s as a simplified operating system for PCs

A lawsuit between AT&T and Berkeley in the early 1990s suspended the release of BSD variants for two years; many developers switched to Linux at this time

Linux distributions

Multiple organizations have developed packaged 'distributions' of Linux, including desktop environments and package installers on top of the standard commands and kernel. These include:

- Debian: a non-commercial distribution maintained by volunteers. Several distributions are built on top of Debian, such as **Ubuntu** (my preferred distribution, now available in Windows 10/11), Linux Mint and Knoppix.

- Red Hat: a distribution family sponsored by the American company of the same name. Includes the free distributions Fedora and CentOS, and commercial packages such as Red Hat Enterprise Linux (RHEL)

Why did it catch on?

Despite limited consumer use*, Unix (and its relatives) has a strong hold on 'industrial' applications (e.g. servers) and academic computing. Why?

– It was distributed freely

– It could run on any computer with a C compiler

– It was well documented

– It was easy to adapt (all configuration files were simple ASCII text, all source code was provided)

– BSD was the first computer operating system to have Internet protocols (TCP-IP)

* Android is Linux and iOS is BSD, so maybe it's not 'limited'