Navigating the Filesystem

Dr. Princess Rodriguez

2025-01-10

Learning Objectives

- Understand the organization of the Filesystem
- Distinguish between Full versus Relative paths
- listing, copying, creating, moving and removing data

Navigating the Filesystem

A filesystem organizes a computer's files and directories into a tree structure.

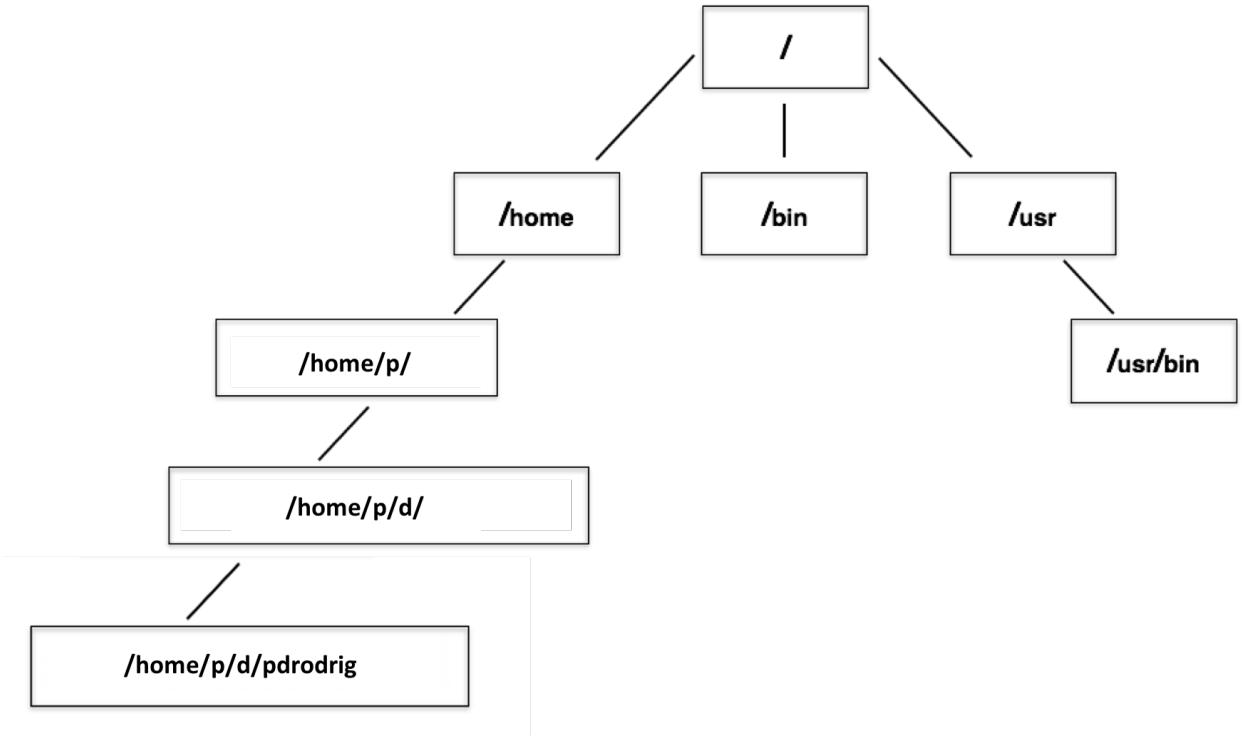


Figure 1: Directory Structure

The image above illustrates the Filesystem.

Filesystem

- The first directory in the filesystem is the **root directory**. It is the parent of all other directories and files in the filesystem. That / or root is the 'top' level.
- Each parent directory contains child directories and/or files.
- Each child directory can also contain more files

Note: When you log in to a remote computer you land on one of the branches of that tree, i.e. your pre-designated "home" directory that usually has your login name as its name (e.g. `/users/username`).

Shortcut: Tab Completion

tab
caps lock

Typing out file or directory names can waste a lot of time and its easy to make typing mistakes. Instead we should get in the habit of using tab complete as a shortcut. The `tab` key is located on the left side of your keyboard, right above the `caps lock` key. When you start typing out the first few characters of a directory name, then hit the `tab` key, Shell will try to fill in the rest of the directory name.

File Paths

The command to check our current location is `pwd`, this command does not take any arguments and it returns the path or address of your **p**resent **w**orking **d**irectory (the folder you are in currently).

```
pwd
```

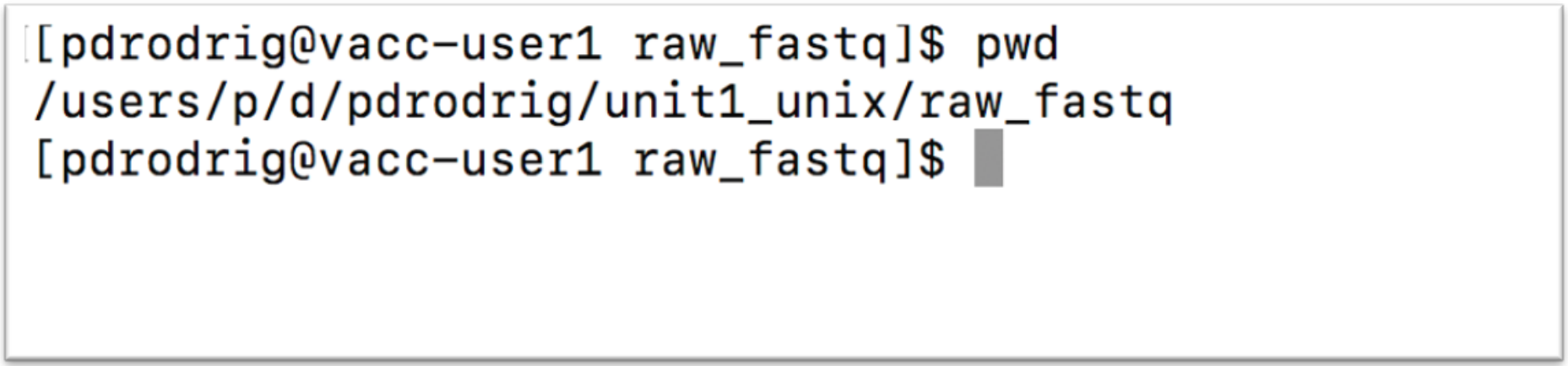


Figure 2: print working directory (pwd)

File Paths continued

Each folder is separated from its "parent" or "child" folder by a "/", and the output starts with the root / directory. So, now you are able to determine the location of `raw_fastq` directory relative to the root directory.

The Tilde ~

The "~" is an abbreviation for the current user's home folder.

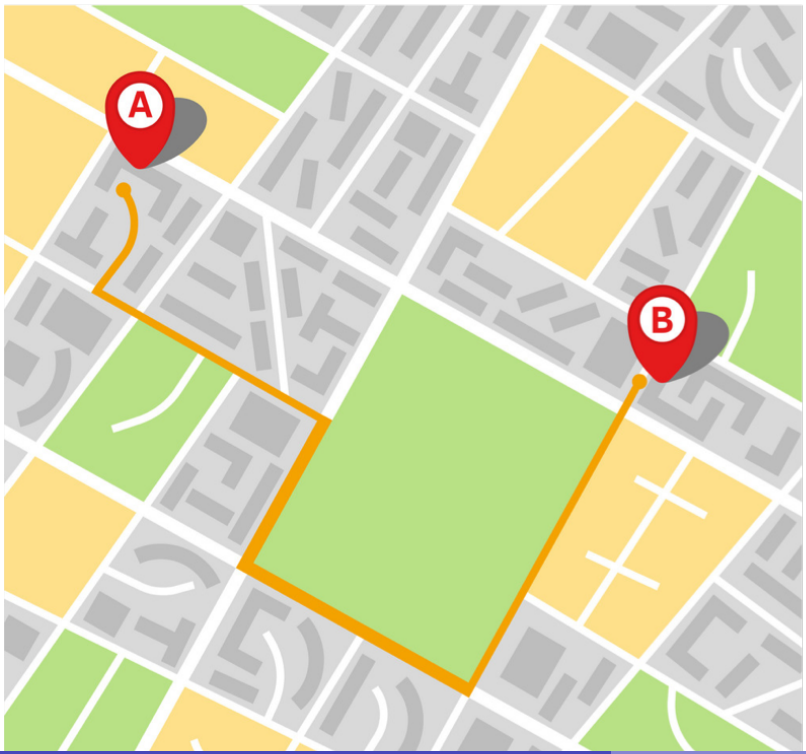
```
cd ~
```

Synopsis of Full versus Relative paths

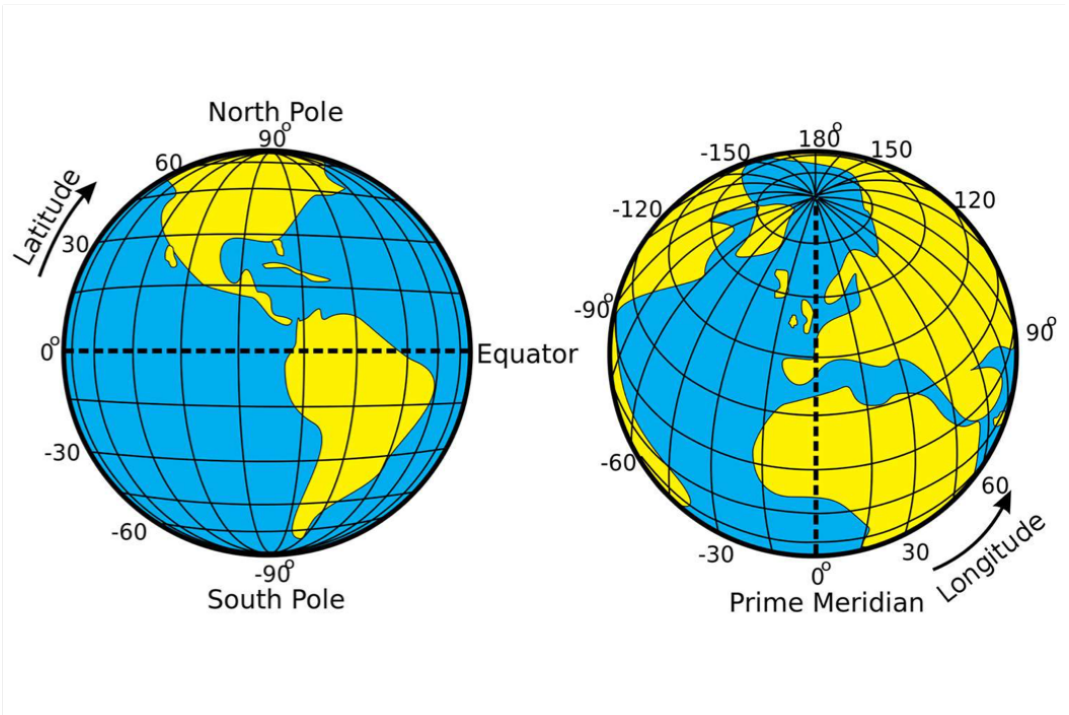
A full path always starts with a /, a relative path does not.

A full path is like GPS coordinates. It tells you exactly where something is no matter where you are right now.

Relative Paths



Absolute Paths



Copying

The copy command has the following syntax:

```
cp  path/to/item-being-copied  path/to/new-copied-item
```

Creating

Next, let's create a directory called `fastq_backup` and we can move the copy of the fastq file into that directory.

```
mkdir fastq_backup
```

Moving

We can now move our copied fastq file in to the new directory. We can move files around using the move command, `mv`, syntax:

```
mv  path/to/item-being-moved  path/to/destination
```

Renaming

The `mv` command has a second functionality. You can use `mv` to rename files too. The syntax is identical to when we used `mv` for moving, but this time instead of giving a directory as its destination, we just give a new name as its destination.

Removing

```
rm  Mov10_oe_1.subset-backup.fq
```

Important notes about `rm`:

- `rm` permanently removes/deletes the file/folder.
- There is no concept of "Trash" or "Recycle Bin" on the command-line. When you use `rm` to remove/delete they're really gone.
- **Be careful with this command!**
- You can use the `-i` argument if you want it to ask before removing, `rm -i file-name`.

Summary of Commands

```
cd              # Change Directory
                   +   used to move throughout the filesystem of a

ls              # List
                  +   list the contents of a directory

pwd             # Print Working Directory
                  +  displays the file path from the root director



cp              # Copy
                  +   used to copy files or directories

mkdir           # Make Directory
                  +   used to make a new directory
```

Summary of Commands continued

```
mv              # Move
                  +   move a file into a directory

rm              # Remove
                  +   used to delete files and directories
```

Citation

This lesson has been developed by members of the teaching team at the Harvard Chan Bioinformatics Core (HBC). These are open access materials distributed under the terms of the Creative Commons Attribution license (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

- *The materials used in this lesson were derived from work that is Copyright © Data Carpentry (http://datacarpentry.org/). All Data Carpentry instructional material is made available under the Creative Commons Attribution license (CC BY 4.0).*
- *Adapted from the lesson by Tracy Teal. Original contributors: Paul Wilson, Milad Fatenejad, Sasha Wood and Radhika Khetani for Software Carpentry (http://software-carpentry.org/) authors: Sheldon McKay, Mary Piper, Radhika Khetani, Meeta Mistry, Jihe Liu date posted: September 28, 2020*

Other parts of this lesson were developed from