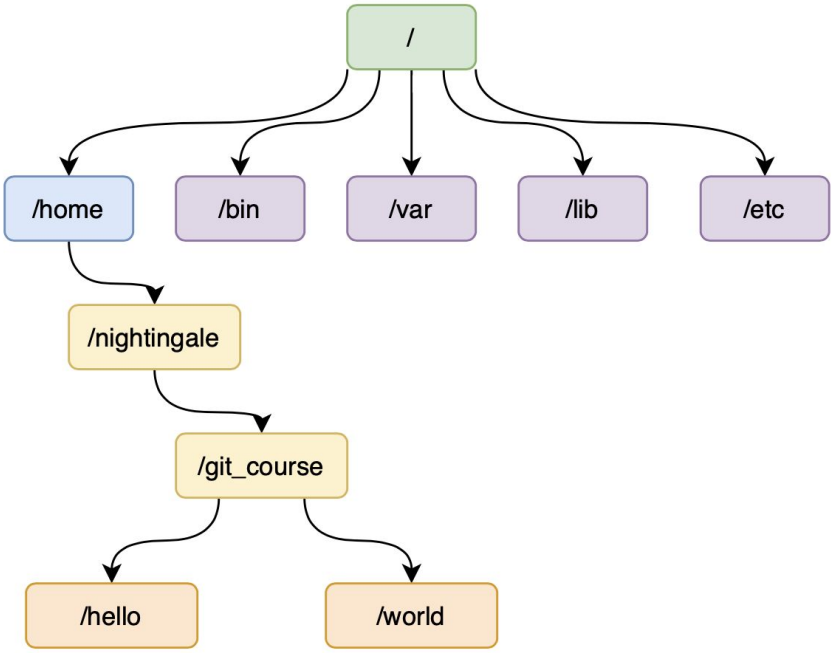
Software Engineering with Python

Command line interface

Alexander Solovyev

File system

Unix filesystem tree



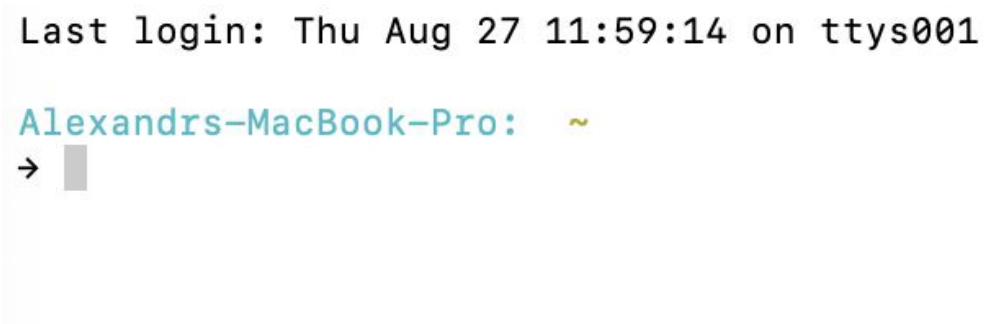
- At the top of the Unix file system is what is called the **root** directory, which is always just “ / ” in Unix

- **Absolute path** to any file in the filesystem is the sequence of directories, starting from the root and ending with the file

command-line interface

A command-line interface (CLI) - text interface to your computer

Referred as *shell, terminal, console, prompt*



Commands

- To give a command to a UNIX system you type the name of the command, along with any associated information, such as a filename

- general form of a UNIX command is:

  command [-option(s)] [argument(s)]

man

If you don’t know what command is doing it’s useful to use **man** command

```
$ man ls

LS(1)                     BSD General Commands Manual                    LS(1)

NAME
     ls -- list directory contents

SYNOPSIS
     ls [-ABCFGHLOPRSTUW@abcdefghiklmnopqrstuwx1%] [file ...]

DESCRIPTION
     For each operand that names a file of a type other than directory, ls
     displays its name as well as any requested, associated information.  For
     each operand that names a file of type directory, ls displays the names
     of files contained within that directory, as well as any requested, asso-
     ciated information.
...
```

Navigation

Tree

- ***Tree*** is the tool which helps you to visualise your file system

```
$ tree
.
|____hi
| |____my_name_is
| | |____alex
|____hello
| |____world
```

ls(list)

- ***ls*** command gives you a list of everything what is in your current directory

```
$ ls
Applications Desktop
```

- If you need to see dotted directories, use ***-a*** option

```
$ ls -a
Applications Desktop .local .bashrc
```

- It could be useful to display this list in **long format**. Here you can get additional information about your files, such as file size etc.

```
$ ls -l
drwx------+   6 nightingale  staff       192 Apr 20 19:26 Music
drwx------+   6 nightingale  staff       192 May 25 02:30 Pictures
drwxr-xr-x+   4 nightingale  staff       128 Dec  4  2018 Public
drwxr-xr-x   80 nightingale  staff      2560 Jun 15 21:17 PycharmProjects
```

You can combine arguments.

The following is how to do an ***ls*** listing sorted by increasing size

```
$ ls -lraS
...
drwx------+   4 nightingale  staff       128 Oct 15  2019 Movies
drwxr-xr-x    4 nightingale  staff       128 Sep  2  2019 FrontendProjects
drwxr-xr-x    5 nightingale  staff       160 Apr  5 07:57 CLionProjects
drwx------@   5 nightingale  staff       160 Dec 19  2018 Applications
...
```

Or same with dotted files

```
$ ls -lraS
drwx------@   5 nightingale  staff       160 Dec 19  2018 Applications
drwxr-xr-x    5 nightingale  staff       160 Mar 20  2019 .npm
drwxr-xr-x    5 nightingale  staff       160 May 21  2019 .matplotlib
```

pwd (print working directory)

pwd command shows absolute path to your current directory

```
$ pwd
/Users/nightingale
```

cd (change directory)

- **cd** command changes your current directory to another

```
$ pwd
/Users/nightingale
$ cd Documents/
$ pwd
/Users/nightingale/Documents
```

- cd command working with absolute and relative paths

```
$ pwd
/Users/nightingale
$ cd /Users/nightingale/Documents
$ pwd
/Users/nightingale/Documents
```

The current working directory is referred to with a single dot *(.)* and the parent directory is referred to with two dots *(..)*.

Manipulation

mkdir (make directory)

mkdir command creates empty directory in your current location

```
$ mkdir hello
$ ls
hello
$ mkdir world
$ ls
hello world
$ mkdir hello/students
$ tree
.
├── hello
│   └── students
└── world

3 directories, 0 files
```

touch

touch command creates file

```
$ touch hello/students/students.txt
$ tree
.
├── hello
│   └── students
│       └── students.txt
└── world

3 directories, 1 file
```

cp(copy)

cp command can be used to copy the contents in **<file_name>** to **<new_file_name>**

- ***cp*** **<file_name> <new_file_name>**

```
$ cp hello/students/students.txt world/copy_students.txt
$ tree
.
├── hello
│   └── students
│       └── students.txt
└── world
    └── copy_students.txt

3 directories, 2 files
```

- ***cp*** **<file_name> <directory_name>**

```
$ cp hello/students/students.txt world
$ tree
.
├── hello
│   └── students
│       └── students.txt
└── world
    └── students.txt
```

mv(move)

mv command is used to change the location of a file or directory as well as to change the name of a file or directory without changing its location

```
$ mv hello/students/students.txt hello/students/good_students.txt
$ tree
.
├── hello
│   └── students
│       └── good_students.txt
└── world
```

```
$ mv hello/students/good_students.txt world
$ tree
.
├── hello
│   └── students
└── world
    └── good_students.txt
```

```
mv hello/ world/
$ tree
.
└── world
    ├── good_students.txt
    └── hello
        └── students
```

rm(remove)

rm command attempts to remove the **non-directory** type files specified on the command line

```
$ tree
.
├── hello
│   └── students
│       └── students.txt
└── world
    └── good_students.txt

$ rm world/good_students.txt

$ tree
.
├── hello
│   └── students
│       └── students.txt
└── world
```

rm(remove)

-R(-r) argument attempts to remove the file hierarchy rooted in each file argument

```
$ tree
.
├── hello
│   └── students
│       └── students.txt
└── world

$ rm -r hello/
$ tree
.
└── world
```