NYC DATA SCIENCE ACADEMY

Introduction to Linux & Python

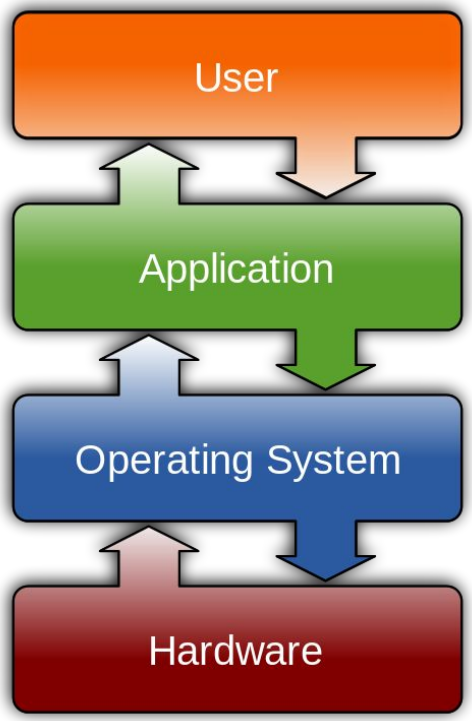
Data Science Bootcamp

OVERVIEW

- Operating Systems and Linux
- File System and File Operations
- Basic file commands
- Creating files
- Python Tutorial

What is an OS?

- An *operating system* (OS) is a program that provides a variety of services designed to facilitate your work on the machine, and to keep different users from interfering with one another.



Linux

- Linux is an open source distribution. In Linux, you can request OS services through a command-line interface (CLI, also called a “shell”).

```
 * Starting OpenBSD Secure Shell server sshd                                    [ OK ]
 * Starting MySQL database server mysqld                                        [ OK ]
Creating MySQL Database
Writing default config to: /home/ubuntu/.jupyter/jupyter_notebook_config.py


    __    _                    ______           ____   _ __
   / /   (_)___  __  ___  __   /_  __/___  ____  / / /__(_) /_____
  / /   / / __ \/ / / / |/_/    / / / __ \/ __ \/ / //_/ / __/ ___/
 / /___/ / / / / /_/ />  <     / / / /_/ / /_/ / / ,< / / /_(__  )
/_____/_/_/ /_/\__,_/_/|_|    /_/  \____/\____/_/_/|_/_/\__/____/


Ubuntu 16.04.5 LTS
Python 3.5.2
mysql  Ver 14.14 Distrib 5.7.23, for Linux (x86_64) using  EditLine wrapper

ubuntu@de6b545ba9c4:~$
```

Exercise 1 - Setting Up Linux Environment

- In this class, you will be using Linux from a Docker container.
- Git pull the course repo to your local machine:

  `https://github.com/nycdatasci/data_engineering`

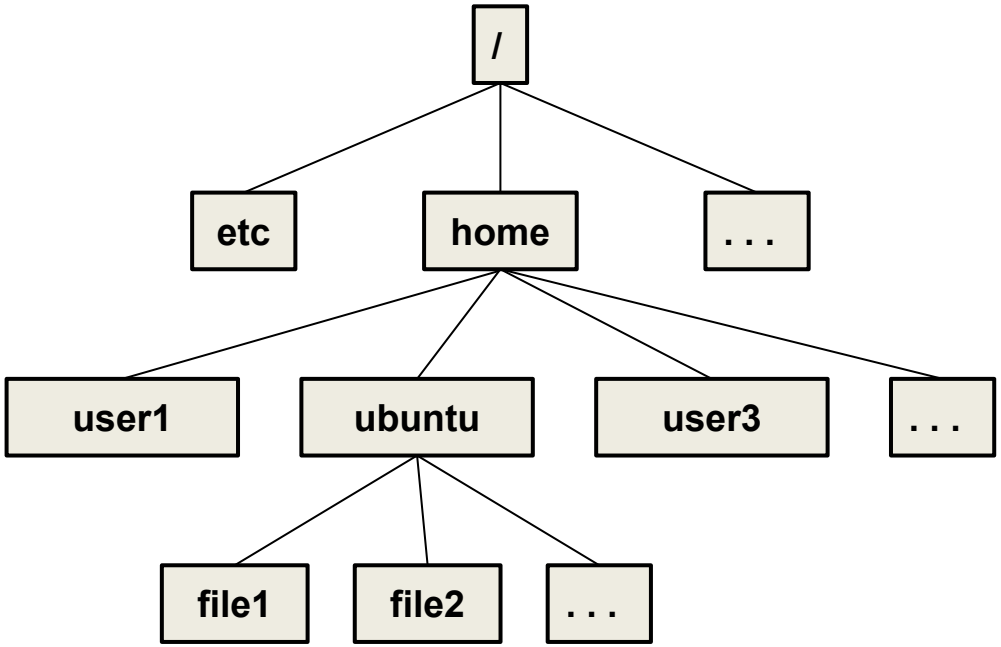
- Follow the instructions from **docker-quickstart.html** to setup the linux environment on your machine.
  - Make sure to mount the course repo to your docker container using `-v` flag.

OVERVIEW

- Operating Systems and Linux
- **File System and File Operations**
- Basic file commands
- Creating files
- Python Tutorial

The Filesystem

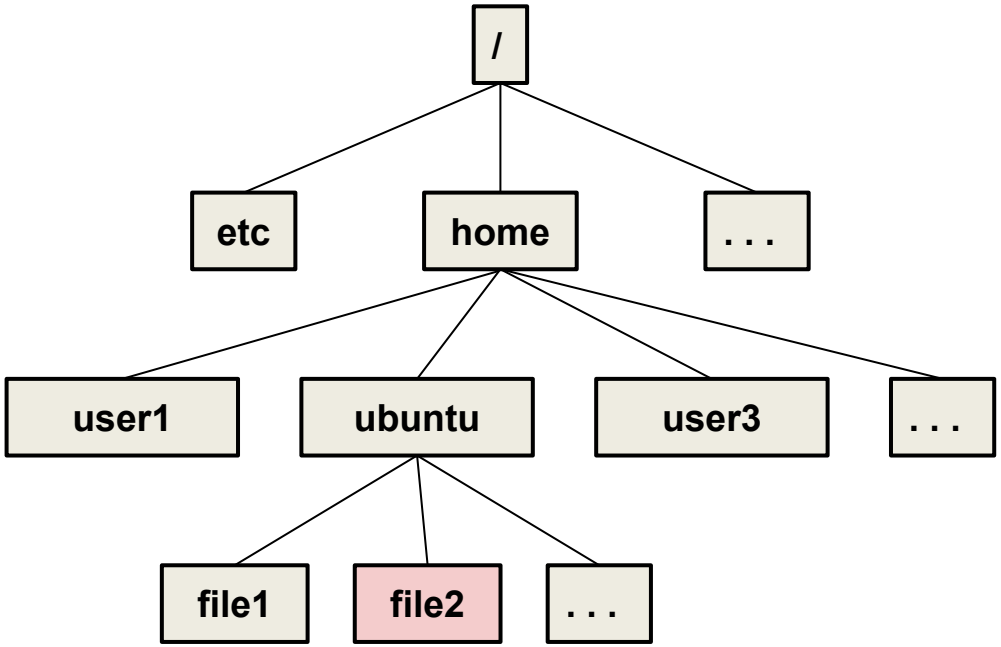
- Filesystem hierarchy, looks like an inverted tree.



- In Unix, the traditional word for folder is *directory*. We use “folder” and “directory” interchangeably.

Pathnames - Absolute Pathname

- To access a file we need to specify the path to a file. For example, the **absolute pathname** for the file2 below is `/home/ubuntu/file2`

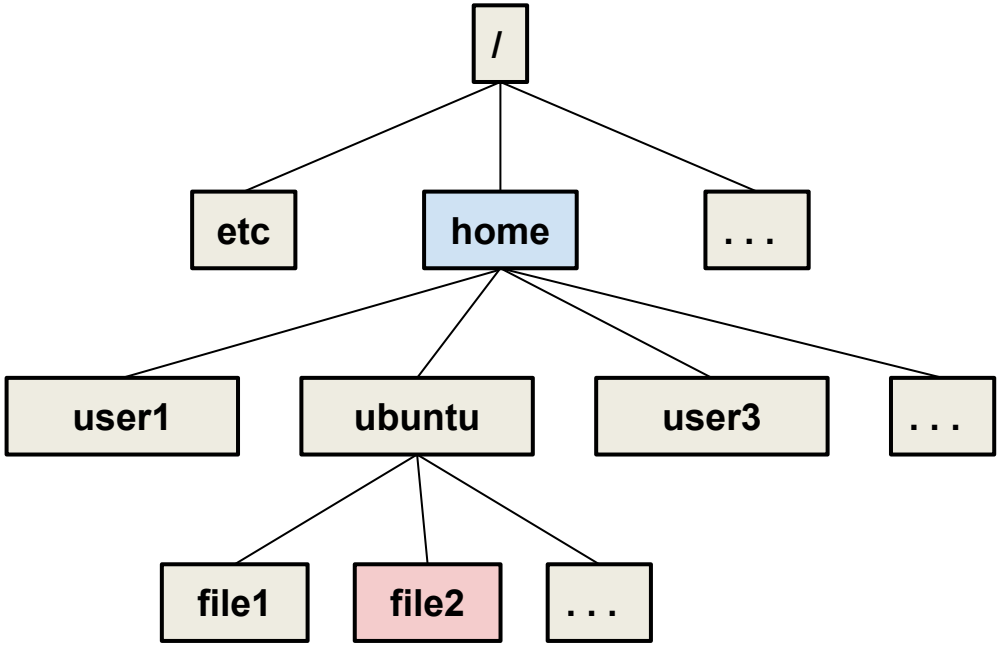


The working directory

- Special directories:
  - A user is always “in” a directory, called the *working directory*. To find your current working directory, use the command `pwd`
  - For each user, there is a directory with the same name as the user. It is user’s *home directory*, and is the working directory when you log on.

Pathnames - Relative Pathname

- Assume our current working directory is home, then to specify the path to file2, we can use the **relative pathname:** ./ubuntu/file2
- The symbol . represents the working directory.



OVERVIEW

- Operating Systems and Linux
- File System and File Operations
- **Basic file commands**
- Creating files
- Python Tutorial

ls: List files in directory

- The `ls` command lists the file at the given path. With no arguments, that defaults to the current working directory.
  - In exercise 1, you used "`ls`" to list the files in your home directory, and "`ls /etc`" to list files in the `/etc` directory.
- Finding documentation about Unix commands:
  - "`man ls`" produces a "man page." This is a standard Unix documentation format.
  - "`ls --help`" produces a similar documentation list.
  - Googling "`ls`" or "`linux ls`" produces many hits, though many are copies of the man page.

ls: List files in directory

- In Linux, commands often have special arguments, called *options*, introduced by a single or double dash.
- "`-a`" produces a file listing that includes "hidden files" - files whose name starts with a period. There are two special hidden files:
  - "." (a single period) is the current directory (the one being listed)
  - ".." (two periods) is its parent directory.
- "`-t`" sorts the file listing by modification date.
- Combine arguments by including several separately or by combining them after a single dash:

```
ls -a -t
```

or

```
ls -at
```

ls: List files in directory

- Argument "`-l`" produces a long form of the file listing, including file ownership, size, permissions, and other information.

```
test_user@ip-172-31-11-204:~$ ls -l /etc
total 808
drwxr-xr-x 3 root root    4096 May 22 10:55 acpi
-rw-r--r-- 1 root root    3028 May 22 10:54 adduser.conf
drwxr-xr-x 2 root root    4096 Jul  1 22:14 alternatives
drwxr-xr-x 3 root root    4096 May 22 10:54 apm
... ...
permissions  | links | user | group     | size | last modification time | name
```

- The permissions are broken into 4 sections. For example: `drwxr-xr--`
  1. `d`: '–' -> file, '`d`' -> directory, '`l`' -> link
  2. `rwx`: permissions for the owner
  3. `r-x`: permissions for members of the group owning the file
  4. `r--`: permissions for other users

  Where '`r`'-> read, '`w`'-> write/modify, '`x`'-> execute, '–'-> no permission

cd: Change the working directory

- The `cd` command changes the current working directory.
  - `cd` *pathname* changes the working directory to that pathname.
  - cd alone changed the working directory to your home directory.
  - Tilde (~) is an alias for your home directory. "`cd ~`" is the same as "`cd`". You can write "`cd ~/subdir`" to go to `subdir` in your home direct.

mkdir: Make a new directory

- `mkdir` creates a new, empty directory.
  - Make a new directory in your home directory:

```
$ cd ~
$ mkdir examples
$ ls examples
$
```

cp: Copy files

- The `cp` command supports making copies of any file. The syntax is:

```
cp [-r] source destination
```

  - Add the `-r` option if you are copying a folder.

- The destination can be a filename or an existing folder.
  - If an existing folder, `source` is copied into that folder.
  - If not an existing folder, a copy of `source` is made, with the `destination` as its name.

- *Be careful*, as the `cp` command can overwrite existing files.

cp: Copy files

- Here is an example of copying.

```
$ cd ~    # go to home directory
$ cp /etc/magic .
$ ls
examples magic

$ cp magic examples
```

- Check that both your home directory and the folder `examples` have the file `magic`.

rm: Delete files

- The `rm` command is for removal of files.
- **Note: once a file is deleted through the command-line, it is removed permanently.**

```
$ rm magic
```

rm: Delete files

- The `rm` command doesn't supports deleting directories. Try the command below:

```
$ rm examples
```

- The rmdir only supports deleting empty directories. Try the command below:

```
$ rmdir examples
```

- It's more common to use: `rm -r`. But this removes files **permanently.**

```
$ rm -r examples
```

cat: Dump file contents to stdout

- The simplest command to view the file contents is `cat` (short for 'concatenate'). It prints the entire file to the console ("standard output").

```
$ cp /etc/hosts nethosts
$ cat nethosts   # We could cat /etc/hosts as well
127.0.0.1   localhost localhost.localdomain localhost4
localhost4.localdomain4
::1         localhost6 localhost6.localdomain6
```

- `cat` can take multiple files as arguments. It dumps them all to standard output, with no line breaks in between.
- Be careful with `cat`. If the file is very large, it could print for a long time. To stop it, type ^C (ctrl-C).

less: View files one screenful at a time

- We can use the `less` command to view longer files without writing everything out to the screen at once. Less will only print one screenful of data to the terminal.

```
$ less /etc/services
```

  - Scroll up/down one line at a time with arrow keys.
  - Use the space bar to scroll through a screen at a time.
  - Use the / key to search for a term: `/apple`
  - Press h to see a quick list of other options.
  - Press q to exit less.

Exercise 2 - Practice file commands

- Your home directories are empty. The first command below get a file `iris.csv` online by `wget`. Then try the command below to work with directories and files:

```
wget https://s3.amazonaws.com/graderdata/iris.csv
less iris.csv
mkdir flowers
cp iris.csv flowers
cp -r flowers irises
ls
cd flowers
rm iris.csv
cd ..
```

OVERVIEW

- Operating Systems and Linux
  - File System and File Operations
  - Basic file commands
- Creating files
  - Python Tutorial

Editing text files

- We will be working with “pure text files” - files with characters but no formatting information.
- We will edit them using vi, an editor popular with programmers.
- You start vi by typing the following command after the linux $ prompt:
  - `$ vi` *file*
- *file* will be opened if it exists, created if not.

vi - Basic operation

- vi is *modal*, meaning you are always in one of two modes:
  - *Insert mode*: Characters you type go into the file. (But careful: the file is not saved until you request it.)
  - *Command mode*: Characters are interpreted as commands, e.g. the character k means “move the cursor up one line”.
- The next two slides have a cheat sheet for basic vi commands.

vi - Basic operation

- Remember vi is modal: you are either in *insert mode* (what you type goes in the file) or *command mode* (what you type is interpreted as a command to the editor).

Enter/exit insert mode	Command mode
a - insert after cursor	x - delete one character
i - insert before cursor	dw - delete one word
o - insert a new line below	dd - delete the line
O - insert a new line above	D - delete the rest of the line
ESC - exit insert mode	u - undo the last action

vi - Basic operation

- Unlike the letter commands, the arrow keys work in either edit or command mode.

Moving cursor	Saving
↑ or k - up one line	ZZ - save and exit
↓ or j - down one line	:wq - save and exit
← or h - backward one character	:w - save without exiting
→ or l - forward one character	:q! - exit without saving

Exercise 3: Linux Commands and vi

- Open file input.txt: `$ vi hello.txt`
- Practice using the vi commands on the previous slide. Remember:
  - You can exit without saving changes by typing `:q!`

OVERVIEW

- Operating Systems and Linux
- File System and File Operations
- Basic file commands
- Creating files
- **Python Tutorial**

Exercise 3: Python Tutorial

- The Python tutorial provides a quick introduction about how to use python.
- Start jupyter notebook by executing `jupyter notebook` from your Linux container and open the jupyter URL from your browser window.
- Follow the tutorial.