THE BIOROBOTICS INSTITUTE

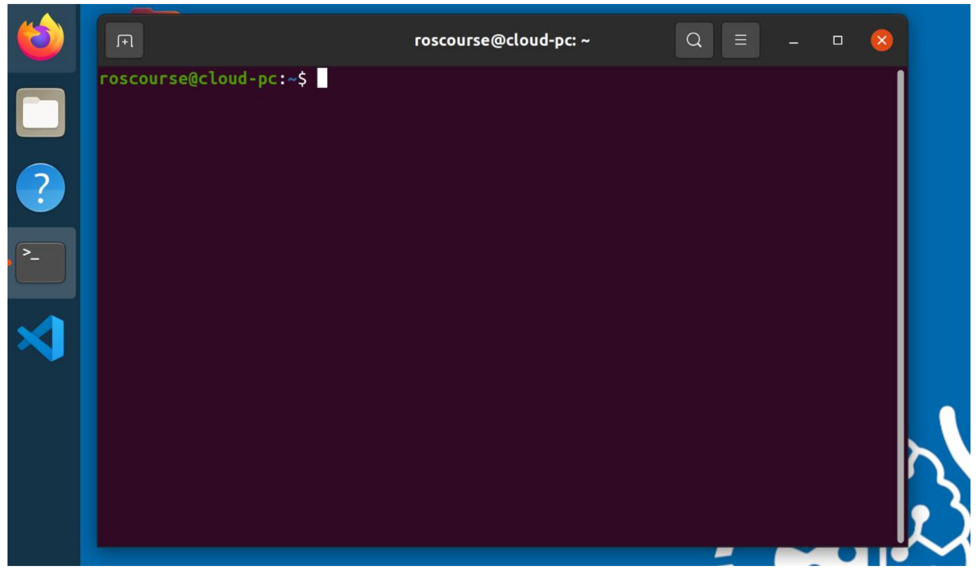
Sant'Anna
School of Advanced Studies – Pisa

Introduction to Ubuntu Command Line

Egidio Falotico

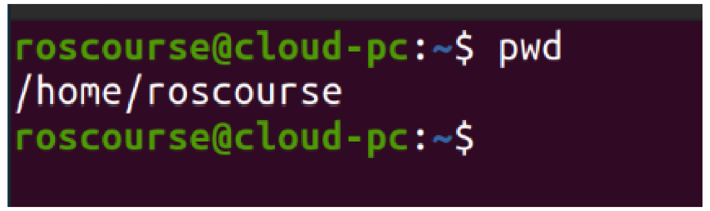
Open a terminal

Press *Ctrl+Alt+t* to open a new terminal (or just click on the black icon in the sidebar



```
roscourse@cloud-pc:~$ 
```

The command *pwd* (print working directory) returns the current directory

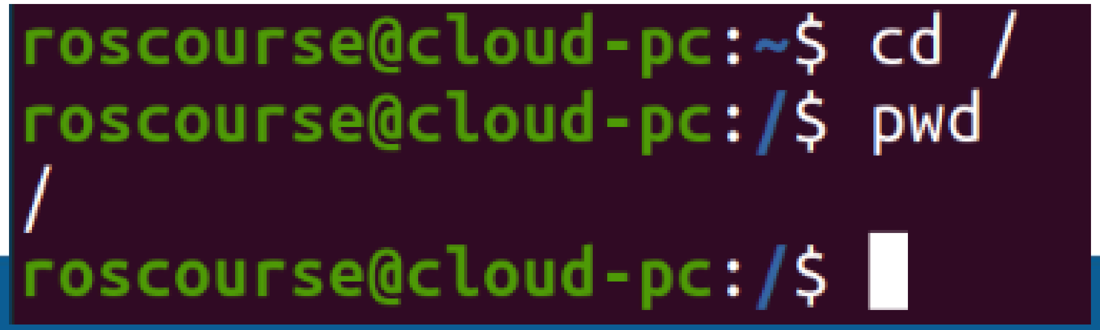


```
roscourse@cloud-pc:~$ pwd
/home/roscourse
roscourse@cloud-pc:~$
```

Change directory

One important concept to understand is that the shell has a notion of a **default location** in which any file operations will take place. This is its **working directory**. If you try to create new files or directories, view existing files, or even delete them, the shell will assume you are looking for them in the current working directory unless you take steps to specify otherwise

You can change the working directory using the cd command, an abbreviation for ‘change directory’



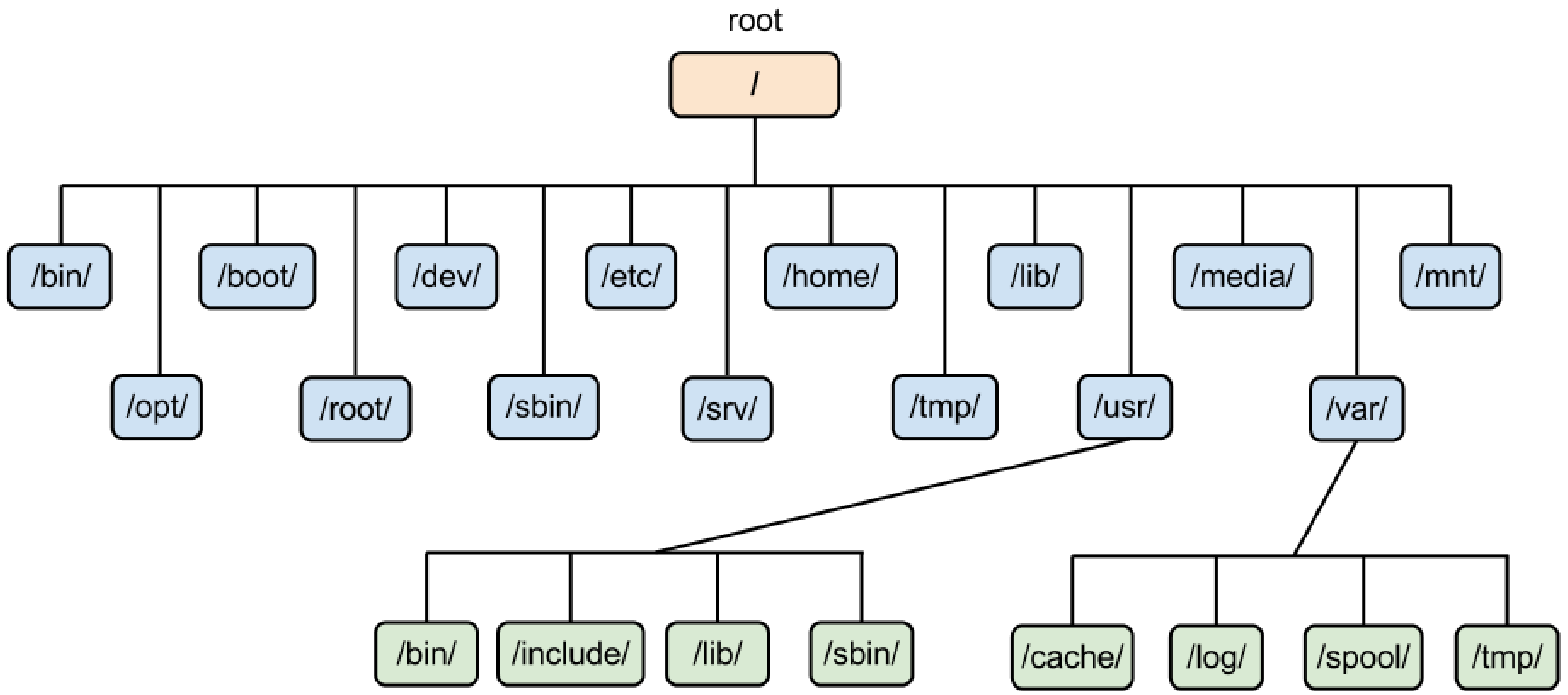
```
roscourse@cloud-pc:~$ cd /
roscourse@cloud-pc:/$ pwd
/
roscourse@cloud-pc:/$ 
```

Change directory

In Windows each drive having its own letter, with your main hard drive typically being "C:".

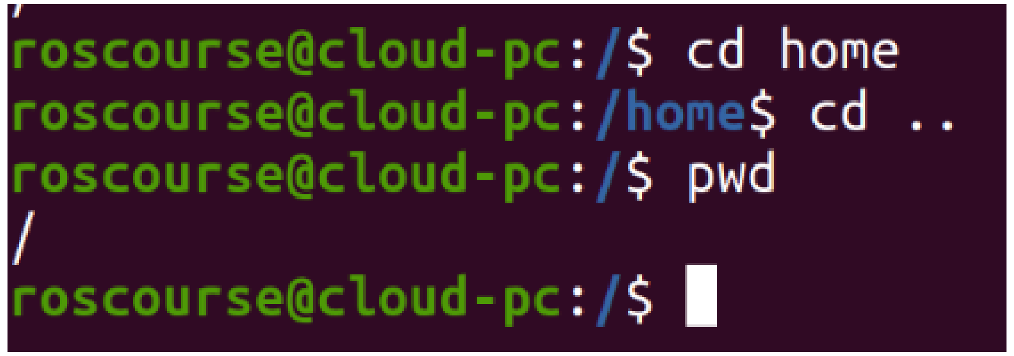
Unix-like systems have a single unified file system, and individual drives can be attached ("mounted") to whatever location in the file system makes most sense.

The "/" directory, often referred to as the root directory, is the base of that unified file system. From there everything else branches out to form a tree of directories and subdirectories.



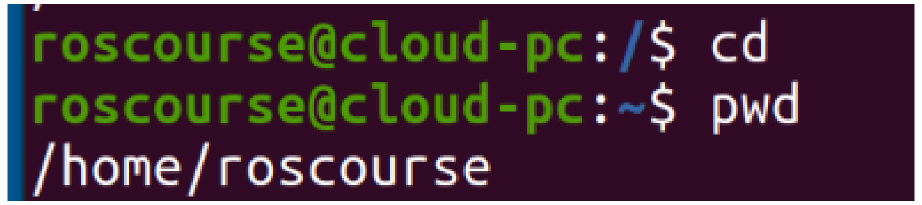
Change directory (go to parent one)

To go up to the parent directory, use the special syntax of two dots (..) when changing directory (note the space between cd and ..). We first move to home and then back using *cd ..*



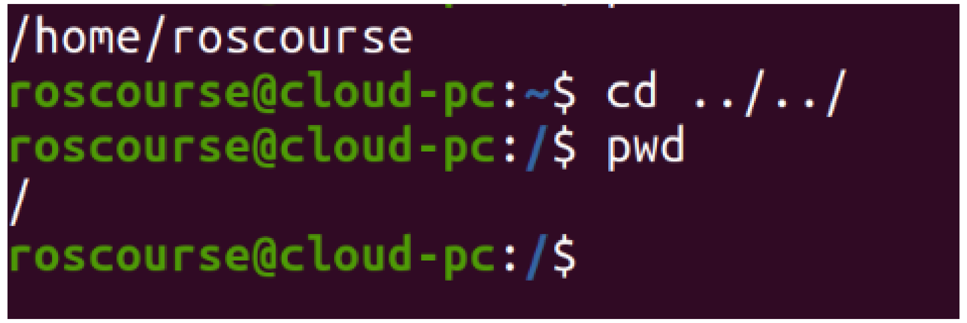
```
roscourse@cloud-pc:/$ cd home
roscourse@cloud-pc:/home$ cd ..
roscourse@cloud-pc:/$ pwd
/
roscourse@cloud-pc:/$ █
```

Typing cd on its own is a quick shortcut to get back to your home directory:



```
roscourse@cloud-pc:/$ cd
roscourse@cloud-pc:~$ pwd
/home/roscourse
```

You can also use .. more than once if you have to move up through multiple levels of parent directories:



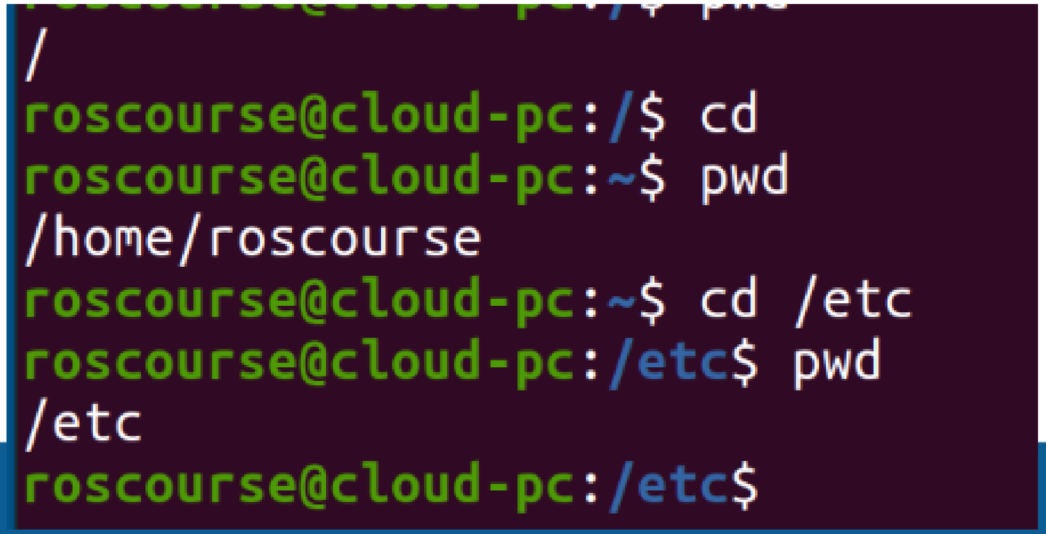
```
/home/roscourse
roscourse@cloud-pc:~$ cd ../../
roscourse@cloud-pc:/$ pwd
/
roscourse@cloud-pc:/$
```

Relative and Absolute path

Most of the examples seen so far use relative paths. That is, the place you end up at **depends on your current working directory**.

You can also use absolute path. No matter what your current working directory is, they will have the same effect.

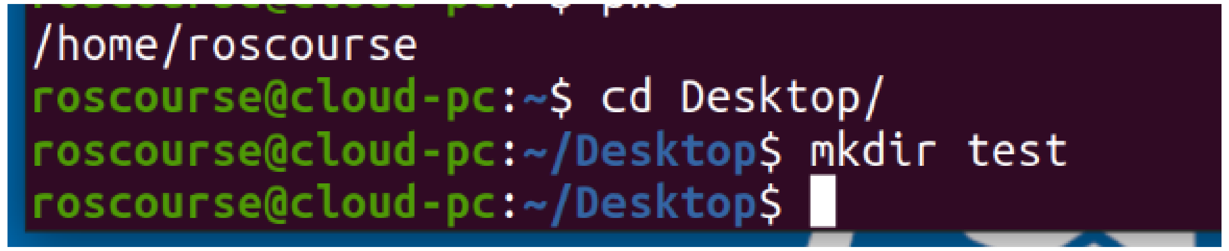
- The first is when you run cd on its own to go straight to your home directory.
- The second is when you used cd / to switch to the root directory. In fact any path that starts with a forward slash (/) is an absolute path. ***You can think of it as saying "switch to the root directory, then follow the route from there"***



```
roscourse@cloud-pc:/$ pwd
/
roscourse@cloud-pc:/$ cd
roscourse@cloud-pc:~$ pwd
/home/roscourse
roscourse@cloud-pc:~$ cd /etc
roscourse@cloud-pc:/etc$ pwd
/etc
roscourse@cloud-pc:/etc$
```

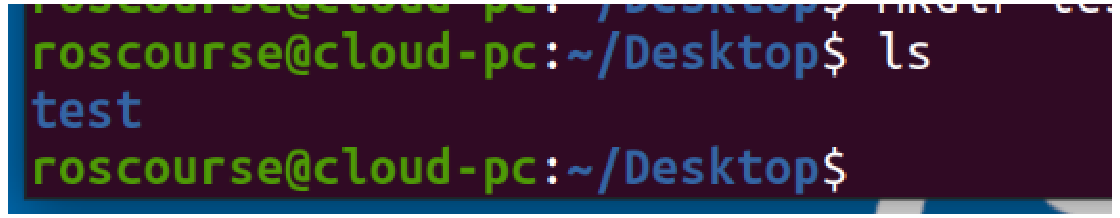
Creating folders and file

To create a new folder you can use *mkdir* is short for ‘make directory’



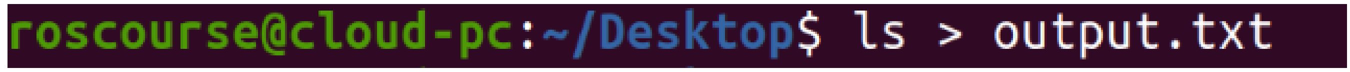
```
/home/roscourse
roscourse@cloud-pc:~$ cd Desktop/
roscourse@cloud-pc:~/Desktop$ mkdir test
roscourse@cloud-pc:~/Desktop$ 
```

Let’s take a look inside the current directory with the *ls* (list) command:



```
roscourse@cloud-pc:~/Desktop$ ls
test
roscourse@cloud-pc:~/Desktop$
```

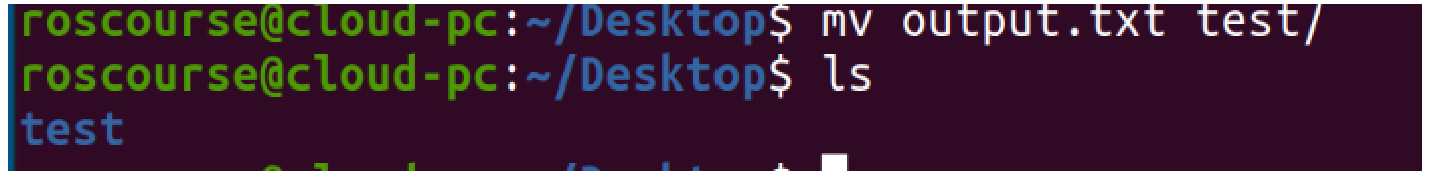
Suppose we wanted to capture the output of that command as a text file that we can look at or manipulate further. All we need to do is to add the greater-than character (”>”) to the end of our command line, followed by the name of the file to write to:



```
roscourse@cloud-pc:~/Desktop$ ls > output.txt
```

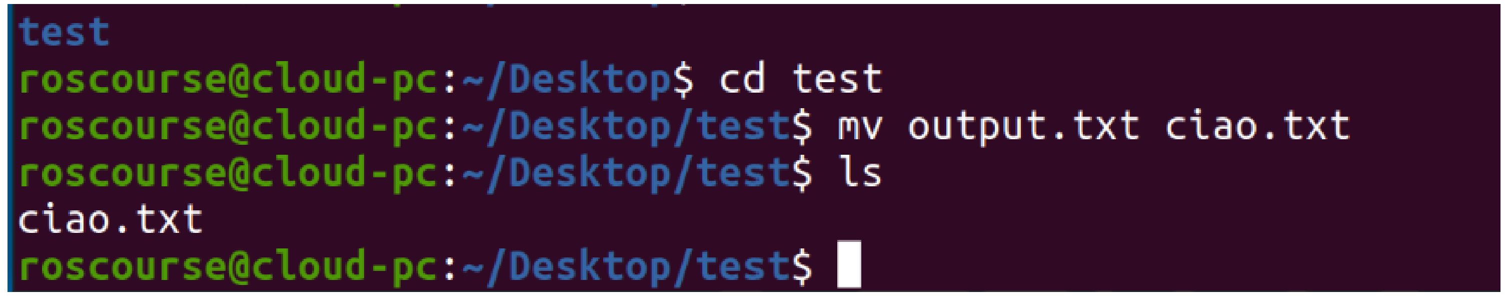
Moving and removing files

Using command *mv* you can move a file into a folder



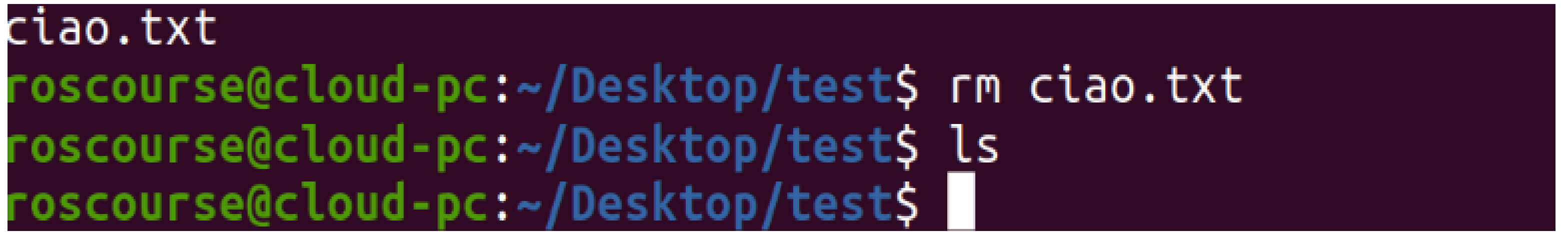
```
roscourse@cloud-pc:~/Desktop$ mv output.txt test/
roscourse@cloud-pc:~/Desktop$ ls
test
```

You can also change the name of the file



```
test
roscourse@cloud-pc:~/Desktop$ cd test
roscourse@cloud-pc:~/Desktop/test$ mv output.txt ciao.txt
roscourse@cloud-pc:~/Desktop/test$ ls
ciao.txt
roscourse@cloud-pc:~/Desktop/test$ 
```

To remove file you can use rm command



```
ciao.txt
roscourse@cloud-pc:~/Desktop/test$ rm ciao.txt
roscourse@cloud-pc:~/Desktop/test$ ls
roscourse@cloud-pc:~/Desktop/test$ 
```