CEG2722: Data Analysis II

Command Line Data Processing

- Lecture 2 : Getting Started -

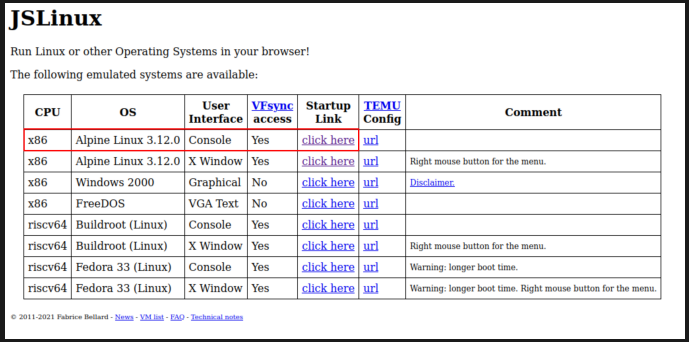
Achraf Koulali

Geospatial Engineering

November 22, 2021

Getting started

- We will be using JSLinux for practice during Lectures.
  - minimal Linux with a terminal from your browser.



JSLinux

Run Linux or other Operating Systems in your browser!

The following emulated systems are available:

CPU	OS	User Interface	VFsync access	Startup Link	TEMU Config	Comment
x86	Alpine Linux 3.12.0	Console	Yes	click here	url	
x86	Alpine Linux 3.12.0	X Window	Yes	click here	url	Right mouse button for the menu.
x86	Windows 2000	Graphical	No	click here	url	Disclaimer.
x86	FreeDOS	VGA Text	No	click here	url	
riscv64	Buildroot (Linux)	Console	Yes	click here	url	
riscv64	Buildroot (Linux)	X Window	Yes	click here	url	Right mouse button for the menu.
riscv64	Fedora 33 (Linux)	Console	Yes	click here	url	Warning: longer boot time.
riscv64	Fedora 33 (Linux)	X Window	Yes	click here	url	Warning: longer boot time. Right mouse button for the menu.

Figure 1: JS Linux terminal emulator

- For practicals, we will be using the University Linux Server.
  - Instructions to access the server can be found in Canvas

The shell command line

- Interprets and enacts typed commands in real time some commands may run on other computers and/or at later time, or in background.

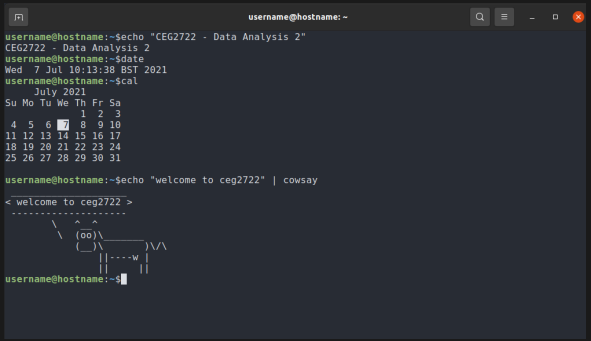


Figure 2: shell prompt: user@machine

The shell command line.

- Almost all Linux distributions provide the "Bourne again shell" (bash).
- There are many shell version such as: csh, tcsh, zsh...
- We will focus on bash

Executing a Command-line Tool

Now that you have a basic understanding of the environment, it is time that you try out some commands.

Type the following in your JSLinux terminal (without the dollar sign) and press Enter

```
$ pwd
/home/user
$ ls
bench.py    hello.c     hello.js    readme.txt
```

Navigating the filesystem using `cd` & `ls`

The Linux filesystem appears as a hierarchy of directories(folders):

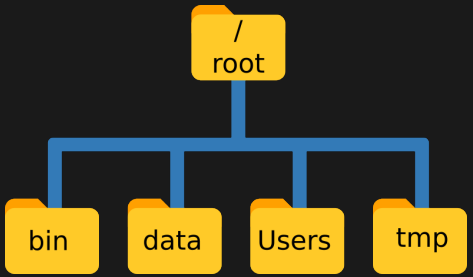


Figure 3: Linux Filesystem

Navigating the filesystem using `cd` & `ls`

- Use the command `cd` (change directory)

```
# this is a comment
# change into dir
$ cd Desktop
# this command prints the the working directory
$ pwd
/home/jake/Desktop
```

Navigating the filesystem using `cd` & `ls`

- The command `ls` to list the files in a directory.

```
$ cd book/ch02/
$ cd data
$ pwd
/home/user/book/ch02/data
$ cd ..
$ pwd
/home/user/book/ch02/
```

- Values that come after the command are called command-line arguments or options.
- The two dots refer to the parent directory.

Command line options

- Linux commands may take one or more arguments.
- Traditionally, arguments beginning “-“ or “–“ are regarded as option flags.

```
$ ls --help
# the option --help displays the help of the command ls
```

Command line options

- Another example of command line

```
# this displays the first 3 lines of the file movies.txt
$ head -n 3 data/movies.txt
Matrix
Star Wars
Home Alone
```

Command line tools

- We use the term "command line" as anything that can be executed from the terminal.
- Each command-line tool is one of the following five types:
  - A binary executable.
  - A shell builtin.
  - **An interpreted script**.
  - **A shell function.**
  - **An alias.**

Command line tools

Interpreterd script

```
def factorial(x):
    result = 1
    for i in xrange(2, x + 1):
        result *= i
    return result

if __name__ == "__main__":
    import sys
    x = int(sys.argv[1])
    print factorial(x)
```

```
$ python fac.py 5
120
```

Command line tools

Shell script

```
$ fac() { (echo 1; seq $1) | paste -s -d\* - | bc; }
$ fac 5
120
```

Command line tools

Alias

```
$ alias mymachine='hostname -f'
$ mymachine
Latitude-5400
```

Command line tools

You can find out the type of a command-line tool with ‘type‘ (which is itself a shell builtin):

```
$ type -a pwd
pwd is a shell builtin
pwd is /bin/pwd
$ type -a cd
cd is a shell builtin
```

Test your knowlegde

http://poll-maker.com/QL52VSN94

Combining command line tools

- The most important way of combining command-line tools is through a pipe("|")

Example: generate a sequence of numbers from 1 to 5

```
$ seq 5
1
2
3
4
5
```

Combining command line tools

We can pipe("|") the ouput of the first command to a second tool, which can be used to filter lines.

Example: how many numbers between 1 and 100 that contain a three

```
$ seq 100 | grep 3 | wc -l
19
```

Redirecting input and output

- The default output of command-line tools is to the terminal.
- We can save outputs to a file: output redirection:

```
$ hostname > mymachine.txt
# here we redirect the name of the
# machine (given by hostname) to the file mymachine.txt
```

Redirecting input and output

- We can also append the output to a file with >>:

```
$ echo "Hello, ceg2722!"
$ echo -n "Hello" > hello-world
$ echo " World" >> hello-world
# -n specifies that echo should not output a trailing newline.
$ cat hello-world
?
```

- We can use the command `cat` to read a file and print it in the terminal.

Working with files

- Data analysis implies using data, and data is often stored in files.
- In this section we introduce how to create, move, copy, rename, and delete files and directories.

Working with files

- To move a file to a different directory you can use:

```
$ mv hello.txt ~/book/ch02/data/
```

- You can also rename files with mv:

```
$ cd data
$ mv hello.txt bye.txt
```

Wording with files

In case you no longer need a file, you delete it with `rm` (Warning!!)

```
$ rm bye.txt
# to remove a directory, specify the -r option,
# which stands for recursive:
$ rm -r book/ch02/data/old
```

- A good practice is to define an alias for the rm command: `alias rm='rm -i'`

Working with files

- In case you want to copy a file, use cp :

```
$ cp server.log server.log.bak
```

Working with files

- To create a directory

```
$ cd data
$ mkdir logs
```

Using Help / documentation

- The most important command to get help is perhaps `man` == manual.

```
$ man cat | head -n 20
?
# you can also use the argument --help to display the usage directions
$ ls --help
```

Test your knowledge

Quiz 2.1: You can use your terminal to verify possible answers.

Test your knowledge

- To change to the Desktop directory:
  1. `cd ~/Desktop/`
  2. `ls ~/Desktop/`
  3. `cd /home/user/Desktop/`

Test your knowledge

- Create a directory called myceg2722:
  1. `mkdir myceg2722`
  2. `wc myceg2722`
  3. `cd myceg2722`

Test your knowledge

- Change the directory to myceg2722
  1. `cd myceg2722`
  2. `echo myceg2722`
  3. `dc myceg2722`

Test your knowledge

- Print the current directory
  1. `pwd`
  2. `ls ./`
  3. `echo ./`

Test your knowledge

- Create a file (myfile.txt) and add the sentence “Hello, CEG2722” to it
  1. `touch myfile.txt; echo "Hello, CEG2722" > myfile.txt`
  2. `ls myfile.txt; pwd`
  3. `cat myfile.txt; echo "Hello, CEG2722" > myfile.txt`

Test your knowledge

- Move the file `myfile.txt` to the the directory Desktop
  1. `cd myfile.txt`
  2. `cd myceg2722; ls ~/Desktop/`
  3. `mv myfile.txt ~/Desktop/`

Test your knowledge

- Change directory to `Desktop` then remove the directory `myceg2722`
  1. `cd ~/Desktop/; rm myceg2722`
  2. `cd ~/Desktop/; rm -r myceg2722`
  3. `cd ~/Desktop/; mr myceg2722`

Summary

- We introduced the Linux filesystem.
- We introduced some basic Linux commands for navigating the filesystem.
- Homework: Using the JSLinux terminal, repeat the examples of this session.