UNIX notes

**Tutorial**

* <http://www.ee.surrey.ac.uk/Teaching/Unix/> for instance – given by Fernando
* **SSH tutorial**: <https://linuxacademy.com/blog/linux/ssh-and-scp-howto-tips-tricks/>
* **Manipulating csv files from command line** <http://bconnelly.net/working-with-csvs-on-the-command-line/>

**Manipulating data with terminal**

**Extract first lines**

head -n 13 input.csv > first13.csv

**Extract columns 2, 4,5,6 from input.csv**

cut -d , -f 2,4-6 input.csv

* -d , : separated by coma
* -f: what columns to extract

**SSH**

**Log in server**

* Run UNIX terminal – maybe Git Bash will work
* Ssh [flegros@vgchead.poly.edu](mailto:flegros@vgchead.poly.edu)  Log into server
* Password: type password here
* Then navigate in the serve with cd and ls commands

**Scp**

* Copy of taxi.csv from Fernando directory to mine

scp flegros@vgchead.poly.edu:/san\_share/fchirigati/taxi.csv flegros@vgchead.poly.edu:/san\_share/flegros

* Copy from remote to local: type IN LOCAL SHELL (no ssh login)

scp flegros@vgchead.poly.edu:/san\_share/fchirigati/taxi.csv /local/dir

* Multiple files from local to remote
  + scp foo.txt bar.txt [your\_username@remotehost.edu](mailto:your_username@remotehost.edu):~
  + scp {foo,bar}.txt your\_username@remotehost.edu:~
* Multiple files from remote to local
  + scp your\_username@remote.edu:/some/remote/directory/\{a,b,c\} ./

**Python live interpreter**

* Type: python
* To exit: exit()
* execfile('helloworld.py') = running script on shell
* Without login in python: python myscript.py

**Setting a python environment on remote machine**

* [Install locally virtualenv.](http://docs.python-guide.org/en/latest/dev/virtualenvs/) <https://virtualenv.pypa.io/en/stable/installation.html>

$ python virtualenv.py my\_new\_env

$ . my\_new\_env/bin/activate

(my\_new\_env)$ pip install package\_name

**Using Remi’s python 2.7**

* Connect to vidaserver1 : ssh [flegros@vidaserver1.poly.edu](mailto:flegros@vidaserver1.poly.edu)
* set path to remi’s python 2.7 folder
  + export PATH=/local\_scratch/remirampin/python-2.7/bin:$PATH

Editing

* vi taxiproc.py
* Navigate with arrow
* i: insert text, exist with Esc
* ZZ: save and exit
* :w → save
* :q! → exit without saving

Opening various programs

* sublime-text
* spyder

**Remote processes while logged out of ssh : screen**

* <https://www.rackaid.com/blog/linux-screen-tutorial-and-how-to>
* screen, then screen -S session name
* Ctrl-a a : new window
* Ctrl-a n : next window
* Ctrl-a p : previous window
* Ctrl-a d : detach from window
* screen -r sessionname : reattach a session
* exit:stop screen or Ctrl-a k

**Installing .jar files**

* Java is needed
* Terminal : java -jar packagename.jar

**Run a file**

* chmod +x filename
* ./ filename

**.so file – Shared Object**

* <http://www.file-extensions.org/so-file-extension>
* Other name = ELF, Executable and Linkable Format
* Standard type file for object code, shared libraries…
* Used in Unix

**Running a program from a script – C script for instance**

1. Preprocessing
2. Compilation proper : .c → .o. High to low level script
3. Linking : all o files linked together to make a program
   * static libraries placed inside the program
   * shared libraries referenced in the program
4. Loading : scanned for references to shared libraries

**Tutorials on libraries**

* <http://tldp.org/HOWTO/Program-Library-HOWTO/shared-libraries.html>

**How to install a library with no administrator rights ?**

* 1. LD\_LIBRARY\_PATH=.:$LD\_LIBRARY\_PATH **→ OUT**
     + Will look for libraries in the directory specified by path before the usual places
     + (eventuellement export)
     + Not efficient
  2. rpath ?? → Nice for a single program
  3. gcc -L  → **OUT** 
     + gcc SaTScanBatch -L /home/vgc/flegros/satscan/software/libraries -l stdc++ -o SaTScanBatch64\_15
     + Error : Cannot create .eh\_frame\_hdr section and others…
  4. Create symbolic link **→ OUT**
  5. LD\_RUN\_PATH <https://www.eyrie.org/~eagle/notes/rpath.html> **→ OUT**
  6. gcc -Wl,-rpath,/path/lib/file <https://www.eyrie.org/~eagle/notes/rpath.html> **→ OUT**
  7. Patchelf tool : modify dynamic linker and RPATH of ELF executables
     + **OUT** because installingp atchelf requires admin rights

**Find what libraries are used by a executable file**

* ldd /path/to/file

→ ldd home/satscan/SaTScanBatch

* Path to libstdc++.so.6 never changed whatever the operation

**Fabio’s tricks on libraries**

1. scl enable devtoolset-3 bash **OUT → 6.0.14**
   * Includes a stock of libraries
2. -Llibrarypath -l’libraryname’ **OUT → Not recognized path, although directory path or library path exists**
   * ‘lib’ and ‘.so’ should be stripped from library name
   * ‘libkevin.so’ → ‘kevin’

**Find files**

* find . -name '\*.pl'
* find and delete
  + find . -name "FILE-TO-FIND" -exec rm -rf {} \;

**Count lines**

wc -l <filename>

**Using Boost**

* Configuration : <http://www.boost.org/doc/libs/1_61_0/doc/html/quickbook/install.html>

**Directories**

**Create hierarchy of directories**

* mkdir -p dir1/dir2/dir3