Introduction to Linux

Francisco Salavert Torres
March 6th, 2017

GDA
International Course on
Genomic Data Analysis

PRINCIPE FELIPE
CENTRO DE INVESTIGACION

What is GNU/Linux?

- GNU/Linux to simplify Linux, is a “free” Operating System (OS).
- By Operating System, we mean the suite of programs and libraries which make our computer work.
- It is a stable, multi-user, multi-tasking system for servers, desktops and laptops.
- Other OS:
  - Microsoft windows (10, 8, 7, Vista, XP...)
  - Apple Mac OS X
  - Sun Solaris
  - ...

GNU/Linux history

- GNU project started by Richard Stallman in 1984 to create a “free” operating system.

- Linux kernel (base of the system) created by Linus Torvalds in 1991.

Linux distributions

- A Linux distribution is an operating system built on top of the Linux kernel that usually includes a very large collection of free and open-source software of all sorts.
- Linux distributions have taken a wide variety of forms, from fully featured desktop, server, laptop, netbook, mobile phone, and tablet operating systems as well as minimal environments.

Why do we use Linux?

- Many useful tools for bioinformatics are mainly developed for UNIX-based systems:
  - Programming languages: Perl, Python, R, C, Java, Bash…
  - Software for data analysis and manipulation:
    - RNA aligners, quantification tools, statistics analysis tools…

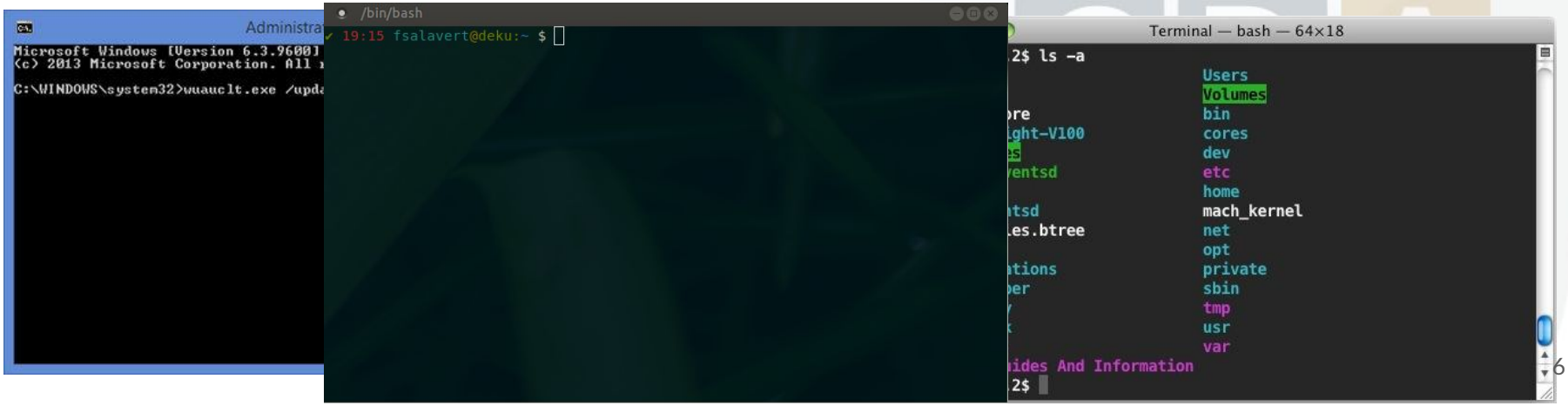
- Linux is more…
  - Scalable: it's quick and easy adding new libraries, tools and modules.
  - Flexible: we can easily adapt the existing code to our own requirements.
  - More Secure than other OS.

- Price, Linux is free

GUI and CLI

- Graphical user interface (GUI)

- Command line interface (CLI)

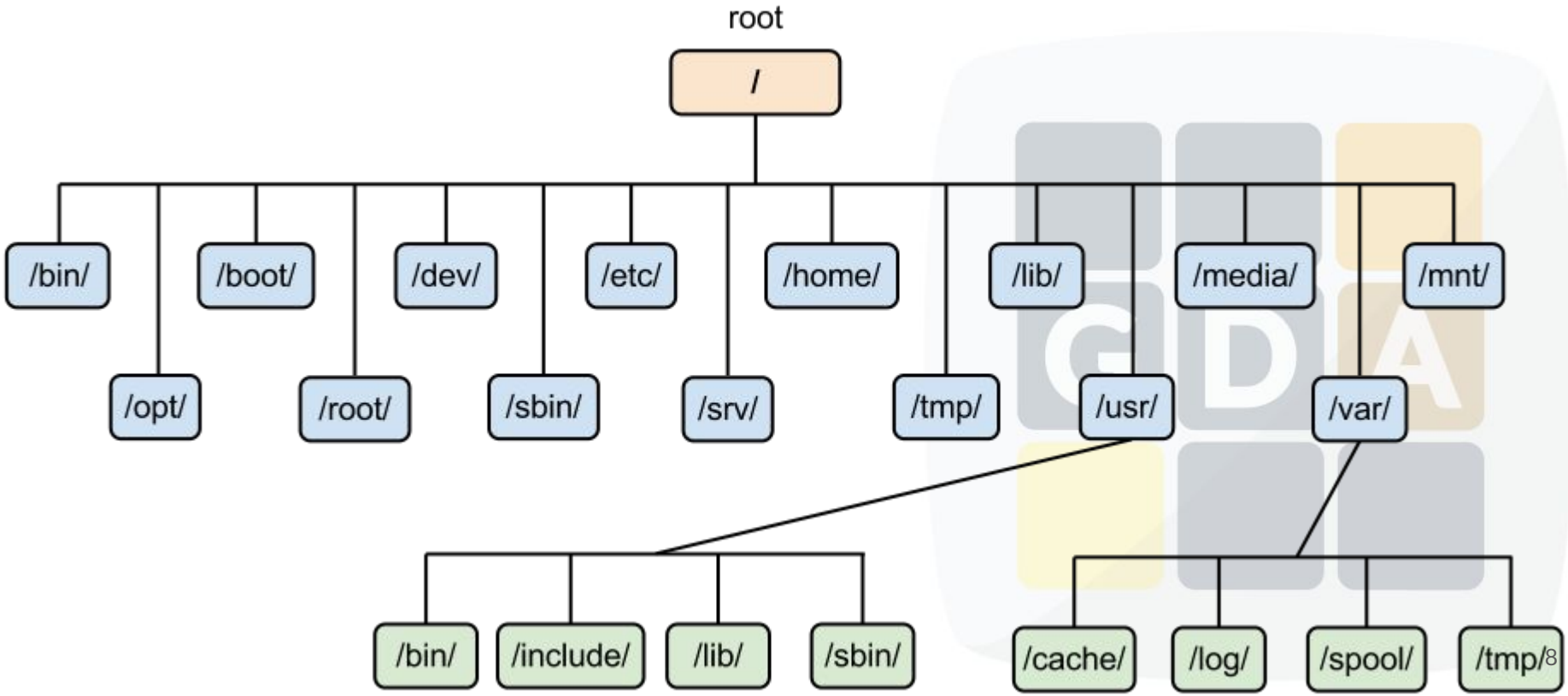


Why do we use Terminal (CLI)?

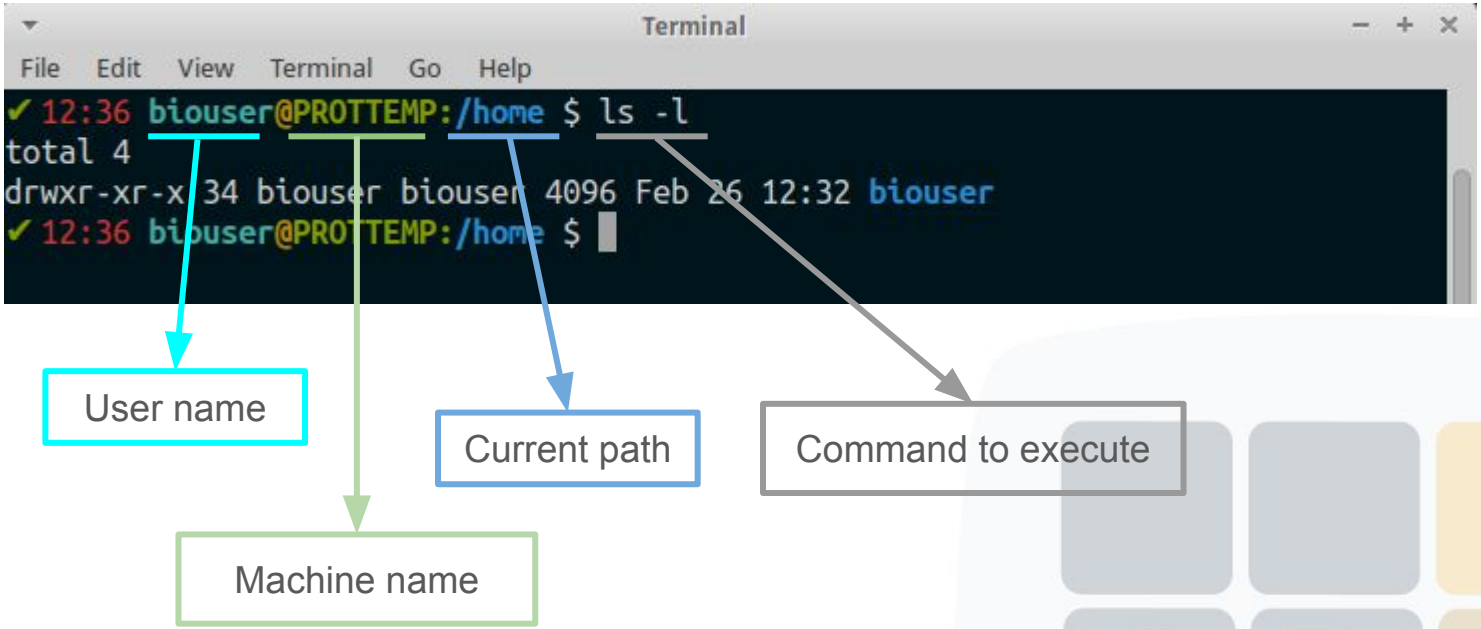
- Many bioinformatic tools are command line programs.
  - bwa
  - samtools
  - ....
- So there is no other choice to learn how to run those programs using the command-line.
- Some tools can not be run on a workstation computer due to the hardware requirements, so we need a cluster.
- To run the tools on a cluster we use the Terminal, because there is no Graphical interface installed.

The Linux file System

- It is organized as a directory tree.
- All starts in / (root)
- The paths are built like this /home/user1/Desktop



Basic command line usage



- The **prompt** is a text message at start of the command line, **$** indicates the end.
- Commands usually contains arguments, some can optional others can be mandatory.
  - ls -l
  - ls path/to/a/directory
- Arguments modifies the default behaviour of the command.

Basic commands

pwd : **p**rint current **w**orking **d**irectory

```
/bin/bash
✔ 15:06 fsalavert@deku:~ $ pwd
/home/fsalavert
✔ 15:06 fsalavert@deku:~ $
```

ls : **l**i**s**t information about the FILEs

- **ls -l**
- **ls -a**
- **ls -la**
- **ls -lah**

```
/bin/bash
✔ 15:09 fsalavert@deku:/ $ ls
bin    dev   initrd.img  lib64       mnt   root  srv  usr
boot   etc   lib         lost+found  opt   run   sys  var
cdrom  home  lib32       media       proc  sbin  tmp  vmlinuz
✔ 15:09 fsalavert@deku:/ $
```

Basic commands

ls -alh

- -a : show all files, including entries starting with . (hidden files).
- -l : list mode show full info (ownership, privileges, size, creation/edition date...)
- -h : human readable, show file sizes in human readable format.

```
/bin/bash
✔ 15:11 fsalavert@deku:/ $ ls -l
total 124
drwxr-xr-x   2 root root 12288 feb 23 11:29 bin
drwxr-xr-x   3 root root  4096 nov 19 12:02 boot
drwxr-xr-x   2 root root  4096 abr  7  2015 cdrom
drwxr-xr-x  18 root root  4580 feb 26 10:09 dev
drwxr-xr-x 157 root root 12288 feb 25 11:40 etc
drwxr-xr-x   6 root root  4096 oct 23  2014 home
lrwxrwxrwx   1 root root    33 abr  7  2015 initrd.img -> boot/initrd.img-3.19.0-10-generic
drwxr-xr-x  25 root root  4096 feb 17 11:14 lib
drwxr-xr-x   2 root root  4096 feb 17 11:14 lib32
drwxr-xr-x   2 root root  4096 feb 17 11:14 lib64
drwx------   2 root root 16384 abr  7  2015 lost+found
drwxr-xr-x   4 root root  4096 may 28  2015 media
drwxr-xr-x   2 root root  4096 oct 23  2014 mnt
drwxr-xr-x   5 root root  4096 dic 14 11:43 opt
dr-xr-xr-x 316 root root     0 feb 10 18:14 proc
drwx------   7 root root  4096 feb 24 16:41 root
drwxr-xr-x  32 root root  1040 feb 26 12:37 run
drwxr-xr-x   2 root root 12288 feb 17 11:14 sbin
drwxr-xr-x   2 root root  4096 mar 26  2015 srv
dr-xr-xr-x  13 root root     0 feb 26 15:10 sys
drwxrwxrwt  42 root root 20480 feb 26 15:10 tmp
drwxr-xr-x  13 root root  4096 dic  1 11:37 usr
drwxr-xr-x  15 root root  4096 may 28  2015 var
lrwxrwxrwx   1 root root    30 abr  7  2015 vmlinuz -> boot/vmlinuz-3.19.0-10-generic
✔ 15:11 fsalavert@deku:/ $
```

Basic commands

ls --help

```
✔ 15:17 fsalavert@deku:/ $ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.
  -a, --all                  do not ignore entries starting with .
  -A, --almost-all           do not list implied . and ..
      --author               with -l, print the author of each file
  -b, --escape               print C-style escapes for nongraphic characters
      --block-size=SIZE      scale sizes by SIZE before printing them; e.g.,
                               '--block-size=M' prints sizes in units of
                               1,048,576 bytes; see SIZE format below
```

The **--help** or **-h** argument is generally used in commands, it shows an overview about how to use the command and all it's allowed parameters.

Basic commands

mkdir : **m**ake **dir**ectory

```
/bin/bash
✓ 15:25 fsalavert@deku:~/projects/examples/test $ mkdir myfolder
✓ 15:25 fsalavert@deku:~/projects/examples/test $ ll
total 12
drwxrwxr-x  3 fsalavert fsalavert 4096 feb 26 15:25 ./
drwxr-xr-x 10 fsalavert fsalavert 4096 feb 26 15:24 ../
drwxrwxr-x  2 fsalavert fsalavert 4096 feb 26 15:25 myfolder/
✓ 15:25 fsalavert@deku:~/projects/examples/test $
```

cd : **c**hange **d**irectory

```
/bin/bash
✓ 15:28 fsalavert@deku:~/projects/examples/test $ ll
total 12
drwxrwxr-x  3 fsalavert fsalavert 4096 feb 26 15:25 ./
drwxr-xr-x 10 fsalavert fsalavert 4096 feb 26 15:24 ../
drwxrwxr-x  2 fsalavert fsalavert 4096 feb 26 15:25 myfolder/
✓ 15:28 fsalavert@deku:~/projects/examples/test $ cd myfolder/
✓ 15:28 fsalavert@deku:~/projects/examples/test/myfolder $ pwd
/home/fsalavert/projects/examples/test/myfolder
✓ 15:28 fsalavert@deku:~/projects/examples/test/myfolder $
```

Basic commands

cd

This is command allows you to move across the directory tree, the equivalent when we are using GUI is to open folders from a window.

Special cases:

- **cd /** / Represents the root directory
- **cd .** . Represents the current directory, so nothing will happen
- **cd ..** .. Represents the parent directory, so you will move up
- **cd ~** ~ Represents the user's home directory located at /home/user

Basic commands

Absolute path:

- An absolute path is defined as the specifying the location of a file or directory from the root directory(/).
  - /home/fsalavert/projects
  - /tmp

Relative path:

- Relative path is defined as path related to the present working directory(pwd).
  - Suppose we are in **/home/user/test** and we want to change to **/home/user/test/myfolder**
  - We can use **cd myfolder/**

Basic commands

cp [**-r**] : **copy** file(s) or directories.

```
/bin/bash
✔ 15:56 fsalavert@deku:~/projects/examples/test $ ls -l
total 4
-rw-rw-r-- 1 fsalavert fsalavert    0 feb 26 15:55 data.txt
drwxrwxr-x 2 fsalavert fsalavert 4096 feb 26 15:56 myfolder
✔ 15:56 fsalavert@deku:~/projects/examples/test $ cp data.txt myfolder/data.txt
✔ 15:56 fsalavert@deku:~/projects/examples/test $ ls -l myfolder/
total 0
-rw-rw-r-- 1 fsalavert fsalavert 0 feb 26 15:56 data.txt
✔ 15:56 fsalavert@deku:~/projects/examples/test $
```

mv : move the file(s) or directories to a new location. Use it also for renaming.

```
/bin/bash
✔ 15:56 fsalavert@deku:~/projects/examples/test $ ll
total 12
drwxrwxr-x  3 fsalavert fsalavert 4096 feb 26 15:55 ./
drwxr-xr-x 10 fsalavert fsalavert 4096 feb 26 15:24 ../
-rw-rw-r--  1 fsalavert fsalavert    0 feb 26 15:55 data.txt
drwxrwxr-x  2 fsalavert fsalavert 4096 feb 26 15:56 myfolder/
✔ 16:01 fsalavert@deku:~/projects/examples/test $ mv data.txt newdata.txt
✔ 16:01 fsalavert@deku:~/projects/examples/test $ ll
total 12
drwxrwxr-x  3 fsalavert fsalavert 4096 feb 26 16:01 ./
drwxr-xr-x 10 fsalavert fsalavert 4096 feb 26 15:24 ../
drwxrwxr-x  2 fsalavert fsalavert 4096 feb 26 15:56 myfolder/
-rw-rw-r--  1 fsalavert fsalavert    0 feb 26 15:55 newdata.txt
✔ 16:01 fsalavert@deku:~/projects/examples/test $
```

Basic commands

rm [**-r**] : **rem**ove file(s) or directories.

```
/bin/bash
✔ 16:13 fsalavert@deku:~/projects/examples/test $ ls -l
total 4
drwxrwxr-x 2 fsalavert fsalavert 4096 feb 26 15:56 myfolder
-rw-rw-r-- 1 fsalavert fsalavert    0 feb 26 15:55 newdata.txt
✔ 16:13 fsalavert@deku:~/projects/examples/test $ rm newdata.txt
rm: remove regular empty file 'newdata.txt'? y
✔ 16:14 fsalavert@deku:~/projects/examples/test $ ls -l
total 4
drwxrwxr-x 2 fsalavert fsalavert 4096 feb 26 15:56 myfolder
✔ 16:14 fsalavert@deku:~/projects/examples/test $
```

rm -r allows you to remove directories, -r means recursive, **watch out!** so all the sub-tree will be deleted.

Basic commands

cat : concatenate files and print on the standard output (screen).

```
/bin/bash
✔ 16:21 fsalavert@deku:~/projects/examples/test $ ls -l
total 8
-rw-rw-r-- 1 fsalavert fsalavert  274 feb 26 16:20 data.txt
drwxrwxr-x 2 fsalavert fsalavert 4096 feb 26 15:56 myfolder
✔ 16:21 fsalavert@deku:~/projects/examples/test $ cat data.txt
 _____________________________________
/ This is the content of the data.txt \
\ file                                /
 -------------------------------------
   \
    \
        .--.
       |o_o |
       |:_/ |
      //   \ \
     (|     | )
    /'\_   _/`\
    \___)=(___/

✔ 16:21 fsalavert@deku:~/projects/examples/test $
```

Basic commands

head : output first lines of files.

tail : output last lines of files.

```
✔ 16:21 fsalavert@deku:~/projects/examples/test $ head -5 data.txt
 _____________________________________
/ This is the content of the data.txt \
\ file                                /
 -------------------------------------
   \
✔ 16:23 fsalavert@deku:~/projects/examples/test $
```

```
✔ 16:27 fsalavert@deku:~/projects/examples/test $ tail -8 data.txt
       .---.
      |o_o |
      |:_/ |
     //   \ \
    (|     | )
   /'\_   _/`\
   \___)=(___/

✔ 16:27 fsalavert@deku:~/projects/examples/test $
```

Basic commands

less : read the text files interactively.

- Use keyboard to move through the file
  - Arrows KEYS-> move through the document
  - Space KEY-> jump n lines
  - Intro KEY-> jump 1 line
  - q KEY-> exit

more : like **less** but with less options.

Basic commands

grep [**-v**] : print lines matching a pattern.

```
✔ 16:31 fsalavert@deku:~/projects/examples/test $ ls -l
total 8
-rw-rw-r-- 1 fsalavert fsalavert  274 feb 26 16:20 data.txt
drwxrwxr-x 2 fsalavert fsalavert 4096 feb 26 15:56 myfolder
✔ 16:31 fsalavert@deku:~/projects/examples/test $ grep "a" data.txt
/ This is the content of the data.txt \
✔ 16:31 fsalavert@deku:~/projects/examples/test $
```

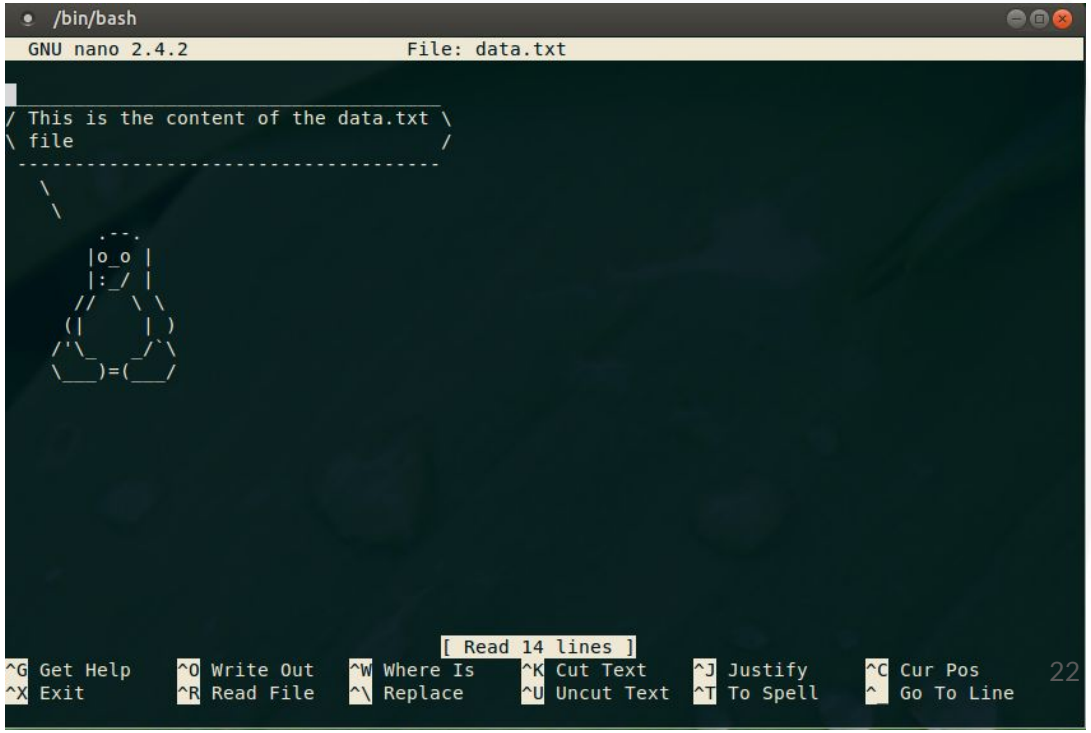
grep -v only prints lines that does not match the pattern.

Basic commands

nano : is a small friendly text editor that runs in the command line.

```
✔ 16:33 fsalavert@deku:~/projects/examples/test $ nano data.txt
```

- Use keyboard to move through the file
  - Arrows KEYS-> move through the document
  - Ctrl + K -> cut lines
  - Ctrl + U -> paste lines
  - Ctrl + X -> Exit
  - …



Basic commands

Operator ; : use it to introduce multiple commands at the same line.

```
✔ 16:39 fsalavert@deku:~/projects/examples/test $ pwd; ls; cd myfolder/
/home/fsalavert/projects/examples/test
data.txt  myfolder
✔ 16:39 fsalavert@deku:~/projects/examples/test/myfolder $
```

Operator > : redirect the output of a command to a file.

```
✔ 16:41 fsalavert@deku:~/projects/examples/test $ ll
total 16
drwxrwxr-x  3 fsalavert fsalavert 4096 feb 26 16:37 ./
drwxr-xr-x 10 fsalavert fsalavert 4096 feb 26 15:24 ../
-rw-rw-r--  1 fsalavert fsalavert  274 feb 26 16:20 data.txt
drwxrwxr-x  2 fsalavert fsalavert 4096 feb 26 15:56 myfolder/
✔ 16:41 fsalavert@deku:~/projects/examples/test $ pwd > pwd.txt
✔ 16:42 fsalavert@deku:~/projects/examples/test $ cat pwd.txt
/home/fsalavert/projects/examples/test
✔ 16:42 fsalavert@deku:~/projects/examples/test $
```

Operator | : redirect the output of a command as input for the next command.

```
✔ 16:43 fsalavert@deku:~/projects/examples/test $ cat data.txt | grep "a"
/ This is the content of the data.txt \
✔ 16:43 fsalavert@deku:~/projects/examples/test $
```

Basic commands

echo : display a line of text.

```
✔ 16:43 fsalavert@deku:~/projects/examples/test $ echo "Hello world"
Hello world
✔ 17:02 fsalavert@deku:~/projects/examples/test $
```

cut : Print selected parts of lines from each FILE to standard output.

```
✔ 17:14 fsalavert@deku:~/projects/examples $ cat file.gff
X       Ensembl Repeat  2419108 2419128 42      .       .       hid=trf; hstart=1; hend=21
X       Ensembl Repeat  2419108 2419410 2502    -       .       hid=AluSx; hstart=1; hend=303
X       Ensembl Repeat  2419108 2419128 0       .       .       hid=dust; hstart=2419108; hend=2419128
X       Ensembl Pred.trans.     2416676 2418760 450.19  -       2       genscan=GENSCAN00000019335
X       Ensembl Variation       2413425 2413425 .       +       .
X       Ensembl Variation       2413805 2413805 .       +       .
✔ 17:14 fsalavert@deku:~/projects/examples $ cut -f1 file.gff
X
X
X
X
X
X
✔ 17:14 fsalavert@deku:~/projects/examples $
```

Basic commands

wc : returns the total lines/words/bytes for a given file.

wc -l : counts lines.

```
✔ 17:18 fsalavert@deku:~/projects/examples $ wc -l file_1.vcf
5030 file_1.vcf
✔ 17:18 fsalavert@deku:~/projects/examples $
```

wc -m : counts characters.

```
✔ 17:20 fsalavert@deku:~/projects/examples $ wc -m file_1.vcf
371151 file_1.vcf
✔ 17:20 fsalavert@deku:~/projects/examples $
```

Basic commands

pwd	Print current working directory	
ls [-alh] directory	list directory content	ls /bin/ ls –alh ~/Desktop
cd path	Move to the given path	cd .. cd /usr/local/bin
cp [-r] source destination	Copy the source file/dir to the given location	cp –r ~/Documents ~/test cp test/myData.txt Documents/
mv source destination	Move the source file/dir to the given location	mv ~/test ~/Documents
rm [-r] source	Remove a file/directory	rm –r ~/Documents/test rm ~/Documents/data.txt
cat file1 file2 ...	Print the content of a text file	cat ~/test/data.txt
mkdir directory	Creates a directory	mkdir test mkdir ~/test/subtest

Basic commands

head file1 file2 ... **tail** file1 file2 ...	Print the head/tail of a text file	head -20 ~/test/data.txt tail -5 ~/test/data.txt
less file1 file2 ...	Read a text file interactively	less ~/test/data.txt
grep PATTERN file	Filter the file content looking for the given PATTERN	grep “hello” ~/test/data.txt
nano file	Edit the given file using the text editor nano	nano ~/test/data.txt
operator **;**	Used to write multiple commands in a line	
operator **>**	Redirect output to a file	
operator **|**	Redirect output to a command (pipe)	
echo message	Display a line of text	
cut -f1 file	Print selected parts of lines from each FILE to standard output	