

Dia 1. I.Introducción al uso de Terminal en UBUNTU (UNIX)

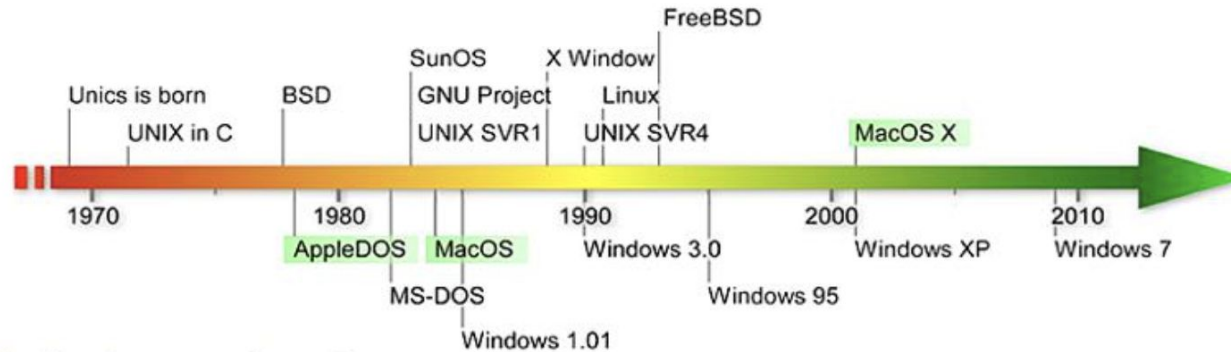


ubuntu

¿Qué es UNIX?



UNIX es una familia de sistemas operativos



Timeline of some operating systems.

The timeline of some UNIX/UNIX-like (top) and other non-UNIX (bottom) operating systems is represented. One can notice that Apple's operating system evolved from non-UNIX (AppleDOS then Mac OS) to UNIX (OS X).

¿Qué es UNIX?



Características claves de los sistemas UNIX

Multiuser

Possible to connect to the same server (remote) and execute different programs at the same time

Multitasking

Multiple processes can run on the same server at the same time

Networking

The network is essential for remote access

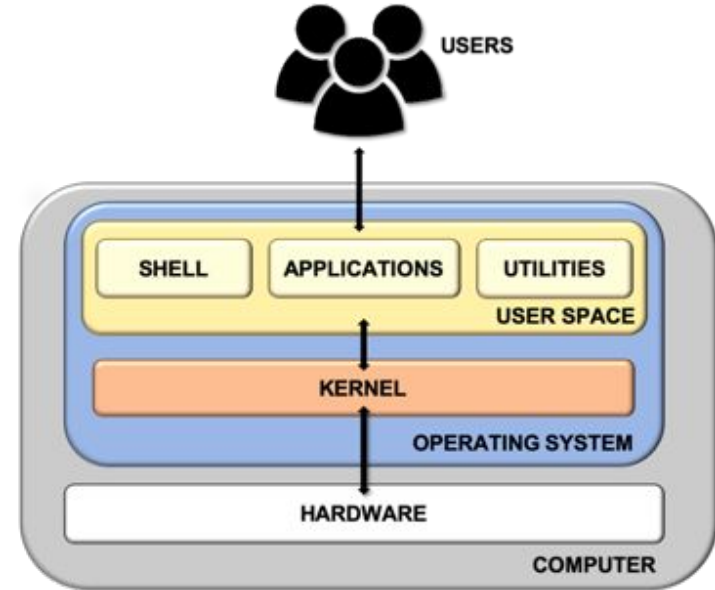
Various user interfaces

Both text-only and graphical interfaces are available

¿Qué es linux?



Linux es un sistema operativo de código abierto basado en Unix que fue desarrollado inicialmente por **Linus Torvalds** en 1991. Se compone de un **kernel (núcleo)** de sistema operativo y un **conjunto de herramientas** (SHELL, aplicaciones y utilidades) que lo acompañan.

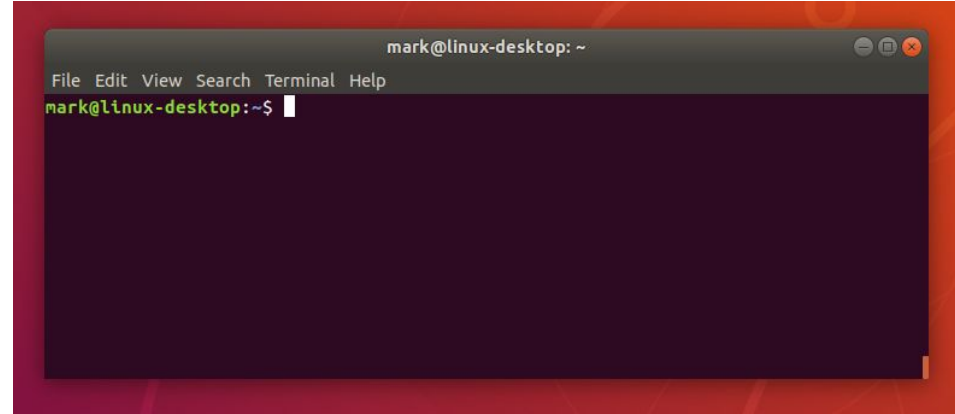


¿Que es la Shell?



En un sistema Linux, **el shell es una interfaz de línea de comandos** que interpreta los comandos del usuario y los archivos de script, y comunica al sistema operativo del servidor qué hacer con ellos.

Las **SHELL** más usadas son **BASH** y **Z SHELL (zsh)**



¿Que es la Shell?



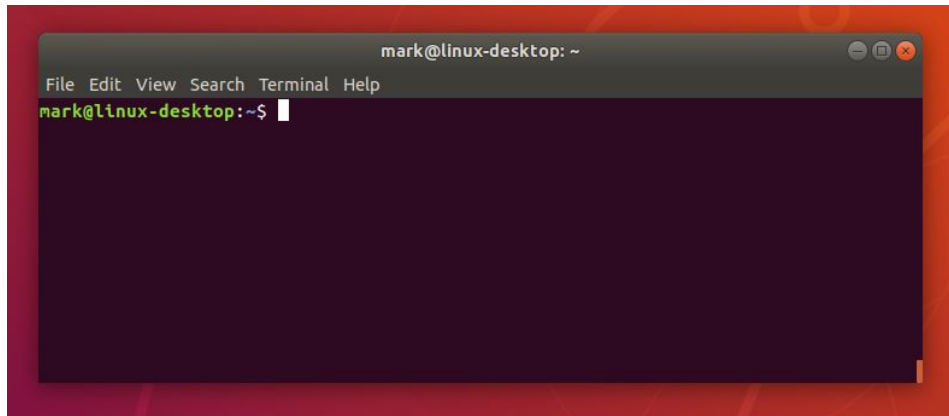
Command Prompt: Indicador de entrada de comandos

mark: El nombre de usuario del usuario actual

linux-desktop: El nombre del host del servidor

~: Ubicación actual.

\$: El símbolo del indicador, después del cual aparecerá la entrada del teclado del usuario



Ejecutando comandos

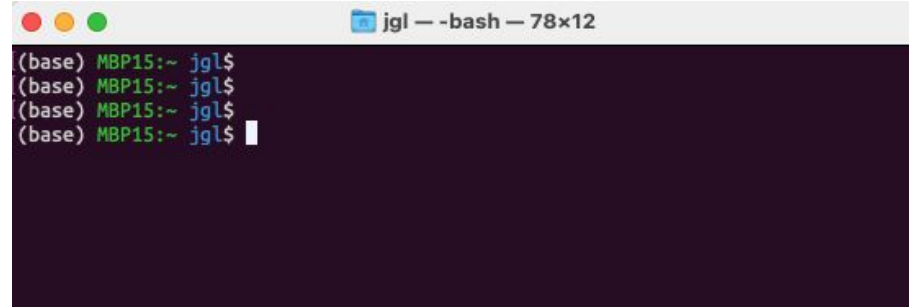


Cada comando es en sí es un pequeño programa (binary program script)

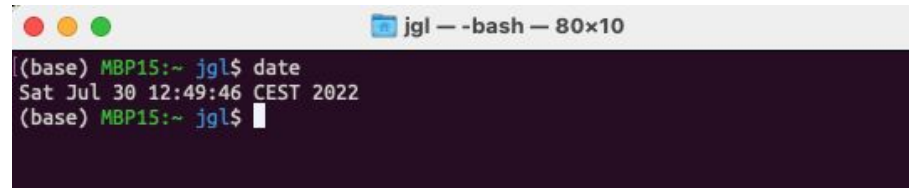
Pulsamos varias veces ENTER

Escribimos comando “Clear”

Ahora escribe comando “date”

A terminal window titled 'jgl - -bash - 78x12'. It shows four consecutive prompts '(base) MBP15:~ jgl\$' with a cursor at the end of each line, indicating that the ENTER key has been pressed multiple times without any commands being entered.

```
(base) MBP15:~ jgl$  
(base) MBP15:~ jgl$  
(base) MBP15:~ jgl$  
(base) MBP15:~ jgl$
```

A terminal window titled 'jgl - -bash - 80x10'. It shows the prompt '(base) MBP15:~ jgl\$' followed by the command 'date' and its output 'Sat Jul 30 12:49:46 CEST 2022'. The prompt is then shown again with a cursor at the end.

```
(base) MBP15:~ jgl$ date  
Sat Jul 30 12:49:46 CEST 2022  
(base) MBP15:~ jgl$
```

Ejecutando comandos



un comando puede tener:

opciones (suelen ir con -)

argumentos

```
[command] [options] [arguments]
```


Ejecutando comandos



comando **echo** toma como argumento el elemento que queremos imprimir en pantalla.

```
echo 'Hello World!'
```

```
## Hello World!
```

everything in Linux is case-sensitive

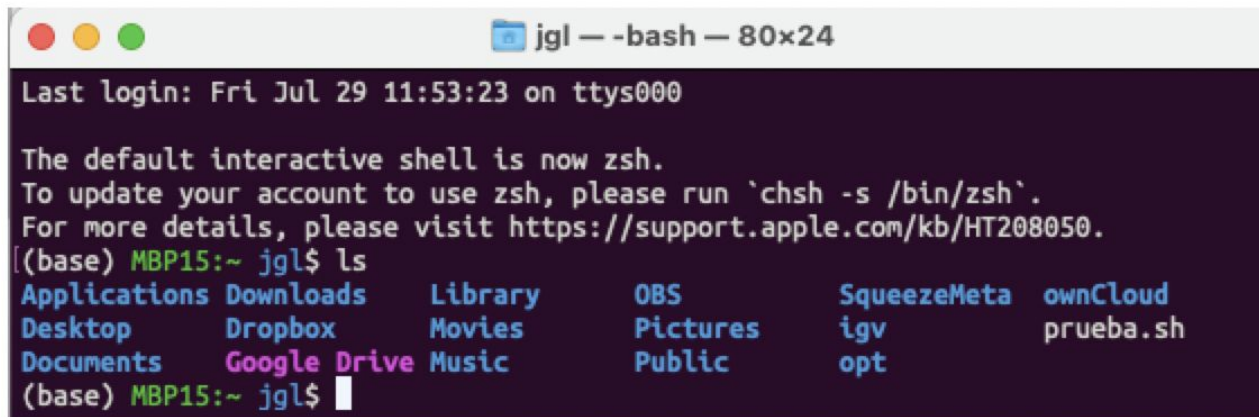
1.2. Listando archivos y directorios



ls (list)

muestra archivos y directorios en la ubicación actual

```
$ ls (short for list)
```

A terminal window titled 'jgl — -bash — 80x24' with standard macOS window controls (red, yellow, green buttons). The terminal shows the output of the 'ls' command. It starts with a login message, then a message about switching to zsh, and finally the output of 'ls' which lists various directories and files in a color-coded format.

```
Last login: Fri Jul 29 11:53:23 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
(base) MBP15:~ jgl$ ls
Applications Downloads Library OBS SqueezeMeta ownCloud
Desktop Dropbox Movies Pictures igv prueba.sh
Documents Google Drive Music Public opt
(base) MBP15:~ jgl$
```

1.2. Listando archivos y directorios



ls (list)

```
$ ls -l (long format)
```

```
jgl — -bash — 80x24
(base) MBP15:~ jgl$ ls -l
total 8
drwx-----@   8 jgl  staff   256 Feb 14 14:30 Applications
drwx-----@  51 jgl  staff  1632 Jul 29 12:06 Desktop
drwx-----+  30 jgl  staff   960 Jul  7 23:03 Documents
drwx-----@ 1095 jgl  staff 35040 Jul 29 11:12 Downloads
drwx-----@  58 jgl  staff  1856 Sep 26 2021 Dropbox
lrwx-----   1 jgl  staff    20 Jul 27 17:20 Google Drive -> /Volumes/GoogleD
rive
drwx-----@  76 jgl  staff  2432 Mar  3 12:49 Library
drwx-----   7 jgl  staff   224 Feb 21 09:48 Movies
drwx-----+   6 jgl  staff   192 Apr 26 2021 Music
drwxr-xr-x    5 jgl  staff   160 Apr 28 2021 OBS
drwx-----+   5 jgl  staff   160 Oct  5 2021 Pictures
drwxr-xr-x+   4 jgl  staff   128 Apr 25 2021 Public
drwxr-xr-x+  10 jgl  staff   600 Nov 27 11:16 SourceMaps
```

1.2. Listando archivos y directorios



Obtener ayuda para cualquier comando

```
man [command_name]
```

1.2. Listando archivos y directorios



ls (list)

```
jgl — bash — 80x24
(base) MBP15:~ jgl$ ls -l
total 8
drwx-----@  8 jgl  staff    256 Feb 14 14:30 Applications
drwx-----@ 51 jgl  staff   1632 Jul 29 12:06 Desktop
drwx-----+ 30 jgl  staff    960 Jul  7 23:03 Documents
drwx-----@ 1095 jgl  staff  35040 Jul 29 11:12 Downloads
drwx-----@  58 jgl  staff   1856 Sep 26 2021 Dropbox
lrwx-----  1 jgl  staff     20 Jul 27 17:20 Google Drive
rive
drwx-----@ 76 jgl  staff   2432 Mar  3 12:49 Library
drwx-----  7 jgl  staff    224 Feb 21 09:48 Movies
drwx-----+  6 jgl  staff    192 Apr 26 2021 Music
drwxr-xr-x  5 jgl  staff    160 Apr 28 2021 OBS
drwx-----+  5 jgl  staff    160 Oct  5 2021 Pictures
drwxr-xr-x+  4 jgl  staff    128 Apr 25 2021 Public
```

columns	Information
1	File permissions ('d' means is a directory)
3	Owner
4	Group
5	Size (bytes)-meaningful for files, but not for directories.
6, 7 and 8	Last modification time

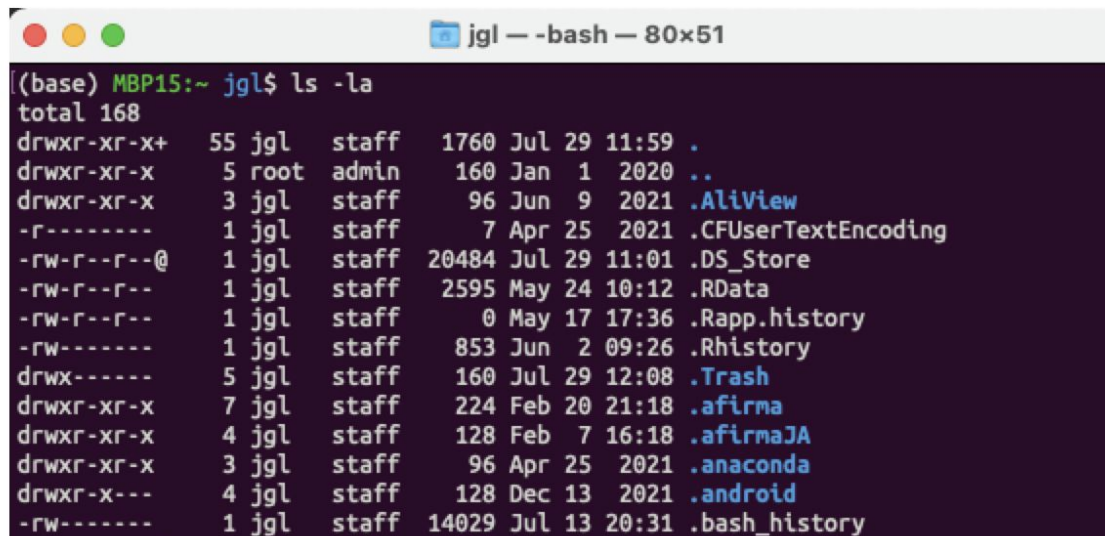
1.2. Listando archivos y directorios



ls (list)

Opción **-a** muestra archivos y directorios ocultos (empiezan por “.”)

```
$ ls -la
```

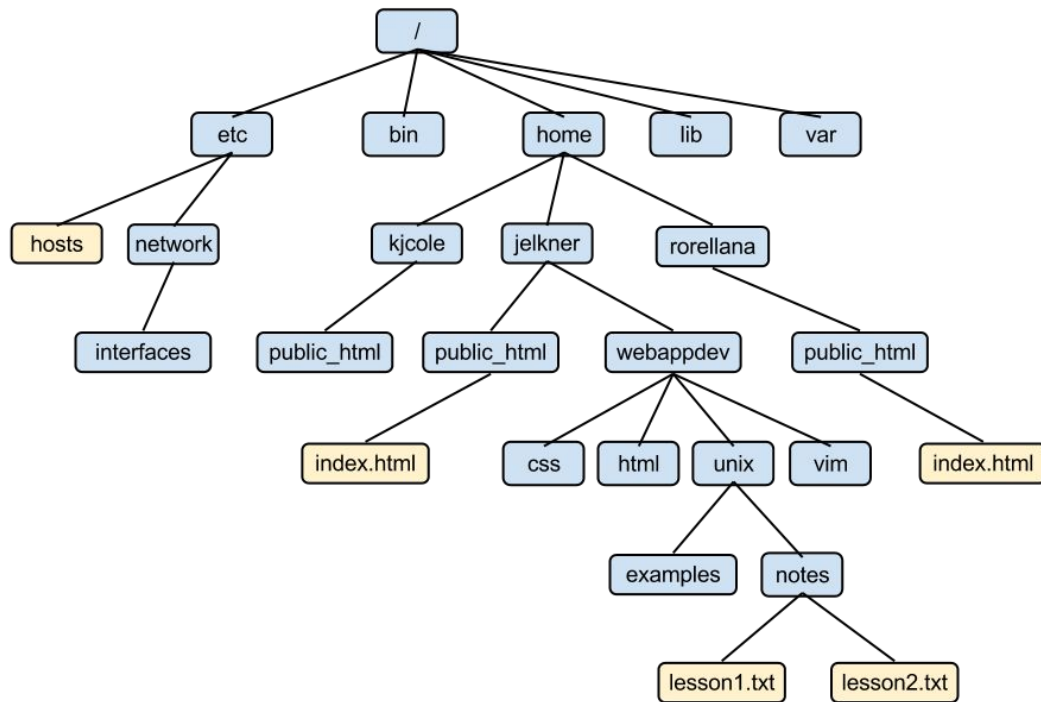


```
(base) MBP15:~ jgl$ ls -la
total 168
drwxr-xr-x+ 55 jgl  staff   1760 Jul 29 11:59 .
drwxr-xr-x   5 root   admin    160 Jan  1  2020 ..
drwxr-xr-x   3 jgl  staff     96 Jun  9  2021 .AliView
-r-----   1 jgl  staff     7 Apr 25  2021 .CFUserTextEncoding
-rw-r--r--@  1 jgl  staff 20484 Jul 29 11:01 .DS_Store
-rw-r--r--   1 jgl  staff 2595 May 24 10:12 .RData
-rw-r--r--   1 jgl  staff   0 May 17 17:36 .Rapp.history
-rw-----   1 jgl  staff  853 Jun  2 09:26 .Rhistry
drwx-----  5 jgl  staff   160 Jul 29 12:08 .Trash
drwxr-xr-x   7 jgl  staff   224 Feb 20 21:18 .afirma
drwxr-xr-x   4 jgl  staff   128 Feb  7 16:18 .afirmaJA
drwxr-xr-x   3 jgl  staff    96 Apr 25  2021 .anaconda
drwxr-x---   4 jgl  staff   128 Dec 13  2021 .android
-rw-----   1 jgl  staff 14029 Jul 13 20:31 .bash_history
```

1.3. Estructura de directorios en Linux



todos los archivos y directorios, en Linux se encuentran bajo el directorio **'root' o raíz representado por '/'**. Si observas la estructura de directorios, te darás cuenta de que es similar a la raíz de una planta.



1.4. Creando directorios



mkdir (make directory)

Crea un directorio en tu directorio de inicio llamado **bioinformatic_course**

```
mkdir bioinformatic_course
```

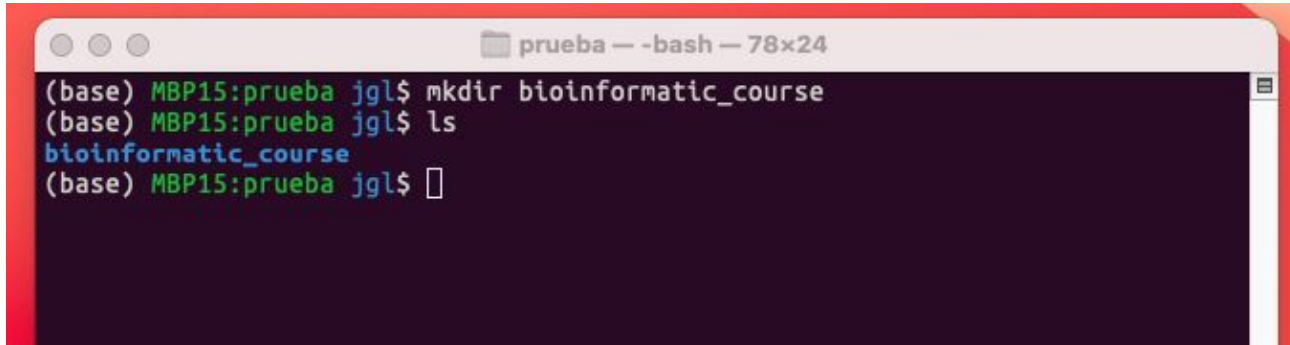
ahora revisa si lo has creado con el comando **ls**

1.4. Creando directorios



mkdir (make directory)

Crea un directorio en tu directorio \$Home llamado **bioinformatic_course**

A terminal window with a dark purple background and a light gray title bar. The title bar contains the text 'prueba — -bash — 78x24'. The terminal shows the following commands and output:

```
(base) MBP15:prueba jgl$ mkdir bioinformatic_course
(base) MBP15:prueba jgl$ ls
bioinformatic_course
(base) MBP15:prueba jgl$
```

1.5. Cambiando de directorio



cd (Change Directory)

El comando `cd [directorio]` significa cambiar el directorio de trabajo actual al directorio especificado en el parámetro `[directorio]`

vamos a movernos al directorio que acabamos de crear

```
cd bioinformatic_course
```

ahora revisa el contenido del directorio ¿que pinta tiene?

1.5. Cambiando de directorio



Ejercicio

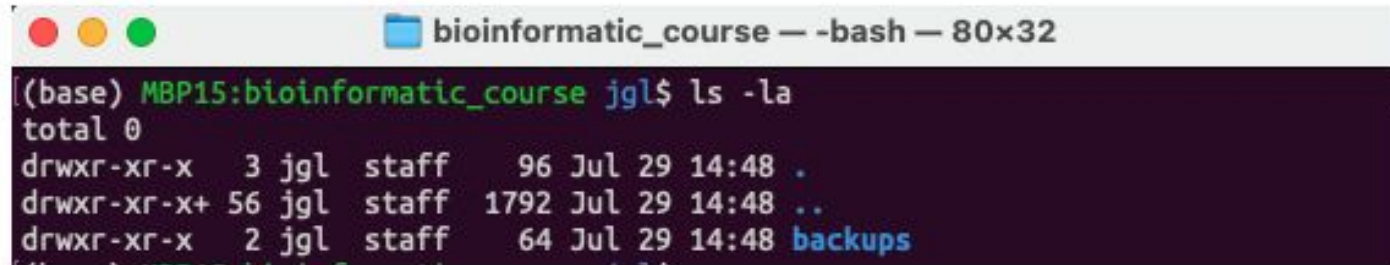
*Crea un nuevo directorio dentro de `bioinformatic_course` llamado: **backup***

1.6. Directorios . y ..



Volvemos a Bioinformatic course y escribimos: **ls -la**

```
$ ls -la
```

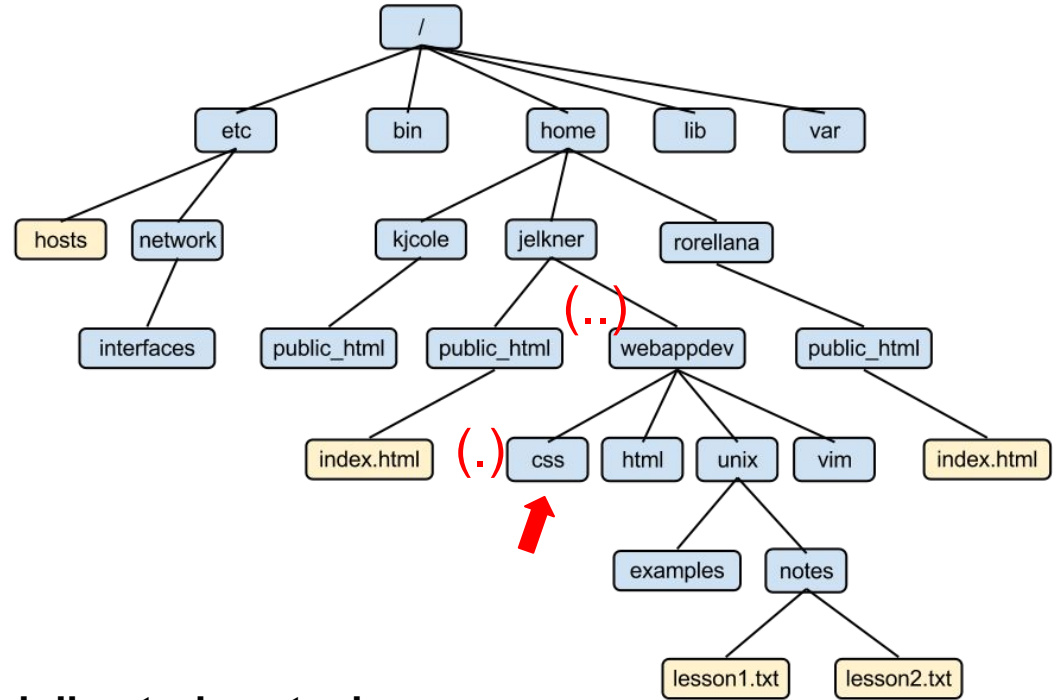
A terminal window titled 'bioinformatic_course — -bash — 80x32'. The prompt is '(base) MBP15:bioinformatic_course jgl\$'. The command 'ls -la' has been executed, showing the following output:

```
total 0
drwxr-xr-x  3 jgl  staff   96 Jul 29 14:48 .
drwxr-xr-x+ 56 jgl  staff 1792 Jul 29 14:48 ..
drwxr-xr-x  2 jgl  staff   64 Jul 29 14:48 backups
```

(.) significa el directorio actual

(..) significa el directorio padre del directorio actual

1.6. Directorios . y ..



(.) significa el directorio actual

(..) significa el directorio padre del directorio actual

1.7. Pathnames



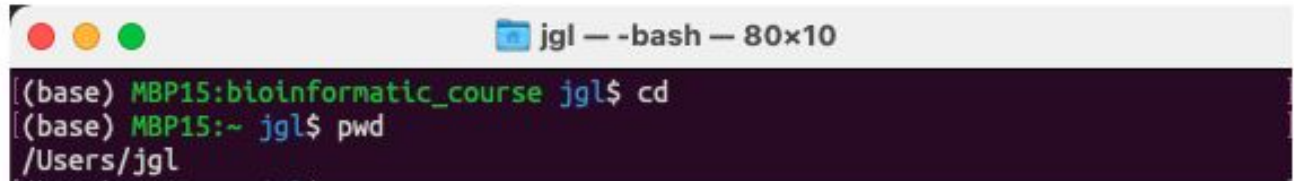
pwd (Print Working Directory)

Los **pathnames** te permiten saber dónde estás en relación con todo el sistema de archivos.

Para averiguar la ruta absoluta (**absolute path**) de tu directorio de inicio (\$HOME):

pwd

/Users/jgl

A terminal window with a dark background and light text. The title bar shows 'jgl - -bash - 80x10'. The terminal content shows a prompt '(base) MBP15:bioinformatic_course jgl\$' followed by the command 'cd'. A second prompt '(base) MBP15:~ jgl\$' is followed by the command 'pwd'. The output of the command is '/Users/jgl'.

directorio Raiz/ directorio Users/ directorio jgl/ (Home)

1.7. Pathnames



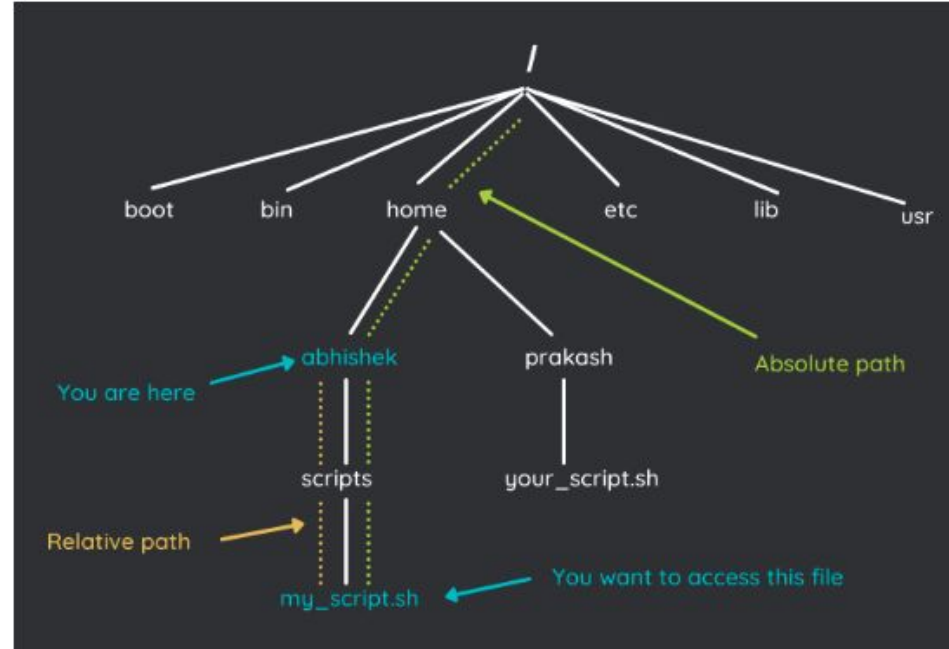
Absolute Path y Relative Path

Absolute Path: siempre comienza desde **Root “/”**.

```
/home/abhishek/scripts/my_script.sh
```

Relative Path: comienza en nuestro directorio actual

```
/scripts/my_script.sh
```



1.7. Pathnames



Ejercicios

1. Utiliza los comandos **ls**, **pwd** y **cd** para explorar el sistema de archivos.

*(Recuerda, si te pierdes, escribe solo **cd** para regresar a tu directorio de inicio)*

Crea otro directorio en **bioinformatic_course** llamado **scripts**

Dentro de **scripts** crea nuevamente un nuevo directorio llamado **python**. Entra en directorio **python**

Obtén **relative path del directorio backup** desde tu directorio actual (**python**). Obten también el **absolute path** del directorio **python**.

2. Intenta usar el comando **tree**. Ve a tu directorio de inicio y escribe **tree bioinformatic_course**.

Dia 1. II Creación, destrucción e inspección en (UNIX)



ubuntu

2.1. Creando Archivos



touch

crea archivos vacíos o actualiza las marcas de tiempo de los archivos existentes.

```
touch journal-2017-01-24.txt  
ls  
## journal-2017-01-24.txt
```

2.1. Creando Archivos



Redirección output >

Puedes usar la redirección de salida escribiendo el signo mayor que > al final de un comando, seguido del nombre del nuevo archivo que contendrá la salida del comando

```
$ echo "I'm in the terminal."  
## I'm in the terminal.
```

```
echo "I'm in the file." > echo-out.txt
```

```
cat echo-out.txt  
## I'm in the file.
```

2.1. Creando Archivos



Redirección output >

¿Qué ocurre si queremos añadir una segunda línea?

```
echo "A third line." > echo-out.txt  
cat echo-out.txt  
## A third line.
```

2.1. Creando Archivos



Redirección output >

¿Qué ocurre si queremos añadir una segunda línea?

```
echo "A third line." > echo-out.txt  
cat echo-out.txt  
## A third line.
```

Hemos sobrescrito el archivo!!!

2.1. Creando Archivos



Creando y editando archivos con NANO

nano es un editor de texto muy sencillo que puede utilizarse en la terminal.

Crea archivo con nano: (desde el home)

```
nano todo.txt
```



```
GNU nano 2.0.6
```

```
File: todo.txt
```



```
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos  
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

2.1. Creando Archivos



Creando y editando archivos con NANO

Una vez terminemos de escribir podemos salvar el archivo pulsando **Ctrl+O** y salimos con **Ctrl+X**

GNU nano 2.0.6 File: todo.txt

```
- email Jaime  
- write bioinformatic protocols  
- write final section of "R command Line"
```

^G Get Help	^O WriteOut	^R Read File	^Y Prev Page	^K Cut Text	^C Cur Pos
^X Exit	^J Justify	^W Where Is	^V Next Page	^U UnCut Text	^T To Spell

2.1. Creando Archivos



Creando y editando archivos con NANO

Podemos ver el contenido del archivo todo.txt

```
cat todo.txt
## - email Jaime
## - write bioinformatic protocols
## - write final section of "R command Line"
```



2.1. Creando Archivos



Convención para el nombre de archivos

Un **directorio** es simplemente un tipo especial de archivo. Por lo tanto, las reglas y convenciones para nombrar archivos también se aplican a los directorios.

1. Se deben evitar caracteres con significados especiales como / * & % , .
2. **Evita** usar **espacios** dentro de los nombres.
3. Usa **solo** caracteres alfanuméricos, **letras y números, junto con _ (guión bajo) y . (punto)**.
4. Deben **comenzar con una letra minúscula**,
5. Deben **terminan con un (.)** seguido de una extensión indican el contenido del archivo. Por ejemplo, todos los archivos que consisten en código Python pueden tener el final .py, por ejemplo, myprogram.py.

mi_script.py

mi_tabla.csv

mi_archivo_tabulado.tsv

mi_archivo_texto_plano1.txt

2.2. Copiando Archivos



cp (copy)

copia [argumento1] en [argumento2]

cp path(file1) path(file2)

```
$ cp ~/Desktop/science.txt /Users/jgl/bioinformatic_course
```

2.2. Copiando Archivos



cp (copy)

copia [argumento1] en [argumento2]

cp path(file1) path(file2)

```
$ cp ~/Desktop/science.txt /Users/jgl/bioinformatic_course
```

```
$ cp ~/Desktop/science.txt .
```

2.2. Copiando Archivos



Ejercicio

Crea una copia de tu archivo todo.txt, haciendo una copia llamada backup_todo.txt

2.3. Moviendo/Renombrando Archivos



mv (move)

mueve o renombra [argumento1] en [argumento2]

```
mv backup_todo.txt python/backup_todo.txt
```

Escribe ahora `ls python/`

2.3. Moviendo/Renombrando Archivos



mv (move)

mueve o renombra [argumento1] en [argumento2]

```
mv backup_todo.txt python/backup_todo.txt
```

Escribe ahora `ls python/`

2.4. Eliminando archivos y directorios



rm (remove)

elimina un archivo.

rm (file)

```
$ cp todo.txt copia_todo.txt
$ ls
$ rm copia_todo.txt
$ ls
```

2.4. Eliminando archivos y directorios



rm (remove)

elimina un directorio.

rm -r directorio

```
$ mkdir nuevo_directorio  
$ ls  
$ rm -r nuevo_directorio  
$ ls
```


2.4. Eliminando archivos y directorios



Ejercicios

Crea un nuevo directorio en bioinformatic_course denominado **/tareas**

crea un archivo con nano que se llame **lista.txt** (anota 2 o 3 tareas)

elimina el archivo **lista.txt**

luego elimina el directorio **/tareas**

2.5. Visualizar un archivo en terminal



cat (concatenate): muestra el contenido de una archivo (tb concatena archivos)

```
$ cat protein.fasta
```

```
$ cat protein.fasta protein2.fasta
```

less: muestra el contenido de una archivo pagina a pagina en terminal. barra espaciadora nueva página. presiona q para salir

```
$ less protein.fasta
```

2.5. Visualizar un archivo en terminal



head: muestra las 10 primeras líneas de un archivos

```
$ head protein.fasta
```

puede modularse el número de líneas con la **opción -n**

```
$ head -n 5 protein.fasta
```

tail: muestra las 10 últimas líneas de un archivos

```
$ tail protein.fasta
```

¿Podemos ver las últimas 15 líneas de un archivo?

2.6. Analizando un archivo



grep

Algunas de otras opciones de grep son:

- **v** muestra las líneas que NO coinciden
- **n** precede cada línea coincidente con el número de línea
- **c** imprime solo el recuento total de líneas coincidentes

Prueba algunas de ellas y observa los resultados diferentes. No olvides que puedes usar más de una opción a la vez, por ejemplo, el número de líneas sin las palabras science o Science es:

```
$ grep -ivc science science.txt
```

2.6. Analizando un archivo



wc (Word Count)

permite realizar un conteo de *lines/word/characters*

```
$ wc science.txt
```

lines/word/character opciones **-l -w -c o** combinadas

```
$ wc -l science.txt
```

2.6. Wildcards



Los caracteres especiales * e ?

El carácter `*` se llama comodín y **coincide con ninguno o más caracteres en un nombre de archivo** (o directorio). Por ejemplo, en tu directorio `bioinformatic_course`:

```
$ touch list1 list22 list33 1list
```

ahora prueba

```
$ ls list*
```

Y prueba

```
$ ls *list
```

2.6. Wildcards



Los caracteres especiales `*` e `?`

El carácter `?` es otro comodín pero en este caso solo **coincide exactamente con un caracter en un nombre de archivo** (o directorio). Prueba

```
$ ls ?list
```

Finalmente puedes eliminar todos los archivos usando `*`

```
$ rm ?list*
```

Ejercicios I



1. Crea un archivo llamado **message.txt** en tu directorio personal y muévelo a otro directorio.
2. Copia el **message.txt** que acabas de mover a tu directorio personal.
3. Crea un nuevo directorio llamado **workbench** en tu directorio personal.
4. Sin cambiar de directorio, crea un archivo llamado **readme.txt dentro de workbench**.
5. Añade los números **1, 2 y 3 a readme.txt** para que cada número aparezca en su propia línea.
6. **Imprime readme.txt** en la terminal.
7. Utiliza la redirección de salida (>) para crear un nuevo archivo en el directorio **workbench** llamado **list.txt** que contenga la salida del comando **ls** en el directorio **bioinformatic_course**.
8. Obtén el **número de caracteres y líneas** que hay en **list.txt** sin abrir el archivo ni imprimirlo en la línea de comandos.
9. Imprime la **primera y la última** línea de **list.txt** en la línea de comandos.

Ejercicios II



1. Descarga el archivo **sequences.fasta** presente en la carpeta de recursos de este repositorio.
2. Imprime el encabezado de la primera secuencia fasta.
3. Imprime el encabezado de la última secuencia fasta.
4. ¿Cuántas secuencias de proteínas contiene sequences.fasta?
5. ¿Cuántas secuencias contienen el motivo **LLR**?
6. Intenta la opción **-B1** para ver el nombre de las secuencias que contienen el motivo **LLR**. ¿Todas las secuencias contienen el mismo número de motivos?
7. Usando nano, modifica el encabezado de la primera secuencia fasta, elimina todas las palabras después de `tr|A0A060IHA6|A0A060IHA6_9RHIZ ID`.

Imprime las cabeceras de las secuencias fasta utilizando grep “^>” nombre de archivo. Explora el uso de -An y -Bn. Que secuencias contienen el motivo LLR?