



MINISTERIO
DE CIENCIA, INNOVACIÓN
Y UNIVERSIDADES



> BU-ISCIII

Sesión 2 - Iniciación al entorno Linux

Miguel Juliá

BU-ISCIII

Unidades Científico Técnicas – SGAFI-ISCIII

17-21 Junio 2019, 7^a Edición
Programa Formación Continua, ISCIII

Índice

Día 1:

- Introducción. Generalidades
- Sistema de ficheros
- Virtualización
- Cuentas de usuario
- Comandos básicos

Día 2:

- Comandos básicos
- Comandos procesamiento de archivos
- Administración ficheros y directorios
- Workflows

Introducción

- Principales componentes un ordenador
- Sistema Operativo (SO)
- Interfaz
- Software Libre
- Distribuciones GNU/LINUX
- Principales elementos del Sistema operativo Linux

Principales componentes un ordenador



- **Procesador o CPU**
 - Controla el funcionamiento del computador y procesa los datos
- **Memoria y dispositivos de almacenamiento**
 - Almacenan datos, instrucciones y Programas
- **Subsistema de interconexión o buses**
 - Proporciona un medio de comunicación entre el procesador, los elementos de almacenamiento y los periféricos.
- **Dispositivos de entrada/salida o periféricos**
 - Transfieren datos entre el computador y el entorno externo

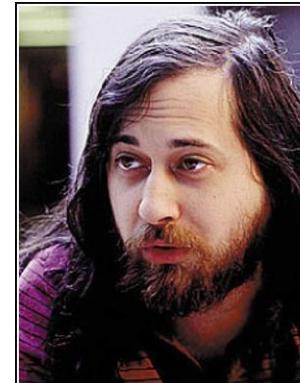
Sistema Operativo (SO)

- **Definición**
 - El SO es un programa o conjunto de programas que actúa como intermediario entre las aplicaciones y el hardware del ordenador, gestionando los recursos del sistema y optimizando su uso
- **Funciones:**
 - Control de la ejecución de los programas
 - Administración de periféricos
 - Gestión de permisos de usuario
 - Control de concurrencia y errores
 - Control de seguridad



Software Libre

- Concepto creado por el movimiento GNU, fundado por Richard Stallman
- ¿Qué es software libre?
 - El código fuente debe ser público y libre
 - Libre significa tener libertad sobre el código para:
 - Modificarlo, adaptarlo o mejorararlo
 - Redistribuirlo o comercializarlo
 - Ejecutarlo para cualquier propósito
 - Estudiar su funcionamiento
 - El software libre se distribuye bajo licencias distintas a las del software privativo, siendo la GNU (General Public License (GPL)) la más popular.

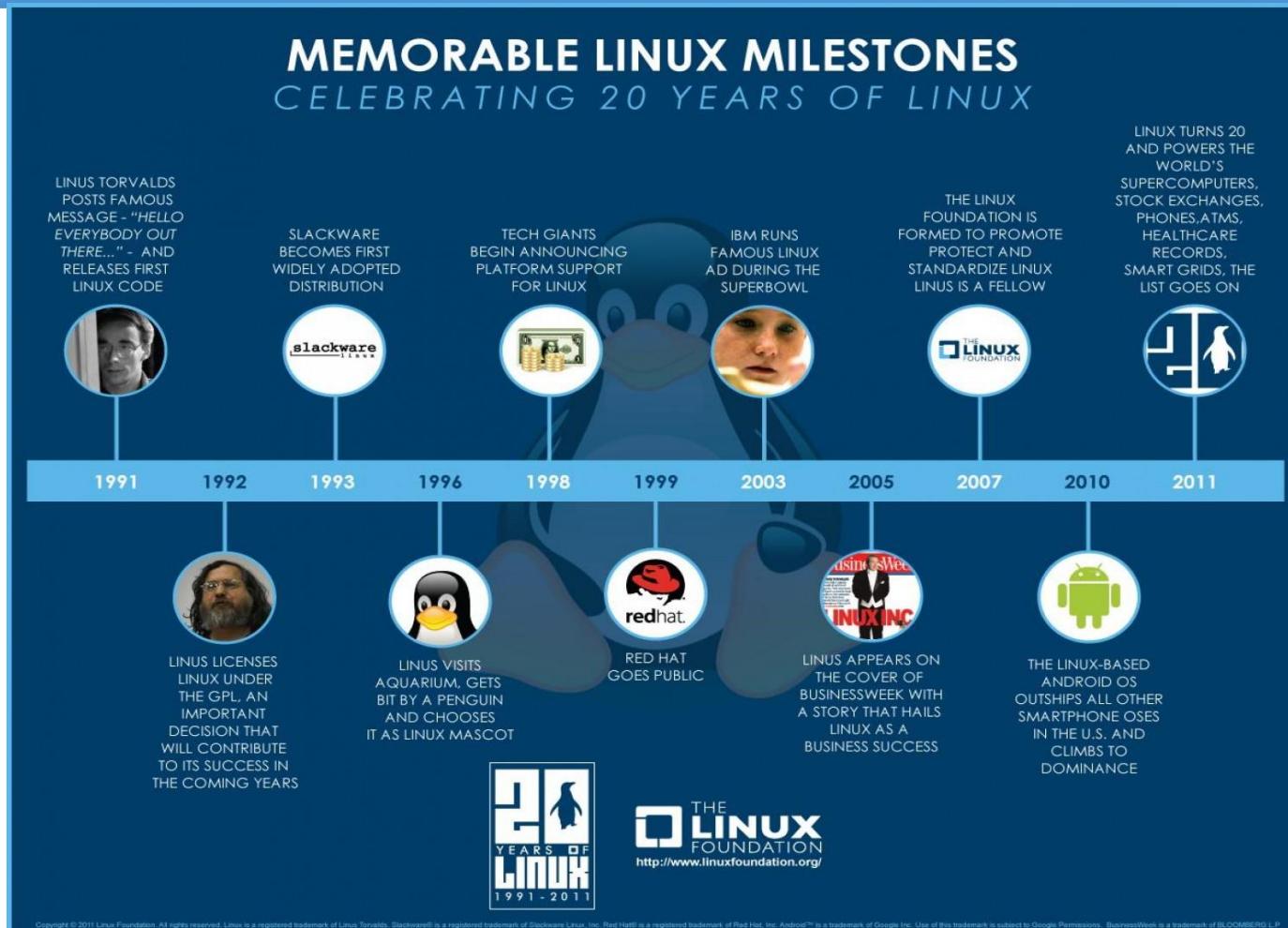


'Free software' is a matter of liberty, not price. To understand the concept, you should think of 'free' as in 'free speech,' not as in 'free beer'.

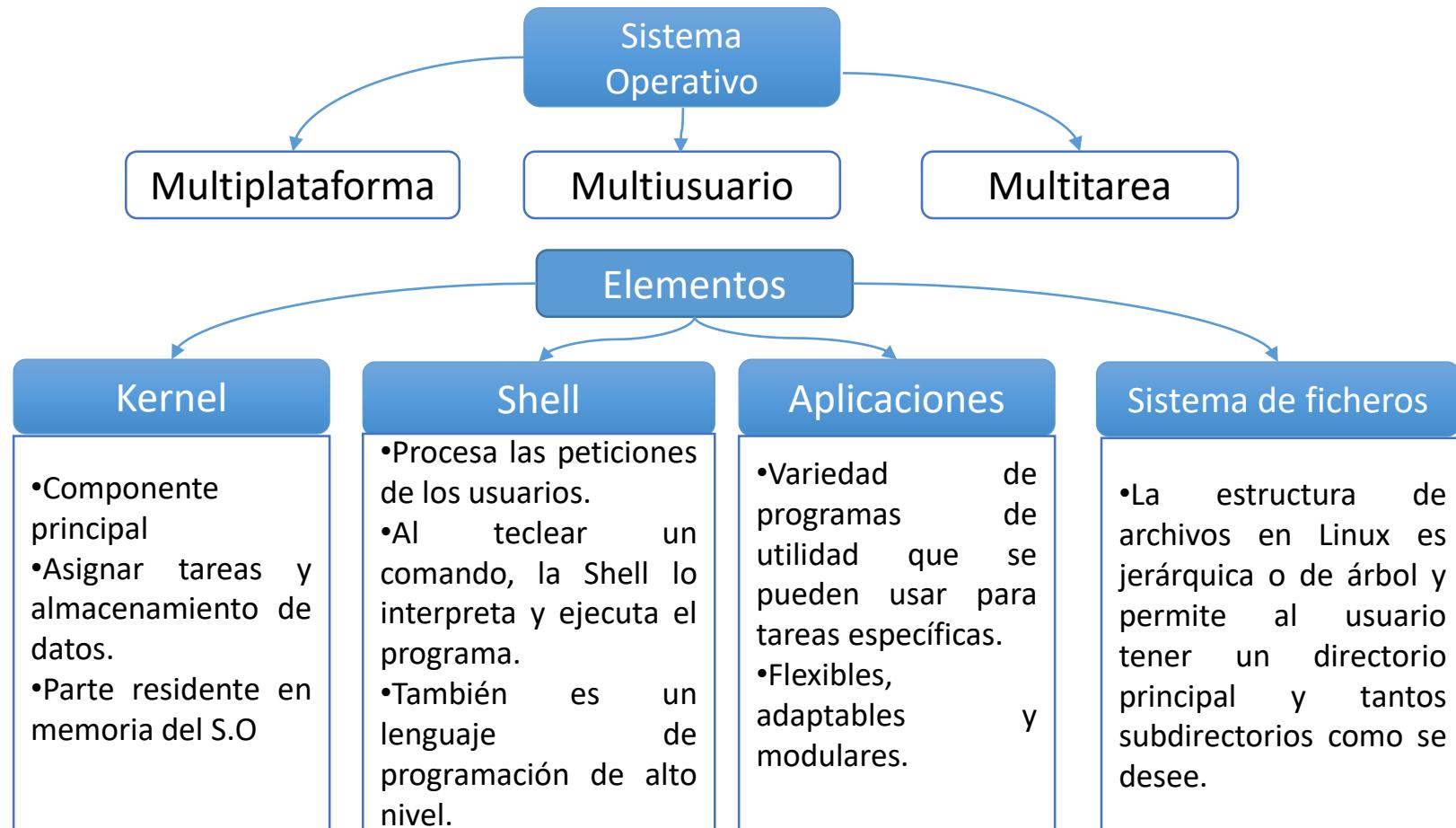
— Richard Stallman —

AZ QUOTES

SO Linux



Elementos del SO Linux



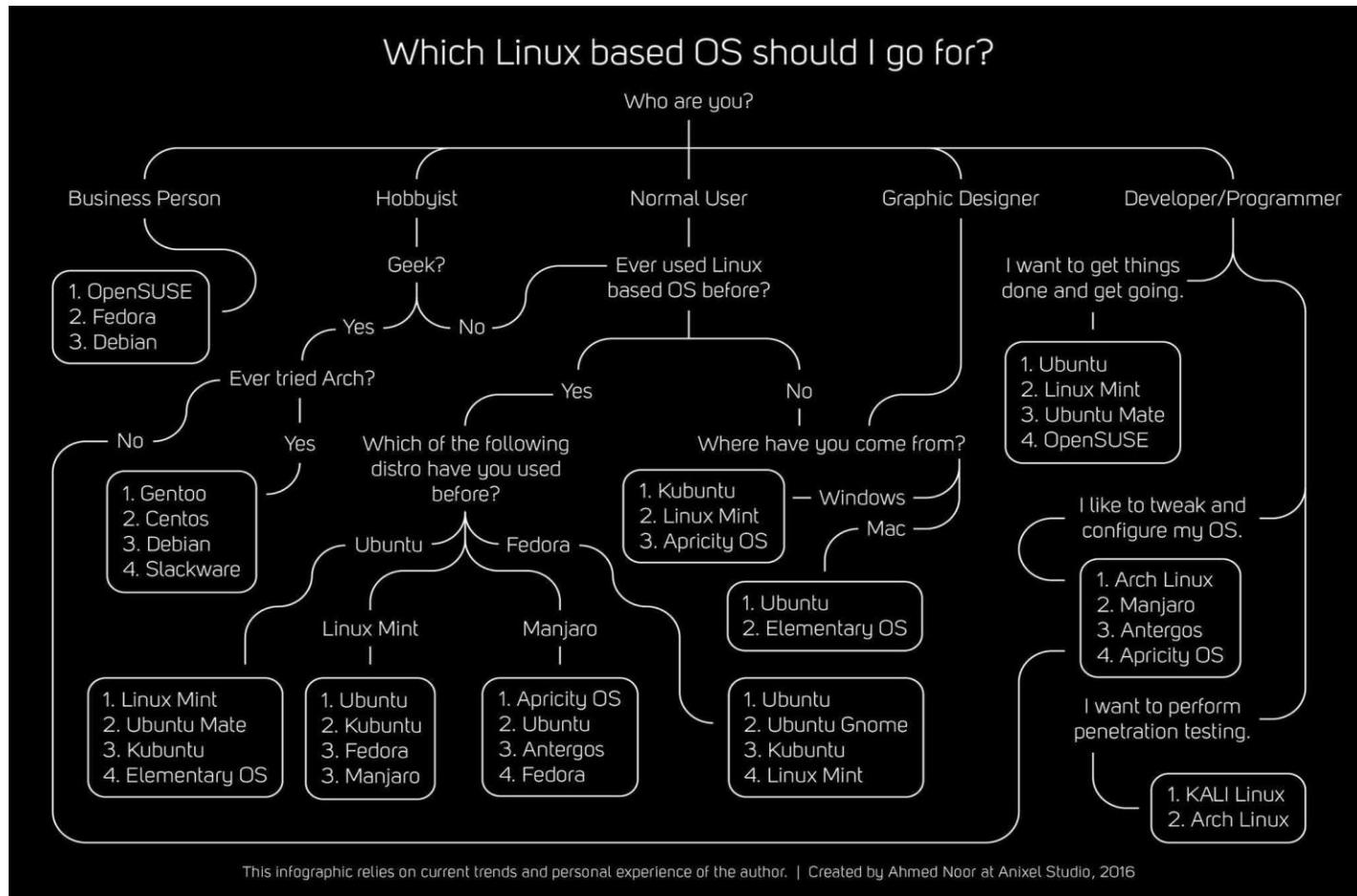
Distribuciones Linux

- Una distribución Linux es un conjunto de aplicaciones y herramientas de configuración reunidas por un grupo, empresa o persona.
- Existen numerosas distribuciones Linux. Cada una de ellas puede incluir cualquier número de software adicional (libre o no) como entornos gráficos, suites ofimáticas, servidores web, servidores de correo, servidores FTP, etc.

Distribuciones Linux

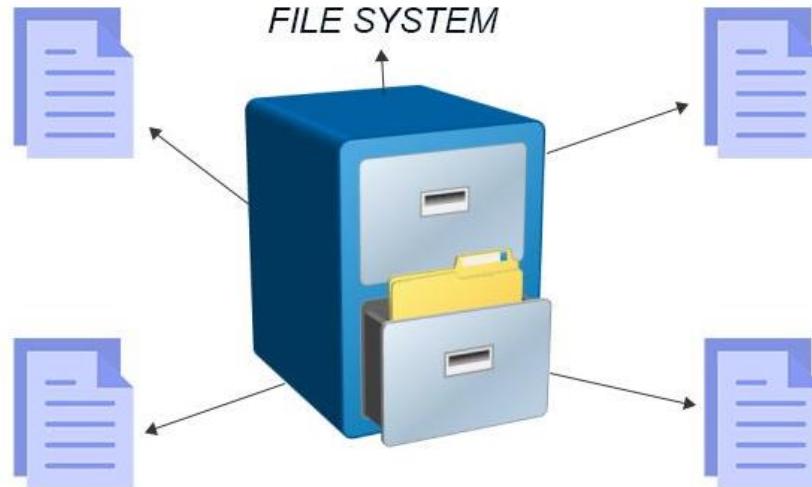


Distribuciones Linux



Sistema de ficheros

- Características
- Árbol de Directorios Linux
- Rutas de acceso.
 - Ruta absoluta y ruta relativa
 - Directorio padre y directorio actual
- Ficheros
- Permisos



Sistema de ficheros

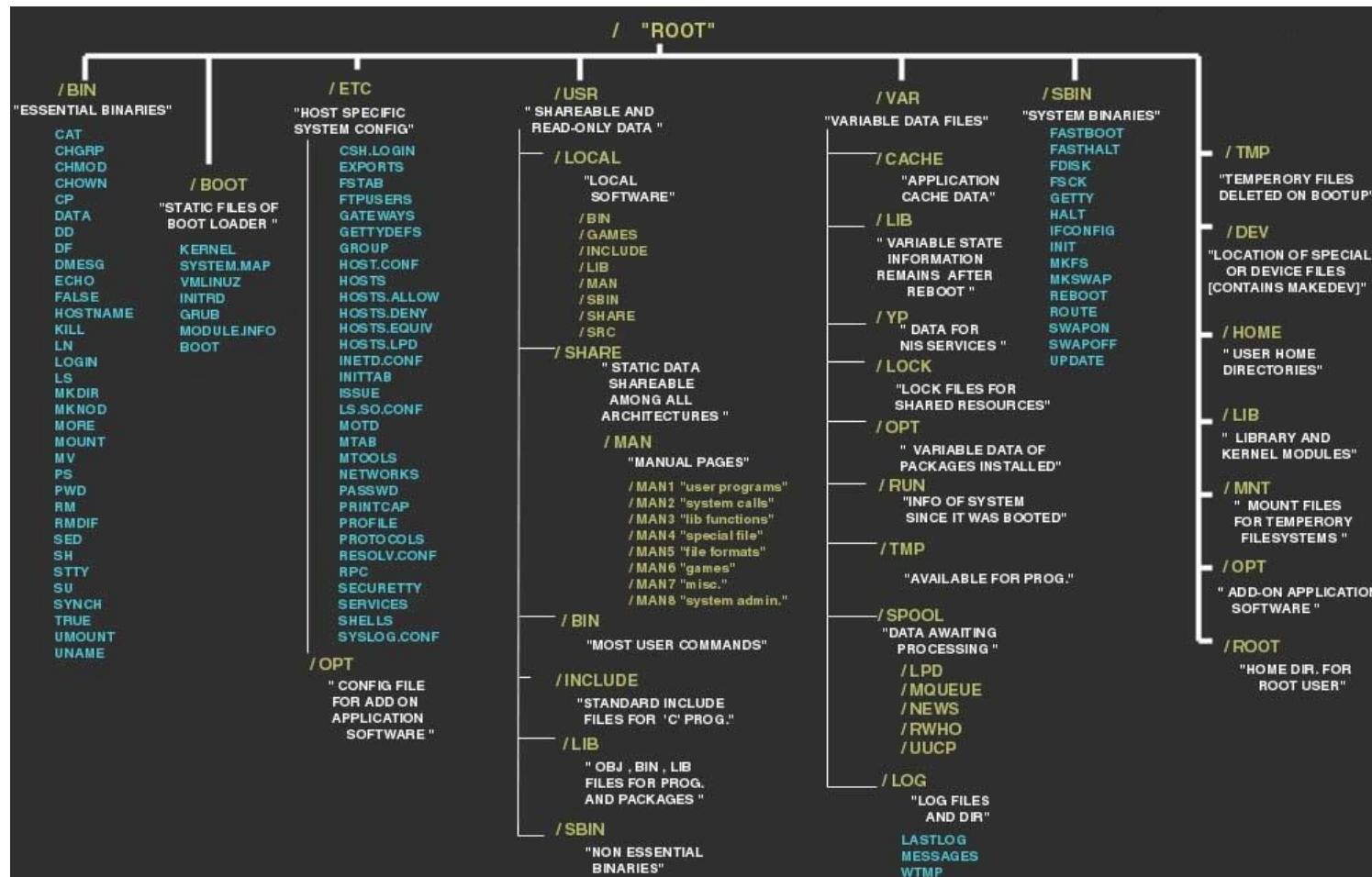
Todo es un fichero

- Los ficheros son ficheros, los directorios son ficheros y los dispositivos se representan con ficheros.
- No existe el concepto de extensiones, un programa es un fichero que tiene permisos para ser ejecutado
- Todo lo que hay en el sistema cuelga del directorio raíz /

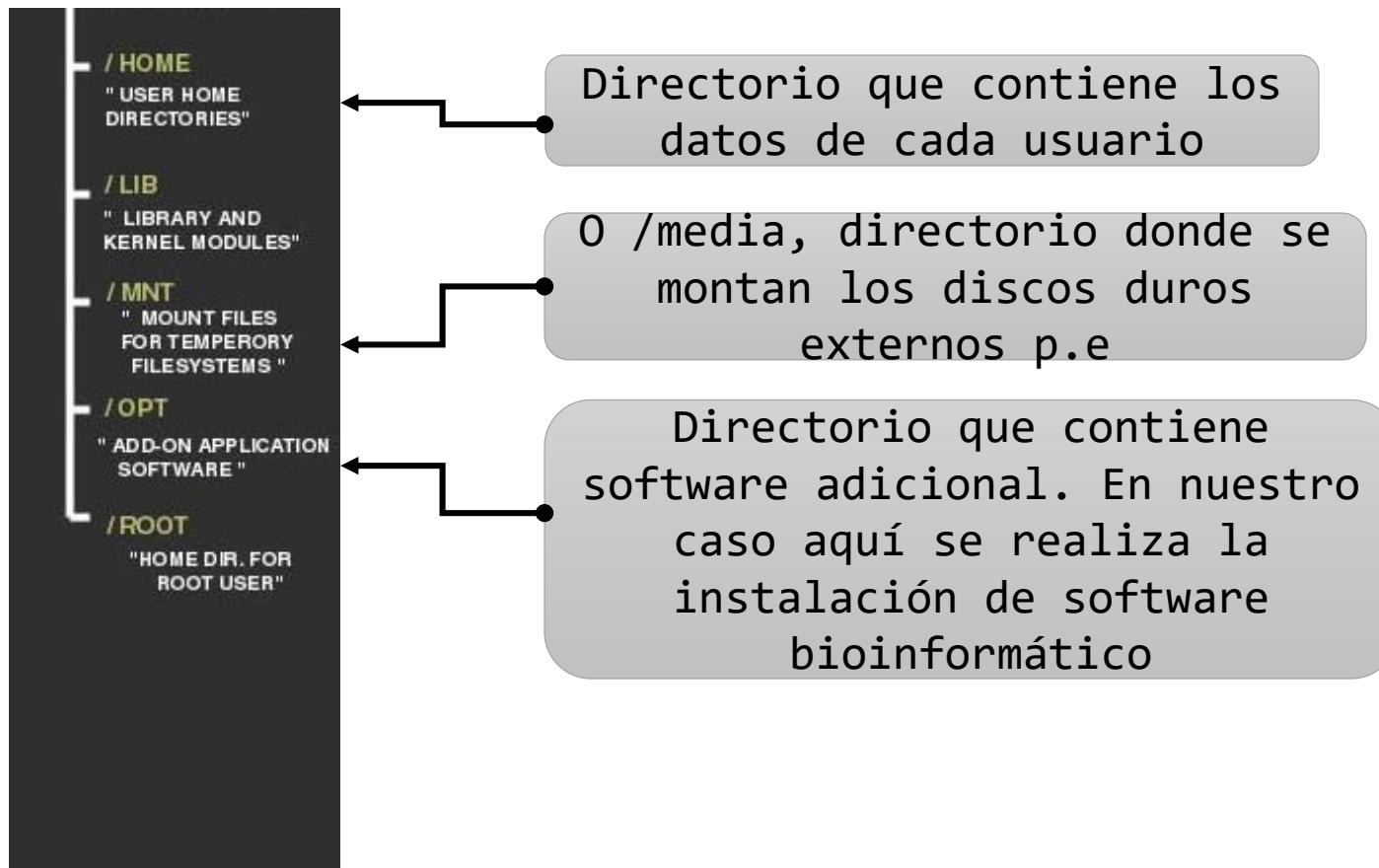
Árbol de directorios



Árbol de directorios



Árbol de directorios

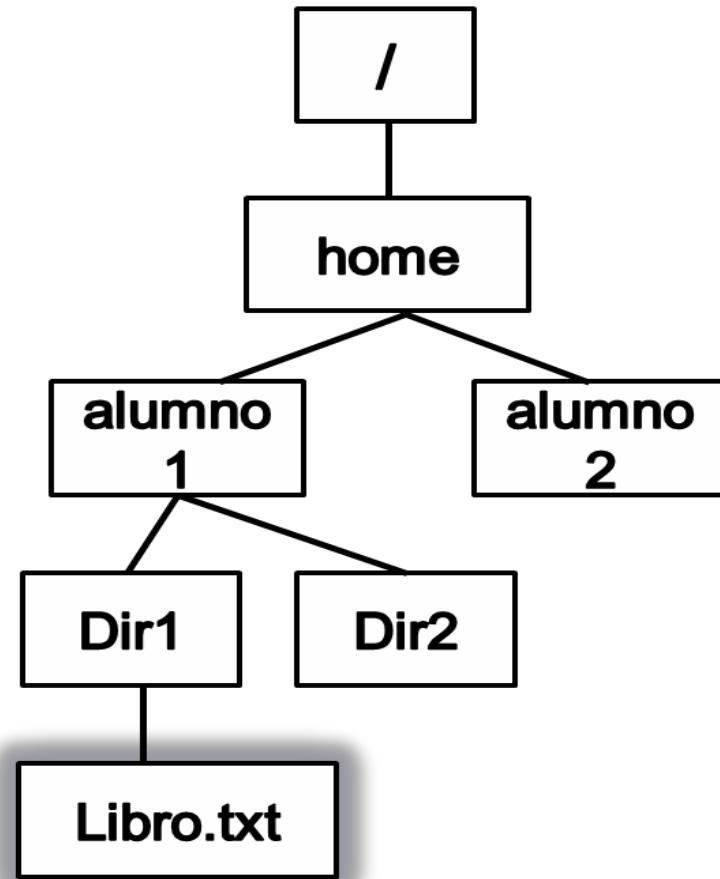


Rutas de acceso

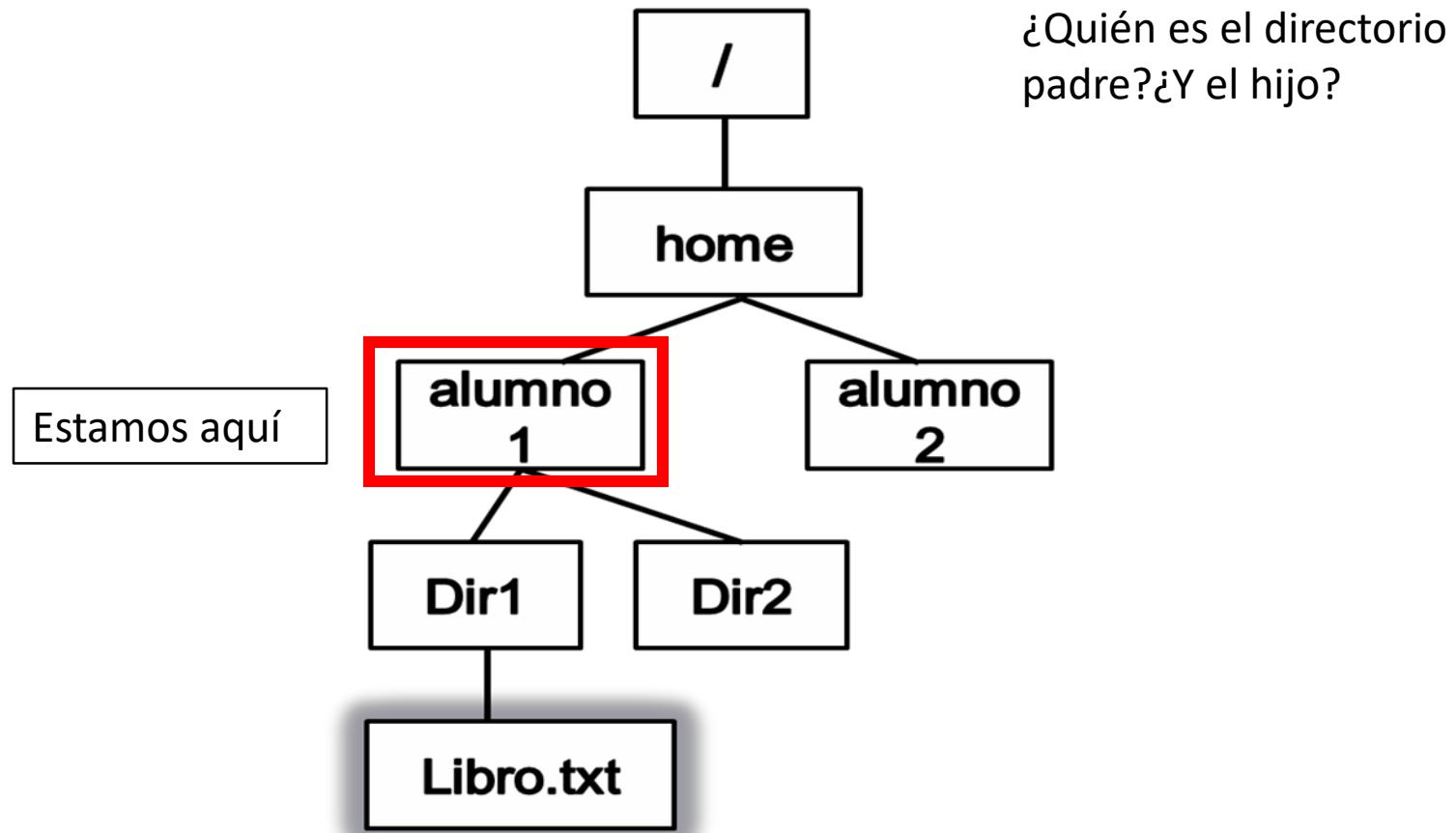
- La ruta o path de un fichero o directorio es la secuencia de directorios que se ha de recorrer para acceder a él, separados por /, es decir es la dirección donde se encuentra el fichero o directorio.
- El nombre (ruta de acceso a un fichero) puede especificarse de dos modos:
 - **Ruta absoluta:** Tomando como punto de partida el raíz
 - `/home/alumno1/dir1/libro.txt`
 - **Ruta relativa:** Tomando como punto de partida el directorio actual
 - `./alumno1/dir1/libro.txt` (desde `/home`)
 - `dir1/libro.txt` (desde `/home/alumno1`)
 - `Libro.txt` (desde `/home/alumno1/dir1`)

Rutas de acceso

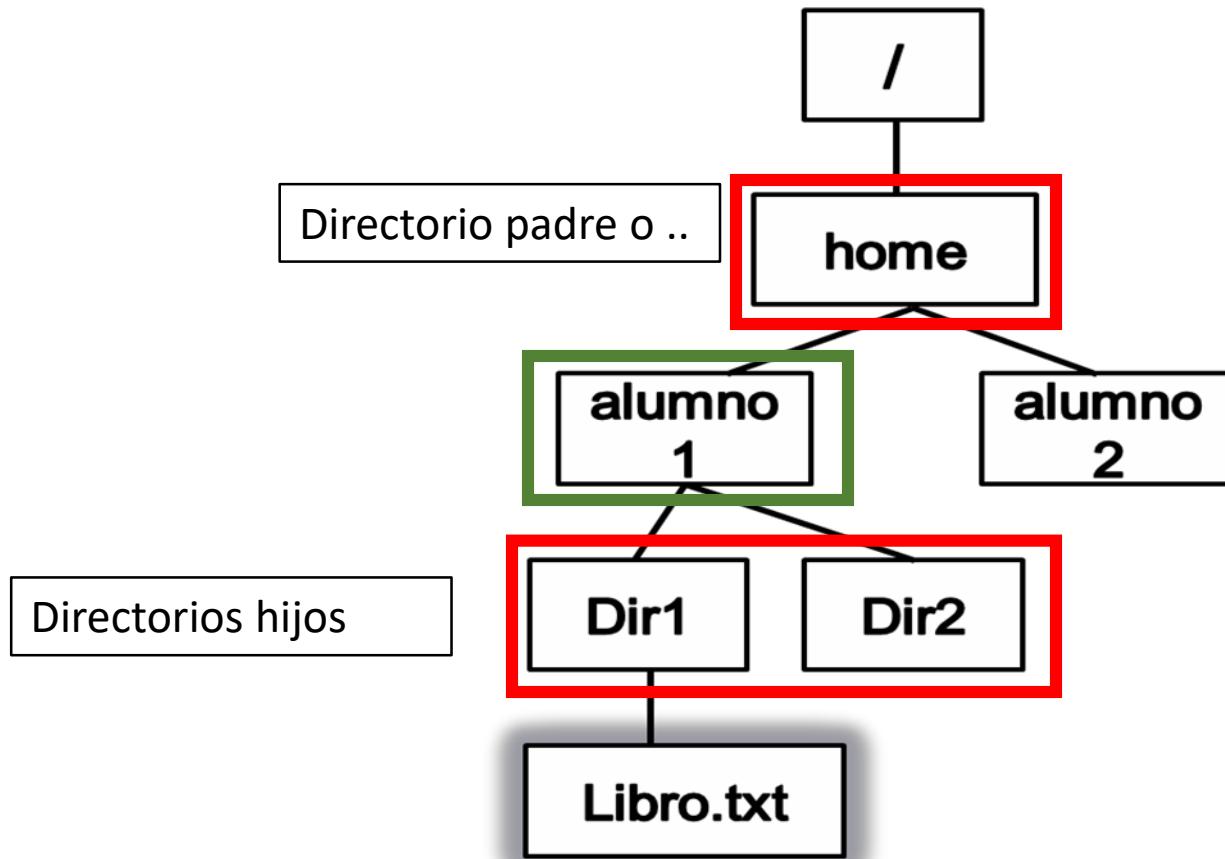
- El directorio actual, se representa por el punto “.”.
- El directorio padre se representa por dos puntos “..”
- Es mas rápido usar “..” para apuntar a un fichero o directorio que en un nivel superior. Ejemplos
 - `../../` Para ir al home desde dir1 o dir2
 - `../../alumno2` Para ir a alumno2 desde dir1 o dir2.
 - `../dir2` Para ir desde dir1 a dir2



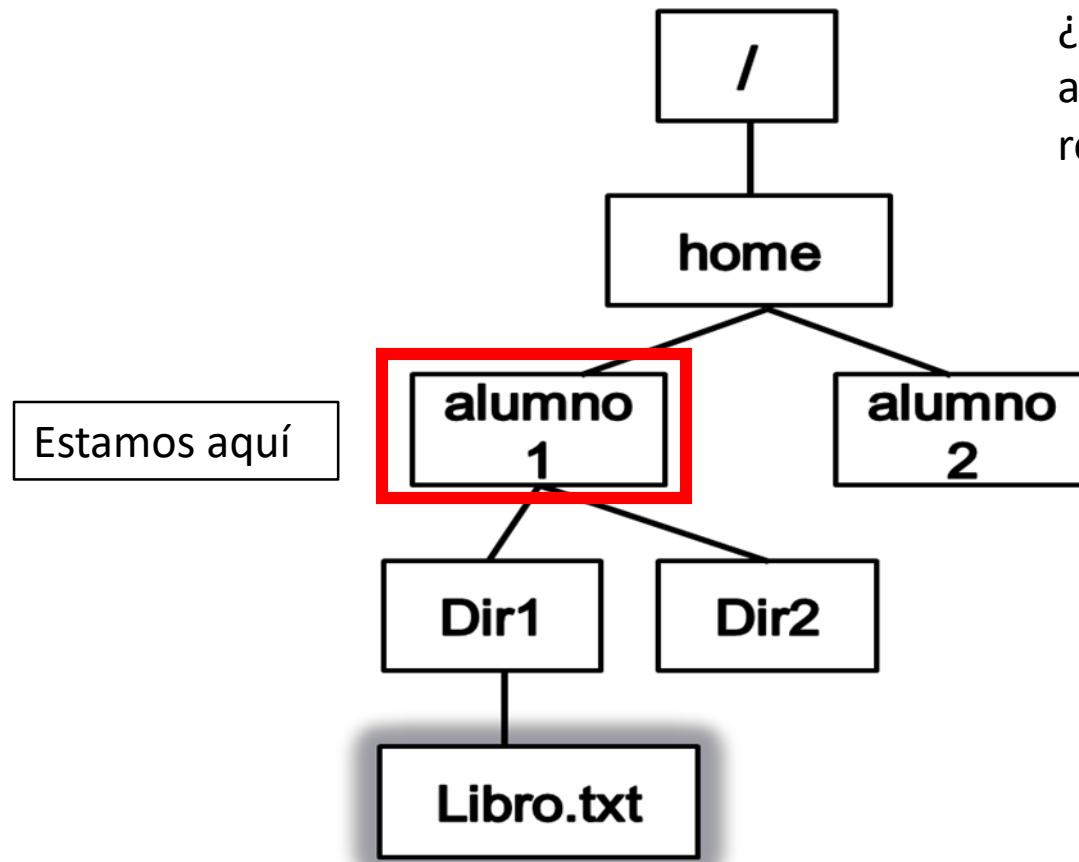
Rutas de acceso



Rutas de acceso

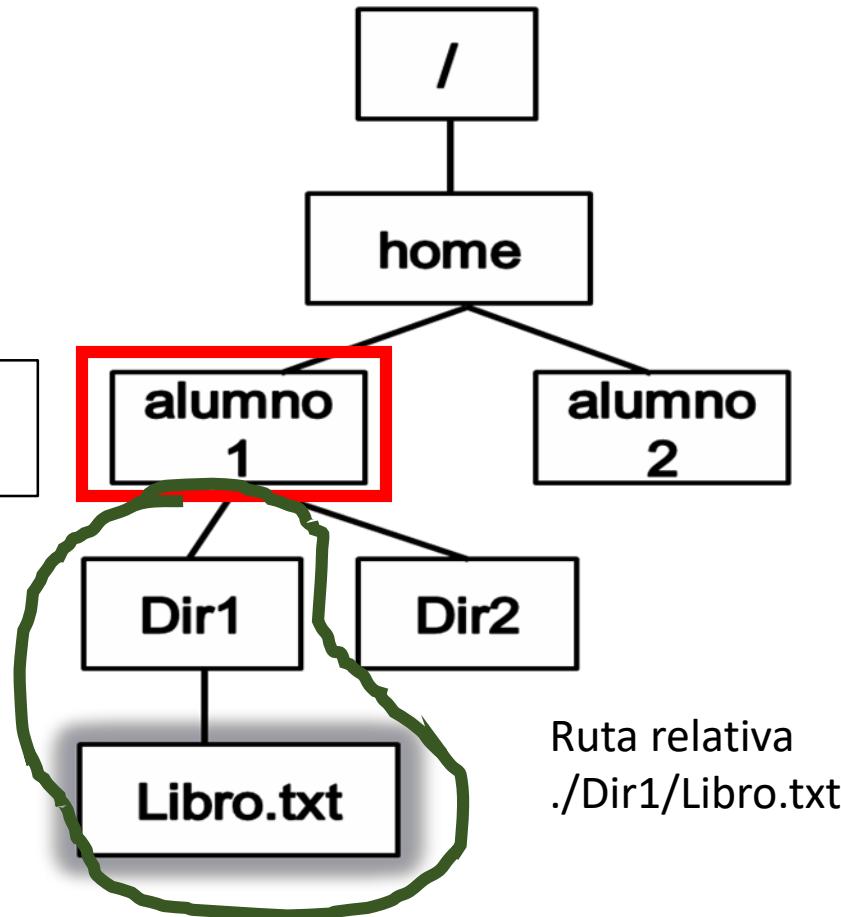


Rutas de acceso

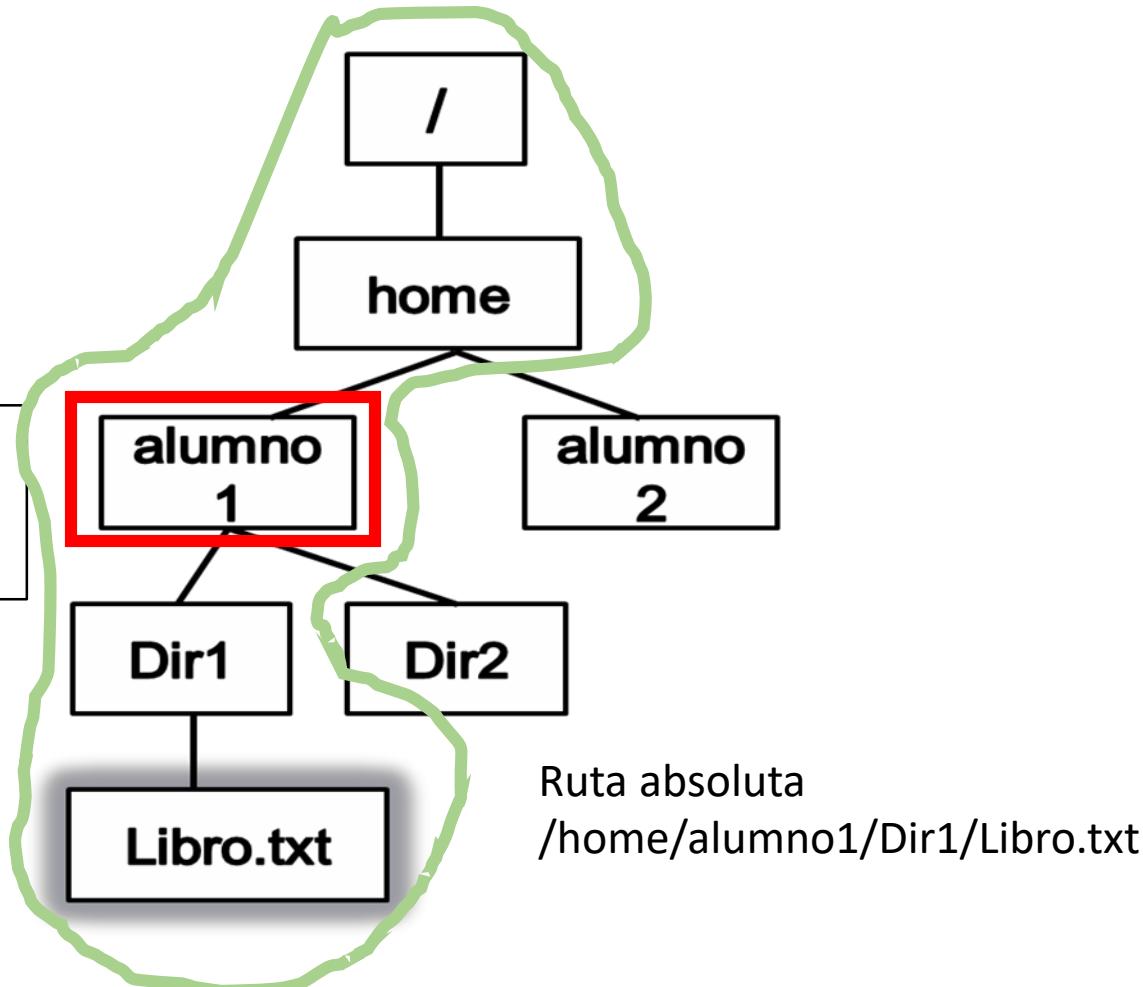


¿Cuál es la ruta absoluta a Libro.txt? ¿Y la ruta relativa?

Rutas de acceso

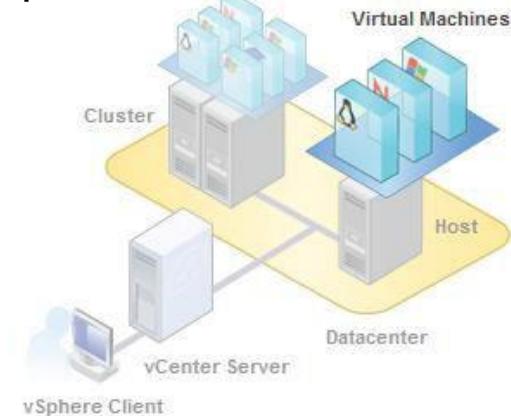


Rutas de acceso



Virtualización

- Mecanismo de abstracción entre los recursos hardware de una máquina y el o los sistemas operativos instalados. Esto se realiza mediante la interfaz llamada Hipervisor
- La virtualización de servidores, permite ‘particionar’ un servidor físico en varios servidores virtuales
- Diferentes máquinas virtuales pueden ejecutar diferentes sistemas operativos y múltiples aplicaciones al mismo tiempo utilizando un solo equipo físico



Resumen Recursos Servidor Bioinfo01

bioinfo01.isciii.es VMware ESXi, 5.5.0, 1331820

Getting Started | Summary | Virtual Machines | Resource Allocation | **Performance** | Configuration | Tasks & Events | Alarms | Permissions

General	
Manufacturer:	HP
Model:	ProLiant DL385 G7
CPU Cores:	32 CPUs x 2,199 GHz
Processor Type:	AMD Opteron(TM) Processor 6274
Processor Sockets:	2
Cores per Socket:	16
Logical Processors:	32
Hyperthreading:	Inactive
Number of NICs:	4
State:	Connected
Virtual Machines and Templates:	34
vMotion Enabled:	No
VMware EVC Mode:	Disabled 
vSphere HA State	 N/A
Host Configured for FT:	No 
Active Tasks:	
Host Profile:	
Image Profile:	ESXi-5.5.0-1331820-standard
Profile Compliance:	 N/A
DirectPath I/O:	Not supported 

Storage	Status	Drive Type
 Local Store	 Warning	Non-SSD
 		

Network	Type	Status
 Inside_bioinformat..	Standard port group	
 		

Resources	
CPU usage: 8753 MHz	Capacity 32 x 2,199 GHz
Memory usage: 98296,00 MB	Capacity 131021,70 MB

Fault Tolerance	
Fault Tolerance Version:	5.0.0-5.0.0-5.0.0
Total Primary VMs:	--
Powered On Primary VMs:	--
Total Secondary VMs:	--
Powered On Secondary VMs:	--

Máquinas virtuales Servidor Bioinfo01

bioinfo01.isciii.es VMware ESXi, 5.5.0, 1331820								
Name	State	Status	Provisioned Space	Used Space	Host CPU - MHz	Host Mem - MB	Alarm Actions	
Ideafix16_CursoNGS	Powered On	Normal	48,00 GB	23,63 GB	21	5134	Enabled	
Ideafix19_CursoNGS	Powered On	Normal	48,00 GB	23,53 GB	21	4523	Enabled	
Ideafix21_CursoNGS	Powered On	Normal	48,00 GB	23,53 GB	21	5849	Enabled	
Ideafix14_CursoNGS	Powered On	Normal	48,00 GB	23,55 GB	21	7056	Enabled	
Ideafix17_CursoNGS	Powered On	Normal	48,00 GB	23,61 GB	153	4001	Enabled	
Ideafix20_CursoNGS	Powered On	Normal	48,00 GB	23,53 GB	21	3711	Enabled	
Ideafix13_CursoNGS	Powered On	Normal	48,00 GB	23,50 GB	109	4172	Enabled	
Asterix01	Powered On	Normal	90,12 GB	90,12 GB	21	20485	Enabled	
Ideafix23_CursoNGS	Powered On	Normal	48,00 GB	23,44 GB	21	1428	Enabled	
Ideafix11_CursoNGS	Powered On	Normal	48,00 GB	23,77 GB	2924	7278	Enabled	
Ideafix15_CursoNGS	Powered On	Normal	48,00 GB	23,55 GB	21	8030	Enabled	
Ideafix18_CursoNGS	Powered On	Normal	48,00 GB	23,57 GB	21	5305	Enabled	
Ideafix03	Powered On	Normal	65,12 GB	50,44 GB	43	6165	Enabled	
Ideafix22_CursoNGS	Powered On	Normal	48,00 GB	23,49 GB	21	4544	Enabled	
Panoramix01	Powered On	Normal	4,74 TB	4,74 TB	2462	2531	Enabled	
Ideafix12_CursoNGS	Powered On	Normal	48,00 GB	23,54 GB	2133	5236	Enabled	
Asterix_CentOs_6_5_basico	Powered Off	Normal	64,56 GB	5,15 GB	0	0	Enabled	
panoramix04_swat	Powered Off	Normal	60,32 GB	6,02 GB	0	0	Enabled	
Ideafix_basico_Ubuntu_1404...	Powered Off	Normal	58,47 GB	8,18 GB	0	0	Enabled	
Ideafix_Ubuntu_1404_Unity_n...	Powered Off	Normal	58,47 GB	8,18 GB	0	0	Enabled	
Asterix-bio	Powered Off	Normal	92,62 GB	60,00 GB	0	0	Enabled	
Bio-linux_curso	Powered Off	Normal	33,33 GB	11,78 GB	0	0	Enabled	
Asterix03_H	Powered Off	Normal	64,56 GB	7,99 GB	0	0	Enabled	
Panoramix03	Powered Off	Normal	60,32 GB	5,09 GB	0	0	Enabled	
biowebserver01	Powered Off	Normal	52,21 GB	7,93 GB	0	0	Enabled	
Asterix02	Powered Off	Normal	82,47 GB	26,52 GB	0	0	Enabled	
Panoramix02	Powered Off	Normal	52,25 GB	50,00 GB	0	0	Enabled	
Ideafix05_nolvm	Powered Off	Normal	71,30 GB	15,60 GB	0	0	Enabled	
Ideafix_Curso	Powered Off	Normal	33,33 GB	15,17 GB	0	0	Enabled	
Asterix_CentOS_6_5_basico_t...	Powered Off	Normal	64,56 GB	5,15 GB	0	0	Enabled	
Ideafix_Ubuntu1404_CURSON...	Powered Off	Normal	33,33 GB	15,17 GB	0	0	Enabled	
Amnesix01	Powered Off	Normal	50,29 GB	8,37 GB	0	0	Enabled	
Ideafix04	Powered Off	Normal	60,47 GB	50,00 GB	0	0	Enabled	
Ideafix03_curso	Powered Off	Normal	91,04 GB	32,74 GB	0	0	Enabled	

Containers I

Linux containers is a generic term for an implementation of operating system-level virtualization for the Linux operating system.

Containers allow us to port pipelines and replicate their exact execution environments across different hardware.

Currently, a number of such implementations exist, and they are all based on the virtualization, isolation, and resource management mechanisms provided by the Linux kernel.

Containers II

Singularity is a free, cross-platform and open-source computer program that performs operating-system-level virtualization.

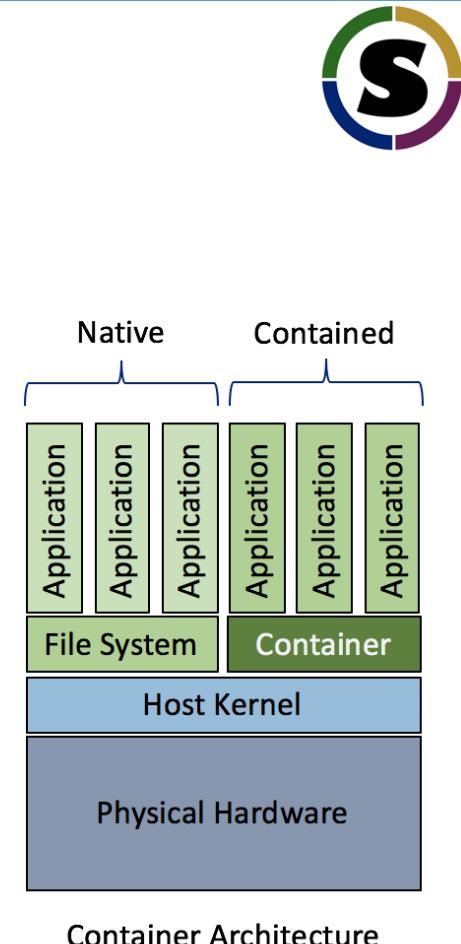
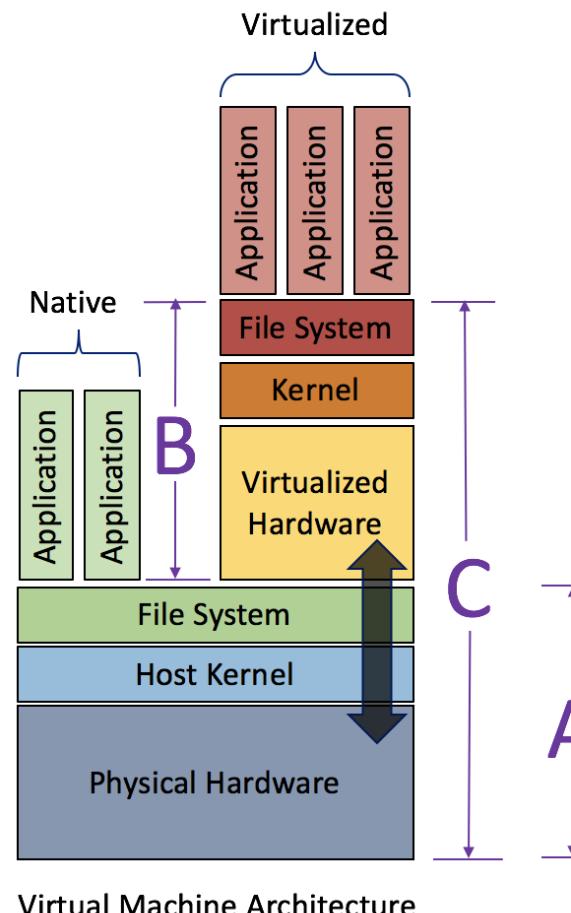
One of the main uses of Singularity is to bring containers and reproducibility to scientific computing and the HPC world.

While Docker is broadly used, Singularity is fully compatible with Docker, plus Singularity does not require root permissions to be executed.

Singularity I



- Applications running within a container will always be “closer” to the physical hardware
 - Notice how close to native a container behaves
- Applications running through a virtual machine will always have multiple levels of indirection
- The container’s proximity to the physical hardware equates to less overhead, higher performance and lower latency



Singularity II

Singularity image runs on the same level as the OS, directly above the kernel, and can access all hardware in the machine.

Not needing to virtualise the hardware and run a kernel again makes this kind of virtualisation really effective.

Filesystem is shared, and some paths are automatically mounted (`/tmp` and `/home`), while the others are optional.

Files of the host system can be created, modified and deleted from the image in the mounted folders.

Día 2

- Nociones básicas comandos
- Línea de comandos
- Sintaxis de los comando
- Ayuda de los comandos
- Historial
- Privilegios administrador
- Comandos uso general
- Comandos manejo de archivos y directorios
- Comandos visualización contenido de archivos
- Redirecciones

Ficheros

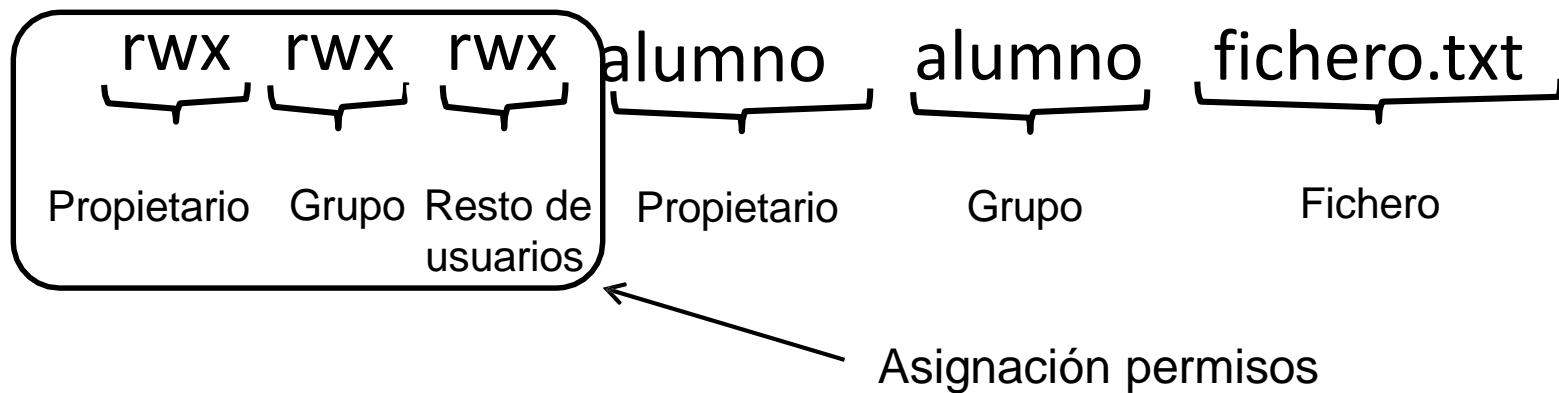
- No tienen extensión obligatoria y la extensión no tiene significado intrínseco
- Si el nombre comienza por “.” permanecen ocultos
- Poseen un propietario y pertenecen a un grupo
- Tipos:
 - Regulares (-)
 - Directorios (d)
 - Enlaces: Simbólicos o blandos (l), Físicos o duros (-)
 - De dispositivo (b, c)

Permisos

- La forma de definir que puede y que no puede hacer un usuario es a través de los permisos de los ficheros.
- Cada fichero tiene asociados TRES tipos de permisos diferentes:
 - **Lectura (r)**: permite ver el contenido del fichero
 - **Escritura (w)**: permite alterar el contenido del fichero
 - **Ejecución (x)**: permite ejecutar el fichero (en un directorio permite acceder a él)
- Se diferencian tres roles de usuarios respecto a ficheros y directorios:
 - Propietario
 - Miembro del grupo del fichero
 - Resto de usuarios

Permisos

- Los permisos de un fichero solo pueden ser alterados por su propietario, usuarios que pertenezcan al grupo y por el administrador
- El sistema almacena estos permisos como una secuencia de 9 bits
- Esta secuencia tiene una estructura en tres grupos: para el propietario, para el grupo del fichero y para los demás



Permisos

- Ejemplo 1: **directorio opt**
 - **drwxr-xr-x root root opt**
 - El propietario puede leer, modificar y ejecutar el fichero.
 - Su grupo y el resto solo leerlo y ejecutarlo.
- Ejemplo 2: **directorio personal**
 - **drwx----- alumno clase alumno**
 - El propietario puede ver, modificar y acceder al contenido del directorio.
 - Su grupo y el resto no pueden hacer nada.
- Ejemplo 3: **directorio tmp**
 - **drwxrwxrwx root root tmp**
 - Todo el mundo puede ver, modificar y acceder al directorio.

Cuentas de usuario

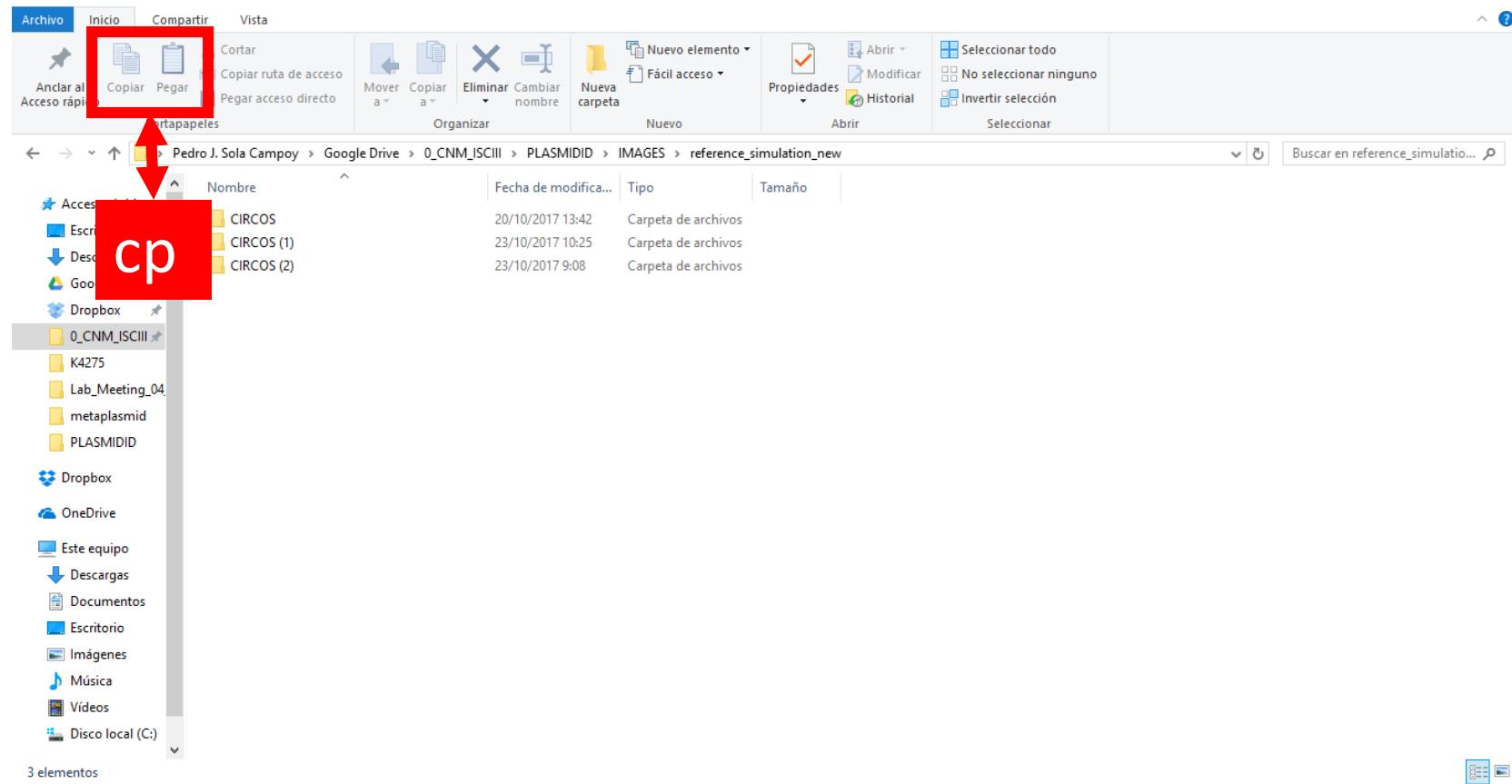
- Para acceder a cualquier sistema Linux necesitamos una **cuenta de usuario**
- La cuenta de usuario engloba los siguientes elementos
 - Nombre de usuario (Login)
 - Contraseña (Password)
 - Identificador de usuario y grupo (UID y GID)
 - Directorio de trabajo (Home directory)
 - Intérprete de órdenes (Shell)
- Al acceder con nuestra cuenta de usuario automáticamente nos ubicamos en nuestro directorio de trabajo (**home**)
 - Un usuario suele tener permiso total de acceso a todos los archivos y subdirectorios de su directorio de trabajo
 - El acceso a otros directorios que no pertenezcan al directorio de trabajo del usuario suele estar limitado o incluso prohibido

Cuentas de usuario

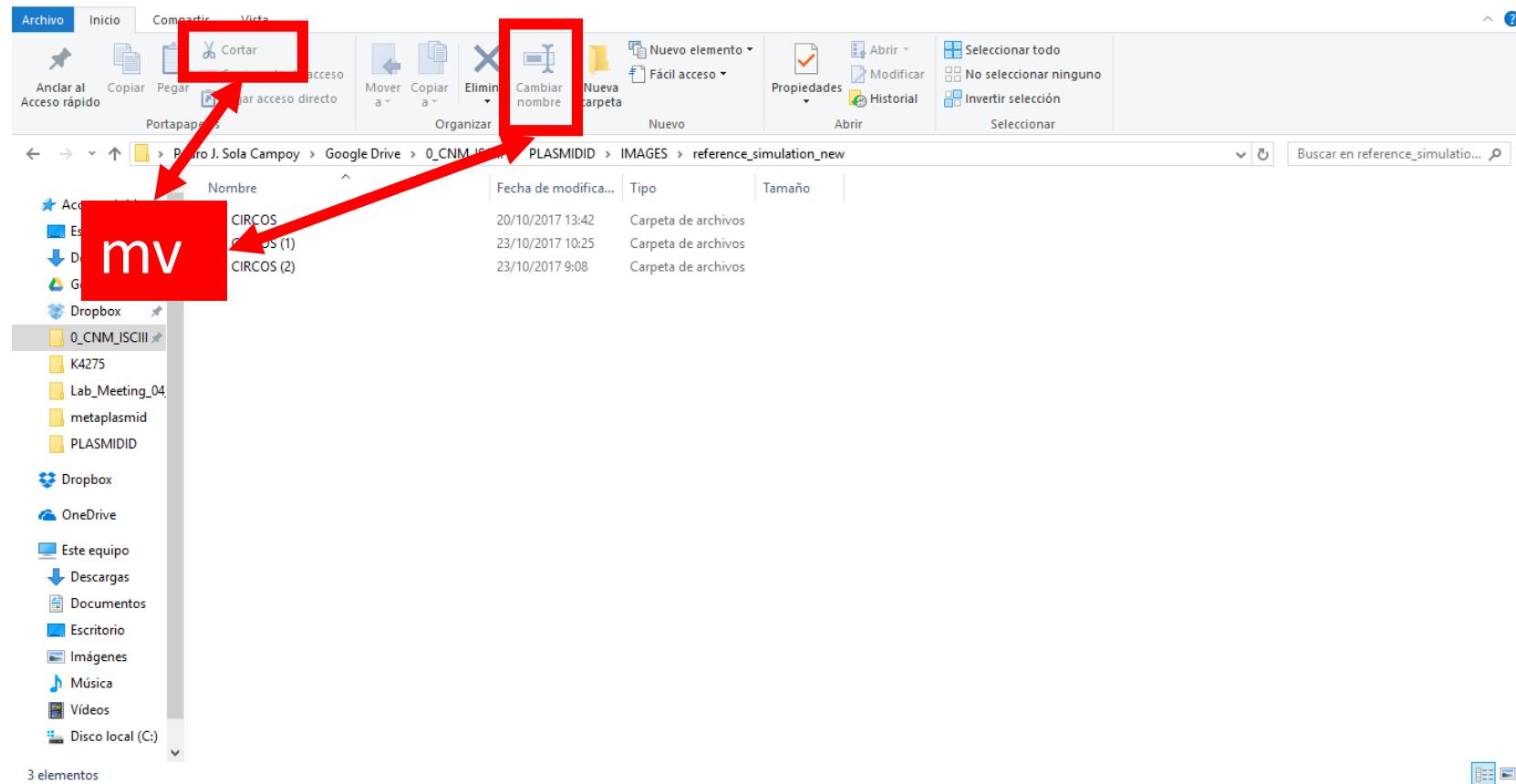
- **La cuenta del superusuario o root**

- El superusuario es un usuario especial que actúa como administrador del sistema
- Tiene acceso a todos los archivos y directorios del sistema
- Tiene capacidad para crear nuevos usuarios o eliminar usuarios
- Puede detener cualquier proceso que se esté ejecutando en el sistema
- Tiene capacidad para detener y reiniciar el sistema
- El **Login** del superusuario suele ser **root**, aunque no es estrictamente necesario

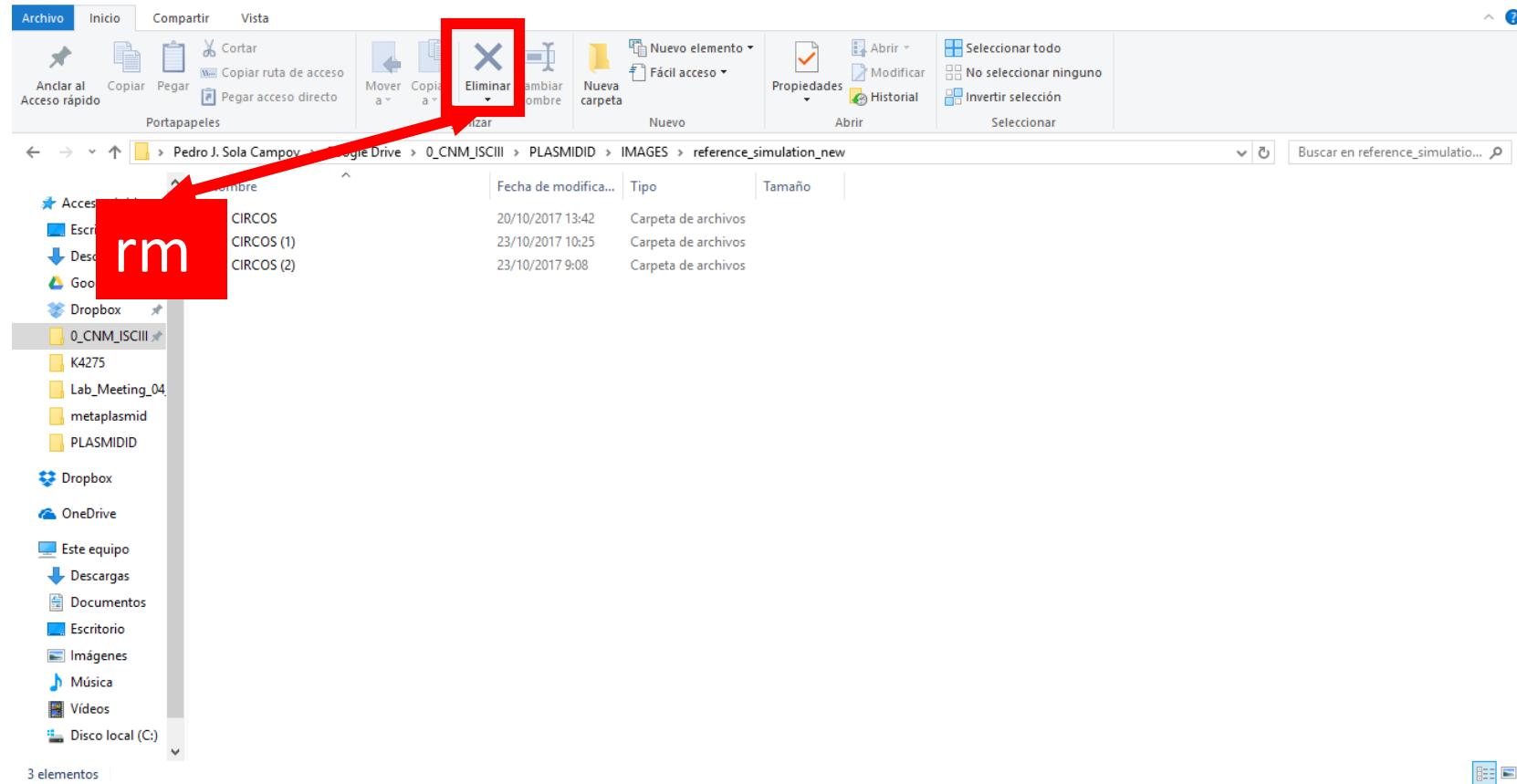
Nociones básicas



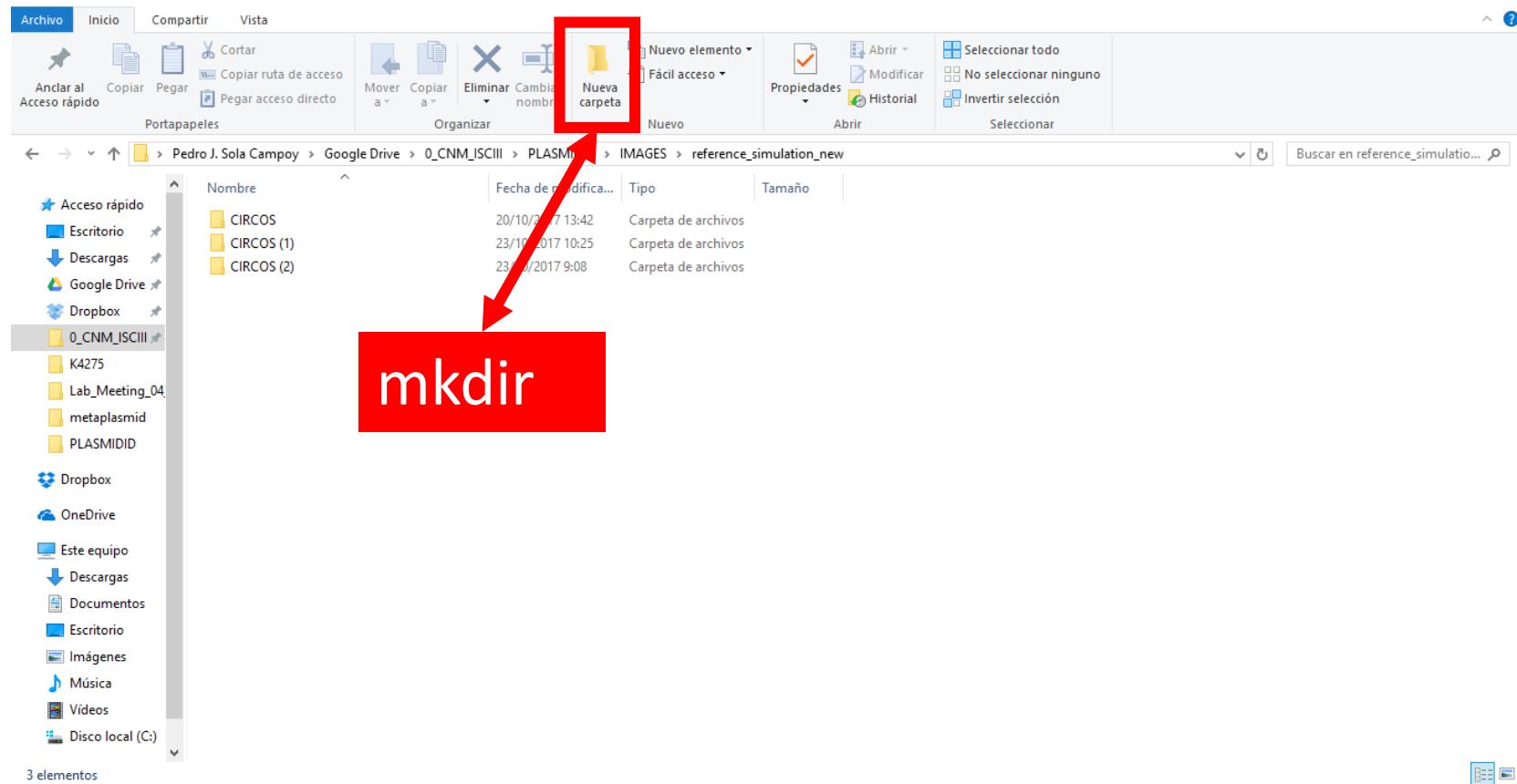
Nociones básicas



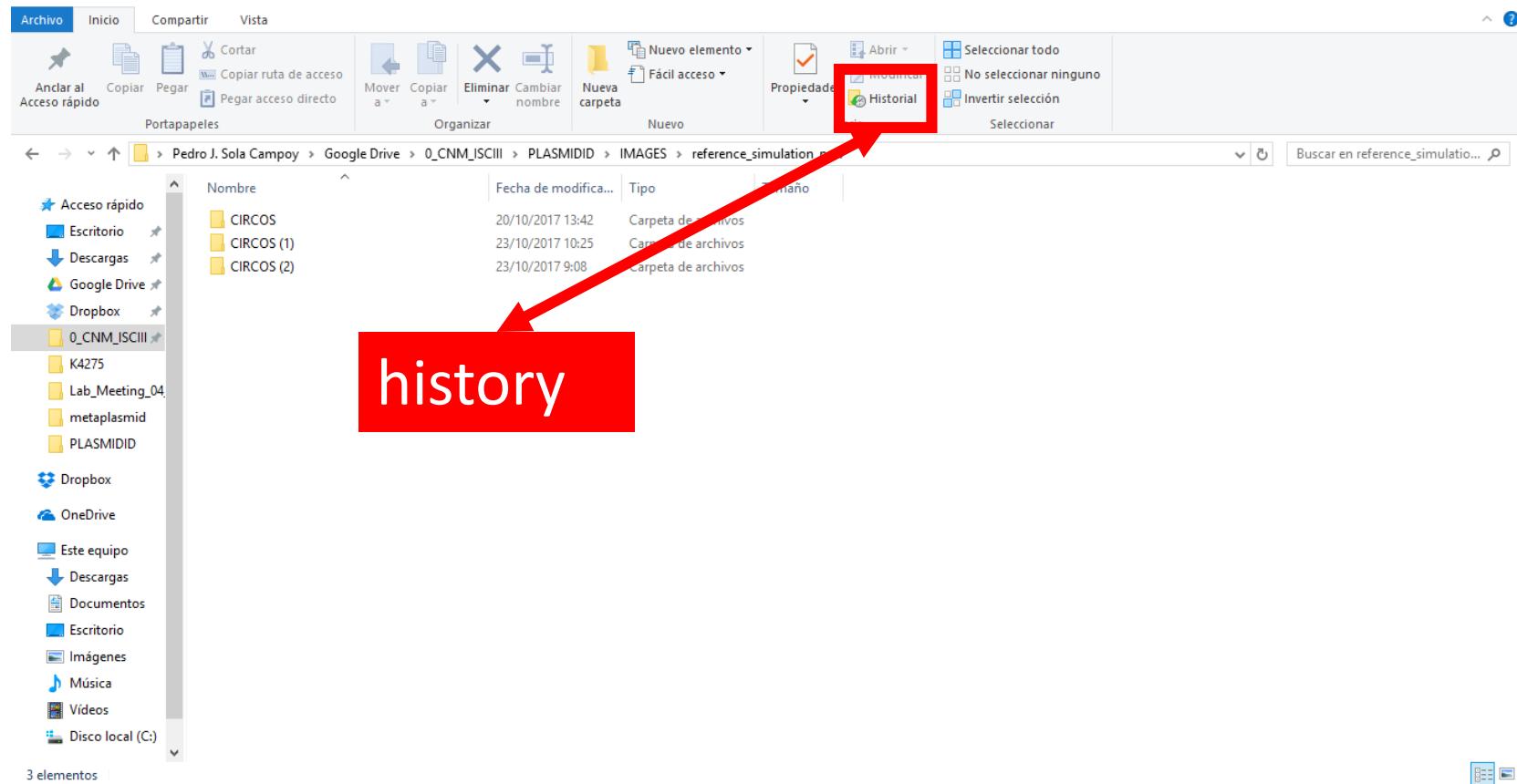
Nociones básicas



Nociones básicas



Nociones básicas



Nociones básicas

- Los comandos hay que teclearlos exactamente
- Las letras mayúsculas y minúsculas se consideran como diferentes (case sensitive)
- Cuando se introduce un comando o ruta de fichero o directorio, *la Shell* permite escribir las primeras letras del mismo y realiza un autorellenado al presionar la tecla del tabulador. Si no puede distinguir entre diversos casos rellenará hasta el punto en el que se diferencien

Línea de comandos

- El Shell espera instrucciones por el teclado en una línea llamada **línea de comandos** o **prompt**
- Esta línea de comandos nos ofrece cierta información fácilmente reconocible:

```
profesor@VM-NGS01:~/Documentos$
```

\$ sin privilegios
con privilegios

Ubicación actual en el árbol de directorios

Hostname (Nombre de la máquina) Usuario conectado

- Esta línea de comandos nos informa de que el usuario profesor sin privilegios de administración esta conectado a la maquina lógica VM-NGS01 en la posición del sistema de ficheros Documentos

Sintaxis de los comandos

- En general la sintaxis de los comandos es:

Comando [-o | --opción] [argumentos]

- Los **parámetros** son opciones del comando y normalmente se escriben mediante un guion y una letra (-r por ejemplo). Se puede ejecutar mas de un parámetro por instrucción escribiendo guion y letra varias veces (-r -a) o uniendo las letras detrás del guion (-ra)
- Algunos parámetros o instrucciones requieren un **argumento**, por ejemplo el nombre de un fichero, que se colocara detrás del parámetro separado por un espacio
- Para escribir más de un comando en la misma línea los separaremos con el carácter “;”
 - Ejemplo: \$ *mkdir dir1* (este comando creara la carpeta dir1 en la ubicación actual de el árbol de directorios)

Sintaxis de los comandos

- Añaden infinita flexibilidad a la invocación de comandos
- La shell los sustituye por otra expresión

?	Único carácter
*	Cadena de caracteres
~	Abreviatura del directorio home
[x-y]	Algún carácter dentro del rango

- Por ejemplo:
 - *.txt representa todos los ficheros que terminen en “.txt”

Ayuda de los comandos

- Es imposible conocer cada uno de los parámetros y argumentos de todas las instrucciones (comandos)
- Todos los manuales de Linux están dentro del propio sistema operativo
- Existen páginas con información completa sobre los comandos. Para acceder a ellas se usa el comando **man** o el argumento **help**. Sintaxis:
 - **man comando** (\$ *man ls*)
 - **comando --help** (\$ *ls --help*)
- Se puede navegar a través de estas páginas con los cursores del teclado.
- Presionando letra **q** (quit) se sale del manual

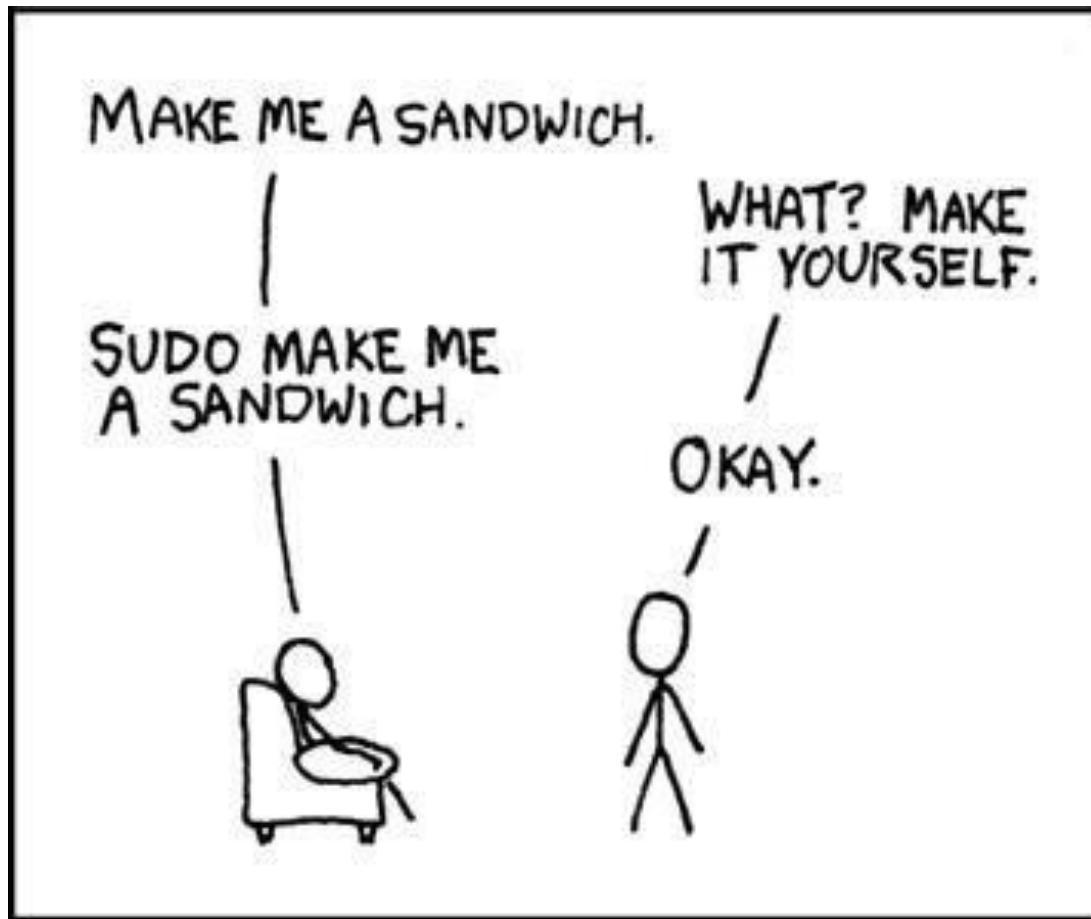
Historial

- La Shell conserva un historial de los comandos que se ejecutan.
- Se pueden consultar con el comando ***history***
- También se puede navegar por este historial pulsando la flecha para arriba o la flecha para debajo de los cursores del teclado
- Es de gran utilidad ya que permite no tener que reescribir los comandos utilizados hace poco
 - **Sintaxis:** \$ history [opciones]
 - **Opciones:**
 - -c Borra el historial

Privilegios de Administrador

- Muchos de los comandos necesitan privilegios para poder ser ejecutados
- Para llamar o ejecutar un comando que requiere derechos de administrador tendremos que colocar antes de la instrucción el comando ***sudo*** (por ejemplo `$ sudo shutdown`). Después de ejecutar instrucciones con sudo el interprete nos pedirá la contraseña de administrador
- También podemos ejecutar el comando ***sudo su*** con el que conseguiremos derechos de root (administrador), evidentemente el interprete también nos pedirá la contraseña de administrador

Privilegios de Administrador



Pipeline frameworks

- Basic scripting for automating scripts is inefficient, difficult to port and maintain, and does not include support for dependencies or reentrancy.
- High-throughput bioinformatic analyses increasingly rely on pipeline frameworks to process sequence and metadata.
- Modern implementations of these frameworks differ on three key dimensions: using an implicit or explicit syntax, using a configuration, convention or class-based design paradigm and offering a command line or workbench interface.
- Choosing the right framework depends on both your lab developers and users needs.

Nextflow I

- Nextflow is a DSL for parallel and scalable computational pipelines.
- It enables scalable and reproducible scientific workflows using software containers.
- It allows the adaptation of pipelines written in the most common scripting languages.
- Its fluent DSL simplifies the implementation and the deployment of complex parallel and reactive workflows on clouds and clusters.

Nextflow II

Fast prototyping

Nextflow allows you to write a computational pipeline by making it simpler to put together many different tasks.

You may reuse your existing scripts and tools and you don't need to learn a new language or API to start using it.

Portable

Nextflow provides an abstraction layer between your pipeline's logic and the execution layer, so that it can be executed on multiple platforms without it changing.

It provides out of the box executors for SGE, LSF, SLURM, PBS and HTCondor batch schedulers and for [Kubernetes](#) and [Amazon AWS](#) cloud platforms.

Continuous checkpoints

All the intermediate results produced during the pipeline execution are automatically tracked.

This allows you to resume its execution, from the last successfully executed step, no matter what the reason was for it stopping.

Reproducibility

Nextflow supports [Docker](#) and [Singularity](#) containers technology.

This, along with the integration of the [GitHub](#) code sharing platform, allows you to write self-contained pipelines, manage versions and to rapidly reproduce any former configuration.

Unified parallelism

Nextflow is based on the *dataflow* programming model which greatly simplifies writing complex distributed pipelines.

Parallelisation is implicitly defined by the processes input and output declarations. The resulting applications are inherently parallel and can scale-up or scale-out, transparently, without having to adapt to a specific platform architecture.

Stream oriented

Nextflow extends the Unix pipes model with a fluent DSL, allowing you to handle complex stream interactions easily.

It promotes a programming approach, based on functional composition, that results in resilient and easily reproducible pipelines.



Nextflow III

Now a simple command works out of the box in any machine:

```
nextflow run //buisciii/main.nf -profile singularity
```

Plus, it give us:

- Dependency and computing automatization
- Resource usage statistics
- Easy to share and maintain
- Reproducibility and re-entrancy
- Transparency



GOBIERNO
DE ESPAÑA

MINISTERIO
DE CIENCIA, INNOVACIÓN
Y UNIVERSIDADES



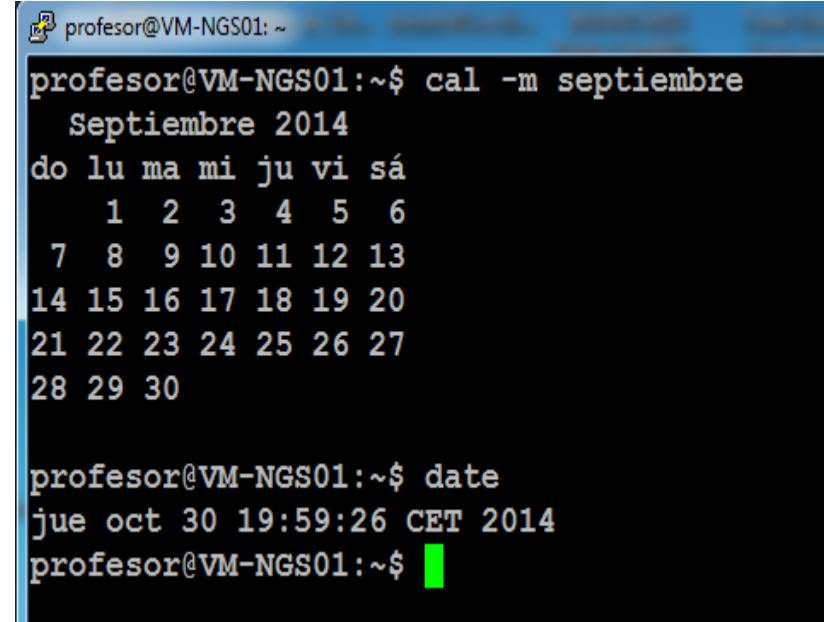
>A_BU-ISCIII

Gracias por su atención

Comandos uso general

- **cal**
 - Muestra el calendario
 - **Sintaxis:** cal [opciones]

- **date**
 - Muestra la fecha del sistema
 - **Sintaxis:** date [opciones]

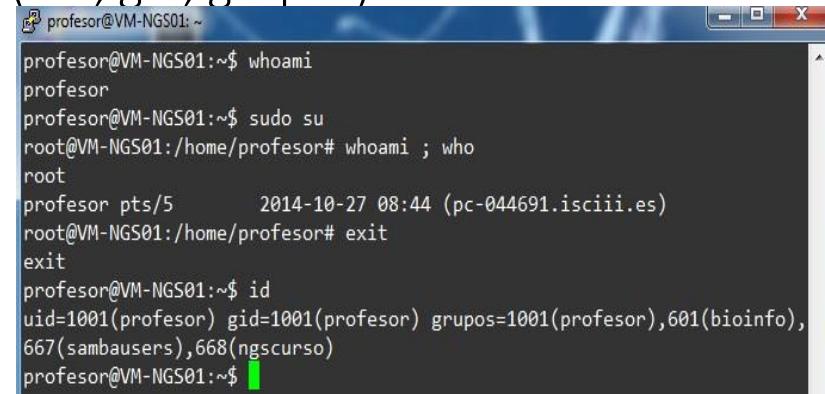


```
profesor@VM-NGS01: ~
profesor@VM-NGS01:~$ cal -m septiembre
Septiembre 2014
do lu ma mi ju vi sá
      1  2  3  4  5  6
    7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30

profesor@VM-NGS01:~$ date
jue oct 30 19:59:26 CET 2014
profesor@VM-NGS01:~$
```

Comandos uso general

- **who**
 - Muestra quien esta conectado en el sistema
 - **Sintaxis:** who [opciones]
- **whoami**
 - Muestra el nombre del usuario
 - **Sintaxis:** whoami [opciones]
- **id**
 - Muestra información del usuario (uid, gid, grupos)
 - **Sintaxis:** id [opciones]



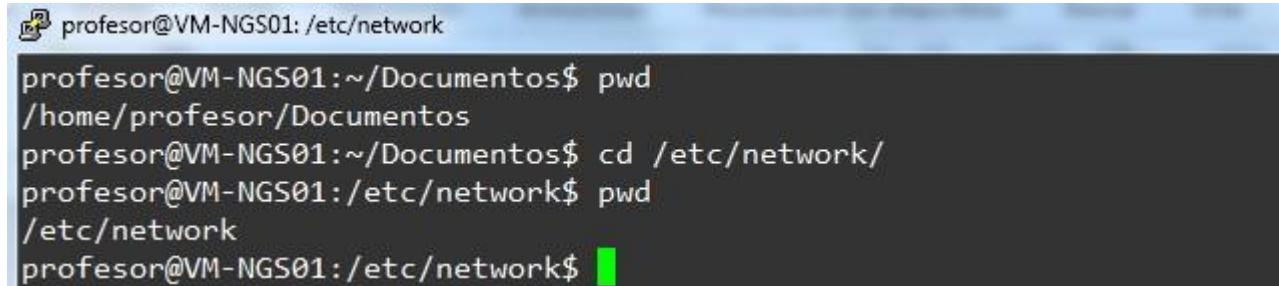
```
profesor@VM-NGS01:~$ whoami
profesor
profesor@VM-NGS01:~$ sudo su
root@VM-NGS01:/home/profesor# whoami ; who
root
profesor pts/5      2014-10-27 08:44 (pc-044691.isciii.es)
root@VM-NGS01:/home/profesor# exit
exit
profesor@VM-NGS01:~$ id
uid=1001(profesor) gid=1001(profesor) grupos=1001(profesor),601(bioinfo),
667(sambausers),668(ngscurso)
profesor@VM-NGS01:~$
```

Comandos uso general

- **clear**
 - Borra la pantalla
 - **Sintaxis:** clear
 - Idem Ctrl + l
- **exit**
 - Termina la sesión de un usuario o sale de la terminal en modo texto
 - **Sintaxis:** exit
- **echo**
 - Muestra en pantalla lo que indicamos en el argumento(cadena de caracteres)
 - **Sintaxis:** echo [opciones] [cadena de caracteres]

Comandos uso general

- **pwd** (print working directory)
 - Muestra la ruta absoluta del directorio en la cual nos encontramos actualmente.
 - **Sintaxis :** pwd

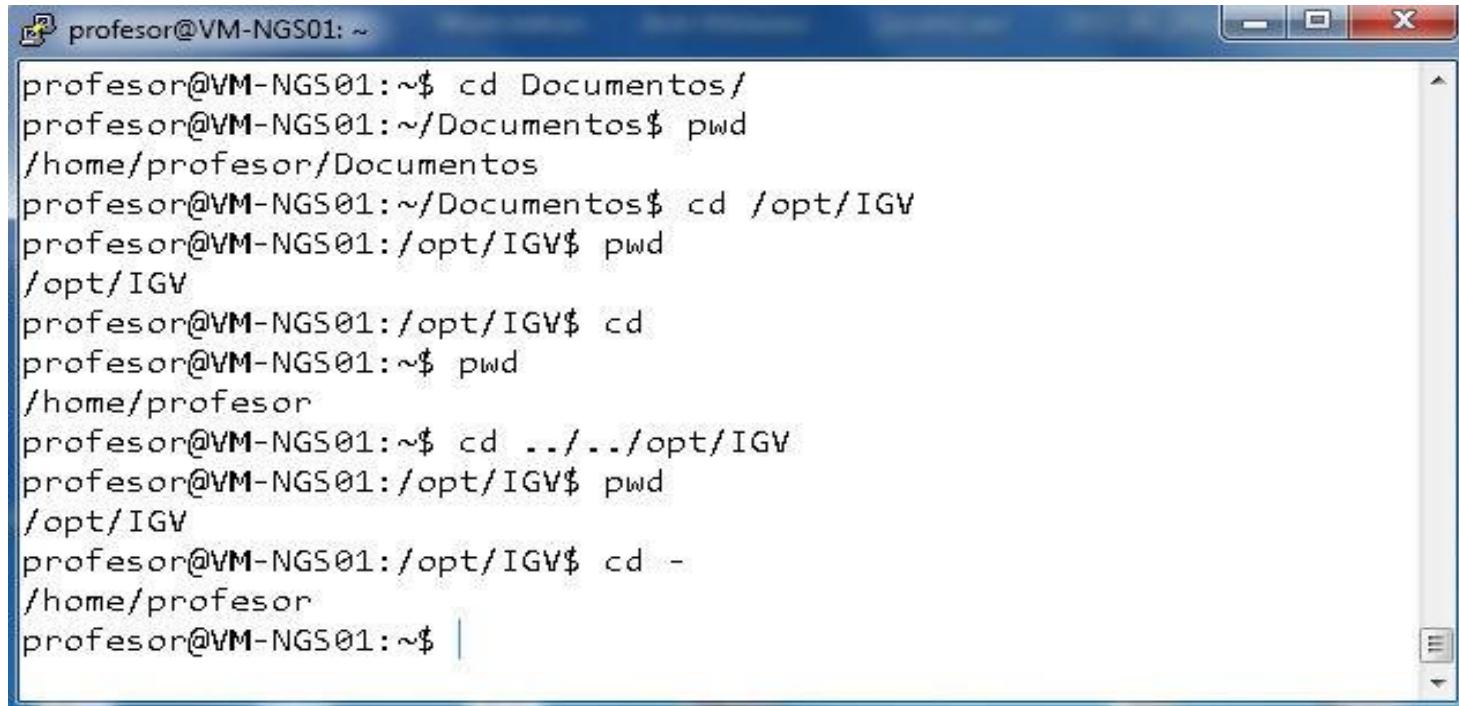


```
profesor@VM-NGS01: /etc/network
profesor@VM-NGS01:~/Documentos$ pwd
/home/profesor/Documentos
profesor@VM-NGS01:~/Documentos$ cd /etc/network/
profesor@VM-NGS01:/etc/network$ pwd
/etc/network
profesor@VM-NGS01:/etc/network$
```

Comandos manejo de archivos y directorios

- **cd** (change directory)
 - El comando cd se utiliza para cambiar de directorios
 - **Sintaxis:** cd [opciones][directorio]
 - **opciones:**
- Retorna al directorio donde se encontraba anteriormente
- .. Cambia al directorio padre (retrocede un directorio)
- ~ Cambia al directorio base del usuario actual
- Sin argumentos cambia al directorio base del usuario actual

Comandos manejo de archivos y directorios

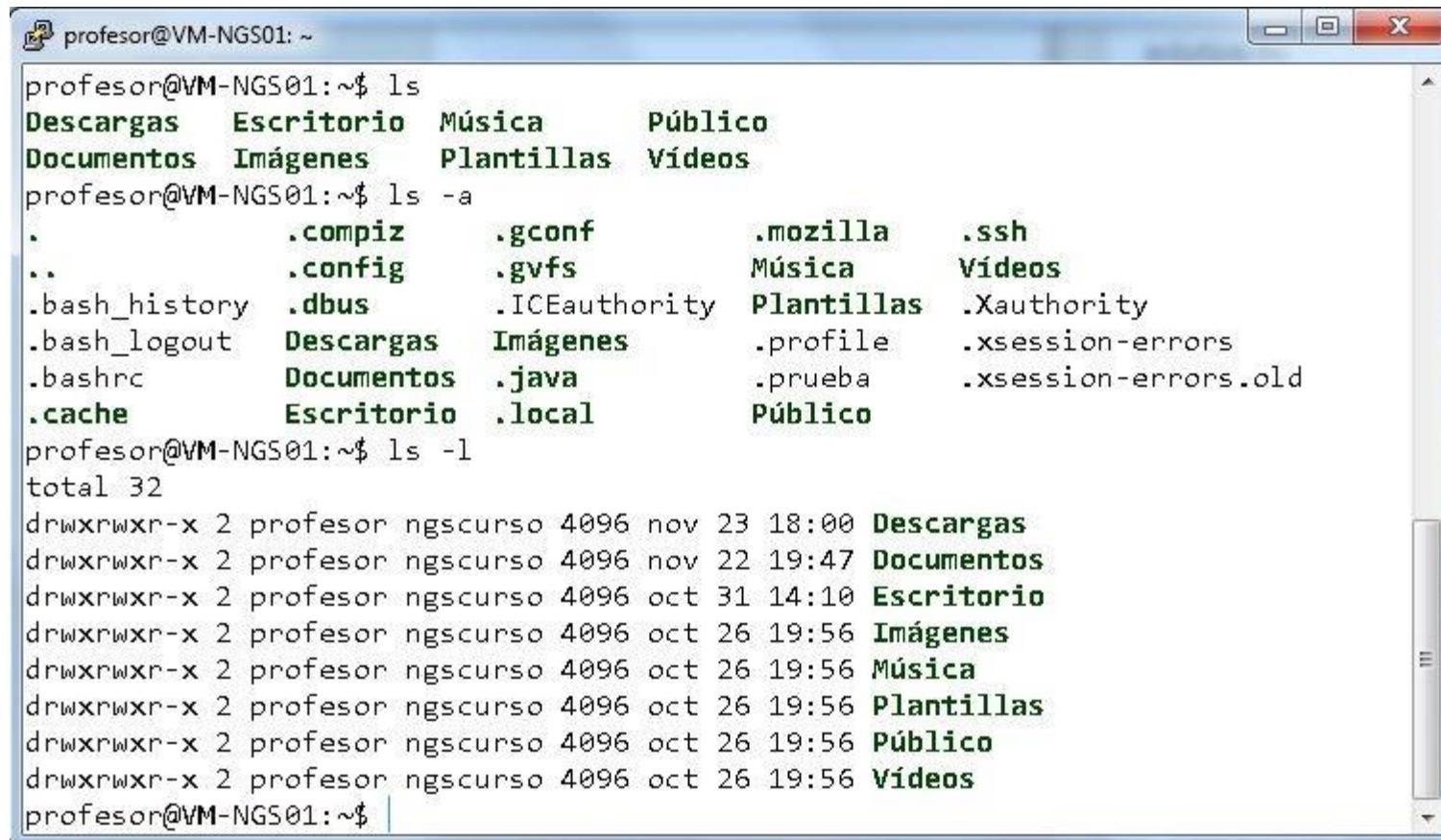


```
profesor@VM-NGS01: ~$ cd Documentos/
profesor@VM-NGS01: ~/Documentos$ pwd
/home/profesor/Documentos
profesor@VM-NGS01:~/Documentos$ cd /opt/IGV
profesor@VM-NGS01:/opt/IGV$ pwd
/opt/IGV
profesor@VM-NGS01:/opt/IGV$ cd
profesor@VM-NGS01:~$ pwd
/home/profesor
profesor@VM-NGS01:~$ cd ../../opt/IGV
profesor@VM-NGS01:/opt/IGV$ pwd
/opt/IGV
profesor@VM-NGS01:/opt/IGV$ cd -
/home/profesor
profesor@VM-NGS01:~$ |
```

Comandos manejo de archivos y directorios

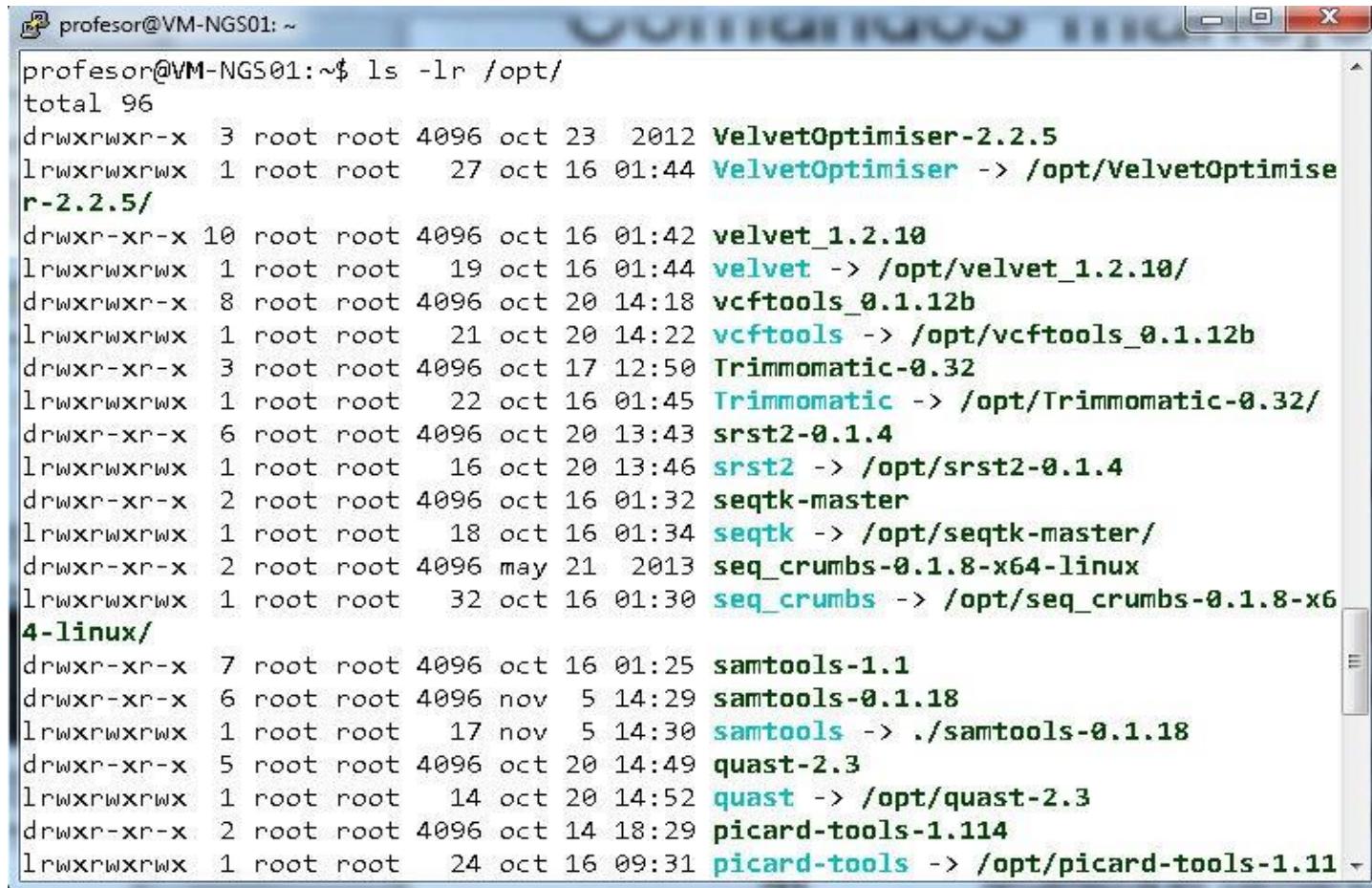
- ls (List)
 - Muestra información sobre ficheros y directorios. Si no se especifica nada muestra información sobre el directorio actual
 - Sintaxis: ls [opciones] [directorio|archivo]
 - opciones:
 - l Muestra la salida en formato largo, con permisos ...
 - r Invierte el orden de listado de los archivos (forma descendente)
 - a Lista todos los archivos, incluyendo aquellos que empiezan con un punto, que representa a los archivos ocultos
 - h Muestra el tamaño de archivo en kilobytes, megabytes, etc
 - t Lista los archivos ordenados por tiempo de modificación Ordena el listado por el tamaño del archivo
 - s

Comandos manejo de archivos y directorios



```
profesor@VM-NGS01: ~
profesor@VM-NGS01:~$ ls
Descargas E Escritorio Música Público
D Documentos Imágenes Plantillas Vídeos
profesor@VM-NGS01:~$ ls -a
. .compiz .gconf .mozilla .ssh
.. .config .gvfs Música Vídeos
.bash_history .dbus .ICEauthority Plantillas .Xauthority
.bash_logout D Descargas Imágenes .profile .xsession-errors
.bashrc D Documentos .java .prueba .xsession-errors.old
.cache E Escritorio .local P blico
profesor@VM-NGS01:~$ ls -l
total 32
drwxrwxr-x 2 profesor ngscurso 4096 nov 23 18:00 Descargas
drwxrwxr-x 2 profesor ngscurso 4096 nov 22 19:47 Documentos
drwxrwxr-x 2 profesor ngscurso 4096 oct 31 14:10 E Escritorio
drwxrwxr-x 2 profesor ngscurso 4096 oct 26 19:56 Imágenes
drwxrwxr-x 2 profesor ngscurso 4096 oct 26 19:56 Música
drwxrwxr-x 2 profesor ngscurso 4096 oct 26 19:56 Plantillas
drwxrwxr-x 2 profesor ngscurso 4096 oct 26 19:56 Público
drwxrwxr-x 2 profesor ngscurso 4096 oct 26 19:56 Vídeos
profesor@VM-NGS01:~$ |
```

Comandos manejo de archivos y directorios

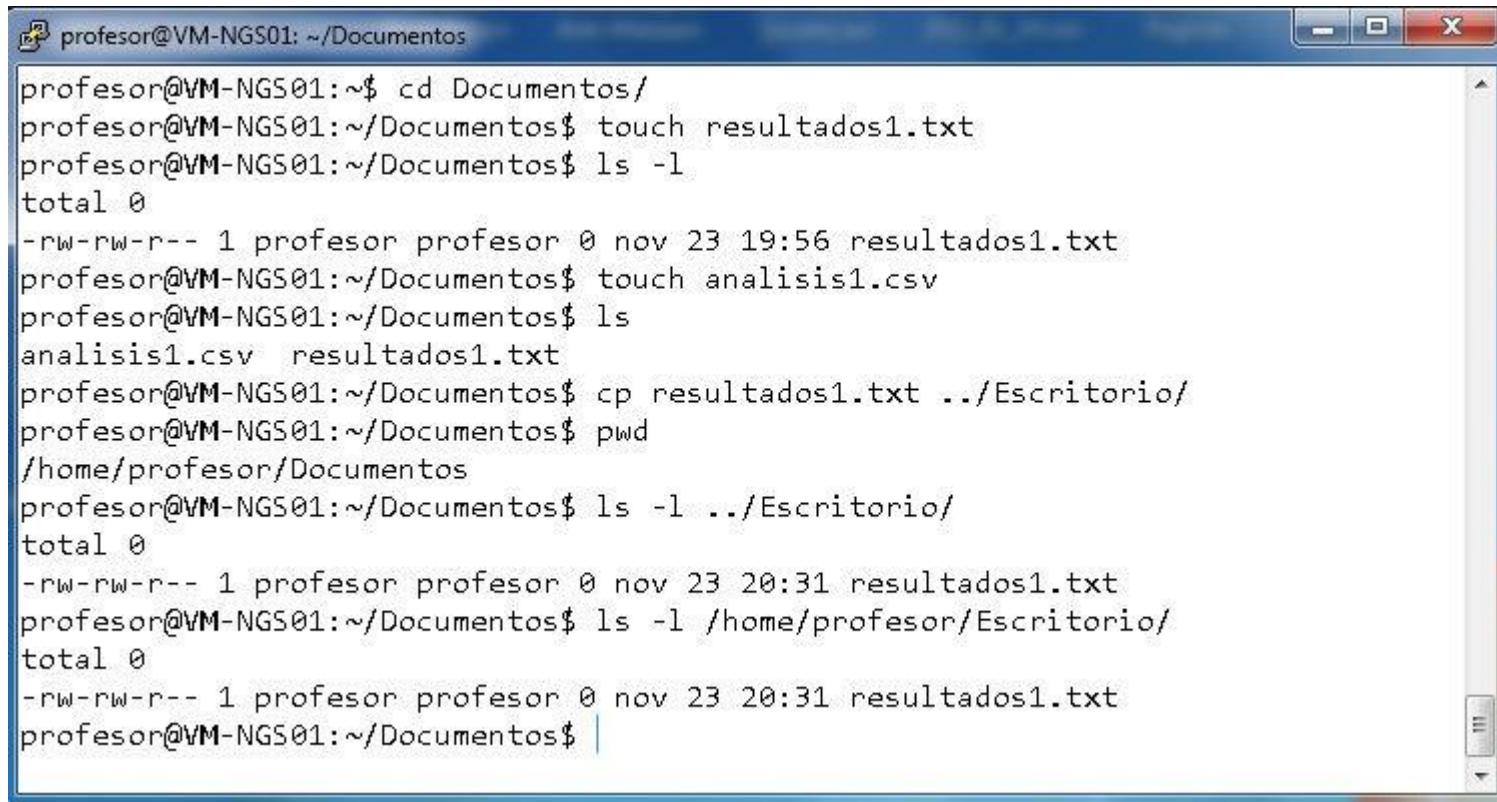


```
profesor@VM-NGS01:~$ ls -lr /opt/
total 96
drwxrwxr-x 3 root root 4096 oct 23 2012 VelvetOptimiser-2.2.5
lrwxrwxrwx 1 root root    27 oct 16 01:44 VelvetOptimiser -> /opt/VelvetOptimiser-2.2.5/
drwxr-xr-x 10 root root 4096 oct 16 01:42 velvet_1.2.10
lrwxrwxrwx 1 root root   19 oct 16 01:44 velvet -> /opt/velvet_1.2.10/
drwxrwxr-x 8 root root 4096 oct 20 14:18 vcftools_0.1.12b
lrwxrwxrwx 1 root root   21 oct 20 14:22 vcftools -> /opt/vcftools_0.1.12b
drwxr-xr-x 3 root root 4096 oct 17 12:50 Trimmomatic-0.32
lrwxrwxrwx 1 root root   22 oct 16 01:45 Trimmomatic -> /opt/Trimmomatic-0.32/
drwxr-xr-x 6 root root 4096 oct 20 13:43 srst2-0.1.4
lrwxrwxrwx 1 root root   16 oct 20 13:46 srst2 -> /opt/srst2-0.1.4
drwxr-xr-x 2 root root 4096 oct 16 01:32 seqtk-master
lrwxrwxrwx 1 root root   18 oct 16 01:34 seqtk -> /opt/seqtk-master/
drwxr-xr-x 2 root root 4096 may 21 2013 seq_crumbs-0.1.8-x64-linux
lrwxrwxrwx 1 root root   32 oct 16 01:30 seq_crumbs -> /opt/seq_crumbs-0.1.8-x64-linux/
drwxr-xr-x 7 root root 4096 oct 16 01:25 samtools-1.1
drwxr-xr-x 6 root root 4096 nov  5 14:29 samtools-0.1.18
lrwxrwxrwx 1 root root   17 nov  5 14:30 samtools -> ./samtools-0.1.18
drwxr-xr-x 5 root root 4096 oct 20 14:49 quast-2.3
lrwxrwxrwx 1 root root   14 oct 20 14:52 quast -> /opt/quast-2.3
drwxr-xr-x 2 root root 4096 oct 14 18:29 picard-tools-1.114
lrwxrwxrwx 1 root root   24 oct 16 09:31 picard-tools -> /opt/picard-tools-1.114
```

Comandos manejo de archivos y directorios

- **touch**
 - Crea un fichero vacío
 - **Sintaxis:** touch [archivo(s)]
- **cp (copy)**
 - El comando cp permite copiar archivos y/o directorios. Si el destino es un archivo existente, el archivo se sobrescribe
 - **Sintaxis:** cp [opciones] [archivo-origen][archivo-destino]
 - (archivo-origen y archivo-destino pueden ser un directorio)
 - **Opciones:**
 - r** Copia directorios recursivamente (Incluyendo todos lo que contenga)
 - p** Preservar los permisos, el usuario y el grupo del archivo a copiar
 - v** Da información en pantalla sobre los archivos que se van copiando

Comandos manejo de archivos y directorios

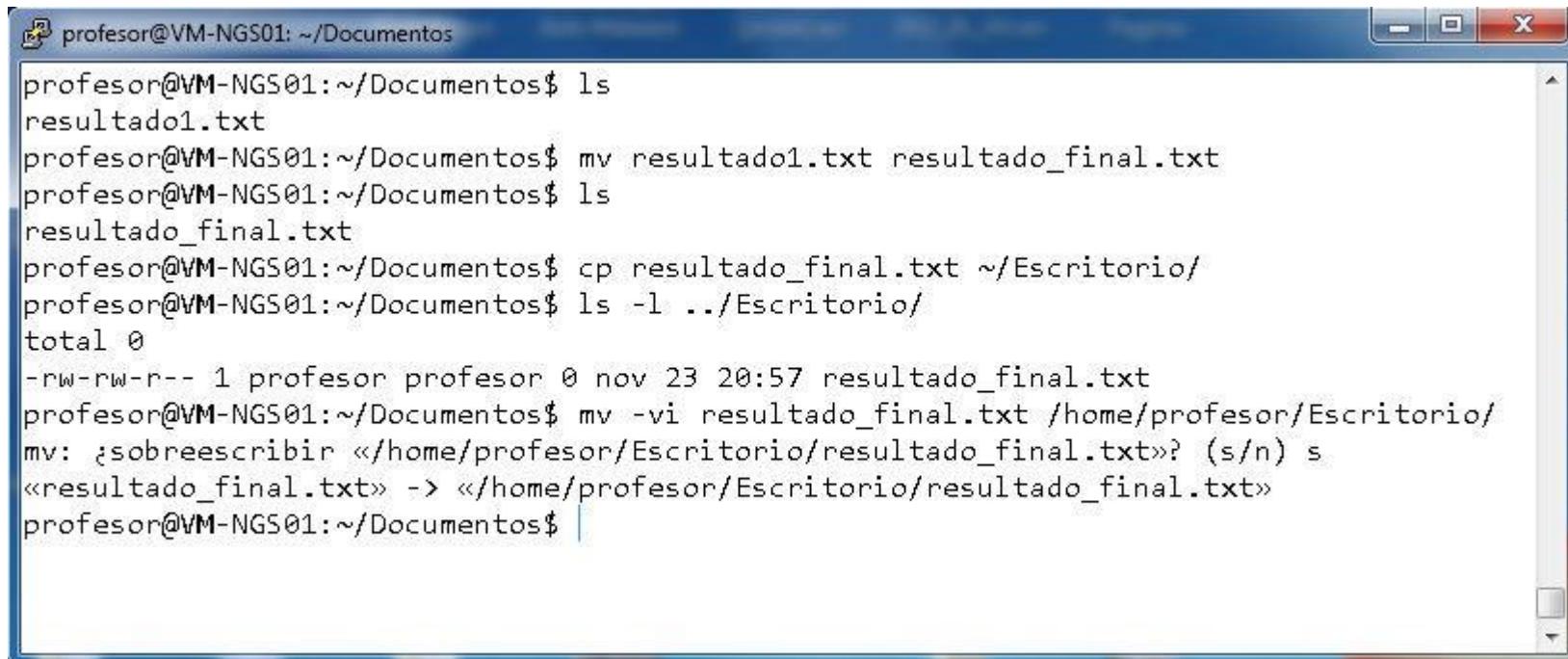


```
profesor@VM-NGS01: ~/Documentos
profesor@VM-NGS01:~/Documentos$ cd Documentos/
profesor@VM-NGS01:~/Documentos$ touch resultados1.txt
profesor@VM-NGS01:~/Documentos$ ls -l
total 0
-rw-rw-r-- 1 profesor profesor 0 nov 23 19:56 resultados1.txt
profesor@VM-NGS01:~/Documentos$ touch analisis1.csv
profesor@VM-NGS01:~/Documentos$ ls
analisis1.csv resultados1.txt
profesor@VM-NGS01:~/Documentos$ cp resultados1.txt ../Escritorio/
profesor@VM-NGS01:~/Documentos$ pwd
/home/profesor/Documentos
profesor@VM-NGS01:~/Documentos$ ls -l ../Escritorio/
total 0
-rw-rw-r-- 1 profesor profesor 0 nov 23 20:31 resultados1.txt
profesor@VM-NGS01:~/Documentos$ ls -l /home/profesor/Escritorio/
total 0
-rw-rw-r-- 1 profesor profesor 0 nov 23 20:31 resultados1.txt
profesor@VM-NGS01:~/Documentos$ |
```

Comandos manejo de archivos y directorios

- **mv (move)**
 - Se usa tanto para mover archivos a otros directorios, como para renombrarlos
 - **Sintaxis:** mv [opción] [origen(s)] [destino]
 - **Opciones:**
 - f Borrar los archivos de destino existentes sin preguntar al usuario
 - i Pregunta por cada archivo a sobrescribir antes de hacerlo
 - v Muestra el nombre de cada archivo a ser movido

Comandos manejo de archivos y directorios



```
profesor@VM-NGS01: ~/Documentos$ ls
resultado1.txt
profesor@VM-NGS01:~/Documentos$ mv resultado1.txt resultado_final.txt
profesor@VM-NGS01:~/Documentos$ ls
resultado_final.txt
profesor@VM-NGS01:~/Documentos$ cp resultado_final.txt ~/Escritorio/
profesor@VM-NGS01:~/Documentos$ ls -l ..../Escritorio/
total 0
-rw-rw-r-- 1 profesor profesor 0 nov 23 20:57 resultado_final.txt
profesor@VM-NGS01:~/Documentos$ mv -vi resultado_final.txt /home/profesor/Escritorio/
mv: ¿sobreescribir «/home/profesor/Escritorio/resultado_final.txt»? (s/n) s
«resultado_final.txt» -> «/home/profesor/Escritorio/resultado_final.txt»
profesor@VM-NGS01:~/Documentos$
```

Comandos manejo de archivos y directorios

- **mkdir (make directory)**
 - Se utiliza para crear directorios
 - **Sintaxis:** mkdir [opciones] [directorios(s)]
 - **Opciones:**
 - v Muestra mensaje confirmando la creación del directorio
- **rmdir (remove directory)**
 - El comando rmdir se utiliza para borrar directorios vacíos
 - **Sintaxis:** rmdir [opciones] [directorios(s)]
 - **Opciones:**
 - v Muestra lo que se está borrando

Comandos manejo de archivos y directorios

- **rm (remove)**
 - Se utiliza para borrar archivos y/o directorios
 - **Sintaxis :** rm [opciones] [archivo(s) | directorios(s)].
 - **Opciones:**
 - f** No pide confirmación para borrar (forzar)
 - r** Borrar todos los archivos que cuelgan de un directorio
 - v** Muestra lo que se está borrando

Comandos manejo de archivos y directorios

```
profesor@VM-NGS01:~/Documentos$ pwd
/home/profesor/Documentos
profesor@VM-NGS01:~/Documentos$ rm -v ..../Escritorio/resultado_final.txt
<..../Escritorio/resultado_final.txt> borrado
profesor@VM-NGS01:~/Documentos$ mkdir -v probatina_dia1 probatina_dia2
mkdir: se ha creado el directorio «probatina_dia1»
mkdir: se ha creado el directorio «probatina_dia2»
profesor@VM-NGS01:~/Documentos$ touch probatina_dia1/gradiente_magnesio.csv
profesor@VM-NGS01:~/Documentos$ cd probatina_dia2
profesor@VM-NGS01:~/Documentos/probatina_dia2$ touch gradiente_temperatura.csv
profesor@VM-NGS01:~/Documentos/probatina_dia2$ rm ..../probatina_dia1
rm: no se puede borrar <..../probatina_dia1>: Es un directorio
profesor@VM-NGS01:~/Documentos/probatina_dia2$ rm -r ..../probatina_dia1/
profesor@VM-NGS01:~/Documentos/probatina_dia2$ cd ..
profesor@VM-NGS01:~/Documentos$ ls
probatina_dia2
profesor@VM-NGS01:~/Documentos$ |
```

Comandos visualización contenido de archivos

- **cat**

- Concatena archivos y los muestra en la terminal, también permite la creación desde la entrada de teclado de un nuevo archivo
- **Sintaxis:** cat [opciones] [archivo(s)]
- **Opciones:**
 - -n Numera todas las líneas

Usando el símbolo **>** se redirecciona la entrada de teclado hacia un archivo. Si se usa el símbolo **>>** se añade la entrada del teclado al contenido de un archivo ya existente
Después de añadir el texto el archivo se cierra pulsado la tecla Enter para empezar una línea nueva y las teclas **ctrl + d** para salir
Ejemplo: \$ cat > notas.txt

Comandos visualización contenido de archivos

```
profesor@VM-NGS01: ~/Documentos
profesor@VM-NGS01:~/Documentos$ cat > resultados.txt
El comando cat permite añadir contenido al fichero
pero se usa sobretodo para visualizarlo
profesor@VM-NGS01:~/Documentos$ cat resultados.txt
El comando cat permite añadir contenido al fichero
pero se usa sobretodo para visualizarlo
profesor@VM-NGS01:~/Documentos$ cat > resultados.txt
Si el archivo tiene texto se sobrescribe
profesor@VM-NGS01:~/Documentos$ cp resultados.txt resultados_copia.txt
profesor@VM-NGS01:~/Documentos$ cat >> resultados_copia.txt
En este caso al usar la 'doble redireccion la informacion no se borra, se añade a la
existente
profesor@VM-NGS01:~/Documentos$ cat resultados.txt resultados_copia.txt > juntar_doc.txt
profesor@VM-NGS01:~/Documentos$ cat juntar_doc.txt
Si el archivo tiene texto se sobrescribe
Si el archivo tiene texto se sobrescribe
En este caso al usar la 'doble redireccion la informacion no se borra, se añade a l
existente
profesor@VM-NGS01:~/Documentos$ |
```

Comandos visualización contenido de archivos

- more
 - Permite ver en pantalla el contenido de uno o mas archivos
 - Muestra el contenido de los ficheros de forma paginada (pantalla a pantalla)
 - Se usa en archivos cortos
 - Sintaxis: more [archivo(s)]
 - Manejo con teclado:
 - **q** Se sale del comando
 - **Espacio** Desplazamiento hasta próxima pantalla
 - **Enter** Desplazamiento de una línea

Comandos visualización contenido de archivos

- **less**

- Permite ver en pantalla el contenido de uno o mas archivos
- Muestra el contenido de los ficheros de forma paginada (pantalla a pantalla) y permite el retroceso
- Se usa en archivos largos
- **Sintaxis:** less [archivo(s)]
- **Manejo con teclado:**

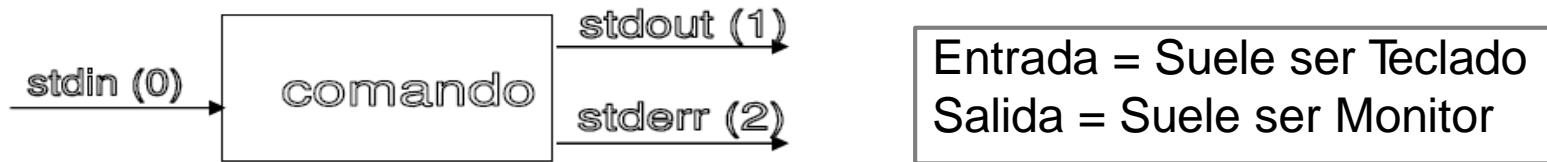
- | | |
|-------------------|---|
| ▪ q | Se sale del comando |
| ▪ Espacio | Desplazamiento hasta próxima pantalla |
| ▪ Enter | Desplazamiento de una línea |
| ▪ Cursores | Desplazamiento por el contenido del archivo |

Editor de texto “nano”

- Se usa desde el terminal (interactivo). Es sencillo de usar y muy adecuado para realizar operaciones simples sobre pequeños archivos de texto
 - ^X = CTRL+X
 - **Sintaxis:** nano [archivo]
 - Lo más básico:
 - ^X Salir
 - ^O Guardar el archivo que se esta editando
 - ^W Buscar desde la posición actual del cursor
 - ^C Indicar la posición actual del cursor
 - ^E Desplazar el cursor al final de la línea en curso
 - ^A Desplazar el cursor al principio de la línea en curso
 - ^K Cortar el texto seleccionado o, si nada esta seleccionado, la línea en curso
 - ^U Pegar
 - ^G Acceder a la ayuda

Redirecciones

- Toda operación dentro de un proceso tiene una entrada y una salida (I/O)
- Linux permite que la entrada y la salida sea redireccionada



- **Símbolos:**

>	Redirecciona la salida a un archivo
2>	Redirecciona la salida de error a un archivo
&>	Redirecciona ambas salidas a un archivo
>>	Redirecciona la salida a un archivo (no sobrescribe)
2>>	Redirecciona la salida a un archivo (no sobrescribe)
<	Redirecciona la entrada

¡OJO! Con '`<`, `>`, `2>`, `&>`' si el nombre del archivo existe, se sobrescribe

Redirecciones

- Redirección de la entrada
 - \$ cat > libro.txt
 - \$ grep vlc < dpkg --list
- Redirección de la salida
 - \$ ls /etc > lista.txt ; ls /usr >> lista.txt
 - \$ sort /tmp/listado > /tmp/listado_ordenado
 - \$ echo \$PATH > /tmp/path.txt
- Redirección de la salida error
 - \$ ls archivo 2> /dev/null
 - \$ ls archivo 2>> archivo.error
 - \$ ls /bin > /tmp/salida.std 2> /tmp/salida.err

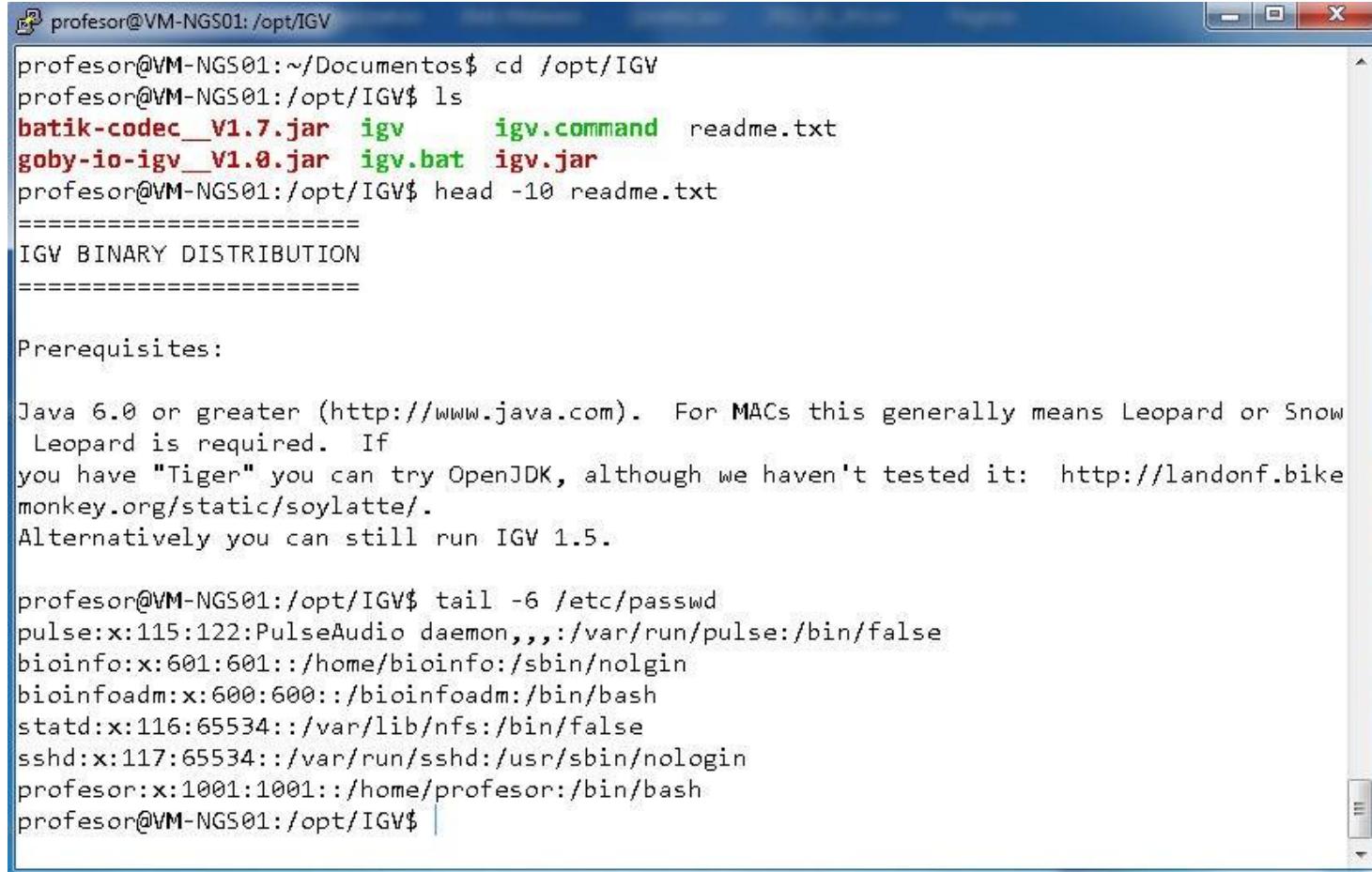
Tuberías

- Conectan la salida de un proceso con la entrada de otro
- Permiten que la salida estándar de un comando (proceso) se conecte con la entrada estándar de otro comando (proceso)
 - **comando1 | comando2 | comando3 | ... | comando N**
 - La salida estándar de comando1 se conecta con la entrada estándar de comando2
 - La salida estándar de comando2 se conecta con la entrada estándar de comando3
- Ejemplos:
 - \$ cat /tmp/nombres.txt | sort
 - \$ cat /tmp/nombres.txt | head -5
 - \$ cat /etc/passwd | wc -l

Comandos procesamiento de archivos

- **head**
 - Muestra las primeras líneas de un archivo (por defecto 10)
 - **Sintaxis:** head [opciones] [archivo(s)]
 - **opciones:**
 - **-n (número)** Muestras las primeras *número* de líneas
- **tail**
 - Muestra las ultimas líneas de un archivo (por defecto 10)
 - **Sintaxis:** tail [opciones] [archivo(s)]
 - **opciones:**
 - **-número** Muestras las últimas *número* de líneas

Comandos procesamiento de archivos



```
profesor@VM-NGS01:~/Documentos$ cd /opt/IGV
profesor@VM-NGS01:/opt/IGV$ ls
batik-codec_V1.7.jar  igv    igv.command  readme.txt
goby-io-igv_V1.0.jar  igv.bat  igv.jar
profesor@VM-NGS01:/opt/IGV$ head -10 readme.txt
=====
IGV BINARY DISTRIBUTION
=====

Prerequisites:

Java 6.0 or greater (http://www.java.com). For MACs this generally means Leopard or Snow Leopard is required. If you have "Tiger" you can try OpenJDK, although we haven't tested it: http://landonf.bikemonkey.org/static/soylatte/. Alternatively you can still run IGV 1.5.

profesor@VM-NGS01:/opt/IGV$ tail -6 /etc/passwd
pulse:x:115:122:PulseAudio daemon,,,:/var/run/pulse:/bin/false
bioinfo:x:601:601::/home/bioinfo:/sbin/nologin
bioinfoadm:x:600:600::/bioinfoadm:/bin/bash
statd:x:116:65534::/var/lib/nfs:/bin/false
sshd:x:117:65534::/var/run/sshd:/usr/sbin/nologin
profesor:x:1001:1001::/home/profesor:/bin/bash
profesor@VM-NGS01:/opt/IGV$
```

Comandos procesamiento de archivos

- **WC**
 - Muestra el número de líneas, palabras o caracteres de los ficheros que se le indiquen, o el tamaño de la línea más larga
 - **Sintaxis:** wc [opciones] [archivo]
 - **opciones:**
 - c Cuenta caracteres
 - w Cuenta palabras
 - l Cuenta líneas
 - L El tamaño de la línea mas larga

Comandos procesamiento de archivos

```
profesor@VM-NGS01:/opt/artemis
profesor@VM-NGS01:~$ cd /opt/artemis ; ls
act artemis_sqlmap dnaphotter images Makefile org seqdb uk
art corba etc lib nsdb README type
profesor@VM-NGS01:/opt/artemis$ wc README
 55 307 2042 README
profesor@VM-NGS01:/opt/artemis$ echo 'El numero de caracteres es: ' ; wc -c README
El numero de caracteres es:
2042 README
profesor@VM-NGS01:/opt/artemis$ echo 'El numero de palabras es: ' ; wc -w README
El numero de palabras es:
307 README
profesor@VM-NGS01:/opt/artemis$ echo 'El numero de lineas es: ' ; wc -l README
El numero de lineas es:
55 README
profesor@VM-NGS01:/opt/artemis$ |
```

Comandos procesamiento de archivos

- **sort**
 - Muestra en orden ascendente el contenido de los ficheros
 - **Sintaxis:** sort [opciones] [archivo(s)]
 - opciones:
 - c Comprueba que el archivo esta ordenado Elimina las líneas duplicadas
 - f No diferencia entre mayúsculas y minúsculas
 - r Invierte el orden de la clasificación
 - o Almacena la salida en el archivo especificado
 - Ejemplos:
 - \$ tail -n 20 /etc/passwd | sort -o usuarios_sistema.txt
 - \$ sort -fr listado_genes -o listado_genes_invertido

Comandos procesamiento de archivos

- **cut**
 - Se utiliza para cortar campos seleccionados de cada línea de un archivo
 - Por defecto usa el tabulador como delimitador de campo
 - **Sintaxis:** cut [opciones] [archivo(s)]
 - **Opciones:**

-c lista Selecciona solamente estos caracteres

-f lista Selecciona solamente estos campos Para

-d Lista: (N = número) especificar el delimitador

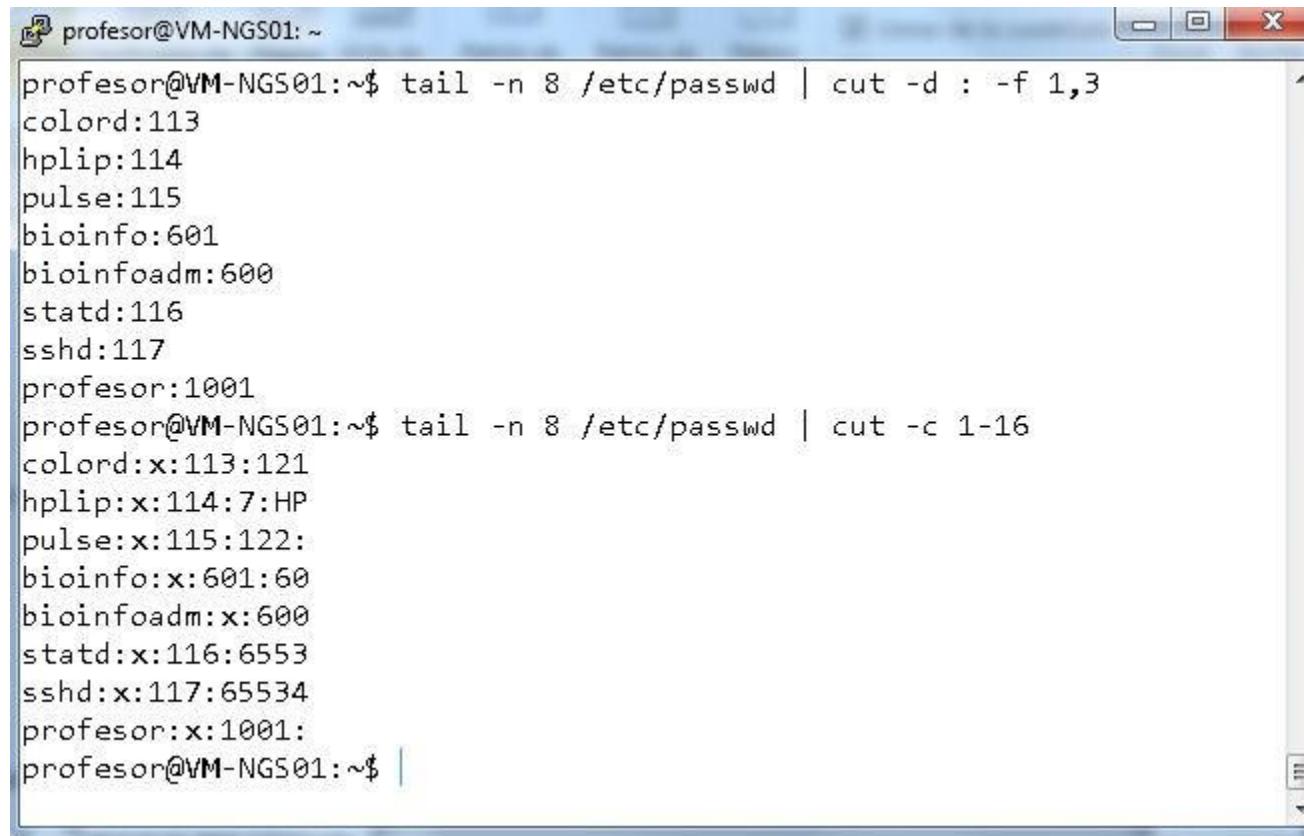
N Desde el primer carácter A

N- partir de N hasta el final Rango

N-M entre N y M

-M Desde el primer carácter hasta M

Comandos procesamiento de archivos



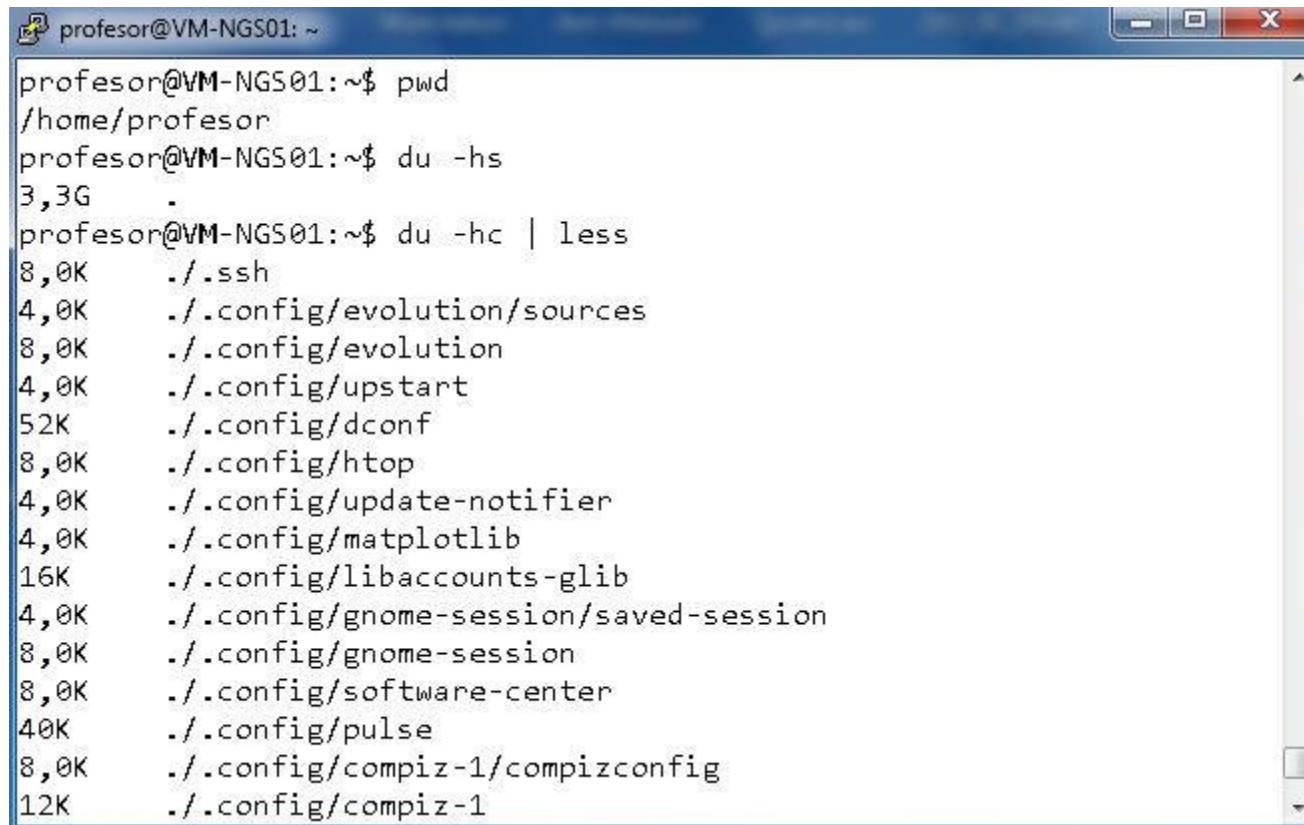
```
profesor@VM-NGS01: ~
profesor@VM-NGS01:~$ tail -n 8 /etc/passwd | cut -d : -f 1,3
colord:113
hplip:114
pulse:115
bioinfo:601
bioinfoadm:600
statd:116
sshd:117
profesor:1001
profesor@VM-NGS01:~$ tail -n 8 /etc/passwd | cut -c 1-16
colord:x:113:121
hplip:x:114:7:HP
pulse:x:115:122:
bioinfo:x:601:60
bioinfoadm:x:600
statd:x:116:6553
sshd:x:117:65534
profesor:x:1001:
profesor@VM-NGS01:~$ |
```

Comandos procesamiento de archivos

- **du**
 - Muestra el espacio que ocupa el fichero o directorio
 - **Sintaxis** : du [opciones] [archivo | directorios].
 - **Opciones:**

- h Muestra los tamaños de archivo en un formato más legible
- s Muestra únicamente los tamaños de los archivos especificados
- c Muestra en pantalla el espacio total ocupado por los archivos especificados

Comandos procesamiento de archivos



A screenshot of a terminal window titled "profesor@VM-NGS01: ~". The window displays the output of several Linux command-line tools used for file system analysis. The commands shown are:

```
profesor@VM-NGS01:~$ pwd
/home/profesor
profesor@VM-NGS01:~$ du -hs
3,3G .
profesor@VM-NGS01:~$ du -hc | less
8,0K ./ssh
4,0K ./config/evolution/sources
8,0K ./config/evolution
4,0K ./config/upstart
52K ./config/dconf
8,0K ./config/htop
4,0K ./config/update-notifier
4,0K ./config/matplotlib
16K ./config/libaccounts-glib
4,0K ./config/gnome-session/saved-session
8,0K ./config/gnome-session
8,0K ./config/software-center
40K ./config/pulse
8,0K ./config/compiz-1/compizconfig
12K ./config/compiz-1
```

Comandos búsqueda de información

- **grep**
 - Examina en los archivos las líneas que concuerdan con la expresión regular dada y lo muestra en la salida estándar
 - Una **expresión regular** es una secuencia de caracteres que forma un patrón de búsqueda
 - ^expresión : Muestra líneas que empiezan por el patrón
 - expresión\$: Muestra líneas que terminan por el patrón
 - Si se pasa por argumento más de un archivo, el nombre del archivo aparece delante de cada línea
 - **Sintaxis:** grep [opciones] [expresión] [archivo(s)]

Comandos búsqueda de información

- **grep**
 - Ejemplos de uso:
 - \$ grep patrón archivo
 - \$ head archivo | grep patrón
 - **Opciones:**
 - n • Muestra el número correspondiente al lugar que ocupa en el fichero la/s línea/s encontrada/s
 - l • Muestra el nombre del fichero donde hubo coincidencias pero no muestra la coincidencia
 - v • Muestra las líneas que no se corresponden con el patrón
 - w • Muestra solo coincidencias donde el patrón ocupa palabras enteras
 - i • Ignora diferencias entre mayúsculas y minúsculas

Comandos búsqueda de información

- **grep -i ttcag secuencia.txt**
 - Considera idénticas las mayúsculas y minúscula en la cadena de búsqueda ttcag (ttcag=TTCAg=TtcaG=etc.)
- **grep -v ^ttcaaagagattaaa secuencia.txt**
 - Lista las líneas que **NO** empiecen por la cadena **ttcaaagagattaaa** del archivo secuencia.txt
- **grep ttcaaagagattaaa\$ secuencia.txt**
 - Lista las líneas que **terminen** por la cadena **ttcaaagagattaaa** del archivo secuencia.txt
- **dpkg --list | grep programa**
 - dpkg --list sirve para ver programas instalados. En este caso grep lista las líneas en las que aparezca el nombre del programa y sino es que no esta instalado

Comandos búsqueda de información

- **find**
 - Permite encontrar ficheros y directorios
 - Busca de forma recursiva desde la ruta dada en la estructura de archivos
 - Si no se especifica ruta busca desde el directorio actual
 - Find necesita:
 - Directorio/s donde buscar
 - Reglas de búsqueda
 - Permite ejecutar comandos por cada archivo coincidente con las reglas
 - Es posible utilizar expresiones regulares o reglas no concretas
 - Cada regla se usa como un modificador: **-regla valor**
 - Una invocación puede usar varias reglas
 - **\$ find /ruta -regla valor -regla valor ...**

Comandos búsqueda de información

- **Opciones**

- name** Archivos cuyo nombre coincide con el valor
- iname** Archivos cuyo nombre coincide con el valor (**no** case sensitive)
- ctime** Archivos creados hace tantos días como indique el valor
- type** Archivos de ese tipo (b, c, d, f)
- user** Archivos de ese usuario
- perm** Archivos con esos permisos
- exec comando** Busca los archivos y ejecuta el comando

Comandos búsqueda de información

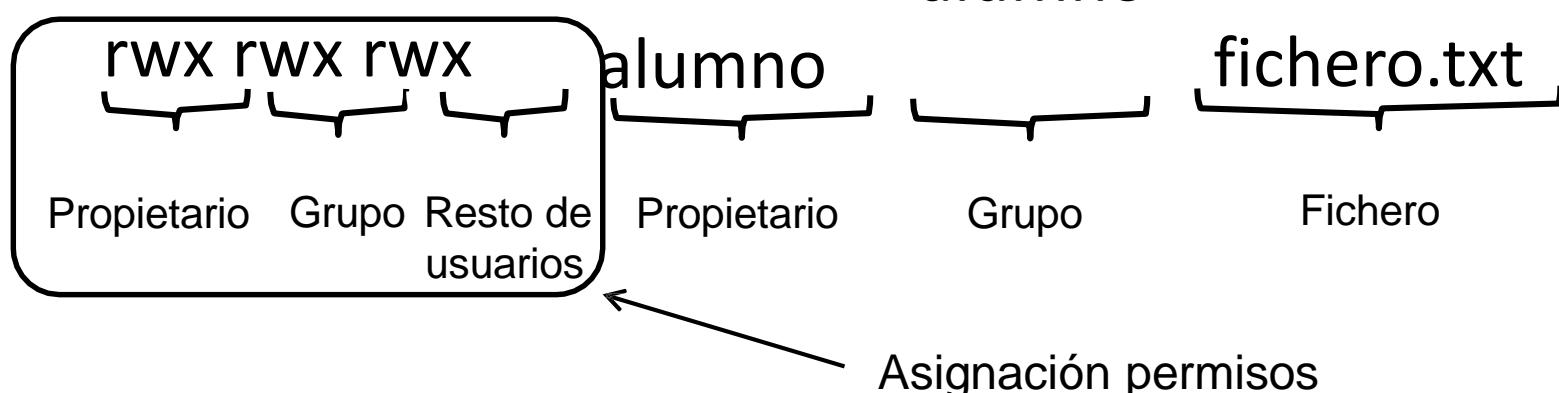
- Ejemplos de uso:
 - Archivos y directorios cuyo nombre empiece con ‘a’ o ‘A’
 - **find /ruta -iname “a*”**
 - Archivos y directorios creados hace menos de tres días.
 - **find /ruta -ctime -3**
 - Archivos creados hace menos de tres días.
 - **find /ruta -type f -ctime -3**
 - Directorios cuyo propietario es root y terminen por ‘.conf’
 - **find /ruta -type d -user root -name “*.conf”**
 - Eliminar archivos cuyo propietario es alumno en el directorio /tmp.
 - **find /tmp -type f -user alumno -exec rm -f {} \;**

Administración ficheros y directorios

- **Asignación de permisos**
 - Comando chmod
- **Asignación de usuario y grupo propietario**
 - Comando chown
 - Comando chgrp

Asignación de permisos

- Los permisos de un fichero solo pueden ser alterados por su propietario, usuarios que pertenezcan al grupo y por al administrador
- El sistema almacena estos permisos como una secuencia de 9 bits.
- Esta secuencia tiene una estructura en tres grupos: para el propietario, para el grupo del fichero y para los demás.



Asignación de permisos

- Para la modificación de los permisos usamos **chmod**
- **Opciones:**
 - **-f** Forzar
 - **-R** Recursivo
- Suele usarse de dos modos:
 - **Modo textual.** Mas sencillo y lento.
 - **Modo octal.** Menos intuitivo pero mas rápido

Asignación de permisos

- Consiste en añadir(+) o eliminar(-) permisos
- A los diferentes actores: propietario(u), grupo(g) o los demás(o)
- **Sintaxis:** chmod [opciones] [u|g|o|a] [+-=] [rwx] archivo
- Ejemplos:
 - Quitar el permiso de lectura al grupo
 - **chmod g-r archivo**
 - Agregar permiso de ejecución al propietario y permiso de lectura para el grupo y otros
 - **chmod u+x,go+r archivo**
 - Agregar permiso de lectura y escritura a todos
 - **chmod a+rwx archivo**

Asignación de permisos

- Modifica los 9 bits a la vez
- Se especifica una nueva combinación de estos bits en modo octal
- **Truco:** r=4, w=2, x=1
- **Sintaxis:** chmod [opciones] modo octal archivo
- Ejemplos:
 - chmod 644 archivo (r w - r - - r - -)
 - chmod 755 archivo (r w x r - x r - x)
 - chmod 400 archivo (r - - - - - - - - -)

Asignación de permisos

```
profesor@VM-NGS01:~$ mkdir pruebas ; cd pruebas ; pwd
/home/profesor/pruebas
profesor@VM-NGS01:~/pruebas$ cp /etc/passwd passwd_copia
profesor@VM-NGS01:~/pruebas$ cp /opt/IGV/readme.txt .
profesor@VM-NGS01:~/pruebas$ ll
total 24
drwxrwsr-x  2 profesor ngscurso 4096 nov 23 21:55 ./
drwxrwsr-x 21 profesor ngscurso 12288 nov 23 21:54 ../
-rw-r--r--  1 profesor ngscurso  2048 nov 23 21:54 passwd_copia
-rw-r--r--  1 profesor ngscurso  1238 nov 23 21:55 readme.txt
profesor@VM-NGS01:~/pruebas$ chmod 600 passwd_copia
profesor@VM-NGS01:~/pruebas$ chmod go+w readme.txt
profesor@VM-NGS01:~/pruebas$ ls -l
total 8
-rw----- 1 profesor ngscurso 2048 nov 23 21:54 passwd_copia
-rw-rw-rw- 1 profesor ngscurso 1238 nov 23 21:55 readme.txt
profesor@VM-NGS01:~/pruebas$ cd ..
profesor@VM-NGS01:~$ chmod -R a-w pruebas/
profesor@VM-NGS01:~$ ls -l pruebas/
total 8
-r----- 1 profesor ngscurso 2048 nov 23 21:54 passwd_copia
-r--r--r-- 1 profesor ngscurso 1238 nov 23 21:55 readme.txt
profesor@VM-NGS01:~$ |
```

Asignación de usuario y grupo propietario

- **chown**
 - Permite cambiar el usuario o grupo propietario de un archivo o directorio
 - **Sintaxis:** chown [opciones] usuario[:grupo] [archivo(s)]
 - **Opciones:**
 - -R: Cambio recursivo
- **chgrp**
 - Permite cambiar el únicamente el grupo propietario de un archivo o
 - directorio
 - **Sintaxis:** chgrp [opciones] grupo [archivo(s)]
 - **Opciones:**
 - -R: Cambio recursivo

Asignación de usuario y grupo propietario

```
profesor@VM-NGS01:~/pruebas$ cd pruebas/ ; ls -l
total 8
-r----- 1 profesor ngscurso 2048 nov 23 21:54 passwd_copia
-r--r--r-- 1 profesor ngscurso 1238 nov 23 21:55 readme.txt
profesor@VM-NGS01:~/pruebas$ chown bioinfoadm:bioinfoadm passwd_copia
chown: cambiando el propietario de «passwd_copia»: Operación no permitida
profesor@VM-NGS01:~/pruebas$ sudo chown bioinfoadm:bioinfoadm passwd_copia
[sudo] password for profesor:
profesor@VM-NGS01:~/pruebas$ ls -l passwd_copia
-r----- 1 bioinfoadm bioinfoadm 2048 nov 23 21:54 passwd_copia
profesor@VM-NGS01:~/pruebas$ chmod ug+rwx passwd_copia
chmod: cambiando los permisos de «passwd_copia»: Operación no permitida
profesor@VM-NGS01:~/pruebas$ sudo chmod ug+rwx passwd_copia
profesor@VM-NGS01:~/pruebas$ groups
profesor bioinfo sambauers ngscurso
profesor@VM-NGS01:~/pruebas$ sudo chgrp ngscurso readme.txt
profesor@VM-NGS01:~/pruebas$ chmod 660 readme.txt
profesor@VM-NGS01:~/pruebas$ ls -l
total 8
-rw-rw---- 1 bioinfoadm bioinfoadm 2048 nov 23 21:54 passwd_copia
-rw-rw---- 1 profesor ngscurso 1238 nov 23 21:55 readme.txt
profesor@VM-NGS01:~/pruebas$ |
```

Procesos

- Concepto
- Identificadores de procesos
- Gestión de procesos
- Terminar procesos
- Primer y segundo plano

Procesos

- **Definición de proceso**
 - Se llama proceso o tarea a cualquier programa en ejecución
- **Ejemplos:**
 - Cuando llamamos a cualquier comando del SO (ls, more, find, ...) se crea un proceso
 - Cuando usamos el navegador Firefox se crea un proceso
 - Cualquier sesión que abre un usuario en el sistema ejecuta un shell, que es un proceso que está presente durante la duración de la sesión

Identificadores de procesos

- Cada proceso tiene asignado un *identificador de proceso o PID*
- PID: Número que identifica de forma única a cada proceso
- Además del PID, cada proceso está relacionado con otros identificadores como el identificador de usuario (*UID*) o el identificador del proceso padre (*PPID*)
- Cuando ejecutamos cualquier orden desde una shell, esta shell es el proceso padre del proceso que ejecuta la orden

Los procesos siguen un modelo en árbol
Tienen padre y pueden tener hijos

Gestión de procesos

- **Comando ps**
 - Visualiza un listado de los procesos activos en la máquina
 - **Sintaxis:** ps [opciones]
 - a • Visualiza los procesos de todos los usuarios.
 - u • Visualiza información de los procesos el usuario
 - x • Visualiza procesos que no están controlados por ninguna terminal
 - e • Visualiza todos los procesos en ejecución
 - f • Visualiza información detallada de los procesos

Gestión de procesos

```
profesor@VM-NGS01:~$ ps aux
USER        PID %CPU %MEM      VSZ      RSS TTY      STAT START   TIME COMMAND
root          1  0.0  0.0  33952   3276 ?      Ss  oct26   0:04 /sbin/init
root          2  0.0  0.0       0       0 ?      S  oct26   1:21 [kthreadd]
root          3  0.0  0.0       0       0 ?      S  oct26   0:01 [ksoftirqd/0]
root          5  0.0  0.0       0       0 ?      S< oct26   0:00 [kworker/0:0H]
```

```
profesor@VM-NGS01:~$ ps aux | grep ssh
profesor  1023  0.0  0.0  15960    912 pts/8      S+  02:51   0:00 grep --color=auto ssh
root     1303  0.0  0.0  61364   3076 ?      Ss  oct26   0:00 /usr/sbin/sshd -D
profesor  5234  0.0  0.0  10616    312 ?      Ss  oct27   0:00 ssh-agent -s
root    25042  0.0  0.0 109784   4300 ?      Ss  oct30   0:00 sshd: profesor [priv]
profesor 25143  0.0  0.0 109784   2052 ?      S  oct30   0:01 sshd: profesor@pts/5
root    26980  0.0  0.0 109784   4280 ?      Ss  02:25   0:00 sshd: profesor [priv]
profesor 27016  0.0  0.0 109784   2056 ?      S  02:25   0:00 sshd: profesor@pts/8
profesor@VM-NGS01:~$
```

Gestión de procesos

- **htop**
 - htop es una pequeña aplicación que nos muestra los procesos y nos permite gestionarlos. También nos enseña el estado de la memoria, la swap y el consumo de CPU
 - Su análogo en Windows es el programa *administrador de sistemas*
 - Se divide en 3 secciones:
 - ↗ La primera resume el uso de CPU, memoria RAM y memoria de intercambio (*swap*), con barras indicadoras horizontales
 - ↗ La segunda sección corresponde a un listado de los procesos actualmente en ejecución en todo el sistema, ordenados por uso de CPU
 - ↗ La tercera sección es un menú que incluye opciones para pedir ayuda, realizar búsquedas, enviar señales a los procesos, desplegar la lista en forma de árbol y la opción de salida, entre otros

Gestión de procesos

```
profesor@VM-NGS01: ~

1 [|||||]
2 [|||]
3 [||]
4 [|||]
Mem[|||||]
Swp[

Tasks: 126, 241 thr; 1 running
Load average: 2.85 2.60
Uptime: 4 days, 05:37:00

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
6033 profesor 20 0 1065M 233M 55820 S 17.4 2.9 14:15.50 /usr/lib/firefox/fire
6077 profesor 20 0 1065M 233M 55820 D 10.7 2.9 3:37.37 /usr/lib/firefox/fire
6118 profesor 20 0 1065M 233M 55820 D 7.4 2.9 2:57.40 /usr/lib/firefox/fire
25945 profesor 20 0 30080 2224 1428 R 4.7 0.0 0:00.67 htop
5610 profesor 20 0 309M 20104 16220 S 0.7 0.2 15:26.19 /usr/lib/vmware-tools
1068 20 0 353M 10032 6228 S 0.7 0.1 0:07.53 NetworkManager
1568 20 0 161M 4672 3768 S 0.0 0.1 7:52.37 /usr/sbin/vmtoolsd
1316 20 0 37144 1012 696 S 0.0 0.0 0:10.17 /usr/sbin/kerneloops
1 20 0 33952 3276 1480 S 0.0 0.0 0:04.57 /sbin/init
358 20 0 20000 1188 608 S 0.0 0.0 0:00.39 upstart-udev-bridge -
366 20 0 51732 1956 1016 S 0.0 0.0 0:00.12 /lib/systemd/systemd-
523 20 0 40120 2360 1036 S 0.0 0.0 0:05.11 dbus-daemon --system
579 20 0 23480 424 212 S 0.0 0.0 0:00.00 rpc.idmapd
594 20 0 249M 1832 1016 S 0.0 0.0 0:00.24 rsyslogd

F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit
```

Terminar procesos

- **Comando kill**
 - En muchas ocasiones nos veremos en la necesidad de cancelar la ejecución de ciertos procesos cuya ejecución no se desea que continúe
 - Para ellos es necesario enviar una señal de cancelación mediante la orden kill
 - La señal de muerte en Linux, es -9
 - Solo root puede matar a cualquier proceso
 - **Sintaxis:** # kill -9 PID

Primer y segundo plano

- Por defecto los procesos se ejecutan en primer plano
- Un proceso o trabajo se está ejecutando en *primer plano (foreground)* si la línea de órdenes está ocupada en la ejecución de dicho proceso (no hay prompt). El usuario no puede ejecutar otros procesos en el mismo shell
- Un proceso que se ejecuta en primer plano se puede detener con:
 - **Ctrl + C**
- Usando & se ejecutan en segundo plano
 - **Sintaxis:** comando [opción] [argumentos] &
- Con el comando **fg** se puede pasar a primer plano un proceso
 - **Sintaxis:** fg [numero_PID]