

Implantación de SO

UT7:

**Administración y
aseguramiento de la
información**



INTRODUCCIÓN A LA SHELL



Linux distingue entre mayúsculas y minúsculas

La consola



- Conjunto formado por el teclado y el monitor que están conectados de forma directa con el ordenador donde esta instalado el sistema Linux.
- Dentro de la misma sesión se pueden ejecutar varios programas a la vez, ya que es multitarea.

Terminal

- Conjunto formado por el teclado y un monitor conectados al ordenador en el que esta funcionando el sistema gracias a un programa de comunicaciones.

Consolas virtuales

- En linux es posible abrir varias consolas simulando varios terminales y trabajar con diferentes usuarios en cada una de ellas, o tener varias sesiones abiertas para un mismo usuario.
- ¿Cómo abrir consolas?
 - Modo texto → (ALT) + (F1-F6)
 - Modo gráfico → (ALT+F7).... (ALT+F12)
- Para cambiar desde la consola gráfica a la de texto debemos añadir la tela CTRL.
 - Ejemplo:
 - CTRL + ALT + F3

Consola de texto

- Al abrir la consola y logarnos se muestra el PROMPT
- PROMPT esta constituido por:
 - **Usuario** que maneja la shell
 - Nombre de la **máquina** desde la que se maneja la shell
 - **Directorio** del sistema de archivos donde se encuentra el usuario.
 - ~ Simboliza el directorio personal del usuario
 - \$ El usuario conectado es un usuario normal
 - # El usuario con privilegios. Normalmente root

```
cicloformativo@cicloformativo-virtual-machine:~$
```

Definición Shell

- Envuelve al Núcleo o Kernel.
- Es la parte de LINUX que se ocupa de comunicarse con el usuario.
- Recoge las instrucciones tecleadas en el terminal y las convierte a sintaxis máquina.
- Es el primer programa y el último que se ejecuta al iniciar una sesión de usuario.
- Hablar de shell es hablar de shell en modo texto

Tipos de shell



- ☐ Bourne Shell (**sh**)
 - Almquist shell (**ash**)
- ☐ Bourne-Again shell (**bash**)
 - Debian Almquist shell (**dash**)
- ☐ Korn Shell (**ksh**)
 - Z shell (**zsh**) + completa

Funciones de la Shell



- Interpreta y ejecuta comandos.
- Configura el entorno de trabajo.
- Asigna valores a las variables.
- Redirecciona entradas y salidas.
- Encauza lista de órdenes o tuberías.
- Es un completo lenguaje-interpretador de programación (scripting)

Otras posibilidades shell bash



- Autocompletar durante la escritura
 - Tecla **TAB**
 - Uso: Teclear varios caracteres y TAB
- Historial de comandos
 - Tecla de cursores (**←↑→↓**)
 - Uso: Permite recuperar comandos ya ejecutados
- Poderosas estructuras de control para realizar scripts → Procesos por lotes
 - If, for, while, select....
- Definición de funciones y alias para comandos

Línea de ordenes

- Una orden acaba con la tecla INTRO.
- Si no está completa, visualiza el signo ">" hasta que esta sea completada.

```
root@cicloformativo-virtual-machine:~# ls "/etc/default
>
>
> "
ls: no se puede acceder a /etc/default

: No existe el archivo o el directorio
root@cicloformativo-virtual-machine:~#
```

Metacaracteres

- `\` Elimina el significado del carácter siguiente
- `?` Sustituye a cualquier carácter
- `*` sustituye a un grupo de caracteres
- `[]` Sustituye a cualquier valor incluido entre los corchetes, un solo carácter
- `[!]` Sustituye todos los caracteres excepto los que están entre corchetes
- Ejemplo:
 - `mi[0a]` → mio o mia
 - `achivo[A-P]` → archivoA, archivoBarchivoP
 - `[!0-3]*` → Los que no empiecen por 0,1, 2 o 3
 - `[12]?a` → empiecen por 1 o 2 lleven un carácter y después terminen a

Recuerda los caracteres especiales:

- `&` Backgroup
- `$` Variable
- `=` Valor a una variable

Agrupamiento de ordenes

- Formas de introducir varias ordenes:
 - Ejecución sucesiva e independiente de todas las ordenes
 - **Orden1;orden2;orden3** `cat Ks1.cfg; ls; cd plantillas`
 - Igual que el anterior, pero se puede considerar como un conjunto
 - **(Orden1;orden2;orden3)** `(cat ks1.cfg; ls; cd Plantillas)`
 - Redirige la salida a un archivo
 - **(Orden1;orden2;orden3) > archivo** `(cat ks1.cfg; ls; cd Plantillas) > PruebaArch`
 - Ejecuta un grupo de ordenes en segundo plano (background)
 - **(Orden1;orden2;orden3) &** `(ls; cd Plantilla; ls)&`
 - Ejecuta la orden2 si y solo si la orden1 ha tenido éxito
 - **Orden1 && orden2** `cd Plantillas && ls`
 - Ejecuta la orden2 si y solo si la orden1 no ha tenido éxito
 - **Orden1 || orden2** `cd Plantillas || ls`
 - Ejecuta la orden1 y encauza o envía su salida como entrada de orden2
 - **Orden1 | orden2** `ls | more`

Realizar estos comandos
sobre el directorio de
trabajo del usuario

Eliminación de significados especiales

- `\` Inhibe la interpretación del carácter siguiente (solo uno). Por ejemplo `"/n"` (salto de línea)

```
echo -n Hola \n mama
```

- `' '` Impide que el shell interprete todo lo incluido entre las comillas.

```
echo 'ls /home/cicloformativo' ls /home/cicloformativo
```

- `" "` Evita la interpretación de los caracteres incluidos, salvo `$`, `\`, `"`,

```
echo "ls /home/cicloformativo" ls /home/cicloformativo
```

- ``` (acento francés) Provoca la ejecución de la orden incluida.

```
echo `ls /home/cicloformativo`
```

UNA SESION EN LINUX

CONCEPTOS BASICOS I

- **SESIÓN**: tiempo que el usuario permanece dentro del sistema.
- **USUARIO**: persona que utiliza el sistema, y esta registrado mediante una CUENTA.
- **CUENTA**: Espacio del sistema reservado para cada usuario, o bien para un conjunto de ellos que trabajen en común.
- **GRUPO**: Conjunto de recursos y ficheros compartidos por un conjunto de usuarios.

CONCEPTOS BASICOS II

- **LOGIN:** Identificación de usuario en una cuenta.
- **PASSWORD:** Palabra de acceso al login introducido en una cuenta.
- **SUPERUSUARIO:** Persona administradora, encargada entre otras cosas de crear las cuentas de usuario, administrar y configurar el sistema. Es **root**.
- En Ubuntu viene desactivada, y utilizamos el comando **sudo** para realizar tareas administrativas como **root**. Introduciendo la contraseña del usuario que tiene permisos **admin** dentro de **sudoers**

Archivos de inicio del shell bash

- Al entrar en el sistema, durante el trabajo en una sesión y al salir del sistema, el SHELL ejecuta una serie de archivos que contiene ordenes para configurar el entorno y personalizar la sesión .
- Ejemplo:
 - Alias
 - Máscaras
 - umask

El Administrador debe asignar a un usuario para permitirle su conexión:

- Nombre de usuario (**login-name**).
- Asignarle un grupo (**group**)
- Una palabra de acceso (**password**)
- Asignarle un shell (**/bin/bash**)
- Asignarle los ficheros **.bash_profile** y **.bashrc** para configurar el entorno del usuario.

Ficheros relacionados con una sesión de usuario.



- Al iniciar una sesión se ejecutan los comandos:
 - 1º Archivo `/etc/profile`
 - 2º En su home `.bash_profile`, que a su vez ejecuta `.bashrc` (Para la shell hijas) --- Entorno trabajo ---
 - 3º Y llama `/etc/bashrc` configurado por `root`.
 - En modo texto se ejecutan al contrario.
- Iniciarle un fichero `.bash_logout`
- `UID` (identifica usuario), `GID` (Identifica al grupo). Se almacena en `/etc/passwd` y `/etc/group`.

Si no existe `bash_profile`, se busca `.bash_login` y en caso de Ubuntu `.profile`

Scriptps se usa `.bashrc` que ejecuta el archivo `/etc/bashrc`

Ficheros relacionados con una sesión de usuario.



- Conviene saber que:
 - `/etc/profile` , invoca los scripts contenidos en `/etc/profile.d` y `/etc/bash.bashrc` si existe
 - `/etc/skel` contiene `.bashrc` y `.profile`. Ficheros maestros que se copian en cada home del usuario
 - Para ejecutar de nuevos los ficheros sin reiniciar la sesión en la shell actual y no shell hija ejecutar:
 - `./etc/profile` //dejando un espacio entre pto y la barra
 - `sources ~/.bashrc` //emplea la orden sources

Cambiar el prompt

- Para cambiar el prompt basta con cambiar el valor de la variable de entorno `$PS1`
- Opciones:
 - `\h`: Hostname o nombre de la maquina*
 - `\T`: la hora actual (formato 24 h)*
 - `\t`: la hora actual*
 - `\d`: la fecha*
 - `\W`: El directorio actual en que nos encontramos*
 - `\w`: El directorio actual en que nos encontramos con path completo*
- Ejemplo:
 - `# PS1="Hola soy \u>"`
 - `# PS1="\u@\h> "`

Cambiar el prompt

- Para que dicho cambio se mantenga en todas las sesiones y no sólo en la shell actual.
- Añadir al final del archivo **.bashrc** que encontraras en tu directorio home, la siguiente orden: **“export PS1=configuracion_prompt”**

Más información:

<https://ubuntulife.wordpress.com/2008/10/25/cambiando-el-prompt-de-la-terminal/>

Variables de entorno (1/2)

- Es un conjunto de caracteres asociados a un nombre.
- Más comunes:

Variable	Descripción
DISPLAY	Donde aparecen la salidas de X-Windows.
HOME	Directorio personal.
HOSTNAME	Nombre de la máquina.
MAIL	Archivo de correo.
PATH	Lista de directorios donde buscar los programas.
PS1	Prompt.
SHELL	Intérprete de comandos por defecto.
TERM	Tipo de terminal.
USER	Nombre del usuario.

Variables de entorno (2/2)

- Define:
nombre_variable=valor
Ejemplo:
nuevo=/etc/default “sin espacios ambos lados =”
- Comprobar su valor
\$nombre_variable
Ejemplo:
echo \$nuevo
- Aplicar a todas las shell y a todas las sesiones
export nombre_variable
Ejemplo:
export nuevo
- Ver todas las variables de entorno definidas
env

Recomendaciones editor I

- Linux *distingue entre mayúsculas y minúsculas*.
- Las ordenes deben escribirse en minúsculas.
- Para **recuperar ordenes** anteriores, mediante la **FECHA ARRIBA**
- Para **seleccionar una orden ejecutada** anteriormente, nos desplazamos con la **fecha arriba** por las ordenes, y cuando aparezca la que deseamos, pulsamos **INTRO**.
- Para completar el nombre de un archivo, pulsar **TAB**
- La variable **\$HISTSIZE** nos guarda el número de ordenes a guardar.
- Para ver este valor, usar la orden **"echo \$HISTSIZE"**
- Para cambiar este valor **"HISTSIZE = nuevo_valor"**

Recomendaciones editor II

- La variable **\$HISTSIZE** nos guarda el número de ordenes a guardar.
 - Para ver este valor, usar la orden “**echo \$HISTSIZE**”
 - Para cambiar este valor “**HISTSIZE = nuevo_valor**”
- La variable **\$HISTFILE** (home/usuario/.bash_history)
 - Para ver este valor, usar la orden “**echo \$HISTFILE**”
 - Para cambiar este valor “**HISTFILE = nuevo_valor**”

Comando `history`

- Visualiza una lista numerada de todos los comandos anteriormente ejecutados.
- Sintaxis:
`# history`

Conexión al sistema

- Introducir:
 - Login
 - Password
- Aparece **Prompt**
 - # → Usuario administrador
 - \$ → Usuario normal

Desconexión al sistema

- Usar
 - EXIT (No es necesario rearrancar la maquina para reiniciar)
 - LOGOUT
 - CTRL + D

Apagar el ordenador

- Comando:
 - **shutdown** [-hr] tiempo [mensaje]
- Parámetros:
 - Tiempo: Indica cuando se va a realizar el cierre del sistema
 - Formas:
 - **hh:mm** → Indica la hora
 - **+m** → Indica los minutos que restan hasta el cierre
 - **now** → Inmediatamente (Equiv +0)
 - **-h** (Equiv **halt**) → Cierra todo el sistema y detiene la maquina
 - **-r** (Equiv **reboot**) → Cierra el sistema y reinicia la maquina

COMANDOS PARA CONOCER LA SINTAXIS DE LOS COMANDOS LINUX

Pedir ayuda en línea (1/3)

- Consultar la sintaxis de un comando, accediendo al manual.
- Dicha información es amplia y suele ser estar escrita mayoritariamente en inglés
- Sintaxis
 - # **man** comando
- Ejemplo
 - Como usar el comando ls
 - # **man** **ls**
- Para salir teclear **q**

Pedir ayuda en línea (2/3)

- Información breve. Generalmente en español
- Sintaxis
 - # Comando -help
 - # help comando
- Ejemplo
 - Como usar el comando ls
 - # ls --help
 - # help ls

Pedir ayuda en línea (3/3)

- Información breve.
- Sintaxis
 - # info comando
- Ejemplo
 - Como usar el comando ls
 - # info ls

Prompt

- Nombre de usuario
- Simbolo @ `cicloformativo@cicloformativo-virtual-machine:~$ pwd`
`/home/cicloformativo`
- Nombre del equipo
- Dos puntos (:)
- Directorio donde nos encontramos
- Simbolo \$ o # para root `root@cicloformativo-virtual-machine:~#`
- Ejemplo:
`cicloformativo@cicloformativo-virtual-machine:/$`
`cicloformativo@cicloformativo-virtual-machine:/home$`

AUTENTIFICARSE COMO ADMINISTRADOR (ROOT)

Más información: <http://www.ubuntu-guia.com/2012/08/comandos-su-y-sudo.html>

Cambiar de usuario

- **su** = switch user
- **Comando su**
 - Permite autenticarte con otro usuario en el mismo terminal en el que estas funcionando sin cerrar la sesión.
 - Sintaxis:
 - **su** <usuario>
 - Ejemplo:
 - # **su** manuel
 - Sintaxis
 - # **su** - (Me logo en esta sesión como usuario root, cargando su shell)
(Elimino las variables de entorno del usuario logado y cargo las de root)
- Para salir utilizamos el comando **exit**

Cambiar de usuario para ejecutar uno o varios comandos como root

- **sudo** = switch user do
- **Comando sudo**
 - Permite ejecutar un comando como root
 - Sintaxis:
sudo comando
 - Ejemplo:
sudo cat /etc/passwd
 - Si no deseo me vuelva a pedir la password durante la actual sesión
sudo -i (Me logo en esta sesión como usuario root, cargando su shell)
(Elimino las variables de entorno del usuario logado y cargo las de root)
- Para poder ejecutar sudo debes estar incluido en la carpeta /etc/sudoer
 - El usuario creado en la instalación esta dentro del grupo administradores o admin (usuarios sudoers)
- Para editar dicho archivo visudo

PRIMERAS ORDENES LINUX

Interpretación de la sintaxis

- **orden** { [- | +] opción..}.....[argumentoEntrada].. [argumentoSalida].. <enter>
- ESCRITURA:
 - **negrita** Debe ser digitado tal y cómo aparece, generalmente son nombres de comandos.
 - { }, [] Lo escrito entre corchetes o paréntesis es opcional.
 - .. Indican que lo que hay antes puede repetirse varias veces.
 - | Separa dos o más opciones/argumentos excluyentes para utilizar con la orden.
 - **normal** Son digitadas por el usuario según los nombre que él utilice.

Comando `clear`

- Permite borrar la pantalla
- Ejemplo:
 `# clear`

Comando `passwd`

- Para cambiar la contraseña de un usuario:
- Sintaxis
 - # `passwd` usuario
- Ejemplo
 - # `passwd` cicloformativo

Para cambiar mi propia password sólo utilizamos el comando `passwd` sin indicar el nombre de usuario

```
root@cicloformativo-virtual-machine:/home/cicloformativo# passwd
Introduzca la nueva contraseña de UNIX:
```

Comando `who`

- Usuarios conectados actualmente con el terminal asociado, fecha y hora de conexión.

- Sintaxis:

`who [opciones]`

- Opciones:

`am I` → Datos del usuario conectado.

`-u` → Usuarios que hay en el sistema.

`-T` → Indica si el terminal permite que los usuarios envíen mensajes al mismo.

`-H` → Imprime la línea de cabecera.

- Ejemplo:

– ¿con qué usuario estoy conectado?

`who am i`

```
root@cicloformativo-virtual-machine:/home/cicloformativo# who i am
cicloformativo pts/1      2015-02-16 22:54 (:0.0)
```

- Usuarios conectados al sistema y cuenta de usuarios

`who -q`

```
root@cicloformativo-virtual-machine:/home/cicloformativo# who -q
cicloformativo cicloformativo
Nº de usuarios=2
```

NOTA: PTS/1 siempre representa al entorno gráfico

Comando `tty`

- Muestra el controlador del terminal asignado y su ruta de acceso.
- Ejemplo:

```
root@cicloformativo-virtual-machine:/home/cicloformativo# tty  
/dev/pts/1
```

Comando **finger**

- Muestra información sobre los usuarios del sistema.
- Nombre del usuario, nombre del terminal, y estado. Tiempo de inactividad, tiempo de entrada, directorio del usuario, shell de entrada, estado del correo
- **Sintaxis:**
finger [opciones] usuarios

Comando logname

- Muestra el nombre de usuario con que se ha accedido al sistema
- Sintaxis:
logname
- Ejemplo:

```
root@cicloformativo-virtual-machine:/home/cicloformativo# logname  
cicloformativo
```

Comando que debe ser instalado para poderse ejecutar con `sudo apt-get install logname`

Comando id

- Muestra la identificación del usuario del sistema.
- Nombre, número de identificador, nombre del grupo e identificador del grupo.
- Sintaxis:

id [opciones]

```
root@cicloformativo-virtual-machine:/home/cicloformativo# id  
uid=0(root) gid=0(root) grupos=0(root)
```

Comando `uname`

- Muestra el nombre del sistema operativo y su versión
- Sintaxis: `uname [opciones]`
- Opciones
 - `-a` Muestra toda la información
 - `-s` Muestra nombre del kernel
 - ...

Comando `date` (1/2)

- Usuario normal:
 - `date`
- Usuario Administrador:
 - Sintaxis: `date MMDDhhmm [CC] [AA][.ss]` // Para cambiar fecha
`date +formato` // Para mostrar fecha en un formato determinado
 - Opciones:
 - `MM` mes (01-12)
 - `DD` día (01-31)
 - `hh` hora (00-23)
 - `mm` minutos (00-59)
 - `CC` Los primeros dígitos del año
 - `AA` El año (00-99)
 - `ss` Los segundos
 - `+formato` Controla como se muestra la fecha.
Formato se compone de un % seguido de un carácter.
(Ver ayuda)

Comando date (2/2)

- Ejemplos:

- Conocer la fecha del sistema, en el formato por defecto

date

- Cambiar la fecha del sistema a 3 de mayo, 12 horas 12 minutos.

date 05031212 (MMDDhhmm)

- Muestra la fecha en formato dd/mm/aa

date +FECHA=%D

Comando cal

- Muestra un calendario en pantalla.
 - Sintaxis: **cal [mes] [año]**
 - **Ejemplo:**
 - Muestra el calendario del mes de mayo del año 2006
- # cal 05 2006**

Comando echo (1/2)

- Muestra la cadena de texto especificada a continuación
- Sintaxis: `echo [-n] [-e] cadena`
- Parámetros:
 - n Suprime la actuación normal de echo, que consiste en que añade una nueva línea a continuación de la salida.
 - e Permite la interpretación de una serie de secuencias de caracteres en la cadena. Es admisible el carácter ESC.

Comando echo (2/2)

- Ejemplo:

- Imprime Hola en la salida estándar

- # **echo hola**

- Imprime hola y a continuación el prompt sin realizar salto de línea por la opción -n

- # **echo -n hola**

- Imprime Hola y el cursor realiza el salto de línea porque interpreta \c debido a la opción -e

- # **echo -e "Hola \c"**

```
root@cicloformativo-virtual-machine:/home/cicloformativo# echo -e "hola \n"
hola
```

```
root@cicloformativo-virtual-machine:/home/cicloformativo# echo "hola \n"
hola \n
```

```
root@cicloformativo-virtual-machine:/home/cicloformativo# echo -e "hola"
hola
```

```
root@cicloformativo-virtual-machine:/home/cicloformativo# echo -n "Hola"
Holaroot@cicloformativo-virtual-machine:/home/cicloformativo#
```

Comando alias

- Son sobrenombres que se le dan a ciertas cosas, como archivos u ordenes.
- Se utiliza para ejecutar comandos siempre con las mismas opciones, renombrar cosas o ignorar errores

- Sintaxis:

`# alias identificador = "orden"`

- Ejemplo:

`# alias dir = "ls"`

`# alias logot = "logout"`

`# alias rm = "rm -i"`

Comando unalias

- Permite eliminar un alias.
- Sintaxis:

unalias alias

- Ejemplo:

– Eliminar alias dir

unalias dir

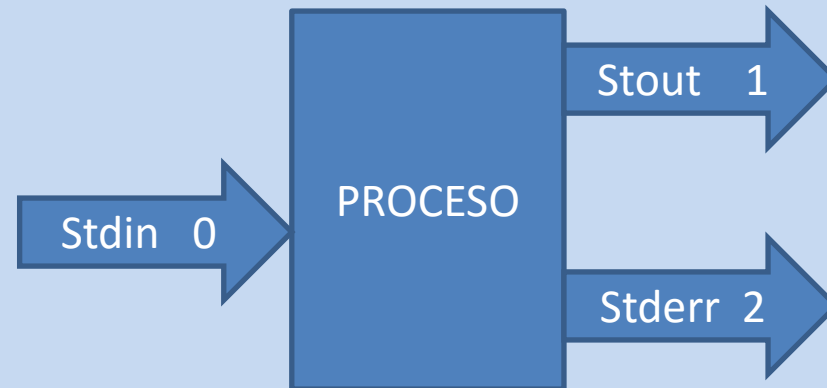
DEBEMOS CONOCER: TRABAJAR CON TUBERIAS

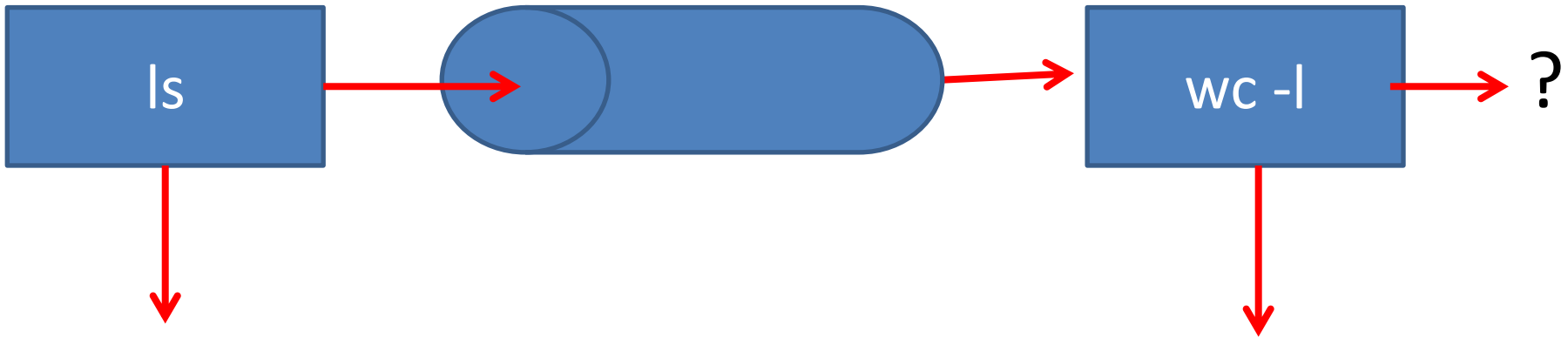
Más información:

<http://hipertextual.com/archivo/2014/07/redirecciones-y-tuberias-bash/>

Debes conocer que:

- Entrada estándar → teclado (stdin)
- Salida estándar → pantalla (stdout)
- Salida especial → Mensajes de error (stderr)





ls | wc -l

TUBERIAS

Redireccionamiento

- El **teclado** y la **pantalla** es tratada como **ficheros**, proporcionando una gran flexibilidad al sistema, permitiendo tratar a los dispositivos igual que a los ficheros.
- Existen 1 entrada (teclado) y 2 salidas (pantalla y de error).

Redireccionamiento de salida (1/2)

- Se utiliza el carácter **>**
- Se cambia la salida estandar (**stdout** = pantalla) por un **fichero**
- Ejemplo:

```
# cat fich1 fich2  > suma  
# cat fich1 fich2 1> suma
```

Envía el contenido de los **fich1** y **fich2** al fichero **suma**.
Si no existe le **crea** y de existir se **sobrescribiría**.

Redireccionamiento de salida (2/2)

- También se utiliza el carácter **>>** para añadir el contenido al final del fichero, en lugar de sobrescribirle

```
# cat fich1 fich2 >> suma  
# cat fich 1 fich2 1>> suma
```

Envía el contenido de los **fich1** y **fich2** al final del fichero
suma

Redireccionamiento de salida de error (1/2)

- Se utiliza para notificar errores o avisos durante la ejecución. Redirecciona la **stdout** a **stderr**
- Ejemplo:

```
cat fich1 fich2 2> errores
```

Redirecciona la salida de errores de la ejecución `cat fich1 y fich2` al fichero `errores`

- Para añadirse más errores sin ser sobrescrito.
- Ejemplo:

```
cat fich1 fich2 2>> errores
```

(sólo válida para la sh Bash)

Redirecciona la salida de errores de la ejecución sin ser sobrescrito

Redireccionamiento de salida de error (2/2)

- La salida estándar y la de errores del comando a ejecutar se guardarán en file.
- Si el archivo file existe se sobrescribirá.
- Ejemplo:

```
cat fich1 fich2 > errores 2>&1
```

- La salida estándar y la de errores del comando a ejecutar se guardarán en file. La información se escribirá al final del archivo.
- Ejemplo:

```
cat fich1 fich2 2>> errores 2>&1
```

Redireccionamiento de entrada

- Se utiliza el carácter `<` seguido de un **fichero**
- Redirige la entrada estándar (**stdin** = teclado) por el contenido de un fichero.
- Ejemplo:

```
cat < fich1
```

Visualiza el contenido del fichero 1

Redireccionamiento de entrada y salida

- Se pueden utilizar las dos direcciones a la vez.
- Ejemplo:

```
# sort < lista > ordenado
```

El comando sort ordena las líneas del fichero lista y almacena el resultado en el fichero ordenado

Encauzamiento

- Nos permite encauzar la salida de un programa con la entrada de otro, sin tener que controlar los ficheros temporales.
- Se conoce como tuberías, fifo o pipes.
- Se utiliza el carácter |
- Ejemplo:

ls | sort

Ejecuta ambos comandos.

Enlazando la salida de ls con la entrada de sort.

Nos muestra todos los ficheros del directorio actual, y aparecen ordenados.

Encauzamiento y redireccionamiento

- Es posible combinar ambos conceptos
- Ejemplo:

```
ls | sort > ordenado
```

Nos guarda el resultado de los dos comandos anteriores en el fichero ordenado.

COMANDOS PARA TRABAJAR CON FICHEROS Y DIRECTORIOS

Comando pwd

- Informa de cuál es el directorio actual

- Sintaxis:

pwd

- Ejemplo:

pwd

```
cicloformativo@cicloformativo-virtual-machine:~$ pwd  
/home/cicloformativo
```

Comando touch

- Cambia los tiempos de modificaciones y acceso de un archivo o crea un nuevo archivo con los tiempos especificados.
- Sintaxis:
touch [opciones] MMDDhhmmAA lista_archivos
- Opciones:
 - a Cambia la hora y fecha de acceso
 - m Cambia la hora y fecha de modificación
 - Por defecto si no se indica opciones toma los valores -am

Comando touch

- Ejemplos:

- Cambiar fecha de creación y modificación

```
# touch -t 0101120191 carta
```

Establece la fecha y hora de modificación y acceso a la hora indicada

```
# touch carta
```

Establece la fecha y la hora de modificación y acceso a la actual

- Permite también crear un fichero vacío cuando el fichero no existe.

```
# touch ficheronuevo
```

Comando **ls** (1/3)

- Lista el contenido de un directorio
- Sintaxis:
 - **ls [opciones] [lista de archivos]**
- Opciones:
 - a Muestra todos los archivos, incluidos los ocultos que comienzan por punto.
 - l Muestra amplia información de cada archivo.
 - x Muestra los archivos en columnas.
 - t Clasifica por fecha y hora de última modificación. Es utilizado con -l.
 - u Clasifica con fecha y hora del último acceso. Es utilizado con -t.
 - r Invierte el orden de clasificación.

Comando ls (2/3)

- Opciones:

- -i Muestra el número de inodo del archivo.
- -R Lista el contenido de los directorios recursivos
- -F Coloca un:
 - / después de las entradas de un directorio (azul)
 - * después de los programas ejecutables (verde)
 - @ después de los enlaces simbólicos (cian)
 - | después de un FIFO
 - Nada después de los archivos normales
- Curiosidad:
 - Las configuraciones del color se realizan en `/etc/DIR_COLORS`
 - Para personalizar para un usuario, copiar este archivo al directorio del usuario y modificar su nombre a `dir_colors`

Comando `ls` (3/3)

- Ejemplos:

<code>ls -l</code>	Muestra toda la información
<code>ls -a</code>	Muestra los archivos ocultos
<code>ls -la</code>	Muestra toda la información, incluso de los ocultos
<code>ls -lt</code>	Muestra toda la información, clasificador por fecha y hora última modificación
<code>ls -F</code>	Muestra los ficheros colocándolos un símbolo según su tipo.
<code>ls -Ri</code>	Muestra el número de inodo y el contenido de los directorios recursivos.
<code>ls -xaF</code>	Muestra en columnas, colocando un símbolo de distinción y muestra incluso los ocultos

Comando `mkdir`

- Crea un nuevo directorio

- Sintaxis:

`mkdir [nombre directorio]`

- Ejemplo:

```
mkdir direct1    Crea el directorio de nombre fich1
```

Rutas de acceso

- Rutas separan los directorios por / y sin dejar espacios en blanco
- Hay dos tipos:
 - **Rutas absolutas**
 - Aquellas que parten del directorio raíz (/)
 - Siempre son válidas
 - **Rutas relativas**
 - Ruta que parte del directorio actual como origen, relativa desde ese directorio
 - Sólo son válidas para indicar desde ese directorio y u otro.

Comando `cd` (1/2)

- Permite cambiar de directorio
- **Sintaxis:**
`cd [nombre directorio]`
- Caracteres permitidos para usar con el nombre del directorio:
 - /** Directorio raiz
 - .** Directorio actual
 - ..** Directorio padre
 - blanco** Directorio inicial del usuario
 - ~** Directorio home del usuario

Comando `cd` (2/2)

- `cd` sin argumentos, se activa el directorio inicial o del usuario.
- `cd ..`, permite ir al directorio padre
- `cd /`, permite a la raiz
- Ejemplo:

<code>cd direct1</code>	Cambia al directorio "direct1"
<code>cd ..</code>	Me permite ir al directorio padre de "direct1"
<code>cd direct1</code>	Cambia al directorio "direct1"
<code>cd /</code>	Me permite ir a la raiz

Comando `rmdir`

- Elimina un directorio

- Sintaxis:

`# rmdir [nombre directorio]`

- Ejemplo:

```
rmdir direct1      Elimina el directorio de nombre fich1
```

COMANDOS PARA MANIPULAR FICHEROS

- PAGINAR**
- VISUALIZAR**
- EDITAR**

Comando `cat` (1/3)

- Imprime en la salida estándar el contenido de uno o varios archivos

- Sintaxis:

`cat [opciones] lista_archivos`

– Opciones:

- `-b` Numera las líneas del fichero que no están en blanco, comenzando por 1.
- `-n` Numera, empezando por 1, todas las líneas de salida.

Comando `cat` (2/3)

- Ejemplo:

```
cat /etc/passwd
```

```
cat -b /etc/passwd
```

```
cat -n /etc/passwd
```

Visualiza y **concatena** por pantalla la visualización de ambos archivos

```
cat /etc/passwd /boot/grub/grub.cfg
```

Comando **cat** (3/3)

- Crear fichero con contenido. Salir con ^D

```
cat > ficherocrear
```

```
Uno
```

```
Dos
```

```
Tres
```

```
^D
```

- Crear fichero con contenido. Salir con la palabra que se indique

```
cat << END > ficherocrear
```

```
Uno
```

```
Dos
```

```
Tres
```

```
END
```

Comando **more**

- Muestra el contenido de un archivo de texto por pantalla pero deteniendose al completarse esta.
 - Al pulsar **INTRO** aparece una nueva línea
 - Al pulsar **ESPACIADORA** aparece una nueva pantalla
- Sintaxis:
more [opciones] fichero
- Opciones:
 - n** **n** es un entero que se utiliza para establecer el tamaño de la ventana a una longitud de n líneas.
 - c** Dibuja cada línea desde arriba hacia abajo limpiando la línea y siempre dibujando la siguiente.

more archivo

Muestra contenido de un archivo por páginas

Comando `less`

- Funciona igual que `more` pero permite recorrer el archivo con las teclas del cursor.
- Sintaxis
`less` archivo
- Ejemplo
`less` /etc/passwd

Comando **tail**

- Permite visualizar el final de un archivo de texto.
- Sintaxis:
tail [opciones] archivo
- Opciones:
 - n** Muestra las últimas n líneas
 - +n** Muestra a partir de la línea n
- Ejemplo:

```
# tail -3 fich
```

Muestra las 3 últimas líneas del fichero fich

Comando `head`

- Permite visualizar el principio de un archivo de texto.
- Sintaxis:
`head [opciones] archivo`
- Opciones:
-n [nº] Muestra las primera n líneas
- Ejemplo:

```
# head -3 fich
```

Muestra las 3 últimas líneas del fichero fich

Comando `sort`

- Permite ordenar y fusionar archivos de texto.
- Sintaxis:
 - `sort [opciones] archivos`
- Opciones:
 - `-u` Se asegura que sólo las líneas únicas pasan a la salida
 - `-r` Realizan una clasificación descendente
- Ejemplo:

```
sort -r fichero      Clasificación descendente de fichero
```


Comando `sort`

- Para archivos creados en columnas, utilizar las siguientes opciones
 - `r` Invierte el orden
 - `t` separador de las columnas
 - `k` campos por los que ordenar
 - `n` indica si el campo por el que se ordena es numérico

```
sort -r -t: -k5 -n archivo
```

Ordenado en orden inverso por el campo 5 que es numérico

Comando `cp` (1/2)

- Copia el contenido de un fichero en otro fichero.
 - **Sintaxis:** `cp [opciones] arch_origen arch_destino`
`cp [opciones] lista_origen direct_destino`
 - **Opciones:**
 - i Pregunta si se quiere sobrescribir archivos destino existentes.
 - r Copia directorios de forma recursiva, copiando todos sus ficheros en el destino pero sin copiar la estructura de directorios.
 - R Copia directorios de forma recursiva
 - u Sobrescribe archivos de destino ya existentes sólo si son más antiguos.
- Lista_origen** Varios archivos de origen

Comando `cp` (2/2)

- Ejemplo:

copia archivo1 a archivo2

```
# cp archivo1 archivo2
```

copia archivo1 y archivo2 en el directorio direc1 de la ruta especificada

```
# cp archivo1 archivo2 /home/user1/direc1
```

Comando **mv**

- Permite mover o renombrar archivos o directorios.
- La operación puede ser destructiva. No pide confirmación.

- Sintaxis:

```
# mv arch1 arch2
```

```
# mv dir1 dir2
```

```
# mv lista_de_archivos dir
```

- Ejemplo:

```
# mv arch1 arch2 /home/etc
```

Mueve los ficheros arch1 y archivo2 al directorio etc

Opción **-u**

Actualiza el destino con el fuente solo si este es más reciente

Comando **rm**

- Elimina archivos y estructuras completas de directorios.
- La operación puede ser destructiva. No pide confirmación.
- Sintaxis:
rm [opciones] lista_archivos
- Opciones:
 - r** Suprime los directorios especificados en la lista de archivos.
 - *-r** Suprime recursivamente los ficheros y directorios del directorio activo
- Ejemplo:

```
# rm /home/etc/pepe
```

Borra todos los archivos del directorio pepe, excepto los directorios

Comando **wc**

- Cuenta el número de caracteres, palabras y líneas de un archivo.
- Sintaxis:
`# wc [opciones] lista_archivos`
- Opciones:
 - c** Cuenta sólo el número de caracteres
 - w** Cuenta sólo el número de palabras
 - l** Cuenta sólo el número de líneas
- Ejemplo:

```
# wc fichero1
```

Cuenta el número de caracteres, palabras y líneas del fichero1

Comandos **ln** (1/2)

- Crea un enlace a un archivo, permitiendo la existencia de dos nombres para acceder a un mismo archivo
- Dos nombres con el mismo inode
- Tipos de enlaces:
 - *Completo*: Para eliminar un fichero previamente hay que eliminar todos sus enlaces.
 - *Simbólicos*: Si se elimina el fichero original se elimina el fichero, y los inodes se quedan colgando.

Comandos `ln` (2/2)

- Sintaxis:

`# ln [-s] arch_origen arch_destino`

`# ln lista_origen directorio_destino`

- Opciones:

Lista_origen: Lista de archivos para enlazar
separados por espacios.

-s Crea un enlace simbólico en vez de completo

- Ejemplo:

```
# ln arch1 enlace_arch
```

Al arch1 se crea un enlace de nombre enlace_arch

Comando file

- Determina el tipo de archivo (directorio, fichero, de texto, vacio, enlace ...)
- Sintaxis:
file lista_archivos
- Ejemplo:

```
# file etc
```

Me indica que se trata de un directorio

Comando `diff`

- Compara dos archivos de texto e informa de lo que hay que hacer en cada uno de ellos para que sea como el otro
- Opciones:
 - r compara recursivamente
 - q compara solo archivos diferentes
- Sintaxis:
 - # `diff [opciones] arc_viejo arch_nuevo`
 - # `diff [opciones] dir1 dir2`

Ejemplos

```
# diff nuevo.txt nuevo2.txt
```

Indica las líneas distintas de ambos ficheros

```
# diff -rq directorio1 directorio 2
```

Indica que ficheros hay diferentes entre ambos directorios

Comando `cmp`

- Compara dos archivos de texto
- Opciones:
 - l lista las diferencias entre ambos ficheros
 - s Muestra como son los archivos
 - 0 → son iguales
 - 1 → son distintos
 - > 1 → hubo un error
- Sintaxis:
 - `cmp [opciones] arch1 arch2`

Ejemplo:

```
# cmp fichero1 fichero 2
```

COMANDOS PARA HACER BÚSQUEDAS DE FICHEROS Y PATRONES

Comando **grep** (1/2)

- Busca patrones que están en los ficheros e informa cuando los encuentra.
- Sintaxis:
 - **grep [opciones] <patron> [ficheros]**
- Significado:
 - ☐ **patron** Puede ser una cadena o una expresión regular como “[Hh]i”
- Opciones:
 - ☐ **^patron** Busca la expresión al principio de la línea
 - ☐ **-v** Proporciona un listado de las líneas que NO contienen la expresión
 - ☐ **-c** Cuenta las líneas concordantes
 - ☐ **-i** Hace que no distinga entre mayúsculas y minúsculas
 - ☐ **-n** Imprime el número de línea que contiene el patrón
 - ☐ **-r** Busca en un directorio de forma recursiva (incluyendo subdirectorios) el patrón

Comando `grep` (2/2)

- Ejemplo:

Busca la cadena “root “ en el archivo `passwd`

`grep root /etc/passwd`

Busca al principio de la línea la palabra dos en el fichero `pruebagrep.txt`

`grep ^dos ./pruebagrep.txt`

Busca al principio de la línea la palabra dos o Dos en el fichero anterior

`grep ^[dD]os pruebagrep.txt`

Número de líneas en el fichero anterior que tiene la palabra dos

`grep -c dos pruebagrep.txt`

Líneas que no contienen la palabra que comienza por dos en el fichero

`grep -v ^dos pruebagrep.txt`

Numera las líneas que contienen la palabra dos en el fichero anterior

`grep -n dos pruebagrep.txt`

Busca en un directorio `/etc` todas las líneas que contienen la palabra `root`

`grep root /etc`

Comando `cut` (1/4)

- Se utiliza en ficheros de texto cuyos campos están separados por un delimitador, para filtrar por lo que se indique.
- Sintaxis:
`# cut opción ... [files] ...`
- Opciones:
 - `-d <delimitador>` define el delimitador
 - `-f <numero, rango>` Indica la columna a filtrar
 - `-c <numero, rango>` Indica los caracteres a filtrar

Comando `cut` (2/4)

- Ejemplo:

- Sea un fichero “pruebacut.txt” con este contenido, con delimitador <space>
 - Radamel Falcao Delantero AtleticoM
Lionel Messi Delantero Barcelona
Diego Lopez Portero Sevilla
Iker Casillas Portero RMadrid
Andres Iniesta Centrocampista Barcelona
- Sacar el primer campo en pantalla de un fichero delimitado por un espacio:

```
m3t4g4m3@nexolinux ~ $ cut -d " " -f 1 fichero.txt  
Radamel  
Lionel  
Diego  
Iker  
Andres
```


Comando `cut` (3/4)

- Ejemplo:

- Sea un fichero “pruebacut.txt” con este contenido, con delimitador <space>
 - Radamel Falcao Delantero AtleticoM
Lionel Messi Delantero Barcelona
Diego Lopez Portero Sevilla
Iker Casillas Portero RMadrid
Andres Iniesta Centrocampista Barcelona
- Sacar el primer y tercer campo en pantalla en un fichero delimitado por espacios:

```
m3t4g4m3@nexolinux ~ $ cut -d " " -f 1,3 fichero.txt  
Radamel Delantero  
Lionel Delantero  
Diego Portero  
Iker Portero  
Andres Centrocampista
```

Comando `cut` (4/4)

- Ejemplo:
 - Sea un fichero “pruebacut.txt” con este contenido, con delimitador <space>
 - Radamel Falcao Delantero AtleticoM
 - Lionel Messi Delantero Barcelona
 - Diego Lopez Portero Sevilla
 - Iker Casillas Portero RMadrid
 - Andres Iniesta Centrocampista Barcelona
- Sacar los 10 primeros caracteres de un fichero

```
m3t4g4m3@nexolinux ~ $ cut -c 1-10 fichero.txt
Radamel Fa
Lionel Mes
Diego Lope
Iker Casil
Andres Ini
```

Comando `find` (1/3)

- Busca y genera una lista de archivos que cumplen los criterios que se han indicado.
- Los archivos pueden ser comparados por:
 - Nombre `-name archivo` (Comodines * y ?)
 - Tamaño `-size n` (Bloques 512b, 1K)
 - Modalidad (rwx) `-perm modo` (Signo menos lo que no concuerda con ese modo)
 - Tipo (c,b,d,p,l,f) `-type x`
 - Nº Enlaces `-links n`
 - Id de usuario `-user id_user` (valor o nombre)
 - `-exec cmd` Por cada archivo se ejecuta cmd.
 - `{}` Indica donde debe aparecer el nombre del archivo en el comando ejecutado
 - `\;` Terminación del comando

Comando `find` (2/3)

- Las opciones pueden agruparse y combinarse para limitar los criterios de búsqueda.
- Deben cumplirse todas (**AND**)
- Otras opciones:
 - **\(\)** Se utilizan para agrupar selecciones
 - **-o** Sustituye al predeterminado AND
 - **!** Operador NOT

Comando `find` (3/3)

- Ejemplos:

```
# find . -name "carta*" -print
```

Busca en el directorio actual (.) y en sus directorios todos los archivos que empiecen por carta y los muestra por pantalla (-print)

```
# find . -name "carta*" -exec ls -l {} \;
```

Busca los archivos que empiezan por carta (*) y realiza un listado largo de ello (ls -l) en el directorio actual y sus subdirectorios

```
# find . ! \( -name "carta*" -o -name "*juan" \) -print
```

Busca una lista de archivos que no empiecen (!) por carta ni termine por juan y lo muestra por pantalla en el directorio actual y sus subdirectorios

Comandos para buscar archivos

which

Sintaxis: `which command`

Indica el directorio donde se encuentra el comando especificado en `command`.

find

Sintaxis: `find path -name file`

Busca los archivos con nombre `file` dentro del directorio `path` y subdirectorios.

Ejemplo:

`find /usr -name "*sch*"` → Busca en el directorio `/usr` y subdirectorios los archivos que su nombre contienen la cadena de texto `sch`.

Ejemplo:

`which` firefox

Comandos para buscar archivos

makewhatis

Sintaxis: makewhatis

Crea una base de datos que contiene todas las órdenes del sistema que aparecen en las páginas del manual online. Únicamente el usuario root puede ejecutar esta orden.

whatis

Sintaxis: whatis keywords

Busca una coincidencia exacta en la base de datos construida mediante la orden makewhatis.

apropos

Sintaxis: apropos keywords

Busca una coincidencia parcial en la base de datos construida mediante la orden makewhatis.

Ejemplos:

whereis mkdir

apropos passwd

PERMISOS EN FICHEROS Y DIRECTORIOS

Permisos

- Para ver los permisos

ls -l

```
lrwxrwxrwx 1 cicloformativo cicloformativo      9 mar 14 16:11 des -> Descargas
drwxr-xr-x 5 cicloformativo cicloformativo    4096 mar  1 17:54 Descargas
drwxr-xr-x 2 cicloformativo cicloformativo    4096 dic 23 16:33 Documentos
-rw-rw-r-- 1 cicloformativo cicloformativo    8917 ene  8 17:09 Ejemplo libreoffice.od
```

1 2 3 4 5 6 7 8 9 10 11 12

1 → Tipo de archivo

- **archivo normal**

d directorio

I enlace simbólico

2 → Permisos de usuario propietario (rwx)

3 → Permisos de grupo principal (rwx)

4 → Permisos para otros usuarios (rwx)

5 → Enlace duro

6 → Usuario propietario

7 → Grupo principal al que pertenece el usuario,
pero puede ser modificado y tomar un valor
secundario.

8 → Tamaño en bytes

9 → Fecha de creación

10 → Hora de creación

11 → Nombre del archivo

Formas de tratar los permisos

- Forma **simbólica** a través de caracteres
 - Forma **octal** a través de números
-
- Comando: **chmod**
 - Permite cambiar los permisos de un archivo (ficheros y directorios)

Permisos

- **Lectura**
 - **Directorio:** Leer los recursos almacenados en él, o sea listar las entradas del directorio.
 - **Archivo:** Leer su contenido
- **Escritura**
 - **Directorio:** Crear y borrar archivos sobre él, o sea las entradas del directorio
 - **Archivo:** Modificar su contenido
- **Ejecución**
 - **Directorio:** Podrá realizar otras funciones dentro de él mediante los permisos de lectura y escritura, o sea permiso de acceso a las entradas del directorio.
 - **Archivo:** Ejecutar el archivo

Cambiar permisos de archivos y directorios en modo simbólico

- Sintaxis

chmod [opciones] modo [,modo] ficheros

Opciones:

- **R** Permite cambiar de forma recursiva en un directorio.

- Grupos de usuarios

- **u**: usuario dueño del fichero
- **g**: grupo de usuarios del dueño del fichero
- **o**: todos los otros usuarios
- **a**: todos los tipos de usuario (dueño, grupo y otros)

- Permisos

- **r**: se refiere a los permisos de lectura
- **w**: se refiere a los permisos de escritura
- **x**: se refiere a los permisos de ejecución

Cambiar permisos de archivos y directorios en modo simbólico

- Se añade o quitar permisos con los operadores **+** y **-**.
- Para ello se indica el tipo de usuario y el permiso que se resta o añade.
- Ejemplos:
 - ✓ Cambia el permiso a todos los grupos de usuarios
`# chmod =r tuArchivo`
 - ✓ Remover los permisos de ejecución a grupos y otros
`# chmod g-x, o-x tuArchivo`
 - ✓ Añadir permiso de ejecución al usuario
`# chmod u+x tuArchivo`
 - ✓ Agrega y remueve a la vez
`# chmod u-x+w tuArchivo`

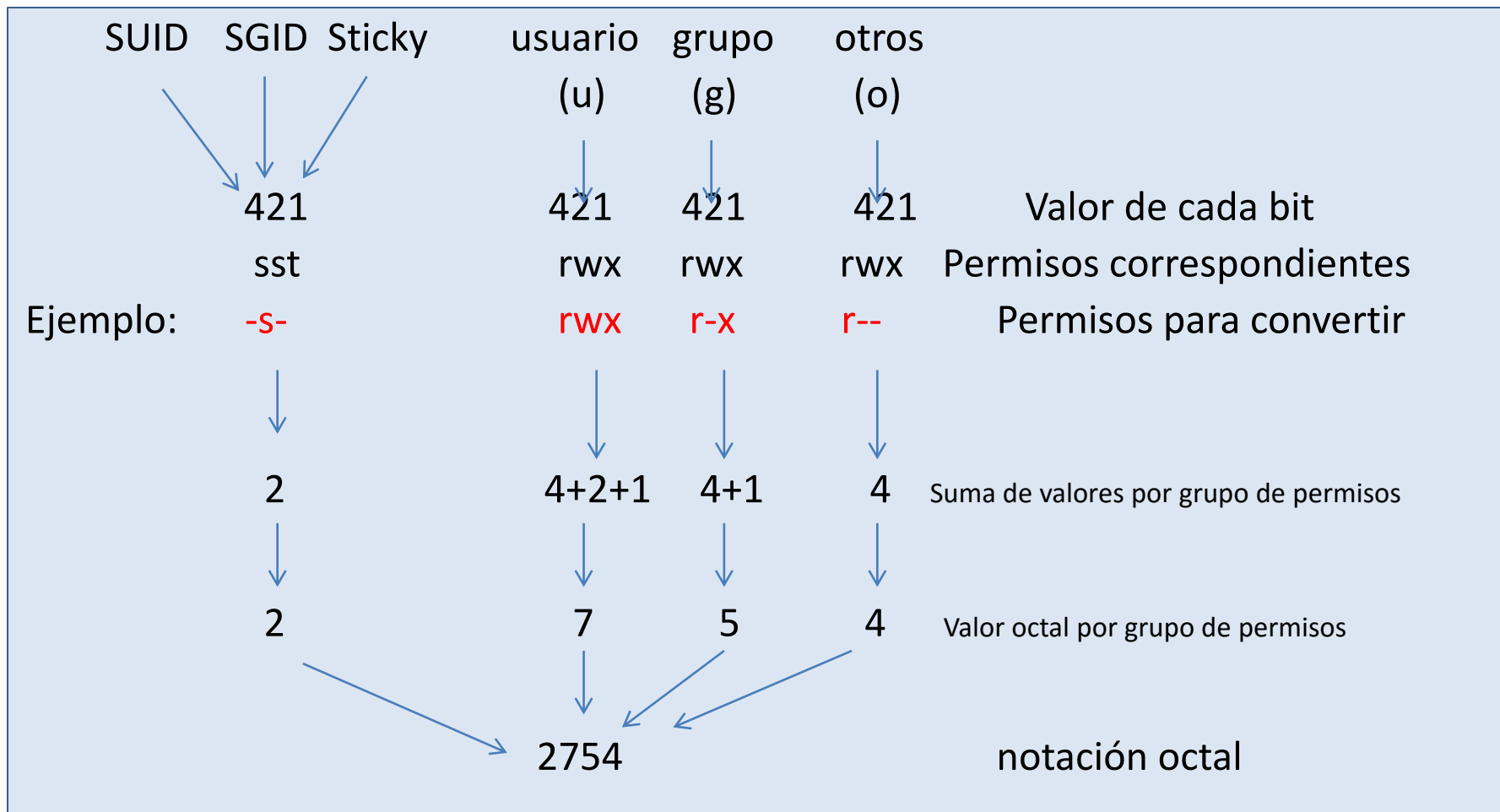
Cambiar permisos de archivos y directorios en modo octal

- Se basa en bit encendido (rwx) o apagado (-)
- La combinación de valores de cada grupo de usuarios (propietario-grupo-otros) forma un número octal (base 8), representado por tres bits y por tanto los valores pueden tomar valores de 0 a 7
 - $r = 4$ (lectura) = 2^2
 - $w = 2$ (escritura) = 2^1
 - $x = 1$ (ejecución) = 2^0
 - $- = 0$ (sin permisos) = 0
- Al asignar los permisos utilizamos números, para lo cual se sumarán los valores y será el que se coloque.

Cambiar permisos de archivos y directorios en modo octal

Permisos	Bits	Valor	Descripción
- - -	$0+0+0$	0	El valor cero significa que no se han asignado permisos
- - x	$0+0+0^0$	1	Sólo se asigna permiso de ejecución
- w-	$0+2^1+0$	2	Sólo se asigna permiso de escritura
- wx	$0+2^1+2^0$	3	Se le asigna permiso de escritura y ejecución
r - -	2^2+0+0	4	Sólo se asigna permiso de lectura
r - x	2^2+0+2^0	5	Se le asigna permiso de lectura y ejecución
rw -	2^2+2^1+0	6	Se le asigna permiso de lectura y escritura
rw x	$2^2+2^1+2^0$	7	Se le asigna permiso de lectura, escritura y ejecución

Cambiar permisos de archivos y directorios en modo octal



Cambiar permisos de archivos y directorios en modo octal

- Ejemplos:

Permisos	Valor	Descripción
rw- --- ---	600	Propietario permiso lectura y escritura
rw- --x --x	711	Propietario permiso lectura, escritura y ejecución. Grupo y otros sólo ejecución
rw- r-x r-x	755	Todos los permisos para el propietario y lectura y ejecución para el grupo y otros
r- r- r--	400	Sólo permiso de lectura para el propietario, grupos y otros.
rw- rw- rw-	777	Puede ser leído, escritura y otros para cualquiera
rw- r-- ---	640	El propietario puede leer y escribir y el grupo solo leer

Cambiar permisos de archivos y directorios en modo octal

- Ejemplos:

```
# chmod 000 foto.jpg
```

```
// quita todos los permisos al archivo foto
```

```
# chmod 700 foto.jpg
```

```
// añadido todos los permisos para el propietario
```

```
# chmod 114 foto.jpg
```

```
//sólo permiso de ejecución para propietario y grupo y lectura para otros
```

Comando **chgrp**

- Se utiliza para cambiar el grupo de un fichero.
- El grupo le puede cambiar el dueño si pertenece a ese grupo y root

- **Sintaxis:**

chgrp [opciones] grupo ficheros

— **Opciones:**

- **R** → Cambia en los directorios recursivamente el dueño

- **Ejemplo:**

chgrp asir mayo

-----	_	fuencis	fuencis	_____	_____	_____	_____	mayo
-----	_	fuencis	asir	_____	_____	_____	_____	mayo

Comando **chown**

- Se utiliza para cambiar el dueño de un fichero.
- El dueño de un fichero sólo lo puede cambiar el usuario root
- **Sintaxis:**

chown [opciones] dueño ficheros

— **Opciones:**

-R → Cambia en los directorios recursivamente el dueño

- **Ejemplo:**

chown raquel foto.png

chown -R root datos/

-----	fuencis	fuencis	_____	_____	_____	_____	foto.png
-----	raquel	fuencis	_____	_____	_____	_____	foto.png

- También permite cambiar usuario y grupo de una sola vez por root

Ejemplo:

chgrp fuencis:asir foto.png //uso como separador :

chgrp -R root. datos/ //uso como separador .

Ejercicios de compresión

- Apunta que puedes hacer o no en estos casos:
 1. Prueba asignar a un directorio únicamente el permiso de lectura (444) e intenta leer su contenido con un `ls` o leer alguno de sus ficheros de los cuales hayas apuntado previamente el nombre.
 2. Prueba asignar a un directorio únicamente el permiso de lectura (111) e intenta leer su contenido con un `ls` o leer alguno de sus ficheros de los cuales hayas apuntado previamente el nombre.
 3. Prueba asignar a un directorio únicamente el permiso de lectura (555) e intenta leer su contenido con un `ls` o leer alguno de sus ficheros de los cuales hayas apuntado previamente el nombre.

Comando `umask`

- Establece la maskara de permisos de directorio y de archivos, es decir los nuevos directorios o archivos que se crean obtienen el valor de los permisos a partir de los valores de `umask`

- Sintaxis

`umask` [opciones] [ficheros]

Opciones:

`-S` → Obtiene la maskara en formato simbólico

- Ejemplo:

`umask`

`umask -S`

`umask 0022` //establece la maskara a 0022

Comando `umask`

- Si el primer bit es 0
 - Para directorios o archivos ejecutables = (777)
 - Para archivos comunes = (666)
 - Los valores resultantes se obtienen restando:
- Ejemplos:
 - Si `umask` es 002, un archivo se crea con los permisos `rw- rw- r-` (666-002=664)
 - Si `umask` es 002, un directorio se crea con los permisos `drwxrwxr-x` (777-002=775)

umask

- Ejemplo:

- Directorios

111 111 111
000 010 100

111 101 011 (rwx r-x -wx)

- Ficheros

110 110 110
000 010 100

110 100 010 (rw- r- -w-)

Bits SUID, SGID y de persistencia (sticky bit)

```
cicloformativo@cicloformativo-virtual-machine:~$ ls -l /usr/bin/passwd  
-rwsr-xr-x 1 root root 41284 sep 13 2012 /usr/bin/passwd
```

- SUID

- Aparece en el grupo de usuario una **s** en lugar de **x**
- Archivo propiedad de root pero debe ser ejecutado por cualquier usuario, es decir puede evocarse un archivo propiedad de otro usuario.

- SGID

- Se aplica a un grupo en lugar de a un usuario.
- Aparece una **s** en lugar de **x** en los permisos de grupo.

- STICKY BIT

- Se aplica a directorios como /tmp y se indica con **t** en el grupo otros

```
cicloformativo@cicloformativo-virtual-machine:~$ ls -ld /tmp  
drwxrwxrwt 15 root root 4096 mar 14 18:17 /tmp
```

Bits SUID, SGID y de persistencia (sticky bit)

- Establecimiento de estos permisos especiales
 - Se utiliza el comando **chmod** pero agregando un número octal (**1 a 7**) extra al principio de los permisos
- Ejemplo:

```
cicloformativo@cicloformativo-virtual-machine:~$ touch mio
cicloformativo@cicloformativo-virtual-machine:~$ ls -l mio
-rw-rw-r-- 1 cicloformativo cicloformativo 0 mar 14 19:08 mio
cicloformativo@cicloformativo-virtual-machine:~$ chmod 4511 mio
cicloformativo@cicloformativo-virtual-machine:~$ ls -l mio
-r-s--x--x 1 cicloformativo cicloformativo 0 mar 14 19:08 mio
```

Observa el valor extra es 4, el resto de permisos se dejan como están (511)

Bits SUID, SGID y de persistencia (sticky bit)

- - - - -	= 0	Predeterminado, sin permisos especiales. No se requiere indicar nada
- - - - -t	= 1	Bit de persistencia, sticky bit
- - - -s - -	= 2	Bit SGID de grupo
- - - -s -t	= 3	Bit sticky y SGID
- -s - - - -	= 4	Bit SUID de usuario
- -s - - -t	= 5	Bit sticky y SUID
- -s -s - - -	= 6	Bit SUID de usuario y UGID de grupo
- -s -s -t	= 7	Bit sticky, SGID y SUID

OJO a los
problemas de
seguridad y a
un uso
inadecuado

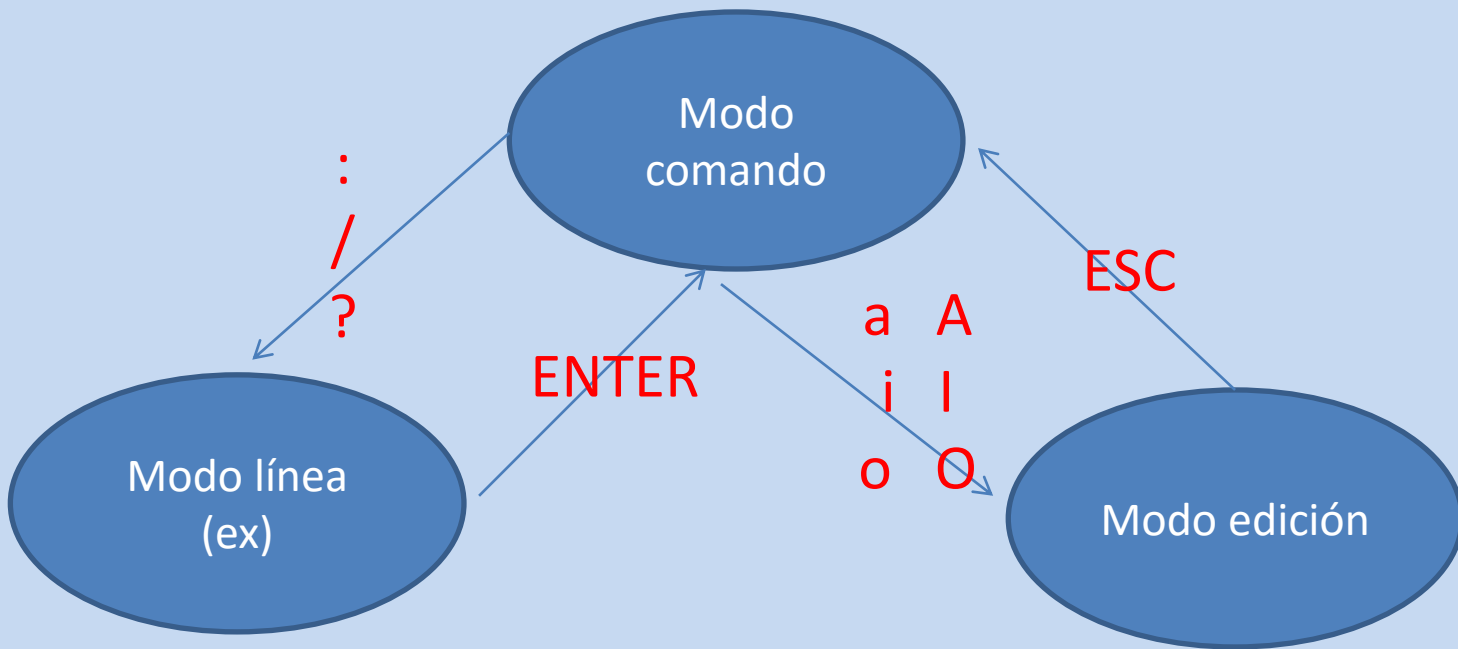
Comandos para editar ficheros para distribuciones debian

- **gedit**
 - Sintaxis: **gedit fichero**
 - Ejemplo: **gedit** /etc/mio.txt
- **nano**
 - Sintaxis: **nano fichero**
 - Ejemplo: **gedit** /etc/mio.txt
- **vi**
 - Sintaxis: **vi fichero**
- **vim** (Previa instalación. Es vi mejorado)
 - Sintaxis: **vim fichero**

vi

- Este editor tiene 3 modos de trabajo:
 - **modo comando**
 - Permite emplear cualquiera de los comandos de: movimiento del cursor, para borrar, alterar, desplazar texto.
 - Al pulsar “vi” se entra en este modo.
 - Para cambiar de un *modo texto a modo comando* pulsar **ESC**.
 - **modo texto**
 - Permite utilizar el teclado para introducir texto.
 - Se entra con **a, A, i, I, o, O**.
 - **modo línea**
 - Permite utilizar el comando **EX** precedido de :

vi



vi

- Consultar
 - Resumen de comandos del editor vi

Compresión y descompresión de archivos en linux

Comando tar (1/2)

- Sirve para empaquetar o desempaquetar archivos, pero no comprime cuando empaqueta.
- Sintaxis:
 - *tar <opciones> <archivo1> <archivo2> ...<archivoN>*
 - **<archivo1>** hasta **<archivoN>** es
 - la lista de archivos a añadir o extraer del archivo empaquetado.
- Opciones
 - **c** cree un nuevo archivo
 - **v** imprime los nombres de los archivos según se des/empaquetan
 - **f** nombre del archivo a des/empaquetar
 - **x** extraer o desempaquetar archivo
 - **t** mostrar un "índice" del archivo tar antes de desempaquetarlo
 - **r** Añade ficheros al final del archivo
 - **u** Se añaden ficheros al archivo si son nuevos o se han modificado.
 - **d** Copara ambos tar y muestra diferencias
- **OJO:** A las rutas. (ver ejemplos en siguiente diapositiva)

Comando tar (2/2)

- Ejemplo:

- **Empaquetar:**

- *# tar cvf backup.tar /etc* empaquetará todos los archivos de /etc en el archivo backup.tar.
- *# tar cvf backup.tar /etc/hosts /etc/group /etc/passwd (*)*

- **Desempaquetar**

- *# tar xvf backup.tar* Extraerá el archivo backup.tar en el directorio actual.
- *# cd /; tar xvf backup.tar (*)*
- *# cd /etc; tar cvf hosts group passwd*
- *# tar tvf backup.tar* Ver índice con ficheros

(*) Lo crea en la raíz, por eso primero me situó

Comando tar y gzip

- Agrupar y comprimir archivo
 - Extensiones: (.tar.gz .tar.z .tgz)
 - Opciones
 - **cz** comprime
 - Ejemplo:
 - **tar -czvf fichero.tar.gz /home/cicloformativo**
- Desagrupar y descomprimir archivo
 - Opciones
 - **xz** descomprime
 - Ejemplo:
 - **tar -xzvf fichero.tar.gz**
- Mostrar contenido de un archivo tar comprimido
 - Opciones
 - **tf** cree un nuevo archivo
 - Ejemplo:
 - **tar -tf fichero.tar.gz**

Comando gzip

- **Comprime o descomprime un archivo**

- `gzip [opciones] archivo`

- **Opciones**

- c El archivo original no se modifica. si no se pone se elimina a continuación.
- d Descomprime el archivo indicado
- r (des)comprime también los archivos de las eventuales carpetas
- numero Factor de compresión

- **Ejemplos:**

- `# gzip test` Comprime y se pierde fichero original
- `# gzip -c test > test.gz` Evita perder el fichero original
- `# gzip -9 backup.gz` Mayor factor de compresión
- `# gzip -d test.gz` Descomprime un fichero

Comando gunzip

- Descomprime un archivo comprimido con gzip.
- Sintaxis:
ungzip archivo
- Es lo mismo que utilizar: "**gzip -d**".
- Ejemplos:
gunzip archivo.gz Descomprime archivo.gz
gunzip -c archivo.gz > nuevo Descomprime **archivo.gz** se redirecciona la salida al archivo **nuevo**.

Ejemplos combinados:

- Comprimir y empaquetar
 - # tar cvf backup.tar /etc //empaqueto
 - # gzip -9 backup.tar //comprimo
- Desempaquetar
 - # gunzip backup.tar.gz //descomprimo
 - # tar xvf backup.tar //desempaqueto

Comando zcat

- Permite ver el contenido de un archivo comprimido sin descomprimirlo
- Ejemplo:
 - `zcat <archivo-de-texto-comprimido>`

Comando ZIP

- Empaqueta y comprime archivos
- Sintaxis:
 - **# zip <nombre_archivo_comprimido> <archivos_a_comprimir>**
 - **# zip <nombre_archivo_comprimido> ***
- Opciones:
 - **r** comprime todos los subdirectorios y archivos bajo el directorio donde estamos
 - **i** Incluye solo los archivos que especificamos
 - **f** agrega los cambios de nuestros archivos en el zip, no los nuevos
 - **q** elimina todos los mensajes informativos
 - **u** actualiza los archivos que hemos modificado, y también agrega archivos a nuestro .zip
 - **x** excluir uno o varios archivos en nuestro .zip
 - **y** guardar los links en el zip como links en vez de guardar el archivo al que apunta el link
- Ejemplo:
zip todo -i *.txt

Comando UNZIP

- Descomprimir archivos
- Ejemplo:
 - # unzip <nombre_archivo_comprimido>

Comandos

bzip2/bunzip2/bzcat/bzip2recover

- El comando bzip2 comprime archivos usando el algoritmo con el doble compresión .
- Extensión es **.bz2**
- Es similar a gzip y tiene las mismas opciones
- Para comprimir → **bzip2.**
- Para descomprimir → **bunzip2.**

Comando cpio

- Empaqueta para después comprimir
- Permite mandarlo a:
 - Pantalla
 - Archivo
 - Dispositivo, como una cinta

Comando cpio

- **Opciones:**

- **-o** (El modo copia-afuera)

- Empaquetar

- **-i** (El modo copia-adentro)

- Desempaquetar.

- **-p** (El modo copia-de-paso)

- Ver

- **Otras opciones menos frecuentes:**

- **- c** Escribe una cabecera en formato ASCII

- **- d** Vuelve a crear una estructura de directorio

- **- m** Retiene la fecha modificación actual

- **- t** Visualiza tabla de contenidos

- **- u** Restauración incondicional

- **- v** Visualiza la lista de ficheros copiados

- **- x** Maneja ficheros especiales de dispositivos

Comando cpio

- Ejemplos:

- `# cpio -iv < archivo.cpio`
 - Sacar los archivos del paquete
- `# cpio -iuv < archivo.cpio`
 - Igual que el anterior comando pero sobrescribe los archivos que existan.
- `# cpio -tv < archivo.cpio`
 - Ver los archivos que contiene el paquete
- `# ls | cpio -ov > archivo.cpio`
 - Ver los archivos que contiene el paquete

Comando unrar

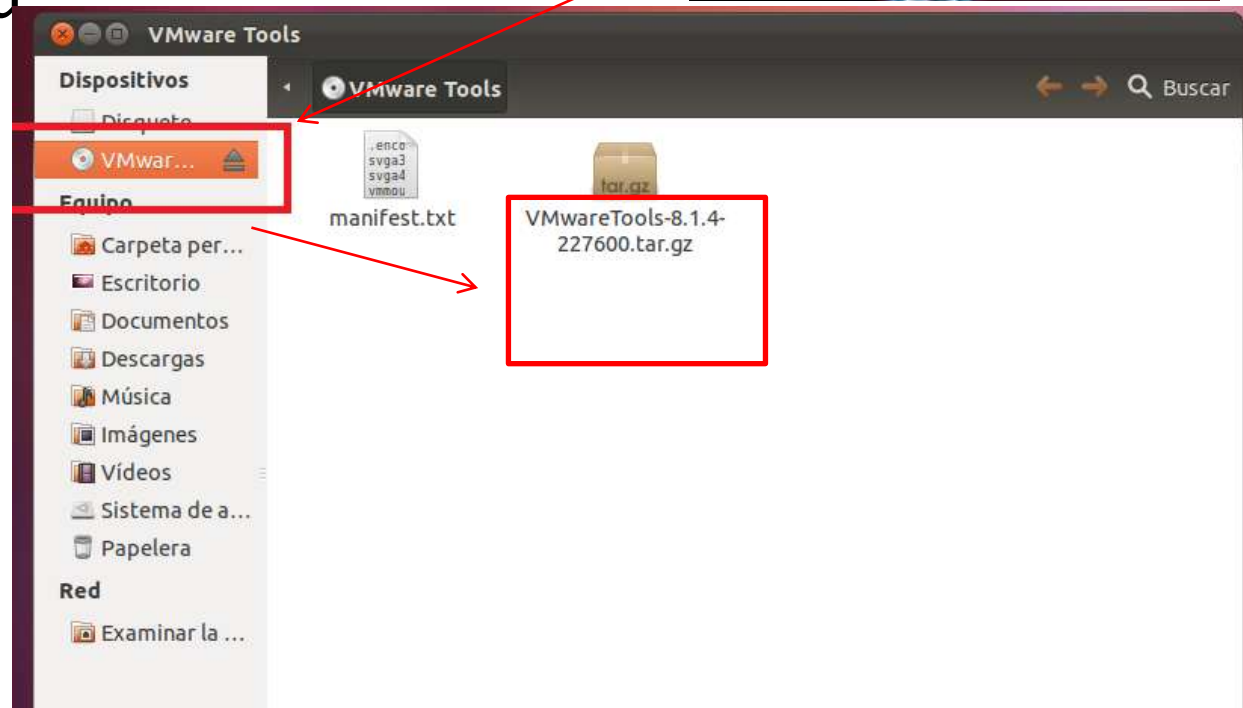
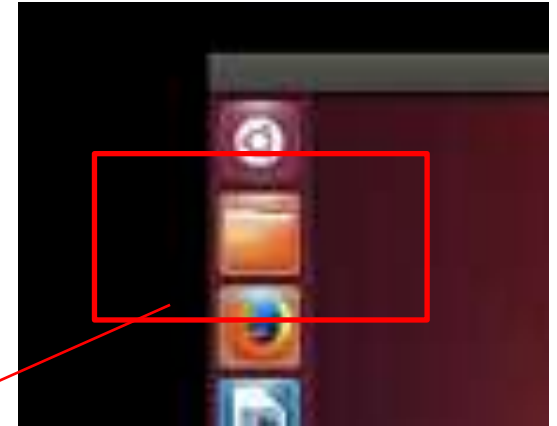
- Permite desempaquetar y descomprimir los archivos que desde Windows han sido empaquetados y comprimidos con el programa propietario rar.
 - Para comprimir:
 - `rar a nombre_fichero_comprimido.rar nombre_fichero_a_comprimir`
 - `rar a nombre_fichero_comprimido.rar *`
 - Para descomprimir:
 - `unrar x nombre_del_rar.rar`
 - `unrar x nombre_del_rar.rar /ruta/destino/descomprimido`

¿CÓMO INSTALAR LAS TOOLS EN VMWARE?

Paso 1

- Con la máquina virtual arrancada, pulsar en el menú **VM / Install vmware Tools**

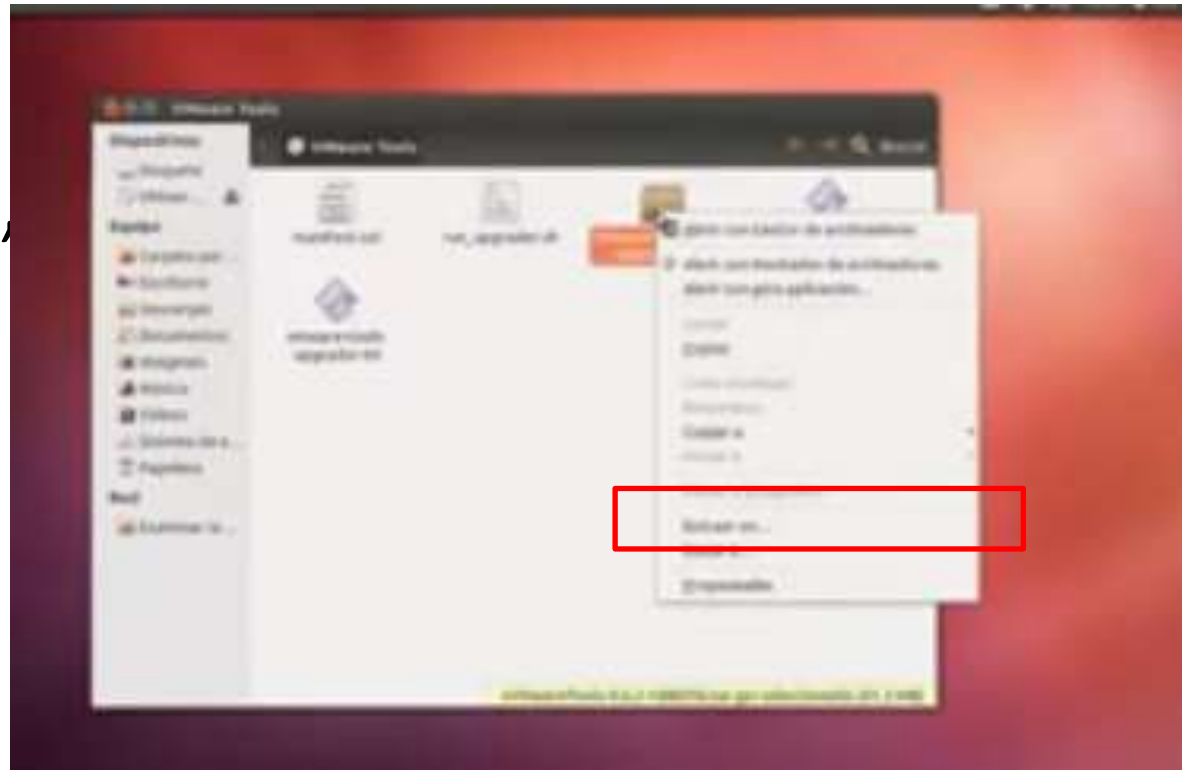
Seleccionar sistema de archivos



Nos monta la unidad CD

Paso 2

- Dar sobre el archivo botón derecho
“extraer en”
- Descomprimir en el “Escritorio”



Paso 3

https://youtu.be/6p_gVnIGfSg

Video borroso pero muy bien explicado

- Abrir un terminal
- Entrar como **superusuario** con “sudo su”
- Situar en el Escritorio y entrar en la carpeta que se nos ha creado haciendo uso de los comodines o tecla TAB
- Ejecutar el comando: **./vmware-install.pl**
- Por defecto dar a todo **enter**
- Reiniciar la máquina

```
kayron@kayron:~/Escritorio/vmware-tools-distrib$ sudo perl vmware-install.pl
[sudo] password for kayron:
Creating a new VMware Tools installer database using the tar4 format.
Installing VMware Tools.
In which directory do you want to install the binary files?
[/usr/bin]
What is the directory that contains the init directories (rc0.d/ to rc6.d/)?
[/etc]
```

Con comandos

- 1. Primero actualizamos el sistema
`sudo apt-get update && apt-get upgrade`
- 2. Luego instalamos las librerías GCC
`sudo apt-get install build-essential`
- 3. Creamos la carpeta `/media/cdrom` por si no está
`sudo mkdir /media/cdrom`
- 4. Una vez creada la carpeta montamos la imagen de VMTools. Para ello tenemos que indicar a la MV que queremos instalar las tools para que nos coloque la imagen en la unidad de CDROM
`sudo mount /dev/cdrom /media/cdrom`
- 5. Copiamos el archivo a la carpeta `/tmp` para poder instalarlo
`cp /media/cdrom/VMware*.tar.gz /tmp`

Con comandos

- 6. Desmontamos el CDROM

sudo umount /media/cdrom

- 7. Ingresamos a la carpeta /tmp y descomprimos el archivo VMware*.gz

Ejemplo: VMwareTools-8.3.7-381511.tar.gz

cd /tmp

- # **tar xzvf VMware*.gz**

- 8. Ingresamos a la carpeta creada y ejecutamos la instalación

cd vmware-tools-distrib/

sudo ./vmware-install.pl

- 9. Presionar a todo yes, la instalación por default. Reiniciamos el servidor

sudo reboot

**PARA USAR EL COMANDO APT-GET
NECESITO CONOCER**

¿Tienes proxy?

- En caso afirmativo sigue estos pasos:
 - `/etc/apt/apt.conf` para usar *apt-get*
 - **Navegador** que uses si quieres *navegar por internet*

Archivo apt.conf

- `Acquire::http::proxy "http://proxy.company.com:80/";`
- `Acquire::https::proxy "https://proxy.company.com:80/";`
- `Acquire::ftp::proxy "ftp://proxy.company.com:80/";`
- En nuestro caso, bastará con poner en dicho fichero:

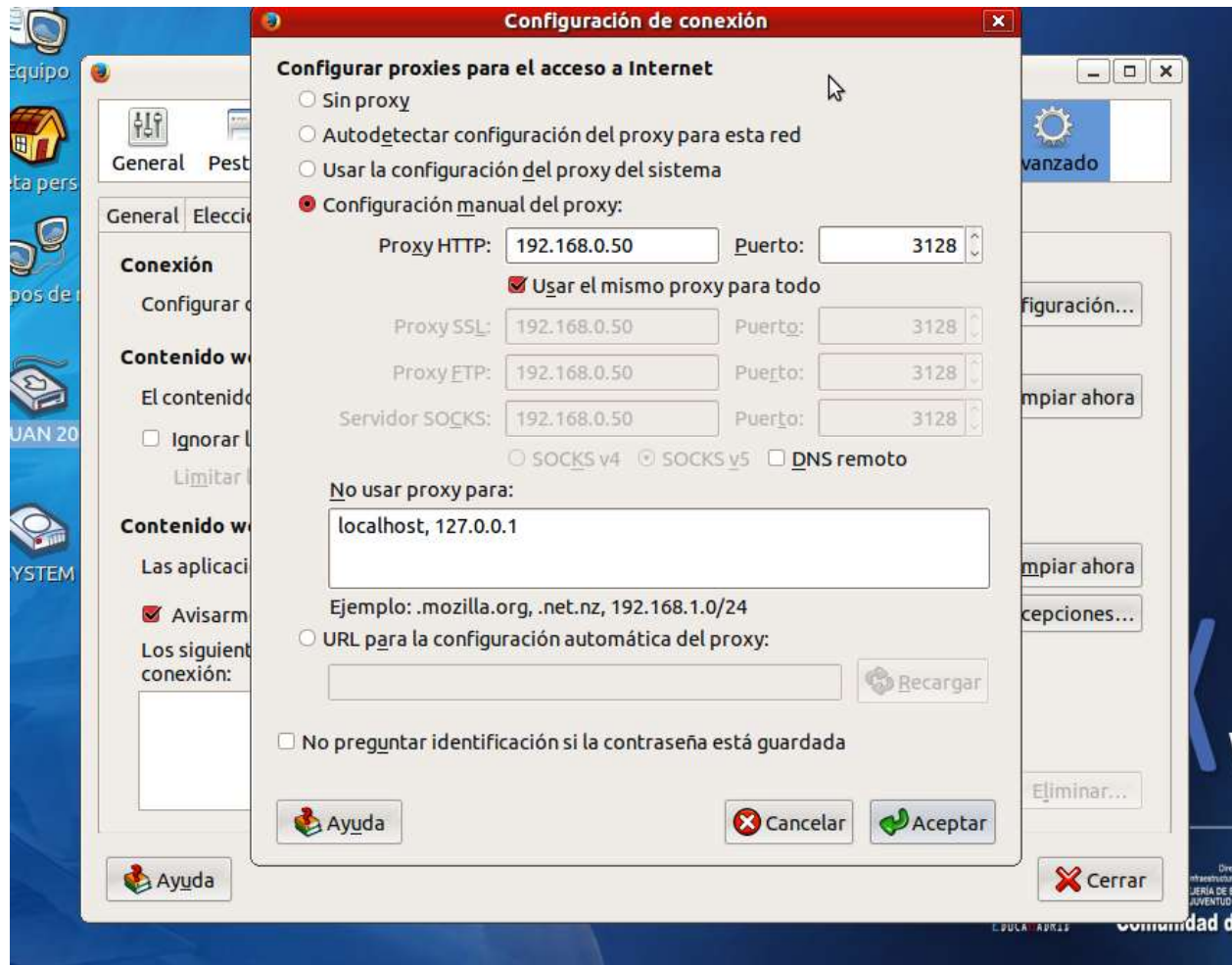
`Acquire::http::proxy "http://192.168.0.50:3128/";`

Navegador Mozilla

- Si queremos navegar desde MV de Ubuntu, abrimos Mozilla y desde **Opciones**, elegimos la opción **Avanzado, Configuración**

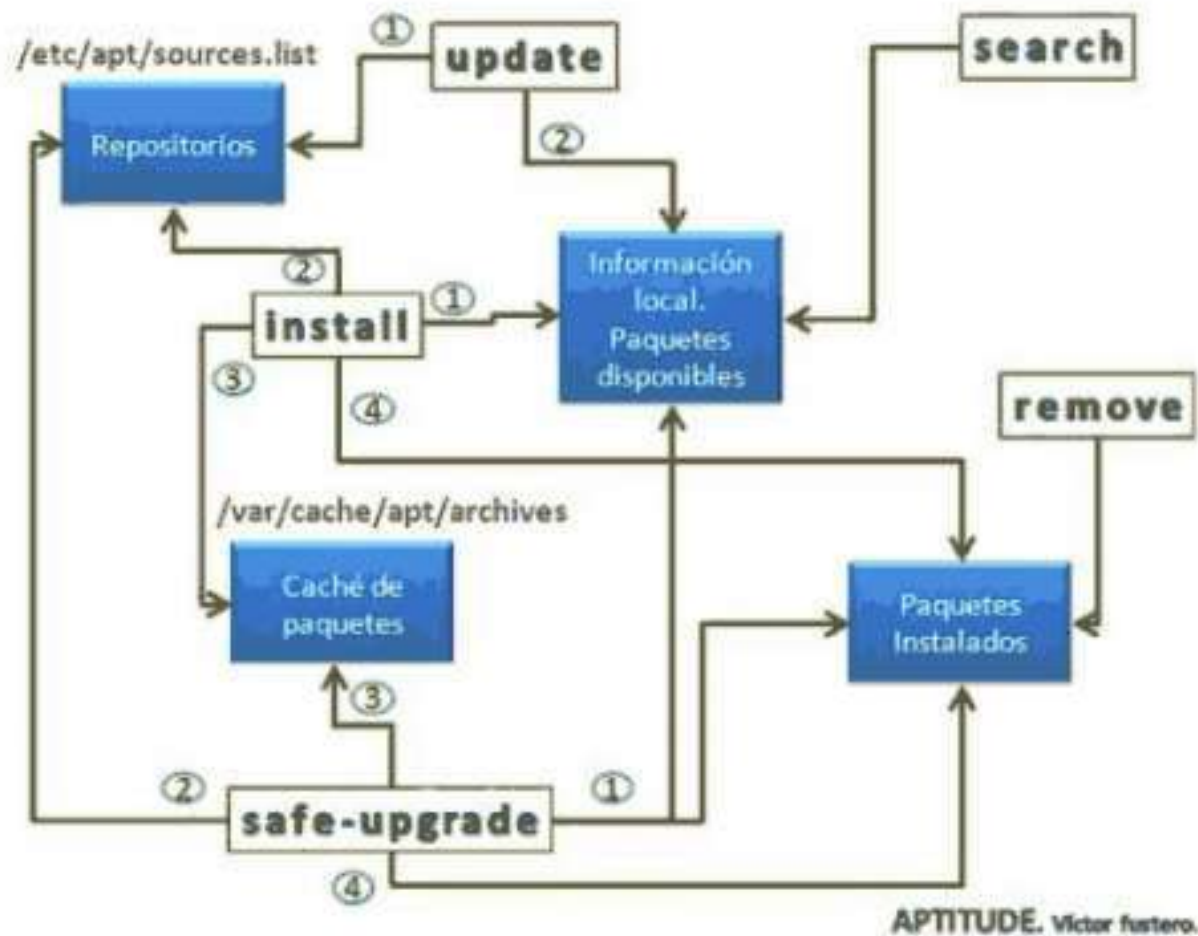


Navegador Mozilla



¿CÓMO INSTALAR SOFTWARE?

Instalación paquetes → apt o aptitude



Los repositorios (1/3)

- Es el lugar local o de internet donde el comando `apt` pueda buscar los paquetes
- En dicho archivo se encuentra las direcciones donde conseguir los paquetes)
- Las fuentes se encuentran en el archivo:
`/etc/apt/sources.list.`

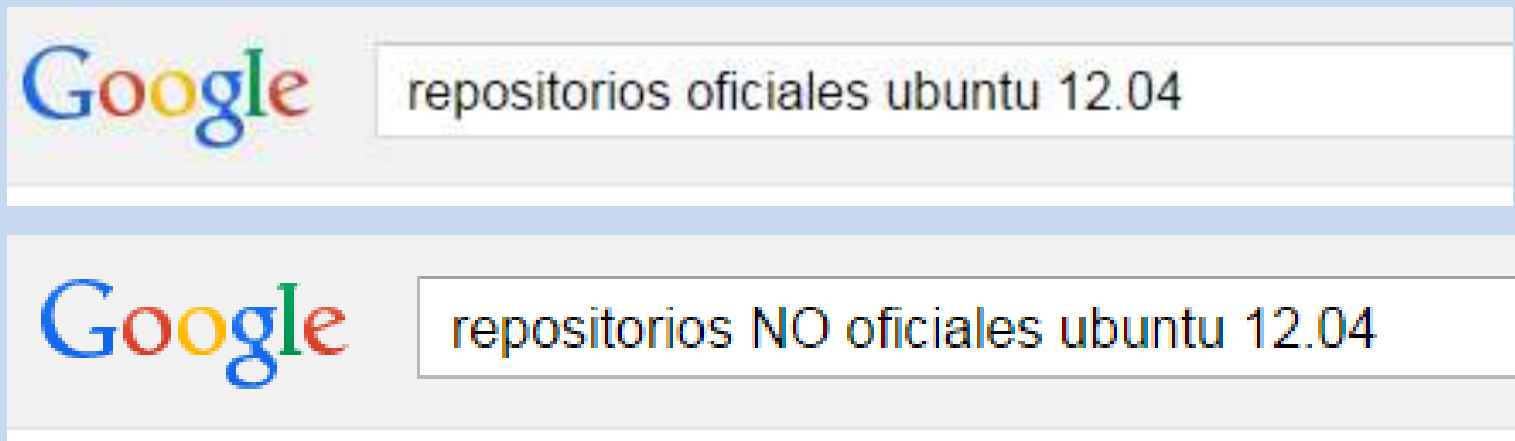
Paquete es un conjunto de programas.
Un programa puede estar formado por varios paquetes

Los repositorios (2/3)

- Encontraremos información como:
 - *# Repositorios*
 - *# Fuentes*
 - *# Método FTP*
 - *# Mirror Oficial de Ubuntu*
 - *# Mirror Oficial de Debían*
 -
- Las líneas comentadas (precedidas de '#') no las tiene en cuenta apt, el resto de líneas son ejecutadas.
- Se puede agregar más fuentes editando dicho fichero como root
- Se puede descomentar alguna de las líneas existentes o escribir manualmente nuevas líneas
- Puede añadir un CD-ROM a la lista de fuentes:
 - *# apt-cdrom add*

¿Dónde buscar repositorios? (3/3)

- www.apt-get.org
- www.google.es



Para añadir repositorios desde el terminal:

<http://elblogdeliher.com/como-anadir-un-repositorio-desde-la-terminal-en-ubuntu/>

apt

- **APT** (*Advanced Packaging Tool*)
 - Es el gestor de paquetes usado por Debían y las distribuciones basadas en Debían, como Ubuntu
 - Debe ser usado preferiblemente con conexión a internet.
 - Para usarle debemos decirle:
 - 1. Decirle que actualice la lista de paquetes disponibles
 - 2. Decirle los paquetes que queremos Instalar/desinstalar

¿Qué hace el comando apt?

- Buscar los paquetes en Internet o localmente.
- Solucionar las dependencias
 - A veces para que un paquete funcione necesita que haya otros paquetes instalados previamente: se dice que 'depende de'
- Descargar de Internet (cuando sea necesario) los paquetes y sus dependencias
- Instalar y configurar en el orden correcto los paquetes y sus dependencias
- Sintaxis:
apt-get [opciones] ordenes [paquetes]

Comando `apt-get` (1/7)

- Actualiza lista de paquetes (siempre antes de usar el comando `apt-get`)

`apt-get update`

Ejemplo:

```
# apt-get update
```

- Actualiza todos los paquete

`apt-get upgrade`

Ejemplo

```
#apt-get upgrade
```

- Para actualizar a otra distribución (Seguramente necesites instalar dicho comando)

`apt-get dist-upgrade`

Comando `apt-get` (2/7)

- Instalar paquetes (Escribir sin extensión ni coletilla)

`apt-get install` paquete

Ejemplos

```
# apt-get install gparted
```

```
# apt-get install gnome-panel
```

- Instalar paquetes cuando hay algún problema por ejemplo con las dependencias

`apt-get -f install` paquete

Ejemplo:

```
#apt-get -f install vlc
```

Comando `apt-get` (3/7)

- Reinstala un paquete cuando se daña o para instalar una versión más reciente.

`apt-get --reinstall install` paquete

Ejemplo:

```
#apt-get --reinstall install vlc
```

- Descarga el paquete a la cache pero sin instalarlo.

`apt-get -d install` paquete

Ejemplo:

```
#apt-get -d install vlc
```

Comando `apt-get` (4/7)

- Instala y desinstala paquete en una orden.

Los paquetes que llevan detrás un “-” serán desinstalados

```
# apt-get install paquete paquete- paquete
```

Ejemplo:

```
# apt-get install gparted vlc- gnome-panel
```

```
# apt-get install gparted- vlc- gnome-panel
```

- Simula una instalación pero sin instalar

```
# apt-get -s install paquete
```

Ejemplo:

```
# apt-get -s install vlc
```

Comando `apt-get` (5/7)

- Desinstala un paquete y borra también sus dependencias.

`apt-get remove` paquete

Ejemplo:

```
#apt-get remove gparted
```

- Desinstala el paquete y borra también todos los ficheros de configuración.

`apt-get --purge remove` paquete

Ejemplo:

```
#apt-get --purge remove gparted
```

Comando `apt-get` (6/7)

- Los paquetes instalados tienen extensión `.deb`
- Los paquetes instalados se conservan en la caché: `/var/cache/apt/archives`.
- *Se pueden borrar pero si necesitamos instalarlos de nuevo, **apt** tiene que volver a descargarlos*
- Borra la cache

`# apt-get autoclean`

Ejemplo `#apt-get autoclean`

Comando `apt-get` (7/7)

- Borra de la cache los paquetes antiguos que ya tienen nueva versión

`# apt-get clean`

Ejemplo

```
#apt-get clean
```

Comando `apt-cache` (1/2)

- Busca en la cache por el nombre o la descripción de un paquete, la palabra que se indique

`apt-cache search palabra`

Ejemplo

```
#apt-cache search particiones
```

- Muestra información de un paquete

`apt-cache show paquete`

Ejemplo:

```
#apt-cache show vlc
```


Comando `apt-cache` (2/2)

- Nos informa sobre las dependencias de 'paquete', es decir, de qué paquetes depende

`# apt-cache depends paquete`

Ejemplo

```
#apt-cache depends vlc
```

Comando `apt-show`

- Lista de paquetes que pueden ser actualizados por una nueva versión
- Necesita instalar el paquete ***apt-show-versions***.

`apt-show-versions -u`

Ejemplo

```
#apt-show-versions -u
```

Comando **aptitude** (1/4)

- aptitude debe ser previamente instalado

```
#apt-get install aptitude
```

- Instalar paquetes

```
# aptitude install <paquete>
```

Ejemplo:

```
#aptitude install vlc
```

Comando **aptitude** (2/4)

- Desinstalar paquetes

aptitude remove <paquetes>

Ejemplo:

```
#aptitude remove vlc
```

- Desinstalar paquetes incluidos ficheros de configuración.

aptitude purge <paquetes>

Ejemplo:

```
#aptitude purge vlc
```

Comando **aptitude** (3/4)

- Actualizar lista de paquetes

aptitude update

Ejemplo:

```
#aptitude update
```

- Actualiza el sistema.
- Instala las nuevas versiones de los paquetes instalados

aptitude safe-upgrade

Ejemplo:

```
#aptitude safe-update
```

Comando **aptitude** (4/4)

- Buscar por el nombre

aptitude search <nombre>

Ejemplo:

```
#aptitude search gparted
```

- Obtener ayuda del comando.

aptitude help

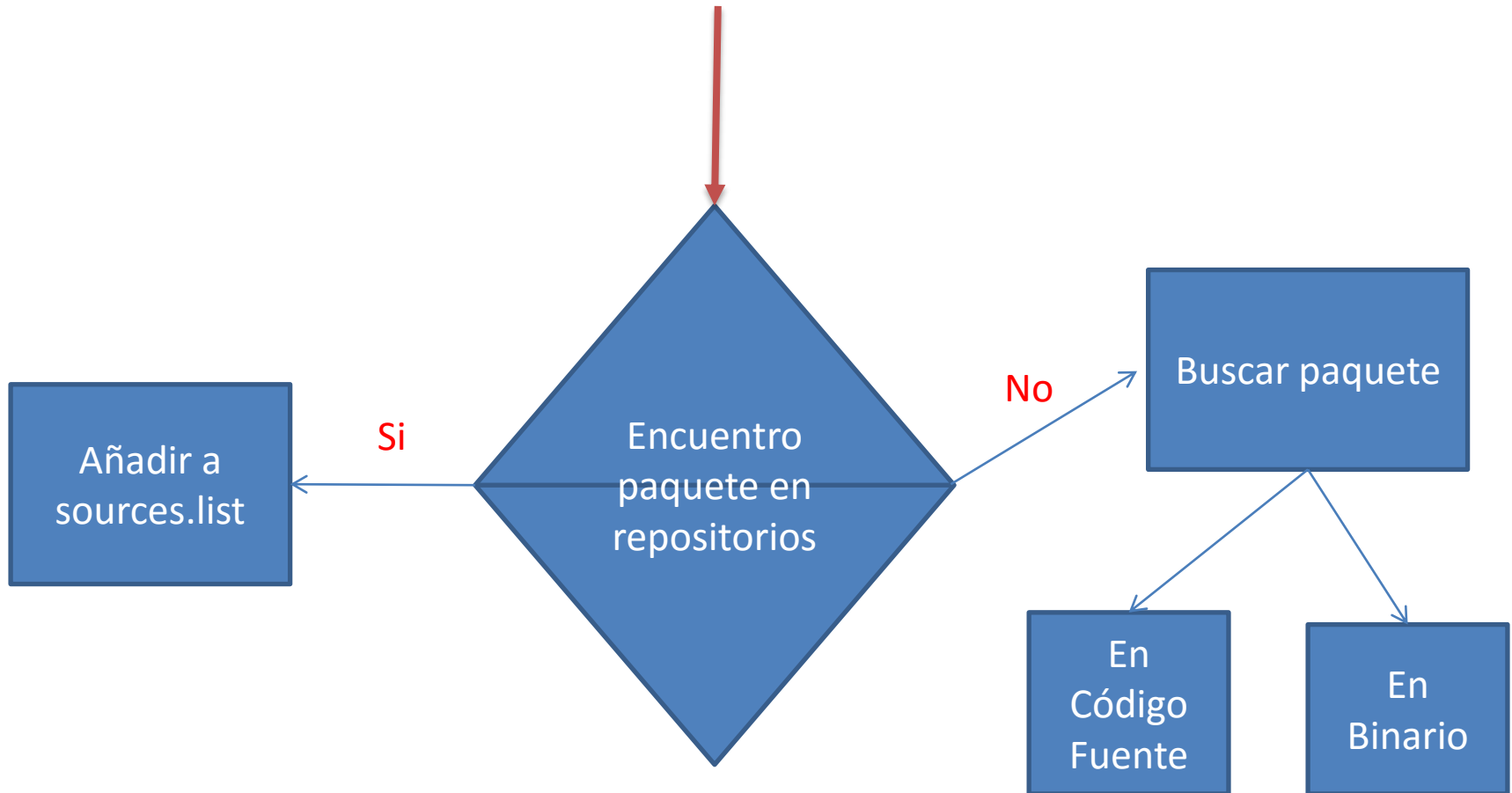
Ejemplo:

```
#aptitude help
```

Otros comandos con aptitude

- Eliminar la cache
`aptitude clean`
- Descargar paquetes y sus dependencias sin instalarlos.
`aptitude install -d nombre_paquete`
- Obtener información de un paquete
`aptitude show paquete`

El paquete **no** esta en el repositorio



FICHEROS BINARIOS

Instalar archivos binarios

(Ejecutar archivos ejecutables)

- Son archivos con extensión **.bin**
- Suelen ser archivos comerciales.
- Si no tiene permiso ejecución, concedérsele
chmod +x [nombre_binario].bin
- Instalar el archivo
./[nombre binario].bin

— Ejemplo

```
root@masterhack-pc:~/Ejemplo# sudo ./genymotion-2.3.1_x86.bin
Installing to folder [/home/masterhack/Ejemplo/genymotion]. Are you sure [y/n] ?
y

- Trying to find VirtualBox toolset ..... OK (Valid version of VirtualB
ox found: 4.3.10_Ubuntur93012)
- Extracting files ..... OK (Extract into: [/home/masterhack/Ejemplo/genym
otion])

Installation done successfully.

You can now use these tools from [/home/masterhack/Ejemplo/genymotion]:
- genymotion
- genymotion-shell

root@masterhack-pc:~/Ejemplo#
```

Instalar archivos binarios con **dpkg**

(Ejecutar archivos ejecutables .rpm, .deb)

- Se encuentran empaquetados en varios formatos, según la distribución: ***.rpm**, ***.deb** (Ubuntu).
- Usar **dpkg** para desempaquetar, configurar e instalarlo.

dpkg -i /ruta/nombre_paquete (lo instala)

dpkg -r /ruta/nombre_paquete (lo desinstala)

– Ver los paquetes que tenemos instalados

dpkg -l

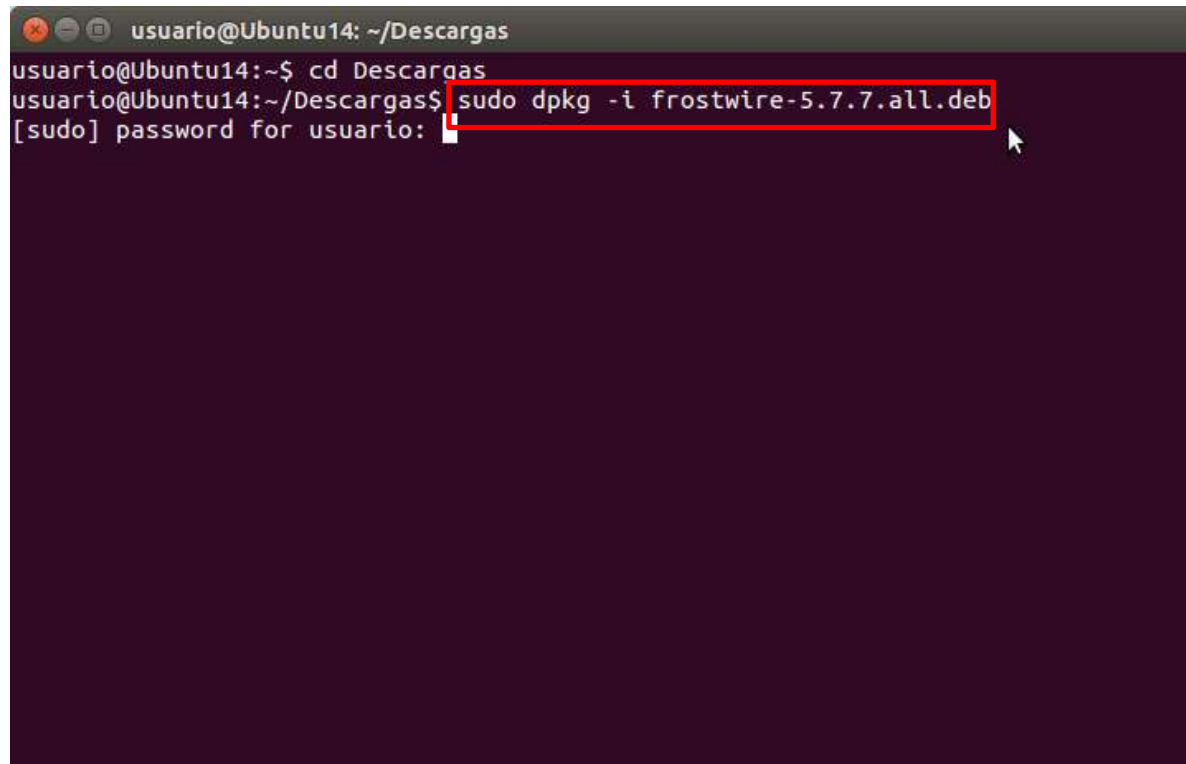
/instalados se indican con dos ii delante del paquete)

Ejemplo:

```
gabriela@Riften:~$ dpkg -l | grep 'chrome'
rc google-chrome-stable                 32.0.1700.102-1
                                amd64      The web browser from Google
ii xserver-xorg-video-openchrome        1:0.3.1-0ubuntu2.1
                                amd64      X.Org X server -- VIA display driver
gabriela@Riften:~$
```

Formato .deb

EJEMPLO instalación



```
usuario@Ubuntu14: ~/Descargas
usuario@Ubuntu14:~$ cd Descargas
usuario@Ubuntu14:~/Descargas$ sudo dpkg -i frostwire-5.7.7.all.deb
[sudo] password for usuario:
```

The image shows a terminal window with a dark purple background. The title bar at the top reads 'usuario@Ubuntu14: ~/Descargas'. The terminal content shows the user navigating to the 'Descargas' directory and running the command 'sudo dpkg -i frostwire-5.7.7.all.deb'. The command is highlighted with a red rectangular box. Below the command, the prompt '[sudo] password for usuario:' is visible, followed by a white cursor character.

Formato .rpm

- Actualizar sistema
apt-get update
- Instalar alien
apt-get install alien
- Convertir .rpm a .deb
alien --to-deb paquete.rpm

Otros archivos ejecutables

¿Cómo ejecutarlos?

Extensión **.sh**

- Son **scripts** generados con **comandos linux** que se ejecutan uno tras otro.
- Situarte en la carpeta donde esta el script o incluirlo en la variable del sistema PATH
- Ejecutar el script
./nombrepaquete.sh o **# sh nombrepaquete.sh**

- Ejemplo

```
/media/nisheet/WORKS sudo sh ./abc.sh
```

```
e@kate-pc:~$ chmod +x ListDir.sh
e@kate-pc:~$ ./ListDir.sh
come
ktop    Downloads    ListDir.sh  Pictures  Templates
uments  examples.desktop Music        Public    Videos
s completes the list of directories
```

Otros archivos ejecutables

¿Cómo ejecutarlos?

Archivos con extensión **.run**

- Son asistentes
- Situarlos en la carpeta
- Ejecutar:

```
# sh ./(archivo).run
```

FICHEROS FUENTES

Instalar ficheros fuentes **.deb** (1/3)

- **Pasos** son:
 - Compilar para generar el archivo binario
 - Instalar
- Antes de comenzar:
 - Instalar instalar un metapaquete llamado **build-essential**
 - # **aptitude update**
 - # **aptitude install build-essential**
 - instalar las librerías de desarrollo del kernel
 - **aptitude install linux-headers '-uname -r'** “son necesarias para los drivers”
 - Y muy importante, leer antes de proceder los archivos como **readme**, **leeme** que suelen tener extensión **txt** y me dicen como instalarlo.

Instalar ficheros fuentes **.deb** (2/3)

- Ejemplo de proceso: (depende de la distribución y versión)
 - Descargar el fichero fuente.
 - Descomprimir el fichero que seguro estará comprimido.
 - Situarte en la carpeta
 - Ejecutar script configure (Comprueba las características del sistema referentes a la compilación)
\configure
 - Compila las fuentes y crea el fichero binario
make
 - Instala la aplicación en el sistema.
make install o **checkinstall** (previa instalación apt-get, no viene por defecto)

Instalar ficheros fuentes **.deb** (3/3)

- Ejemplo:
 - Instalar fuente del paquete libpanelapplet
 - Es una aplicación de mensajería instantánea
 - `sudo aptitude install libpanelapplet`
 - `tar xvfz sensors-applet-0.5.1.tar.gz`
 - `cd sensors-applet-0.5.1`
 - `./configure --prefix=/usr`
 - `Make`
 - `make install`
 - o mejor
 - `sudo checkinstall` (porque facilita la desinstalación)

Instalación de programas sin conexión a internet

- En un ordenador con conexión internet.
 - Descargar el paquete **dpkg-dev** (que por dependencias puede necesitar otros paquetes).
 - Descargar los paquetes o programas necesarios (los que queramos transportar al ordenador sin conexión).
 - Copiar toda la carpeta **/var/cache/apt/archives/**, o sólo los paquetes necesarios, en nuestro medio de transporte (CD, USB, etc.), sin olvidar **dpkg-dev** y sus dependencias.

**¿CÓMO INSTALAR SI TENGO
PROBLEMAS CON LA CONEXIÓN DE
INTERNET?**

Instalación de programas sin conexión a internet

- En un ordenador sin conexión internet (1/2).
 - Crear una carpeta para el repositorio local, por ejemplo: `/home/usuario/repositorio/`
 - Crear una subcarpeta para los paquetes, por ejemplo: `/home/usuario/repositorio/paquetes/`
 - Copiar los paquetes `*.deb`, desde el medio de transporte a la carpeta `/home/usuario/repositorio/paquetes/`,
 - Instalar el paquete `dpkg-dev`
 - # `cd /home/usuario/repositorio/paquetes`
 - # `dpkg -i dpkg-dev`

Instalación de programas sin conexión a internet

- En un ordenador sin conexión internet (2/2).
 - Creamos el archivo (Packages) con las cabeceras de los paquetes:
`cd /home/usuario/repositorio`
`dpkg-scanpackages paquetes /dev/null > Packages`
 - Creamos el archivo comprimido de las cabeceras (comprimimos el packages):
`gzip -c Packages > Packages.gz`
 - Modificamos el `/etc/apt/sources.list` tecleando:
`gedit /etc/apt/sources.list`
 - Insertamos al principio del archivo **deb**
file:/home/usuario/repositorio/ ./
 - También sería conveniente desactivar los repositorios de Internet comentando (un '#' delante) las líneas que haya descomentadas.
 - Ahora ya podemos actualizar la lista de paquetes (**apt-get update**) e instalar los paquetes que necesitemos:
`apt-get update`
`apt-get install nombre_programa`