



Nombre		Curso	
Apellidos		Fecha	

Practica 1-5.- Intérprete de comandos o Shell

shell

El shell o consola es el sistema de comandos de Linux.

Para abrilo usamos

- En la barra de tareas encontrará el icono de un monitor con una *shell*.
- O Pulsando Ctrl+F2 ... Ctrl+F6 ... para acceder a los diferentes terminales por defecto.
- Dependiendo de la distribución puede cambiar en la actual de ubuntu se tiene que hacer en el menu contextual

Si hace clic con el ratón sobre este icono se abrirá la ventana de la *consola* en la que puede introducir comandos. La ventana que aparece es un intérprete de comandos que generalmente se tratará de un Bash (Bourne again shell). El Bash fue desarrollado en el marco del proyecto GNU y es, sin duda, el derivado más extendido de la Bourne Shell (sh). Cuando abra el Bash, verá sobre la primera línea el llamado *Prompt* (indicador de mandatos) que normalmente consta del nombre del usuario y del de la computadora, así como de la localización actual en el árbol de directorios (Path), pero que también puede ser configurado de forma individual. Cuando el cursor se encuentre detrás de este prompt, podrá introducir comandos directamente.

\$ >

El intérprete de comandos o shell del sistema es la interfaz entre el usuario y el sistema operativo. La función del shell es recibir las órdenes del usuario a través de la línea de comandos, interpretarlas, ejecutarlas y mostrar su resultado. Resulta muy útil aprender a utilizar el shell del sistema ya que aunque al principio puede parecer un poco difícil, resulta fundamental para obtener el máximo rendimiento del sistema. El shell permite interactuar directamente con el sistema y con sus ficheros de configuración.

Para iniciar el intérprete de comandos tienes que ir al menú Aplicaciones / Accesorios, ejecutar la herramienta Terminal y aparecerá el terminal del sistema.

Una vez que accedes al sistema se muestra un prompt con el siguiente aspecto o parecido:

:usuario@ubuntu-virtual-machine~\$

donde usuario es el nombre del usuario que está utilizando el sistema, @ubuntu-virtual-machine indica el nombre del equipo. A continuación, se muestra el directorio en el que se encuentra. En el caso de que se encuentre el carácter ~ es porque está en el directorio home. Por último, el símbolo \$ o # indica si es un usuario normal (\$) o es el administrador del sistema (#).

El usuario root es el administrador del sistema y puede realizar cualquier tarea de administración. En algunas distribuciones puede acceder directamente al sistema como usuario root, pero otras distribuciones, como Ubuntu, te obligan a acceder al sistema con un usuario sin privilegios de administrador y luego cambiar de usuario.



Si deseas ejecutar una tarea de forma puntual como root puedes utilizar el comando sudo de la siguiente forma:

```
$ sudo <comando>
```

Si necesita ejecutar múltiples tareas puedes obtener un shell de root ejecutando sudo bash o su:

```
$ sudo bash
```

```
#
```

Además, si lo deseas, puedes activar la cuenta de root al establecer su contraseña:

```
$ sudo passwd root
```

Comandos

Los comandos se componen de distintos elementos. En primer lugar se debe introducir siempre la palabra de **comando** y seguidamente los **parámetros** u **opciones**. Los comandos se ejecutan sólo cuando presiona **Return**. Hasta ese momento podrá editar la línea de comandos sin ningún problema, añadir cualquier opción o corregir fallos. Uno de los comandos más utilizados es **ls**, que se puede usar sólo o con los llamados argumentos. Si sólo introduce **ls** en la consola, se le mostrará el contenido del directorio en el que se encuentra en ese momento. Las opciones van precedidas de un guión. Si introduce el comando **ls -l**, podrá ver el contenido del mismo directorio, sólo que de forma detallada. Verá, además del nombre del archivo, la fecha en que se creó, el tamaño del archivo en bytes y otras características que explicaremos más adelante. Una de las opciones más importantes que también está disponible para muchos otros comandos es **--help**. Si por ejemplo introduce **ls --help**, se visualizarán todas las opciones para el comando **ls**.

Con el comando **ls** puede ver también el contenido de otros directorios que no sean el actual. Introduzca, por ejemplo, **ls -l Desktop** y seguidamente verá el contenido del subdirectorio Desktop. **ls** es el comando en sí, **-l** es una opción para este comando y **Desktop** es un parámetro que se le pasa al comando **l**.

Aunque a lo largo del curso aprenderá a utilizar el shell del sistema, a continuación puedes ver los comandos más utilizados en los sistemas GNU/Linux.

A continuación mostramos información sobre los comandos más utilizados en GNU/Linux.

Manejo de archivos

- **cd** cambia el directorio
- **compress** comprime archivos en formato .Z
- **cp** copia un archivo
- **chmod** cambia los permisos de un archivo o directorio
- **chown** cambia el propietario del archivo o directorio
- **df** muestra el espacio libre en disco
- **du** muestra el espacio en disco utilizado
- **fdformat** formatea un disquete
- **fdisk** particiona unidades
- **file** determina el tipo de archivo a través del análisis parcial de su contenido
- **find** encuentra un archivo
- **fsck** chequea el sistema de archivos



- gzip descomprime un archivo en formato GZip
- ln crea un enlace simbólico o físico depende la opción
- ls sirve para listar el contenido de un directorio
- mkdir crea un directorio
- mkfs crea un nuevo sistema de archivos
- mkswap crea un espacio de intercambio
- more exhibe el contenido de un archivo
- mount monta una unidad o partición en el sistema de archivos
- mv mueve un archivo y se utiliza para renombrar
- pwd devuelve la cadena correspondiente al directorio actual
- rm borra un archivo
- rmdir borra un directorio
- swapon activa el espacio de intercambio
- swapoff desactiva el espacio de intercambio
- tar empaqueta o desempaqueta un archivo en formato .tar
- type muestra la ubicación de un archivo señalando su "path"
- umount desmonta una unidad montada

Manejo de procesos

- free muestra la memoria libre y utilizada
- halt apaga la maquina
- kill mata un proceso señalando el numero de este
- ldd muestra las librerías necesarias para ejecutar un proceso
- ps muestra todos los procesos q se están ejecutando en el sistema
- pstree lo mismo que ps pero en forma de árbol
- reboot reinicia el sistema
- shutdown cierra el sistema
- top monitorea procesos y el estado del sistema
- uname muestra información del sistema

Manejo de usuarios

- adduser crea una cuenta de usuario
- chsh cambia la shell de un usuario
- groups muestra el listado de grupos de usuarios del sistema id muestra la información de usuario y grupo de un determinado usuario
- logout para salir del sistema y permitir el ingreso a otro usuario
- passwd cambia el password de un determinado usuario
- su le da privilegios de root a un usuario
- talk permite chatear con otros usuarios
- users lista los usuarios conectados al sistema
- who muestra información de los usuarios conectados al sistema
- whoami muestra información nuestra

Otros

- cal muestra el calendario
- date muestra el día y la hora
- info muestra la ayuda de un comando
- man muestra las paginas del manual de un comando



- startx para iniciar XWindow
- Comandos para red
- ping el tan famoso ping que manda paquetes esperando una respuesta
- ifconfig para ver las placas de red
- netstat testeo de red
- host "destino" muestra ip de "destino"
- nmap "ip de destino" pequeño programa para analizar ip's o rangos de ellas Indica el shell en el que tienes los permisos de root.

Tipos de ruta

Cuando se trabaja con comandos es habitual tener que pasar como parámetros archivos o directorios. Para indicar un archivo o directorio se utiliza la ruta o path, que puede ser absoluta o relativa. Es importante que tengamos en claro la diferencia entre indicar una ruta relativa y una absoluta cuando realizamos tareas sobre los directorios desde la consola de comandos.

Rutas relativas

Las rutas relativas indican el camino para encontrar un elemento, *pero basándonos en el directorio desde el que se ejecuta la orden*, osea, desde el directorio en donde nos encontramos posicionados. Para saber si son correctas o no tenemos que saber siempre desde dónde se han utilizado.

Para acuñar mejor el concepto, veamos algún ejemplo.

Si me encuentro en el directorio /home y quisiera moverme al directorio /home/usuario/carpeta1 debería indicar el camino hasta donde quiero ir partiendo desde donde me encuentro. Veamos

`cd usuario/carpeta1`

En la practica: Utilizamos el comando `pwd` para ver el directorio en el que nos encontramos e ingresamos la ruta relativa hacia el directorio /home/cristian/carpeta1.

`cd cristian/carpeta1`

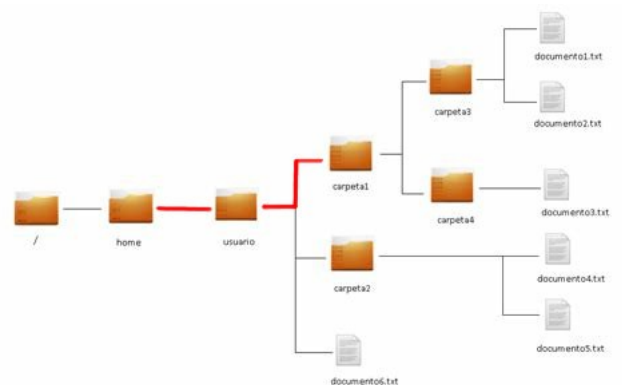
```
[cristian@ARCH home]$ pwd
/home
[cristian@ARCH home]$ cd cristian/carpeta1
```

Una vez confirmemos, nos habremos movido al directorio solicitado. Para corroborar basta con ejecutar el comando `pwd`.

```
[cristian@ARCH carpeta1]$ pwd
/home/cristian/carpeta1
[cristian@ARCH carpeta1]$
```

Este es un ejemplo de ruta relativa.

También es posible movernos hacia taras haciendo uso de rutas relativas, pero para ello debemos tener en claro el concepto de directorio padre.



Directorio padre

Llamamos directorio padre, al directorio que contiene a aquel en el que nos encontramos. Es fundamental para la construcción de rutas relativas saber que `..` indica al directorio padre. Por lo tanto si ejecuto `cd ..` estoy dando la orden de cambiar de directorio al padre del directorio actual, es decir, al que está justo antes en la estructura jerárquica.

Suponiendo que nos encontremos en el directorio `/home/cristian/carpeta1`, y listamos los directorios contenidos, incluyendo el contenido oculto, con el comando `ls -la` obtendremos la siguiente salida.

```
[cristian@ARCH carpeta1]$ ls -la
total 8
drwxr-xr-x  2 cristian users 4096 jun  5 15:59 .
drwx----- 35 cristian users 4096 jun  5 16:27 ..
[cristian@ARCH carpeta1]$
```

Veremos que nos encontramos con los directorios ocultos `.` y `..`. El primero hace referencia al directorio donde nos encontramos y el segundo respectivamente hace referencia al **directorio padre**.

Para movernos hacia atrás haciendo uso de rutas relativas podemos indicarle al comando `cd` que se mueva de a su directorio padre (un nivel dentro de la jerarquía hacia atrás) o de varios a la vez dependiendo de hacia donde queramos movernos.

En la practica, y sabiendo que nos encontramos en el directorio `/home/cristian/carpeta1` siendo este contenido por el directorio por su directorio padre `/cristian` (directorio del usuario), podremos movernos hacia el con la orden

```
cd ..
[cristian@ARCH carpeta1]$ ls -la
total 8
drwxr-xr-x  2 cristian users 4096 jun  5 15:59 .
drwx----- 35 cristian users 4096 jun  5 16:56 ..
[cristian@ARCH carpeta1]$ cd ..
[cristian@ARCH ~]$
```

Si verificamos la posición en la que nos encontramos, veremos que nos habremos movido a `/home/cristian`

```
[cristian@ARCH ~]$ pwd
/home/cristian
[cristian@ARCH ~]$
```

¿Pero, cuantos directorios padres podemos movernos para atrás de forma simultanea? Muy simple, la cantidad de niveles padre con los que contemos hacia atrás tomando como referencia donde es que nos encontramos.

Vemos como seria esto en la practica. Acudimos al mismo ejemplo y nos posicionamos en el directorio `/home/cristian/carpeta1` y vamos a suponer que nos queremos mover haciendo uso de rutas relativas hacia el directorio Raiz (`/`).

Identificamos nuestra posición con `pwd`.

```
[cristian@ARCH carpeta1]$ pwd
/home/cristian/carpeta1
[cristian@ARCH carpeta1]$
```

Donde observamos que hacia atrás tenemos 3 niveles, `/cristian`, `/home` y `/`. entonces nos movemos 3 directorios padres hacia atrás

`cd ../../..` y listamos

Linux

```
[cristian@ARCH carpeta1]$ cd ../../..
[cristian@ARCH /]$ ls -l
total 56
lrwxrwxrwx 1 root root 7 may 23 11:18 bin -> usr/bin
drwxr-xr-x 4 root root 4096 jun 5 13:59 boot
drwxr-xr-x 20 root root 3300 jun 5 00:41 dev
drwxr-xr-x 91 root root 4096 jun 5 13:59 etc
drwxr-xr-x 4 root root 4096 sep 29 2018 home
srw----- 1 root root 0 nov 1 2018 kdeinit5_0
lrwxrwxrwx 1 root root 7 may 23 11:18 lib -> usr/lib
lrwxrwxrwx 1 root root 7 may 23 11:18 lib64 -> usr/lib
drwx----- 309 root root 16384 sep 29 2018 lost+found
drwxr-xr-x 2 root root 4096 ago 21 2018 mnt
drwxr-xr-x 7 root root 4096 jun 5 13:50 opt
dr-xr-xr-x 176 root root 0 jun 5 00:41 proc
drwxr-xr-x 14 root root 4096 may 6 19:17 root
drwxr-xr-x 23 root root 600 jun 5 00:43 run
lrwxrwxrwx 1 root root 7 may 23 11:18/sbin -> usr/bin
drwxr-xr-x 4 root root 4096 sep 29 2018 srv
dr-xr-xr-x 13 root root 0 jun 5 00:41 sys
drwxrwxrwt 16 root root 420 jun 5 17:53 tmp
drwxr-xr-x 9 root root 4096 jun 5 14:00 usr
drwxr-xr-x 12 root root 4096 jun 5 00:41 var
-rw-r--r-- 1 root root 17 sep 29 2018 vconsole.conf
[cristian@ARCH /]$
```

Veremos que haciendo uso de rutas relativas nos movimos al directorio Raíz (/).

Una observación para hacer es que el *único directorio que no tiene padre es la propia raíz del sistema (/)*, ya que este es el directorio desde donde nacen todos los directorios.

Rutas absolutas

El sistema de ficheros es una estructura jerárquica que en el caso de Linux tiene una raíz que se indica cuando se pone solamente el carácter barra /. La raíz contiene los directorios principales del sistema que a su vez tendrán subdirectorios en su interior, esta estructura de directorios se corresponde con el estándar de jerarquías de directorios FHS. Cuando yo quiero indicar dónde se encuentra un elemento usando una ruta absoluta, tendré que indicarle todos los directorios por los que hay que pasar *empezando desde la raíz del sistema*, SIEMPRE.

Si lo llevamos a la practica haciendo uso del ejemplo anterior

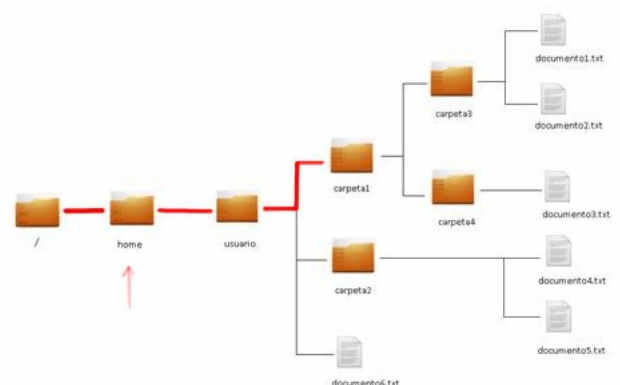
Vamos a suponer que nos encontramos nuevamente en el directorio /home y que nuevamente nos queremos mover hacia el directorio /home/cristian/carpeta1 pero en este caso haciendo uso de una ruta absoluta. La forma correcta de utilizar una ruta absoluta es indicarle el camino desde el directorio raíz (/) hasta donde se encuentra el directorio a donde nos queremos trasladar, independientemente de donde nos encontremos.

`cd /home/usuario/carpeta1`

En la practica seria

Igual que en el ejemplo anterior, hacemos uso del comando `pwd` para ver el directorio en el que nos encontramos e ingresamos la ruta absoluta hacia el directorio /carpeta1.

`cd /home/cristian/carpeta1`



```
[cristian@ARCH home]$ pwd
/home
[cristian@ARCH home]$ cd /home/cristian/carpeta1
```



Aceptamos, y nos habremos movido al directorio indicado utilizando una ruta absoluta. Para corroborar la posición basta con ejecutar el comando `pwd`.

```
[cristian@ARCH carpetal]$ pwd
/home/cristian/carpetal
[cristian@ARCH carpetal]$
```

Estas rutas suelen ser bastante largas, pero como ventaja tienen que funcionan siempre, independientemente del lugar desde el que ejecute la orden. Es muy recomendable utilizar la facilidad que brinda la shell de completar el nombre de un elemento del sistema de ficheros pulsando la tecla tabulador. Ahorrará mucho tiempo y posibles errores de tipeo (teniendo en cuenta que se discriminan mayúsculas y minúsculas en los nombres de los directorios), a esta acción la conocemos como autocompletado.

Comandos que te haran falta hoy

mkdir crea un directorio con las rutas adecuadas.

Touch crea archivos vacios

cp copia archivos

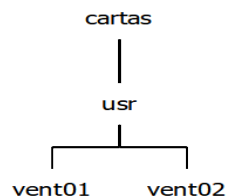
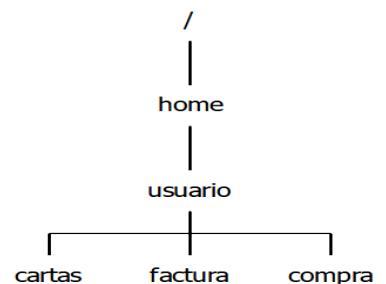
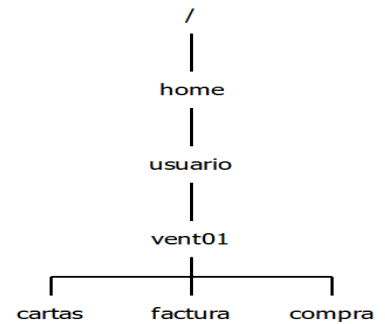
mv mueve y renombra archivos

Practica 1-5 Linux Shell 1

1. Que es el shell?
2. Como lo abrimos ?
3. Cual es el shell por defecto?
4. Hay otros shells? Enumera algunos
5. Que es *Prompt*?
6. Que es un *Path*?
7. Que suele salir en el *Prompt*?
8. que significa el carácter `~` ?
9. que significa el carácter `$` ?
10. que significa el carácter `#`?
11. que es el usuario root ey que puede realizar?
12. Como podemos acceder al usuario root?
13. Que hace `$ sudo bash`?
14. Como podemos activar al usuario root con su contraseña?
15. De que tres elementos se componen los comandos? Y como se distinguen entre ellos?



16. Cuando se ejecutan los comandos?
17. Que hace ls?
18. Que hace ls -l?
19. Que hace Mkdir?
20. Usa mkdir , md, cp y touch para hacer esta estructura en tu linux.
21. Que hace pwd?
22. Que es una ruta? Y que tipos hay?
23. Que caracteriza una ruta absoluta?
24. Que caracteriza una ruta relativa?
25. Que es el directorio padre y como accedemos a el?
26. - La empresa Novedades Cinematográficas S.L. se dedica a la venta y distribución de películas de video. La organización de la información se encuentra almacenada según el siguiente esquema, que es idéntico para los usuarios vent01 y vent02.
27. - Crear un fichero en cada una de las carpetas con el nombre cartas-v01.txt, factura-v01.txt, compras-v01.txt.
28. - Copia estos ficheros en las carpetas respectivas de vent02.
29. - Cambia los nombres a los ficheros que acabas de copiar para que se llamen igual pero con la terminación -v02.
30. -El departamento de informática ha decidido que este tipo de organización crea bastantes problemas cuando se trata de compartir información y ha decidido estructurarlo así:
31. 6.- Y dentro de cada uno de los directorios cartas, factura y compra habrá un subdirectorío para cada uno de los usuarios:



32. - Mueve los ficheros a las carpetas correspondientes
33. - Muestra un listado completo de todos los directorios de tu directorio de trabajo.