

## CONSOLA Y CONSOLAS VIRTUALES

Una de las características propias de Linux es que es un S.O. multiusuario. Esto permite que varios usuarios puedan estar usando simultáneamente el sistema a través de lo que se denomina consola virtual o **VCs**. Como ya se dijo, el sistema que sirve a los demás usuarios se llama host.

Una consola se puede abrir para el usuario root o para cualquier otro usuario. Para hacer esto, lo único que se debe hacer es presionar **Alt+F2**, y aparecerá:

### **login:**

Para ingresar, se debe digitar el login o nombre de usuario y posteriormente el password o contraseña.

El límite de consolas que se pueden utilizar lo pone, en un principio, la cantidad de teclas de función (F1-F12). Entonces para cambiar entre consolas solo tienes que combinar las teclas **Alt+Fx** donde *x* es un número 1-12.

Puede ser útil hacer uso de varias consolas al mismo tiempo, a pesar de ser una sola persona la que utiliza la máquina. Se puede correr una aplicación en una y la misma en otra, o directamente correr dos distintas en diferentes consolas.

## EL SHELL

El SHELL es el encargado de establecer una comunicación entre el núcleo (kernel) y el usuario del Sistema Linux, o sea, que es una interface con nuestro Sistema Operativo. Entonces gracias a él podremos dar las ordenes necesarias para que nuestro sistema informático realice las tareas que necesitamos.

Como hemos mencionado anteriormente en numerosas ocasiones, UNIX es un sistema operativo multitarea y multiusuario. La multitarea es muy útil, y una vez la haya probado, la usará continuamente. Muchas de las características que se tratarán en esta sección son proporcionadas por el intérprete de comandos. Hay que tener cuidado en no confundir UNIX (el sistema operativo) con el intérprete de comandos. El intérprete de comandos proporciona la funcionalidad sobre el UNIX.

El intérprete de comandos no es sólo un intérprete interactivo de los comandos que tecleamos, es también un potente lenguaje de programación, el cual permite escribir guiones, que permiten juntar varias ordenes en un archivo. El uso de los guiones (scripts) del intérprete de comandos es una herramienta muy potente que le permitirá automatizar e incrementar el uso de UNIX.

## TIPOS DE INTÉRPRETES DE COMANDOS

Hay varios tipos de intérpretes de comandos en el mundo UNIX. Los dos más importantes son el **"Bourne shell"** y el **"C shell"**.

El intérprete de comandos Bourne, usa una sintaxis de comandos como la usada en los primeros sistemas UNIX, como el System III. El nombre del intérprete Bourne en la

mayoría de los UNIX es /bin/sh (donde sh viene de "shell", interprete de comandos en inglés).

El intérprete C usa una sintaxis diferente, a veces parecida a la del lenguaje de programación C, y en la mayoría de los sistemas UNIX se encuentra como /bin/csh.

Bajo Linux hay algunas diferencias en los intérpretes de comandos disponibles. Dos de los más usados son el "**Bourne Again Shell**" o "**Bash**" (/bin/bash) y **Tcsh** (/bin/tcsh). Bash es un equivalente al Bourne con muchas características avanzadas de la C shell. Como Bash es un super-conjunto de la sintaxis del Bourne, cualquier guión escrito para el interprete de comandos Bourne standard funcionará en Bash. Para los que prefieren el uso del interprete de comandos C, Linux tiene el Tcsh, que es una version extendida del C original.

El tipo de intérprete de comandos que decida usar es puramente una cuestion de gustos. Algunas personas prefieren la sintaxis del Bourne con las características avanzadas que proporciona Bash, y otros prefieren el más estructurado intérprete de comandos C. En lo que respecta a los comandos usuales como cp, ls..etc, es indiferente el tipo de intérprete de comandos usado, la sintaxis es la misma. Solo, cuando se escriben scripts para el intérprete de comandos, o se usan características avanzadas aparecen las diferencias entre los diferentes intérpretes de comandos.

## **CARACTERÍSTICAS DEL SHELL**

### **Flujos**

El shell le otorga a cada programa en ejecución 3 flujos: Entrada Estándar (teclado, archivo, dispositivo E/S), Salida Estándar (terminal, archivo, dispositivo E/S) y Error Estándar (terminal, archivo, dispositivo E/S).

### **Ejecución del Shell**

Una vez que usted se ha registrado en la consola virtual, el sistema carga el intérprete de comandos en memoria. Éste lee primero el archivo /etc/profile que contiene la configuración común del entorno para todo el sistema y sus usuarios, luego lee el archivo .bash\_profile ubicado en el directorio home del usuario recién ingresado.

### **¿Cómo se escriben los comandos?**

La sintaxis para escribir un comando en el prompt de bash es:

**comando [parámetro1] [ parametro2] ...**

### **Capitalización de las Letras**

Recordemos que siempre los comandos y parámetros en GNU/Linux son "**case sensitive**", que quiere decir que "algo" no es igual a "ALGO" u "AlGo".

### **Historial de Comandos**

El archivo .bash\_history es un archivo de historial con todos los comandos utilizados por el usuario y como es lógico se encuentra en el directorio home de dicho usuario. Éste archivo aloja una cantidad de comandos definida previamente. Se accede al historial por medio de las teclas **FLECHA ARRIBA** y **FLECHA ABAJO**.

Si necesita escribir una cantidad excesivamente grande de parámetros y ha llegado al final de la línea de comandos, puede hacer uso del símbolo "\" seguido de un "enter" para poder continuar con la escritura de dichos parámetros en la línea siguiente y también sirve para ver que escribimos anteriormente.

### **Completado de Línea**

Se puede hacer que el shell complete la línea de comandos cuando se introduzcan las primeras letras y se presione la tecla TAB. También se puede utilizar esta propiedad para expandir la ruta a un directorio determinado.

*Ejemplo:* `cd /ho` (presionamos TAB) y el shell nos devolverá `/home/`

### **Comillas**

Las comillas controlan la forma en que el shell expandirá las órdenes que estén encerradas entre ellas.

Existen tres tipos de comillas, las dobles ("), las sencillas (') y las inversas (`).

- Las comillas inversas indican al shell que tendrá que reemplazar lo que está encerrado entre ellas con su resultado.
- Las comillas sencillas (') le dicen al sistema que no hagan ninguna expansión.
- Las comillas dobles tienen casi la misma funcionalidad que las simples pero con la salvedad de que lo que se incluya dentro de estas pasará a ser como una cadena simple de caracteres a excepción de las comillas inversas (`), el signo dólar (\$), la diagonal (\) y las mismas comillas dobles (").