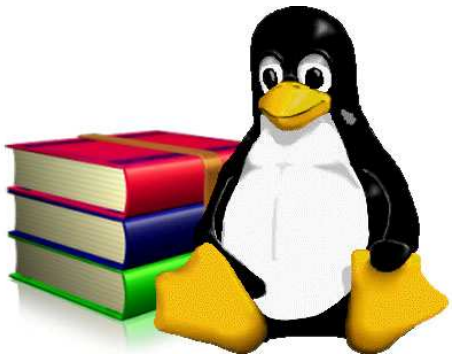
Manual Comandos Linux



Iniciamos otro mini manual sobre comandos Linux , espero que os guste.

Comandos básicos Linux

Para conectarnos a un servidor Linux necesitamos un login y contraseña y mediante telnet conectamos al servidor.

Login , Contraseña

Si queremos salir tecleamos : **shutdown** (administrador) **exit**

Who nos permite ver los usuarios conectados

Who I am no dice el usuario que somos.

passwd si deseamos cambiar la contraseña.

date para ver la fecha del servidor

cal para mostrar el calendario.

Logname para ver nuestro nombre de usuario

Uname pasa saber el linux que utilizamos

tty nos muestra el terminal en el que estamos.

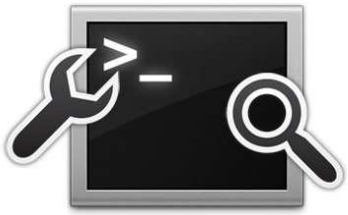
echo para visualizar texto en pantalla.

clean para limpiar la pantalla.

La parte trabajadora del sistema es el nucleo que se llama KERNEL una manera de utilizarlo mediante el shell y por encima de este el entorno gráfico.

Algunos shell podrían ser.

- Sh
- bash (Linux)
- Ksh
- Csh



Cualquier orden en Unix/Linux tiene la siguiente sintaxis.

orden -opciones parámetros (minúsculas) Ej. **ls -li**

Hay tres maneras de agrupar órdenes.

La primera y básica.

orden1; orden2; orden3 > archivo 1

Con el paréntesis creamos otro shell independiente si redireccionamos el paréntesis afecta a todo , en cambio en el primer caso solo afecta al caso 3.

(orden1; orden2; orden3) > archivo 1

Las llaves , el interior se ejecuta en el mismo shell y el redireccionamiento afecta a las tres ordenes.

{ orden1; orden2; orden3 } > archivo 1

Se ejecuta la orden 2 solo su la orden 1 a tenido exito.

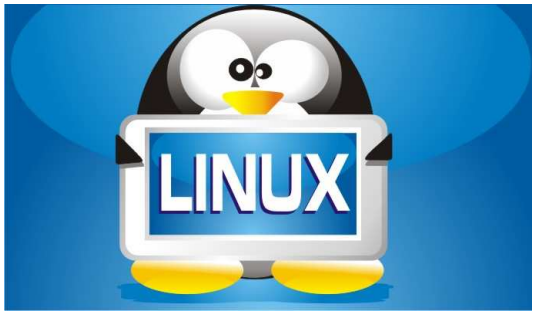
orden1 && orden2 > archivo1

Se ejecuta la orden 1 y solo si falla , entonces se ejecuta la 2.

orden1 || orden2 > archivo1

Encauza la salida de la orden1 en la entrada de la orden2.

orden1 | orden2 > archivo1



Ponemos **$** a todo lo que va detrás de una variable.

$Logname

” ***** **?** “ Són caracteres especiales

[1-9] especificamos el rango de carácteres que puede ocpar esa posición.

\ **$Logname** elimina el significado , la función de **$**.

Por ejemplo:

echo \ **$Logname** > Mostraría ” **$Logname**“.

‘ ‘ todo lo que pongamos dentro de comillas simples pierde su función.

“ “ conseguiremos lo mismo menos estos tres caracteres **$** “ \

Si ponemos echo ‘**ls -li**‘ aparecerá en pantalla el comando ejecutado dentro de las comillas.

echo ‘ **date+ D** ‘ nos mostrará la fecha entre comillas.

COMANDOS REFERENTES A LOS DIRECTORIOS/ARCHIVOS

El comando **ls** . Quizás uno de los comandos más utilizados, sirve para listar archivos y algunos de sus atributos són.

ls -l **lista**

ls -i **inodo**

ls -a **ocultos**

Otros comandos que se utilizan habitualmente.

cd para cambiar de directorio.

pwd para saber en que directorio nos encontramos.

mkdir crear un directorio.

rmdir borrar directorio vacio.

touch crear un archivo vacio.

Si encontramos un archivo con un punto delante , de esta forma **.archivo** , el punto delante lo convierte en archivo oculto y con la orden **ls** no lo veremos , necesitaremos utilizar el comando **ls -a** mencionado anteriormente.

cp para copiar archivos o directorios.

mv mover y renombrar archivos.

ln enlace normal, duro.

ln -s para crear un enlace simbólico.

cat visualizar contenido.

rm borrar archivos

rm -r directorios vacios.

FILTROS

more

Lista ficheros incluidos los ocultos de sistema.

```
ls -la | more
```

Lista los ficheros de un directorio de forma paginada.

```
ls -lh
```

sort ordenar la salida del comando.

grep buscar palabra.

Muestra las últimas líneas de un archivo, 10 por defecto.

```
tail -n 5 archivo
```

Muestra las 5 últimas líneas del archivo.

```
tail -f 5 archivo
```

nl para numerar las lineas.

wc no dice cuantas lineas tiene.

Muestra el contenido de un fichero de forma paginada.

```
zcat fichero
zmore fichero
zless fichero
```

REDIRECCIONAMIENTOS

stdin

entrada estandar para datos,el teclado .

stdout

salida estandar para los programas,la pantalla.

stderr

salida estandar para los mensajes de error,la pantalla .

Redirecciones, un redireccionador redirige la salida de un comando a un fichero

(<) redireccionar la entrada.

Sintaxis : **comando < fichero**

(>) redireccionar la salida.

`Sintaxis` **`: comando > fichero`**

(>>) volcar la ejecución de un comando en otro.

`Sintaxis` **`: comando >> fichero`**

2> o **2>>** se vuelca en /dev/null (papepelara)

Para ver linea 5 de un archivo por jemplo **cat 1hc -5 |tail -1**