

## Opciones de Comandos Básicos

## Opciones de Comandos.

### TR.

Comando utilizado para tratar la información.

`tr [opciones] "patrónA" "patrónB"`

### Opciones destacadas:

- **-s 'patrón':** Sustituye todos los patrones seguidos por uno mismo. Es decir, si como patrón le pasamos un espacio en blanco, recoge todos los espacios en blanco contiguos y los sustituye por uno sólo.

```
salva@UbuntuSalva:~/uni10$ ls -l
total 8
-rw-rw-r-- 1 salva salva 0 oct 23 17:04 Hardware
drwxrwxr-x 2 salva salva 4096 oct 23 17:04 JAVASCRIPT
drwxrwxr-x 2 salva salva 4096 oct 23 17:04 SHELL
```

```
salva@UbuntuSalva:~/uni10$ ls -l | tr -s ' '
total 8
-rw-rw-r-- 1 salva salva 0 oct 23 17:04 Hardware
drwxrwxr-x 2 salva salva 4096 oct 23 17:04 JAVASCRIPT
drwxrwxr-x 2 salva salva 4096 oct 23 17:04 SHELL
```

- **-d 'patrón':** Elimina todas las coincidencias del patrón.

**Elimina todos los números de la salida.**

```
salva@UbuntuSalva:~/uni10$ ls -l | tr -d [0-9]
total
-rw-rw-r--  salva salva  oct  : fic
-rw-rw-r--  salva salva  oct  : Hardware
drwxrwxr-x  salva salva  oct  : JAVASCRIPT
drwxrwxr-x  salva salva  oct  : SHELL
```

- **-t 'patrónA' 'patrónB':** Intercambia todos los patronesA por los patronesB.

**Sustituye los espacios en blanco por ":"**

```
salva@UbuntuSalva:~/uni10$ ls -l | tr -s ' ' | tr -t ' ':''
total:12
-rw-rw-r--:1:salva:salva:64:oct:23:17:25:fic
-rw-rw-r--:1:salva:salva:0:oct:23:17:04:Hardware
drwxrwxr-x:2:salva:salva:4096:oct:23:17:04:JAVASCRIPT
drwxrwxr-x:2:salva:salva:4096:oct:23:17:04:SHELL
```

- Es importante notar que por defecto no funciona sobre ficheros. Para trabajar con ficheros deberemos redireccionar la entrada de datos (<).
  - Ejemplo: `tr -s ' ' < fichero.txt`

```
salva@UbuntuSalva:~/uni10$ cat fic
L4 c0d1f1c4c10n h4ck3r sust1tuy3 c13rt4s v0c4l3s p0r nÚm3r0s
```

```
salva@UbuntuSalva:~/uni10$ tr -t [0134] [oieal] < fic
La codificacion hacker sustituye ciertas vocales por números
```

Es importante saber que podemos hacer uso de palabras reservadas que representan una secuencia de caracteres:

**[::digits:]** → Todos los números.

**[::lower:]** → Caracteres en minúsculas.

**[::upper:]** → Caracteres en mayúsculas.

**[::space:]** → Espacios en blanco.

## **CUT.**

Comando especialmente útil para seleccionar partes de un fichero o de una información devuelta por un comando.

`cut [opciones] "Fichero"`

### **Opciones destacadas:**

- **-d:** Permite indicar un delimitador.
- **-f:** Una vez especificado un delimitador el fichero lo dividimos en varios campos (fields). Con la opción **-f** podemos quedarnos con 1 o varios campos.
  - Es importante saber que con **"-f3"** nos quedaríamos con el campo 3 y con **"-f3-"** nos quedaríamos desde el campo 3 hasta el último campo.
- **-c:** Permite quedarnos con una serie de caracteres.

```
salva@UbuntuSalva:~/uni10$ ls -l
total 8
-rw-rw-r-- 1 salva salva 0 oct 23 17:04 Hardware
drwxrwxr-x 2 salva salva 4096 oct 23 17:04 JAVASCRIPT
drwxrwxr-x 2 salva salva 4096 oct 23 17:04 SHELL
```

**Seleccionamos el campo 3 de la salida original.**

```
salva@UbuntuSalva:~/uni10$ ls -l | tr -s ' ' | cut -d ' ' -f3
salva
salva
salva
```

**Seleccionamos el campo 3 en adelante.**

```
salva@UbuntuSalva:~/uni10$ ls -l | tr -s ' ' | cut -d ' ' -f3-
salva salva 0 oct 23 17:04 Hardware
salva salva 4096 oct 23 17:04 JAVASCRIPT
salva salva 4096 oct 23 17:04 SHELL
```

**Seleccionamos el primer carácter (tipo de archivo).**

```
salva@UbuntuSalva:~/uni10$ ls -l | tr -s ' ' | cut -c1
t
-
d
d
```

Seleccionamos del carácter 2 al 10 (permisos)

```
salva@UbuntuSalva:~/uni10$ ls -l | tr -s ' ' | cut -c2-10
otal 8
rw-rw-r--
rwxrwxr-x
rwxrwxr-x
```

## GREP.

`grep [opciones] "patrón de búsqueda" "Fichero1" "Fichero2" ... "FicheroN".`

### Opciones destacadas:

- **-c:** Número de ocurrencias.
- **-i:** No distingue entre mayúsculas y minúsculas.
- **-v:** Selecciona todas las líneas sin ocurrencias.
- **-w:** Sólo devuelve si la coincidencia es una palabra completa. Una palabra es identificada cuando va rodeada de un separador, por ejemplo un espacio en blanco o “.”.
- **-r:** Busca de forma recursiva dentro de la carpeta.
  - `grep -r "patrón" "Directorio"`
- **-L:** Devuelve los ficheros donde NO hay matches (ocurrencias). Habitualmente usado junto con la opción de recursividad.
  - `grep -r -L "patrón" "Directorio"`
- **-I:** Devuelve los ficheros donde SÍ hay ocurrencias. Habitualmente usado junto con la opción de recursividad.
  - `grep -r -I "patrón" "Directorio"`
- **-q:** Modo silencioso, lo que hace que no devuelva nada por pantalla. Para saber si ha habido ocurrencias, una vez ejecutado el comando, comprobaremos el contenido de \$?, si es 0 es que sí ha habido ocurrencia, en caso contrario no. Es útil cuando hacemos un script y no queremos que se vayan mostrando ejecuciones grep por el terminal.

```
salva@UbuntuSalva:~$ cat lista
salva
pepe
jaime
álvaro
fulgencio
juande
higinio
pedro
pablo
```

Si \$? vale 1 es que NO ha habido coincidencia.

```
salva@UbuntuSalva:~$ grep -q federico lista
salva@UbuntuSalva:~$ echo $?
1
```

Si \$? vale 0 es que SÍ ha habido coincidencia.

```
salva@UbuntuSalva:~$ grep -q pepe lista
salva@UbuntuSalva:~$ echo $?
0
```

- **-e:** Equivalente al comando “egrep”. Permite el uso de **expresiones regulares**.

Busca una línea que comience por “salva:”.

```
salva@UbuntuSalva:~$ grep -E ^salva: /etc/passwd
salva:x:1000:1000:salva,,,:/home/salva:/bin/bash
```

Busca uno u otro patrón.

```
salva@UbuntuSalva:~$ grep -E -w "salva|pedro" /etc/passwd
salva:x:1000:1000:salva,,,:/home/salva:/bin/bash
pedro:x:1001:1001:Pedro,23,2,23,sd:/home/pedro:/bin/bash
```

Selecciona aquellas líneas que terminan con el patrón “bash”.

```
salva@UbuntuSalva:~$ grep -E bash$ /etc/passwd
root:x:0:0:root:/root:/bin/bash
salva:x:1000:1000:salva,,,:/home/salva:/bin/bash
pedro:x:1001:1001:Pedro,23,2,23,sd:/home/pedro:/bin/bash
jaimito:x:1002:1002:Jaimito,23,23,23,23:/home/jaimito:/bin/bash
examencito:x:1005:1000:MI PRUEBA DE EXAMEN:/home/salva:/bin/bash
```

## HEAD.

Muestra las primeras líneas de un fichero. Por defecto muestra las primeras 10 líneas.

```
salva@UbuntuSalva:~$ cat listaNumeros
1
2
3
4
5
6
7
8
9
10
```

- **-n ‘líneas’:** Muestra las primeras X líneas.

```
salva@UbuntuSalva:~$ head -n 3 listaNumeros
1
2
3
```

- **-n -‘líneas’:** Muestra todo menos las últimas X líneas.

```
salva@UbuntuSalva:~$ head -n -3 listaNumeros
1
2
3
4
5
6
7
```

## TAIL.

Muestra las últimas líneas de un fichero. Especialmente útil para hacer seguimiento de las entradas en un fichero o log.

- **-n “líneas”**: Muestra las últimas X líneas.

```
salva@UbuntuSalva:~$ tail -n 3 listaNumeros
8
9
10
salva@UbuntuSalva:~$
```

- **-n +“líneas”**: Imprime todo a partir de la línea X (incluida)

```
salva@UbuntuSalva:~$ tail -n +3 listaNumeros
3
4
5
6
7
8
9
10
salva@UbuntuSalva:~$
```

- **-f “fichero”**: Cada vez que se actualiza el fichero nos avisa, mostrando las nuevas líneas insertadas. Útil para el seguimiento de logs en la depuración de problemas.

## LS.

Lista los recursos de un directorio.

- -l: Muestra información detallada (aprender de memoria la salida).
- -p: Añade una "/" a los directorios. Nos permite diferenciar en un script los ficheros de directorios.

```
salva@UbuntuSalva:~/uni10$ ls -lp
fic Hardware JAVASCRIPT/ SHELL/
```

- -t: Ordena por fecha de modificación.

```
salva@UbuntuSalva:~/uni10$ ls -lt
total 12
-rw-rw-r-- 1 salva salva 64 oct 23 17:25 fic
-rw-rw-r-- 1 salva salva 0 oct 23 17:04 Hardware
drwxrwxr-x 2 salva salva 4096 oct 23 17:04 JAVASCRIPT
drwxrwxr-x 2 salva salva 4096 oct 23 17:04 SHELL
```

- -s: Ordena por tamaño del fichero/directorio.

```
salva@UbuntuSalva:~/uni10$ ls -ls
total 16
4 -rw-rw-r-- 1 salva salva 64 oct 23 17:25 fic
4 -rw-rw-r-- 1 salva salva 2875 oct 23 18:03 Hardware
4 drwxrwxr-x 2 salva salva 4096 oct 23 17:04 JAVASCRIPT
4 drwxrwxr-x 3 salva salva 4096 oct 23 18:03 SHELL
```

- -r: Invierte la salida. Se suele combinar con las opciones -s o -t.

```
salva@UbuntuSalva:~/uni10$ ls -lrt
total 12
drwxrwxr-x 2 salva salva 4096 oct 23 17:04 SHELL
drwxrwxr-x 2 salva salva 4096 oct 23 17:04 JAVASCRIPT
-rw-rw-r-- 1 salva salva 0 oct 23 17:04 Hardware
-rw-rw-r-- 1 salva salva 64 oct 23 17:25 fic
```

## **SORT.**

Permite ordenar alfabéticamente o numéricamente una salida. Por defecto ordena alfabéticamente.

```
sort [opciones] "Fichero1"
```

- **-r**: invierte la salida.
- **-n**: ordena numéricamente.
- **-M**: ordena por meses (ene, feb, mar, ...).
- **-t**: "delimitador": Permite indicar un delimitador para identificar los campos o columnas. Funciona igual que la opción **-d** del comando cut.
- **-k**: "campo": Permite ordenar la salida en base a una columna.
- **-u**: Sólo muestra los campos únicos (no duplicados)

### Ejemplos:

Ordena los ficheros por fecha (meses). Para ello seleccionamos la sexta columna.

```
ubuntu@salva:~$ ls -l | tr -s ' ' | sort -M -t ' ' -k6
total 28
----rw---- 1 ubuntu ubuntu 191 sep 24 08:38 mifichero.txt
drwxrwxrwx 4 ubuntu ubuntu 4096 oct 19 09:24 Mi_Carpeta
drwxrwxr-x 2 ubuntu ubuntu 4096 oct 19 09:24 carpeta
-rw-rw-r-- 1 ubuntu ubuntu 0 oct 19 10:53 micarpeta.txt
-rwxrwxrwx 1 ubuntu ubuntu 151 oct 22 19:41 ejemploCase.sh
-rwxrwxrwx 1 ubuntu ubuntu 244 oct 22 19:25 ejemploFuncion.sh
-rwxrwxrwx 1 ubuntu ubuntu 720 oct 22 20:11 ejemploArgumentos.sh
-rwxrwxrwx 1 ubuntu ubuntu 76 oct 21 17:53 recorreVector.sh
```

Ordena de forma numérica por la columna 2, que representa las notas de un listado de alumnos.

```
ubuntu@salva:~$ cat lista
salva 5
pepe 4
juan 2
maria 10
juana 5
ana 9
sara 10
julian 2

ubuntu@salva:~$ sort -n -t' ' -k2 lista
juan 2
julian 2
pepe 4
juana 5
salva 5
ana 9
maria 10
sara 10
```



En el siguiente ejemplo ordenamos alfabéticamente pero eliminamos de la salida los campos repetidos.

```
salva
pepe
ana
salva
salva
ana
juan
maria
juana
ana
salva
sara
julian
ubuntu@salva:~$ sort -u lista
ana
juan
juana
julian
maria
pepe
salva
sara
```

### UNIQ.

`uniq [opciones] "Fichero"`

Permite eliminar o detectar líneas duplicadas contiguas, por lo que suele venir acompañado del comando sort.

- **-D:** Imprime todas las líneas duplicadas.
- **-i:** No tiene en cuenta diferencia entre mayúsculas y minúsculas.

```
salva
pepe
ana
salva
salva
ana
juan
maria
juana
ana
salva
salva
sara
julian
ubuntu@salva:~$ sort lista | uniq -D
ana
ana
ana
ana
salva
salva
salva
salva
```