

SCRIPTING

```
> ip a | grep 'inet' | cut -d ' ' -f 6 | cut -d '/' -f 1
```

cut extrae columnas indicándole el delimitador. en el ejemplo anterior extrae las columna sexta delimitados (las columnas) por espacios. y luego la primera columna, teniendo en cuenta que los delimitadores son /.

Comando tr (translate): la entrada le llega por una tubería y “traduce”, sustituye por pantalla caracteres. Lo hace carácter por carácter, si quiero cambiar tres, tendré que darle 3 para sustituir

```
> cat text.txt | tr 'acb' '?X-'
```

```
> cat text.txt | tr -s ' ' '*' ← con el parámetro '-s' sustituye uno o más espacios en blanco por un solo asterisco.
```

```
> ip a | grep 'inet' | tr -s ' ' '*' | cut -d '*' -f 3
```

```
egrep '[[a-zA-Z0-9_]]*[[^@].@][a-zA-Z0-9_]*.?[a-zA-Z0-9]{2,6}'  
emails.txt
```

SCRIPTING

La variable “?” guarda las salidas de error que son códigos entre 0-255. 0 es que el comando se ha ejecutado correctamente.

```
> id student && echo “usuario nuevo” ← Ejecución condicionada de comandos, la condición es &&. Si el comando anterior se ejecuta con salida de error “0”, se ejecuta el segundo.
```

```
> id student || echo “usuario nuevo” ← ejecución condicionada de comandos, si el comando primero tiene error (= código de error mayor que cero), se ejecuta el echo.
```

> id student &>/dev/null || echo "usuario nuevo" ← lo mismo que el anterior pero la salida de error lo envía a la "papelera linux".

> test -f file.txt ; echo \$? ← comprueba si es un **fichero regular**
-d file.txt ; echo \$? ← comprueba si existe y es un **directorio**
-L file.txt ; echo \$? ← comprueba si existe y es un **soft link**

OJO!!: Test no da output por sí solo.

>test -d /dir3 || mkdir /dir3

Test también **compara cadena como código ascii**

>test cadena1 == cadena2; echo \$?
>test cadena1 == cadena1; echo \$?
>test cadena1 != cadena2; echo \$?

Para **comprar cadenas numéricas**:

>test 3 -eq 3 ; echo \$? ← -eq = **igual**
>test 3 -ne 3 ; echo \$? ← -ne = **no igual**
-gt = nº estrictamente **mayor que** otro
-lt = nº estrictamente **menor que** otro

OJO!!: En lugar de la palabra test se pueden usar **[]** o **[[]]**

>test A -eq B
>**[** A -eq B

SCRIPTING

Comando sed ← se aplica sobre los ficheros pero no los cambia.
Sustituye cadenas. Este comando permite utilizar expresiones regulares.

>sed 's/cat/dog/g' file ← saca por pantalla el fichero file sustituyendo cat por dog.

>sed 's/c.*/dog/g' file

>sed -i 's/cat/dog/g' file ← la opción -i cambia el fichero

>sed -i.ori 's/cat/dog/g' file ← cambia el fichero y crea una copia con extensión .ori

>sed 's:/var/www/html:/custom:' /etc/httpd/conf/httpd.conf ← Aquí el separador "/" lo sustituimos por "." para no tener que escapar los "/" de las rutas.

para que la shell haga cosas numéricas:

C=\${A+B} ←con dólar corchete interpreta la cadena como numérica.
ó

C=\$((A+B))

#!/bin/bash

[sin lo lo está ejecutando root exit 3 ← usar && y ||
comprobar si el usuario ya existe

IF

if condición

then

acción

fi

NOTA: La condición es un comando, y se ejecuta la acción sólo si la condición no da error.

Ejem: ls && echo hecho bien ← con if se haría así:

```
if ls
then
    echo hecho bien
fi
```

Se ejecuta la acción si la condición da error:

```
if !condición
then
    accion
fi
```

Si la condición no da error, se ejecuta acción1, y si da error, ejecuta acción2:

```
if condición
then
    acción1
else
    acción2
fi
```

Ejemplo:

```
if id user
then
    echo ya existe
else
    useradd user
fi
```

Mirar si ssh está ejecutándose, y si no lo está que lo arranque:

```
if systemctl status sshd
then
    echo Está activo
else
    systemctl start sshd
fi
```

OJO!!: `systemctl status cups; echo $?` ←para saber si da error o no un comando

Que diga si server15 está apagado o no:

```
if ping -c3 server15.example.com &>/dev/null
then
    echo Está UP
else
    echo Está Down
fi
```

Si la hora es antes que las 14:00 que diga que es antes de las 2, y si es después de las 14:00 que diga que es después de las 2:

```
if test $(date +%H) -lt 14 &>/dev/null
then
    echo son antes de las 14
else
    echo son más de las 14
fi
```