

Ordena alfabéticamente líneas de texto y las muestra en la salida estándar
Formato: **sort** [opciones] *fichero*
-b ignora blancos al principio de línea
-f no distingue mayúsculas/minúsculas
-r orden inverso
-n ordena numéricamente
-k POS1[, POS2] ordena según los campos desde POS1 o POS2, o el final si no está POS2 (el primer campo es 1)
-t *char* establece el carácter separador entre campos
-u unique

Permite **unir** texto de varios ficheros, uniendo las líneas de cada uno de los ficheros
Formato: **paste** [opciones] fichero1 [fichero2] . . .
-s pega los ficheros secuencialmente
-d especifica los caracteres delimitadores en la salida

Escribe **partes** seleccionadas de un fichero a la salida estándar; puede usarse para seleccionar columnas o campos de un fichero específico
Formato: **cut** [opciones] fichero
-b, -c, -f corta por bytes, caracteres o campos,
-d fija el carácter delimitador entre campos (por defecto, TAB) respectivamente

Descarta todas (menos una) las líneas idénticas sucesivas en el fichero
Formato: **uniq** [opciones] fichero
-d muestra las líneas duplicadas (sin borrar)
-u muestra solo las líneas sin duplicación
-i ignora mayúsculas/minúsculas al comparar
-c muestra el número de ocurrencias de cada línea
-s n no compara los n primeros caracteres
-f n no compara los n primeros campos
-t c usa el carácter c como separador de campos

Muestra el número de líneas, palabras y bytes de un fichero
Formato: **wc** [opciones] fichero
-l muestra solo el número de líneas
-w muestra solo el número de palabras
-c muestra solo el número de bytes
Ejemplo: wc -w quijote.txt 33 quijote.txt
wc -c quijote.txt 178 quijote.txt

Borra caracteres o **reemplaza** unos por otros
Formato: **tr** [opciones] set1 set2
-d borra los caracteres especificados en set1
-s reemplaza caracteres repetidos por uno único
Ejemplo: tr au pk < quijote.txt
En kn lkgr de lp Mpnchp, de kyo nombre
tr`a-z``A-Z` < quijote.txt EN UN LUGAR DE
LA MANCHA, DE CUYO NOMBRE..

- Algunas tareas de administración que necesitan búsqueda de ficheros:
 - Asegurarse de que no haya ficheros demasiado grandes.
 - Asegurarse que ningún usuario utilice demasiado espacio de disco.
 - Buscar donde está un determinado fichero.
 - Borrar los ficheros que pertenecen a un usuario en particular.
 - Visualizar los nombres de todos los ficheros modificados en los últimos dos días...
- find** [lista-directorios-búsqueda] [expresión]
 - Expresión : [opciones][tests][acciones][operadores]
 - opciones : modifican la forma de operar de find.
 - tests : condiciones de búsqueda a verificar por los ficheros a extraer.
 - acciones : operaciones a realizar en los ficheros encontrados.
 - operadores : agrupar expresiones.
 - Expresión por defecto : -print (acción de visualización)
- Opciones :
 - daystart : para tests usando tiempo tomarlo desde el día actual.
 - depth : procesar contenido de directorio antes de directorio.
 - maxdepth *numero* : máximo de directorios a descender (positivo).
 - mindepth *numero* : a que nivel empezar los tests.
 - xdev : no buscar en otras particiones.
- Tests :
 - amin *n1* -cmin *n2* -mmin *n3*
 - [último acceso | cambio estado | datos modificados] del fichero hace n1, n2, n3 minutos.
 - atime *n1* -ctime *n2* -mtime *n3*
 - lo mismo que el anterior pero n1, n2 y n3 son días.
 - name *patrón_fichero* -iname *patrón_fichero* -regex *ER*
 - Sensible/insensible a mayúsculas, expresiones regulares
 - perm *modo*
 - los permisos del fichero concuerdan con modo
 - size *n*[bck]
 - el fichero utiliza n unidades (b=bloques, c=bytes, k=kbytes) de espacio
 - user *nombre* -uid *n* -gid *n* -type *c*
 - los valores numéricos pueden especificarse, también, con +n y -n.
- Acciones :
 - Ejecutar comandos sobre los ficheros :
 - exec *comando* ; ejecuta sin preguntar
 - ok *comando* ; pregunta antes de ejecutar
 - en comando se utiliza {} para referirse al fichero encontrado por find.
 - Visualización diversa de información sobre el fichero : mirar man.
- Ejemplos :
 - find - maxdep 2
 - find . -user david
 - find /home -size +2500k -mtimes -7
 - find . -exec grep hola {} \;
 - find / -iname *.bak -ok rm {} \;

ER	concuerta con
.	cualquier carácter
[]	cualquiera de los caracteres entre corchetes, p.e. [abc] concuerda con a, b o c; [a-z] concuerda con cualquier letra minúscula
[^]	cualquier carácter que no esté entre corchetes
^	principio de línea
\$	final de línea
*	0 o más ocurrencias de la expresión regular anterior
\(\)	permite agrupar ER
\	escapa un metacarácter

crear partición
parted /dev/sda mklable gpt mkpart P2 ext4 2TiB 4TiB
chequear alineado partición
parted /dev/sda align-check optimal
borrar una partición parted /dev/sdb3 rm
mkfs [options] [-t type fs-options] device [size]

lsblk muestra discos y particiones en modo árbol
blkid muestra el dispositivo, atributos y el tipo de sistema de ficheros
umpe2fs muestra información de ext2/ext3/ext4
tune2fs ajusta parámetros en ext2/ext3/ext4
df muestra el espacio libre en particiones
du muestra el espacio de disco usado por ficheros y subdirectorios

Sed, permite realizar transformaciones básicas de un flujo de entrada
Sustituir apariciones de c1 por c2 en todo el fichero:
sed 's/c1/c2/g' fichero > fichero2
Sustituir apariciones de c1 por c2 en las líneas 1 a 10:
comando l sed '1,10 s/c1/c2/g'
Eliminar las líneas 2 a 7 del fichero sed '2,7 d' fichero > fichero2

Test -r el fichero existe y tiene permisos de escritura
Test -w el fichero existe y tiene permisos de escritura
Test -x el fichero existe y tiene permisos de ejecución
Test -f es un fichero regular
Test -d es un directorio

Montaje: sudo **mount** -t ext3 /dev/sda4 /home
Desmontaje/liberación: sudo umount /home
***umount** [opciones] {directorioldispositivo}
-a desmonta todo de /etc/mtab
Buscar Referencias: fuser -c /home
Listar montajes: mount o cat /etc/mtab
Listar dispositivos de bloque: lsblk

Creación de un volumen lógico:
lvcreate -L 256GB --name lv_var vg_pool
Como **resultado** tendremos un nuevo enlace en /dev/vg_pool/lv_var
Ahora hay que **dar formato** al volumen lógico: # mkfs -t ext4 /dev/vg_pool/lv_var
Y el volumen lógico ya está listo para el **montaje**:
mount -t ext4 /dev/vg_pool/lv_var /var
Agrandar/reducir un LV:
vextend -L+1GB /dev/vg_pool/lv_var
Agrandar el sistema de ficheros:
resize2fs /dev/vg_pool/lv_var

número de parámetros
***\$** todos los parámetros como una secuencia
@\$ todos los parámetros explícitamente separados
?\$ Contiene el estado de la salida, 0 ejecución correcta, todo lo demás ejecución incorrecta
\$\$ contiene el id del proceso en curso
#{var} calcula la longitud de la variable

-> ejecutar test es lo mismo que hacer []

Búsqueda, en un fichero (o más), de líneas que contienen un determinado patrón de caracteres:
grep [opciones] ER [ficheros...]
-c : numero de líneas capturadas.
-l : visualizar solo nombres de ficheros con líneas capturadas.
-v : extraer todas las líneas menos las que correspondan a ER.
-E : usa expresiones regulares extendidas no se especifica fichero, se usa la entrada estándar. Es. Lo mismo que usar egrep.
-q: no saca el resultado por pantalla

Borra el /tmp todos los días laborables a las 4:30 am:
30 4 * * 1-5 rm -rf /tmp/*
Ejecutar estos comandos con crontab -e
Escribe la hora, cada 15 minutos, durante la noche:
0,15,30,45 0-8,20-23 * * * echo Hora:\$(date)>>/tmp/horas

MANDAR CORREO ELECTRÓNICO
Con el comando at, automáticamente manda un. Correo al usuario después de la ejecución
Definir en /var/spool/cron/crontabs l usuari al que se le mandará un correo (MAILTO=user)
Comando mail: echo "\$listaNodos" | mail -s "procesos matados" root

```

Comprueba que sea una contraseña segura
read -p "Please enter the new password: " password
echo "$password" | grep '[:digit:]' | grep '[:lower:]' | grep '[:upper:]' | grep -q -E '.{8,}'
if [ $? -ne 0 ]; then
    echo "Invalid password: $password"
    exit 1
fi
echo $USER:"$password" | chpasswd

```

```

if test $(id -g $USER) -eq 0; then #comprueba que el usuario tenga privilegios
    if test $# -ne 2; then #comprobacion del numero de parametros
        echo "Numero incorrecto de parametros"
    else
        case $1 in
            -a) #añadir un usuario
                OldIFS=IFS
                while IFS=, read -r user password name #leer del fichero
                do
                    if [ "$user" ] && [ "$password" ] && [ "$name" ]; then
                        if test $(grep -c "$user" /etc/passwd) -gt 0; then #comprueba si el usuario ya existe
                            echo "El usuario '$user' ya existe"
                        else
                            useradd -c "$name" -m -U -K UID_MIN=1815 -k /etc/skel "$user"
                            echo "$user:$password" | chpasswd

                            chage -M 30 "$user"
                            echo "'$name' ha sido creado"
                        fi
                    else
                        echo "Campo invalido"
                    fi
                done <"$2"
                IFS=OldIFS;;

            -s) #eliminar un usuario
                mkdir /extra 2> /dev/null
                mkdir /extra/backup 2> /dev/null
                OldIFS=IFS
                while IFS=, read -r user da igual #leer del fichero
                do
                    if test $(grep -c "$user" /etc/passwd) -gt 0; then
                        if test $(grep -c /home/"$user" /etc/passwd) -eq 0; then
                            usermod -m -d /home/"$user" "$user"
                        fi
                        tar -cf /extra/backup/"$user".tar /home/"$user" &>/dev/null && userdel -r -f "$user" 2>/dev/null
                    fi
                done <"$2"
                IFS=OldIFS;;

            *)
                echo "Opcion invalida" >&2 ;;

        esac
    fi
else
    echo "Este script necesita privilegios de administracion"
fi

```

```

Agrupa usuarios por terminal
#!/bin/bash
EX_OSFILE=72
if [ ! -r /etc/passwd ]; then
    echo "Unable to open /etc/passwd"
    exit $EX_OSFILE
fi
grep 'bash$' /etc/passwd | cut -d':' -f1

```

Álvaro Pomar Martínez
796808

```

Enviar email de los nodos caídos (haciendo 3 pings)
lista_caidos=""
ip_prefix="172.16.1"
for i in $(seq 1 17) ; do
    if ! ping -q -c3 -W1 "${ip_prefix}.${i}" > /dev/null
    2>&1
    then
        lista_caidos+="${ip_prefix}.${i}\n"
    fi
done
if [ "x${lista_caidos}" != "x" ]
then
    echo -e "$lista_caidos" | mail -s "errores en nodos"
root
fi

```