

UD 8. **Redireccionamientos** **y utilidades de uso** **frecuente en Linux**



UD 8 – Redireccionamientos y utilidades de uso frecuente

INDICE DE CONTENIDOS

1.Redireccionamientos

2.Utilidades de uso frecuente

UD 8 – Redireccionamientos y utilidades de uso frecuente

2. Redireccionamientos

Los programas son piezas de código que se deben comunicar con “el mundo exterior” para poder interactuar con el usuario.



**PROGR
AMA**

De este modo, podemos identificar flujos de entrada (input) que permiten introducir información (desde el teclado o el ratón, por ejemplo).

Y también, podemos identificar flujos de salida (output) que permiten que el programa muestre información (por ejemplo, escribiendo en la pantalla del ordenador).

UD 8 – Redireccionamientos y utilidades de uso frecuente

Los sistemas operativos abren tres canales o flujos de comunicación para cada programa:

- Uno de entrada de datos (**stdin** –standar input)
- Y dos de salida. Estos canales se separan según el tipo de datos que transportan:
 - Uno para la salida de información “válida” (**stdout** – standard output)
 - Y otro para la salida de “mensajes de error” (**stderr** – standard error) .



UD 8 – Redireccionamientos y utilidades de uso frecuente

Estos canales de comunicación son “como tuberías” que interconectan dos elementos, en un extremo está nuestro programa y en el otro extremo el sistema operativo “conecta” un dispositivo (como el teclado, la pantalla...).

De este modo, si tú no indicas lo contrario, los sistemas operativos realizan las siguientes conexiones:



El flujo de entrada se conecta con el teclado y los dos flujos de salida se mandan a la pantalla.

Veamos un ejemplo práctico de esto:

UD 8 – Redireccionamientos y utilidades de uso frecuente

```
kike@kike-VirtualBox:~$ sudo adduser pedro
Añadiendo el usuario 'pedro' ...
Añadiendo el nuevo grupo 'pedro' (1001) ...
Añadiendo el nuevo usuario 'pedro' (1001) con grupo 'pedro' ...
Creando el directorio personal '/home/pedro' ...
Copiando los ficheros desde '/etc/skel' ...
sh: 1: find: not found
Introduzca la nueva contraseña de UNIX: 
Vuelva a escribir la nueva contraseña de UNIX: 
passwd: contraseña actualizada correctamente
Cambiando la información de usuario para pedro
Introduzca el nuevo valor, o presione INTRO para el predeterminado
Nombre completo []: Pedro López Ramos
Número de habitación []: 23
Teléfono del trabajo []: 654321876
Teléfono de casa []: 955235499
Otro []: 
chfn: el nombre contiene caracteres ilegales (no ASCII): «Pedro López Ramos»
¿Es correcta la información? [S/n] S
kike@kike-VirtualBox:~$
```

El S.O. “llama” al programa **adduser** con el parámetro **pedro** y abre los tres canales de datos, conectándolos al teclado (entrada) y a la pantalla (salida estándar y mensajes de error)

Lo recuadrado en azul se corresponde con el flujo de entrada al programa que teclea el usuario

Lo recuadrado en rojo se corresponde con el flujo de mensajes de error del programa y que se muestran en la pantalla

El resto de información que no está recuadrada se corresponde con el flujo de salida estándar del programa



ENTRADA

**PROGR
AMA**

SALIDA ESTÁNDAR

SALIDA DE
ERRORES



UD 8 – Redireccionamientos y utilidades de uso frecuente

Acabamos de describir la situación “por defecto” o “de base”. Sin embargo, desde la línea de comandos, podemos indicar que **queremos “conectar” nuestro programa con otros dispositivos que no sean el teclado o la pantalla.**

Por ejemplo, podríamos querer “desviar” la salida de errores a un fichero de texto para tener guardado el resultado de la ejecución de un programa y poder mirar posteriormente si hubo algún problema.



UD 8 – Redireccionamientos y utilidades de uso frecuente

O también podríamos querer que la entrada de datos de nuestro programa no sea desde el teclado sino que sea el contenido de un fichero el que se considere como entrada:



Veamos cómo se desvían los flujos de entrada y salida en Linux desde la línea de comandos:

REDIRECCIONAR LA ENTRADA: para que los datos de entrada se tomen de un fichero en vez de usar el teclado se usa la siguiente sintaxis: **Comando < fichero_de_entrada**

Veamos un ejemplo

UD 8 – Redireccionamientos y utilidades de uso frecuente

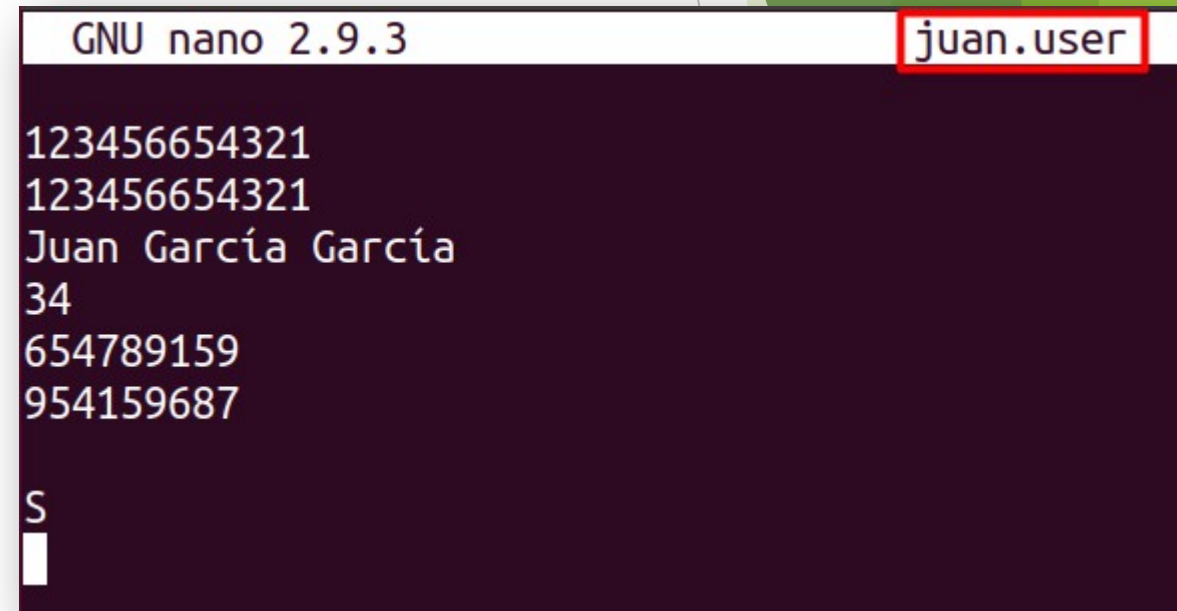
Vamos a alimentar el comando **sudo adduser juan** con los datos de un fichero en vez de utilizar el teclado.

Para ello vamos a crear un fichero llamado **juan.user** con los datos que hubiéramos teclado.

OJO: fíjate en que todas las líneas llevan un INTRO al final.

Las dos primeras líneas serán para la contraseña.

Después vienen los datos asociados al 'nombre completo', 'número de habitación', 'teléfono del trabajo', 'teléfono de casa', 'otro'. Y, por último, la respuesta a la pregunta '¿Es correcta la información [S/n]?'



```
GNU nano 2.9.3 juan.user
123456654321
123456654321
Juan García García
34
654789159
954159687
S
█
```

UD 8 – Redireccionamientos y utilidades de uso frecuente

Por último, vamos a lanzar el comando y alimentar el canal de entrada con el fichero creado.

Comando
o

Redireccionamiento
o de la entrada

```
kike@kike-VirtualBox:~$ sudo adduser juan < juan.user
Añadiendo el usuario `juan' ...
Añadiendo el nuevo grupo `juan' (1002) ...
Añadiendo el nuevo usuario `juan' (1002) con grupo `juan' ...
El directorio personal `/home/juan' ya existe. No se copiará desde `/etc/skel'.
Introduzca la nueva contraseña de UNIX: Vuelva a escribir la nueva contraseña de UNIX: p
asswd: contraseña actualizada correctamente
Cambiando la información de usuario para juan
Introduzca el nuevo valor, o presione INTRO para el predeterminado
Nombre completo []: Número de habitación []: Teléfono del trabajo []:
Teléfono de casa []: Otro []: chfn: el nombre contiene caracteres ilegales (no
ASCII): «Juan García García»
kike@kike-VirtualBox:~$
```

UD 8 – Redireccionamientos y utilidades de uso frecuente

Si usamos la utilidad **finger** para averiguar información del usuario recién creado, podemos ver que los datos coinciden con los que escribimos en el fichero:

```
kike@kike-VirtualBox:~$ finger juan
Login: juan
Directory: /home/juan
Office: 34, 654789159
Never logged in.
No mail.
No Plan.
kike@kike-VirtualBox:~$
```

Name: Juan García García
Shell: /bin/bash
Home Phone: 954159687

La utilidad finger se instala con: **sudo apt install finger**

Como vemos, hemos conseguido cambiar el flujo de entrada de nuestro comando (**sudo adduser juan**) para que use el fichero (**juan.user**) en vez de esperar a que el usuario introduzca los datos por teclado.

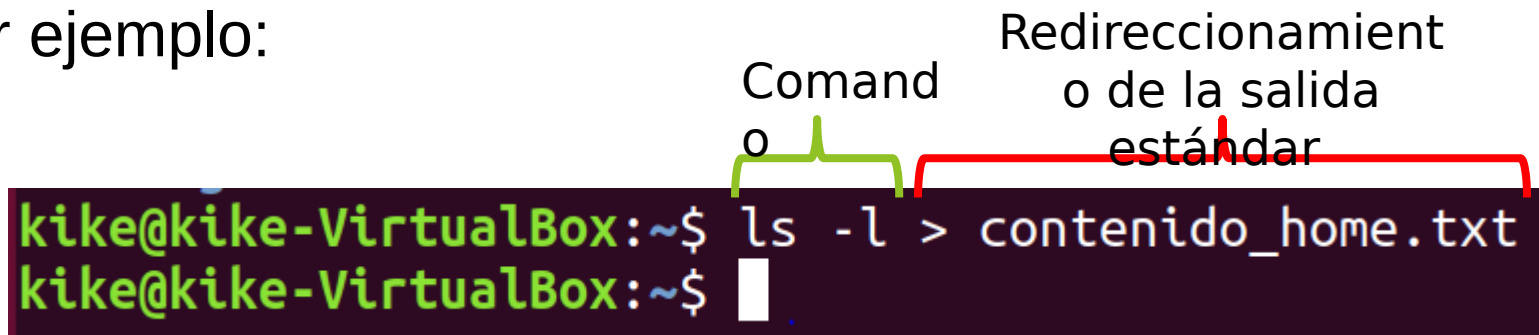
Realizar el **ejercicio 1** del boletín de problemas

UD 8 – Redireccionamientos y utilidades de uso frecuente

REDIRECCIONAR LA SALIDA ESTÁNDAR: en este caso nos interesa que los mensajes que pueda emitir un programa destinados al usuario NO se muestren en la pantalla sino que se escriban en un fichero de texto. Se utilizará la siguiente sintaxis:

Comando > fichero_de_salida

Por ejemplo:



The image shows a terminal window with a dark background and green text. The prompt is `kike@kike-VirtualBox:~$`. The command entered is `ls -l > contenido_home.txt`. Above the command, there are two annotations: a green bracket labeled "Comando" pointing to `ls -l`, and a red bracket labeled "Redireccionamiento o de la salida estándar" pointing to `> contenido_home.txt`. The second line of the terminal shows the prompt `kike@kike-VirtualBox:~$` followed by a cursor.

Con el comando `ls -l` mostramos una lista detallada de los archivos y carpetas contenidos en el directorio actual. Al usar el redireccionamiento de la salida, esta información no aparecerá en la pantalla sino que se guardará en el fichero

UD 8 – Redireccionamientos y utilidades de uso frecuente

Si abrimos el fichero podemos ver el contenido que se ha guardado.



```
GNU nano 2.9.3 contenido_home.txt
total 84
drwxr-xr-x 2 kike kike 4096 nov  9 13:02 Apuntes
-rw-r--r-- 1 kike kike   56 ene 11 11:05 busquedas.dns
drwxr-xr-x 2 kike kike 4096 ene  4 12:54 config
-rw-r--r-- 1 kike kike    0 ene 11 13:12 contenido_home.txt
drwxr-xr-x 2 kike kike 4096 ene  4 16:25 Descargas
drwxr-xr-x 2 kike kike 4096 ene  4 13:00 Documentos
drwxr-xr-x 2 kike kike 4096 sep 23 19:55 Escritorio
-rwxrwxrwx 1 kike kike 8980 sep 23 19:39 examples.desktop
-rw-r--r-- 1 kike kike    2 ene 11 12:16 fichero
drwxr-xr-x 3 kike kike 4096 nov  9 13:07 ficheros
drwxr-xr-x 2 kike kike 4096 ene  4 13:36 Imágenes
-rw-r--r-- 1 kike kike   73 ene 11 10:47 juan.user
drwxr-xr-x 2 kike kike 4096 sep 23 19:55 Música
drwxr-xr-x 2 kike kike 4096 nov  9 13:00 Otras
```

Vamos a hacer otra prueba con el siguiente comando: `ls > contenido_home.txt`

Si abrimos de nuevo el fichero vemos que se ha borrado el contenido anterior y en su lugar aparece el nuevo resultado. Así que el **nuevo contenido “machaca” el contenido previo** del fichero.



```
GNU nano 2.9.3 contenido_home.txt
Apuntes
busquedas.dns
config
contenido_home.txt
Descargas
Documentos
Escritorio
examples.desktop
fichero
ficheros
Imágenes
juan.user
Música
Otras
pendrive
Plantillas
```

UD 8 – Redireccionamientos y utilidades de uso frecuente

Pero ¿y si lo que queremos es añadir al fichero el nuevo contenido sin borrar lo que hubiera previamente?

Para redirigir la salida de un programa a un fichero **añadiendo** el nuevo contenido se usa el operador >>

Ejemplo:

```
kike@kike-VirtualBox:~$ echo "Hoy es..." > hoy.txt
kike@kike-VirtualBox:~$ date >> hoy.txt
kike@kike-VirtualBox:~$
```

El primer comando escribe el texto “Hoy es...” en el fichero. El segundo comando **añade** la fecha del sistema sin eliminar el texto anterior. El contenido del fichero es, por tanto, la “suma” de ambos resultados:

```
GNU nano 2.9.3 hoy.txt
Hoy es...
lun ene 11 19:32:11 CET 2021
```

Realizar los ejercicios
2 al 3 del boletín de
problemas

UD 8 – Redireccionamientos y utilidades de uso frecuente

REDIRECCIONAR LA SALIDA DE ERRORES: en este caso nos interesa que los mensajes de error que pueda emitir un programa NO se muestren en la pantalla sino que se escriban en un fichero de texto.

Se utilizará la siguiente sintaxis:

Comando `2> fichero_de_errores`

Recordamos que los programas emiten 2 posibles salidas:

- **La estándar** (1), que se desvía mediante `>` o `>>` pero también puede escribirse con la sintaxis `1>` o `1>>`
- **La de errores** (2), que se desvía mediante `2>` o `2>>`, en el primer caso se desvía a un fichero “machando” el contenido que tuviese y en el segundo caso, los mensajes se añaden al final del fichero sin borrar el contenido previo.



ERROR

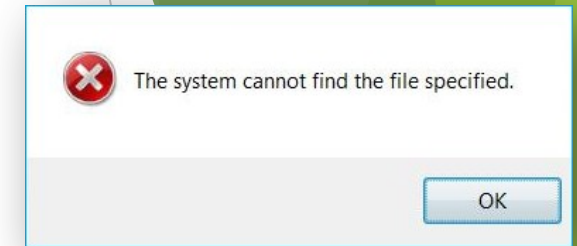
UD 8 – Redireccionamientos y utilidades de uso frecuente

Veamos un ejemplo:

Comand
o

Redireccionamiento
de la salida de
errores

```
cp no_existe.txt Descargas 2> resultado_copia
```



Dado que no existe ningún fichero llamado *no_existe.txt* no puede ser copiado a la carpeta de *Descargas*.

El mensaje de error que devuelve el comando *cp* se manda al fichero *resultado_copia*, que tendrá el siguiente contenido:

```
GNU nano 2.9.3                                resultado_copia
cp: no se puede efectuar `stat' sobre 'no_existe.txt': No existe el archivo o el directorio
```

Desviar la salida de errores de un script a un fichero es bastante habitual para poder consultar posteriormente el resultado de la ejecución.

UD 8 – Redireccionamientos y utilidades de uso frecuente

REDIRECCIONAR VARIOS FLUJOS: simplemente vamos a combinar las distintas sintaxis para redireccionar varios flujos simultáneamente. Un posible ejemplo:

Comando < fichero_entrada 2> fichero_de_errores

Ejemplo:

```
kike@kike-VirtualBox:~$ sudo adduser maria < maria.user 2>> errores_usuario
Añadiendo el usuario `maria' ...
Añadiendo el nuevo grupo `maria' (1003) ...
Añadiendo el nuevo usuario `maria' (1003) con grupo `maria' ...
Creando el directorio personal `/home/maria' ...
Copiando los ficheros desde `/etc/skel' ...
Cambiando la información de usuario para maria
Introduzca el nuevo valor, o presione INTRO para el predeterminado
    Nombre completo []:      Número de habitación []:      Teléfono del tra
bajo []:      Teléfono de casa []:      Otro []: ¿Es correcta la información? [S
/n] kike@kike-VirtualBox:~$
```

Realizar los ejercicios
4 al 5 del boletín de
problemas

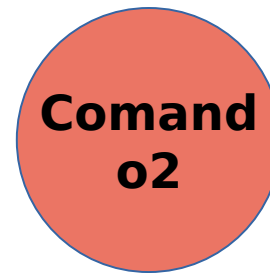
Tomamos la entrada desde un fichero y la salida de errores la mandamos a otro fichero

UD 8 – Redireccionamientos y utilidades de uso frecuente

CONECTAR LA SALIDA DE UN PROCESO A LA ENTRADA DE OTRO PROCESO: en este caso nos interesa **encadenar** dos o más procesos de modo que la salida de uno sea la entrada del siguiente. La sintaxis que se usa es la siguiente:

Comando1 | Comando2

Gráficamente sería algo como:



tubería.
Se pueden encadenar
tantos comandos como se
deseen:

El carácter | se obtiene
pulsando la tecla **Alt-Gr** y,
sin soltarla, la tecla del **1**

Para poner un ejemplo del
uso de tubería
necesitamos conocer los
comandos que
estudiamos en el
siguiente apartado...

UD 8 – Redireccionamientos y utilidades de uso frecuente

2. Utilidades de uso frecuente

La primera utilidad que vamos a estudiar “conecta” con el apartado anterior ya que normalmente se usa en combinación con la notación de “tuberías”.

grep (Global Regular Expression Print) es un comando que permite **filtrar un texto** y seleccionar sólo las líneas que cumplan con una condición.

Es uno de los comandos más usados en Linux. En su notación básica (sin “tuberías”) tendríamos:

grep “texto a buscar” fichero



UD 8 – Redireccionamientos y utilidades de uso frecuente

Partimos del fichero “usuarios.txt” de la imagen y hacemos algunos ejemplos:

```
kike@kike-VirtualBox:~$ grep "juan" usuarios.txt
juan2020
juan2019
juanjo2020
juanfran
```

```
kike@kike-VirtualBox:~$ grep "18" usuarios.txt
alberto2018
hugo2018
```

Este último ejemplo pero con la notación de “tuberías” equivalente:

```
kike@kike-VirtualBox:~$ cat usuarios.txt | grep "18"
alberto2018
hugo2018
```

```
GNU nano 2.9.3 usuarios.txt
juan2020
rosa2019
alberto2018
tomas
josema2020
admin
juan2019
rosa
hugo2018
carlos2019
alberto
fernando
kike2020
hugo
ursula2021
juanjo2020
juanfran
jose
ramon2017
```


UD 8 – Redireccionamientos y utilidades de uso frecuente

Como vemos, nos permite buscar algo en uno o **varios ficheros de texto**:

```
kike@kike-VirtualBox:~$ grep "21" usuarios.txt fecha_hoy
usuarios.txt:ursula2021
fecha_hoy:lun ene 11 19:30:28 CET 2021
```

Con tuberías:

```
kike@kike-VirtualBox:~$ cat usuarios.txt fecha_hoy | grep "21"
ursula2021
lun ene 11 19:30:28 CET 2021
```

Esto nos permite buscar rápidamente información en archivos de configuración o del sistema:

```
kike@kike-VirtualBox:~$ sudo cat /etc/passwd | grep "kike"
kike:x:1000:1000:kike,,,:/home/kike:/bin/bash
kike@kike-VirtualBox:~$
```



UD 8 – Redireccionamientos y utilidades de uso frecuente

Grep presenta algunos **modificadores** interesantes:

```
kike@kike-VirtualBox:~$ grep -n "juan" usuarios.txt
1:juan2020
7:juan2019
16:juanjo2020
17:juanfran
```

Con **-n** se
imprime el
Número de línea
del fichero

```
kike@kike-VirtualBox:~$ grep -v "20" usuarios.txt
tomas
admin
rosa
alberto
fernando
hugo
juanfran
jose
```

Con **-v** se
inVierte el
patrón de
búsqueda y nos
muestra lo que
NO COINCIDA

```
kike@kike-VirtualBox:~$ grep -c "juan" usuarios.txt
4
```

Con **-c** se CUENTA el
número de veces que se
cumple el patrón de
búsqueda

UD 8 – Redireccionamientos y utilidades de uso frecuente

Más modificadores interesantes:

```
kike@kike-VirtualBox:~$ grep -xn "jose" usuarios.txt
18:jose
kike@kike-VirtualBox:~$ grep -n "jose" usuarios.txt
5:josema2020
18:jose
```

Con **-x** solo se muestra la coincidencia eXacta con el patrón de búsqueda

```
kike@kike-VirtualBox:~$ grep -i "JUAN" usuarios.txt
juan2020
juan2019
juanjo2020
juanfran
```

Con **-i** se Ignoran las diferencias de mayúsculas y minúsculas en el patrón de búsqueda

```
kike@kike-VirtualBox:~$ cp usuarios.txt usuarios2.txt
kike@kike-VirtualBox:~$ grep -l "kike" usu*.txt
usuarios2.txt
usuarios.txt
```

Con **-l** se muestran los nombres de fichero (fiLename) en los que aparezca el patrón de búsqueda

UD 8 – Redireccionamientos y utilidades de uso frecuente

Además, también podemos aplicar **expresiones regulares al patrón de búsqueda** al igual que cuando usamos ? o * en la línea de comandos. Sin embargo, se usa una sintaxis diferente más potente y que es un **estándar** que encontraréis en otros contextos.

Lo primero que tenemos que hacer es **dejar de pensar como cuando usamos la línea de comandos**. Fíjate en la diferencia:

```
kike@kike-VirtualBox:~$ ls *a*
busquedas.dns  examples.desktop  maria.user  salida.txt  usuarios.txt
errores_usuario  fecha_hoy  resultado_copia  usuarios2.txt
```

```
kike@kike-VirtualBox:~$ grep "a" usuarios.txt
juan2020
rosa2019
alberto2018
tomas
josema2020
...
```

En la línea de comandos tenemos que especificar el **CONTENIDO COMPLETO DEL TEXTO: *a*** (algo+a+algo).

Sin embargo, en grep nos basta con decir qué esperamos encontrar **EN ALGÚN PUNTO DEL TEXTO**.

UD 8 – Redireccionamientos y utilidades de uso frecuente

También podemos indicar si queremos que nuestro patrón de búsqueda esté al principio o al final de una línea:

```
kike@kike-VirtualBox:~$ grep "^a" usuarios.txt
alberto2018
admin
alberto
```

El ^ hace que el patrón de búsqueda esté al comienzo de la línea

```
kike@kike-VirtualBox:~$ grep "a$" usuarios.txt
rosa
```

El \$ hace que el patrón de búsqueda se sitúe al final de la línea

También hay caracteres especiales o comodines:

```
kike@kike-VirtualBox:~$ grep "jose" usuarios.txt
josema2020
jose
kike@kike-VirtualBox:~$ grep "jose." usuarios.txt
josema2020
```

El . Se reemplaza por 1 y solo 1 carácter

UD 8 – Redireccionamientos y utilidades de uso frecuente

Y un poco más:

```
kike@kike-VirtualBox:~$ grep ".l." usuarios.txt
alberto2018
carlos2019
alberto
ursula2021
```

Aquí queremos todo lo que tenga un 'l' con un carácter por delante y otro por detrás

```
kike@kike-VirtualBox:~$ grep "201[89]" usuarios.txt
rosa2019
alberto2018
juan2019
hugo2018
carlos2019
```

Entre [] podemos poner una lista de caracteres admitidos en el patrón de búsqueda

```
kike@kike-VirtualBox:~$ grep "201[0-9]" usuarios.txt
rosa2019
alberto2018
juan2019
hugo2018
carlos2019
ramon2017
```

Entre [] también podemos poner un rango de caracteres admitidos: [0-9] o [A-Z] o [a-z]

Realiza los ejercicios del 4 al 5 del boletín

UD 8 – Redireccionamientos y utilidades de uso frecuente

La segunda utilidad de uso frecuente que vamos a estudiar es **sort** y nos permite ordenar un bloque de información alfanumérica. Su sintaxis básica es:

sort fichero

Esto mostraría en pantalla
el contenido del fichero
ordenado.

```
kike@kike-VirtualBox:~$ sort usuarios.txt
admin
alberto
alberto2018
carlos2019
fernando
hugo
hugo2018
jose
josema2020
juan2019
juan2020
juanfran
juanjo2020
kike2020
ramon2017
rosa
rosa2019
tomas
ursula2021
```



UD 8 – Redireccionamientos y utilidades de uso frecuente

También podemos combinarlo con el operador para redirigir la salida a un fichero:

```
sort usuarios.txt > usuarios_ordenados.txt
```

O bien ordenar varios ficheros y volcarlos a otro fichero de resultados:

```
sort usuarios.txt usuarios2.txt > total_usu_ordenados.txt
```

Otra opción sería usar una “tubería” para que la salida de un comando sea la entrada de otro:

```
kike@kike-VirtualBox:~$ ls -l | sort
drwxr----- 2 kike kike 4096 nov 19 08:26 Recetas
drwxr-xr-x 2 kike kike 4096 ene 25 12:23 sorpresa
drwxr-xr-x 2 kike kike 4096 ene 28 08:56 Documentos
drwxr-xr-x 2 kike kike 4096 ene 28 08:57 Escritorio
drwxr-xr-x 2 kike kike 4096 ene 28 10:05 Descargas
```



UD 8 – Redireccionamientos y utilidades de uso frecuente

¿Cómo ordena sort?

Cuando **sort** se encuentra letras ordena en orden alfabético y, si hay dos letras iguales, pone primero la minúscula y después la mayúscula.

Observa:

```
kike@kike-VirtualBox:~$ cat letras.txt
```

```
A  
B  
C  
E  
a  
b  
c  
d
```

```
kike@kike-VirtualBox:~$ sort letras.txt
```

```
a  
A  
b  
B  
c  
C  
d  
E
```


UD 8 – Redireccionamientos y utilidades de uso frecuente

Cuando **sort** se encuentra una secuencia de **dígitos** también los ordena por orden alfabético. Observa:

```
kike@kike-VirtualBox:~$ cat numeros.txt
91
9
911
3
44
1111
56
```

```
kike@kike-VirtualBox:~$ sort numeros.txt
1111
3
44
56
9
91
911
```

Esto nos puede **venir bien** si estamos ordenando números de teléfonos o DNIs.

Pero nos **viene mal** si queremos entender cada secuencia de dígitos como un **NÚMERO** que queremos ordenar de menor a mayor.

UD 8 – Redireccionamientos y utilidades de uso frecuente

Si queremos que **sort** interprete una secuencia de dígitos como **un número** tenemos que usar el modificador **-n**. Observa:

```
kike@kike-VirtualBox:~$ cat numeros.txt
91
9
911
3
44
1111
56
```

```
kike@kike-VirtualBox:~$ sort -n numeros.txt
3
9
44
56
91
911
1111
```

El modificador **-r** invierte el orden (Reverse) tanto con números como con letras.

```
kike@kike-VirtualBox:~$ sort -rn numeros.txt
1111
911
91
56
44
9
3
```

UD 8 – Redireccionamientos y utilidades de uso frecuente

Hasta ahora **sort** ordena un texto tomando la información desde el principio de la línea pero ¿qué pasa si tenemos un texto organizado por columnas y queremos ordenar por una columna en concreto?

```
kike@kike-VirtualBox:~$ ls -l
total 140
drwxr-xr-x 2 kike kike 4096 nov  9 13:02 Apuntes
-rw-r--r-- 1 kike kike  56 ene 11 11:05 busquedas.dns
-rw-r--r-- 1 kike kike 198 ene 11 13:35 contenido_home.txt
drwxr-xr-x 2 kike kike 4096 ene 28 10:05 Descargas
drwxr-xr-x 2 kike kike 4096 ene 28 08:56 Documentos
-rw-r--r-- 1 kike kike 160 ene 12 12:42 errores_luis.user
-rw-r--r-- 1 root sudo  76 ene 14 18:11 errores.txt
-rw-r--r-- 1 kike kike 242 ene 14 11:58 errores_usuario
drwxr-xr-x 2 kike kike 4096 ene 28 08:57 Escritorio
-rwxrwxrwx 1 kike kike 8980 sep 23 19:39 examples.desktop
-rw-r--r-- 1 kike kike  36 ene 11 19:30 fecha_hoy
-rw-r--r-- 1 kike kike   2 ene 11 12:16 fichero
```

Columna 1 2 3 4 5 ...

Sort entiende que el carácter ESPACIO es el separador de columnas

UD 8 – Redireccionamientos y utilidades de uso frecuente

Usaremos el modificador **-k** seguido del **número de columna** por la que queremos ordenar. Observa los siguientes ejemplos:

```
kike@kike-VirtualBox:~$ ls -l | sort -k5
total 140
-rw-r--r-- 1 kike kike 0 ene 14 18:11 salida.txt
-rw-r--r-- 1 kike kike 160 ene 12 12:42 errores_luis.user
-rw-r--r-- 1 kike kike 164 ene 28 17:29 usuarios.txt
-rw-r--r-- 1 kike kike 164 ene 28 17:33 usuarios2
-rw-r--r-- 1 kike kike 164 ene 28 17:38 usuarios2
-rw-r--r-- 1 kike kike 16 ene 28 18:57 letras.txt
-rw-r--r-- 1 root root 1963 ene 19 19:05 Partida.3
-rw-r--r-- 1 kike kike 198 ene 11 13:35 contenido
-rw-r--r-- 1 kike kike 22 ene 28 19:15 numeros.txt
-rw-r--r-- 1 kike kike 242 ene 14 11:58 errores_u
-rw-r--r-- 1 kike kike 2 ene 11 12:16 fichero
-rw-r--r-- 1 kike kike 328 ene 28 17:39 total_usu
-rw-r--r-- 1 kike kike 36 ene 11 19:30 fecha_hoy
```

```
kike@kike-VirtualBox:~$ ls -l | sort -k5n
total 140
-rw-r--r-- 1 kike kike 0 ene 14 18:11 salida.txt
-rw-r--r-- 1 kike kike 2 ene 11 12:16 fichero
-rwxrwxrwx 1 root root 8 ene 25 13:15 log_sistema -> /var/lo
-rw-r--r-- 1 kike kike 16 ene 28 18:57 letras.txt
-rw-r--r-- 1 kike kike 22 ene 28 19:15 numeros.txt
-rw-r--r-- 1 kike kike 36 ene 11 19:30 fecha_hoy
-rw-r--r-- 1 kike kike 39 ene 11 19:32 hoy.txt
-rw-r--r-- 1 kike kike 56 ene 11 11:05 busquedas.dns
-rw-r--r-- 1 kike kike 75 ene 14 11:57 maria.user
-rw-r--r-- 1 root sudo 76 ene 14 18:11 errores.txt
-rw-r--r-- 1 kike kike 92 ene 12 19:43 resultado_copia
-rw-r--r-- 1 kike kike 160 ene 12 12:42 errores_luis.user
```

UD 8 – Redireccionamientos y utilidades de uso frecuente

Si queremos que el **separador de columnas** sea otro, lo especificamos con **-t** seguido del carácter que usaremos como separador. Observa:

```
kike@kike-VirtualBox:~$ cat /etc/passwd | sort -t: -k7
juan:x:1002:1002:Juan García García,34,654789159,954159687:/home/juan:/bin/bash
kike:x:1000:1000:kike,,,:/home/kike:/bin/bash
maria:x:1003:1003:María García García,22,654789159,954159687:/home/maria:/bin/bash
pedro:x:1001:1001:Pedro López Ramos,23,654321876,955235499:/home/pedro:/bin/bash
root:x:0:0:root:/root:/bin/bash
gdm:x:121:125:Gnome Display Manager:/var/lib/gdm3:/bin/false
gnome-initial-setup:x:120:65534::/run/gnome-initial-setup:/bin/false
hplip:x:118:7:HPLIP system user,,,:/var/run/hplip:/bin/false
speech-dispatcher:x:111:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
```

En este caso estamos ordenando por la última columna del archivo que indica qué intérprete de comandos usa cada usuario.

UD 8 – Redireccionamientos y utilidades de uso frecuente

Por último, podemos mezclar varios comandos encadenando la salida de uno con la entrada de otro.

Ejemplo:

```
kike@kike-VirtualBox:~$ cat /etc/passwd | grep "bash" | sort
juan:x:1002:1002:Juan García García,34,654789159,954159687:/home/juan:/bin/bash
kike:x:1000:1000:kike,,,:/home/kike:/bin/bash
maria:x:1003:1003:María García García,22,654789159,954159687:/home/maria:/bin/bash
pedro:x:1001:1001:Pedro López Ramos,23,654321876,955235499:/home/pedro:/bin/bash
root:x:0:0:root:/root:/bin/bash
```

En este caso la salida de *cat* la filtramos para quedarnos con las líneas que contengan “*bash*”. Las líneas resultantes las ordenamos utilizando *sort*.

Realiza los ejercicios del 1 al 3 del
boletín

UD 8 – Redireccionamientos y utilidades de uso frecuente

El tercer bloque de utilidades de uso frecuente que vamos a estudiar sirve para **empaquetar y comprimir archivos**.

La filosofía del software de Linux es muy de: *“Haz una sola cosa pero hazla bien”*.

Con esta premisa en Linux se construyen dos herramientas:

- **Tar**: sirve para empaquetar varios archivos en uno solo pero sin comprimir el archivo resultante.
- **Gzip**: sirve para comprimir 1 solo archivo.



UD 8 – Redireccionamientos y utilidades de uso frecuente

La utilidad que **empaqueta archivos** se llama **tar** (Tape Archiver) y se utilizaba inicialmente para escribir varios archivos en cintas magnéticas que se usaban para hacer backups de sistemas.

La utilidad que **comprime un archivo** se llama **gzip** (GunZip).

Normalmente se encadenan los dos procesos:

1. Se empaquetan varios archivos en un fichero **.tar**
2. Se comprime el archivo **.tar** y se obtiene un **.tar.gz** o también **.tgz**

En este formato nos podemos encontrar mucho software de Linux



UD 8 – Redireccionamientos y utilidades de uso frecuente

¿Y por qué no se usa ZIP como en Windows?

La herramienta **ZIP** para la compresión de múltiples archivos nació como un **software de pago** para sistemas Windows.

Esto hizo que la comunidad Linux desarrollase la pareja **tar** (para empaquetar) + **gzip** (para comprimir).

Hoy en día, en Linux existen dos comandos que hacen lo mismo que la herramienta ZIP de Windows. Estos comandos son: **zip** y **unzip**.

Sin embargo, se sigue usando la pareja **tar+gzip**, **veamos porqué**.



UD 8 – Redireccionamientos y utilidades de uso frecuente

Comparativa:

UTILIDAD ZIP	UTILIDADES TAR + GZIP
Está diseñada para Windows y es capaz de guardar y restaurar los permisos y atributos del sistema de ficheros de MSDOS	Está diseñada para Linux y es capaz de guardar y restaurar los permisos (-rwxr-xr-x) y atributos de sistema de ficheros de Linux
Permite comprimir y descomprimir varios archivos	Gzip permite comprimir un solo archivo pero unido al comando Tar tiene la misma funcionalidad
Cuando se comprimen varios archivos, ZIP comprime cada uno independientemente y después los “empaqueta” en el archivo final. Esto consigue una tasa de compresión más baja.	Con tar + gzip, primero empaquetamos varios archivos y después comprimimos el archivo resultante. Esto consigue una tasa de compresión más alta.

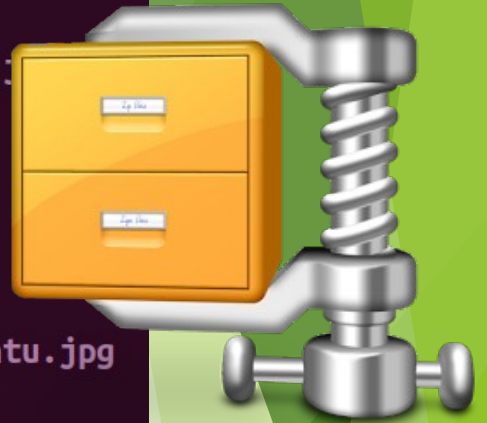


UD 8 – Redireccionamientos y utilidades de uso frecuente

Utilidad gzip

Veamos cómo funciona **gzip** en primer lugar:

```
kike@kike-VirtualBox:~/Descargas$ ls
cliente                perro-feo.jpg
enlace-duro-docum.docx queen_under_pressure_official_video_-8840728843474474226.mp3
kike@kike-VirtualBox:~/Descargas$
kike@kike-VirtualBox:~/Descargas$ gzip -v enlace-duro-docum.docx
enlace-duro-docum.docx: 16.3% -- replaced with enlace-duro-docum.docx.gz
kike@kike-VirtualBox:~/Descargas$
kike@kike-VirtualBox:~/Descargas$ ls
cliente                perro-feo.jpg
enlace-duro-docum.docx.gz queen_under_pressure_official_video_-8840728843474474226.mp3
kike@kike-VirtualBox:~/Descargas$
```



Gzip comprime el archivo indicado, creando un nuevo archivo con el mismo pero con la extensión .gz, finalmente borra el archivo original.

La opción -v (Verbose) hace que se nos muestre el mensaje informativo:

```
enlace-duro-docum.docx: 16.3% -- replaced with enlace-duro-docum.docx.gz
```

Si no se pone esta opción, este mensaje no aparece.

UD 8 – Redireccionamientos y utilidades de uso frecuente

Si queremos conservar el archivo original solo tenemos que añadir la opción `-k` (Keep = guardar):

```
kike@kike-VirtualBox:~/Descargas$ gzip -vk perro-feo.jpg
perro-feo.jpg:  0.3% -- replaced with perro-feo.jpg.gz
kike@kike-VirtualBox:~/Descargas$
kike@kike-VirtualBox:~/Descargas$ ls
cliente                perro-feo.jpg.gz      simpson_ubuntu.jpg
enlace-duro-docum.docx.gz  queen_under_pressure_official_video_-8840728843474474226.mp3
perro-feo.jpg            simbolico
```

Si queremos descomprimir usamos `-d` (decompress):

```
kike@kike-VirtualBox:~/Descargas$ ls
cliente                perro-feo.jpg.gz      simpson_ubuntu.jpg
enlace-duro-docum.docx.gz  queen_under_pressure_official_video_-8840728843474474226.mp3
perro-feo.jpg            simbolico
kike@kike-VirtualBox:~/Descargas$ gzip -d enlace-duro-docum.docx.gz
kike@kike-VirtualBox:~/Descargas$ ls
cliente                perro-feo.jpg.gz      simpson_ubuntu.jpg
enlace-duro-docum.docx  queen_under_pressure_official_video_-8840728843474474226.mp3
perro-feo.jpg            simbolico
kike@kike-VirtualBox:~/Descargas$
```

UD 8 – Redireccionamientos y utilidades de uso frecuente

Utilidad tar

Veamos cómo funciona:

```
kike@kike-VirtualBox:~$ tar cvf textos.tar *.txt
contenido_home.txt
desordenado.txt
errores.txt
hoy.txt
letras.txt
numeros.txt
salida.txt
total_usu_ordenados.txt
usuarios2.txt
usuarios_ordenados.txt
usuarios.txt
```

En este caso estamos **empaquetando** todos los ficheros *.txt y guardándolos en un fichero llamado **textos.tar**

```
kike@kike-VirtualBox:~$
```

```
kike@kike-VirtualBox:~$ ls
```

Apuntes	errores_luis.user	fichero	maria.user	Plantillas	sorpresa
busquedas.dns	errores.txt	ficheros	Música	Público	textos.tar
contenido_home.txt	errores_usuario	hoy.txt	numeros.txt	Recetas	total_usu_ordenados.txt
Descargas	Escritorio	Imágenes	Otras	resultado_copia	usuarios2.txt
desordenado.txt	examples.desktop	letras.txt	Partida.java	salida.txt	usuarios_ordenados.txt
Documentos	fecha_hoy	log_sistema	pendrive	snap	usuarios.txt

```
kike@kike-VirtualBox:~$
```


UD 8 – Redireccionamientos y utilidades de uso frecuente

Descripción de las opciones:

- **v** : emite un mensaje descriptivo de cada fichero añadido al paquete (v = Verbose)
- **f** : permite especificar el nombre del fichero .tar en el que se guardará la información (f = File).
- **c** : indica que se realizará un manejo del fichero anterior creando uno nuevo (c = Create). Esto implica que si ya existía el fichero previamente se borrarán los datos que hubiese.



UD 8 – Redireccionamientos y utilidades de uso frecuente

Estas mismas opciones también nos valen para empaquetar todo el contenido de una **carpeta**:

```
kike@kike-VirtualBox:~$ tar cvf descargas.tar Descargas
Descargas/
Descargas/enlace-duro-docum.docx
Descargas/queen_under_pressure_official_video_-8840728843474474226.mp3
Descargas/simbolico
Descargas/perro-feo.jpg
Descargas/cliente
Descargas/simpson_ubuntu.jpg
```

Si, en vez de crear un nuevo fichero, queremos añadir a un fichero ya existente usaremos la opción **r**, en vez de la opción **c**:

```
kike@kike-VirtualBox:~$ tar rvf textos.tar maria.user
maria.user
kike@kike-VirtualBox:~$
```

UD 8 – Redireccionamientos y utilidades de uso frecuente

Si queremos listar los ficheros que hay contenidos en un paquete .tar usamos la opción **t** (Test):

```
kike@kike-VirtualBox:~$ tar tf textos.tar
contenido_home.txt
desordenado.txt
errores.txt
hoy.txt
letras.txt
numeros.txt
salida.txt
total_usu_ordenados.txt
usuarios2.txt
usuarios_ordenados.txt
usuarios.txt
maria.user
```

También podemos borrar un archivo del paquete con la sintaxis:

```
kike@kike-VirtualBox:~$ tar f textos.tar --delete maria.user
```

UD 8 – Redireccionamientos y utilidades de uso frecuente

Por último, si queremos “desempaquetar” los ficheros de un .tar usamos la opción x (eXtract):

```
kike@kike-VirtualBox:~$ cp textos.tar ./Documentos/
kike@kike-VirtualBox:~$ cd Documentos/
kike@kike-VirtualBox:~/Documentos$ tar xvf textos.tar
contenido_home.txt
desordenado.txt
errores.txt
hoy.txt
letras.txt
numeros.txt
salida.txt
total_usu_ordenados.txt
usuarios2.txt
usuarios_ordenados.txt
usuarios.txt
kike@kike-VirtualBox:~/Documentos$ ls
contenido_home.txt  errores.txt  letras.txt  salida.txt  total_usu_ordenados.txt  usuarios_ordenados.txt
desordenado.txt     hoy.txt     numeros.txt  textos.tar  usuarios2.txt            usuarios.txt
```