

Introducción a Linux

Módulo 5



Comandos de filtro

Introducción

Los comandos **bzcat**, **cat**, **cut**, **head**, **less**, **md5sum**, **nl**, **od**, **paste**, **sed**, **sha256sum**, **sha512sum**, **sort**, **split**, **tail**, **tr**, **uniq**, **wc**, **xzcat**, **zcat** son filtros porque tienen la capacidad de **procesar el texto frente a redireccionamiento de la entrada estándar**, **luego de tuberías** o bien porque **el resultado es útil para volcarlo en un archivo redireccionando la salida estándar**.



Comandos de filtro

Comando cat

El comando cat pasa el contenido de un archivo a la salida estándar (STDOUT), el resultado será mostrar en pantalla el contenido de un archivo.

Opciones:

- s quita las líneas en blanco repetidas
- A muestra los caracteres no imprimibles (\$: nueva línea, ^I: tab).

[code]

```
# cat /etc/passwd
```

```
root:x:0:0:root:/root:/bin/bash
```

```
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
```

```
bin:x:2:2:bin:/bin:/bin/sh
```

```
sys:x:3:3:sys:/dev:/bin/sh
```

```
(....salida cortada...)
```

```
# cat -A /etc/passwd
```

```
root:x:0:0:root:/root:/bin/bash$
```

```
daemon:x:1:1:daemon:/usr/sbin:/bin/sh$
```

```
bin:x:2:2:bin:/bin:/bin/sh$
```

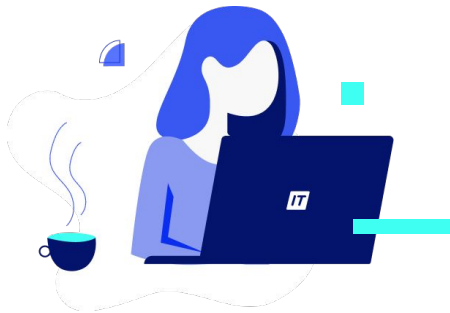
```
(....salida cortada...)
```

```
[/code]
```

Ejemplo

Redireccionar la salida de un archivo a otro:

```
# cat /etc/passwd > otro.txt
```



Comando cut

El comando **cut** permite separar en secciones el contenido de un archivo o el contenido recibido por STDIN

Opciones:

- d Define un delimitador para separar en campos (el predeterminado es TAB)
- c Cantidad de caracteres.
- f Campo o campos a mostrar.

Mostrar los primeros 8 caracteres por línea.

```
[code]
# cut -c1-8 /etc/passwd
root:x:0
daemon:x
bin:x:2:
(...salida cortada...)
```

Toma como delimitador el signo ":" y se muestra la primera columna.

```
# cut -d":" -f1 /etc/passwd
root
daemon
bin
(...salida cortada...)
```

Muestra los primeros 8 caracteres y luego del 20 al 25.

```
# cut -c1-8,20-25 /etc/passwd
root:x:0ot:/bi
daemon:x:/usr/
bin:x:2:/bin/s
(...salida cortada...)
```

Toma como delimitador el signo ":" y se muestra la primera columna.

```
# cut -d":" -f1 /etc/passwd  
root  
daemon  
bin  
(....salida cortada...)
```

Toma como delimitador el signo ":" mostrando la primera y la sexta columna.

```
# cut -d":" -f1,6 /etc/passwd  
root:/root  
daemon:/usr/sbin  
bin:/bin  
sys:/dev  
sync:/bin  
(....salida cortada...)  
[/code]
```

Comando head

El comando **head** permite mostrar las primeras líneas de un archivo. De manera predeterminada mostrará las primeras 10 líneas.

Opciones:

- n [número]** Muestra el número de líneas especificado (anteponiendo un signo “-”, muestra todo excepto esa cantidad de líneas).
- c [número]** Muestra el número de caracteres especificado (número seguido de **k** muestra en kilobytes).

Muestra los 6 primeros caracteres.

```
# head -c6 /etc/passwd
root:x

# head -n 3 /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
```


Comando less

El comando **less** es un visor de archivos de texto. Es ideal para leer archivos largos sin editarlos.

Existe una herramienta más rudimentaria llamada **more**. El programa **less** tiene la ventaja de poder desplazarse en cualquier dirección del documento usando las flechas del teclado.

Muchos comandos de **less** son iguales a los usados en el editor Vi. Por ejemplo, para buscar texto se usa **/**.

```
# less /var/log/messages
```

Comando nl

El comando **nl** permite numerar las líneas de un archivo con diferentes formatos.

El archivo puede tener definidas secciones para poder numerarlas. Con **“\:\:”** se define la sección encabezado, con **“\:”** el cuerpo y con **“\:”** el pie.

-h[opción] Formato encabezado (header)

-f[opción] Formato pie (footer)

-b[opción] Formato cuerpo (body)

Los posibles **formatos** son:

- a** Numera todas las líneas.
- t** Numera las líneas que no están vacías.
- n** No numera las líneas.

A continuación, veamos algunos ejemplos de su utilización.

```
[code]
# nl /etc/passwd
1 root:x:0:0:root:/root:/bin/bash
2 daemon:x:1:1:daemon:/usr/sbin:/bin/sh
3 bin:x:2:2:bin:/bin:/bin/sh
(...salida cortada...)

# cat prueba.txt
\:\:\:
encabezado
\:\:
cuerpo
line1
line2
line3
\:
pie
\:\:\:
encabezado
\:\:
cuerpo
line1
line2
line3
\:
pie
```

Numera todas las líneas incluyendo el encabezado.

```
# nl -h a prueba.txt
1 header

2 line1
3 line2
4 line3

footer

1 header

2 line1
3 line2
4 line3

footer
```

Numera los encabezados, pero no las líneas del cuerpo.

```
# nl -b n -h a prueba.txt
1 header

line1
line2
line3

footer

1 header

line1
line2
line3

footer
```

Numera todas las líneas incluyendo el pie.

```
# nl -f a nl
```

```
header
```

```
1 line1
```

```
2 line2
```

```
3 line3
```

```
4 footer
```

```
header
```

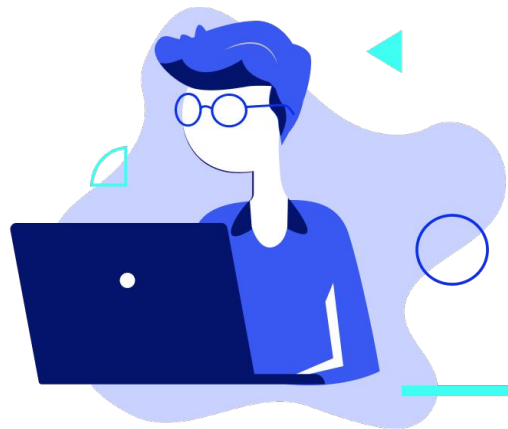
```
1 line1
```

```
2 line2
```

```
3 line3
```

```
4 footer
```

```
[/code]
```



Comando od

El comando **od** (*octal dump*) se utiliza para visualizar el contenido de un archivo en varios formatos (Octal, Hexadecimal, Caracteres visibles y no visibles).

No es muy utilizado.

```
# od -c /etc/passwd  
0000000  r  o  o  t  :  x  :  0  :  0  :  r  o  o  t  :
```

Comando paste

El comando **paste** se utiliza para unir dos o más archivos en columnas en una única salida.

Opciones:

- d [delimitador]** Separa las columnas con el delimitador definido. El predeterminado es tab.
- s** El contenido del archivo en una única línea. Si son más de un archivo separará por línea cada uno.

```
[code]
[rino@restauracion ~]$ cat file1
Casa
Et
Telefono
[rino@restauracion ~]$ cat file2
Telefono
Casa
Telefono
```

Unir ambos archivos

```
[rino@restauracion ~]$ paste file1 file2
Casa      Telefono
Et        Casa
Telefono   Telefono
```

Une ambos archivos, pero una única línea por archivo.

```
[rino@restauracion ~]$ paste -s file1 file2
Casa      Et        Telefono
Telefono   Casa      Telefono
[/code]
```

Une ambos archivos separando el contenido por una arroba.

```
[rino@restauracion ~]$ paste -d'@' file1 file2
Casa@Telefono
Et@Casa
Telefono@Telefono
```

Comando sort

El comando **sort** permite ordenar el contenido de un archivo en varias formas.

Opciones:

- f Ignora la diferencia entre mayúsculas y minúsculas
- r Ordena de manera inversa
- k Define columna para ordenar
- M Ordena por mes (sólo inglés)
- n Ordena de manera numérica
- d Toma letras, números y espacios en blanco

[code]

```
# sort /etc/passwd
```

```
avahi:x:109:122:Avahi mDNS
daemon,,,:/var/run/avahi-daemon:/bin/false
backup:x:34:34:backup:/var/backups:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
clamav:x:111:124::/var/lib/clamav:/bin/false
(...salida cortada...)
```

```
# sort -r /etc/passwd
```

```
www-data:x:33:33:www-data:/var/www:/bin/sh
vde2-net:x:107:118::/var/run/vde2:/bin/false
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
```

```
[rino@restauracion ~]$ ls -la |sort -n -k 5
-rw-rw-r-- 1 rino rino 2408172 Jun 22 22:49
wikimedia 2.pdf
-rw-r--r-- 1 rino rino 2710249 May 26 13:48
ibmsamqt-3.2.3.1-2.i386.rpm
-rw-rw-r-- 1 rino rino 3360412 Jun 10 00:38
vlc-0.7.0-1.i386.rpm
(...salida cortada...)
[/code]
```


Comando split

El comando **split** divide un archivo en varios, ya sea por cantidad de líneas o un tamaño determinado.

Sintaxis

[code]

```
split [opciones] [archivodeentrada] [archivodesalida]
```

Opciones:

-l [número] Cantidad de líneas para cortar (Se puede omitir la "l" Ej: usar -3, para 3 líneas)

-b [número] Cantidad de bytes para cortar

Crearé varios archivos de 10 líneas. El resultado se guardará en passwd-cortadoXX donde XX serán letras incrementándose.

```
# split -l 10 /etc/passwd passwd-cortado
```

[/code]

Comando tac

El comando **tac** pasa el contenido de un archivo a la salida estándar (STDOUT) de manera invertida, el resultado será mostrar en pantalla el contenido de un archivo comenzando por el final.

```
[code]
[root@restauracion ~]# cat tac
uno
dos
tres
[root@restauracion ~]# tac tac
tres
dos
uno
[/code]
```

Comando tail

El comando **tail** nos permite mostrar las últimas líneas de un archivo. De manera predeterminada mostrará las últimas 10 líneas.

Opciones:

- n[numero]** Muestra el número de líneas especificado (anteponiendo un signo "+", muestra comenzando en la línea especificada hasta el final).
- c[numero]** Muestra el número de caracteres especificado (número seguido de k muestra en kilobytes).
- f** Captura el archivo y muestra la salida mientras se va modificando.

Ejemplo:

```
[root@restauracion ~]# tail -n 1 prueba.txt  
tres
```

Comando tr

El comando **tr** traduce cadenas de texto tomadas por STDIN y devolviendo el resultado por STDOUT.

Sintaxis

```
tr string1 string2
```

Opciones

- d** Borra los caracteres en el string1 de la salida.
- s** Borra los caracteres repetidos.
- c** Se usa para completar el string1 o todos los caracteres que no están.

Parámetros

- a-z** Todos los caracteres de la “a” a la “z”.
- ** Un carácter \
- \nnn** Un carácter ASCII con el valor octal nnn.

Caracteres de control

- \a** bell
- \b** backspace
- \f** form feed
- \n** newline
- \r** carriage return
- \t** horizontal tab
- \v** vertical tab

Convierte las minúsculas en mayúsculas.

```
[code]
[root@restauracion ~]# cat file1 | tr a-z A-Z
1 JACK WALLEN
2 JESSICA WALLEN
3 JOHNNY WALLEN
4 JERI WALLEN
[/code]
```

También convierte las minúsculas en mayúsculas.

```
[code]
[root@restauracion ~]# cat file1 | tr '[:lower:]'
'[:upper:]'
1 JACK WALLEN
2 JESSICA WALLEN
3 JOHNNY WALLEN
4 JERI WALLEN
[/code]
```

```
[code]
[root@restauracion ~]# cat prueba.txt
1 Jack      Wallen
2 Jessica   Wallen
3 Johnny    Wallen
4 Jeri      Wallen
[/code]
```

Remueve los espacios repetidos

```
[code]
[root@restauracion ~]# cat file1 | tr -s
':blank: '
1 Jack Wallen
2 Jessica Wallen
3 Johnny Wallen aa
4 Jeri Wallen
[/code]
```

```
[code]
[root@restauracion ~]# cat file1
1      Jack Wallen
2 Jessica   Wallen
3 Johnny   Wallen aa
4 Jeri Wall en
[/code]
```

Quita todos los caracteres no imprimibles, excepto el de nueva línea “**enter**” (*newline*).

```
[code]
[root@restauracion ~]# cat file1 | tr -dc
':print: \n'
1 Jack Wallen
2 Jessica   Wallen
3 Johnny Wallen aa
4 Jeri Wallen
[/code]
```

Comando uniq

El comando **uniq** elimina líneas duplicadas en un archivo.

Opciones

- c Muestra la cantidad de repeticiones
- d Muestra sólo lo que se repite
- u Muestra lo que nunca se repite

```
[code]
[root@restauracion ~]# cat rino.txt
Juan
Pedro
Maria
Rino
Rino
Marina
Pedrito
Hola
Hola
[root@restauracion ~]# uniq rino.txt
Juan
Pedro
Maria
Rino
Marina
Pedrito
Hola
```

Muestra la cantidad de repeticiones.

```
[root@restauracion ~]# uniq -c rino.txt
1 Juan
1 Pedro
1 Maria
2 Rino
1 Marina
1 Pedrito
2 Hola
```

Muestra lo que nunca se repite.

```
[root@restauracion ~]# uniq -u rino
Juan
Pedro
Maria
Marina
Pedrito
[/code]
```

Muestra sólo lo que se repite

```
[root@restauracion ~]# uniq -d rino
Rino
Hola
```


Comando **wc**

El comando **wc** se utiliza para contar líneas, caracteres y palabras.

Opciones:

- w Cantidad de palabras
- l Cantidad de líneas
- c Cantidad de caracteres

Si no se especifican opciones mostrará cantidad de líneas, palabras y caracteres.

```
# wc /etc/passwd
43  64 2099 /etc/passwd
```

Cantidad de líneas

```
# wc -l /etc/passwd
43 /etc/passwd
```

Cantidad de palabras

```
# wc -w /etc/passwd
64 /etc/passwd
```

Cantidad de caracteres

```
# wc -c /etc/passwd
2099 /etc/passwd
```

**¡Sigamos
trabajando!**