

LINUX USER • Línea de Comandos: Cat

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply these licenses to your programs, too.



Examinando ficheros con `cat`, `more`, `less`, `head` y `tail`

LÉELO TODO

Herramientas prácticas como `cat`, `less`, y `head` son ideales para visionar textos. **POR HEIKE JURZIK**

No siempre es necesario arrancar un editor para visionar los ficheros de texto en la consola. Muy a menudo, un simple visor es mucho más rápido. Además, los programas que vemos en el número de este mes tienen mucho más que ofrecer. Si se está buscando un método para concatenar múltiples ficheros en uno solo, `cat` es la herramienta apropiada. Este artículo también examina cómo funcionan para los paginadores `more` y `less`, enseñando los métodos más convenientes para el uso de estas herramientas. Y si se está interesado en coger las primeras o las últimas líneas de un fichero, vemos cómo hacerlo con `head` y `tail`.

El gato con botas

`cat` permite mostrar un fichero de texto en la salida estándar:

```
cat fichero.txt
```

Podemos añadir números de línea a esta salida, podemos escribir `-n`. Si se pretende mostrar con `cat` un fichero binario en lugar de uno de texto, la shell mostrará todo tipo de extraños caracteres. Para resetear la pantalla se escribe `reset` en la línea de comandos (puede que tengas que hacerlo a ciegas).

Pero `cat` puede hacer mucho más. El nombre del programa deriva del verbo inglés "to concatenate" (es decir, concatenar), ya que `cat` realmente lo que

permite es pegar juntos dos o más ficheros. Para concatenar dos ficheros de texto en uno se usan `cat` y el operador `>`:

```
cat fichero1 fichero2 > fichero3
```

Se aconseja ser prudente si ya existe el `fichero3`, ya que `cat` lo reescribirá. Pero puede usarse el operador `>>` para decirle a `cat` que adjunten los dos primeros al tercero, sin sobrescribir su contenido:

```
cat fichero1 fichero2 >> fichero3
```

Más o Menos

Si necesitamos ver un fichero de texto en la shell, pero permitiéndonos mover arriba y abajo y buscar en el fichero, lo mejor es usar un paginador como `more` o `less`. Para ver el fichero página a página, se escribe:

```
more fichero
```

Se presiona la barra espaciadora para bajar una línea y la tecla Enter para subir otra. Una línea de estado en la parte inferior de la página muestra la posición de la página actual a modo de un porcentaje del fichero completo. [B] ("back") lleva a la página anterior, [H] presenta la ayuda y [Q] ("quit") sale del programa. `more` sale automáticamente cuando llega al final del fichero.

Además, `more` posee unos cuantos atajos de teclado más: si se está interesado en el

número de la línea actual, se pulsa la tecla del signo igual (=), la combinación de [Shift + ;] y [F] (:f) presenta el nombre del fichero con el número de la línea. Para buscar en el fichero, se escribe el carácter barra (/) y se introduce la clave de búsqueda.

Si fuera necesario, `more` puede presentar múltiples ficheros de texto simultáneamente. Para hacerlo, simplemente es suficiente pasar todos los nombres de los ficheros a la herramienta cuando ésta se arranca. El paginador indica el nombre de cada fichero:

```
.....:
wget.txt
.....:
```

Puede presionarse `:n` ("next", es decir, siguiente) o `:p` ("previous" o previo) para saltar de un fichero a otro. El comando `:p` lleva al comienzo del fichero actual; si se presiona `:p` por segunda vez, vuelve al fichero de texto previo.

Para editar un fichero no es necesario dejar `more` y arrancar tu editor favorito. En lugar de eso, se presiona [V] mientras se está viendo el fichero. Con ello se arranca por defecto el editor `vi`, pudiéndose entonces modificar el fichero. Si se prefiere usar `more` para invocar a un editor diferente hay que modificar el valor de la variable de entorno `EDITOR`, la cual es evaluada por otros muchos programas. Por ejemplo, si se prefiere trabajar con el editor `Xemacs`, puede definirse la variable con el siguiente comando:

```
export EDITOR=xemacs
```

Cuando se sale del editor, se vuelve a *more*. Para modificar permanentemente el valor de la variable de entorno *EDITOR*, se añade el comando *export* al fichero de configuración *bash*, *~/bashrc*, y se reanaliza el fichero como sigue:

```
source ~/.bashrc
```

Las opciones permiten influir en el modo en el que reacciona *more*; éstas pueden ser especificadas cuando se arranca la herramienta. Por ejemplo, si se necesita editar un fichero con muchas líneas vacías, puede introducirse

```
more -s fichero
```

para decirle al paginador que reduzca múltiples líneas vacías a una sola. Para restringir la salida a 10 líneas en la ventana actual, se añade *-10* como una opción de arranque, y para evitar que *more* se olvide de esta configuración cuando se desplaza a través del fichero se añade la opción *-p*.

Si se desea que *more* use los mismos parámetros de la línea de comandos cada vez que se corre la herramienta, puede establecerse la variable de entorno *MORE* y conservar mucho de lo escrito. Para hacerlo se añade lo siguiente (o algo similar) al fichero de configuración *bash* *~/bashrc*:

```
export MORE=" -10 -p -s"
```

seguidamente, para reanalizar el fichero se corre el siguiente comando:

```
source ~/.bashrc
```

Menos es Más

less es el sucesor de *more*: reconoce los mismos atajos de teclado, aunque añade unas cuantas funciones propias. Si no se desea usar la barra espaciadora y [B] para moverse, pueden emplearse los cursores en su lugar. [G] salta al comienzo del fichero y [Shift + G] al final. *less* no solamente dispone de una búsqueda hacia adelante (/), sino que con el interrogante (?) podemos movernos hacia atrás.

A diferencia de *more*, este paginador no sale automáticamente cuando se alcanza el final del fichero, sino que es preciso salir explícitamente del programa escribiendo [Q]. Mientras que *more* no limpia la presentación cuando sale y presenta el último trozo de texto en la pantalla, *less* la deja

limpia. Este comportamiento puede cambiarse si se desea, estableciendo el parámetro *-X* en la línea de comandos. Al igual que en *more*, puede definirse una variable de entorno para *less* y así especificar los parámetros por defecto. Por ejemplo, para usar la opción *-X* permanentemente hay que añadir la siguiente línea a *~/bashrc*:

```
export LESS=" -X"
```

Como en *more*, *less* analiza la variable *EDITOR* y arranca el editor de texto cuando se pulsa [V]. Los atajos de teclado que se usan para visionar múltiples ficheros son idénticos. Si se pasan múltiples ficheros al paginador, puede presionarse *:n* para moverse al fichero siguiente y *-p* para volver al anterior. A diferencia de *more*, *less* recuerda la posición en el fichero cuando se escribió el comando.

Otra característica práctica del comando *less* es que, si se está trabajando con múltiples ficheros, muestra el nombre del fichero siguiente en la línea de estado de la parte inferior de la página:

```
textfiles.txt (file 1 of 2) ↘
(END) - Next: spellcheck.txt
```

Para evitar tener que presionar *:p* para cambiar al fichero previo, puede establecerse la opción *-e* cuando se arranca *less*. Con esta opción se le dice que cambie automáticamente al fichero siguiente cuando llegue al final del actual y se pulse la barra espaciadora, [Cursor hacia abajo] o [Avance de Página].

less dispone de una característica práctica para controlar los cambios de los ficheros. Si se presiona [f], el paginador muestra las líneas añadidas recientemente. Esto resulta extremadamente práctico mientras se usa *less* para ver ficheros log. La línea de estado presenta *Waiting for data...(interrupt to abort)* para mostrar que *less* está esperando cambios (Figura 1). [Ctrl + C] sale del modo "follow" y vuelve a las funciones de paginador normal.

Paquetes Empaquetados

Los comandos *zless* y *zmore* hacen posible que se vean los ficheros comprimidos

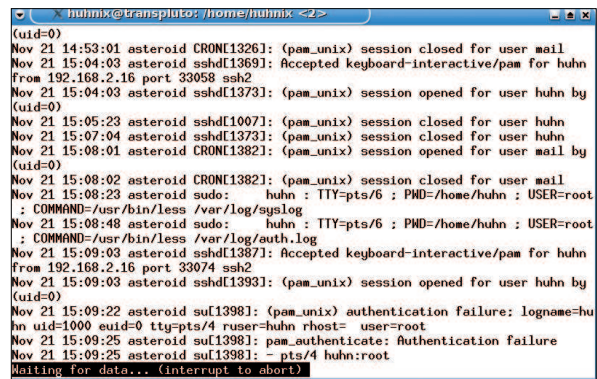


Figura 1: Pulsa [F] para que *less* muestre nuevas líneas de un fichero.

o zipeados. Las herramientas ofrecen los mismos atajos de teclado que los paginadores de los que heredan. En lugar de usar una secuencia de comandos complicada como:

```
zcat Xinerama-HOWTO.gz | less
```

puede escribirse

```
zless Xinerama-HOWTO.gz
```

para leer el fichero comprimido en la pantalla.

De Cabo a Rabo

Si se está interesado en las primeras o en las últimas líneas de un fichero de texto, *head* y *tail* son la herramienta que necesitas:

```
head <I>fichero.txt<I>
```

presenta las primeras diez líneas de un fichero.

tail es el equivalente de *head* para el final de fichero. Al igual que *head*, puede manejar como entrada múltiples nombres de ficheros.

tail posee muchos extras prácticos: *-f* es con toda probabilidad el más comúnmente utilizado. Al igual que ocurriera con el paginador *less*, esta opción permite que se active el modo "follow" y se registre cualquier cambio. Esto es útil para controlar los ficheros log. El comando siguiente:

```
tail -f /var/log/messages
```

envía a *tail* a un bucle infinito, donde controla un fichero en busca de cambios y presenta las últimas entradas. Para abandonar el modo "follow" se presiona [Ctrl + C]. ■