



Victorhck
in the free world
omnia sunt communia!

Victorhck in the free world

Blog personal sobre openSUSE, GNU/Linux y software libre

[Aprender Vim](#) [El programador mediocre](#) [Guía de openSUSE para recién llegados](#) [Contacto](#) [Invítame a una cerveza](#)

El comando uniq de #GNU

Software Libre 21/10/2021 3 minutos

El comando `uniq` sirve para encontrar texto duplicado adyacente en un archivo



Os comandos de **GNU coreutils** son las herramientas básicas de nuestros sistemas **GNU** con kernel Linux similares a las de sistemas Unix, para gestionar desde la línea de comandos el sistema y operaciones con archivos de texto.

Vamos a repasar las opciones y uso del **comando `uniq`** que sirve para **encontrar cadenas de texto duplicado** en un archivo y mostrar diferentes resultados.

Antes de empezar para que el comando `uniq` considere a un texto repetido, este debe estar en líneas adyacentes, consecutivas, juntas o como quieras llamarles.

Es por eso que normalmente se utiliza junto con el **comando `sort`** que lo que hace es justamente ordenar el texto y queda así agrupado por líneas similares. (¿os apetece un próximo artículo sobre ese comando?)

Ejecutando `uniq`

Consideremos que tenemos un archivo que tiene las siguientes 11 líneas. Vemos el contenido con el comando `cat`:

```
1 cat archivo.txt
2 Yo vivo en América
3 Yo vivo en África
4 Yo vivo en Oceanía
5 Yo vivo en Europa
6 Yo Vivo en Asia
7 Yo vivo en la luna
8 Yo vivo en Europa
9 Yo vivo en Marte
10 Yo vivo en Asia
11 Yo vivo en el Polo Norte
12 Yo vivo en europa
```

Podemos ordenarlo con el comando `sort` y ejecutar el comando `uniq` sin más argumentos. Y nos muestra 10 líneas únicas, ya que «Yo vivo en Europa» está duplicado:

```
1 sort archivo.txt | uniq
2 Yo vivo en África
3 Yo vivo en América
4 Yo vivo en Asia
5 Yo Vivo en Asia
6 Yo vivo en el Polo Norte
7 Yo vivo en europa
8 Yo vivo en Europa
9 Yo vivo en la luna
10 Yo vivo en Marte
11 Yo vivo en Oceanía
```

Ignorando mayúsculas y minúsculas

De manera predeterminada, el comando `uniq` de GNU coreutils diferencia entre mayúsculas y minúsculas. Podemos impedir esa diferenciación con la opción `-i` lo que nos deja 8 resultados:

```
1 sort archivo.txt | uniq -i
2 Yo vivo en África
3 Yo vivo en América
4 Yo vivo en Asia
5 Yo vivo en el Polo Norte
6 Yo vivo en europa
7 Yo vivo en la luna
8 Yo vivo en Marte
9 Yo vivo en Oceanía
```

Número de ocurrencias encontradas

El comando `uniq` nos puede mostrar el número de ocurrencias encontradas para cada cadena con la opción `-c` y además con la opción de sin tener en cuenta la diferencia entre mayúsculas y minúsculas:

```
1 sort archivo.txt | uniq -ic
2 1 Yo vivo en África
3 1 Yo vivo en América
4 2 Yo vivo en Asia
5 1 Yo vivo en el Polo Norte
6 3 Yo vivo en europa
7 1 Yo vivo en la luna
8 1 Yo vivo en Marte
9 1 Yo vivo en Oceanía
```

Mostrar solo las líneas duplicadas

Podemos hacer que `uniq` nos muestre únicamente las líneas que ha encontrado duplicadas con la opción `-d`

```
1 sort archivo.txt | uniq -id
2 Yo vivo en Asia
3 Yo vivo en europa
```

Solo muestra una vez cada una de las línea duplicadas, es decir, si hay 5 líneas iguales, solo muestra 1 como muestra. Si quieres ver todas las líneas duplicadas, no solo una por cada grupo utiliza la opción `-D`

```
1 sort archivo.txt | uniq -iD
2 Yo vivo en Asia
3 Yo Vivo en Asia
4 Yo vivo en europa
5 Yo vivo en Europa
6 Yo vivo en Europa
```

Filtrando los caracteres o campos a comprobar

De manera predeterminada `uniq` comprueba la totalidad de las líneas para comparárlas, pero quizás debido a la manera en la que guardamos los datos en nuestro archivo queremos filtrar el texto que queremos que compare.

Compararemos caracteres con la opción `-w` y un número que indicará los primeros caracteres a comparar. Si buscamos las líneas iguales ignorando mayúsculas y minúsculas de los 7 primeros caracteres lo haremos:

```
1 sort archivo.txt | uniq -iw 7
2 Yo vivo en África
```

Solo encuentra una ocurrencia, porque todas empiezan con el mismo texto.

Ahora en vez de decirle que compruebe los primeros 7 caracteres, vamos a decirle que ignore los primeros 11 caracteres y empiece a comparar con el resto mediante la opción `-s`

```
1 sort archivo.txt | uniq -cs 11
2 1 Yo vivo en África
3 1 Yo vivo en América
4 2 Yo vivo en Asia
5 1 Yo vivo en el Polo Norte
6 1 Yo vivo en europa
7 2 Yo vivo en Europa
8 1 Yo vivo en la luna
9 1 Yo vivo en Marte
10 1 Yo vivo en Oceanía
```

En vez de indicar el número de caracteres a ignorar, también podemos hacerlo por campos. Un campo es un grupo de caracteres separados por un espacio en blanco. Con la opción `-f` vamos a indicarle al comando que ignore los primeros 3 campos. Lo que dará un resultado idéntico al anterior:

```
1 sort archivo.txt | uniq -cf 3
2 1 Yo vivo en África
3 1 Yo vivo en América
4 2 Yo vivo en Asia
5 1 Yo vivo en el Polo Norte
6 1 Yo vivo en europa
7 2 Yo vivo en Europa
8 1 Yo vivo en la luna
9 1 Yo vivo en Marte
10 1 Yo vivo en Oceanía
```

Estás serían las maneras de utilizar este comando escrito para GNU por Richard M. Stallman y David MacKenzie. Quizás útil a la hora de incluirlo en algún script.

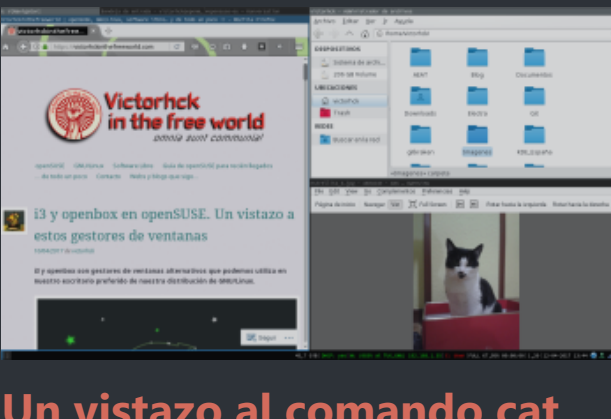
¿Te ha parecido instructivo?



Comparte este artículo:

Correo electrónico Mastodon Mastodon, GNUsocial diaspora, Hubzilla mername Reddit Telegram Pocket Twitter

Relacionado



Un vistazo al comando `cat` de GNU/Linux para mostrar el contenido de archivos en la consola

19/02/2019

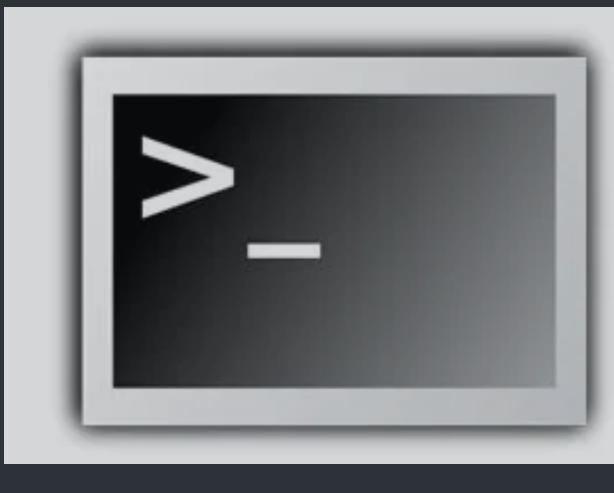
En «GNU/Linux»



El comando `comm` de #GNU

19/10/2021

En «Software Libre»



Creando archivos de texto desde la terminal de #Linux

16/06/2020

En «GNU/Linux»

Etiquetado: GNU, softwarelibre, tutorial

Publicado

21/10/2021

< Dona tu Raspberry Pi para colaborar con las campañas de GNU Health

#openSUSE Tumbleweed revisión de la semana 42 de 2021 >

Me gustaría saber tu opinión. Deja un comentario (Puedes usar Markdown)

Escribe un comentario.

Comentarios

Este sitio utiliza Akismet para reducir el spam. Conoce cómo se procesan los datos de tus comentarios.

Busca en el blog

Buscar ...

Canal Telegram



Estadísticas del sitio

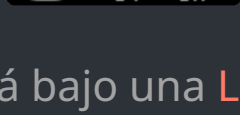
4.439.229 hits

Licencia del blog

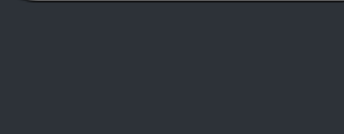


Esta obra está bajo una Licencia Creative Commons Atribución-Compartir Igual 4.0 Internacional.

Licencia del blog



Esta obra está bajo una Licencia Creative Commons Atribución-Compartir Igual 4.0 Internacional.



Busca en el blog

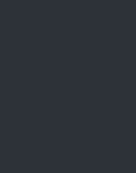
Buscar ...

Comentarios Rebloguear Suscribirse ...

Donar con LiberaPay



Canal Telegram



RSS del blog

RSS - Entradas

Sigue el blog por correo electrónico

Introduce tu dirección de correo electrónico para seguir este Blog y recibir las notificaciones de las nuevas publicaciones.

Dirección de correo electrónico



Únete a otros 1.014 suscriptores