

www.photocase.com

La colección de herramientas diffutils

VIVE LA DIFFÉRENCE

El juego de herramientas diffutils ayuda a comparar ficheros de texto, descubrir y presentar las diferencias entre ficheros e incluso sincronizarlos automáticamente. **POR HEIKE JURZIK**

Diffutils es una colección de útiles herramientas. diff compara dos ficheros de texto, presentando las diferencias en la línea de comandos y almacenando los cambios sugeridos en un fichero patch, el cual puede ser usado para modificar automáticamente un fichero si fuera necesario. Si necesitamos comparar tres ficheros de texto, diff3 será la herramienta apropiada. Sigue leyendo para más detalles y trucos.

Una Perspectiva Diferente

El programa diff compara dos ficheros de texto (o dos ficheros de diferentes directorios) línea por línea y presenta las diferencias, de haberlas, en la línea de comandos. Para descubrir si existen diferencias entre dos ficheros,

podemos arrancar el programa con el parámetro `-q` y los nombres de ambos ficheros:

```
$ diff -q fichero1 fichero2
ficheros fichero1 and
fichero2 ↵
differ
```

Si estamos interesados en la diferencia, omitimos la opción, y diff escribirá lo que encuentra a la shell (Figura 1). Además de las líneas en las que difieren, también podremos ver una ligera referencia críptica que dice lo que has de cambiar para asegurar que los ficheros sean absolutamente idénticos.

Los cambios sugeridos siguen siempre la misma pauta: junto a los números de líneas o rango (número de líneas separadas por comas), veremos

una de estas tres letras, *a*, *c* o *d*, donde *a* representa a “append” (añadir), *c* para “change” (cambiar) y *d* para “delete” (borrar). La siguiente línea:

```
2,3c2,3
```

en la Figura 1 significa que tendremos que modificar las líneas 2 y 3 en el primer o segundo fichero para sincronizarlos. diff usaría una expresión como la que sigue:

```
3a,13,15
```

para decirnos que necesitamos insertar de la línea 13 a la 15 desde el segundo fichero, detrás de la línea 3 en el primer fichero. Si diff dijera

```
4-7,d8
```

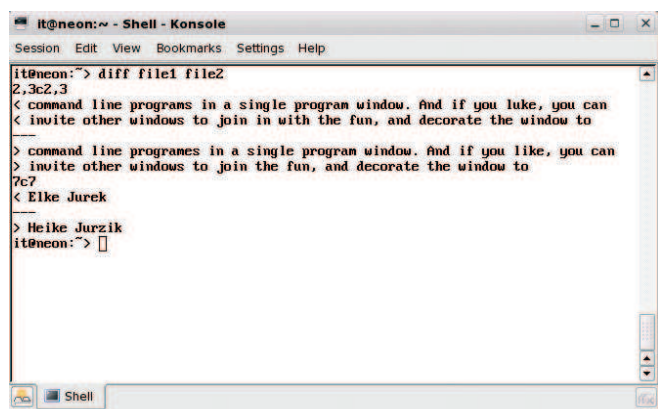


Figura 1: diff puede mostrarte las diferencias entre dos ficheros de texto en la línea de comandos.

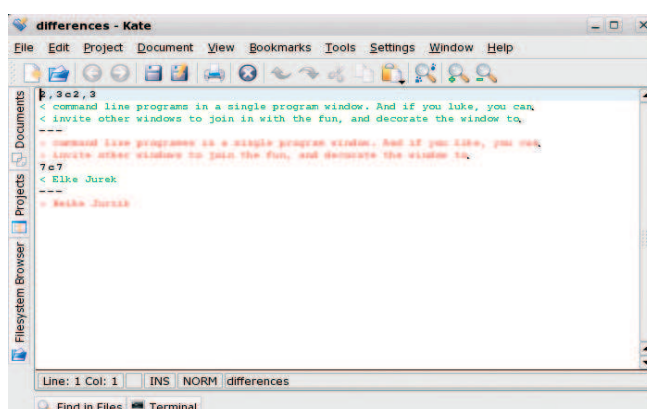
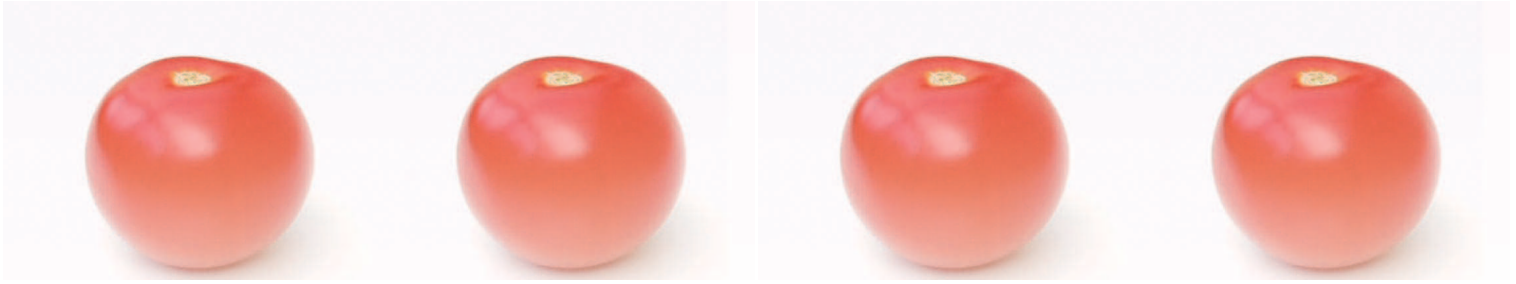


Figura 2: La sintaxis destacada hace que la salida de diff sea más fácil de entender.



significaría que necesitamos borrar de la línea 4 a la 7 en el primer fichero, o bien insertar las líneas después de la línea 8 en el segundo fichero si preferimos modificar este último.

Si los dos blancos o destinos a comparar contienen un importante número de líneas vacías en distintas posiciones, esto puede hinchar considerablemente la salida de diff. Sin embargo, podemos decirle que ignore las líneas en blanco estableciendo otro parámetro, *-B*.

Existen también varias opciones para manipular los blancos. Por ejemplo, puedes decirle que ignore los tabuladores (*-E*), un único tabulador (*-b*), o cualquiera (*-w*). Si no deseamos diferenciar entre una mayúscula y una minúscula, podemos usar la opción *-i*.

Contexto Idóneo

El programa diff usa otros parámetros de la línea de comandos para ayudarnos a mejorar la legibilidad si es complicado entender la salida:

```
diff -c fichero1 fichero2
```

Esto añade algún formato, y por defecto, tres líneas de contexto, a la salida. Si necesitamos más contexto, añadimos el número de líneas con el parámetro *-C*:

```
diff -C 5 fichero1 fichero2
```

diff marca el primer fichero con asteriscos, y el segundo con guiones. La primera cosa que vemos son los nombres de los ficheros y la última fecha de cambio:

```
*** fichero1      2006-10-04
00:11:09.000000000 +0200
-- fichero2      2006-10-04
03:05:03.000000000 +0200
```

La salida restante también se organiza por fichero.

Las líneas con diferencias son identificadas con una exclamación, y no existe marcaje para líneas que coinciden. Las secciones nuevas se marcan con un signo + (más) y las líneas que faltan con un signo - (menos).

Cara a Cara

Otra opción interesante es *-y* que le dice al programa que presente los dos ficheros cara a cara. Se necesita una ventana terminal grande para este formato de salida, e incluso aunque se posea uno, las líneas pueden cortarse. Todo puede llegar a ser más legible si añadimos el parámetro *—suppress-common-lines* para eliminar líneas a juego de la salida. En este escenario, probaremos con la opción *-W* seguida de un valor numérico para

el número de columnas que deseamos ver:

```
diff -y -
suppress-common-lines
-W 70 fichero1 fichero2
```

Parcheado

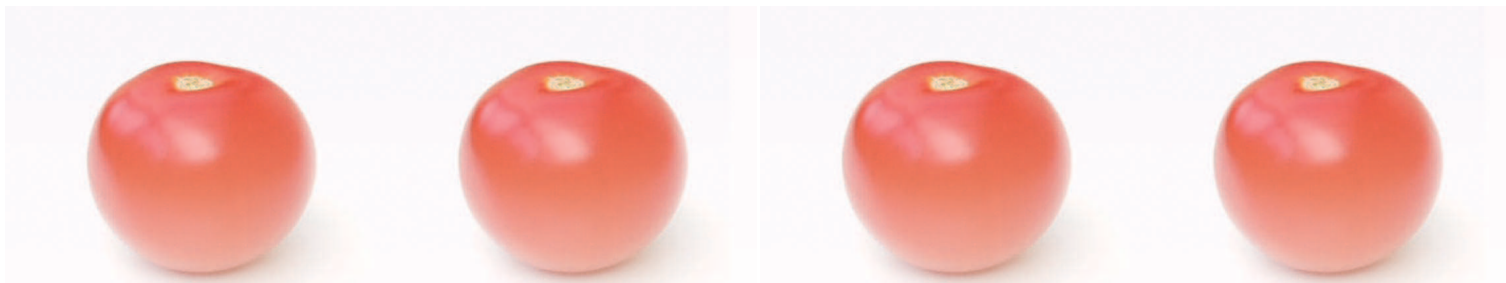
El poder real de diff se hace completamente claro si usamos la herramienta para la sincronización automatizado de ficheros, lo cual evita la necesidad del parcheado manual. diff

Listado 2: La Herramienta diff3

```
01 $ diff3 fichero1 fichero2
    fichero3
02 ====2
03 1:6c
04 3:6c
05 _____
06
    _____
07
08 2:6c
09     Elke Jurek
10 [...]
11 ====
12 1:14,16c
13     Heike Jurzik
14 2:10,14c
15 El programa screen nos da
    la posibilidad de ejecutar
    múltiples consolas
    virtuales
16 [...]
17 3:14,16c
18 Heike Jurzig
19 [...]
20 ====3
21 1:257a
22 2:255a
23 3:255,256c
```

Listado 1: Salida diff

```
01 *** fichero1      2006-10-04 U
02     00:11:09.000000000 +0200 U
03 -- fichero2      2006-10-04
04     03:05:03.000000000 +0200
05 *****
06 *** 3,17 ****
07 [...]
08 !   Heike Jurzik
09 [...]
10 -- 3,9  —
11 [...]
12 !   Elke Jurek
13 [...]
14 *****
15 *** 25,31 ****
16 [...]
17 -   Montando la escalera
18 [...]
19 -- 252,256  —
20 [...]
21 +
22 + ¡Aquí hay texto!
```



dispone de algunas opciones que manipulan lo siguiente:

```
diff -u fichero1 fichero2 > patching
```

Esto crea una salida en un formato unificado, y el fichero resultante es referido como un fichero de parcheado. Ambos ficheros se listan uno debajo del otro en la salida, junto con las últimas fechas de cambio y tres líneas de contexto para mejorar la legibilidad. Puedes especificar un número de líneas diferente. El siguiente comando le dice a diff que añada a la salida diez líneas iguales:

```
diff -U 10 fichero1 fichero2 > patching
```

Es hora de sacar aguja e hilo y ponerse a coser. La herramienta para hacer esto en la línea de comandos se denomina acertadamente patch. Por razones de seguridad, especifica la opción `-b` para decirle al programa que guarde un backup del fichero original. La opción `<` señala al fichero parcheado:

```
$ patch -b < patching
patching fichero fichero1
```

Adicionalmente, se almacena un backup del original del *fichero 1* con el sufijo *.orig*.

Para Todo

Si ejecutamos diff con los nombres de dos directorios, el programa comparará cada fichero que exista en ambos directorios. Como alternativa, se le pueden pasar los nombres del fichero requerido y del directorio a diff. La herramienta buscará en los directorios el fichero especificado, y comparará los dos ficheros si la búsqueda tuvo éxito.

De manera predeterminada, diff no

creará ninguna salida si dos ficheros son idénticos, pero si necesitamos un informe completo, usaremos el parámetro `-s`:

```
$ diff -s dir1 dir2
ficheros dir1/fichero1 and dir2/fichero1 are identical
diff -s dir1/fichero2 dir2/fichero2
6c6
< Elke Jurek
--
> Elke Jurzig
Only patch in dir1:
```

Si queremos incluir subdirectorios, podremos hacerlo con la opción `-r`, la cual le dice a diff que busque recursivamente a través del árbol principal. También podemos excluir ficheros desde la comparación. Lo conseguiremos con la etiqueta `-x`, y se especifican los candidatos para la exclusión usando comillas.

Para excluir todos los ficheros que acaban en *.orig*, debemos hacer lo siguiente:

```
diff -s -x "*.orig" dir1 dir2
```

Lo Bueno, si Breve, Tres Veces Bueno

diffutils incluye un programa que puede comparar tres ficheros. La herramienta, acertadamente conocida como diff3, se ejecuta contra tres ficheros de texto (Listado 2).

La salida de diff3 es distinta de la normal que ofrece diff. La herramienta diff3 usa secciones diferentes, separadas por signos iguales, mostrando dónde son diferentes los ficheros unos de otros. El signo igual sin un número indica que los tres ficheros son distintos; el número 1, 2 ó 3 nos indica qué fichero es diferente de los otros.

Al igual que diff, proporciona los números de línea y rangos, y dice si

necesitas añadir (*a*) o cambiar (*c*) algo.

Común Denominador

diff3 guarda más ases en la manga. Permite suponer que dos usuarios han modificado el mismo fichero de texto; en este caso, podemos usar diff3 para mezclar los ficheros. Para ello, especificamos el fichero original como el segundo argumento en la línea de comandos de diff3, y los dos ficheros modificados como los argumentos primero y tercero:

```
diff3 -m fichero1 original fichero2 > salida
```

El parámetro `-m` es para “mezclar”, y la salida aquí se redirige a través del operador `>` a un fichero denominado *salid*. Si la acción causa conflictos sin solución, el fichero de salida señalará hacia las áreas problemáticas:

```
<<<<<< fichero1
||||||| original
```

En este caso, modificaremos manualmente los ficheros con nuestro editor favorito. ■

CONSEJO

Si la salida diff se desplaza fuera de la pantalla, podemos dirigirla a las búsquedas *less* o *more* de la siguiente manera: `diff fichero1 fichero2 | less`. Alternativamente, es posible que prefiramos redirigir la salida a un fichero, así: `diff fichero1 fichero2 > diferencias`. Si luego abrimos el fichero en un editor de texto que soporta sintaxis destacada, las diferencias serán destacadas en diferentes colores para que sea más fácil su visualización (Figura 2).