

# Uso básico de SED

## Qué es sed

SED (Stream EDitor) es un editor de flujos y ficheros de forma no interactiva. Permite modificar el contenido de las diferentes líneas de un fichero en base a una serie de comandos o un fichero de comandos (-f fichero\_comandos).

Sed recibe por stdin (o vía fichero) una serie de líneas para manipular, y aplica a cada una de ellas los comandos que le especifiquemos a todas ellas, a un rango de las mismas, o a las que cumplan alguna condición.

Por ejemplo:

```
Sustituir apariciones de cadena1 por cadena2 en todo el fichero:  
# sed 's/cadena1/cadena2/g' fichero > fichero2
```

```
Sustituir apariciones de cadena1 por cadena2 en las líneas 1 a 10:  
# comando | sed '1,10 s/cadena1/cadena2/g'
```

```
Eliminar las líneas 2 a 7 del fichero  
# sed '2,7 d' fichero > fichero2
```

```
Buscar un determinado patrón en un fichero:  
# sed -e '/cadena/ !d' fichero
```

```
Buscar AAA o BBB o CCC en la misma línea:  
# sed '/AAA/!d; /BBB/!d; /CCC/!d' fichero
```

```
Buscar AAA y BBB y CCC:  
# sed '/AAA.*BBB.*CCC/!d' fichero
```

```
Buscar AAA o BBB o CCC (en diferentes líneas, o grep -E):  
# sed -e '/AAA/b' -e '/BBB/b' -e '/CCC/b' -e d  
# gsed '/AAA\|BBB\|CCC/!d'
```

## Formato de uso

El formato básico de uso de sed es:

```
# sed [-ns] '[direccion] instruccion argumentos'
```

Donde:

- **[direccion]** es opcional, siendo un número de línea (N), rango de números de línea (N,M) o búsqueda de regexp (/cadena/) indicando el ámbito de actuación de las instrucciones. Si no se especifica [direccion], se actúa sobre todas las líneas del flujo.
- **Instruccion** puede ser:
  - **i** = Insertar línea antes de la línea actual.
  - **a** = Insertar línea después de la línea actual.

- **c** = Cambiar línea actual.
  - **d** = Borrar línea actual.
  - **p** = Imprimir línea actual en stdout.
  - **s** = Sustituir cadena en línea actual.
  - **r fichero** = Añadir contenido de "fichero" a la línea actual.
  - **w fichero** = Escribir salida a un fichero.
  - **!** = Aplicar instrucción a las líneas no seleccionadas por la condición.
  - **q** = Finalizar procesamiento del fichero.
- **-n**: No mostrar por stdout las líneas que están siendo procesadas.
  - **-s**: Tratar todos los ficheros entrantes como flujos separados.

## Ejemplos de sustitución

Reemplazar cadenas:

```
# sed 's/^Host solaris8/Host solaris9/g' fichero > fichero2
```

Reemplazar cadenas sólo en las líneas que contienen una cadena:

```
# sed '/cadena_a_buscar/ s/vieja/nueva/g' fichero > fichero2
```

Reemplazar cadenas sólo en determinadas líneas:

```
# sed '5,6 s/vieja/nueva/g' fichero > fichero2
```

Reemplazar múltiples cadenas (A o B):

```
# sed 's/cadenasrc1|cadenasrc2/cadena_nueva/g'
```

Sustituir líneas completas (c) que cumplan o no un patrón:

```
# echo -e "línea 1\nlínea 2" | sed '/1/ cPrueba'
Prueba
línea 2
# echo -e "línea X 1\nlínea 2" | sed '/1/ !cPrueba'
línea 1
Prueba
```

## Ejemplos de Inserción

Insertar 3 espacios en blanco al principio de cada línea:

```
# sed 's/^/   /' fichero
```

Añadir una línea antes o después del final de un fichero (\$=última línea):

```
# sed -e '$i Prueba' fichero > fichero2
# sed -e '$a Prueba' fichero > fichero2
```

Insertar una línea en blanco antes de cada línea que cumpla una regex:

```
# sed '/cadena/{x;p;x;}' fichero
```

Insertar una línea en blanco detrás de cada línea que cumpla una regex:

```
# sed '/cadena/G' fichero
```

Insertar una línea en blanco antes y después de cada línea que cumpla una regex:

```
# sed '/cadena/{x;p;x;G;}' fichero

Insertar una línea en blanco cada 5 líneas:
# sed 'n;n;n;n;G;' fichero

Insertar número de línea antes de cada línea:
# sed = filename | sed 'N;s/\n/\t/' fichero

Insertar número de línea, pero sólo si no está en blanco:
# sed '/./=' fichero | sed '/./N; s/\n/ /'

Si una línea acaba en \ (backslash) unirla con la siguiente:
# sed -e :a -e '/\\$/N; s/\\n//; ta' fichero
```

## Ejemplos de Selección/Visualización

Ver las primeras 10 líneas de un fichero:

```
# sed 10q
```

Ver las últimas 10 líneas de un fichero:

```
# sed -e :a -e '$q;N;11,$D;ba'
```

Ver un rango concreto de líneas de un fichero:

```
# cat -n fich2 | sed -n '2,3 p'
```

```
2      línea 2
```

```
3      línea 3
```

(Con cat -n, el comando cat agrega el número de línea).

(Con sed -n, no se imprime nada por pantalla, salvo 2,3p).

Ver un rango concreto de líneas de varios ficheros:

```
# sed '2,3 p' *
```

```
línea 2 fichero 1
```

```
línea 3 fichero 1
```

```
línea 2 fichero 2
```

```
línea 3 fichero 2
```

(-s = no tratar como flujo sino como ficheros separados)

Sólo mostrar la primera línea de un fichero:

```
# sed -n '1p' fichero > fichero2.txt
```

No mostrar la primera línea de un fichero:

```
# sed '1d' fichero > fichero2.txt
```

Mostrar la primera/última línea de un fichero:

```
# sed -n '1p' fichero
```

```
# sed -n '$p' fichero
```

Imprimir las líneas que no hagan match con una regexp (grep -v):

```
# sed '/regexp/!d' fichero
```

```
# sed -n '/regexp/p' fichero
```

Mostrar la línea que sigue inmediatamente a una regexp:

```
# sed -n '/regexp/{n;p;}' fichero
```

Mostrar desde una expresión regular hasta el final de fichero:

```
# sed -n '/regexp/, $p' fichero
```

Imprimir líneas de 60 caracteres o más:

```
# sed -n '/^.\{60\}/p' fichero
```

```
Imprimir líneas de 60 caracteres o menos:  
# sed -n '/^.\{65\}/!p' fichero  
# sed '/^.\{65\}/d' fichero
```

## Ejemplos de Borrado

Eliminar un rango concreto de líneas de un fichero:  
# sed '2,4 d' fichero > fichero2.txt

Eliminar todas las líneas de un fichero excepto un rango:  
# sed '2,4 !d' fichero > fichero2.txt

Eliminar la última línea de un fichero  
# sed '\$d' fichero

Eliminar desde una línea concreta hasta el final del fichero:  
# sed '2,\$d' fichero > fichero2.txt

Eliminar las líneas que contentan una cadena:  
# sed '/cadena/ d' fichero > fichero2.txt  
# sed '/^cadena/ d' fichero > fichero2.txt  
# sed '/^cadena\$/ d' fichero > fichero2.txt

Eliminar líneas en blanco (variación del anterior):  
# comando | sed '/^\$/ d'  
# sed '/^\$/d' fichero > fichero2.txt

Eliminar múltiples líneas en blanco consecutivas dejando sólo 1:  
# sed '/./,/^\$/!d' fichero

Añadir una línea después de cada línea:  
# echo -e "línea 1\nlínea 2" | sed 'aPrueba'  
línea 1  
Prueba  
línea 2  
Prueba

Eliminar espacios al principio de línea:  
# sed 's/^ \*//g' fichero

Eliminar todos los espacios que haya al final de cada línea:  
# sed 's/ \*\$//' fichero

Eliminar espacios sobrantes a principio y final de línea, o ambos:  
# sed 's/^[ \t]\*\$//' fichero  
# sed 's/[ \t]\*\$//' fichero  
# sed 's/^[ \t]\*\$//;s/[ \t]\*\$//' fichero

Eliminar tags HTML:  
# sed -e :a -e 's/<[^>]\*>//g;/</N;//ba' fichero

Borrar líneas duplicadas no consecutivas de un fichero:  
# sed -n 'G; s/\n/&&; /^([ -~]\*\n\).\*\n\1/d; s/\n//; h; P' fichero

Eliminar líneas en blanco y comentarios bash:  
# comando | sed '/^\$/ d'  
# sed '/^\$/d; / \*#/d' fichero > fichero2.txt

## Uso de salida selectiva

Salir a nuestra voluntad antes de acabar el fichero:

```
# sed -e '/uno/ s/uno/1/' -e '/salir/ q' fichero > fichero2.txt
# sed 10q fichero
# sed q fichero
```

Equivalencia de -e con ";":

```
# sed -e '/AAA/b' -e '/BBB/b' -e 'd' == sed '/AAA/b;/BBB/b;d'
```

Usar 'q' apropiadamente reduce tiempo de procesamiento:

```
# sed -n '10,20p' fichero
# sed -n '21q;10,20p' fichero -> Más rápido que el anterior.
```

Conversión de CRLF de DOS a formato UNIX (LF):

```
# sed 's/.$//' fichero
```

Conversión de LF de UNIX a formato DOS (CRLF):

```
# sed 's/$'"/`echo \\`r`/" fichero
```

Obtener el Subject de un correo, pero sin cadena "Subject: ":

```
# sed '/^Subject: */!d; s///;q' fichero
```

Imprimir párrafo (cadenas entre 2 líneas en blanco) si contiene XXX:

```
# sed -e '/./{H;$!d;}' -e 'x;/XXX/!d;' fichero
```

Imprimir párrafo si contiene (1) XXX y ZZZ o bien (2) XXX o ZZZ :

```
# sed -e '/./{H;$!d;}' -e 'x;/XXX/!d;/ZZZ/!d'
```

```
# sed -e '/./{H;$!d;}' -e 'x;/XXX/b' -e '/ZZZ/b' -e d
```