Supongamos que tenemos una partición de Dagtos en la que hemos borrado muchos archivos recientemente y nos interesa que todo ese «espacio libre» esté lleno de ceros, y no de los datos originales que se encontraban en el disco antes de ser borrados. Podemos realizar esta tarea de diversas formas:

CAT

Utilizando el dispositivo /dev/zero podemos crear un archivo que ocupe todo el espacio libre que tiene disponible la partición, ejecutando como root:

```
cat /dev/zero > /Particiones/Datos/ArchivoConCeros.tmp
```

Cuando la CLI nos notifique que la partición se ha quedado sin espacio libre para seguir creando el archivo, deberemos sincronizar los datos de la memoria con los datos del disco, ejecutando:

```
sync
```

Y, finalmente, borraremos el archivo para volver a liberar ese espacio libre.

```
rm -f /Particiones/Datos/ArchivoConCeros.tmp
```

Toda la operación en una única línea:

```
cat /dev/zero > /Particiones/Datos/ArchivoConCeros.tmp; sync; rm -f
/Particiones/Datos/ArchivoConCeros.tmp
```

.

SFILL

sfill es un comando que se encuentra dentro del paquete **secure-delete** del repositorio Universe de Ubuntu, o el repositorio contrib de Debian. Hace exactamente lo mismo que hicimos con el comando cat pero de forma automática y sin indicarle el nombre del archivo (sólo indicamos el punto de montaje de la partición). Lo ejecutamos de la siguiente forma:

```
sfill -v -z /Particiones/Datos/
```

Hay otros parámetros que se le pueden pasar:

-i borra todo el espacio libre de los inodos, no el espacio libre de disco.

-I borra todo el espacio de disco, no el espacio libre de los inodos.

ZEROFREE (Sólo para ext2, ext3 y ext4)

Parecido al comando anterior, lo ejecutamos con:

```
zerofree -v /Particiones/Datos/
```

TRIM

Si el disco al que llenaremos su espacio libre con ceros es una unidad de estado sólido compatible con la tecnología TRIM, podemos utilizar el comando fstrim para hacer lo mismo que arriba, de la siguiente foma:

```
fstrim -v /Particiones/Datos/
```

.