

# Gestión de Dispositivos de Almacenamiento

## Sumario

- 1 Particionado
- 2 Utilizando como base para el SA un disco duro virtual
- 3 Utilizando como base para el SA un archivo en un SA existente
  - ◆ 3.1 Creación de un dispositivo de bloques virtual sobre un archivo
  - ◆ 3.2 Creación de las particiones con fdisk
- 4 Creación de particiones con cfdisk
- 5 El comando sfdisk

## Particionado

El primer paso para la creación de sistemas de archivos es la preparación de los dispositivos sobre los que se desplegarán los mismos.

Entendemos por dispositivo todo aquel soporte de almacenamiento de datos compatible con el kernel de Linux. Empezaremos nuestro recorrido partiendo de la base de que existe un dispositivo conectado a la máquina listo para ser utilizado. Podría tratarse, por ejemplo, de alguno de los siguientes:

1. Disco virtual ¿vacío? conectado a una máquina virtual
2. Archivo en un sistema de archivos existente sobre el cual podríamos crear un dispositivo de bloques virtual
3. Volumen de almacenamiento lógico LVM
4. Volumen constituido por un array de discos de tipo RAID

para las opciones 3 y 4 necesitaríamos conocer las herramientas de gestión correspondientes, por ejemplo para gestionar un RAID software podríamos utilizar mdadm, y para LVM las versiones modernas de GNU/Linux disponen de herramientas para ello. Sin embargo, vamos a centrarnos en una de las dos primeras, 1 y 2, ya que más adelante veremos aspectos relacionados con RAID y gestión lógica de volúmenes al hablar del sistema de archivos ZFS.

A partir de ahora nos referiremos al sistema de archivos como **SA**

## Utilizando como base para el SA un disco duro virtual

Esta es la opción más sencilla y habitual, conectar un dispositivo de almacenamiento a nuestra máquina que utilizaremos como base para la creación de particiones y del SA.

El procedimiento consiste en, desde el panel de gestión de nuestra máquina virtual (en Virtualbox o virt-manager) conectar un nuevo dispositivo a la máquina virtual.

Una vez arrancado el sistema nuestro dispositivo será representado por el kernel como un dispositivo especial de bloques en /dev, como por ejemplo **/dev/sdb** o **/dev/sdc**

Para ver los dispositivos de bloques en uso podemos ejecutar el comando

```
lsblk
```

La salida de este comando mostrará los dispositivos de bloques (es decir dispositivos de almacenamiento) disponibles en el sistema y, en caso de estar divididos en particiones, las correspondientes particiones)

## Utilizando como base para el SA un archivo en un SA existente

Otra opción muy cómoda y nada invasiva con un sistema operativo existente es la utilización de un archivo como base para la creación de un dispositivo de bloques virtual sobre el que crearemos nuestro SA

## Creación de un dispositivo de bloques virtual sobre un archivo

Comenzamos creando el archivo. Siempre como usuario root ejecutamos los comandos:

```
dd if=/dev/zero of=./fsraw bs=1M count=10000
```

el comando anterior crea un archivo de nombre **fsraw**, en el directorio de trabajo de 10GB de capacidad (count=10000 bloques de bs=1Mega byte de tamaño)

En este punto ya podríamos crear el SA directamente sobre el archivo anterior, sin embargo vamos a ejecutar un paso adicional consistente en la creación de un dispositivo de bloques asociado al archivo, para a continuación crear el SA sobre ese dispositivo.

Ejecutamos el comando:

```
losetup
```

Este comando, que debería mostrar una salida vacía, muestra los dispositivos **/dev/loopX** (que van desde loop0 hasta loop8) que están siendo utilizados. Suponiendo que **/dev/loop0** no está siendo utilizado, en este caso no deberíamos ver ese dispositivo listado en la salida del comando **losetup**, vamos a utilizarlo como dispositivo de bloques asociado al archivo **fsraw** creando con anterioridad. La opción -P permite definir particiones dentro del dispositivo de bloques

Para ello ejecutamos:

```
losetup -P /dev/loop0 ./fsraw
```

Tras la ejecución de los pasos anteriores disponemos del dispositivo de bloques **/dev/loop0** como base para continuar con los procedimientos que siguen. Si volvemos a ejecutar el comando anterior

```
losetup
```

veremos como ahora sí aparece el dispositivo **/dev/loop0** en el listado

Una vez dispongamos del dispositivo procederemos a crear las particiones necesarias para albergar los SA

En el siguiente vídeo puede verse el proceso de creación del dispositivo virtual de bloques

```
root@debian:~# dd if=/dev/zero of=./fsraw count=10000 bs=1M
```



## Creación de las particiones con fdisk

fdisk es una herramienta de gestión de particiones para dispositivos de bloques basados en el formato MBR (o tipo d.o.s.), si queremos utilizar un formato de particionado basado en GPT necesitaríamos utilizar otra herramienta, como **cfdisk** o **gdisk**.

Usaremos esta herramienta para crear las particiones en el dispositivo.

Supongamos que disponemos de un dispositivo **/dev/sdb** (en caso de utilizar /dev/loop0 sustituirlo por éste en los comandos correspondientes).

Vamos ahora a abrir una terminal. Una vez en ella, como usuario root, para ello escribimos el comando:

```
su -
```

**NOTA:** En caso de no tener habilitado el usuario root (algo habitual en Ubuntu) ejecutamos el comando

```
sudo -i
```

Una vez en la sesión de root ejecutamos a continuación el comando

```
fdisk
```

La ejecución del comando nos mostrará un listado con las opciones de uso más importantes. **Este comando sirve para crear y gestionar particiones de un dispositivo** de almacenamiento. Se utiliza en reemplazo del comando **parted** que sirve para gestionar las particiones de un disco duro. **Con fdisk podemos gestionar las particiones de un modo más cómodo y menos propenso a errores que parted**

Ejecutad el comando

```
fdisk -l
```

para ver un listado de los dispositivos y particiones en él.

**IMPORTANTE:** Identificad cual de los dispositivos corresponden con el disco que habéis insertado para la práctica (en nuestro caso **/dev/sdb** o **/dev/loop0**), es importante comprobar siempre esto para evitar equivocaciones y acciones potencialmente destructivas)

Para la identificación propuesta podemos utilizar el comando

```
lsblk
```

el cual nos muestra una visualización en forma de árbol de los dispositivos de bloques, además de información adicional sobre los mismos, como el espacio disponible, los números **MAJ (major)** y **MIN (minor)** del dispositivo, el tipo y el punto de montaje

Vamos a ejecutar el comando para el disco que hemos conectado a la máquina (también puede hacerse utilizando el dispositivo de bloques /dev/loop0):

```
fdisk /dev/sdb
```

Ahora entrará en modo interactivo para recibir vuestras órdenes. Antes de nada invocad la orden de ayuda (**pulsando m**)

A continuación os mostrará las posibles acciones y su uso. Guiándoos por las opciones que aparecen en el listado, realizad los siguientes pasos: (Adjuntad capturas de pantalla)

1. Imprimir por pantalla la tabla de particiones el disco
2. Eliminar la partición anteriormente creada
3. Crear una nueva partición de la mitad de tamaño del disco
4. Crear otra partición con el espacio restante
5. Establecer el código identificador del sistema de archivos que va a albergar la partición (ver nota más abajo)
6. Verificar la tabla de particiones

7. Imprime de nuevo la tabla de particiones

8. Guarda los cambios en disco y sal (Sino se ejecuta este paso no guardará los cambios)

Indicando el número de alguna de las acciones anteriores entramos en la funcionalidad correspondiente de fdisk, el cual nos asistirá con mensajes de texto e indicaciones de como completar la tarea seleccionada.

**IMPORTANTE:** Es necesario establecer el **identificador del sistema de archivos en la partición** para luego poder crear el sistema de archivos correspondiente. fdisk utiliza para eso la opción **t**. Por tanto, si por ejemplo vamos a crear un sistema de archivos NTFS en una partición deberemos previamente establecer el código identificador del sistema de archivos con la opción **t** de fdisk, en este caso el código sería **7**. Para ver los códigos de sistemas de archivos podemos utilizar la opción **l** de fdisk.

En el siguiente vídeo puede verse como crear particiones en un dispositivo utilizando la utilidad fdisk

Bienvenido a fdisk (util-linux 2.29.2).

Los cambios solo permanecerán en la memoria, hasta que decida escribirlos.  
Tenga cuidado antes de utilizar la orden de escritura.

El dispositivo no contiene una tabla de particiones reconocida.

Se ha creado una nueva etiqueta de disco DOS con el identificador de disco 0x9a51d414.

Orden (m para obtener ayuda): n

Tipo de partición

p primaria (0 primaria(s), 0 extendida(s), 4 libre(s))

e extendida (contenedor para particiones lógicas)

Seleccionar (valor predeterminado p)

Se está utilizando la respuesta predeterminada p.

Número de partición (1-4, valor predeterminado 1):

Primer sector (2048-20479999, valor predeterminado 2048):

Último sector, +sectores o +tamaño{K,M,G,T,P} (2048-20479999, valor predeterminado 20479999): +4G

Crea una nueva partición 1 de tipo 'Linux' y de tamaño 4 GiB.

Orden (m para obtener ayuda):

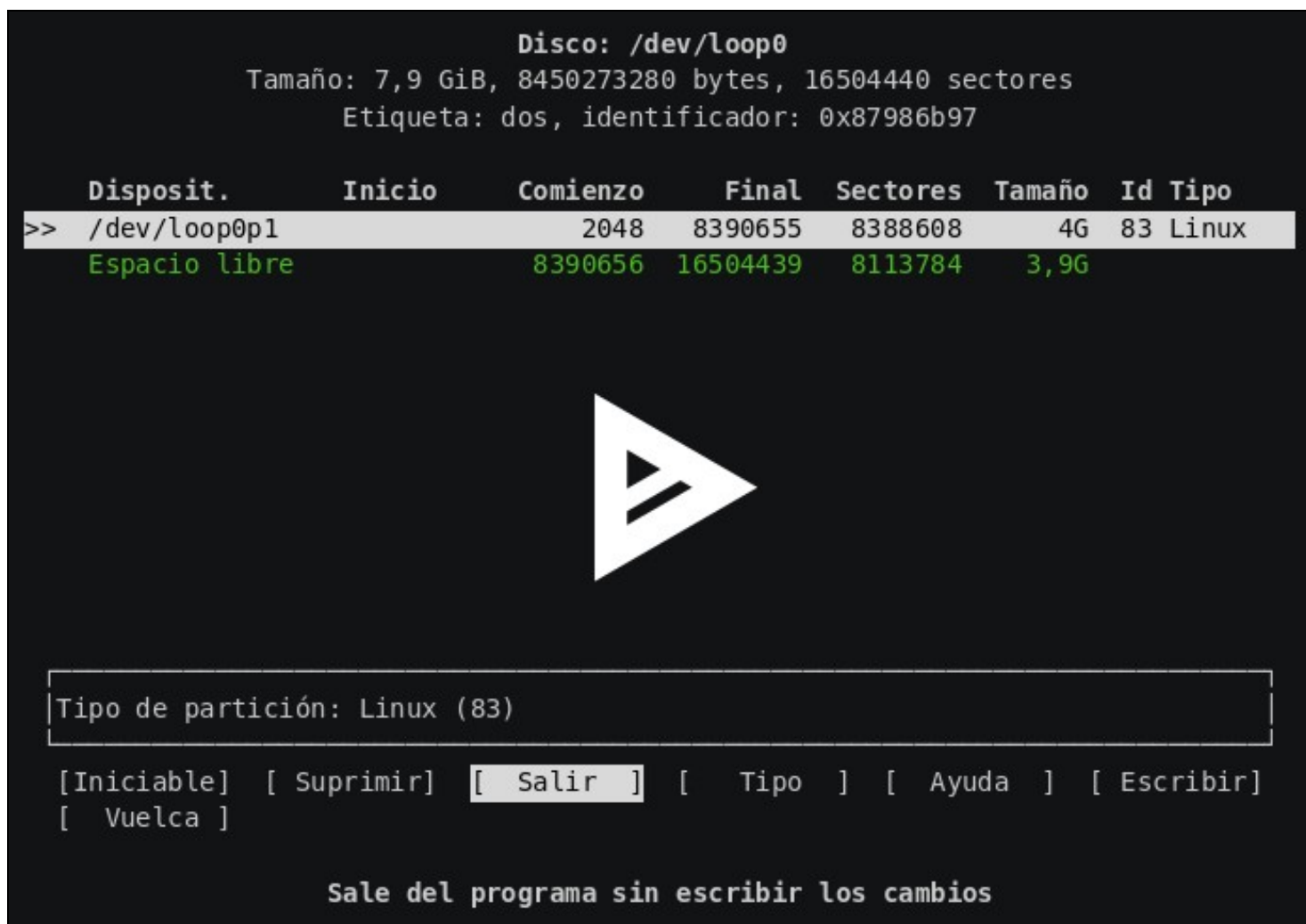
## Creación de particiones con cfdisk

El comando fdisk anterior permite gestionar las particiones de discos MBR. Existe otra herramienta, que además dispone de una interfaz ncurses (en modo texto pero con interfaz más utilizable), que soporta las opciones de creación del comando fdisk y permite también crear particiones en discos de formato GPT. Se trata de la utilidad **cfdisk**

El siguiente comando permite definir y gestionar particiones con la herramienta cfdisk. Siempre como root:

```
cfdisk /dev/sdb
```

En el siguiente vídeo puede verse como crear particiones en un dispositivo utilizando la utilidad cfdisk



## El comando sfdisk

El comando sfdisk se utiliza para comprobar la configuración de particiones de un dispositivo de bloques. Un uso muy común de este comando es copiar a un archivo de texto la tabla de particiones de un dispositivo. Ejecutemos:

```
sfdisk -l /dev/sdb
```

El comando anterior lista todas las particiones del dispositivo /dev/sdb

El comando:

```
sfdisk -l /dev/sda > /tmp/partable
```

Escribe la configuración de la tabla de particiones a un archivo. Este archivo puede ser utilizado luego para recuperar la información de la tabla de particiones

Otro uso muy interesante es poder copiar tablas de particiones de unos dispositivos a otros, por ejemplo:

```
sfdisk -d /dev/sda | sfdisk /dev/sdb
```

copiará la tabla de particiones del dispositivo /dev/sda en /dev/sdb

Veamos el proceso en el siguiente vídeo

```
root@debian:~# sfdisk -d /dev/
```





*Copiar tabla de Particiones con sfdisk*

[Volver](#)

JavierFP 18:31 12 nov 2018 (CET)