Tema 7

Información del sistema. Dispositivos de almacenamiento

Administración de Sistemas Operativos

Índice

[Esquema 3](#_Toc170140067)

[Material de estudio 4](#_Toc170140068)

[7.1. Introducción y objetivos. 4](#_Toc170140069)

[7.2. Particiones y discos. 5](#_Toc170140070)

[7.3. Administración de particiones y Esquemas de particionamiento. 6](#_Toc170140071)

[7.4. Sistemas de ficheros. 15](#_Toc170140072)

[7.5. Administrador de volúmenes lógicos. 19](#_Toc170140073)

[7.6. Referencias Bibliográficas. 24](#_Toc170140074)

[A fondo 25](#_Toc170140075)

[Entrenamientos 26](#_Toc170140076)

[Test 31](#_Toc170140077)

Esquema

Tabla

Descripción generada automáticamente

Material de estudio

7.1. Introducción y objetivos.

En esta unidad vamos a entender todas las posibilidades que se nos presentan para gestionar de una forma eficiente los dispositivos de almacenamiento desde el punto de vista del software.

Por otro lado, profundizaremos en los Volúmenes Lógicos y sus herramientas de administración (LVM) que permite gestionar y administrar el almacenamiento en sistemas Linux.

Con LVM, puedes crear volúmenes lógicos (LV) y grupos de volúmenes lógicos (VG) que utilizan varios discos físicos.

Los objetivos que se pretenden conseguir al final de esta unidad son:

* **Comprender que es una partición, que tipos hay y como se crean**: Entenderemos que es una partición, cual es su razón de ser, cuantos tipos diferentes hay y para que se utiliza cada una de ellas y como crearlas sin utilizar herramientas con interfaz gráfica
* **Entender que es el sistema de ficheros y que tipos hay**: Comprender que es el sistema de archivos, la importancia que tiene y los diferentes tipos que hay con sus ventajas y desventajas
* **Asignar un sistema de ficheros a una partición determinada**: Aprenderemos que todas las particiones necesitan su sistema de ficheros y a como asignárselo.
* **Montar un sistema de ficheros**: Entenderemos el concepto de punto de montaje y por que es tan importante en el mundo de los sistemas operativos Linux.
* **Desfragmentar un sistema de ficheros**: Abarcaremos el problema de la fragmentación, que herramientas existen para desfragmentar
* **Administrar los volúmenes lógicos del sistema**: Los estudiantes deben saber cómo identificar y depurar errores en los scripts, además de aprender a optimizar el rendimiento de los mismos.

7.2. Particiones y discos.

El particionamiento nos permite dividir un disco en unidades de almacenamiento lógicas denominadas particiones que nos proporcionan una serie de ventajas como la organización y aislamiento de datos, copias de seguridad sencillas, posibilidad de utilizar distintos tipos de sistemas de archivos, instalación de distintos sistemas operativos…

En Windows cada partición tiene asignado un volumen representado por una letra (C:, D:, E: …) y cada uno de los volúmenes alberga su propia estructura de directorios.

El sistema operativo de Linux por defecto cuenta con particionamiento simple teniendo una única partición que ocupa todo el disco y el punto de montaje de su sistema de ficheros es el directorio raíz ( / ) que integra los principales directorios del sistema de Linux, no obstante, es recomendable crear otras particiones que únicamente contengan directorios clave como /boot, /home, /var o /usr. Precisamente una de las características de Linux es la flexibilidad de ubicar cada directorio en distintas particiones compartiendo todas ellas una única estructura de directorios común.

Los discos duros y SSDs de nuestro equipo pueden estructurar sus particiones atendiendo a dos estándares: MBR (que permite hasta 4 particiones primarias o 3 primarias y una extendida) y GPT (permite un número casi ilimitado y no distingue entre particiones lógicas o primarias) más adelante en la unidad profundizaremos explicando lo que es una partición primaria, extendida y lógica.

Hasta hace relativamente poco, en Linux, se tenía que hacer una partición dedicada para la zona de intercambio o swap que se encarga de alojar los datos que la memoria RAM ya no puede absorber evitando así un fallo de sistema y se los devuelve cuando deje de estar tan saturada.

Las distribuciones más modernas ya no utilizan la partición swap dedicada si no un fichero llamado swapfile que hace la misma función con la diferencia de que este fichero asigna espacio de la partición actualmente en uso.

7.3. Administración de particiones y Esquemas de particionamiento.

En un esquema de particiones utilizamos los conceptos de divisiones o **particiones primarias** para referirnos a la parte de disco encargada almacenar el sistema operativo y **particiones extendidas** para referirnos otras divisiones meramente organizativas. Las particiones extendidas pueden contener a su vez hasta 11 subdivisiones denominadas **particiones lógicas**, estas particiones lógicas son las que utilizamos para organizar los datos y archivos adicionales que tenemos en nuestro ordenador.

Como mencionamos en el apartado anterior existen dos tipos de estándares que se utilizan para estructurar las particiones de nuestro disco:

* **MBR (Master Boot Record o Registro de Arranque Maestro)**

Es el esquema de particionamiento de los sistemas con estándar BIOS. Es más antiguo, pero aún se sigue empleando por su compatibilidad con sistemas operativos.

Tanto en Linux como en Windows con el uso de particiones extendidas y lógicas se pueden crear un máximo de 15 unidades totales, esto incluye 3 particiones primarias y una extendida, esta última, con hasta 11 particiones lógicas en su interior.

* **GPT (UUID Partición Table o Tabla de Particiones UUID)**

Esquema de particionamiento de los sistemas con estándar UEFI (Unified Extensible Firmware Interface). A cada partición se le asigna un identificador global único (UUID). Si bien la lista de sistemas operativos que admiten GPT es más limitada, la mayoría de los sistemas operativos modernos son compatibles.

El estándar UEFI mejora al tradicional de BIOS y solventa multitud de limitaciones al ser más flexible, potente y fácil de usar.

* Teóricamente puede contener un número ilimitado de particiones pero el límite de particiones primarias que soporta el sistema operativo de Linux es 256, en el caso de Windows es 128 particiones.
* Soporta discos de tamaño superior al esquema MBR que limitaba en 2.2TB la capacidad máxima, despreciando en resto.
* Mejora el tiempo de arranque y dispones de una interfaz amigable.
* Es más seguro que el sistema BIOS ya que realiza comprobaciones de originalidad, además hace copias de la estructura de la tabla de particiones.
* Permite trabajar en modos de 32 y 64 bits.
* Se puede conectar a Internet.
* Consta de un gestor de arranque propio no vinculado al sistema operativo.

Los discos con formato GPT deben tener una partición del sistema EFI, esta partición ha de estar formateada en FAT32 y suele ser la primera. En ella se encuentran principalmente los cargadores de arranque de los sistemas operativos, firmware UEFI que inician los sistemas de comprobación, imágenes kernel y otras utilidades.

Un disco con esquema GPT tiene una parte llamada “Protective MBR”, que evita que programas de gestión de particiones antiguos que no reconocen GPT puedan dañar la estructura de datos contenida en su partición.

Los programas de gestión y edición de particiones permiten al administrador crear, redimensionar, eliminar, o fusionarlas particiones de disco. el tipo de partición. La herramienta de interfaz gráfica más conocida para este propósito es “gparted”.

Los comandos citados a continuación deben ejecutarse con permisos de super usuario (root) o con permisos de administración del sistema similares.

Podemos consultar de forma resumida la información sobre los dispositivos conectados en el sistema tales como unidades de memoria y sus particiones con el comando [ **lsblk** ]

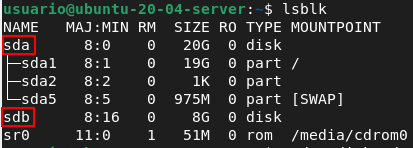


Figura 1. Comando lsblk. Fuente: elaboración propia.

Después de ejecutar el comando “lsblk” podemos observar que tenemos dos discos duros conectados, sda (con 20GB y tres particiones) y sdb (con 8GB).

* **Comando “fdisk -l”**

Con este comando obtendremos un listado de todas las particiones existentes en el sistema incluso los nuevos discos instalados aunque aún no estén formateados además de toda información relativa a los discos y USB externos conectados. Si queremos ver la información de un solo disco en específico debemos añadir al argumento del comando el nombre del dispositivo (Ej. **fdisk –l /dev/sdb**).

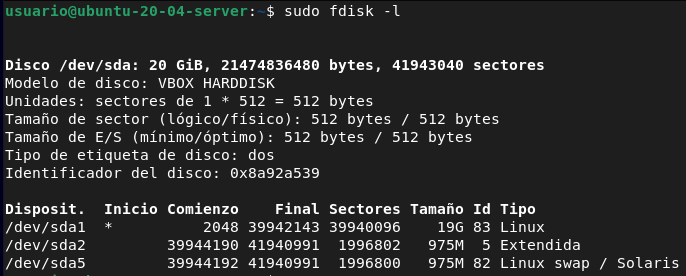


Figura 2. Comando fdisk -l. Fuente: elaboración propia.

* **Comando “gdisk”**

La herramienta gdisk está diseñada para administrar particiones de disco en sistemas Linux con tabla de particionado UUID (GPT) desde la línea de comandos. Se trata de una utilidad bastante completa que nos plantea un menú interactivo por consola, sin interfaz gráfica funcionando mediante la selección de distintas opciones.

(Ej. gdisk /dev/sdb):

|  |  |
| --- | --- |
| **Opción** | **Comando** |
| **?** | Muestra la sección de ayuda para ver el listado de opciones que tenemos para aplicar sobre el disco. |
| **p** | Muestra información detallada de la tabla de particiones asociadas al disco. |
| **d** | Borra una partición en concreto. |
| **n** | Crea una nueva partición, el siguiente paso sería decir si es primaria (pulsando “p”) o extendida (pulsando “e”) |
| **t** | Cambia el tipo de partición. |
| **q** | Sale del programa sin guardar cambios. |
| **w** | Confirma los cambios aplicados sobre el disco y sale del programa. |
| **v** | Verifica el disco en busca de errores |

Tabla 1. Opciones del comando “gdisk”. Fuente: elaboración propia.

A continuación vamos a crear una partición de tipo GPT en el disco (sdb) recién instalado, sin particiones anteriores.

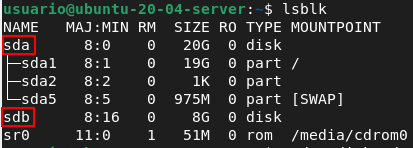


Figura 3. Información de dispositivos conectados, comando. Fuente: elaboración propia.

Comenzamos el proceso de creación utilizando “gdisk /dev/sdb” para entrar al menú de opciones de la herramienta gdisk, recordemos que pulsando “?” se nos muestra el listado con todas las opciones disponibles y una breve descripción de cada una.

Para asignar una nueva partición desde cero seleccionamos la opción “n”, en el proceso de creación nos pide introducir un número que por defecto será el uno, de esto resulta el nombre de la partición (sdb1), seguidamente nos solicita que asignemos un valor al primer sector, lo recomendable es pulsar enter para permitir que se establezca el valor calculado por defecto, a continuación nos solicitará introducir el tamaño de partición que deseamos, en nuestro caso elegiremos +2G.

Por último nos pide definir el tipo de partición mediante el HEX CODE, una vez más, lo más conveniente es pulsar enter ya que por defecto se establecen en 8300 que corresponde con partición de Linux pero si se desea utilizar un tipo diferente existe la opción L que muestra un listado de todos los tipos de partición disponibles.

Cuando terminamos de crear la partición antes de salir del panel de opciones debemos guardar los cambios realizados con “w”, si seleccionamos la opción “q” saldremos del menú sin guardar cambios.

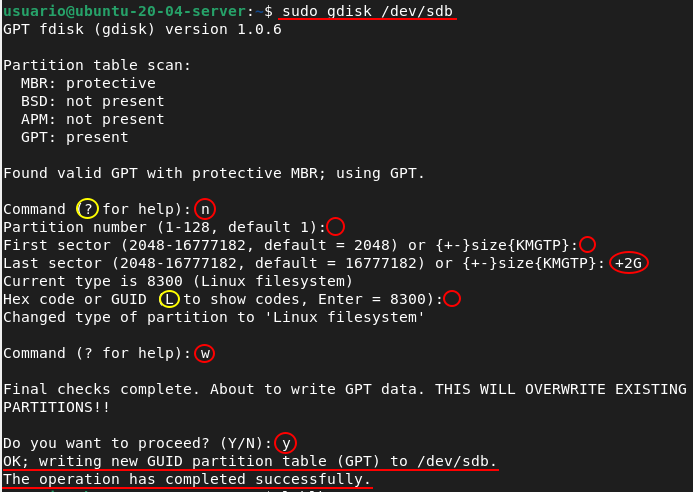


Figura 4. Creación de una tabla de particiones y una partición de tipo GPT. Fuente: elaboración propia.

Una vez creada la partición usaremos de nuevo el comando “lsblk” para observar cómo ha quedado el esquema de particiones para el disco sdb:

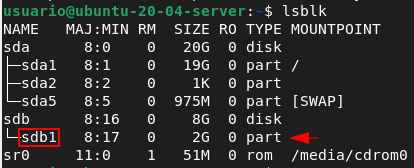


Figura 5. Información de dispositivos conectados, comando lsblk. Fuente: elaboración propia.

Ahora tenemos un disco nuevo con una partición sdb1 pero antes de poder usarla debemos darle formato o formatearla, es decir, asignarle un sistema de ficheros. En nuestro caso vamos a utilizar ext4. Cada partición cuenta con un único sistema de ficheros pudiendo ser este distinto para entre ellas.

Estudiaremos más a fondo los sistemas de ficheros en el siguiente punto de la unidad.

* **Comando “mkfs.<sistema de archivos elegido>”**

Formatea la nueva partición al sistema de archivos elegido (Ej. mkfs.ext4 /dev/sdb1).

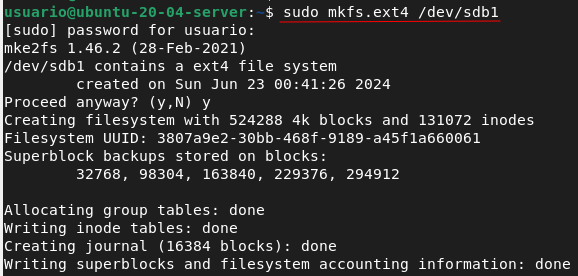


Figura 6. Ejemplo asignación sistema de archivos a la partición sdb1. Fuente: elaboración propia.

Podemos comprobar la asignación del sistema de archivos que hemos asociado con la opción “ -f ” del comando “lsblk”.

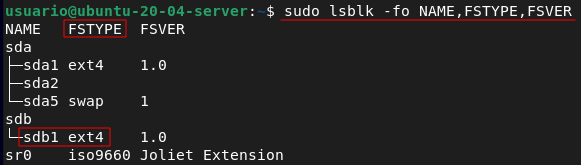


Figura 7. Comando lsblk -fo. Fuente: elaboración propia

Se observa también que utilizando la opción “–o” del comando “lsblk” podemos especificar la lista de columnas que deseamos visualizar para simplificar la salida de datos que nos muestra el comando.

* **Comando “mount y umount”**

Montar un disco, significa adjuntarle el sistema de ficheros un directorio específico dentro del árbol o estructura de directorios de Linux, este directorio recibe el nombre de “**punto de montaje”.** Existen dos tipos de montaje, automático y el manual. Las diferencias, ventajas y desventajas entre ellos son:

* Montaje automático:

Las particiones se montan automáticamente al arrancar el sistema. Para configurar esta opción debemos modificar el archivo de configuración **“/etc/fstab”** y añadirle la línea correspondiente.

* Ventaja: No requiere de intervención manual cada vez que se inicia el sistema.
* Desventaja: Si la partición no está disponible (Ej. un USB desconectado), puede generar un error durante el inicio.
* Montaje manual:

Se realiza mediante el comando [ **mount** ].

* Ventaja: Permite un control más preciso sobre la opción montar y desmontar según sea necesario.
* Desventaja: Requiere de intervención manual cada vez que sea necesario montar la partición ya que el montaje no persiste tras apagar el sistema.

Antes de comenzar con el montaje de la partición debemos profundizar en el significado UUID. Cada partición del disco tiene asignado un número único e identificativo (UUID), este identificador es el que se indica en el fichero de configuraciones “/etc/fstab” para establecer el punto de montaje. El comando [ **blkid** ] muestra el UUID de las particiones de un disco en un sistema de archivos.

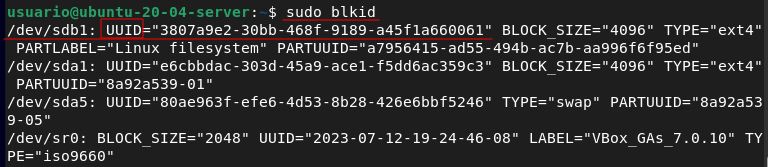


Figura 8. Comando blkid mostrando los UUID. Fuente: elaboración propia

Para establecer el punto de montaje automático al iniciar el sistema tenemos debemos modificar el fichero de configuración /etc/fstab con el editor que prefiramos y añadir una línea más, correspondiente a la partición que debe contener la siguiente información:

* <file system>: Define el UUID.
* <mount point>: Directorio de montaje.
* <type>: Tipo de sistema de ficheros.
* <options>: Opciones de montaje (lectura y escritura es la opción por defecto).
* <dump>: Indica cuando se debe realizar una copia de seguridad, las posibles entradas son 0 (no hace copia) y 1 (hace copia de archivos).
* <pass>: Referida al contexto en la utilización fsck (que verifica y repara sistemas de archivos) pass determina el orden de verificado de estos archivos, las entradas pueden ser 0 (los archivos no se comprueban), 1 (asigna la prioridad más alta de verificación de archivos, estos son los archivos raíz) y 2 (se comprueban después del sistema de archivos de raíz).

El ejemplo de la línea a añadir siguiendo el esquema de información es el siguiente:

**UUID=3807a9e2-30bb-468f-9189-a45fla660061 /mnt/particion\_datos ext4 defaults 0 2**

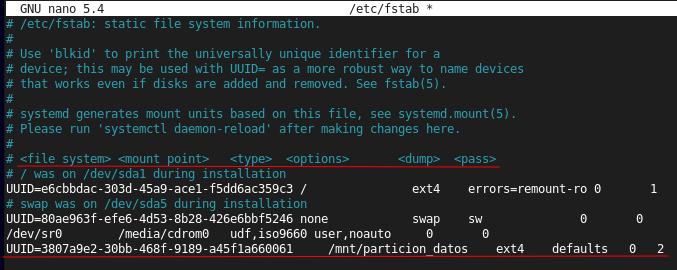


Figura 9. Fichero de configuración /etc/fstab. Fuente: elaboración propia

Tras añadir la línea y guardar el archivo, hay que reiniciar el sistema para que los cambios se persistan. Con el reinicio, el sistema chequea los discos existentes y los monta en el arranque. Para verificar que se ha montado correctamente en el directorio indicado ejecutamos de nuevo el comando [lsblk]:

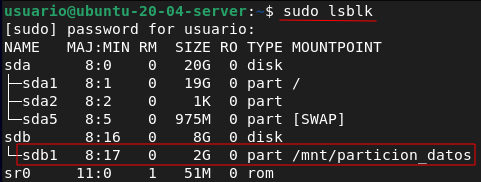


Figura 10. Comando lsblk y punto de montaje. Fuente: elaboración propia

Para desmontar basta con comentar añadiendo el símbolo ( # ) al inicio de la línea añadida en el fichero /etc/fstab y volver a reiniciar.

El comando [ **mount** ] asigna el punto de montaje de forma manual a una partición, pero esta opción de montaje no es definitiva, habrá que volver a montar cada vez que lo necesitemos (Ej. **mount /dev/sdb1 /mnt/particion\_datos2**).

Para revertir los cambios, es decir, desmontar o desasignar el punto de montaje utilizamos el comando [ **umount** ] seguido del nombre de la partición (Ej. **umount /dev/sdb1**).

Veamos un ejemplo práctico en la siguiente imagen:

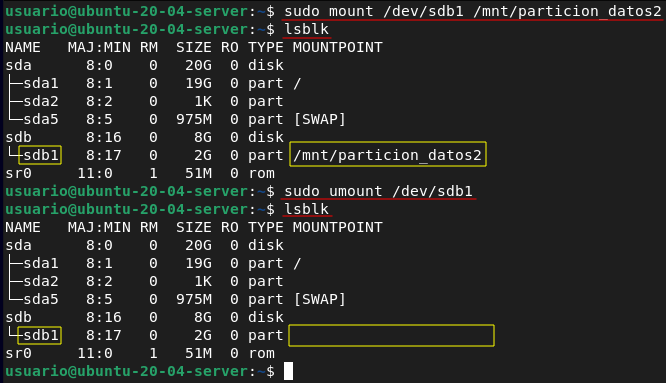


Figura 11. Comando mount, umount y lsblk. Fuente: elaboración propia

7.4. Sistemas de ficheros.

Como ya sabemos cualquier archivo digital, que necesitemos conservar, se guardan en dispositivos de almacenamiento como discos HDD, SSD, unidades de memoria USB, etc. Cada uno de estos dispositivos cuenta con una capacidad o cantidad de espacio libre, pero ¿Cómo se organizan todos los bits de datos en su interior? Es en este punto donde aparece el concepto de “sistema de ficheros”.

**El sistema de ficheros** (también conocido como sistema de archivos), es un elemento fundamental para cualquier sistema operativo ya que **su función principal es organizar el modo en que se almacenan y recuperan los datos** en un medio de almacenamiento.

Sin un sistema de ficheros los dispositivos de almacenamiento serían un cumulo de datos sin forma. Para evitarlo, el sistema de ficheros lleva a cabo las siguientes tareas:

* Gestiona la asignación de espacio a los archivos.
* Gestiona errores en los discos duros
* Facilita el acceso a los datos.
* Almacena metadatos útiles de cada fichero
* Controla el espacio libre y el ocupado.

Existe una gran cantidad de sistemas de ficheros distintos con sus ventajas y desventajas, incluso algunos de ellos están especialmente diseñados para un propósito concreto, pero cabe destacar que no todos los sistemas de archivos son compatibles con todos los sistemas operativos.

Debido a su naturaleza, Linux cuenta con muchas distribuciones diferente y soporta una gran variedad de sistemas de ficheros por lo que vamos a explicar los más conocidos:

* **Ext2 (Extended File System 2)**

Es el sistema de ficheros de alto rendimiento que usaba Linux en el pasado para discos duros y sistemas de almacenamiento extraíbles. Fue diseñado como una extensión del sistema de ficheros extendido original (Ext) para ofrecer mejor rendimiento sobre todo en términos de velocidad de lectura y escritura.

* **Ext3 (Extended File System 3)**

Es la versión mejorada de ext2 con algunas mejoras como la previsión de pérdida de datos por fallos de disco o cortes de corriente. Sus principales ventajas son la fiabilidad y la compatibilidad ext2 ya que ambos sistemas comparten el mismo formato.

* **Ext4 (Extended File System 4)**

Se trata de la última versión de la familia de sistemas de ficheros ext. En la actualidad es el sistema más difundido dentro de la comunidad GNU/Linux, considerado el estándar. Su principal ventaja es la ampliación de los límites de tamaños para ficheros, ahora de hasta 16TB y del propio sistema de ficheros que puede llegar hasta los 1024PB (PetaBytes). Pero las mejoras en eficiencia también son notorias, tanto en la reducción del uso de CPU como en la velocidad de lectura y escritura.

* **FAT32 (File Allocation Table – 32 bits)**

Se trata de un sistema de archivos desarrollado por Windows y evolucionado del anterior FAT16. Este sistema también es compatible con Linux y se utiliza principalmente en pequeñas unidades flash y discos extraíbles debido a su amplia compatibilidad y soporte con un gran número de dispositivos. FAT32 admite un tamaño máximo de archivo individual de 4GB y un tamaño máximo de volumen de 2TB.

* **NTFS (New Technology File System)**

Es el sistema más actual desarrollado y utilizado por Windows desde la llegada de Windows NT 3.1 aunque también es compatible con Linux. Este sistema de ficheros presenta varias ventajas respecto de su predecesor FAT32:

* Robustez: Es más resistente a la corrupción de archivos lo que ayuda a reducir la posible pérdida de datos
* Seguridad: Permite la encriptación de contenido para aumentar la privacidad.
* Tamaño de volumen: Admite volúmenes de hasta 16 exabytes.
* Tamaño de archivo: Puede gestionar archivos del tamaño máximo del volumen, es decir, de hasta 16 exabytes.

Todos los medios de almacenamiento, ya sean discos HDD, discos SSD, unidades de almacenamiento USB o particiones, necesitan su propio sistema de ficheros para poder realizar su función. Dicho sistema de archivos se asigna y puede cambiarse posteriormente en el proceso conocido como formateo o lo que es lo mismo, dar formato a la unidad.

Podemos formatear una unidad de disco o una partición recién creada desde la consola mediante el comando [ mkfs ] desarrollado en el punto 7.3 de la unidad

# Sintaxis del comando

**sudo mkfs.sistema\_de\_archivos <ruta\_de\_disco/particion>**

# Ejemplo

**sudo mkfs.ext4 /dev/sdb1**

Cabe destacar que cada sistema de ficheros cuenta con un UID o numero de identificación propio utilizado para asignarle un punto de montaje, por ejemplo el sistema ext4 cuenta con un UID de 16 bytes mientras que el sistema NTFS cuenta con un UID de 8 bytes.

* **Fragmentación.**

La fragmentación de un sistema de archivos ocurre cuando un conjunto de datos, relacionados entre sí, no se encuentran almacenados juntos si no dispersos en diferentes posiciones de memoria no contiguas en el disco.

La fragmentación y sus características dependen de cada sistema de archivos. En sistemas de archivos NTFS y FAT la desfragmentación es común para mejorar su rendimiento, en cambio, los sistemas de archivos Linux, en general, no necesitan desfragmentarse, puesto que dejan un espacio de “n” bloques entre ficheros evitando que el futuro crecimiento de un fichero suponga la división de bloques en otra parte del disco. Por otro lado, si un fichero reduce su tamaño, el sistema de archivos libera ese espacio para que otro fichero pueda ocuparlo.

No obstante, el comando [ **e4defrag** ] es una utilidad que nos permite desfragmentar sistema de archivos ext4 instalado en un disco o partición determinados, (Recomendada en dispositivos de poco espacio en disco).

El comando [ e4defrag ] cuenta con algunas opciones útiles que se recogen en la siguiente tabla:

* **Comando e4defrag**

|  |  |
| --- | --- |
| **Opción** | **Efecto** |
| **-c** | Realiza una simulación y muestra un recuento de la desfragmentación sin llegar a efectuarla. |
| **-v** | Muestra el recuento para cada archivo antes y después de la desfragmentación en caso de realizarse. |

Tabla 2. Opciones disponibles del comando e4defrag. Fuente: elaboración propia.

Vamos a ver un ejemplo de desfragmentación para el sistema de ficheros de la partición sdb1 con la opción -v (al ser una partición nueva encontrará un 0% de fragmentación).

# Sintaxis del comando

**sudo e4defrag -v <ruta\_partición>**

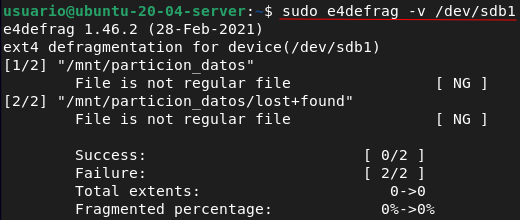


Figura 12. Ejecución del comando e4defrag. Fuente: elaboración propia.

7.5. Administrador de volúmenes lógicos.

Para administrar los volúmenes lógicos primero debemos entender el concepto de “grupo de volúmenes”.

Hasta el momento conocemos los discos o unidades de almacenamiento cuya capacidad total puede dividirse de forma lógica en diferentes partes denominadas particiones. El tamaño máximo de una partición no puede exceder el tamaño total del disco de forma que estamos limitados en caso de necesitar particiones más amplias. La creación de grupos de volúmenes viene a solucionar este inconveniente.

Un grupo de volúmenes o VG por sus siglas en ingles “Volume Group” consiste en la fusión lógica de dos o más volúmenes físicos para que puedan trabajar como una única unidad. Veamos un ejemplo:

Si tenemos dos unidades de disco con 500GB de espacio cada una, el primer paso es convertir las unidades de disco a volúmenes físicos, para después poder fusionarlos en un solo grupo de volúmenes con una capacidad total de 1TB.

Cabe destacar que no es posible convertir una unidad de disco particionada en volumen físico, se deberán eliminar todas las particiones antes de continuar el proceso, (podemos eliminarlas con la opción “d” de la herramienta gdisk, explicada en el punto 7.3 de esta unidad).

Ahora que entendemos que es un grupo de volúmenes introducimos el concepto de volúmenes lógicos,

Los volúmenes lógicos son las partes lógicas en las que se puede dividir un grupo de volúmenes, es decir, son el equivalente a las particiones de una unidad física pero aplicadas a un grupo de volúmenes.

Las ventajas que ofrece un volumen lógico es que puede ser mas grande que cualquiera de los volúmenes físicos que componen el grupo pero siempre más pequeños que el tamaño total del grupo de volúmenes.

Vamos a ver un ejemplo de cómo crear un grupo de volúmenes a partir de dos unidades de disco físicas y después crearemos varias unidades lógicas dentro del grupo, pero antes conviene recoger algunos conceptos en la siguiente tabla:

|  |  |
| --- | --- |
| **Opción** | **Función** |
| **Unidad de disco** | Unidad de almacenamiento hardware o disco físico del equipo. |
| **Volumen físico (PV)** | Es cada uno de los componentes físicos que pueden formar parte de un grupo de volúmenes. |
| **Grupo de volúmenes (VG)** | Conjunto de volúmenes físicos que suman sus capacidades para trabajar como una única unidad a nivel lógico. |
| **Volumen lógico**  **(LV)** | Son las partes en las que podemos dividir un grupo de volúmenes, no hay restricciones en cuanto al número de volúmenes lógicos que se pueden crear, el límite lo marca la capacidad total del grupo. |

Tabla 3. Conceptos de (LVM). Fuente: elaboración propia.

Partimos de un sistema con 2 unidades de disco sin particionar: sdb (con un tamaño de 8 GB) y sdc (con un tamaño de 10 GB). Lo comprobamos con el comando [ lsblk ]

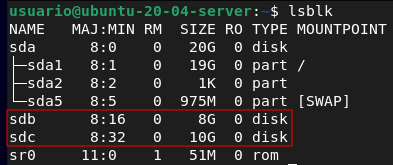


Figura 13. Información de dispositivos conectados, comando lsblk. Fuente: elaboración propia.

Para administrar los volúmenes lógicos, existe una herramienta denominada “lvm2” en Debian y derivados que debemos descargar e instalar primero con el siguiente comando:

**sudo apt install lvm2 -y**

Los comandos que vamos a ejecutar a continuación solo están disponibles si contamos con lsvm2 instalado en nuestro sistema.

El primer paso es convertir nuestras dos unidades lógicas en volúmenes físicos, para ello utilizamos el siguiente comando:

**sudo pvcreate /dev/sd{b,c}**

Posteriormente comprobamos que se han creado correctamente con el comando [ **sudo** **pvscan** ] que devuelve la lista de volúmenes físicos disponibles y el grupo de volúmenes al que pertenecen si fuera el caso.

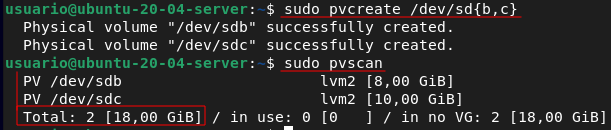


Figura 14. Comando pvcreate. Fuente: elaboración propia.

El siguiente paso es crear el grupo de volúmenes compuesto por los dos volúmenes físicos creados en el punto anterior, para ello debemos ejecutar el siguiente comando:

**sudo vgcreate grupo\_prueba /dev/sd{b,c}**

Posteriormente comprobamos que el grupo se han creado correctamente con el comando [ **sudo** **vgscan** ] y el comando [ **sudo** **pvscan** ] que devuelve la lista de volúmenes físicos disponibles y el grupo de volúmenes al que pertenecen.

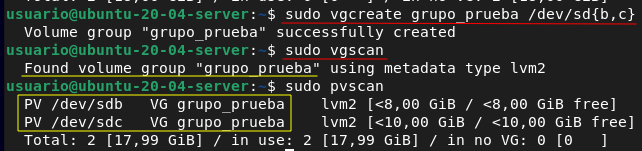


Figura 15. Comando vgcreate. Fuente: elaboración propia.

Podemos obtener información detallada del grupo de volúmenes utilizando el comando [ **sudo vgdisplay** ].

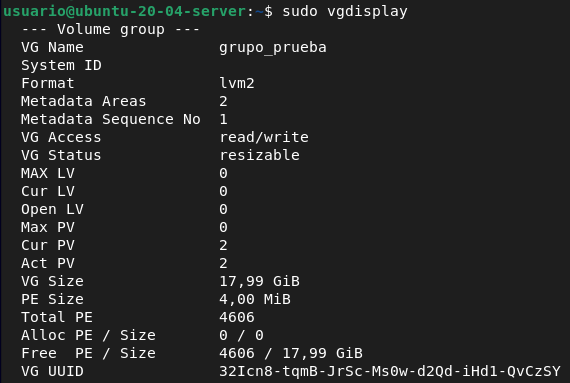


Figura 16. Comando vgdisplay. Fuente: elaboración propia.

Por último, crearemos el volumen lógico de 11GB dentro del grupo con el comando:

**sudo lvcreate - - size 11G grupo\_prueba**

Posteriormente comprobamos que el grupo se han creado correctamente con el comando [ **sudo** **lvdisplay** ] que nos muestra información de los volúmenes lógicos.

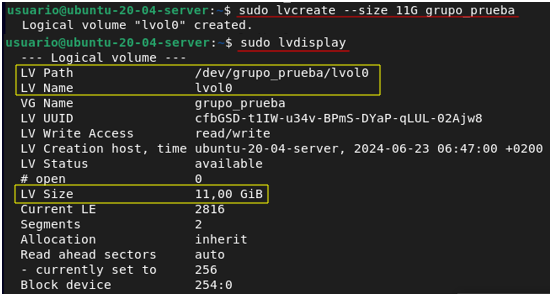


Figura 17 Comando lvcreate. Fuente: elaboración propia.

El último paso sería dar formato y montar el nuevo volumen lógico tal y como hemos visto en el punto 7.3 de esta unidad.

7.6. Referencias Bibliográficas.

Podemos encontrar más información sobre todos los conceptos estudiados en la unidad accediendo a al siguiente enlace:

<https://wiki.debian.org/es/FrontPage?action=show&redirect=P%C3%A1ginaInicial>

Para obtener información más detallada sobre cualquiera de los apartados es recomendable recurrir al manual oficial de RedHat visitando el siguiente enlace:

<https://docs.redhat.com/en/documentation/Red_Hat_Enterprise_Linux/4/html/Cluster_Logical_Volume_Manager/ch_Introduction-CLVM.html#ch_Introduction-CLVM>

Cabe destacar que podemos consultar una breve descripción de cada comando y de sus opciones ejecutando el comando de ayuda - -help o man dentro de la línea de comandos, pongamos como ejemplo el comando read para visualizar como recibir está ayuda :

nano --help o man gdisk g

A fondo

Creando particiones y volúmenes lógicos con LVM en Linux.

Sergio Geek. (2023). Creando particiones y volúmenes lógicos con LVM en Linux. [Video] https://www.youtube.com/watch?v=NYYvFZPxxXo

En este video encontraremos un apoyo visual al material didáctico de la unidad donde se repasarán los apartados estudiados.

**Cómo CREAR PARTICIONES en LINUX Desde el TERMINAL con Fdisk**

El Rincón del Hacker. (2022). Cómo CREAR PARTICIONES en LINUX Desde el TERMINAL con Fdisk. [Video] https://www.youtube.com/watch?v=VDsfvHICZaU

En este video encontraremos como crear una tabla de particiones en nuestro disco en Linux a través de los comandos. Usaremos la herramienta fdisk y una serie de comandos para crear la partición primaria, la partición extendida y las particiones lógicas.

Entrenamientos

Entrenamiento 1

* Enunciado

Partiendo de un disco recién instalado (sdb) con 10GB de capacidad, crea una partición de 5GB llamada sdb1 y asígnale un sistema de ficheros ext4.

* Solución

Primero debemos lanzar la herramienta gdisk pasándole como argumento el disco sobre el cual queremos hacer la partición:

* **gdisk /dev/sdb**

A continuación entraremos en el modo menú de la consola y deberemos introducir las opciones correspondientes.

Comenzaremos con la opción “**n**” para crear una nueva partición.

El segundo paso es asignarle un número a la partición, por defecto será el 1 si solo pulsamos enter se establecerá el nombre de la partición como el nombre del disco seguido del número 1, (sdb1).

El tercer paso nos solicita el tamaño del primer sector que también dejaremos por defecto.

el cuarto paso nos pregunta por el tamaño de la partición por lo que deberemos informar que serán 5GB en el siguiente formato (**+5G**).

Por último nos solicita el código Hex para definir el tipo de partición que también dejaremos por defecto ya que este código es el 8300 que corresponde con una partición Linux.

Tras seguir estos pasos solo nos queda seleccionar la opción “**w**” para guardar los cambios y proceder con “**Y**” para confirmarlos.

Con la partición (sdb1) ya creada debemos asignarle un sistema de ficheros ext4. Esta tarea la podemos llevar a cabo mediante el siguiente comando:

* **mkfs.ext4 /dev/sdb1**

Entrenamiento 2

* Enunciado

Monta la el sistema de ficheros de la partición sdb1 en el directorio /mnt/entrenamiento1 de tal manera que no se desmonte cuando apagamos el equipo y persista con un nuevo arranque del sistema operativo.

* Solución

Recordemos que existen dos formas de montar el sistema de ficheros en la partición seleccionando un punto de montaje, el ejercicio especifica que el montaje debe persistir después de apagar el equipo lo que significa que debe montarse de forma automática al arrancar el sistema, este efecto lo podemos conseguir mediante el sistema de montaje automático.

El primer paso es consultar el código UUID correspondiente a la partición en cuestión mediante el comando:

* **sudo blkid**

Una vez que tenemos el UUID debemos editar el fichero de configuración correspondiente que se encuentra en la ruta /etc/fstab y añadir una línea más siguiendo la siguiente sintaxis:

<<**UUID de la partición**>> <<**ruta del punto de montaje**>> <<**tipo de sistema de** **archivos elegido**>> <<**opciones de montaje “defaults”>>** <<**valor 0 o 1 para la realización de copias de seguridad**>> <<**2 para el orden de comprobación**>>

El aspecto de la línea introducida debe ser el siguiente:

**UUID=3807a9e2-30bb-468f-9189-a45fla660061 /mnt/particion\_datos ext4 defaults 0 2**

Para finalizar, deberemos reiniciar el equipo para que el sistema recorra todos los archivos de configuración en el arranque incluido /etc/fstab y actualice la configuración.

Entrenamiento 3

* Enunciado

Efectúa la desfragmentación de la partición sdb1 y después elimina dicha partición.

* Solución

Dado que estamos ante una partición con sistema de ficheros ext4 basta con utilizar el comando [ **e4defrag** ] seguido de la ruta de la partición para Para desfragmentar el sistema de ficheros.

Si queremos que nos informe del porcentaje que estaba fragmentado y de los cambios tras la desfragmentación deberemos añadir la opción -v al comando quedando de la siguiente manera:

* **sudo e4defrag -v /dev/sdb1**

Una vez desfragmentado, el ejercicio nos pide que eliminemos la partición, para ello debemos volver a la herramienta [ **gdisk** ] y pasarle como parámetro la ruta de la partición que queremos eliminar como en el siguiente comando:

* **sudo gdisk /dev/sdb1**

Y una vez dentro del menú seleccionar la opción ( d ) para eliminar la partición y comprobamos que ya no existe con el siguiente comando:

* **sudo lsblk**

Entrenamiento 4

* Enunciado

Partiendo de un equipo con dos discos recién instalados, sdb (8GB) y sdc (7GB) crea un grupo de volúmenes con una capacidad total de 15GB que se llame grupo\_entrenamiento.

* Solución

Para crear un grupo de volúmenes partiendo de dos unidades de disco físico sin particionar, primero debemos instalar la herramienta de administración de volúmenes lógicos denominada [ lsvm2].

* **sudo apt install lsvm2 -y**

A continuación debemos convertir las unidades de disco en volúmenes físicos, esta tarea podemos llevarla a cabo mediante el siguiente comando:

* **sudo pvcreate /dev/sd{b,c}**

AY por último debemos unir los dos volúmenes lógicos creados en un solo grupo de volúmenes con una capacidad de almacenamiento igual a la suma de las capacidades de los volúmenes lógicos que lo conforman.

Esta ultima tarea podemos llevarla a cabo mediante la ejecución del siguiente comando:

* **sudo vgcreate grupo\_entrenamiento4 /dev/sd{b,c}**

Es recomendable comprobar los cambios que se han ido realizando con los siguientes comandos de control:

**vgscan** 🡪 Muestra los grupos de volúmenes existentes en el sistema.

**pvscan** 🡪 Muestra los volúmenes físicos disponibles en el sistema.

**vgdisplay** 🡪 Muestra información detallada del o los grupos de volúmenes.

Entrenamiento 5

* Enunciado

Crea un volumen lógico de 12GB en el grupo de volúmenes del ejercicio anterior, ¿es posible? ¿Es posible crear un volumen lógico de 16GB?

* Solución

Podemos crear volúmenes lógicos de un tamaño superior a los volúmenes físicos que componen el grupo de volúmenes pero nunca podemos exceder el tamaño del propio grupo con un volumen lógico.

Para crear un volumen lógico debemos ejecutar el siguiente comando:

* **sudo lvcreate - - size 12G grupo\_entrenamiento**

Podemos comprobar que el volumen lógico se haya creado ben con el comando

* **sudo lvcreate**

Este comando nos muestra información detallada sobre todos los volúmenes lógicos que hay en el sistema.

Respecto a la segunda pregunta, la respuesta es No, el tamaño máximo de un volumen lógico nunca puede exceder el tamaño máximo del grupo al que pertenece.

Test

**1.** ¿Qué es una partición?

A. La asignación de una letra a una unidad de memoria.

\_B. La división de un disco en unidades de almacenamiento lógicas.

C. La creación de un sistema de ficheros”.

D. Un directorio de log de Linux que almacena entradas a disco.

Las divisiones de un disco en unidades de almacenamiento lógicas denominadas se denominan particiones.

**2.** ¿Cuántas particiones acepta un disco GUID (GPT)?

A. 11.

B. 4.

C. 15.

\_D. ilimitado.

Teóricamente un número ilimitado de particiones pero el tope de particiones primarias que soporta el sistema operativo de Linux es 256, en el caso de Windows es 128 particiones.

**3.** ¿Qué muestra el comando fdisk -l?

\_A. Un listado de particiones existentes.

B. Un listado de discos existentes.

C. Un listado de ayuda con comandos.

D. Da error, comando inexistente.

Con este comando obtendremos un listado de todas las particiones existentes en el sistema.

**4.** ¿Cuál es la opción utilizada para mostrar la ayuda en gdisk?

A. -lt.

B. ¿?

\_C. ?

D. -l

El carácter ? muestra la sección de ayuda para ver el listado de posibilidades que tenemos para aplicar sobre el disco.

**5.** ¿ Para qué se utiliza la herramienta gdisk?

A. Para formatear particiones.

\_B. Para administrar particiones.

C. Para desfragmentar particiones.

D. Para actualizar los repositorios de Linux.

La herramienta gdisk está diseñada para administrar particiones de disco en sistemas Linux.

**6.** ¿Cuál de las siguientes opciones es correcta para crear una nueva partición?

A. d.

B. a.

C. p.

\_D. n.

El carácter n crea una nueva partición.

**7.** ¿Cuál es el comando utilizado para dar formato a una partición?

\_A. mkfs.

B. lsblk.

C. mount.

D. blkid.

Formatea la nueva partición al sistema de archivos elegido.

**8.** ¿Con qué comando podemos visualizar los dispositivos conectados y sus particiones?

A. mkfs.

\_B. lsblk.

C. mount.

D. blkid.

El comando lsblk nos muestra de forma resumida los dispositivos conectados

**9.** ¿Cuál es el comando para montar una partición en un directorio?

A. amount.

B. umount.

\_C. mount.

D. -mount.

Para asignar un punto de montaje utilizamos el comando mount.

**10.** ¿Cuántas particiones acepta como máximo un disco MBR?

\_A. 15

B. 11

C. 128

D. ilimitado

Tanto en Linux como en Windows con el uso de particiones extendidas y lógicas se pueden crear un máximo de 15 unidades totales.

**11.** ¿Cuál es la opción para eliminar una partición con gdisk?

A. 2>/dev/null.

\_B. d.

C. e4defrag.

D. delete.

El carácter d borra la partición

**12.** ¿Cuál es la función del sistema de ficheros o sistema de archivos?

A. Es un sistema de control de tráfico de red

B. Crear particiones primarias para los ficheros más pesados

C. Archivar los ficheros con más de un año de antigüedad

\_D. Organizar como se almacenan y recuperan los datos en un medio

El sistema de ficheros asigna espacio a los ficheros y los almacena de forma ordenada facilitando el acceso a los datos y controlando el espacio libre y ocupado del disco.

**13.** ¿Cuál de los siguientes sistemas de ficheros no es compatible con Linux?

A. ext4

B. NTFS

C. ext3

\_D. HDMI

HDMI son las siglas High Definition Multimedia interface que nada tiene que ver con el sistema de ficheros.

**14.** ¿Cuándo aparece la "fragmentación en el sistema de ficheros"?

\_A. Cuando los datos que guardan relación están dispersos por el disco

B. Cuando el disco es antiguo y comienza a agrietarse

C. Cuando tarda más de 1 segundo en abrir los ficheros

D. Cuando se crean mas de 100 particiones lógicas.

El sistema de ficheros sufre fragmentación cuando existen datos relacionados entre si que están almacenados en posiciones de memoria distintas y no contiguas,.

**15.** ¿Cuál es la utilidad del comando d4defrag?

A. Se utiliza para encriptar contraseñas.

B. Se utiliza para desfragmentar sistemas de ficheros FAT32

\_C. Se utiliza para desfragmentar sistemas de ficheros ext4

D. Se utiliza para desfragmentar sistemas de ficheros muy antiguos

El comando d4frag fue diseñado para desfragmentar únicamente sistemas de ficheros del tipo ext4.

**16.** ¿Qué es un grupo de volúmenes?

\_A. Un conjunto de volúmenes físicos

B. Un conjunto de unidades físicas

C. Un conjunto programas para controlar el audio del equipo

D. Es un comando de gestión de usuarios

Un grupo de volúmenes unifica varios volúmenes físicos para formar una unidad de almacenamiento única.

**17.** ¿Para qué se utiliza la opción -c del comando e4defrag?

A. Para continuar con la desfragmentación sin pedir confirmación

\_B. Para simular una desfragmentación

C. Es un comando de sistema necesario para arrancar el equipo

D. Es el comando utilizado para cambiar el tipo de sistema de ficheros

La opción -c sirve para hacer una simulación sin llegar a ejecutarla con fines informativos

**18.** ¿Qué función tiene la herramienta lvm2 en Debian?

A. Es una herramienta que facilita la descarga de software más rápido

B. Posibilita el uso de repositorios privados de Linux

C. Es una herramienta de apoyo a bash scripting

\_D. Posibilita la administración de volúmenes lógicos

La herramienta lvm2 ofrece todos los comandos necesarios para la gestión de los volúmenes lógicos.

**19.** ¿Cuál es el comando para visualizar información de grupos de volúmenes?

\_A. vgdisplay

B. vgviewer

C. vgmanager

D. vglist

El comando sudo vdisplay muestra información detallada sobre los grupos de volúmenes existentes en el sistema.

**20.** ¿Cuál es el comando utilizado para convertir una unidad de disco a volumen físico?

A. createpv

\_B. pvcreate

C. lvcreate

D. svcreate

Con el comando pvcreate podemos convertir unidades de disco sin particionar en volúmenes físicos listos para poder forman un nuevo grupo de volúmenes.