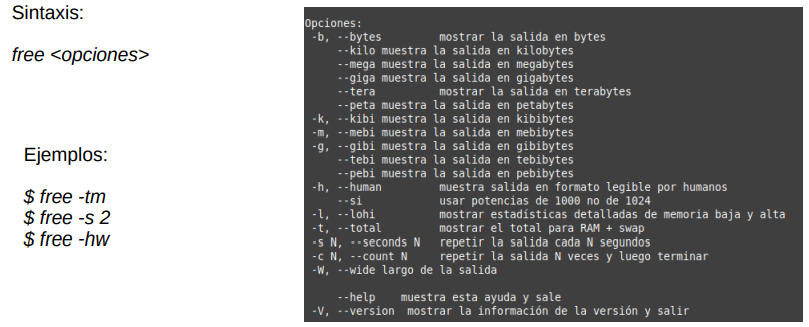
Resumen SO Segundo Parcial Practica

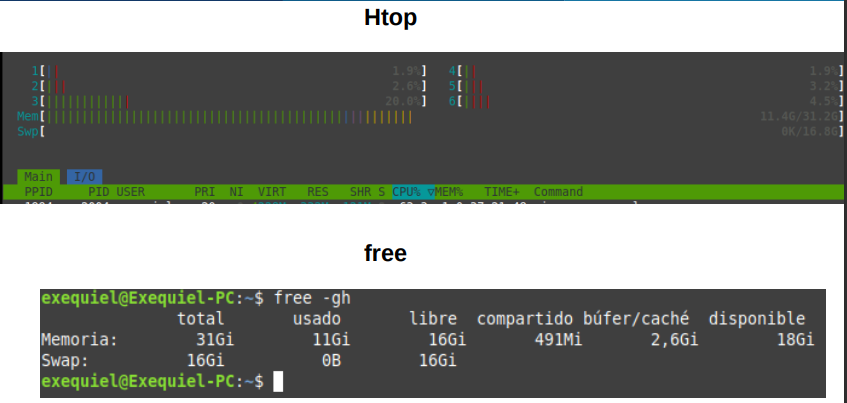
**Espacio de intercambio:** el espacio de intercambio en Debian y otros SO basados en GNU/Linux es una forma de memoria virtual. *En pocas palabras, esto significa que si el sistema se queda sin memoria física (RAM), transferirá algunos de los datos menos utilizados en la RAM a esto espacio.* El espacio de intercambio también es fundamental para los procesos de las funciones “Suspender” e “Hibernar” de Linux. El intercambio se configura principalmente como particiones separadas dedicadas (recomendado) o como archivos especialmente creados que residen en un sistema de archivos existente.

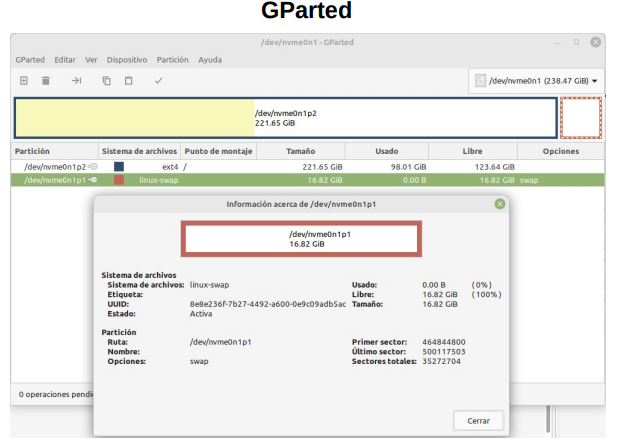
A partir de la serie de núcleos Linux 2.4, el núcleo de Linux puede manejar hasta 32 particiones de intercambio separadas o archivos de intercambio en un momento dado. Sin embargo, esto es muy poco común, ya que los discos duros son mucho mas lentos que la RAM. La mayoría de los sistemas solo tendrán una partición de intercambio o una partición de intercambio por instancia de Linux. El momento más común para configurar el intercambio es durante el proceso de instalación de Debian, pero se puede crear o modificar en cualquier momento.

**Comandos:**

~ **free** – muestra la utilización de memoria del sistema







**Tamaño del área de intercambio:** la cantidad recomendada de espacio de intercambio ha sido tradicionalmente el doble de la cantidad de memoria del sistema. Esto ha cambiado con el tiempo de una vez y media la memoria del sistema, ambas respuestas son líneas de base decentes, pero se están volviendo cada vez menos respuestas útiles a la pregunta a medida que pasa el tiempo. Hay muchas variables sobre su sistema y el uso previsto que determinaran el intercambio de sistema disponible que querrá tener.

**¿Tiene la intención de utilizar las funciones de hibernación y suspensión?**

Si tiene la intención de usar cualquiera de los dos, deberá asegurarse de que su espacio de intercambio disponible sea lo suficientemente grande como para contener toda la información que su sistema en ejecución tendrá en la RAM. Si tiene la intención de utilizar Hibernate, debe asegurarse de que su instalación de Debian tenga acceso a una partición o archivo de intercambio que NO se comparta con otra instancia de GNU/Linux

**¿Cuánta memoria tiene tu máquina?**

Si su máquina tiene una gran cantidad de RAM disponible y no necesita Hibernar o Suspender, entonces será más seguro tener menos intercambio disponible para el sistema. Si su máquina tiene muy poca memoria RAM disponible, como en el caso de las máquinas más antiguas y las netbooks, debe asegurarse de crear un espacio de intercambio que sea lo suficientemente grande. Duplicar la memoria del sistema sería una buena sugerencia en este caso.

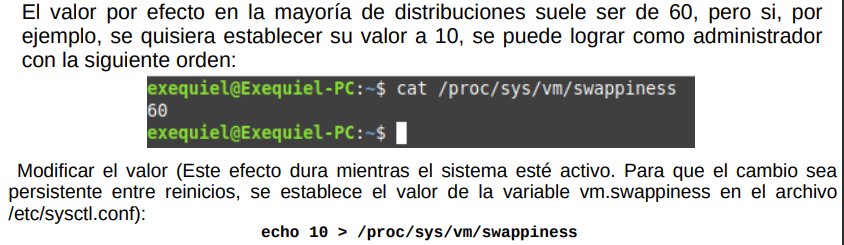
**¿Cuánta memoria ocuparán las aplicaciones que pretende ejecutar?**

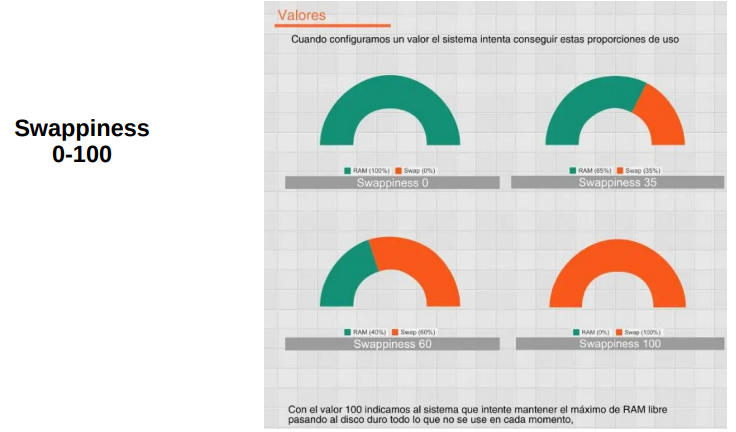
El uso de memoria de un sistema Debian GNU/Linux variará mucho según las aplicaciones que elija ejecutar. Por ejemplo, ejecutar un entorno de escritorio más ligero como Openbox o Xfce y sus aplicaciones asociadas utilizará menos memoria que GNOME o KDE.

**¿Cuánto disco duro tienes disponible?**

Si usa un disco de estado sólido pequeño o una llave USB para la instalación de Debian, este espacio en el disco duro puede ser escaso. En tales casos, podría tener más sentido instalar más memoria del sistema si es necesario que usar un gran espacio de intercambio. Esto también tiene la ventaja de acelerar el sistema, ya que la velocidad de lectura/escritura de los discos duros es mucho más lenta que la de la RAM

**Swappines**: es una propiedad del núcleo de Linux, que permite ajustar el equilibrio entre el uso de espacio de intercambio (swap) y la memoria de acceso aleatorio (RAM). El swappines puede tomar valores desde el 0 hasta el 100. Si se establece 0 el núcleo intentara no hacer intercambio, mientras que si se establece 100 en el sistema intentara mantener la memoria de acceso aleatorio lo mas libre posible haciendo intercambio.



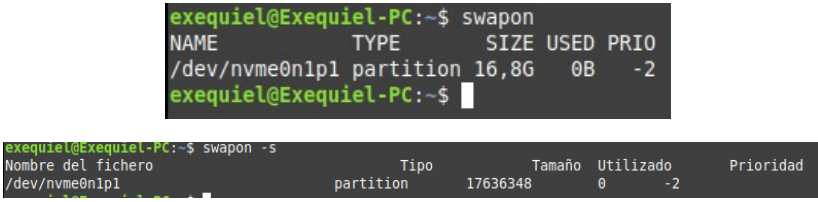


~ **swapon** -> Lista de la swap disponibles con sus tamaños y prioridad

~ **swapon -s** -> Lista la swap disponibles y el tipo (archivo/partición)

~ **swapon -a** -> Para habilitar todas las áreas de intercambio

~ **swapoff -a** -> Para deshabilitar todas las áreas de intercambio



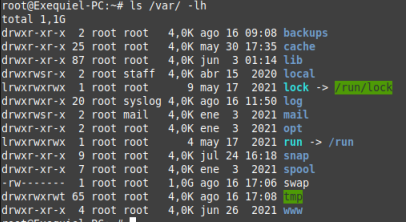
**Incorporamos otra swap de tipo archivo**

***Creamos el archivo de 16GB con el comando dd:***

dd if=/dev/zero of=/var/swap bs=4096k count=256

**Establecemos permisos limitados (por seguridad):**

Chmod 600 /var/swap



**Incorporamos una swap de tipo de archivo:**

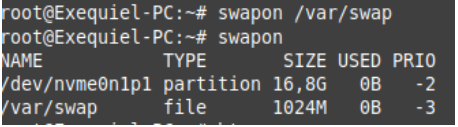
Establecemos el formato de mkswap:

mkswap /var/swap

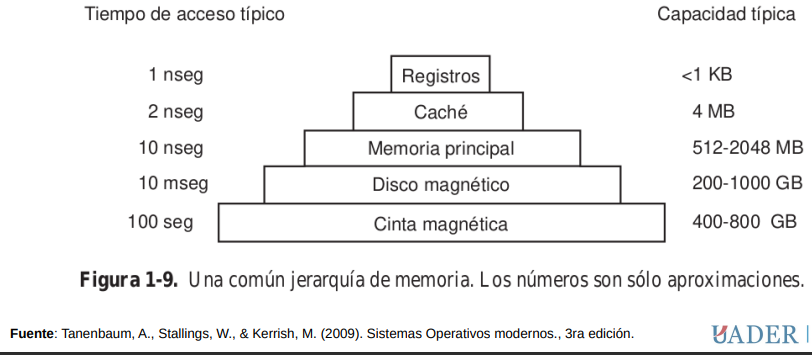


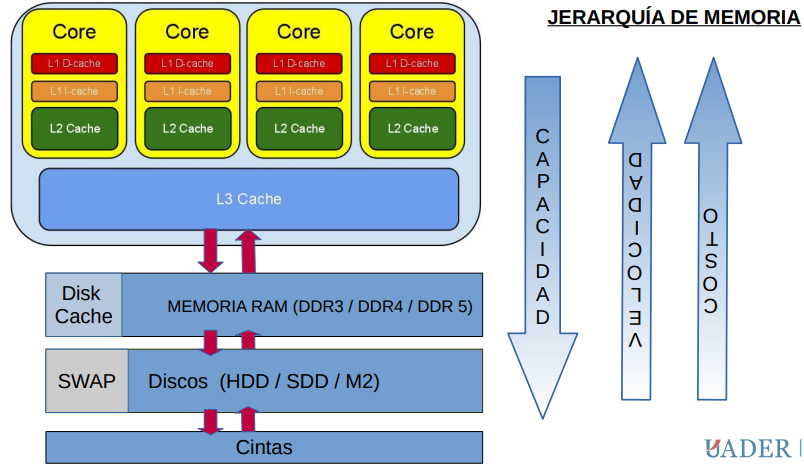
Incorporamos el archivo a la swap:

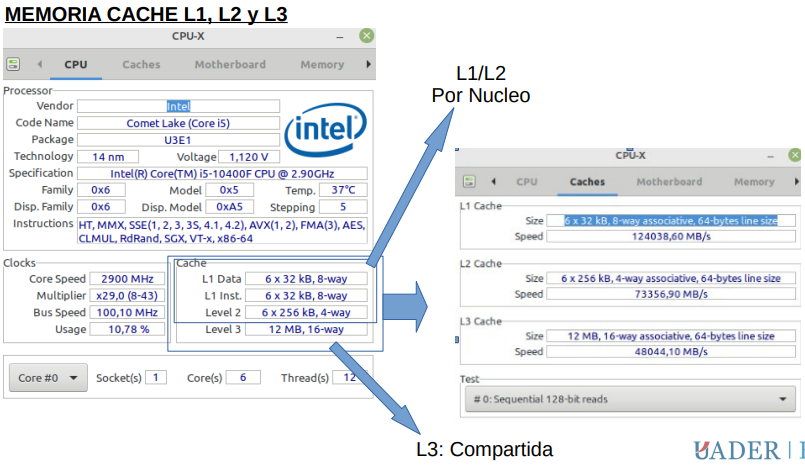
swapon /var/swap

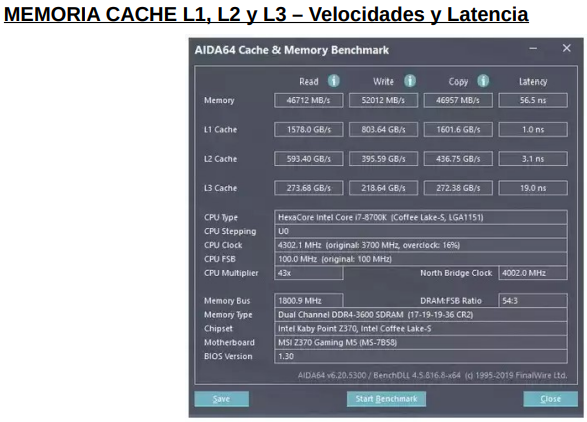


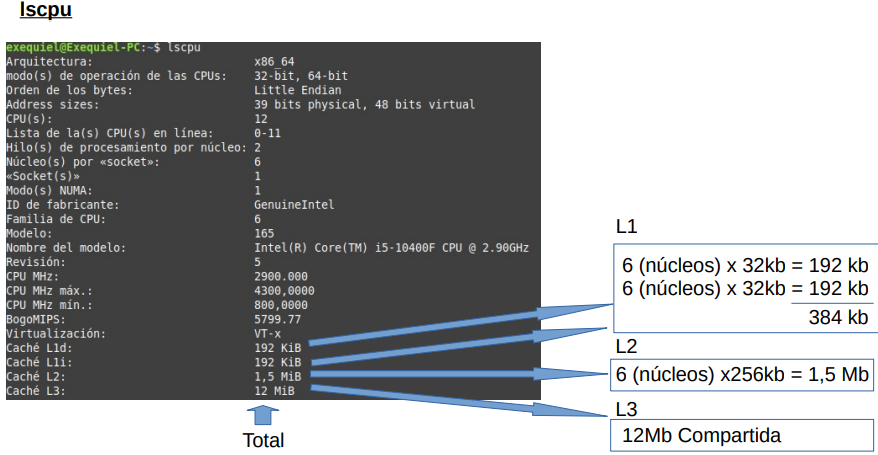
**Jerarquía de memoria:** el sistema de memoria esta construido como una jerarquía de capas. Las capas superiores tienen mayor velocidad, menor capacidad y mayor costo por bit que las capas inferiores, a menudo por factores me mil millones o más.









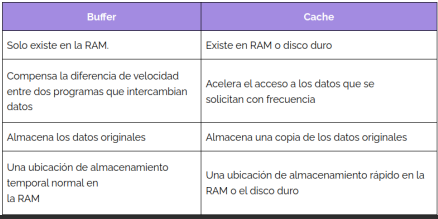


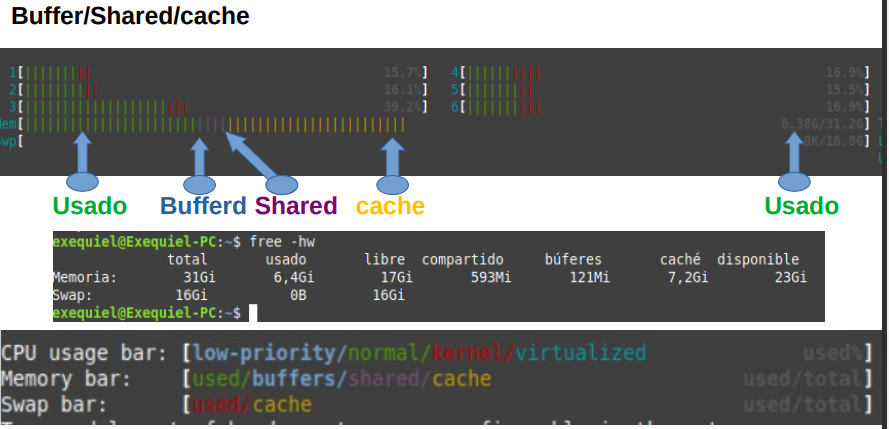
**Buffer/Shared/cache:**

**Cache:** el almacenamiento en cache es el proceso de almacenar temporalmente una copia de un recurso determinado para que las solicitudes posteriores al mismo recurso se procesen más rápido.

**Buffer:** es un área de almacenamiento temporal en la memoria principal (RAM) que almacena la transferencia de datos entre dos o mas dispositivos o entre una aplicación y un dispositivo. El almacenamiento en búfer compensa la diferencia en las velocidades de transferencia entre el remitente y el receptor de los datos.

**Shared:** La memoria compartida, permite que dos o mas procesos compartan una zona común de memoria, representa la forma mas rápida de IPC, debido a que no hay ningún tráfico entre cliente/servidor, si no quedo todo el acceso es directo a la memoria.





(Apreciar los colores de la barra de htop: en verde es lo usado, lo azul representa el Bufferd, en violeta el Shared y en amarillo el cache)

**Comando sync:** de forma predeterminada, el kernel de Linux escribe datos en el disco de forma asíncrona. Las escrituras se almacenan en búfer (caché) en la memoria y se escriben en el dispositivo de almacenamiento en el momento óptimo. El comando de sincronización fuerza una escritura inmediata de todos los datos almacenados en caché en el disco. Ejecute la sincronización si prevé que el sistema sea inestable o que el dispositivo de almacenamiento deje de estar disponible repentinamente y desea asegurarse de que todos los datos se escriban en el disco. Se pueden sincronizar archivos individuales o todo el sistema de archivos que contiene los archivos especificados. Si no se proporcionan argumentos, se sincronizan todos los sistemas de archivos montados.

**¿Cómo borrar el caché en Linux?**

Cada sistema Linux tiene tres opciones para borrar el caché sin interrumpir ningún proceso o servicio.

**1.Borre PageCache únicamente.**

#sync; echo 1 > /proc/sys/vm/drop\_caches

**2.Borrar Dentries e inodos.**

#sync; echo 2 > /proc/sys/vm/drop\_caches

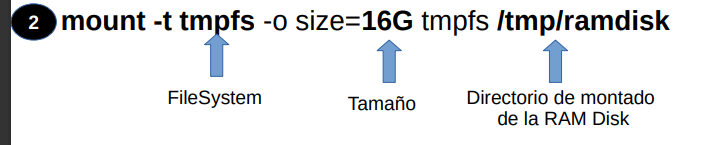
**3.Borrar cache de página, dentries e inodos.**

#sync; echo 3 > /proc/sys/vm/drop\_caches

**RAM Disk – Utilizando tmpfs**

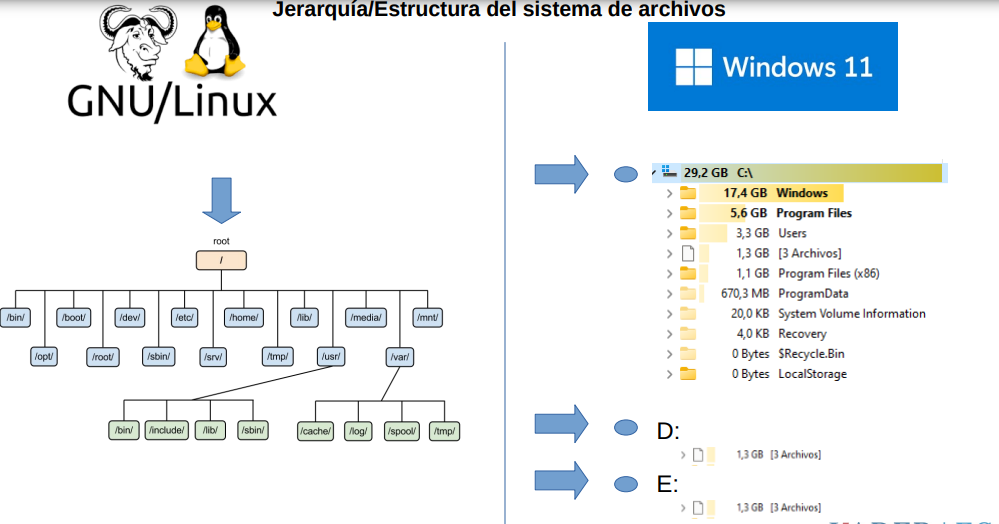
**Tmpfs** es un Sistema de archivos que guarda todos sus archivos en la memoria virtual. Dado que tmpfs vive completamente en la memoria caché de la página y en el intercambio, todas las páginas de tmpfs se mostrarán como “Shmem” en /proc/meminfo y shared (compartido en español) en free.

1. **mkdir** /tmp/ramdisk -> crear un directorio vacío.
2. **mount -tmpfs** -o size=**16BG** tmpfs **/tmp/ramdisk**



**IMPORTANTE:** Los datos desaparecerán cuando reinicies o apagues el equipo, puesto que la RAM es una memoria volátil. Por eso, lo que quieras conservar, hazle una copia en una partición del disco duro.

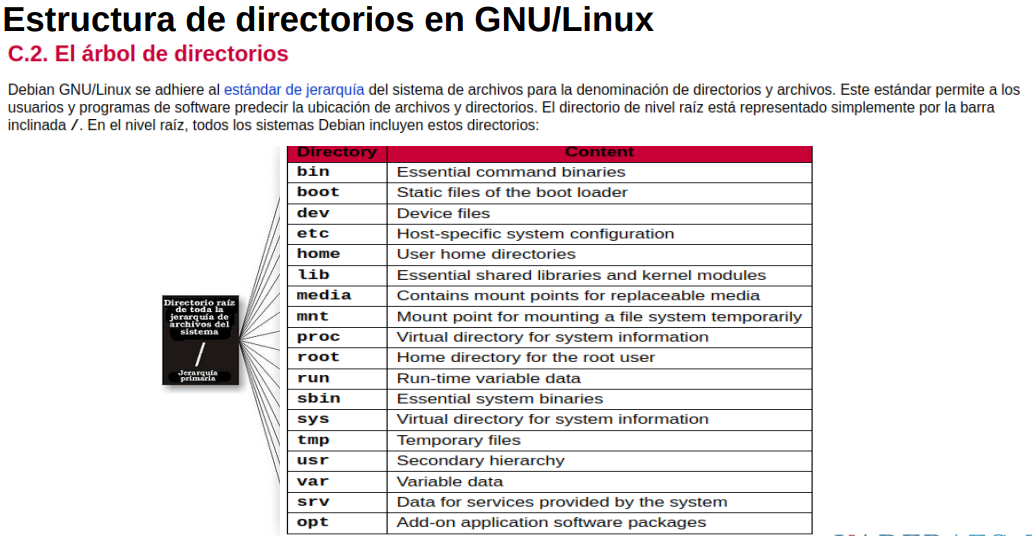
**Estructura de directorios:** la estructura de los directorios es un árbol jerárquico.



**Estándar de jerarquía del sistema de archivos (Filesystem Hierarchy Standard - FHS)¿Qué es?** El estándar de jerarquía del sistema de archivos (en inglés Filesystem Hierarchy Standard, también conocido por sus siglas FHS) es una norma que define los directorios principales y sus contenidos **en el sistema operativo GNU/Linux y otros sistemas de la familia Unix.**

**¿Para qué se diseñó?**  Se diseñó originalmente en 1994 para estandarizar el sistema de archivos de las distribuciones de Linux, basándose en la tradicional organización de directorios de los sistemas Unix. En 1995 se amplió el ámbito del estándar a cualquier Unix que sea voluntariamente.

El 3 de junio de 2015 se lanzó FHS 3.0



**Estructura de directorios en GNU/Linux:**

**/**  -> Directorio raíz del sistema. De aquí “cuelgan” todos los directorios del sistema

**/bin**  -> Es una abreviatura de binarios o ejecutables. Aquí se encuentran los diferentes dispositivos del sistema.

**/dev** En este directorio están los archivos que representan los diferentes dispositivos del sistema.

**/etc** contiene todos los archivos de configuración del sistema y de cada uno de los programas.

**/sbin** Aquí encontraras utilidades para la administración del sistema. Estas utilidades son accesibles solamente para el root o super usuario.

**/home** Este directorio contiene los subdirectorios personales de los usuarios del sistema. Cada usuario posee su propio directorio en el que puede almacenar archivos personales, documentos, programas y archivos de configuración.

**/lib** el directorio lib posee librerías compartidas utilizadas por el sistema a la hora de arrancar y correr comandos propios.

**/proc** Los archivos que contiene residen en memoria, hacen referencia a procesos e información del sistema.

**/usr** es la segunda mayor sección del sistema de archivo. Contiene archivos de documentación, librerías, código fuente, binarios usados por programas que no son del sistema.

**/var** en este directorio encontraremos datos variables como pueden ser: archivos de logs, temporales, spool de impresión, etc.

**/tmp** Contiene archivos temporales del sistema. Los programas pueden escribir en este directorio datos que necesiten mientras estén en ejecución.

**/root** Éste es el directorio personal de usuario root.

**¿Qué es un sistema de archivos?**

Es un modelo abstracto que permite manipular los datos almacenados en un espacio de disco de una forma determinada.

**¿Qué nos permite hacer un sistema de archivos?**

Un tratamiento consistente de la información almacenada físicamente en disco. Una jerarquía de archivos y protección de los archivos.

**¿Qué es un sistema de archivos con Journaling?**

Es un sistema que sirve para da mayor seguridad a la integridad de los datos que contenemos en nuestros discos rígidos. Se recomienda para sistemas de alta disponibilidad.

**Algunos sistemas de archivos sin Journaling**

Ext2

Diseñado originalmente por Rémy Card. Desaroolado para GNU/Linux. Los limites son un máximo de 2 TB de archivo, y de 4 TB de partición.

FAT (12/16/32)

Diseñado por Bill Gates y Matc McDowland en 1977. Desarrollado para DOS/Windows. El tamaño máximo de un archivo FAT32 es de 4 GB.

**Algunos sistemas de archivos con Journaling**

**ext3**

Diseñado por la Comunidad de Software Libre. Desarrollado para GNU/Linux. Los límites son un máximo de 2 TB de archivo, y de 16 TB de partición.

**ReiserFS**

Diseñado por la empresa Namesys, liderado por Hans Reiser. Desarrollado para GNU/Linux. Los límites son un máximo de 8 TB de archivo, y de 16 TB de partición.

**XFS**

Diseñado por la empresa SGI. Desarrollado para Unix IRIX y GNU/Linux. Los límites son un máximo de 8 EB de archivo, y de 9 EB de partición.

**JFS**

Diseñado por por la empresa IBM. Desarrollado para Unix AIX y GNU/Linux. El tamaño máximo de partición va de 512 TB hasta 4 PB (en función del block size).

**NTFS**

Diseñado por Microsoft. Desarrollado para Windows NT/2000/XP/Vista. Los límites son un máximo de 16 TB de archivo, y de 256 TB de partición.

**ext4**

Diseñado por Mingming Cao, Dave Kleikamp, Alex Tomas, Andrew Morton y otros programadores. Desarrollado para GNU/Linux. Los límites son un máximo de 16 TB de archivo, y de 1 EB de partición. Deja de ser experimental en la versión de Kernel 2.6.28.

**Montado de sistemas de archivos:**

GNU/Linux no trabajo con unidades, entonces, para poder acceder a un dispositivo (disco rigido, cdrom, pendrive) necesitamos incorporarlo a nuestro sistema de archivos.

Para eso nos valemos de los comandos **mount** y **umount.**

Ellos se encargan de poder acceder a un dispositivo, a través de un directorio en nuestro sistema (normalmente ubicados en /media)

**Tipos de archivo en GNU/Linux**

* **Archivos o ficheros ordinarios:** estos pueden ser archivos de textos, documentos ejecutables, etc.
* **Directorios:** también conocidos como carpetas y son tipos de archivos especiales que agrupan otros ficheros de forma estructurada.
* **Archivos de dispositivos:** representan dispositivos conectados al sistema. Se dividen en tres clases:
  + **Archivos de dispositivos:** representan dispositivos conectados al sistema. Se dividen en tres clases:
    - **Seriales:** se lee o escribe carácter por carácter, ej: teclados, mouses y MODEM’s.
    - **Bloque:** permiten leer y/o escribir datos por bloques en forma aleatoria, ej: discos rígidos, disquetes, memorias, etc.
    - **Especiales:** se utilizan en pantallas, dispositivos gráficos, etc

**Tipos de dispositivos en GNU/Linux**

* /dev/hd**X** (hard device) -> Dispositivos **IDE**
* /dev/plip**X** -> Puertos **Paralelos**
* /dev/ttyS**X** -> Puerto **Serie**
* /dev/fd**X** (floppy device)
* /dev/sd**X** (scsi device) -> Dispositivos **IDE o SCSI**, dispositivos **USB**
* /dev/nvme0nX -> dispositivos **M2**

**Comandos para el manejo de archivos:**

**mkdir -** crea directorios

Sintaxis: mkdir <opciones> <directorios>

*Opciones:*

**-p:** crea la estructura completa de directorio

**-v:** imprime un mensaje por pantalla de cada directorio creado

Ejemplos: $mkdir -p /home/usuario/dir1/subdir1

$mkdir -v /home/usuario/dir2

**rm ­–** remueve archivos o directorios

Sintaxis: rm <opciones> <archivos>

*Opciones:*

**-r:** remueve un directorio en forma recursiva, eliminando todo su contenido

**-i:** pide confirmación antes de remover el archivo

**-f:** no pide confirmación para remover

Ejemplos: $rm -r /home/usuario/dir1 // $rm -i /home/usuario/dir2

**cp** – copia archivos y directorios

Sintaxis: **cp** <opciones> origen destino

Opciones:

**-v:** imprime por pantalla los archivos copiados

**-r:** permite copiar recursivamente el contenido de un directorio

**-i:** pide confirmación antes de sobrescribir un archivo

Ejemplos: $cp -v /etc/hosts /home/usuario/

$cp -i /etc/hosts /home/usuario/

**mv** – mueve (renombra) archivos y directorios

Sintaxis: **mv** <opciones> origen destino

Opciones:

**-v**: imprime por pantalla los archivos movidos

**-f**: borra los archivos existentes sin pedir confirmación

**-i**: pide confirmación al usuario antes de sobreescribir archivos

Ejemplos:

$mv -i /home/usuario/hosts /home/usuario/archivo-hosts

$mv -v /home/usuario/archivo-hosts /home/usuario/hosts

**ln** – crea enlaces entre archivos

Sintaxis: **ln** <opciones> destino nombre-enlace

Opciones:

**-v**: imprime por pantalla los archivos enlazados

**-s**: crea un enlace simbólico

**-i**: pide confirmación al usuario antes de sobreescribir archivos

Ejemplos:

$ln -s /home/usuario/hosts /home/usuario/enlace-hosts

$ln -v /home/usuario/hosts /home/usuario/otro-enlace-hosts

**mount** – monta un sistema de archivos

Sintaxis: **mount** <opciones> <dispositivo> <directorio>

Opciones:

**-o**: permite especificar opciones de montado (solo lectura, lectura/escritura)

**-a**: monta todos los sistemas de archivos mencionados en fstab

**-t**: permite especificar el tipo de sistema de archivos (vfat, ext2, ext3, reiserfs)

Ejemplos: $mount // $mount -t iso9660 /dev/hdc /mnt

**umount** – desmonta un sistema de archivos

Sintaxis: **umount** <opciones> <dispositivo> <directorio>

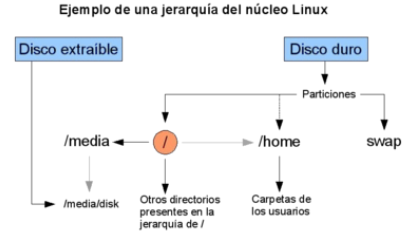
Opciones:

**-n**: desmonta un sistema de archivos sin modificar /etc/mtab

**-a**: desmonta todos los sistemas de archivos mencioados en mtab

**-t**: permite especificar el tipo de sistema de archivos (vfat, ext2, ext3, reiserfs)

Ejemplos: $umount -a // $umount /mntD



**Cómo linux maneja el montado de sistemas de archivos**

Utiliza el archivo /etc/fstab (filesystem table)

En el se especifica la ubicación de cada sistema de archivos y sus opciones de montado.

Se usa principalmente de la carga del sistema operativo para el montado de particiones en el árbol de directorio de Linux.

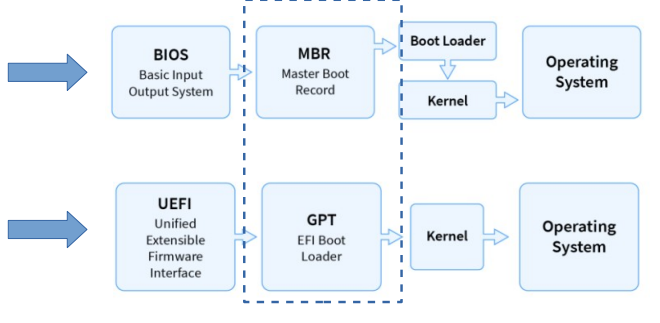
Sirve como atajo a la hora de montar dispositivos o sistemas de archivos.

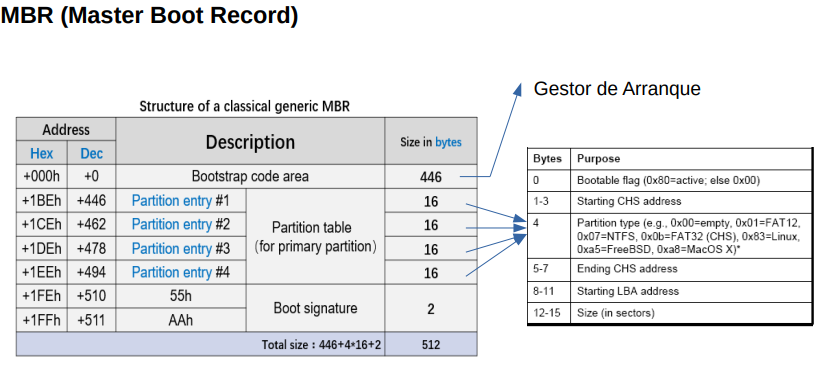
**Particiones:** Una partición de disco, es el nombre genérico que recibe cada división presente en una sola unidad física de almacenamiento de datos. Los dispositivos de almacenamiento se pueden dividir en una o más particiones, con sistemas de archivos independientes en cada partición.

Existen distintas tablas de partición (ej:**MBR(msdos)** y **GPT(gpt)**), y distintos tipos de particiones según la tabla.

Las particiones pueden utilizarse para varios fines, puedes tener una dedicada a guardar datos sensibles con medidas de seguridad que no interfieran en el resto del sistema, como copias de seguridad, o utilizarla para instalar diferentes sistemas operativos o dividirlo.

**Tabla de particiones:**





**Tablas de particiones GUID (GPT):** La tabla de partición GUID (GPT) es un esquema de partición basado en el uso de un **identificador único global (GUID)**. La GPT se desarrolló para hacer frente a las limitaciones de la tabla de partición **MBR**, especialmente con el limitado espacio de almacenamiento máximo direccionable de un disco. A **diferencia** de **MBR**, que no puede direccionar un almacenamiento mayor de **2 TiB (equivalente a unos 2,2 TB)**, GPT se utiliza con discos duros de mayor tamaño; el tamaño **máximo** **direccionable** del disco **es** de **2,2 ZiB**. Además, GPT, por defecto, permite crear hasta **128 particiones primarias**. Este número puede ampliarse asignando más espacio a la tabla de particiones.

**Partes de los discos:**

Los discos GPT utilizan el **Direccionamiento Lógico de Bloques** (LBA Logical Block Addressing) y la disposición de las particiones es la siguiente:

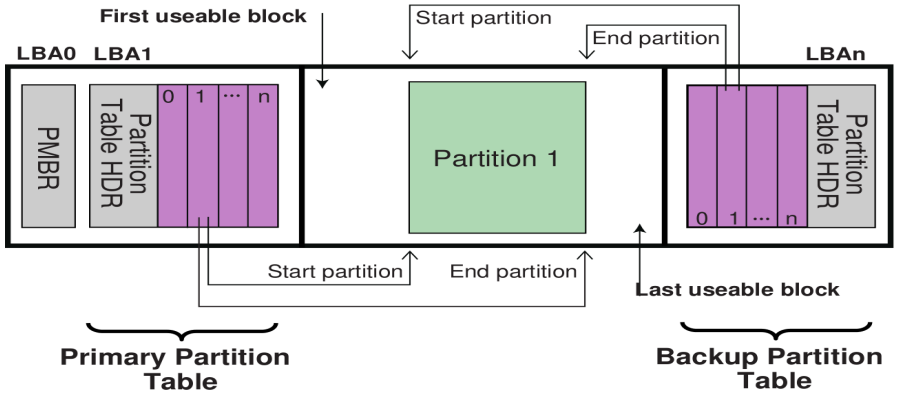
● **Para mantener la compatibilidad** con los discos MBR, el primer sector **(LBA 0)** de GPT está reservado para los datos de MBR y se llama "MBR protector".

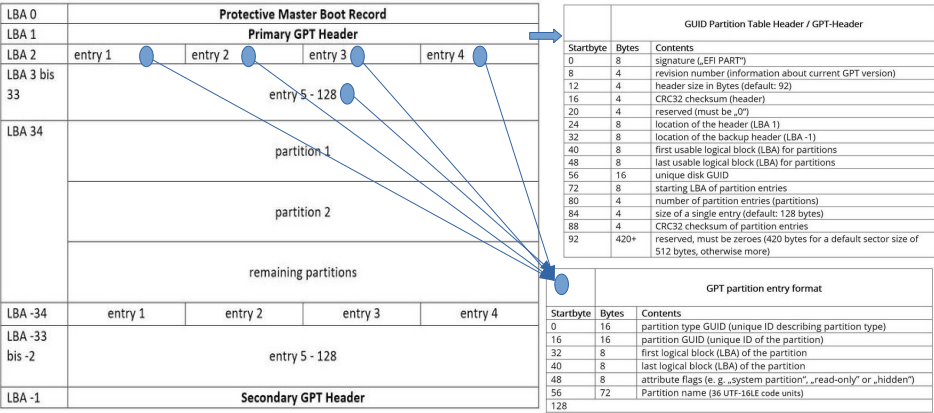
● **La cabecera GPT primaria** comienza en el segundo bloque lógico **(LBA 1)** del dispositivo. La cabecera contiene el GUID del disco, la ubicación de la tabla de particiones primaria, la ubicación de la cabecera GPT secundaria, y las sumas de comprobación CRC32 de sí misma y de la tabla de particiones primaria. También especifica el número de entradas de partición en la tabla.

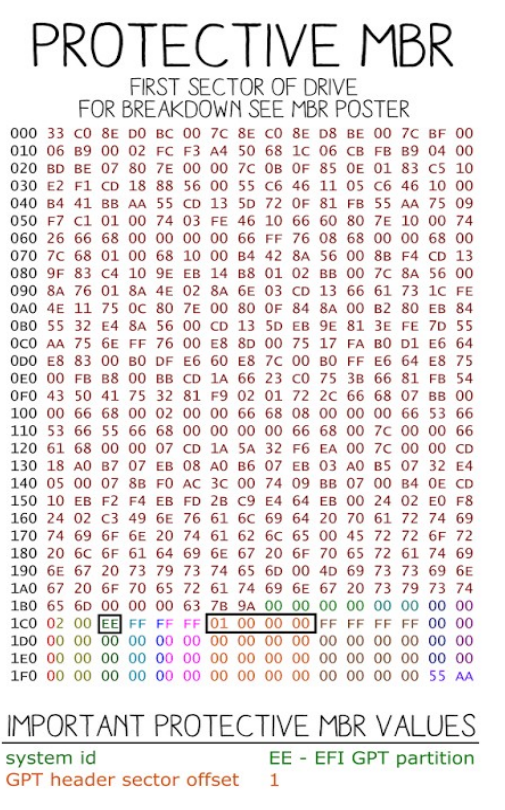
● **La GPT primaria incluye comienza en bloque LBA 2 hasta LBA 33**, por defecto 128 entradas de partición, cada una con un tamaño de entrada de 128 bytes, su GUID de tipo de partición y su GUID de partición única.

● **La GPT secundaria** es idéntica a la GPT primaria. Se utiliza principalmente como una tabla de respaldo para la recuperación en caso de que la tabla de partición primaria se corrompa.

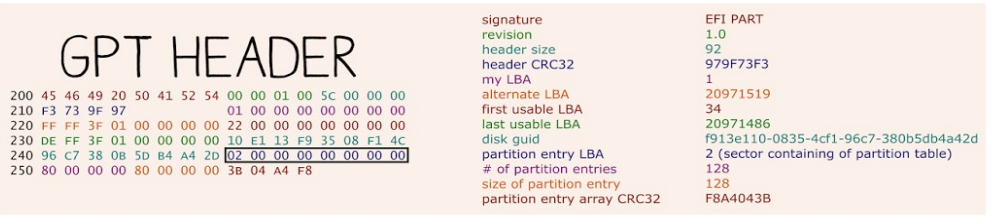
● **La cabecera secundaria de GPT** se encuentra en el último sector lógico del disco y puede utilizarse para recuperar la información de GPT en caso de que la cabecera primaria esté dañada. Contiene el GUID del disco, la ubicación de la tabla de particiones secundarias y la cabecera GPT primaria, las sumas de comprobación CRC32 de sí mismo y de la tabla de particiones secundarias, y el número de posibles entradas de partición.





**MBR**

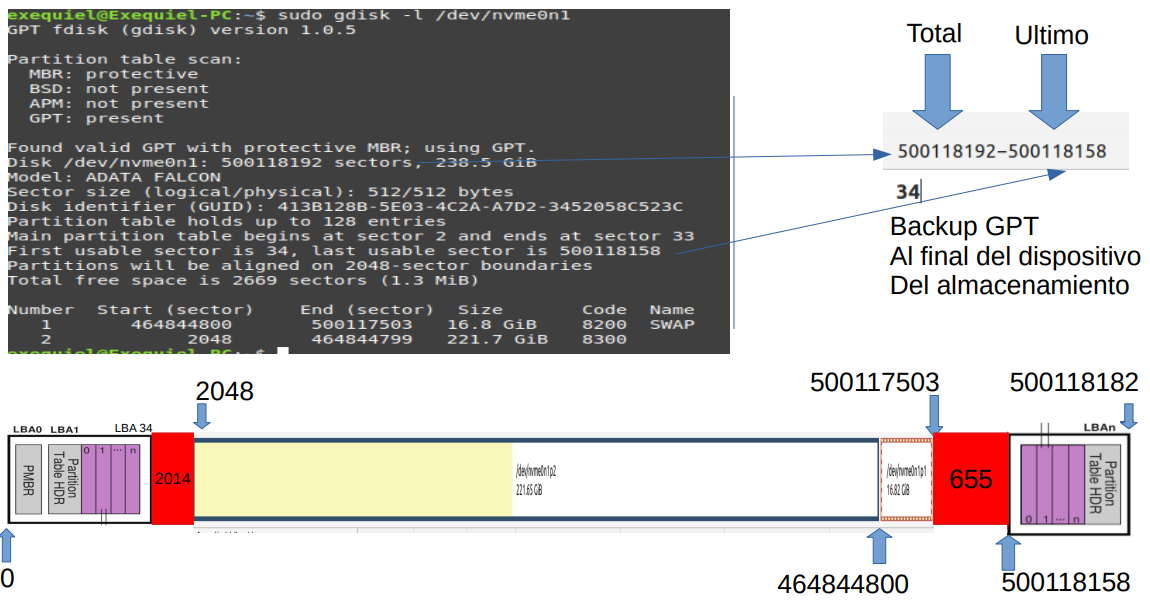
**GPT:**

****

**¿Por qué no se ocupan todo el espacio del disco?**

**GPT** al final del disco reserva espacio para dejar lugar para un backup

**Comando: sudo gdisk -l /dev/device\_name**



**¿Por qué si compras un disco de 500GB, en realidad son 465 aprox?**

Esto se debe a que muchos usuarios usan distintos tipos de mediciones

(binario) y otros (decimal), ambos dan un resultado parecido, pero no igual.

Ademas de esto, también la pc calcula las capacidades distintas a como la calculan las empresas:

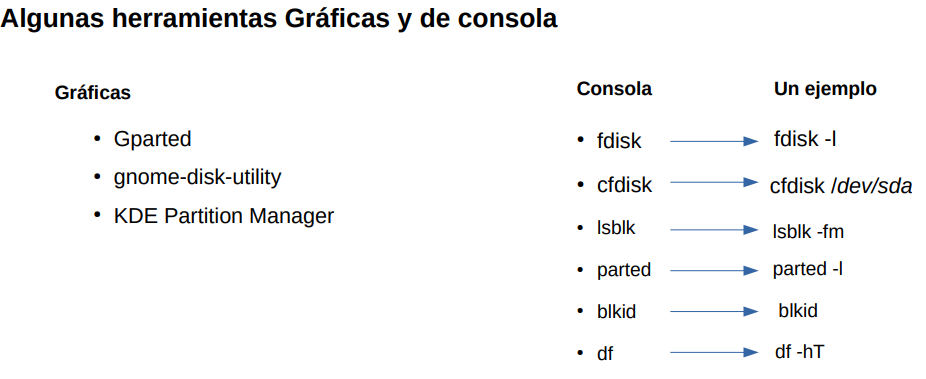
Por ejemplo, si el embalaje muestra una capacidad de 500 GB, se calcula de la siguiente manera: 500 GB x 500 x 1000 MB x 1000 KB x 1000 Byte x 500 000 000 000 bytes. El sistema operativo de su PC se calculará de la siguiente manera: 500 000 000 000 bytes/1024 KB/1024 MB/1024 GB, o aproximadamente, 465 GB.

Esquema de particiones recomendado por RedHat:

* Una partición **swap**
* Una partición **/boot**
* Una partición **/**
* Una partición **/home**

**Espacio de swap recomendado:**

**­**

****

**fdisk**: El comando fdisk es una**utilidad basada en texto para visualizar y administrar particiones de disco duro** en Linux

**cfdisk:** El comando se utiliza para crear, eliminar y modificar particiones en un dispositivo de disco

**lsblk:** El comando lsblk en Linux enumera los dispositivos de bloque.

**parted:** herramienta de línea de comandos que le permite administrar fácilmente las particiones del disco duro.

**blkid:** nos muestra una lista de particiones con información tal como: nombre de dispositivo de bloque, UUID (Universally Unique IDentifier), Etiqueta y tipo de sistema de archivos.

**df:** muestra el uso de espacio en disco en todos los sistemas de ficheros montados actualmente.

**comandos raid**

mdadm -D /dev/md0 -> para realizar las funciones necesarias relacionadas a la administración de conjuntos de múltiples dispositivos

mdadm /dev/md0 --add /dev/sda1

**Comandos de disco**

fdisk -l /dev/sda

cfdisk /dev/sda -> el mejorcito

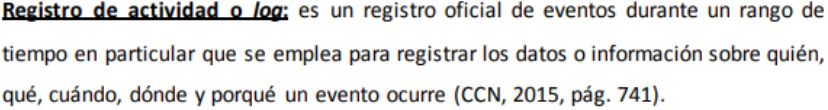
lsblk -lf

no empieza en 513mb el mbr y arranca en 2048, por simetria (casi siempre)

se pierden datos

**Logs**

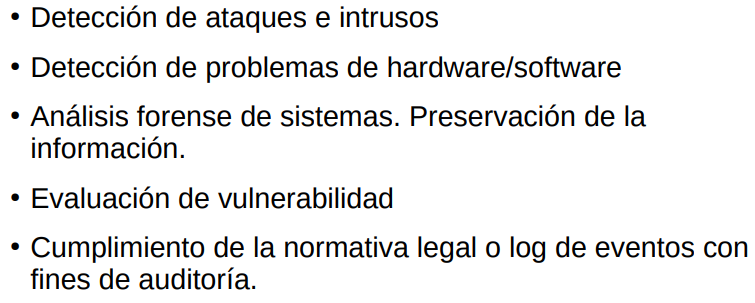
Los logs son registros de los eventos que se generan en los servidores, aplicaciones, redes y sistemas de una organización. Cada uno de estos archivos contiene información relacionada a un evento específico que ocurrió dentro de un equipo, sistema o red.(Vieda, 2013)

****

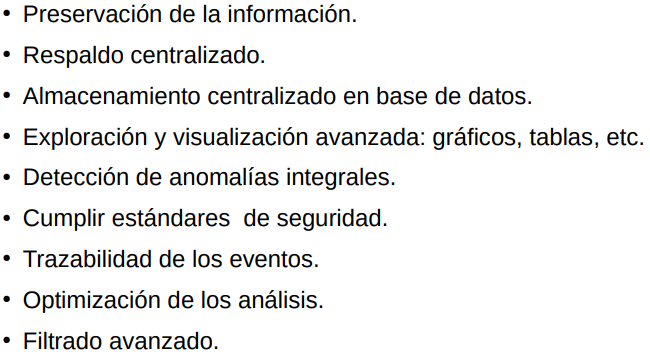
**Logs de sistema:** Generalmente generados por el demonio (ej:Rsyslogd). Registran información relacionada con el funcionamiento del sistema operativo. Algunos ejemplos de logs de sistema son los que registran información sobre los servicios, los que registran los accesos al equipo, los mensajes del sistema, etc.

**Logs de programas:** Son aquellos que registran cronológicamente los eventos más importantes mientras estamos usando una aplicación o programa. En este caso, los logs pueden ser generados por la propia aplicación o por un demonio (ej:Rsyslogd).

**Ventajas de logs:**



**Ventajas servidor de log:**



**Rotacion de logs:**

Cuando hablamos de rotación de logs, nos referimos a lo siguiente:

**1)**Determinar el tamaño máximo permitido para un archivo de log. **2)**Cuando se alcanza dicho tamaño, borrar el archivo, renombrarlo o comprimirlo y crear uno nuevo. **3)**Especificar por cuánto tiempo deseamos mantener los registros, ya sea comprimidos o no.

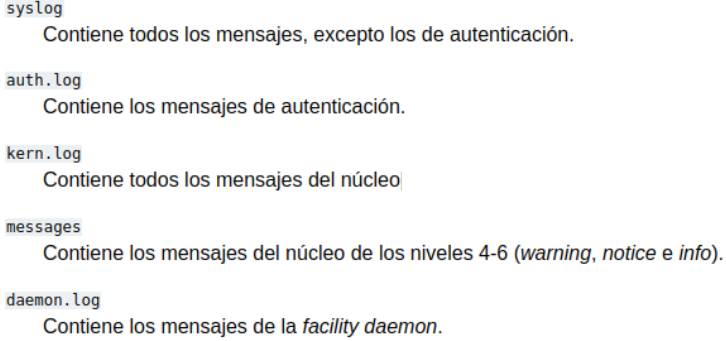


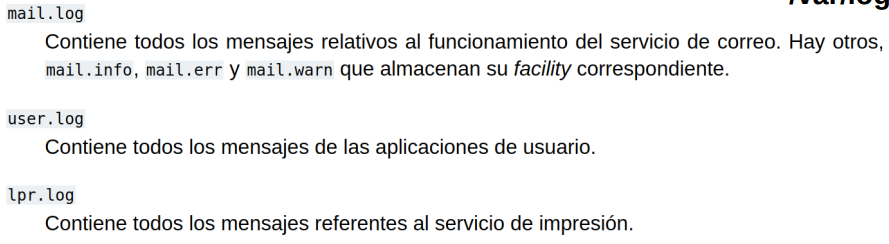
**GNU/Linux -> Logrotate**

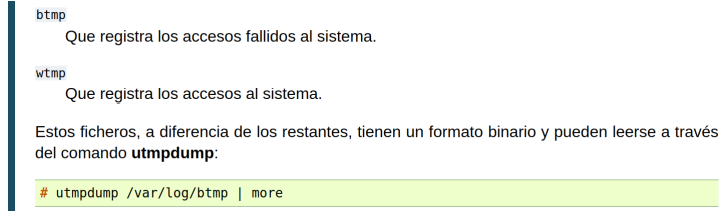
es una utilidad de sistema que administra la compresión y rotación de archivos de logs en sistemas Linux. Si los logs no se rotan, comprimen y depuran de manera periódica, eventualmente pueden consumir todo el espacio en disco disponible en el sistema

*esta en /etc/logratate.conf*

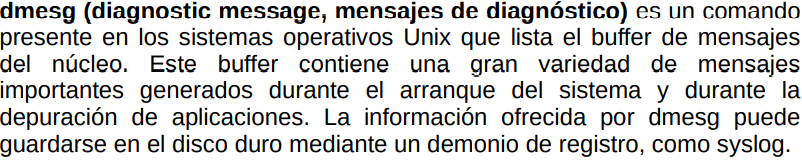
**Archivos** de registros mas importantes en GNU/Linux -> /**var/log**

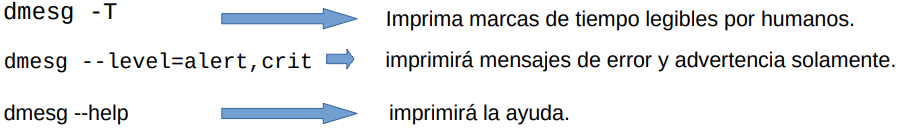






**Comando dmesg:**





**Herramientas gráficas de logs:**

Gnome Logs.

Ksystemlog: muestra todos los registros del sistema, agrupados por servicios generales.