

832 lines (505 loc) · 56.1 KB

Preview Code Blame



Practica 1

[Siguiete](#)
[Anterior](#)

El objetivo de esta práctica es que el alumno se familiarice con los conceptos básicos del sistema operativo GNU/Linux, así como con su entorno y comandos principales.

- [Preguntas/Dudas](#)
- [1\) Características de GNU/Linux](#)
- [2\) Distribuciones de GNU/Linux](#)
- [3\) Estructura de GNU/Linux](#)
- [4\) Kernel](#)
- [5\) Intérprete de comandos \(Shell\)](#)
- [6\) Sistema de Archivos \(File System\)](#)
- [7\) Particiones](#)
- [8\) Arranque \(bootstrap\) de un Sistema Operativo](#)
- [9\) Archivos](#)
- [10\) Indique qué comando es necesario utilizar para realizar cada una de las siguientes acciones.](#)
- [11\) Investigue su funcionamiento y parámetros más importantes](#)
- [12\) Investigue su funcionamiento y parámetros más importantes](#)

1) Características de GNU/Linux:

a) Mencione y explique las características más relevantes de GNU/Linux.

Es multiusuario

Cumple simultáneamente las necesidades de dos o más usuarios que comparten los mismos recursos, un sistema multiusuario permite que los usuarios accedan a los datos o procesos de una única maquina, desde diferentes computadoras o terminales

Es multitarea y multiprocesador

Puede realizar varias operaciones, funciones o ejecuciones al mismo tiempo

Los SO multitarea permiten realizar varias operaciones gracias a una operación denominada cambio de contexto que actúa de la siguiente manera, quita un proceso del CPU, ingresa uno nuevo, y luego vuelve a ingresar el proceso que quitó del CPU en una especie de cola de ejecución, sin que el procesador se entere de todo lo que está pasando, de modo, que pueda realizar varias tareas simultáneas. Es como una especie de engaño, que permite mejorar el rendimiento de los ordenadores y facilitar el trabajo a los usuarios.

Multiprocesador se refiere al número de procesadores del sistema, que es más de uno y éste es capaz de usarlos todos para distribuir su carga de trabajo.

Es altamente portable

Se puede usar en cualquier ordenador que tenga el SO para el que fue programado sin la necesidad de una instalación previa; significa que no hace falta la instalación de bibliotecas adicionales en el sistema para que pueda funcionar.

Posee diversos intérpretes de comandos, de los cuales algunos son programables

- Es el método de interacción entre el usuario y la máquina (shell)
- CLI (interfaz de línea de comando) un ejemplo sería la terminal bash.

Permite el manejo de usuarios y permisos

Nos permite darle permisos particulares a usuarios y archivos del file system.

Todo es un archivo (hasta los dispositivos y directorios):

Todo es un fichero. Los directorios son ficheros, los ficheros son ficheros, y los dispositivos son ficheros. A veces a los dispositivos se les llama nodos, pero siguen siendo ficheros.

Los sistemas de ficheros de GNU/Linux se organizan en una estructura jerárquica, de tipo árbol.

El nivel más alto del sistema de ficheros es / o directorio raíz. Todos los demás ficheros y directorios están bajo el directorio raíz. Por ejemplo, `/home/jebediah/cheeses.odt` muestra la ruta completa al fichero `cheeses.odt` que está en el directorio `jebediah`, que a su vez está bajo el directorio `home`, que por su parte está bajo el directorio raíz (/).

Cada directorio puede estar en una partición diferente (/temp, /home, etc.)

En Linux, el sistema de archivos se organiza en una estructura de árbol con una raíz denominada "/" (barra). Dentro de esta estructura de árbol, cada partición puede montarse en un directorio diferente, lo que significa que los archivos y directorios que se encuentran dentro de esa partición se harán accesibles a través del directorio montado.

Por ejemplo, si tienes una partición llamada `/dev/sda1` y quieres montarla en el directorio `/home`, los archivos y directorios en esa partición estarán disponibles en el directorio `/home`. Si tienes otra partición llamada `/dev/sda2` y quieres montarla en `/temp`, los archivos y directorios en esa partición estarán disponibles en el directorio `/temp`.

En resumen, en Linux, cada partición puede montarse en un directorio diferente en la estructura de árbol de archivos del sistema, lo que permite una mejor organización y gestión de los datos.

Es case sensitive

Sensible a las mayúsculas y minúsculas

Es código abierto

Software cuyo código fuente y otros derechos que normalmente son exclusivos para quienes poseen los derechos de autor, son publicados bajo una licencia de código abierto o forman parte del dominio público. En las licencias compatibles con la Open Source Definition el propietario de los derechos de autor permite a los usuarios utilizarlo, modificarlo, estudiarlo, aprovechar su documentación, redistribuir el software, a cualquiera, para cualquier propósito, ya sea en su forma modificada o en su forma original etc.

b) Mencione otros sistemas operativos y compárelos con GNU/Linux en cuanto a los puntos mencionados en el inciso a.

- A diferencia de Windows y Mac, Linux no pertenece a ninguna compañía, sino que su desarrollo depende de la colaboración de un gran número de empresas y profesionales.
- Su uso esta mas enfocado a programadores dado que sus interfaces son menos amigables.
- Los demás sistemas suelen ser en su mayoría pagos y enfocados mas a lo comercial
- El manejo de permisos de Linux vuelve mas difícil la creación de un virus para dicho sistema mientras que en Windows es mas común.

c) ¿Qué es GNU?

GNU es un SO de tipo Unix pero libre, diseñado por miles de programadores, así como una gran colección de programas informáticos que componen al sistema, desarrollado por y para el GNU.

Está formado en su totalidad por software libre, mayoritariamente bajo términos de copyleft. GNU es el acrónimo recursivo de "GNU's Not Unix" (**GNU no es Unix**), nombre elegido debido a que GNU sigue un diseño tipo Unix y se mantiene compatible con este, pero se distingue de Unix por ser software libre y por no contener código de Unix.

d) Indique una breve historia sobre la evolución del proyecto GNU

- El proyecto GNU fue iniciado por Richard M. Stallman con el propósito de crear un sistema operativo completo y libre: el sistema GNU.
- Se baso principalmente en 4 libertades
 - **Libertad 1:** la libertad para ejecutar el programa con cualquier fin.
 - **Libertad 2:** La libertad de estudiar cómo trabaja el programa y de adecuarlo para que haga lo que usuario desee
 - **Libertad 3:** la libertad de redistribuir el programa de manera de colaborar con el resto de la sociedad.
 - **Libertad 4:** la libertad de hacer pública y distribuir a terceros la versión mejorada.

Pondria más pero a nadie le importa la historia de linux salu2.

e) Explique qué es la multitarea, e indique si GNU/Linux hace uso de ella.

Puede realizar varias operaciones, funciones o ejecuciones al mismo tiempo

Los SO multitarea permiten realizar varias operaciones gracias a una operación denominada cambio de contexto que actúa de la siguiente manera, quita un proceso del CPU, ingresa uno nuevo, y luego vuelve a ingresar el proceso que quitó del CPU en una especie de cola de ejecución, sin que el procesador se entere de todo lo que está pasando, de modo, que pueda realizar varias tareas simultáneas. Es como una especie de engaño, que permite mejorar el rendimiento de los ordenadores y facilitar el trabajo a los usuarios.

Linux es Multitarea: La multitarea no consiste en hacer que el procesador realice más de un trabajo al mismo tiempo (un solo procesador no tiene esa capacidad), lo único que realiza es presentar las tareas de forma intercalada para que se ejecuten varias simultáneamente. Por lo tanto en Linux es posible ejecutar varios programas a la vez sin necesidad de tener que parar la ejecución de cada aplicación.

GNU/Linux si es multitarea.

f) ¿Qué es POSIX?

POSIX (Interfaz de sistema operativo portátil) es un conjunto de interfaces de sistema operativo estándar basadas en el sistema operativo Unix .

Definen una interfaz y un entorno estándar que puede utilizar un sistema operativo para proporcionar acceso a aplicaciones compatibles con POSIX.

El estándar también define un intérprete de comandos (shell) y programas de utilidad comunes . POSIX admite la portabilidad de aplicaciones en el nivel del código fuente , por lo que las aplicaciones se pueden crear para ejecutarse en cualquier sistema operativo compatible con POSIX.

2) Distribuciones de GNU/Linux:

a) ¿Qué es una distribución de GNU/Linux? Nombre al menos 4 distribuciones de GNU/- Linux y cite diferencias básicas entre ellas.

Una distribución es un conjunto de software de GNU, así como programas de sistema, librerías y aplicaciones de usuario, con también software de terceros (a veces), con una versión de un kernel de Linux, todo compilado, empaquetado, configurado y preparado para su instalación en una computadora.

En resumen una distribución de GNU/Linux es el sistema operativo preparado para ser instalado y usado en una computadora.

Desarrollar una distribución es elegir de entre todos los programas de sistema (compiladores, intérpretes, etc.) y aplicaciones de usuario (entornos gráficos de escritorio, editores de texto, paquetes de ofimática, etc.) de GNU y de software de terceros (a veces, no siempre), los que se crean convenientes según la distribución que se quiera crear, más pesada, más ligera, etc., sumado a una versión de un kernel de Linux.

Distribución GNU/Linux = Software GNU + Software de terceros (no siempre) + Kernel de Linux.

Cualquier persona con conocimientos de programación puede crear una distribución GNU/Linux, por eso hay cientos de ellas.

Es tan indispensable el kernel de Linux como el conjunto de programas y librerías de GNU. Por ello cuando a una distribución GNU/Linux se le llama Linux a secas, es una falta de respeto hacia todo el software que GNU ha desarrollado y continua desarrollando como software libre y gratuito.

Debian, Opensuse, Fedora, Ubuntu.

- **Debian:** Se orienta a la estabilidad, en términos de ciclo de desarrollo. Sus actualizaciones y versiones se liberan cuando están listas y bien probadas. Lo cual es bueno para servidores, pero por otra parte nos limita compatibilidad con dispositivos de hardware más novedosos. Son muy exigentes en lo que respecta a Software Libre.
- **Fedora:** Se orienta a innovación con un ciclo de desarrollo corto de seis meses, y un tiempo de soporte de un año sobre sus paquetes. Lo más nuevo se encuentra en Fedora, a veces la gente desconfía de la solidez de sus paquetes sin embargo lo que sucede es que están en fase de maduración y se le integran mejores funcionalidades. Por ahí leí que Fedora es como probar lo que Linux ofrecerá dentro de seis meses.
- **Opensuse:** Novel se quedó sin sistema operativo para su plataforma de servidores y redes, así que compro Suse. Su panel de control para configuración lo hace fácil de configurar. Es una buena elección para aprender de servidores, pero posiblemente para cosas más sofisticadas dejarán las herramientas de configuración y harán ajustes personalizados. El mundo de Software Libre les critica fuertemente sus acuerdos de interoperabilidad con Microsoft.
- **Ubuntu:** Se propuso cambiarle la cara al mundo Linux y ciertamente lo ha logrado. Apuntando a ser fácil de usar hace bien simple la instalación de drivers y decodificadores privativos, cosa que se le critican los fundamentalistas del software libre. Su programa de enviar discos gratis hasta la puerta de tu casa ha significado su creciente popularidad. Tiene ciclo de desarrollo rápido, pero sus versiones tienen ciclos de soporte alternado, entre periodos largos y cortos, tratando de tener estabilidad de soporte y flexibilidad de desarrollo.

b) ¿En qué se diferencia una distribución de otra?

Las mayores diferencias se encuentran sobre todo en distribuciones comerciales, donde se han introducido herramientas para una fácil instalación, detección de software etc...

Diferencias:

- Proceso de instalación.
- Ayudantes o wizard que te ayuden en la configuración de tu máquina.
- Gestores de paquetes.
- Secuencia en el proceso de arranque.

c) ¿Qué es Debian? Acceda al sitio 1 e indique cuáles son los objetivos del proyecto y una breve cronología del mismo.

Debian es una comunidad de desarrolladores , que mantiene un sistema operativo GNU basado en software libre. El sistema se encuentra precompilado, empaquetado y en formato deb para múltiples arquitecturas de computador y para varios núcleos.

El proyecto Debian fue anunciado inicialmente 1993 por Ian Murdock. Debian 0.01 fue lanzado el 16 de agosto de 1993 y la primera versión estable fue hecha en 1996.

Nació como una apuesta por separar en sus versiones el software libre del software no libre. El modelo de desarrollo del proyecto es ajeno a motivos empresariales o comerciales, siendo llevado adelante por los propios usuarios, aunque cuenta con el apoyo de varias empresas en forma de infraestructuras.

Debian no vende directamente su software, lo pone a disposición de cualquiera en Internet, aunque sí permite a personas o empresas distribuirlo comercialmente mientras se respete su licencia.

La primera adaptación del sistema Debian, siendo también la más desarrollada, es Debian GNU/Linux, basada en el núcleo Linux, y como siempre utilizando herramientas de GNU

Los nombres de las versiones de Debian son tomados de la película Toy Story. Hasta la fecha, hay quince versiones estables (con sus respectivas revisiones) enumeradas en la tabla adjunta:

3) Estructura de GNU/Linux:

a) Nombre cuales son los 3 componentes fundamentales de GNU/Linux.

Kernel, shell y Filesystem.

b) Mencione y explique la estructura básica del Sistema Operativo GNU/Linux.

Estructura. Nos ocuparemos de tres componentes principales:

- **Kernel:** El núcleo se podría definir como el corazón de este sistema operativo. Es a grandes rasgos, el encargado de que el software y el hardware de una computadora puedan trabajar juntos.
- **Shell:** Es un programa que recibe lo que se escribe en una terminal de texto y lo convierte en instrucciones para el sistema operativo.
- **Filesystem:** Es la forma lógica en que dentro de un SO se organizan y se administran los archivos.

4) Kernel:

a) ¿Qué es? Indique una breve reseña histórica acerca de la evolución del Kernel de GNU/Linux.

El kernel es la estructura principal del SO, es el que se encarga de que el software y el hardware puedan trabajar juntos.

Es un núcleo monolítico híbrido, y está licenciado bajo la licencia GPL v2.

Evolución

En 1991 Linus Torvalds inicia la programación del kernel Linux basándose en Minix (un clon de Unix desarrollado por Tenenbaum en 1987).

En octubre de 1991 sale la primera versión oficial Linux(0.02) .

En 1992 se une Linux con el proyecto Gnu de Richard Stallman , la versión 1.0 aparece el 14 de marzo del 94' , luego el desarrollo de Gnu/Linux se sigue desarrollando por miles de programadores, en el 96' adoptan como mascota oficial a Tux (pingüinito).

En julio de 1996 se lanza la versión 2.0 y se define la nomenclatura de versionado. Se desarrollo hasta febrero de 2004 y termino con la 2.0.40. En enero de 1999 se lanza la versión 2.2, que provee mejoras de portabilidad entre otras y se desarrolla hasta febrero de 2004 terminando en la versión 2.2.26.

En 2001 se lanza la versión 2.4 y se deja de desarrollar a fines del 2010 con la 2.4.37.11. La versión 2.4 fue la que catapulto a GNU/Linux como un SO estable y robusto.

Durante este período es que se comienza a utilizar Linux más asiduamente Núcleo.

A fines del 2003 se lanza la versión 2.6, Esta versión ha tenido muchas mejoras para el SO dentro de las que se destacan soporte de hilos, mejoras en la planificación y soporte de nuevo hardware.

El 3 de agosto de 2011 se lanza la versión 2.6.39.4 anunciándose la misma desde meses previos como la última en su revisión.

El 17 de julio de 2011 se lanza la versión 3.01, No agrega mayores cambios.

La decisión del cambio son los 20 años del SO y no superar los 40 números de revisión. Totalmente compatible con 2.6. La última versión estable es la 5.19.6 (agosto de 2022).

b) ¿Cuáles son sus funciones principales?

Funciones principales: Administrar la memoria, CPU y E/S, administración de procesos y comunicación y concurrencia.

c) ¿Cuál es la versión actual?

La versión del kernel actual es 5.16. Lo demas a nadie le importa

¿Cómo se definía el esquema de versionado del Kernel en versiones anteriores a la 2.4?

En la versión 2.0 se define la nomenclatura del versionado:

Modo de Versionar: A.B.C.[D]

- A) Denota Versión. Cambia con menor Frecuencia. en 1994 (versión 1.0), en 1996 (versión 2.0) y en 2010(3.0).
- B) Denota Mayor revisión. Antes de la versión 2.6, los números impares indicaban desarrollo, los pares producción.
- C) Denota Menor revisión. Solo cambia cuando hay nuevos drivers o características. ----- la D se agrega a partir de la visión 2.4 -----
- D) Cambia cuando se corrige un grave error sin agregar nueva funcionalidad. (casi no se usan en la rama de versiones 3.x, 4.x y 5.x viéndose reflejado en el componente C del SO).

Ejemplo actualidad: versión 5.19.6

¿Qué cambió en el versionado se impuso a partir de la versión 2.6?

- Antes de la versión 2.6, los números impares indicaban desarrollo, los pares producción.
-

d) ¿Es posible tener más de un Kernel de GNU/Linux instalado en la misma máquina?

Sí. Se puede instalar en la misma máquina, pero 2 núcleos corriendo simultáneamente en Linux es imposible

e) ¿Dónde se encuentra ubicado dentro del File System?

Se encuentra ubicado el `/lib/`: librerías esenciales compartidas y módulos de kernel

Directorio raíz de toda la jerarquía de archivos del sistema.

\ Jerarquia Primaria

- `/bin/` Comandos binarios esenciales de usuario
 - `/boot/` Archivos estáticos del selector de arranque
 - `/dev/` Archivos de unidades
 - `/etc/` Configuración del sistema de Host Especifico
 - `/home/` Directorio "home" de usuario
 - `/lib/` Librerías esenciales compartidas y módulos de Kernel
 - `/media/` Punto de Montaje para medios removibles
 - `/mnt/` Punto de Montaje temporal para sistemas de archivos
 - `/opt/` Agregados de paquetes de Software y Aplicaciones
 - `/sbin/` Binarios de Sistema
 - `/srv/` Datos para los servicios provistos por este sistema
 - `/tmp/` Archivos temporales
 - `/usr/` Unidades y aplicaciones de (Multi) usuario
 - `/var/` Variables de archivo
 - `/root/` Directorio "Home" del usuario Root
 - `/proc/` Documentación del sistema de archivos virtual del Kernel y las condiciones de los procesos en archivos de texto
-

f) ¿El Kernel de GNU/Linux es monolítico? Justifique.

El kernel Gnu/Linux si es **monolítico**.

Kernel monolítico quiere decir que toda funcionalidad que implementa el SO se ejecuta en modo kernel o supervisor, ejemplo cuando un usuario quiere ejecutar un proceso se pasa a modo kernel se resuelve todo ahí y después regresa a modo usuario cuando esa resolución termino, toda la lógica se encuentra en el modo kernel.

Ventajas el modelado, el diseño implica menos tiempo en la resolución de las cosas. (cambio de modo - resuelvo lo que tengo que resolver - vuelvo a modo usuario).

También existe el **kernel microkernel**, se trata de que el modo kernel o supervisor este el menos tiempo posible y trata de dejar en modo usuario diferentes componentes para que se ejecuten en modo usuario y que hagan de apoyo al modo kernel.

Ejemplo: el SO tiene un conjunto de procesos que van a ir cambiando cada 3 segundos, cada vez que cambia de proceso eso debe estar en modo kernel, ahora la selección de cuál de esos 3 procesos va a ser seleccionado para a entrar a ejecutarse se puede hacer en modo usuario.

Básicamente el kernel microkernel reduce al máximo el uso del modo kernel y delega cosas al modo usuario cosa que no sucede con el modo kernel.

5) Intérprete de comandos (Shell):

a) ¿Qué es?

La shell también conocida como CLI (interfaz de línea de comando), es el que gestiona la interacción entre el usuario y el sistema operativo.

(es solo un programa que hará de mediador entre nosotros y el kernel del SO. Puede ser tanto gráfico Ej. La interfase X-Window, como de texto Linux (Ej. El bash).

La Shell no forma parte básico del sistema operativo si no que la misma dialogo con el kernel.

La shell es iniciada por un proceso llamado login y dado que cada usuario tiene asignado una shell por defecto la misma se inicia cada vez que un usuario comienza a trabajar en su estación de trabajo, es decir se loguea en una terminal. Dentro del contenido del archivo /etc/passwd, se puede ver cual es la shell que cada usuario tiene asignada por defecto.

b) ¿Cuáles son sus funciones?

Es el encargado de ejecutar programas a partir del ingreso de comandos.

Ejemplo: control de procesos, redirección de entrada/salida, listado y lectura de ficheros, protección, comunicaciones y un lenguaje de órdenes para escribir programas por lotes o (scripts o guiones).

c) Mencione al menos 3 intérpretes de comandos que posee GNU/Linux y compárelos entre ellos.

- **Bourne Shell (sh)** Creado por S. Bourne, es el más utilizado en la actualidad. Su símbolo del sistema es \$. Es el shell estándar y el que se monta en casi todos los sistemas UNIX/Linux.
- **Korn Shell (ksh)** Escrito por David Korn, amplía el shell del sistema añadiendo historial de órdenes, edición en línea de órdenes y características ampliadas de programación.
- **Bourne Again Shell (bash)** Fue creado para usarlo en el proyecto GNU. BASH, por lo tanto, es un shell o intérprete de comandos GNU que incorpora la mayoría de distribuciones de Linux. Es compatible con el shell sh. Además, incorpora algunas características útiles de ksh y csh, y otras propias como la edición de línea de comandos, tamaño ilimitado del historial de comandos, control de los trabajos y procesos, funciones y alias, cálculos aritméticos con números enteros, etc. Su símbolo del sistema es nombre_usuario@nombre_equipo.

Los intérpretes se diferencian entre sí básicamente en la sintaxis de sus comandos y en la interacción con el usuario.

d) ¿Dónde se ubican (path) los comandos propios y externos al Shell?

PATH: es una variable y su contenido es una cadena que contiene rutas de directorios separadas por dos puntos.

Por ejemplo, para mostrar el valor de una variable en la shell se debe escribir:

- `echo $ nombrevariable`
- en el caso de la variable path seria
- `echo$ PATH`

la variable PATH contiene una lista de directorios separados por dos puntos. Estos son los directorios en los que el shell busca el comando que el usuario escribe desde el teclado.

La búsqueda no se realiza en el orden en el que están los directorios en la variable PATH.

- Los **comandos propios**, también conocidos como built-in commands, son aquellos que están integrados en la propia shell. Estos comandos suelen ser los más básicos y necesarios para la gestión de la sesión y del sistema, como por ejemplo: cd (para cambiar de directorio), echo (para imprimir texto en la pantalla), exit (para salir de la sesión), entre otros.
- Por otro lado, los **comandos externos** son aquellos que se encuentran en el sistema operativo y que son invocados desde la shell. Estos comandos pueden ser programas completos o scripts que realizan una tarea específica. Ejemplos de comandos externos son ls (para listar los archivos en un directorio), cat (para mostrar el contenido de un archivo), grep (para buscar patrones en un archivo), entre otros.

En resumen, los comandos propios están integrados en la propia shell y los comandos externos son programas o scripts que se ejecutan desde la shell.

e) ¿Por qué considera que el Shell no es parte del Kernel de GNU/Linux?

La principal razón es que es muy reemplazable, en caso de que falle, se puede reiniciar y todo tendría que seguir andando.

Y también

La shell no forma parte del kernel de Linux porque es una aplicación que proporciona una interfaz de línea de comandos para interactuar con el sistema operativo, mientras que el kernel es la parte central del sistema operativo que administra los recursos del hardware y proporciona servicios básicos al software. Aunque la shell y el kernel trabajan juntos, son dos componentes distintos con diferentes funciones y responsabilidades.

f) ¿Es posible definir un intérprete de comandos distinto para cada usuario?

Si, es posible.

¿Desde dónde se define?

Se definen en `etc/passwd`

¿Cualquier usuario puede realizar dicha tarea?

No, cada usuario puede definir su o sus intérpretes

6) Sistema de Archivos (File System):

a) ¿Qué es?

Es la forma en que dentro de un SO se organizan y se administran los archivos.

Los usuarios de Linux hacen una distinción del filesystem. Dicen que un filesystem es un esquema de programación que se utiliza para organizar y buscar archivos en una partición. Mientras que el sistema de archivos se refiere a todos los archivos de la computadora.

Lo que esto significa, básicamente, es que el filesystem es la estructura que se utiliza para ver, buscar y usar los archivos mediante Ubuntu. Mientras que el sistema de archivos son todos los archivos separados de esa estructura y el formato de esos archivos.

b) Mencione sistemas de archivos soportados por GNU/Linux.

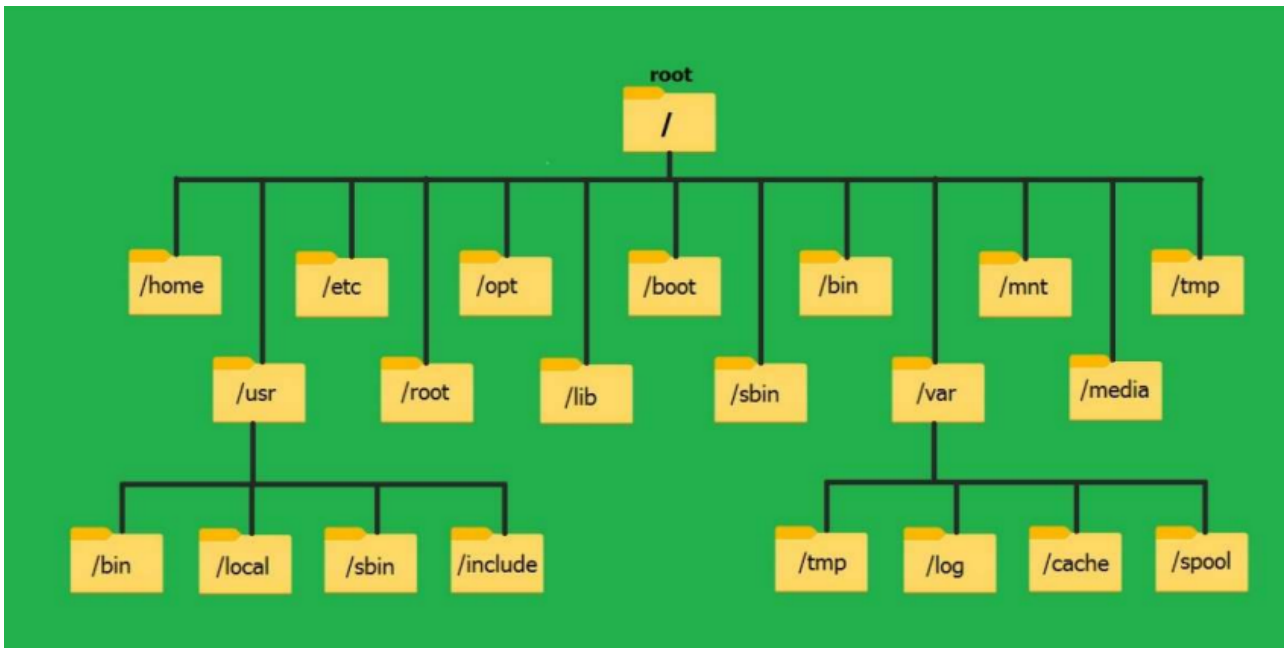
A continuación veremos una lista con algunos filesystem utilizados hoy en día:

- ext2
 - ext3
 - ReiserFS
 - XFS
-

c) ¿Es posible visualizar particiones del tipo FAT y NTFS en GNU/Linux?

Si. En estos momentos existen distribuciones de GNU-Linux que pueden realizar operaciones de lectura y escritura sobre ellas.

d) ¿Cuál es la estructura básica de los File System en GNU/Linux?



Mencione los directorios más importantes e indique qué tipo de información se encuentra en ellos.

- **/** Tope de la estructura de directorios. Es como el C:\ (raíz).
- **/home** Se almacenan archivos de usuarios (Mis documentos)
- **/var** Información que varía de tamaño (logs, BD, spools)
- **/etc** Archivos de configuración del sistema.
- **/bin** Archivos binarios y ejecutables.
- **/dev** Enlace a dispositivos.
- **/usr** Aplicaciones de usuarios

¿A qué hace referencia la sigla FHS?

(Filesystem Hierarchy Standard - Estándar de jerarquía del sistema de archivos).

FHS: es el estándar que siguen todos los sistemas operativos unix, para organizar el sistema de archivos.

7) Particiones:

a) Definición

Es el nombre que recibe cada división de una sola unidad física de almacenamiento de datos. (es un pedazo de una unidad).

Toda partición tiene su propio sistema de archivos (formato); generalmente, casi cualquier sistema operativo interpreta, utiliza y manipula cada partición como un disco físico independiente, a pesar de que dichas particiones estén en un solo disco físico.

Tipos de particiones

Existen 3 tipos diferentes de particiones:

- **Partición primaria** Son las divisiones crudas o primarias del disco, solo puede haber 4 de éstas o 3 primarias y una extendida. Depende de una tabla de particiones. Un disco físico completamente formateado consiste, en realidad, de una partición primaria que ocupa todo el espacio del disco y posee un sistema de archivos. A este tipo

de particiones, prácticamente cualquier sistema operativo puede detectarlas y asignarles una unidad, siempre y cuando el sistema operativo reconozca su formato (sistema de archivos).

- **Partición extendida** También conocida como partición secundaria es otro tipo de partición que actúa como una partición primaria; sirve para contener múltiples unidades lógicas en su interior. Fue ideada para romper la limitación de 4 particiones primarias en un solo disco físico. Solo puede existir una partición de este tipo por disco, y solo sirve para contener particiones lógicas. Por lo tanto, es el único tipo de partición que no soporta un sistema de archivos directamente.
- **Partición lógica** Ocupa una porción de la partición extendida o la totalidad de la misma, la cual se ha formateado con un tipo específico de sistema de archivos (FAT32, NTFS, ext2,...) y se le ha asignado una unidad, así el sistema operativo reconoce las particiones lógicas o su sistema de archivos. Puede haber un máximo de 23 particiones lógicas en una partición extendida.

Ventajas

- **Facilidad de reinstalación.** Si separamos los archivos Windows de nuestra información personal, será mucho más fácil reinstalar Windows. Esto es porque nos «ata» todo lo que tenemos guardado, evitando formatear o intentando hallar otras soluciones antes que esa. De hecho, podemos clonar la partición de Windows para hacer restauración.
- **Copias de seguridad.** Las copias de seguridad son las que nos salvan de los problemas catastróficos (que me lo digan a mí). Siempre hay que funcionar sobreseguro, lo que se traduce en tener backups de toda la información, o de la más importante. Las particiones nos dan la ventaja de tener un volumen para copias de seguridad.
- **Más seguridad.** Imaginad que nuestra partición de Windows queda infectada por un virus y perdemos toda la información hallada en ella. Tener nuestra información importante en otra partición nos salva de un gran apuro, especialmente si el virus es ransomware. Igualmente, instalad un antivirus y funcionad con él.
- **Mejor organización.** Esto depende de la persona, pero es innegable que podemos organizarnos mucho mejor teniendo varias particiones. Así, podemos clasificar toda nuestra información de mayor a menor importancia, o según el tipo de información, tamaño, etc.
- **Instalar dos o más sistemas operativos.** En mi opinión, es uno de los motivos principales por los que las personas hacen particiones. En el pasado, hice una partición para usar OS X y Windows, creando un arranque dual que me permitiera elegir el sistema operativo que quisiese iniciar. Al fin y al cabo, tenéis un volumen listo para usar.
- **Máquinas virtuales.** Quienes trabajéis con ellas, haréis particiones para instalar una máquina virtual en ella. Es una gran idea para separar nuestro PC de nuestra máquina virtual. Por tanto, podemos tener ambos contenidos separados; de lo contrario, si algo falla, podríamos perder toda la información de nuestro PC y máquina virtual.

Desventajas

- **Experiencia más lenta.** Si tenemos muchas particiones, puede que nos cueste encontrar un archivo en concreto. Aquí hay que ser muy metódico y ordenado para saber qué partición es vital, como cuál es prescindible. Al final, tardamos en encontrar algo por tener demasiadas particiones.
- **Posibilidad de errores.** Existe la posibilidad de que surjan más fallos cuando tenemos muchas particiones. Son meras cuestiones de probabilidad, así que no hay que volverse loco.
- **Desorden en los volúmenes.** Es fácil que esto ocurra; de hecho, me ha pasado muchas veces a mí. Cuando hacemos una partición, formateamos, volvemos a crear un volumen, etc., podemos crear particiones extendidas que no van a ningún sitio. En mi caso, se trataba de una partición de 700 MB en un HDD de 2 TB: el impacto es mínimo. Sin embargo, no tiene gracia que nos pase en un SSD de poca capacidad.
- **Innecesario para el usuario medio.** Al final, las personas que utilizan el PC para «3 cosas», no requieren particiones, ni se quieren liar a aprender cómo funciona esto. Por tanto, es una opción limitada a aquellos que buscan cierta utilidad.

b) ¿Cómo se identifican las particiones en GNU/Linux? (Considere discos IDE, SCSI y SATA).

El primer disco duro **IDE** se conoce como `/dev/hda`. Si tenemos un segundo disco duro IDE se llamará `/dev/hdb`, etc.

Los discos duros **SCSI** y **SATA** se denominan /dev/sda, /dev/sdb, etc. En cada disco son representadas **añadiendo un número decimal al nombre del disco**: sda1 y sda2 representan a la primera y segunda partición en la primera unidad de disco SCSI en el sistema.

c) ¿Cuántas particiones son necesarias como mínimo para instalar GNU/Linux? Nómbrelas indicando tipo de partición, identificación, tipo de File System y punto de montaje.

Si bien el número **mínimo absoluto de particiones sería uno (1)**, una **instalación típica de GNU / Linux tendrá al menos dos (2)** particiones: la partición raíz (indicada como /) y la partición de **intercambio**. Puede optar por tener todos los archivos en una sola partición dentro de la mayoría de las distribuciones durante la instalación.

El esquema de particiones estándar para la mayoría de las instalaciones domésticas de Linux es el siguiente: Una partición de 12-20 GB para el sistema operativo, que se monta como / (llamada “**raíz**” o **root**) Una partición más pequeña que se utiliza para aumentar su RAM, montada y denominada **/swap**. Una mampara más grande para uso personal, montada como **/home**

Para una instalación de **GNU/Linux saludable se recomiendan 3 particiones swap, root y home.**

Hay 2 tipos de particiones principales en un sistema Linux:

- **Partición de datos** datos normales del sistema Linux, incluida la partición raíz que contiene todos los datos para iniciar y ejecutar el sistema; y.
- **Partición de intercambio** expansión de la memoria física de la computadora, memoria extra en el disco duro.
- **Tipo de filesystem** Sistema de archivos Ext2, ext3 y ext4: Así como Apple y Microsoft tienen sus propios sistemas, estos tres (cada uno evolución del anterior) son los utilizados por las distribuciones GNU/Linux. El principal inconveniente es que sólo puede ser utilizado en esta familia de sistemas operativos.

Tipo de filesystem Sistema de archivos **Ext2, ext3 y ext4**: Así como Apple y Microsoft tienen sus propios sistemas, estos tres (cada uno evolución del anterior) son los utilizados por las distribuciones GNU/Linux. El principal inconveniente es que sólo puede ser utilizado en esta familia de sistemas operativos.

d) Ejemplifique diversos casos de particionamiento dependiendo del tipo de tarea que se deba realizar en su sistema operativo.

Integración de nuevas adquisiciones Si forma una empresa nueva. La empresa recién formada no utiliza las mismas aplicaciones para las nóminas, el inventario y la facturación. Tiene previsto consolidar las dos empresas en un solo conjunto de aplicaciones, pero esta consolidación tardará un tiempo. Mientras tanto, debe reducir el coste de los centros de datos con rapidez.

Por tanto, decide crear particiones lógicas para las aplicaciones utilizadas por la empresa recién adquirida. Instala un sistema operativo y las aplicaciones utilizadas por la empresa nueva en la partición lógica. Si las cargas de trabajo combinadas necesitan más recursos.

Creación de varios entornos de cliente El usuario suministra servicios e-commerce de alta disponibilidad a diversos clientes. Proporciona recursos de sistema, aplicaciones y soporte técnico a cada cliente, y cada cliente puede configurar y utilizar independientemente las aplicaciones ejecutadas en los recursos de sistema suministrados. En un entorno de este tipo, es esencial aislar a los clientes para que sólo tengan acceso a sus recursos. Sin embargo, dedicar un servidor físico a cada cliente tiene un coste prohibitivo, y no permite aumentar ni disminuir fácilmente la cantidad de recursos de sistema utilizados por cada cliente.

Por tanto, decide crear una partición lógica para cada cliente. Instala un sistema operativo y aplicaciones en cada partición lógica. A continuación, puede utilizar el particionamiento dinámico para añadir recursos a particiones lógicas o eliminar recursos de ellas según sea necesario. Si un cliente deja de utilizar el servicio, puede suprimir la partición lógica de dicho cliente y reasignar los recursos a otras particiones lógicas

e) ¿Qué tipo de software para particionar existe? Mencínelos y compare

Existen 2 tipos:

- **Destructivos:** permiten crear y eliminar particiones (fdisk)

- **No destructivo:** permiten crear, eliminar y modificar particiones

8) Arranque (bootstrap) de un Sistema Operativo:

a) ¿Qué es el BIOS?

La BIOS (Sistema Básico de Entrada/Salida) es un software que inicializa y comprueba todos los componentes de hardware durante el arranque de la máquina. También prepara el equipo para que el Sistema Operativo se cargue y se ejecute. Su nombre viene de las siglas en inglés Basic Input Output System.

La Bios es un software de bajo nivel que se encuentra en el Motherboard. Cuando se arranca la computadora el bios se ejecuta, realizando el POST (power – on Self -test) que incluye rutinas que, entre otras actividades, fijan valores de señales internas y ejecutan test internos (ram , teclado etc).

¿Qué tarea realiza?

- **POST** POST es un acrónimo de Power-On Self Test (Autocomprobación de encendido) que se ejecuta en tu PC en el momento en que la enciendes. El POST prueba el hardware de tu computadora y se asegura de que no exista ninguna avería ni haya errores presentes en su sistema operativo. El POST comprueba todo, desde el teclado y la unidad de disco, hasta la velocidad de la RAM en una computadora y los puertos integrados. Si todo está en orden, POST continuará como de costumbre y permitirá que tu PC se inicie normalmente. Si se detecta un error, el BIOS emitirá un mensaje de error que puede aparecer en forma de texto en la pantalla o una serie de pitidos indicativos de dicho error. Estos sonidos siempre son señales para mensajes determinados, por lo que, si te sucede esto, deberás verificar lo que ello significa para el hardware de tu computadora.
- **Configuración del CMOS** Tu PC almacena todas las configuraciones de bajo nivel como la hora del sistema y la configuración de hardware dentro del CMOS. Esto significa que cada cambio que realices en la estructura del BIOS se guardará en este chip de memoria especial llamado Semiconductor Complementario de Óxido Metálico, o CMOS (por sus siglas en inglés). La configuración del CMOS es responsable, a su vez, de configurar tu contraseña, hora y fecha.
- **Cargador de arranque** El cargador de arranque, programa que vive dentro de la EPROM o ROM de tu computadora, tiene la tarea de leer el sector de arranque del disco duro de tu PC para moverse a lo largo de todo el proceso de carga del sistema operativo.

b) ¿Qué es UEFI?

UEFI (Unified Extensible Firmware Interface o Interfaz de Firmware Extensible Unificada). Es la Alianza entre varias compañías con el objetivo de modernizar el proceso de arranque, querían reemplazar a la BIOS. Sus funciones son parecidas a los de la BIOS, pero mejoradas. UEFI aporta criptografía, autenticación por red, y una interfaz gráfica.

¿Cuál es su función?

Usa el sistema GPT (GUID Partition table) para solucionar algunas limitaciones del MBR, tales como la cantidad de particiones y capacidad máxima del dispositivo particionado.

c) ¿Qué es el MBR?

Sector de arranque – MBR

El Master Boot Record es el primer sector del disco físico.

Se ubica en el cilindro 0, cabeza 0, sector 1. En todos los discos existe un MBR, pero el mismo es tenido en cuenta si el disco contiene las particiones del sistema.

El tamaño de MBR coincide con el tamaño estandar de sector del disco, generalmente 512 bytes. Estos bytes contienen lo siguiente:

¿Que es el MBC?

El MBC es un pequeño código que permite arrancar el sistema operativo. La tarea del MBC consiste en buscar una entrada de la tabla de particiones marcada con el valor 0x80 indicando que esta entrada es booteable lo cual indica que la partición correspondiente a esta entrada es booteable. La última acción del BIOS es leer el MBC. Lo lleva a memoria y lo ejecuta.

d) ¿A qué hacen referencia las siglas GPT?

Una partición GPT GUID Partition Table es un estándar diseñado para realizar la configuración de las tablas de particiones y su ubicación en medios de almacenamiento como los discos duros físicos.

Puede ser utilizado independientemente de la UEFI.

¿Qué sustituye? Indique cuál es su formato.

MBR es la tabla de particiones tradicional que soporta los sistemas operativos más antiguos, mientras que **GPT** es un nuevo sustituto que no tiene límites en cuanto al tamaño del disco y el número de particiones que se pueden crear

Las ventajas de GPT sobre MBR

- Soporta discos duros más grandes que 2TiB.
- Permite crear particiones teóricamente ilimitadas.
- Contiene una verificación de redundancia cíclica para comprobar la integridad de sus datos.
- Contiene la copia de seguridad del encabezado GPT primario y las entradas de la partición que protege mejor los datos del disco.

e) ¿Cuál es la funcionalidad de un “Gestor de Arranque”? ¿Qué tipos existen? ¿Dónde se instalan? Cite gestores de arranque conocidos.

Un gestor de arranque, es un programa que se encarga del inicio del ordenador, antes del sistema operativo, y que nos permite elegir el sistema operativo que queremos cargar, es decir se encarga del bootstrap (carga) o booteo del sistema.

Es el encargado de cargar el sistema operativo (kernel) de un ordenador en la memoria RAM.

Se ejecuta luego del código de la BIOS.

Los datos de un sistema operativo deben cargarse directamente en la memoria RAM al iniciarse el dispositivo. Esto es posible con el bootloader, también conocido como gestor de arranque. El bootloader suele ejecutarse directamente al arrancar un dispositivo usando algún medio que sea booteable, es decir, que sirva como unidad de arranque, como puede ser un disco duro, un CD o DVD, o un stick booteable. El medio de arranque recibe la información acerca de dónde se encuentra el bootloader por parte del firmware del ordenador (BIOS, por ejemplo). Al proceso completo se le denomina inicio, arranque o, en inglés to boot.

En lo que a la ubicación de los bootloaders se refiere, se han consolidado dos opciones:

- **El bootloader se guarda en el primer bloque del medio booteable o de arranque (MBR):** Está estrechamente relacionada con el principio de los Master Boot Records, que no solo contienen la referencia necesaria por parte del firmware hacia el bootloader, sino también el software de inicio en sí. Para el record se reserva siempre el primer bloque o sector de memoria disponible en el medio. Este bloque o sector también es llamado, precisamente por esta función tan importante, boot block o boot sector.
- **El bootloader se guarda en una partición específica del medio de arranque:** El sistema operativo usa como lugar de almacenamiento del bootloader una partición seleccionada, si bien el sistema de archivos de base y la tabla de particiones utilizada no tienen por qué coincidir en absoluto. El firmware siempre es un factor decisivo y, este tipo de ubicación del bootloader, requiere además un formato de archivo específico del gestor de arranque. En dispositivos con UEFI se trata, por ejemplo, del formato PE/COFF (Portable Executable / Common Object File Format).

Algunos gestores de arranque: GRUB

- GRUB 2
- LILO o SYSLINUX.
- NTLDR
- GAG
- YaST

GRand Unified Bootloader(grub): gestor de arranque múltiple que se usa para iniciar dos o más sistemas operativos instalados en un mismo ordenador.

f) ¿Cuáles son los pasos que se suceden desde que se prende una computadora hasta que el Sistema Operativo es cargado (proceso de bootstrap)?

Cuando aprietas el botón de arranque de un ordenador, lo primero que aparece en la pantalla son los datos acerca del hardware del equipo. El software responsable de esta información es, en primer lugar, el ya mencionado firmware, que los productores de los dispositivos por lo general implementan en una memoria flash en la placa base o mainboard. En la mayoría de los ordenadores de mesa y portátiles se apuesta por el ya veterano BIOS (Basic Input/Output System) o por el más reciente UEFI (Unified Extensible Firmware Interface). Ambos programas reúnen diferentes datos del hardware y para generar, entre otras cosas, una lista completa de todas las unidades de disco disponibles en el dispositivo.

Una vez se ha completado este proceso, el firmware comprueba uno a uno los soportes de datos encontrados en busca del bootloader o, más concretamente, la llamada boot signature o boot record. La búsqueda se inicia siempre en los medios extraíbles conectados (CD/DVD, stick USB, discos duros externos, etc.) y luego pasa a los discos duros internos. En estos últimos, el bootloader (o la boot signature) suele estar en el Master Boot Record (MBR), que también aloja la tabla de particiones del soporte de datos. Cuando se encuentre un bootloader, este se cargará y se iniciará así el sistema. Si la búsqueda, en cambio, no tiene éxito, el firmware mostrará un mensaje de error.

g) Analice el proceso de arranque en GNU/Linux y describa sus principales pasos.

El flujo de control durante el arranque es desde el BIOS, al gestor de arranque y al núcleo (kernel)).

- **Kernel:** Este inicia el planificador (para permitir la multitarea) y ejecuta el primer espacio de usuario (fuera del espacio del núcleo) y el programa de inicialización (que establece el entorno de usuario y permite la interacción del usuario y el inicio de sesión), momento en el que el núcleo se inactiva hasta que sea llamado externamente.
- La etapa del cargador de arranque no es totalmente necesaria. Determinadas BIOS pueden cargar y pasar el control a Linux sin hacer uso del cargador. Cada proceso de arranque será diferente dependiendo de la arquitectura del procesador y el BIOS.
- En el apagado, Init es llamado a cerrar toda las funcionalidades del espacio de usuario de una manera controlada, de nuevo a través de secuencias de comandos, tras lo cual el Init termina y el núcleo ejecuta el apagado.

h) ¿Cuáles son los pasos que se suceden en el proceso de parada (shutdown) de GNU/Linux?

- Se notifica a los usuarios este hecho.
- Se bloquea el sistema para que nadie más pueda acceder exceptuando el root.
- Se envía la señal SIGTERM (señal de terminación) a todos los procesos no definidos en inittab(contiene un registro para cada proceso que define los niveles de ejecución para ese proceso) para el siguiente run level, provocando que terminen su ejecución de modo ordenado.

i) ¿Es posible tener en una PC GNU/Linux y otro Sistema Operativo instalado? Justifique

Si es posible ya lo vimos anteriormente gracias a las particiones de disco instalar múltiples sistema operativos o a través de maquinas virtuales.

9) Archivos

a) ¿Cómo se identifican los archivos en GNU/Linux?

Un nombre de archivo puede tener entre 1 y 255 caracteres. recomendable emplear los caracteres con significado especial en Linux, que son los siguientes: = \ ^ ~ ' " ` * ; - ? () ! & ~ < >

b) Investigue el funcionamiento de los editores vi y mcedit, y los comandos cat y more.

- **VI** Es el editor de texto clásico en UNIX. Puede usarse en cualquier tipo de terminal con un mínimo de teclas.

MODOS DE VI: Existen tres modos o estados de vi:

- **Modo comando:** Este es el modo en el que se encuentra el editor cada vez que se inicia. Las teclas ejecutan acciones (comandos) que permiten mover el cursor, ejecutar comandos de edición de texto, salir de vi, guardar cambios, etc.
- **Modo inserción o texto:** Este es el modo que se usa para insertar el texto. Existen varios comandos que se pueden utilizar para ingresar a este modo.
- **Modo línea o ex:** Se escriben comandos en la última línea al final de la pantalla.
- **MCEDIT**

Al igual que Vi funciona como gestor de archivos

- **cat** Es la abreviatura de concatenar. Esto se refiere al hecho de que cat puede ser utilizado para combinar múltiples archivos en un archivo, también se puede utilizar para crear nuevos archivos y para mostrar el contenido de los archivos existentes.
- **more** Es un comando para ver (pero no modificar) el contenido de un archivo o comando y visualizarlo por páginas.

c) Cree un archivo llamado "prueba.exe" en su directorio personal usando el vi. El mismo debe contener su número de alumno y su nombre.

- Utilice el siguiente comando para crear un archivo (en este ejemplo, .htaccess). También puede editar un archivo existente con el mismo comando.

```
nvim fabianmartinezzrincon.txt
```

- Pulse la tecla de la letra i para cambiar al modo de entrada.
- Inserte el contenido deseado o realice las modificaciones deseadas.
- Pulse la tecla ESC para salir del modo de entrada.
- Guarde el nuevo archivo o los cambios realizados con el siguiente comando:

```
:wq
```

d) Investigue el funcionamiento del comando file. Pruébalo con diferentes archivos. ¿Qué diferencia nota?

Permite detectar el tipo y formato de un archivo

file [opciones] archivo

- b) muestra solo el tipo de archivo sin ninguna información adicional.
- i) fuerza el uso de la sintaxis MIME para mostrar el tipo de archivo.
- z) trata los archivos como si estuvieran comprimidos.

10) Indique qué comando es necesario utilizar para realizar cada una de las siguientes acciones. Investigue su funcionamiento y parámetros más importantes:

- mkdir "ISO 2022" Cree la carpeta ISO2017
- cd 'ISO 2022' Acceda a la carpeta (cd)
- touch ISO2022-1 ISO2022-2 Cree dos archivos con los nombres iso2017-1 e iso2017-2 (touch)
- ls Liste el contenido del directorio actual (ls)
- pwd Visualizar la ruta donde estoy situado (pwd)
- find ./'ISO 2022' -name "ISO*" Busque todos los archivos en los que su nombre contiene la cadena "iso*" (find)
- df Informar la cantidad de espacio libre en disco (df)
- who Verifique los usuarios conectado al sistema (who)
- vi ISO2022-1 Acceder a el archivo iso2017-1 e ingresar Nombre y Apellido
- tail ISO2022-1 Mostrar en pantalla las últimas líneas de un archivo (tail).

11) Investigue su funcionamiento y parámetros más importantes:

- shutdown El comando de apagado (Shutdown) te permite apagar, reiniciar y detener tu sistema

```
sudo shutdown
sudo shutdown now
sudo shutdown [time]
sudo shutdown 11:50
sudo shutdown +5
sudo shutdown +2 "System update"
sudo shutdown -r
sudo shutdown +3 -r "Update System"
sudo shutdown -c
```

Options	Description
-a	To control access to the "shutdown" command, it employs the control access file "/etc/shutdown.allow."
-k	Instead of shutting down, deliver warning messages as though the shutdown is actual.
-P	Tells the system to power down before shutting down.
-f	It skips fsck after reboot.
-F	After reboot it forces fsck.
-H	This option orders the system to descend into the boot the monitor on computers which provide support to it if -h is also supplied.

- sudo reboot Sirve para reiniciar el equipo
- sudo halt El comando halt detiene la CPU del ordenador
- locate El comando locate es una alternativa útil, ya que es más rápido que find para realizar búsquedas. Eso se debe a que sólo escanea tu base de datos de Linux en lugar de todo el sistema. Además, la sintaxis es más fácil de

escribir `sudo apt install locate`

- `uname` Se usa para verificar la información del sistema

```
uname -s
uname -r
uname -v
uname -n
uname -m
uname -p
uname -i
uname -o
uname -a
```

- `gmesg` (No entiendo porque esta diferente) El comando `dmesg` es una utilidad de Linux que muestra mensajes relacionados con el kernel recuperados del búfer de anillo del kernel. `dmesg`
- `lspci` El comando `lspci` lista todos los componentes tipo pci como son las tarjetas de red, tarjetas de sonido o tarjetas de televisión.
- `at` Ejecuta comandos a la hora especificada.
- `netstat` Los administradores de sistemas utilizan `netstat` el comando de Linux para ver información sobre las conexiones de red `sudo apt install net-tools`
- `mount` Se utiliza para montar dispositivos y particiones para su uso por el sistema operativo

```
sudo apt install nfs-common
sudo mkdir -p /mnt/client_shareddirectory
sudo mount [nfs_server]:/[nfs_shareddirectory] [client_mountpoint]
```

- `umount` El comando `umount` le permite eliminar un sistema de archivos remoto que esté montando en la actualidad `umount --all`
- `head` Este comando sirve principalmente para mostrar al principio de un archivo (de texto) o para reducir a lo especificado los datos mostrados por un comando de Linux
- `losetup` `losetup` de comandos de Linux se utiliza para fijar el dispositivo de bucle. **parámetros:**
 - `d` dispositivo extraíble.
 - `e` Iniciar cifrado codificación.
 - `o` <número de traducción> Establecer el número de conversión de datos.
- `write` sirve para enviar un mensaje a otro usuario del sistema.
 - `write` usuario
 - Escribo aquí lo que
 - quiera que le llegue y luego cierro.
 - Control-D
- `mkfs` Se utiliza para dar formato a un dispositivo de almacenamiento de bloque con un determinado sistema de archivos
- `fdisk` (con cuidado) Permite al usuario crear particiones en el disco duro de la misma manera que su contraparte de MS-DOS

12) Investigue su funcionamiento y parámetros más importantes:

- a) Indique en qué directorios se almacenan los comandos mencionados en el ejercicio anterior.

El directorio `/bin` es un directorio estático y es donde se almacenan todos los binarios necesarios para garantizar las funciones básicas a nivel de usuario. Solo almacena los ejecutables de usuario, ya que los binarios necesarios para tareas administra/vas gestionadas por el usuario `root` o súper-usuario del sistema se encuentran en el directorio `/sbin`.

