Lo anterior está en el cuaderno. 5)

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

**. Bash (Bourne Again Shell)**

* **Descripción**: Bash es uno de los intérpretes de comandos más utilizados en sistemas GNU/Linux y es el shell predeterminado en muchas distribuciones de Linux, como Ubuntu y Fedora. Es una mejora del shell original de Unix, llamado Bourne Shell (sh), y ofrece muchas características adicionales, como historial de comandos, autocompletado, scripting avanzado, y funciones de control de trabajos.
* **Características Clave**:
  + Historial de comandos accesible con las teclas de flecha.
  + Redirección de entrada y salida.
  + Soporte para scripting con estructuras de control, funciones y variables.
  + Completado automático de comandos y nombres de archivos.

**2. Zsh (Z Shell)**

* **Descripción**: Zsh es un shell poderoso que incorpora características de Bash, csh (C Shell) y ksh (Korn Shell), además de muchas funcionalidades propias. Es conocido por su autocompletado avanzado, corrección de errores de tipeo, y su facilidad de personalización. Zsh es muy popular entre los desarrolladores y usuarios avanzados por su flexibilidad y capacidad de extensión.
* **Características Clave**:
  + Autocompletado avanzado y sensible al contexto.
  + Corrección automática de errores de tipeo en los comandos.
  + Integración de temas y plugins (como con el marco Oh My Zsh).
  + Globbing avanzado para selección de archivos y directorios.

**3. Fish (Friendly Interactive Shell)**

* **Descripción**: Fish es un shell moderno y amigable con el usuario, diseñado para ser fácil de usar y más intuitivo que los shells tradicionales como Bash y Zsh. Fish proporciona autocompletado sofisticado, resaltado de sintaxis en la línea de comandos y una configuración sencilla. No sigue estrictamente la sintaxis POSIX de otros shells, lo que lo hace menos compatible con scripts de shell tradicionales, pero es apreciado por su facilidad de uso interactiva.
* **Características Clave**:
  + Autocompletado con sugerencias y documentación en línea.
  + Resaltado de sintaxis en la línea de comandos.
  + Sin necesidad de configuración inicial compleja.
  + Configuración basada en web para ajustes del shell.

Bash es ideal para scripting tradicional y compatibilidad, Zsh ofrece personalización y funcionalidades avanzadas para usuarios más experimentados, y Fish proporciona una experiencia de usuario amigable y moderna.

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

En sistemas GNU/Linux, los comandos se pueden clasificar en comandos propios del shell (comandos internos o incorporados) y comandos externos (programas y scripts independientes). La ubicación de estos comandos varía dependiendo de si son internos o externos al shell. En el caso de los comandos propios, son parte del propio shell y se implementan directamente en el código del intérprete de comandos. No existen como archivos separados en el sistema de archivos, sino que están integrados en el ejecutable del shell. Para los comandos externos, son programas independientes que existen como archivos ejecutables en el sistema de archivos. Estos comandos se ejecutan iniciando un proceso separado y suelen estar ubicados en directorios estándar que están incluidos en la variable de entorno PATH.

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

El **shell** no es parte del **kernel** de GNU/Linux debido a las diferencias en sus roles y responsabilidades dentro del sistema operativo. El kernel maneja las tareas fundamentales del sistema, mientras que el shell actúa como un intermediario, proporcionando una interfaz de usuario para interactuar con esas funciones del sistema. Ademas debemos tener en cuenta que, debido a que el shell es independiente del kernel, existen múltiples opciones de shells, cada uno con sus propias características y funcionalidades, lo que permite a los usuarios y administradores de sistemas elegir el shell que mejor se adapte a sus necesidades.

(f) ¿Es posible definir un intérprete de comandos distinto para cada usuario? ¿Desde dónde se define? ¿Cualquier usuario puede realizar dicha tarea?

Es posible definir un intérprete de comandos distinto para cada usuario en un sistema GNU/Linux. El intérprete de comandos (o shell) predeterminado de un usuario se especifica en el archivo de configuración de usuarios y puede ser diferente para cada usuario del sistema.

Cada usuario en un sistema Linux tiene una cuenta de usuario asociada con un shell específico. Esta información se almacena en el archivo /etc/passwd, que contiene datos sobre las cuentas de usuario. Puedes cambiar el shell predeterminado de un usuario utilizando el comando ‘chsh’, mientras que para especificar un shell diferente para un usuario:

chsh -s /ruta/al/nuevo/shell usuario

Este comando solicitará la contraseña del usuario y, luego de ingresarla correctamente, cambiará el shell predeterminado.

No cualquier usuario puede definir un intérprete de comandos distinto para otros usuarios. La capacidad de cambiar el shell predeterminado de un usuario está generalmente restringida por permisos y roles dentro del sistema.

 **Cambio de Shell para Propio Usuario**: Los usuarios pueden cambiar su propio shell utilizando chsh sin necesidad de permisos adicionales.

 **Cambio de Shell para Otros Usuarios**: Solo los administradores del sistema (root) tienen la capacidad de cambiar el shell predeterminado para otros usuarios. Esto se hace para mantener la seguridad y la integridad del sistema.

6). Sistema de Archivos (File System):

(a) ¿Qué es?

El **sistema de archivos** (file system) es una estructura y un conjunto de reglas que un sistema operativo utiliza para gestionar y organizar archivos en un dispositivo de almacenamiento. Proporciona una forma de almacenar, recuperar, organizar y manipular archivos y directorios en el sistema.

Funciones:

 **Organización de Datos**: El sistema de archivos organiza los datos en bloques o sectores en el dispositivo de almacenamiento, permitiendo al sistema operativo y a las aplicaciones gestionar y acceder a estos datos de manera eficiente.

 **Gestión de Archivos y Directorios**: Permite crear, leer, escribir, modificar y eliminar archivos y directorios. También proporciona estructuras para organizar los archivos en jerarquías y sistemas de directorios.

 **Control de Acceso**: Gestiona los permisos de acceso a los archivos y directorios, determinando quién puede leer, escribir o ejecutar un archivo.

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

Algunos: ext4 (Fourth Extended Filesystem) – XFS - Btrfs (B-tree File System) - FAT32 (File Allocation Table 32)

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

Para visualizar

**lsblk**: Muestra información sobre los dispositivos de bloques y sus particiones

fdisk para listar las particiones en un dispositivo,

En GNU/Linux, es posible visualizar y montar particiones FAT y NTFS utilizando herramientas y comandos específicos. FAT es compatible de forma nativa y se puede montar directamente, mientras que NTFS requiere la instalación del paquete ntfs-3g para el soporte completo.

(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. ¿A qué hace referencia la sigla FHS?

En GNU/Linux, la estructura básica del sistema de archivos sigue un esquema jerárquico y modular

**1. Raíz del Sistema de Archivos**

* **Directorio Raíz (/)**: Todo el sistema de archivos en GNU/Linux está basado en un único punto de raíz (/). Todos los archivos y directorios están organizados bajo este directorio raíz, formando una estructura de árbol invertido.

**\*\*2. Directorios Principales**

•/ Tope de la estructura de directorios. Es como el C:\

• /home Se almacenan archivos de usuarios (Mis documentos)

• /var Informacion que varıa de tamaño (logs, BD, spools)

• /etc Archivos de configuracion

• /bin Archivos binarios y ejecutables

• /dev Enlace a dispositivos

• /usr Aplicaciones de usuarios

**\*\*3. Inodos**

* **Inodos**: Cada archivo y directorio en un sistema de archivos GNU/Linux tiene un inodo asociado. Un inodo es una estructura de datos que almacena metadatos sobre el archivo o directorio, como permisos, propietario, tamaño, y la ubicación de los datos en el disco.

**\*\*4. Puntos de Montaje**

* **Puntos de Montaje**: Directorios en la jerarquía del sistema de archivos donde se montan otros sistemas de archivos. Por ejemplo, una partición adicional puede ser montada en /mnt o /media.

**\*\*5. Sistema de Archivos Virtual**

* **/proc y /sys**: Son sistemas de archivos virtuales que proporcionan una interfaz para interactuar con el kernel y el hardware del sistema. No corresponden a archivos en el disco físico, sino a datos generados por el sistema.

**\*\*6. Journaling**

* **Journaling**: Algunos sistemas de archivos (como ext4) utilizan journaling para mantener un registro de las transacciones. Esto ayuda a proteger la integridad de los datos en caso de fallos del sistema o apagones inesperados.

FHS (**Estándar de Jerarquía de Sistema de Ficheros)** es un estándar definido para organizar y estructurar los directorios y archivos en sistemas operativos tipo UNIX, incluyendo GNU/Linux. Este estándar proporciona una guía que especifica la ubicación de archivos y directorios en el sistema de archivos, lo cual facilita la consistencia y la interoperabilidad entre diferentes distribuciones de Linux y otros sistemas UNIX.

7. Particiones:

(a) Definición. Tipos de particiones. Ventajas y Desventajas.

Una **partición** es una división lógica de un disco duro o cualquier otro dispositivo de almacenamiento, que permite al sistema operativo gestionar y utilizar el espacio de manera organizada. En términos simples, una partición es como un segmento del disco que actúa como un "compartimento" independiente, separando datos para diferentes propósitos o sistemas operativos. Cada partición se comporta como un disco independiente, aunque físicamente formen parte del mismo dispositivo de almacenamiento.

### ****Tipos Comunes de Particiones****

1. **Partición Primaria:**
   * Es una partición principal que puede contener un sistema de archivos. Tradicionalmente, un disco duro puede tener hasta cuatro particiones primarias.
   * Una de estas particiones puede ser marcada como activa para arrancar un sistema operativo.
2. **Partición Extendida:**
   * Actúa como un contenedor para particiones lógicas, permitiendo superar la limitación de cuatro particiones primarias. Solo puede haber una partición extendida en un disco, pero puede contener múltiples particiones lógicas.
3. **Partición Lógica:**
   * Particiones dentro de la partición extendida. No hay un límite práctico en la cantidad de particiones lógicas que puedes crear, lo cual permite una gran flexibilidad en la organización del espacio en disco.
4. **Partición de Arranque (Boot Partition):**
   * Una partición específica que contiene los archivos necesarios para arrancar el sistema operativo. Puede ser la misma que la partición del sistema o una separada.
5. **Partición de Intercambio (Swap Partition):**
   * En sistemas Linux, es una partición utilizada para gestionar la memoria virtual. Actúa como una extensión de la RAM física y ayuda a mejorar el rendimiento del sistema cuando la memoria RAM está llena.

### ****Ventajas de las Particiones****

1. **Organización y Estructuración de Datos:**
   * Las particiones permiten organizar diferentes tipos de datos en secciones separadas del disco. Por ejemplo, puedes tener una partición para el sistema operativo, otra para datos personales, y otra para copias de seguridad. Esto facilita la gestión y el acceso a los datos.
2. **Seguridad y Aislamiento de Datos:**
   * Al separar el sistema operativo y los datos del usuario en particiones diferentes, se reduce el riesgo de pérdida de datos en caso de que una partición se corrompa o se vea comprometida. Esto también protege el sistema operativo de errores en los archivos de usuario.
3. **Facilita la Instalación de Múltiples Sistemas Operativos:**
   * Las particiones permiten la coexistencia de múltiples sistemas operativos en el mismo disco duro. Cada sistema operativo puede estar en su propia partición, lo que evita conflictos y facilita la gestión de diferentes entornos de trabajo.
4. **Mejora el Rendimiento:**
   * Al separar los archivos del sistema y los archivos de datos en particiones diferentes, se pueden reducir los tiempos de búsqueda y mejorar el rendimiento general del sistema. Esto es especialmente útil para servidores y entornos de alto rendimiento.
5. **Facilita el Mantenimiento y la Actualización del Sistema:**
   * En caso de actualización o reinstalación del sistema operativo, tener datos y aplicaciones en particiones separadas evita la pérdida de información. Solo la partición del sistema operativo necesita ser formateada, mientras que los datos en otras particiones permanecen intactos.
6. **Optimización del Uso del Espacio en Disco:**
   * Permite asignar el espacio de almacenamiento de manera más eficiente, adaptándose a las necesidades específicas de cada tipo de datos o aplicación. Por ejemplo, puedes ajustar el tamaño de la partición de intercambio (swap) para optimizar el rendimiento.
7. **Mejor Copia de Seguridad y Recuperación:**
   * Las particiones facilitan la realización de copias de seguridad y la restauración de datos, ya que es posible clonar o restaurar particiones individuales sin afectar a todo el disco.

### ****Desventajas de las Particiones****

1. **Espacio Desperdiciado:**
   * Al asignar un tamaño fijo a cada partición, puede haber espacio no utilizado en una partición mientras que otra partición puede quedarse sin espacio. La gestión incorrecta del espacio puede llevar a una subutilización del disco.
2. **Complejidad en la Gestión:**
   * El uso de múltiples particiones puede complicar la gestión del sistema. Requiere una planificación cuidadosa para decidir cómo particionar el disco, cuánto espacio asignar a cada partición, y cómo ajustarlas en caso de cambios en las necesidades.
3. **Limitaciones en Redimensionamiento:**
   * Aunque es posible redimensionar particiones, hacerlo puede ser arriesgado y llevar a la pérdida de datos si no se realiza correctamente. También puede requerir herramientas adicionales y tiempo significativo.
4. **Impacto en el Rendimiento si No Se Gestiona Correctamente:**
   * Si no se planifica adecuadamente, las particiones pueden afectar negativamente el rendimiento. Por ejemplo, si una partición se llena mientras otras tienen espacio libre, el sistema puede ralentizarse o incluso fallar.
5. **Problemas de Arranque con Múltiples Sistemas Operativos:**
   * La configuración de múltiples sistemas operativos en particiones diferentes puede complicar el proceso de arranque. Puede requerir la instalación y configuración de un gestor de arranque (bootloader) adecuado para gestionar la selección del sistema operativo.
6. **Riesgo de Pérdida de Datos:**
   * Aunque las particiones pueden mejorar la seguridad, también existe el riesgo de pérdida de datos si una partición se corrompe o si se realiza un mal manejo de las particiones durante la instalación o el mantenimiento.
7. **No Todas las Herramientas Soportan Redimensionamiento Dinámico:**
   * Algunas herramientas y sistemas de archivos no permiten redimensionar particiones en vivo sin un proceso complicado y riesgo de pérdida de datos, lo que limita la flexibilidad en comparación con otras soluciones de almacenamiento.

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

En GNU/Linux, las particiones se identifican de manera lógica utilizando nombres específicos asignados por el sistema operativo. Esta identificación puede variar dependiendo del tipo de disco (SATA, SCSI, NVMe, etc.) y del número de discos instalados en el sistema.

Los discos y sus particiones se identifican en GNU/Linux a través de nombres de dispositivos que se encuentran en el directorio /dev. Estos nombres siguen un esquema estándar:

* **Discos SATA, SCSI y USB (usando sd):**
  + Un disco se identifica como /dev/sdX, donde X es una letra que representa el disco (por ejemplo, a, b, c).
    - Por ejemplo, el primer disco se denominará /dev/sda, el segundo /dev/sdb, y así sucesivamente.
  + Las particiones dentro de un disco se identifican con un número adicional:
    - Por ejemplo, la primera partición del primer disco será /dev/sda1, la segunda partición del mismo disco será /dev/sda2, etc.
* **Discos IDE (antiguos, usando hd):**
  + Los discos IDE se identificaban como /dev/hdX.
    - Por ejemplo, /dev/hda para el primer disco IDE.
  + Las particiones se identifican de manera similar, con números:
    - Por ejemplo, /dev/hda1 para la primera partición.
* **Dispositivos NVMe (usando nvme):**
  + Los discos NVMe tienen un esquema diferente debido a su alto rendimiento y capacidades avanzadas.
    - Se identifican como /dev/nvmeXnY, donde X es el número de controlador y Y el número del dispositivo.
    - Las particiones se identifican como /dev/nvmeXnYpZ, donde Z es el número de partición.
    - Por ejemplo, /dev/nvme0n1p1 representa la primera partición en el primer dispositivo NVMe.

### ****2. Herramientas de Línea de Comandos para Identificación de Particiones****

Existen varias herramientas y comandos en GNU/Linux que se pueden utilizar para listar y verificar las particiones en el sistema:

* **lsblk (List Block Devices):**
  + Muestra una lista de todos los dispositivos de bloque (discos y particiones) conectados al sistema de manera jerárquica.

lsblk

* + La salida muestra los discos, sus particiones y los puntos de montaje, si los hay.
* **fdisk -l:**
  + Muestra una lista detallada de todas las particiones en los discos. Es útil para obtener una vista rápida de las particiones y su esquema.

sudo fdisk -l

* **parted (Partition Editor):**
  + Permite interactuar con las tablas de particiones de los discos. Con parted, puedes listar particiones y realizar operaciones en ellas.

sudo parted -l

* **blkid:**
  + Muestra información sobre los dispositivos de bloque, incluyendo UUIDs, etiquetas, tipos de sistema de archivos, etc.

sudo blkid

* **df -h:**
  + Muestra el uso de espacio en disco para cada sistema de archivos montado. Aunque no lista particiones no montadas, es útil para ver el espacio utilizado y libre.

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.

Para instalar un sistema operativo GNU/Linux, se requieren un mínimo de **dos particiones** para garantizar un funcionamiento adecuado. Sin embargo, en algunos casos, puede bastar con una sola partición, aunque esto no es recomendable por razones de seguridad, mantenimiento y rendimiento.

### ****1. Partición Raíz (****/****)****

* **Tipo de partición:** Primaria.
* **Identificación:** Generalmente se identifica como /dev/sda1 (o similar, dependiendo del disco y sistema).
* **Tipo de File System:** Ext4 (común en muchas distribuciones), aunque también se puede utilizar Btrfs, XFS, o ext3.
* **Punto de montaje:** / (Raíz).
* **Descripción:** Esta es la partición principal donde se instalan todos los archivos del sistema operativo, incluidos los archivos de sistema, aplicaciones y configuraciones. Es fundamental para el funcionamiento del sistema. En una instalación mínima, esta partición también contendrá los directorios de usuario (/home), archivos de arranque (/boot), etc.

### ****2. Partición de Intercambio (****swap****)****

* **Tipo de partición:** Puede ser primaria o lógica.
* **Identificación:** Por ejemplo, /dev/sda2 (o similar).
* **Tipo de File System:** No tiene un sistema de archivos tradicional; es una partición de intercambio especial.
* **Punto de montaje:** No tiene punto de montaje en el sistema de archivos. Es utilizado por el kernel para gestión de memoria virtual.
* **Descripción:** La partición de intercambio actúa como una extensión de la memoria RAM física. Se utiliza para mejorar el rendimiento del sistema cuando la RAM está llena y también para funciones como la suspensión a disco (hibernación). Aunque es posible instalar GNU/Linux sin una partición de intercambio, su presencia es altamente recomendada para evitar problemas de rendimiento y estabilidad, especialmente en sistemas con poca RAM.

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

### ****Configuración Básica para Uso de Escritorio****

Ideal para usuarios domésticos o de oficina que necesitan un sistema simple pero funcional.

* **Partición Raíz (/):**
  + **Identificación:** /dev/sda1
  + **Tipo de File System:** Ext4
  + **Punto de montaje:** /
  + **Tamaño:** 20-30 GB
  + **Descripción:** Contiene el sistema operativo y las aplicaciones. Tamaño suficiente para programas y actualizaciones.
* **Partición de Intercambio (swap):**
  + **Identificación:** /dev/sda2
  + **Tipo de File System:** Swap
  + **Punto de montaje:** No aplica
  + **Tamaño:** 2-4 GB (depende de la RAM, idealmente igual al tamaño de la RAM si es menor a 8GB)
  + **Descripción:** Utilizado para memoria virtual y suspensión a disco.
* **Partición /home:**
  + **Identificación:** /dev/sda3
  + **Tipo de File System:** Ext4
  + **Punto de montaje:** /home
  + **Tamaño:** Resto del espacio en disco disponible
  + **Descripción:** Almacena los datos y configuraciones personales de los usuarios. Facilita la preservación de datos durante actualizaciones o reinstalaciones del sistema operativo.

### ****2. Configuración para un Servidor Web****

Configuración típica para un servidor que aloja sitios web y aplicaciones.

* **Partición Raíz (/):**
  + **Identificación:** /dev/sda1
  + **Tipo de File System:** Ext4 o XFS
  + **Punto de montaje:** /
  + **Tamaño:** 20-30 GB
  + **Descripción:** Contiene archivos del sistema operativo y software básico.
* **Partición de Intercambio (swap):**
  + **Identificación:** /dev/sda2
  + **Tipo de File System:** Swap
  + **Punto de montaje:** No aplica
  + **Tamaño:** 4-8 GB (dependiendo de la RAM del servidor)
  + **Descripción:** Memoria virtual para mejorar el rendimiento en cargas de trabajo pesadas.
* **Partición /var:**
  + **Identificación:** /dev/sda3
  + **Tipo de File System:** Ext4 o XFS
  + **Punto de montaje:** /var
  + **Tamaño:** 50-100 GB (dependiendo del tamaño y número de sitios web)
  + **Descripción:** Almacena archivos de registro, sitios web y bases de datos temporales. Separa estos archivos de otros datos del sistema para mejorar la seguridad y gestión de espacio.
* **Partición /home:**
  + **Identificación:** /dev/sda4
  + **Tipo de File System:** Ext4
  + **Punto de montaje:** /home
  + **Tamaño:** Según las necesidades de los usuarios, generalmente menos prioritario en un servidor web.
* **Partición /var/www:**
  + **Identificación:** /dev/sda5
  + **Tipo de File System:** Ext4 o XFS
  + **Punto de montaje:** /var/www
  + **Tamaño:** Depende del tamaño esperado de los archivos web y las bases de datos.

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

### ****GParted (GNOME Partition Editor)****

* **Descripción:** GParted es una herramienta gráfica popular y fácil de usar basada en GTK para gestionar particiones de disco. Es parte del entorno de escritorio GNOME y permite a los usuarios crear, eliminar, redimensionar, mover, y copiar particiones.
* **Características:**
  + Soporta una amplia variedad de sistemas de archivos, incluyendo ext2, ext3, ext4, FAT16, FAT32, NTFS, y más.
  + Interfaz gráfica de usuario (GUI) intuitiva.
  + Permite operaciones de redimensionamiento y movimiento de particiones sin pérdida de datos.
  + Compatible con dispositivos de almacenamiento conectados por USB y otros medios extraíbles.
* **Ventajas:**
  + Interfaz fácil de usar, ideal para usuarios principiantes y avanzados.
  + Buen soporte de documentación y comunidad activa.
  + Disponible en formato de Live CD, lo que permite particionar discos antes de instalar un sistema operativo.
* **Desventajas:**
  + Las operaciones pueden ser lentas en discos muy grandes.
  + Limitaciones para algunas operaciones avanzadas en sistemas de archivos no soportados de manera nativa.

### ****2. fdisk (Fixed Disk)****

* **Descripción:** fdisk es una herramienta de línea de comandos utilizada para manipular la tabla de particiones. Es una de las herramientas de particionamiento más antiguas y ampliamente disponibles en casi todas las distribuciones de Linux.
* **Características:**
  + Soporte para particiones MBR (Master Boot Record).
  + Permite crear, eliminar, y listar particiones.
  + Ligero y rápido, ideal para operaciones rápidas.
* **Ventajas:**
  + Muy ligero y consume pocos recursos.
  + Disponible en casi todas las distribuciones de Linux de manera predeterminada.
  + Rápido para tareas simples de particionamiento.
* **Desventajas:**
  + Interfaz de usuario no intuitiva y basada en línea de comandos, lo que puede ser difícil para principiantes.
  + Soporte limitado para tablas de particiones GPT y sistemas de archivos modernos.

### ****3. parted (GNU Partition Editor)****

* **Descripción:** parted es una herramienta de línea de comandos más avanzada que fdisk. Es capaz de trabajar con tablas de particiones GPT (GUID Partition Table) y manejar discos de gran tamaño.
* **Características:**
  + Soporta MBR y GPT.
  + Permite redimensionar, mover, y copiar particiones, así como crear y eliminar.
  + Compatible con muchos sistemas de archivos (ext2/3/4, FAT16/32, NTFS, entre otros).
* **Ventajas:**
  + Versátil y potente, adecuado para particionamiento avanzado.
  + Más flexible que fdisk en términos de soporte de particiones GPT y manipulación de grandes volúmenes de disco.
  + Buena documentación y soporte en la comunidad de GNU/Linux.
* **Desventajas:**
  + Requiere conocimiento avanzado para su uso efectivo.
  + No tiene interfaz gráfica, lo cual puede ser un obstáculo para algunos usuarios.

### ****4. KDE Partition Manager****

* **Descripción:** Similar a GParted, KDE Partition Manager es una herramienta gráfica de gestión de particiones diseñada para el entorno de escritorio KDE.
* **Características:**
  + Soporta una amplia gama de sistemas de archivos.
  + Interfaz gráfica intuitiva y amigable, diseñada para integrarse con KDE.
  + Permite redimensionar, mover, copiar y gestionar particiones fácilmente.
* **Ventajas:**
  + Interfaz gráfica agradable e intuitiva para usuarios de KDE.
  + Facilita tareas complejas de particionamiento mediante una interfaz de usuario coherente.
* **Desventajas:**
  + Puede no estar disponible o completamente integrado en otros entornos de escritorio no basados en KDE.
  + Requiere más recursos en comparación con herramientas basadas en línea de comandos como fdisk o parted.

### ****5. cfdisk****

* **Descripción:** cfdisk es una versión más amigable de fdisk, con una interfaz de usuario basada en texto (semi-gráfica) que facilita la navegación y manipulación de particiones.
* **Características:**
  + Ofrece una interfaz de usuario más amigable en comparación con fdisk.
  + Compatible con particiones MBR.
  + Opciones de menú para crear, eliminar, y cambiar el tipo de partición.
* **Ventajas:**
  + Interfaz más fácil de usar que fdisk.
  + Ligero y rápido.
  + Disponible en la mayoría de las distribuciones de Linux por defecto.
* **Desventajas:**
  + Soporte limitado para particiones GPT.
  + Menos funcionalidades en comparación con herramientas gráficas como GParted.

### ****6. GNOME Disks (o Discos)****

* **Descripción:** Es una utilidad de gestión de discos y particiones incluida en muchas distribuciones basadas en GNOME. Permite a los usuarios realizar tareas básicas de particionamiento y gestión de discos.
* **Características:**
  + Interfaz gráfica simple y directa.
  + Ideal para particionamiento básico y tareas de configuración de discos.
  + Soporta particiones MBR y GPT.
* **Ventajas:**
  + Muy fácil de usar, adecuado para principiantes.
  + Integrado en muchas distribuciones basadas en GNOME, por lo que no requiere instalación adicional.
* **Desventajas:**
  + Capacidades limitadas en comparación con herramientas como GParted para particionamiento avanzado.
  + Menos opciones y flexibilidad para operaciones complejas.

### ****7. System Rescue CD****

* **Descripción:** No es una herramienta de particionamiento en sí, sino una distribución de Linux de rescate que incluye varias herramientas para gestión de discos y particiones, como GParted, fdisk, parted, entre otras.
* **Características:**
  + Herramienta de arranque para recuperación y gestión de sistemas de archivos.
  + Incluye múltiples utilidades para particionamiento, copia de seguridad, recuperación de datos, y diagnóstico.
* **Ventajas:**
  + Excelente para recuperación de sistemas y mantenimiento de discos.
  + Proporciona un entorno en vivo donde no se necesita un sistema operativo instalado para operar.
* **Desventajas:**
  + Requiere arrancar desde un CD/USB, lo que puede ser menos conveniente para cambios rápidos.
  + No es una herramienta dedicada, sino un conjunto de herramientas.

La elección de la herramienta de particionamiento dependerá de tus necesidades específicas y tu nivel de experiencia. GParted y KDE Partition Manager son excelentes opciones para usuarios que prefieren una interfaz gráfica. Para tareas más avanzadas o específicas, parted ofrece gran flexibilidad desde la línea de comandos. Herramientas como fdisk y cfdisk son útiles para tareas rápidas y simples, mientras que GNOME Disks es ideal para usuarios que prefieren una herramienta básica y fácil de usar.

8. Arranque (bootstrap) de un Sistema Operativo:

(a) ¿Qué es el BIOS? ¿Qué tarea realiza?

**BIOS** es el acrónimo de **"Basic Input/Output System"** (Sistema Básico de Entrada/Salida). Es un firmware fundamental que se encuentra en la placa base de las computadoras, almacenado en un chip de memoria no volátil (generalmente una memoria ROM o flash). Es uno de los primeros programas que se ejecuta cuando se enciende una computadora, antes de que se inicie el sistema operativo.

### ****Tareas y Funciones del BIOS****

1. **Autodiagnóstico de Encendido (POST, Power-On Self-Test):**  
   Una de las primeras tareas del BIOS es llevar a cabo una serie de comprobaciones iniciales del hardware, conocidas como POST. Estas pruebas incluyen verificar la memoria RAM, la integridad de la CPU, los dispositivos de almacenamiento y otros componentes esenciales para asegurarse de que todo funciona correctamente antes de arrancar el sistema operativo. Si se detecta algún problema, el BIOS generalmente mostrará un mensaje de error o emitirá una serie de pitidos para indicar el problema específico.
2. **Inicialización del Hardware:**  
   El BIOS configura y prepara los dispositivos de hardware para que el sistema operativo pueda utilizarlos. Esto incluye configurar controladores de almacenamiento, puertos de entrada/salida, tarjetas de video, teclado, y otros periféricos. El BIOS asegura que los componentes del hardware estén correctamente inicializados y listos para interactuar con el sistema operativo.
3. **Gestión del Arranque del Sistema (Bootstrapping):**  
   El BIOS determina desde qué dispositivo se iniciará el sistema operativo (por ejemplo, un disco duro, una unidad USB, o un CD/DVD). Esta tarea es conocida como "orden de arranque" o "boot order". Tras finalizar el POST, el BIOS busca un dispositivo de arranque válido y transfiere el control al gestor de arranque del sistema operativo que se encuentra en dicho dispositivo, permitiendo que el sistema operativo se cargue.
4. **Interfaz de Configuración del Sistema:**  
   El BIOS proporciona una interfaz de configuración (a menudo accesible presionando una tecla como F2, Del, o Esc durante el inicio) donde los usuarios pueden ajustar la configuración del hardware, como la velocidad del reloj del CPU, el orden de arranque, la configuración de la memoria, habilitar o deshabilitar dispositivos integrados, y otras configuraciones avanzadas de hardware.
5. **Interacción con el Firmware del Hardware:**  
   El BIOS actúa como un intermediario entre el sistema operativo y el hardware. Proporciona rutinas de software básicas que permiten al sistema operativo y al software de aplicación comunicarse con los componentes de hardware sin tener que saber los detalles específicos de cómo funciona cada dispositivo.

(b) ¿Qué es UEFI? ¿Cuál es su función?

**UEFI** (Unified Extensible Firmware Interface, por sus siglas en inglés) es una interfaz de firmware estándar moderna que se utiliza como reemplazo del BIOS tradicional en las computadoras. UEFI ofrece una interfaz más avanzada y versátil para la inicialización de hardware y el arranque de sistemas operativos. A diferencia del BIOS, que estaba basado en la tecnología de los años 80, UEFI fue diseñado para satisfacer las necesidades de las computadoras modernas, proporcionando más funcionalidades y mejor rendimiento.

### ****Funciones y Características de UEFI****

1. **Interfaz de Arranque Avanzada:** UEFI proporciona una interfaz gráfica de usuario (GUI) más moderna y fácil de usar en comparación con la interfaz de texto simple del BIOS tradicional. Esto permite una mejor experiencia de usuario y facilita la navegación y configuración del firmware.
2. **Soporte para Discos de Gran Capacidad:** UEFI utiliza la tabla de particiones GPT (GUID Partition Table) en lugar de la antigua tabla de particiones MBR (Master Boot Record). Esto permite a UEFI soportar discos de más de 2 terabytes (TB) de capacidad, superando la limitación de tamaño de partición de 2 TB que tenía MBR.
3. **Arranque Seguro (Secure Boot):** UEFI incorpora una función de seguridad llamada Secure Boot, que ayuda a proteger el sistema contra malware y rootkits durante el proceso de arranque. Secure Boot solo permite que se ejecuten programas de arranque que estén firmados digitalmente y autorizados por el fabricante de la placa base o por una entidad de confianza, lo que reduce el riesgo de ataques de software malicioso.
4. **Velocidad de Arranque Mejorada:** UEFI mejora el tiempo de arranque en comparación con el BIOS tradicional. Utiliza un proceso de inicialización más eficiente, lo que puede resultar en tiempos de inicio del sistema más rápidos.
5. **Compatibilidad con Arquitecturas Modernas:** UEFI está diseñado para ser compatible con diferentes arquitecturas de CPU (como x86, x64 y ARM), lo que facilita su implementación en una variedad de dispositivos, desde computadoras de escritorio y portátiles hasta servidores y dispositivos móviles.
6. **Gestión de Hardware y Controladores:** UEFI puede interactuar con controladores y dispositivos directamente, permitiendo una mejor gestión de hardware. Además, puede cargar controladores UEFI directamente desde el firmware, lo que mejora la compatibilidad y el soporte para nuevos dispositivos sin necesidad de actualizaciones de firmware complejas.
7. **Soporte de Red:** UEFI incluye soporte para el arranque desde una red (PXE Boot) directamente desde el firmware, lo que es especialmente útil en entornos corporativos para la instalación o actualización de sistemas operativos de forma remota.
8. **Compatibilidad con BIOS:** UEFI incluye un modo de compatibilidad llamado CSM (Compatibility Support Module), que permite que los sistemas operativos y aplicaciones diseñados para BIOS tradicional se ejecuten en un entorno UEFI. Esto garantiza una transición más suave y compatibilidad hacia adelante.

### ****Ventajas de UEFI sobre BIOS****

* **Seguridad Mejorada:** Con características como Secure Boot, UEFI proporciona un entorno de arranque más seguro contra amenazas y malware.
* **Soporte para Grandes Capacidades de Almacenamiento:** UEFI soporta particiones mayores a 2 TB gracias al uso de la tabla de particiones GPT.
* **Interfaz Gráfica de Usuario (GUI):** UEFI ofrece una interfaz gráfica más moderna que facilita la configuración y administración del sistema.
* **Flexibilidad y Escalabilidad:** UEFI es más flexible, con mejor soporte para tecnologías emergentes y dispositivos modernos, y puede ser actualizado más fácilmente que un BIOS tradicional.
* **Mejor Rendimiento:** UEFI generalmente proporciona tiempos de arranque más rápidos y mejores opciones de gestión de energía.

(c) ¿Qué es el MBR? ¿Que es el MBC?

**MBR** significa **"Master Boot Record"** o "Registro de Arranque Maestro". Es una estructura de datos especial que se encuentra en el primer sector (sector 0) de un disco duro o dispositivo de almacenamiento particionado, y es crucial para el proceso de arranque del sistema operativo en computadoras basadas en BIOS. El MBR fue introducido originalmente en los años 80 por IBM y es compatible con casi todos los sistemas operativos antiguos y modernos.

#### **Funciones del MBR:**

1. **Tabla de Particiones:**
   * El MBR contiene la tabla de particiones, que define las particiones primarias presentes en el disco. Esta tabla puede contener hasta cuatro entradas de particiones principales, o tres particiones primarias y una partición extendida (que a su vez puede contener múltiples particiones lógicas).
2. **Código de Arranque (Bootloader):**
   * El MBR también incluye un pequeño fragmento de código ejecutable llamado "gestor de arranque" o "código de arranque". Este código es el encargado de cargar el sistema operativo desde una de las particiones activas (usualmente la primera). En términos simples, este código dice a la computadora dónde buscar el sistema operativo en el disco.
3. **Identificación del Disco:**
   * El MBR almacena un identificador de disco único (Disk Signature) que ayuda a los sistemas operativos a reconocer y gestionar los discos de manera única.
4. **Límite de Tamaño:**
   * Uno de los principales inconvenientes del MBR es que sólo puede gestionar discos de hasta 2 terabytes (TB) debido a su limitación en el número de bits utilizados para almacenar la información de las particiones.

#### **Ventajas y Desventajas del MBR:**

* **Ventajas:**
  + Compatibilidad amplia con sistemas operativos antiguos y modernos.
  + Simplicidad y fiabilidad, probado durante décadas de uso.
* **Desventajas:**
  + Limitación en el número de particiones (máximo de cuatro particiones primarias).
  + No soporta discos de más de 2 TB.
  + Mayor vulnerabilidad a corrupción de datos; si el MBR se daña, el sistema no puede arrancar.

### ****¿Qué es el MBC (Master Boot Code)?****

El **MBC** (Master Boot Code) se refiere al código de arranque maestro, que es la parte del MBR que contiene el gestor de arranque básico o código de arranque inicial. Este código es responsable de iniciar el proceso de carga del sistema operativo. Es un componente crítico del MBR y suele ocupar los primeros 446 bytes del sector 0 del disco duro. Su función es localizar y cargar el cargador de arranque (bootloader) del sistema operativo desde la partición activa.

#### **Funciones del MBC:**

1. **Inicialización del Hardware:**
   * Realiza tareas básicas de inicialización y verificación de hardware antes de pasar el control al sistema operativo.
2. **Carga del Bootloader Secundario:**
   * El MBC busca la partición activa marcada en la tabla de particiones del MBR y transfiere el control al bootloader específico de esa partición. Esto podría ser un cargador de arranque específico de un sistema operativo (como GRUB, LILO, o el cargador de arranque de Windows).
3. **Transición de Control:**
   * Una vez que el MBC ha hecho su trabajo de localizar y cargar el bootloader secundario, transfiere el control a este último, que luego se encarga de iniciar el sistema operativo.

### ****Relación entre MBR y MBC:****

* **El MBR** es la estructura de datos completa que incluye tanto la tabla de particiones como el código de arranque (MBC).
* **El MBC** es solo una parte del MBR, específicamente el código de arranque que se encarga de iniciar el proceso de arranque del sistema operativo.

(d) ¿A qué hacen referencia las siglas GPT? ¿Qué sustituye? Indique cuál es su formato.

Las siglas **GPT** significan **"GUID Partition Table"** (Tabla de Particiones GUID). Es una estructura de particiones moderna que reemplaza al antiguo esquema de particiones **MBR** (Master Boot Record). GPT forma parte del estándar UEFI (Unified Extensible Firmware Interface), diseñado para superar las limitaciones del MBR y ofrecer una gestión más avanzada y flexible de las particiones en discos duros.

**Formato y Estructura:**

* **Estructura Principal:** GPT utiliza GUIDs (Globally Unique Identifiers) para identificar particiones, en lugar de los identificadores de partición del MBR. Esto permite una gestión más precisa y única de las particiones.
* **Encabezado GPT:** Al igual que el MBR, GPT tiene un encabezado de partición al principio del disco que contiene la información de la tabla de particiones. Sin embargo, también incluye un encabezado de copia de seguridad al final del disco para mejorar la protección contra la corrupción de datos.
* **Tabla de Particiones Primaria y Secundaria:** GPT mantiene dos copias de la tabla de particiones: una primaria al inicio del disco y una copia de seguridad al final del disco, lo que ayuda a mejorar la integridad y recuperación de datos en caso de fallos.

**Formato del Encabezado GPT:**

* **Encabezado GPT Principal:** Se encuentra en el primer sector del disco y contiene la información sobre la ubicación de la tabla de particiones GPT, el número total de particiones, y otros detalles relevantes.
* **Tabla de Particiones GPT:** Se encuentra justo después del encabezado GPT y contiene las entradas de partición, cada una de las cuales incluye detalles sobre el tipo de partición, el tamaño, el punto de inicio y el GUID único.
* **Encabezado GPT de Respaldo:** Se encuentra en el último sector del disco y contiene una copia de respaldo del encabezado GPT y la tabla de particiones para proporcionar una capa adicional de recuperación.

(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** (o **bootloader** en inglés) es un programa esencial que se encarga de iniciar el sistema operativo en una computadora. Su funcionalidad principal es cargar el núcleo del sistema operativo (kernel) en la memoria y transferirle el control para que pueda comenzar a ejecutarse. El gestor de arranque se ejecuta justo después del BIOS o UEFI, y su tarea es preparar el entorno para el sistema operativo.

 **Gestores de Arranque Basados en BIOS:**

* **GRUB (GRand Unified Bootloader):** Uno de los gestores de arranque más populares en sistemas Linux. GRUB puede manejar múltiples sistemas operativos y configuraciones complejas. La versión más reciente, GRUB 2, ofrece más características y flexibilidad.
* **LILO (LInux LOader):** Un gestor de arranque más antiguo y menos flexible que GRUB, pero todavía utilizado en algunos sistemas Linux. Su uso ha disminuido en favor de GRUB.

 **Gestores de Arranque Basados en UEFI:**

* **Windows Boot Manager:** Utilizado en sistemas Windows con UEFI. Gestiona el arranque de Windows y puede interactuar con otros gestores de arranque en configuraciones duales o múltiples.
* **rEFInd:** Un gestor de arranque para sistemas basados en UEFI que proporciona una interfaz gráfica para seleccionar sistemas operativos. Es útil para sistemas con múltiples sistemas operativos, especialmente en configuraciones duales de Windows y Linux.

### ****Ubicación de Instalación****

* **En el MBR (Master Boot Record):** En sistemas basados en BIOS, el gestor de arranque se instala en el primer sector del disco duro (MBR). Este sector contiene el código de arranque inicial que transfiere el control al gestor de arranque completo.
* **En la partición EFI:** En sistemas basados en UEFI, el gestor de arranque se instala en una partición especial llamada "Partición del Sistema EFI" (ESP). Esta partición está formateada con un sistema de archivos compatible con UEFI, como FAT32, y contiene los archivos necesarios para el arranque.

### ****Gestores de Arranque Conocidos****

* **GRUB (GRand Unified Bootloader):** Ampliamente utilizado en distribuciones de Linux. Soporta una amplia variedad de sistemas operativos y configuraciones avanzadas.
* **GRUB 2:** La versión más reciente de GRUB, que ofrece características adicionales y mayor flexibilidad.
* **LILO (LInux LOader):** Más antiguo que GRUB, pero todavía en uso en algunos sistemas.
* **Windows Boot Manager:** El gestor de arranque para sistemas Windows modernos, compatible con UEFI.
* **rEFInd:** Un gestor de arranque para sistemas UEFI, conocido por su interfaz gráfica y su capacidad para manejar múltiples sistemas operativos

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

### ****1. Encendido del Hardware****

* **Encendido de la Computadora:**
  + Cuando se enciende la computadora, se suministra energía a todos los componentes del hardware.

### ****2. Ejecución del POST (Power-On Self-Test)****

* **Autodiagnóstico de Encendido:**
  + El BIOS/UEFI realiza un POST para verificar que el hardware esencial, como la memoria RAM, el procesador y los dispositivos de almacenamiento, esté funcionando correctamente. Si se detectan errores, el sistema puede emitir pitidos o mostrar mensajes de error.

### ****3. Inicialización del Hardware****

* **Configuración del Hardware:**
  + El BIOS/UEFI configura el hardware básico, como el teclado, la pantalla y los discos duros, para asegurarse de que todos los dispositivos estén listos para funcionar.

### ****4. Carga del Gestor de Arranque****

* **Ubicación del Gestor de Arranque:**
  + El BIOS/UEFI busca el gestor de arranque en el sector de arranque del disco duro. En sistemas BIOS, esto es el MBR (Master Boot Record), mientras que en sistemas UEFI, es la partición del sistema EFI (ESP).
* **Ejecutar el Gestor de Arranque:**
  + El BIOS/UEFI transfiere el control al gestor de arranque, que es responsable de iniciar el sistema operativo. Este paso puede implicar la carga del código de arranque desde el MBR o la partición EFI.

### ****5. Carga del Núcleo del Sistema Operativo****

* **Gestión de Particiones:**
  + El gestor de arranque lee la tabla de particiones para encontrar la partición donde está instalado el sistema operativo y carga el núcleo (kernel) del sistema operativo desde esa partición.
* **Cargar el Núcleo:**
  + El gestor de arranque carga el archivo del núcleo del sistema operativo en la memoria. El núcleo es el componente central del sistema operativo, responsable de gestionar el hardware y los recursos del sistema.

### ****6. Transferencia de Control al Núcleo del Sistema Operativo****

* **Inicio del Núcleo:**
  + Una vez que el núcleo está cargado en la memoria, el gestor de arranque transfiere el control al núcleo del sistema operativo. El núcleo comienza su proceso de inicialización.

### ****7. Inicialización del Sistema Operativo****

* **Configuración del Núcleo:**
  + El núcleo del sistema operativo comienza a configurar el hardware y los controladores necesarios. Esto incluye la inicialización de dispositivos, la gestión de la memoria, y el establecimiento de la comunicación entre el hardware y el software.
* **Carga de Servicios y Controladores:**
  + El núcleo carga los servicios y controladores necesarios para que el sistema operativo funcione correctamente. Esto incluye la configuración de redes, dispositivos de almacenamiento y otros periféricos.

### ****8. Inicio de los Procesos de Usuario****

* **Ejecutar el Proceso Inicial:**
  + El núcleo inicia el proceso inicial del sistema operativo, que es el primer proceso que se ejecuta después de la carga del núcleo (en Linux, este es el proceso init o systemd).
* **Inicialización de la Interfaz de Usuario:**
  + El proceso inicial puede iniciar otros procesos de usuario y servicios esenciales. En un sistema gráfico, esto incluirá el gestor de ventanas y otros componentes del entorno de escritorio.
* **Mostrar el Escritorio:**
  + Finalmente, el sistema operativo completa el proceso de arranque y muestra la interfaz gráfica de usuario (escritorio) o la línea de comandos, dependiendo de la configuración del sistema.

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

### ****Encendido del Hardware****

* **Suministro de Energía:**
  + Cuando se enciende la computadora, el hardware recibe energía y comienza a funcionar.

### ****2. Ejecución del POST (Power-On Self-Test)****

* **Autodiagnóstico:**
  + El BIOS o UEFI realiza un POST para verificar el estado del hardware básico (memoria, procesador, etc.). Si todo está en orden, el proceso continúa; si se detectan problemas, se puede mostrar un mensaje de error o emitir pitidos.

### ****3. Inicialización del Hardware****

* **Configuración de Hardware:**
  + El BIOS/UEFI configura el hardware básico, como la CPU, la memoria RAM, y los dispositivos de almacenamiento, para prepararse para el arranque.

### ****4. Carga del Gestor de Arranque (Bootloader)****

* **Ubicación del Gestor de Arranque:**
  + El BIOS/UEFI busca el gestor de arranque en el sector de arranque del disco duro. En sistemas con BIOS, esto es el MBR (Master Boot Record), mientras que en sistemas con UEFI, se busca en la partición del sistema EFI (ESP).
* **Ejecutar el Gestor de Arranque:**
  + El BIOS/UEFI transfiere el control al gestor de arranque, que es el programa que prepara el entorno para el sistema operativo. En GNU/Linux, el gestor de arranque más común es **GRUB (GRand Unified Bootloader)**.

### ****5. Selección del Núcleo del Sistema Operativo****

* **Menú de Arranque (si aplica):**
  + GRUB (u otro gestor de arranque) puede presentar un menú que permite al usuario seleccionar el sistema operativo o la versión del núcleo que desea iniciar. Si no se realiza una selección, el gestor de arranque carga el núcleo predeterminado.
* **Carga del Núcleo:**
  + El gestor de arranque carga el núcleo del sistema operativo (kernel) desde la partición del sistema de archivos. En el caso de GRUB, esto implica leer la configuración de arranque y cargar el archivo del núcleo en la memoria.

### ****6. Transferencia de Control al Núcleo****

* **Inicio del Núcleo:**
  + Una vez cargado en la memoria, el gestor de arranque transfiere el control al núcleo del sistema operativo. El núcleo comienza a inicializar el hardware y establecer el entorno para los procesos de usuario.

### ****7. Inicialización del Núcleo****

* **Configuración del Hardware:**
  + El núcleo inicializa y configura el hardware, cargando controladores necesarios para el funcionamiento de dispositivos como discos duros, tarjetas de red y otros periféricos.
* **Montaje del Sistema de Archivos Raíz:**
  + El núcleo monta el sistema de archivos raíz, que es la base para todas las operaciones del sistema operativo. El punto de montaje principal es /, y es donde se encuentran los archivos esenciales del sistema.

### ****8. Ejecución del Proceso Init****

* **Lanzamiento del Proceso Init:**
  + El núcleo inicia el primer proceso del sistema operativo, conocido como **init** o **systemd** (en sistemas modernos). Este proceso es responsable de iniciar y gestionar otros servicios y procesos del sistema.
* **Configuración del Entorno:**
  + Init/systemd lee los scripts de inicio y archivos de configuración para iniciar servicios esenciales del sistema, como el servicio de red, el servidor de impresión, y otros daemons.

### ****9. Inicialización de Servicios y Daemons****

* **Inicio de Servicios:**
  + Init/systemd arranca servicios del sistema que son necesarios para el funcionamiento del sistema operativo, incluyendo servicios de red, almacenamiento, y otros componentes esenciales.
* **Cargar el Entorno de Usuario:**
  + El sistema operativo inicia el entorno de usuario, que puede ser una interfaz gráfica (como GNOME, KDE, XFCE) o una línea de comandos, dependiendo de la configuración del sistema.

### ****10. Mostrar la Interfaz de Usuario****

* **Interfaz Gráfica o Línea de Comandos:**
  + Finalmente, el sistema operativo muestra la interfaz de usuario al usuario. En sistemas gráficos, se inicia el gestor de ventanas y se muestra el escritorio. En sistemas de línea de comandos, se presenta un terminal o consola.

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

El proceso de parada (shutdown) en GNU/Linux es el proceso que se lleva a cabo para cerrar de manera segura el sistema operativo y apagar el hardware. Este proceso garantiza que todos los datos se guarden correctamente y que el sistema se apague de manera ordenada

### ****1. Solicitud de Apagado****

* **Comando de Apagado:**
  + El proceso de parada generalmente comienza cuando un usuario o un administrador emite un comando para apagar el sistema, como shutdown, poweroff, halt, o mediante una opción en el entorno gráfico. Este comando solicita al sistema que inicie el proceso de apagado.

### ****2. Envío de Señales a los Procesos****

* **Envío de Señales de Terminación:**
  + El sistema operativo envía señales a los procesos en ejecución para que se terminen de manera ordenada. La señal SIGTERM (terminar) se envía primero, dando a los procesos la oportunidad de finalizar correctamente y guardar cualquier dato necesario.
* **Espera para la Terminación de Procesos:**
  + El sistema espera un período de tiempo para que los procesos respondan a la señal SIGTERM. Si algunos procesos no se detienen después de este tiempo, se les envía una señal SIGKILL (matar) para forzarlos a detenerse.

### ****3. Desmontaje de Sistemas de Archivos****

* **Desmontaje de Sistemas de Archivos:**
  + El sistema operativo desmonta todos los sistemas de archivos montados. Esto incluye cerrar de manera segura todos los archivos abiertos y asegurarse de que todos los datos se hayan escrito en el disco. Este paso es crucial para evitar la corrupción de datos.

### ****4. Detención de Servicios y Daemons****

* **Parada de Servicios y Daemons:**
  + El sistema operativo detiene todos los servicios y daemons en ejecución. Estos servicios incluyen servidores de red, bases de datos, y otros procesos de fondo necesarios para el funcionamiento del sistema.

### ****5. Sincronización de Datos****

* **Sincronización de Datos:**
  + El sistema operativo sincroniza todos los datos en la memoria con el disco para asegurarse de que no se pierdan cambios. Esto se hace utilizando el comando sync para escribir todos los datos pendientes al disco.

### ****6. Detención del Núcleo del Sistema Operativo****

* **Parada del Núcleo:**
  + Una vez que todos los procesos, servicios y sistemas de archivos están detenidos, el núcleo del sistema operativo finaliza su operación. Este paso es el último antes de apagar el hardware.

### ****7. Apagado del Hardware****

* **Envío de Comando de Apagado al Hardware:**
  + El sistema operativo envía un comando al hardware para apagar la computadora. En sistemas basados en BIOS, esto puede implicar el uso de instrucciones específicas del hardware. En sistemas basados en UEFI, el proceso puede ser ligeramente diferente.
* **Desconexión de Energía:**
  + Finalmente, el hardware recibe la señal para cortar la energía y apagar el sistema. En este punto, la computadora se apaga completamente.

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

Este tipo de configuración se conoce como **arranque múltiple** o **dual-boot, donde** la clave es gestionar adecuadamente el particionamiento del disco y configurar correctamente el gestor de arranque para permitir la selección entre los sistemas operativos durante el arranque. Con la configuración adecuada, puedes aprovechar las ventajas de ambos sistemas operativos en una sola máquina.

### ****1. Espacio en Disco y Particionamiento****

Para instalar múltiples sistemas operativos en una misma PC, es necesario gestionar el espacio en disco de manera adecuada. Esto implica:

* **Particionamiento del Disco:**
  + Debes crear particiones separadas en el disco para cada sistema operativo. Cada partición debe ser formateada con un sistema de archivos compatible con el sistema operativo que la utilizará. Por ejemplo, puedes tener una partición para GNU/Linux y otra para Windows.
* **Espacio en Disco:**
  + Asegúrate de que hay suficiente espacio en disco para cada sistema operativo y para sus datos. Además, es útil dejar espacio libre para los datos compartidos si planeas usar ambos sistemas operativos.

### ****2. Instalación de Sistemas Operativos****

* **Instalación del Primer Sistema Operativo:**
  + Generalmente, se recomienda instalar el primer sistema operativo primero, especialmente si es Windows. Esto es porque Windows suele instalar su propio gestor de arranque y puede sobrescribir cualquier gestor de arranque existente.
* **Instalación del Segundo Sistema Operativo:**
  + Después de instalar el primer sistema operativo, instala el segundo sistema operativo (por ejemplo, GNU/Linux). Durante la instalación de GNU/Linux, el instalador generalmente detectará la presencia del otro sistema operativo y configurará el gestor de arranque (como GRUB) para permitir la selección entre los sistemas operativos durante el arranque.

### ****3. Configuración del Gestor de Arranque****

* **Gestor de Arranque Común:**
  + El gestor de arranque se encarga de permitir la selección del sistema operativo al inicio. En una configuración con GNU/Linux y otro sistema operativo, como Windows, GRUB (el gestor de arranque común en GNU/Linux) suele ser instalado y configurado para manejar ambos sistemas operativos. GRUB detecta la instalación de otros sistemas operativos y añade opciones en su menú para seleccionar entre ellos.
* **Modificación de la Configuración:**
  + Después de instalar GNU/Linux, puedes necesitar ajustar la configuración del gestor de arranque para asegurarte de que todos los sistemas operativos aparezcan en el menú y se inicien correctamente.

9. Archivos:

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

En GNU/Linux, los archivos se identifican mediante una combinación de varios atributos y propiedades que permiten al sistema y a los usuarios gestionar y acceder a ellos.

### ****1. Ruta del Archivo (Path)****

* **Ruta Absoluta:**
  + Una ruta absoluta especifica la ubicación completa del archivo desde el directorio raíz del sistema de archivos. Comienza con /, el directorio raíz. Por ejemplo, /home/usuario/documentos/archivo.txt es una ruta absoluta.
* **Ruta Relativa:**
  + Una ruta relativa especifica la ubicación del archivo en relación con el directorio de trabajo actual. Por ejemplo, si estás en /home/usuario, una ruta relativa para acceder a archivo.txt en el directorio documentos sería documentos/archivo.txt.

### ****2. Nombre del Archivo****

* **Nombre:**
  + El nombre del archivo es la parte del identificador que sigue al último / en una ruta. Por ejemplo, en /home/usuario/documentos/archivo.txt, archivo.txt es el nombre del archivo.

### ****3. Inodo****

* **Número de Inodo:**
  + Cada archivo en un sistema de archivos de GNU/Linux tiene un identificador único conocido como **número de inodo**. El inodo almacena información sobre el archivo, como su ubicación en el disco, permisos, propietario, y fechas de creación y modificación.
  + Puedes encontrar el número de inodo de un archivo utilizando el comando ls -i. Por ejemplo, ls -i archivo.txt muestra el número de inodo de archivo.txt.

### ****4. Permisos y Propietario****

* **Permisos:**
  + Los permisos de archivo determinan quién puede leer, escribir o ejecutar el archivo. Se representan como una serie de caracteres en la salida de ls -l y se dividen en permisos para el propietario del archivo, el grupo asociado y otros usuarios. Por ejemplo, -rw-r--r-- indica permisos de lectura y escritura para el propietario y solo lectura para el grupo y otros usuarios.
* **Propietario y Grupo:**
  + Cada archivo tiene un propietario y un grupo asociado. Esta información se muestra en la salida de ls -l y ayuda a determinar quién puede acceder y modificar el archivo.

### ****5. Tipo de Archivo****

* **Tipo de Archivo:**
  + El tipo de archivo se indica al principio de la línea en la salida de ls -l. Por ejemplo:
    - - para archivos regulares.
    - d para directorios.
    - l para enlaces simbólicos.
    - c para archivos de dispositivo de carácter.
    - b para archivos de dispositivo de bloque.

### ****6. Tamaño del Archivo****

* **Tamaño:**
  + El tamaño del archivo se muestra en bytes en la salida de ls -l. Indica cuántos bytes de datos contiene el archivo.

### ****7. Fechas de Modificación****

* **Fechas:**
  + Se registran varias fechas para cada archivo, como:
    - **Fecha de Creación:** La fecha en que el archivo fue creado (en sistemas de archivos que soportan esta información).
    - **Fecha de Última Modificación:** La última vez que el contenido del archivo fue modificado.
    - **Fecha de Último Acceso:** La última vez que el archivo fue accedido o leído.

### ****8. Enlaces (Links)****

* **Enlaces Duros:**
  + Un enlace duro es una referencia adicional al mismo inodo. Los enlaces duros tienen el mismo número de inodo y apuntan al mismo archivo en el sistema de archivos.
* **Enlaces Simbólicos (Symlinks):**
  + Un enlace simbólico es un archivo que actúa como un puntero a otro archivo o directorio. Se puede identificar por el prefijo l en ls -l y muestra la ruta al archivo al que apunta.

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

**1. Editor vi**

vi es uno de los editores de texto más antiguos y utilizados en sistemas Unix y Linux. Es un editor de texto modal, lo que significa que tiene diferentes modos de operación (normal, inserción, visual, comando) para editar texto y navegar por el archivo.

* **Modos principales de vi:**
  + **Modo Normal:** Es el modo por defecto al abrir vi. En este modo, puedes navegar por el archivo usando las teclas de dirección, y ejecutar comandos.
  + **Modo de Inserción:** Permite editar el texto. Para entrar en este modo, presiona i en el modo normal. Puedes escribir texto directamente. Para salir de este modo y volver al modo normal, presiona Esc.
  + **Modo de Comando:** Aquí puedes ejecutar comandos para tareas como guardar, salir, buscar, etc. Para entrar en este modo desde el modo normal, presiona : (dos puntos).
* **Comandos comunes en vi:**
  + i: Entra en modo de inserción antes del cursor.
  + Esc: Vuelve al modo normal.
  + :w: Guarda el archivo.
  + :q: Sale de vi.
  + :wq: Guarda y sale.
  + :q!: Sale sin guardar cambios.
  + /texto: Busca "texto" en el archivo.
  + dd: Elimina una línea.
  + yy: Copia una línea.
  + p: Pega el contenido copiado.

vi es popular por su eficiencia y disponibilidad en prácticamente todos los sistemas Unix/Linux, aunque puede ser difícil de aprender para los principiantes debido a su interfaz no intuitiva y dependiente de comandos.

**2. Editor mcedit**

mcedit es el editor de texto integrado en Midnight Commander (MC), un administrador de archivos de texto basado en consola similar a Norton Commander.

* **Interfaz:** mcedit ofrece una interfaz más amigable que vi, similar a otros editores de texto convencionales. Tiene un menú en la parte superior que facilita el acceso a funciones comunes como abrir, guardar, buscar y reemplazar.
* **Uso básico:** Puedes navegar por el archivo utilizando las teclas de flecha, y editar texto directamente sin necesidad de cambiar de modos.
* **Comandos comunes en mcedit:**
  + F2: Guarda el archivo.
  + F10: Sale del editor.
  + F3: Busca en el archivo.
  + F4: Reemplaza texto.
  + F5: Copia texto.
  + F6: Mueve texto.

mcedit es ideal para usuarios que prefieren una interfaz más intuitiva y menos dependiente de comandos en comparación con vi.

**3. Comando cat**

cat (abreviatura de "concatenate") es un comando en Linux utilizado para concatenar y mostrar el contenido de archivos.

* **Funciones principales:**
  + Mostrar el contenido de un archivo:

cat archivo.txt

* + Concatenar varios archivos y mostrarlos juntos:

cat archivo1.txt archivo2.txt

* + Crear un nuevo archivo concatenando otros archivos:

cat archivo1.txt archivo2.txt > nuevoarchivo.txt

* + Añadir contenido a un archivo existente:

cat archivo1.txt >> archivoexistente.txt

cat es útil para visualizar rápidamente el contenido de archivos pequeños o combinar el contenido de varios archivos.

**4. Comando more**

more es un comando utilizado para visualizar el contenido de archivos de texto de manera interactiva, permitiendo navegar por el archivo página a página.

* **Uso básico:**

more archivo.txt

Muestra el contenido del archivo pantalla por pantalla. Presiona Espacio para avanzar una página, Enter para avanzar una línea, y q para salir.

* **Características de more:**
  + Permite visualizar archivos largos que no caben en una sola pantalla.
  + Es más limitado en comparación con less, otro comando similar que permite desplazarse tanto hacia adelante como hacia atrás en el archivo.

more es útil para visualizar grandes archivos de texto sin abrumar al usuario mostrando todo el contenido a la vez

(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.

Ahora lo testeo

<https://youtu.be/VO6EoN-a7QA?si=LsYFFVVmq0okpvlC> (preguntar)

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

El comando file en GNU/Linux se utiliza para determinar el tipo de un archivo. Este comando analiza el contenido de un archivo para identificar su tipo, en lugar de basarse únicamente en la extensión del archivo.

El comando file funciona leyendo ciertos datos específicos dentro del archivo para determinar su tipo. Utiliza una base de datos de "magias" o firmas de archivos conocidas que permiten identificar diferentes tipos de archivos, como archivos de texto, binarios ejecutables, imágenes, archivos comprimidos, etc. Estos datos se encuentran en una base de datos llamada "magic" que el comando consulta.

Ejemplo de uso: file documento.txt

### ****Cómo Determina**** file ****el Tipo de Archivo****

El comando file utiliza un enfoque en varios pasos para determinar el tipo de archivo:

1. **Pruebas de Sistema de Archivos:** Primero, verifica si el archivo es un tipo especial del sistema de archivos, como un enlace simbólico o un archivo de dispositivo.
2. **Pruebas de Magia:** Luego, usa una base de datos de patrones (conocida como la base de datos de "magia") que contiene firmas para varios tipos de archivos. Esta prueba intenta encontrar patrones específicos en el archivo que coincidan con las firmas en la base de datos.
3. **Pruebas de Lenguaje:** Si las pruebas anteriores fallan, el comando file analiza el contenido del archivo para determinar si es un archivo de texto de un tipo específico, como código fuente de un lenguaje de programación.

10.

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

(a) Cree la carpeta ISO2017 mkdir ISO2017

(b) Acceda a la carpeta (cd) cd ISO2017

(c) Cree dos archivos con los nombres iso2017-1 e iso2017-2 (touch)

touch iso2017-1 iso2017-2

(d) Liste el contenido del directorio actual (ls) ls

(e) Visualizar la ruta donde estoy situado (pwd) pwd

(f) Busque todos los archivos en los que su nombre contiene la cadena “iso\*” (find)

find . -name "iso\*"

(g) Informar la cantidad de espacio libre en disco (df)

df -h

(h) Verifique los usuarios conectado al sistema (who) who

(i) Acceder al archivo iso2017-1 e ingresar Nombre y Apellido

Vi iso2017 y escribir nombre y apellido

(j) Mostrar en pantalla las últimas líneas de un archivo (tail). tail iso2017-1

11.

Investigue su funcionamiento y parámetros más importantes:

(a) shutdown

Se utiliza para apagar o reiniciar el sistema de manera segura.

**Parámetros comunes:**

* -h o --halt: Apaga el sistema.
* -r o --reboot: Reinicia el sistema después de apagar.
* -c: Cancela un apagado programado.
* -P: Apaga el sistema (igual que -h).
* now: Apaga el sistema inmediatamente.
* +m: Define el tiempo de apagado en minutos desde el momento actual.
* [mensaje]: Envía un mensaje a todos los usuarios antes de apagar.

(b) reboot

Reinicia el sistema.

**Parámetros comunes:**

* -f o --force: Fuerza el reinicio sin invocar el comando shutdown.
* -p: Apaga la alimentación al reiniciar.

(c) halt

Detiene el sistema. Es equivalente a shutdown -h.

**Parámetros comunes:**

* -f: Fuerza el apagado sin invocar el comando shutdown.
* --halt: Detiene el sistema sin apagar la alimentación

(d) locate

Busca archivos en el sistema mediante una base de datos previamente indexada.

**Parámetros comunes:**

* -i: Ignora mayúsculas y minúsculas.
* -r: Utiliza expresiones regulares para buscar.
* -c: Muestra el número de coincidencias encontradas.

(e) uname

Muestra información sobre el sistema operativo y el núcleo

**Parámetros comunes:**

* -a: Muestra toda la información disponible.
* -r: Muestra la versión del kernel.
* -n: Muestra el nombre de red del sistema.
* -m: Muestra el tipo de hardware.

(f) gmesg

Muestra los mensajes del kernel.

 **Parámetros comunes:**

* -c: Limpia el búfer del mensaje después de mostrarlo.
* -T: Muestra la marca de tiempo legible en el formato de tiempo humano.

(g) lspci

Lista la información de todos los dispositivos PCI en el sistema.

**Parámetros comunes:**

* -v: Muestra información detallada.
* -vv: Muestra información aún más detallada.
* -k: Muestra información sobre los controladores de los dispositivos.

(h) at

Programa tareas para ejecutarse en un momento específico.

**Parámetros comunes:**

* atq: Muestra las tareas pendientes en la cola de at.
* atrm: Elimina una tarea de la cola de at.

(i) netstat

Muestra estadísticas de red, incluyendo conexiones activas y tablas de enrutamiento.

**Parámetros comunes:**

* -a: Muestra todas las conexiones (incluyendo las de escucha).
* -t: Muestra solo las conexiones TCP.
* -u: Muestra solo las conexiones UDP.
* -n: Muestra direcciones y puertos en formato numérico.
* -r: Muestra la tabla de enrutamiento.

(j) mount

Se utiliza para montar sistemas de archivos en directorios específicos.

**Parámetros comunes:**

* -t: Especifica el tipo de sistema de archivos.
* -o: Especifica opciones de montaje (ej., ro para solo lectura).
* -a: Monta todos los sistemas de archivos definidos en /etc/fstab.

(k) umount

Se utiliza para desmontar sistemas de archivos.

**Parámetros comunes:**

* -l: Desmontaje perezoso, que desmonta cuando no está en uso.
* -f: Fuerza el desmontaje.

(l) head

Muestra las primeras líneas de un archivo.

**Parámetros comunes:**

* -n: Especifica el número de líneas a mostrar

(m) losetup

Configura y controla dispositivos de bucle (loop devices).

**Parámetros comunes:**

* -a: Muestra todos los dispositivos de bucle activos.
* -f: Encuentra un dispositivo de bucle libre y lo asigna.
* -d: Des

(n) write

Envía un mensaje a otro usuario que está conectado al sistema.

EJ: write usuario [tty]

(ñ) mkfs

Se utiliza para crear un sistema de archivos en una partición.

**Parámetros comunes:**

* -t: Especifica el tipo de sistema de archivos (e.g., ext4, vfat).

(o) fdisk (con cuidado)

Es una herramienta de particionamiento de discos.

**Parámetros comunes:**

* -l: Lista las particiones en un dispositivo.
* -n: Añade una nueva partición.
* -d: Borra una partición existente.
* -t: Cambia el tipo de partición.

12.

Investigue su funcionamiento y parámetros más importantes:

1. Indique en qué directorios se almacenan los comandos mencionados en el ejercicio anterior.
2. **/bin**: Este directorio contiene comandos esenciales que están disponibles tanto para usuarios normales como para el usuario root. Aquí se encuentran comandos básicos que son fundamentales para el funcionamiento del sistema, incluso en modo de usuario único (single-user mode).
3. **/sbin**: Contiene comandos de administración de sistemas esenciales que normalmente solo son utilizados por el superusuario (root). Los comandos relacionados con la administración del sistema y el arranque, como shutdown, reboot, halt, mkfs, y fdisk, suelen encontrarse aquí.
4. **/usr/bin**: Este directorio alberga la mayoría de los comandos de usuario que no son esenciales para el arranque o la recuperación del sistema, pero que son necesarios para las operaciones diarias. Comandos como locate, uname, dmesg, head, y netstat suelen estar en este directorio.
5. **/usr/sbin**: Contiene comandos de administración de sistema que no son críticos para el funcionamiento básico del sistema pero que son importantes para la administración avanzada del sistema, como lspci, at, y otros comandos de red o configuración de hardware.

### ****Ubicaciones Comunes de Algunos de los Comandos Mencionados:****

* **Comandos de administración y arranque del sistema**: (shutdown, reboot, halt)
  + Ubicación común: /sbin
* **Comandos de gestión de archivos y sistemas de archivos**: (mkfs, fdisk, mount, umount)
  + Ubicación común: /sbin y /usr/sbin
* **Comandos de información y diagnóstico del sistema**: (uname, dmesg, lspci, netstat, head)
  + Ubicación común: /bin y /usr/bin
* **Comandos de programación de tareas y mensajes**: (at, write)
  + Ubicación común: /usr/bin
* **Comandos de gestión de dispositivos de bucle**: (losetup)
  + Ubicación común: /sbin

Para verificar la ubicación específica de un comando en tu sistema, se puede utilizar el comando which:ej: which fdisk.