

ISO prac 1

1. Características de GNU/Linux:

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

Es un sistema operativo de código abierto, su código fuente está disponible para cualquier persona que quiera usarlo o modificarlo o distribuirlo, esto no significa que sea gratis

Es estable y confiable

Es seguro ya que tiene una buena arquitectura en términos de seguridad, tiene permisos de usuario estrictos y la comunidad ayuda a detectar y corregir errores.

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

Por ejemplo windows es un sistema operativo propietario entonces su código no está abierto a la comunidad, no se puede modificar, etc

Windows también fue históricamente criticado por su poca confiabilidad y no es tan seguro por su gran cuota de mercado. Y no es tan personalizable como linux.

Otro ejemplo sería macOS

Es también un sistema propietario pero usa componentes de código abierto como el núcleo Darwin.

Es muy estable y confiable igual que linux pero puede variar su velocidad con el hardware. Es menos personalizable que linux. Y aunque se lo conoce por enfocarse en la seguridad no es inmune a virus y malware.

(c) ¿Qué es GNU?

El sistema GNU es el sistema operativo similar a Unix, constituido en su totalidad por software libre. GNU'S NOT UNIX (GNU). Aunque GNU iba a ser compatible con Unix, no sería lo mismo porque sería libre.

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

En 1983, Richard Stallman, un programador y activista por la libertad del software, anunció el proyecto GNU con la intención de crear un sistema operativo completamente libre, similar a Unix pero sin software propietario.

En 1991 se creó el núcleo Linux, el núcleo funcional por Linus Torvalds.

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

La multitarea es la capacidad de un sistema operativo para ejecutar múltiples procesos o tareas al mismo tiempo. Tiene 2 tipos la cooperativa que los procesos ceden voluntariamente el control al sistema operativo para que se pueda ejecutar otro proceso y la preventiva que el sistema operativo decide cuanto tiempo se le asigna a cada proceso.

Linux hace uso de la multitarea preventiva porque este es conocido por poder gestionar eficientemente múltiples procesos al mismo tiempo.

(f) ¿Qué es POSIX?

POSIX (Portable Operating System Interface) es un conjunto de estándares desarrollados por la IEEE (Institute of Electrical and Electronics Engineers) para mantener la compatibilidad entre sistemas operativos. Estos estándares definen la interfaz de programación de aplicaciones (API), comandos y utilidades comunes que deben cumplir los sistemas operativos para ser considerados compatibles 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 de GNU/Linux es una versión completa de un sistema operativo basada en el núcleo Linux y en componentes del sistema GNU, que incluye herramientas y aplicaciones que permiten a los usuarios interactuar con la máquina. Estas distribuciones pueden variar en términos de administración de paquetes, entornos de escritorio, configuraciones de seguridad y objetivos de uso, adaptándose a diferentes necesidades y preferencias de los usuarios.

Ubuntu:

- **Tipo:** Distribución basada en Debian.
- **Gestor de paquetes:** APT (Advanced Package Tool).
- **Entorno de escritorio:** GNOME (anteriormente usaba Unity).
- **Uso:** Orientada a usuarios novatos y desktops; ofrece facilidad de uso y un gran soporte comunitario.
- **Ciclo de lanzamiento:** Versiones cada 6 meses, con versiones LTS (Long Term Support) cada 2 años.

Fedora:

- **Tipo:** Distribución patrocinada por Red Hat.
- **Gestor de paquetes:** DNF (Dandified YUM).
- **Entorno de escritorio:** GNOME (por defecto, pero se pueden instalar otros).
- **Uso:** Enfocada en ofrecer las últimas características y tecnologías; ideal para desarrolladores y usuarios avanzados.
- **Ciclo de lanzamiento:** Versiones cada 6 meses.

Debian:

- **Tipo:** Una de las distribuciones más antiguas y estables.
- **Gestor de paquetes:** APT.
- **Entorno de escritorio:** Múltiples opciones (GNOME, KDE, XFCE, entre otros).
- **Uso:** Conocida por su estabilidad y la gran cantidad de paquetes disponibles; utilizada tanto en servidores como en desktops.
- **Ciclo de lanzamiento:** Lanzamientos menos frecuentes, con un enfoque en la estabilidad.

Arch Linux:

- **Tipo:** Distribución de rolling release (actualizaciones continuas).
- **Gestor de paquetes:** Pacman.
- **Entorno de escritorio:** Múltiples opciones, el usuario elige su entorno al instalar.
- **Uso:** Orientada a usuarios avanzados; permite una personalización total del sistema, ideal para quienes desean aprender más sobre Linux.
- **Ciclo de lanzamiento:** No tiene un ciclo de lanzamiento fijo; se actualiza constantemente.

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

Ubuntu:

- **Tipo:** Distribución basada en Debian.
- **Gestor de paquetes:** APT (Advanced Package Tool).

- **Entorno de escritorio:** GNOME (anteriormente usaba Unity).
- **Uso:** Orientada a usuarios novatos y desktops; ofrece facilidad de uso y un gran soporte comunitario.
- **Ciclo de lanzamiento:** Versiones cada 6 meses, con versiones LTS (Long Term Support) cada 2 años.

Fedora:

- **Tipo:** Distribución patrocinada por Red Hat.
- **Gestor de paquetes:** DNF (Dandified YUM).
- **Entorno de escritorio:** GNOME (por defecto, pero se pueden instalar otros).
- **Uso:** Enfocada en ofrecer las últimas características y tecnologías; ideal para desarrolladores y usuarios avanzados.
- **Ciclo de lanzamiento:** Versiones cada 6 meses.

Debian:

- **Tipo:** Una de las distribuciones más antiguas y estables.
- **Gestor de paquetes:** APT.
- **Entorno de escritorio:** Múltiples opciones (GNOME, KDE, XFCE, entre otros).
- **Uso:** Conocida por su estabilidad y la gran cantidad de paquetes disponibles; utilizada tanto en servidores como en desktops.
- **Ciclo de lanzamiento:** Lanzamientos menos frecuentes, con un enfoque en la estabilidad.

Arch Linux:

- **Tipo:** Distribución de rolling release (actualizaciones continuas).
- **Gestor de paquetes:** Pacman.
- **Entorno de escritorio:** Múltiples opciones, el usuario elige su entorno al instalar.
- **Uso:** Orientada a usuarios avanzados; permite una personalización total del sistema, ideal para quienes desean aprender más sobre Linux.
- **Ciclo de lanzamiento:** No tiene un ciclo de lanzamiento fijo; se actualiza constantemente.

(c) ¿Qué es Debian? Acceda a <https://www.debian.org/intro/about> e indique cuáles son los objetivos del proyecto y una breve cronología del mismo

Debian es una distribución de sistema operativo de código abierto que se basa en el núcleo Linux y en herramientas del proyecto GNU. Los objetivos del proyecto Debian incluyen proporcionar un sistema operativo completamente libre, mantener un entorno estable y seguro, y fomentar la colaboración entre los desarrolladores.

Cronología Breve:

- 1993: Ian Murdock inicia el proyecto Debian.
- 1996: Se lanza la primera versión oficial (Debian 1.1).
- 1999: Se adopta el Sistema de Gestión de Paquetes APT.
- 2005: Se establece la Social Contract que formaliza el compromiso con el software libre.

3. Estructura de GNU/Linux:

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

Los tres componentes fundamentales de GNU/Linux son:

1. El núcleo (Kernel): Es el corazón del sistema operativo, que gestiona el hardware y proporciona servicios esenciales a las aplicaciones.
2. Las herramientas del sistema GNU: Incluyen bibliotecas, utilidades y herramientas que permiten a los usuarios interactuar con el sistema y realizar tareas básicas.
3. Los entornos de escritorio: Proporcionan la interfaz gráfica de usuario, facilitando la interacción del usuario con el sistema, como GNOME, KDE o XFCE.

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

Núcleo (Kernel): Controla el hardware y gestiona recursos como la memoria y los procesos.

Sistema de archivos: Organiza y gestiona el almacenamiento de datos, facilitando el acceso y la manipulación de archivos y directorios.

Interfaz de usuario: Puede ser gráfica (GUI) o de línea de comandos (CLI), permitiendo la interacción entre el usuario y el sistema.

Utilidades del sistema: Herramientas y programas que permiten realizar tareas específicas y administrar el sistema operativo.

4. Kernel:

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

Es el núcleo del sistema operativo, responsable de gestionar el hardware y proporcionar servicios esenciales a las aplicaciones.

Reseña histórica: El desarrollo del kernel de GNU/Linux comenzó en 1991 con Linus Torvalds, quien publicó la primera versión (0.01). Desde entonces, ha evolucionado significativamente, incorporando nuevas características, mejoras en la seguridad y compatibilidad con una variedad de hardware. En 1992, se lanzó la versión 0.95, y el kernel ha continuado creciendo con contribuciones de una amplia comunidad de desarrolladores, llevando a lanzamientos estables y versiones a largo plazo.

(b) ¿Cuáles son sus funciones principales?

Las funcionalidades principales del kernel de GNU/Linux son:

1. Gestión de procesos: Controla la creación, ejecución y finalización de procesos, gestionando la multitarea.
2. Gestión de memoria: Administra la memoria RAM, asignando y liberando espacios para los procesos.

3. Gestión de dispositivos: Facilita la comunicación entre el hardware y el software, manejando drivers para diversos dispositivos.
4. Sistema de archivos: Proporciona una interfaz para almacenar, recuperar y gestionar datos en dispositivos de almacenamiento.
5. Seguridad y control de acceso: Implementa políticas de seguridad para proteger los recursos del sistema.

(c) ¿Cuál es la versión actual? ¿Cómo se definía el esquema de versionado del Kernel en versiones anteriores a la 2.4? ¿Qué cambió en el versionado se impuso a partir de la versión 2.6?

La versión actual del Kernel de Linux es la 6.11, lanzada el 2 de octubre de 2023. En cuanto al esquema de versionado anterior a la versión 2.4, se utilizaba un formato que consistía en tres números: **X.Y.Z**. Aquí, **X** representaba cambios importantes, **Y** cambios menores y **Z** correcciones de errores. Con la llegada de la versión 2.6, se adoptó un nuevo esquema de versionado más ágil, separando la versión principal de las versiones de desarrollo y usando un ciclo de lanzamientos más frecuente, que incluía versiones estables y versiones de prueba.

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

Sí, es posible tener más de un kernel de GNU/Linux instalado en la misma máquina. Esto permite al usuario elegir cuál kernel utilizar al iniciar el sistema. La mayoría de las distribuciones gestionan esto a través de un gestor de arranque como GRUB, que permite seleccionar entre las diferentes versiones del kernel disponibles. Esta práctica es útil para probar nuevas versiones o para solucionar problemas con un kernel específico.

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

En el sistema de archivos de GNU/Linux, los kernels instalados suelen ubicarse en el directorio `/boot`. Allí, encontrarás archivos de imagen del kernel (por ejemplo, `vmlinuz-<versión>`), así como otros archivos relacionados, como `initramfs` y configuraciones del kernel. Este directorio es esencial para el arranque del sistema, y el gestor de arranque (como GRUB) utiliza esta ubicación para presentar las opciones de kernel durante el inicio.

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

El kernel de GNU/Linux es considerado híbrido. Aunque tiene características de un kernel monolítico al integrar varios servicios en un solo espacio de direcciones, también permite la carga de módulos. Esto significa que se pueden añadir o quitar funcionalidades en tiempo de ejecución sin necesidad de reiniciar el sistema. Esta flexibilidad permite optimizar el rendimiento y la estabilidad, ya que los módulos pueden ser actualizados de forma independiente. En resumen, combina elementos monolíticos y de microkernel.

5. Intérprete de comandos (Shell):

(a) ¿Qué es?

Shell: Es un intérprete de comandos que actúa como interfaz entre el usuario y el sistema operativo. Permite a los usuarios ejecutar comandos, scripts y programas, facilitando la administración del sistema. La shell también permite automatizar tareas a través de scripts y ofrece funcionalidades como la manipulación de archivos, la gestión de procesos y la redirección de

entradas y salidas. Existen diferentes tipos de shells, como Bash, Zsh y Fish, cada una con sus propias características y funcionalidades.

(b) ¿Cuáles son sus funciones?

Interacción con el usuario: Permite a los usuarios ejecutar comandos y programas.

Ejecución de scripts: Facilita la automatización de tareas mediante scripts.

Gestión de procesos: Controla la ejecución de programas y permite la manipulación de procesos.

Manipulación de archivos: Proporciona comandos para crear, eliminar, copiar y mover archivos y directorios.

Redirección y tuberías: Permite enviar la salida de un comando como entrada a otro, así como redirigir la salida a archivos.

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

Bash (Bourne Again Shell): Es el intérprete más común y predeterminado en muchas distribuciones. Ofrece potentes características de scripting y una amplia compatibilidad con otros shells.

Zsh (Z Shell): Ofrece características avanzadas, como autocompletado mejorado, globbing avanzado y personalización. Es popular entre los desarrolladores y usuarios avanzados.

Fish (Friendly Interactive Shell): Se centra en la facilidad de uso, con autocompletado inteligente y resaltado de sintaxis. Es menos compatible con scripts de otros shells, pero más amigable para principiantes.

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

Los comandos propios del shell, conocidos como built-ins, se encuentran integrados dentro del propio intérprete y no tienen un path específico. Por otro lado, los comandos externos se encuentran en directorios del sistema como:

- **/bin:** Comandos esenciales del sistema.
- **/usr/bin:** Comandos de usuario más comunes.
- **/sbin:** Comandos de administración del sistema.
- **/usr/sbin:** Comandos de administración adicionales.

(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 porque desempeña un papel diferente. El kernel se encarga de la gestión de hardware, procesos y memoria, mientras que el shell actúa como interfaz entre el usuario y el sistema operativo, interpretando comandos y ejecutando programas. Esta separación permite que el shell ofrezca funcionalidades más accesibles y flexibles, facilitando la interacción del usuario con el sistema sin afectar directamente al kernel.

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

Sí, es posible definir un intérprete de comandos distinto para cada usuario en GNU/Linux. Esto se puede hacer configurando el archivo de perfil del usuario, como `.bashrc`, `.bash_profile`, o `.profile`, dependiendo del shell utilizado.

Cualquier usuario puede realizar esta tarea, pero los cambios solo afectarán al usuario que edite su archivo de configuración. Los administradores del sistema también pueden establecer un intérprete de comandos por defecto para todos los usuarios en el archivo `/etc/passwd`.

6. Sistema de Archivos (File System):

(a) ¿Qué es?

Un **sistema de archivos** (File System) es una estructura que organiza y gestiona el almacenamiento de datos en un dispositivo, permitiendo que el sistema operativo y los usuarios guarden, recuperen y manipulen archivos y directorios.

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

ext4: Uno de los más utilizados, conocido por su estabilidad y rendimiento.

Btrfs: Ofrece características avanzadas como snapshots y gestión de volúmenes.

XFS: Optimizado para manejar archivos grandes y carga pesada.

FAT32: Comúnmente utilizado en dispositivos USB y compatibilidad con otros sistemas.

NTFS: Usado por sistemas Windows, permite la lectura y escritura en GNU/Linux.

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

Sí, es posible visualizar particiones del tipo FAT y NTFS en GNU/Linux. La mayoría de las distribuciones de GNU/Linux incluyen soporte para estos sistemas de archivos. Para acceder a ellos, es posible montar las particiones utilizando los comandos `mount` o herramientas gráficas. Además, existen utilidades como `ntfs-3g` que permiten la lectura y escritura en particiones NTFS.

(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?

`/`: Directorio raíz que contiene todo el sistema de archivos.

`/bin`: Comandos esenciales del sistema.

`/etc`: Archivos de configuración del sistema.

`/home`: Directorios personales de los usuarios.

`/var`: Archivos de datos variables, como registros y bases de datos.

`/usr`: Contiene aplicaciones y archivos de usuario.

7. Particiones:

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

Una partición es una división lógica en un disco duro que permite organizar y gestionar datos de manera más eficiente, facilitando la instalación de múltiples sistemas operativos o la separación de datos.

Tipos de particiones:

1. Primarias: Pueden contener sistemas operativos; hay un límite de cuatro.
2. Extendidas: Actúan como contenedoras para particiones lógicas; permiten más de cuatro particiones.

3. Lógicas: Se encuentran dentro de particiones extendidas; pueden contener datos o sistemas operativos.

Ventajas y Desventajas:

Ventajas:

- Mejora la organización de datos.
- Facilita la instalación de varios sistemas operativos.
- Permite recuperar datos de particiones separadas en caso de fallo.

Desventajas:

- Puede complicar la gestión del disco.
- Requiere cuidado en la redimensión y eliminación de particiones para evitar la pérdida de datos.

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

En GNU/Linux, las particiones se identifican a través de convenciones de nombres que dependen del tipo de disco:

- IDE: Se identifican como /dev/hda, /dev/hdb, etc., donde la letra representa el disco y el número indica la partición (por ejemplo, /dev/hda1 para la primera partición del primer disco).
- SCSI: Se identifican como /dev/sda, /dev/sdb, etc., siguiendo la misma lógica (por ejemplo, /dev/sda1).
- SATA: Se identifican igual que los discos SCSI, usando la misma nomenclatura.

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

Para instalar GNU/Linux, como mínimo se necesitan dos particiones:

1. Partición raíz:
 - Tipo de partición: Primaria
 - Identificación: /dev/sda1
 - Tipo de File System: ext4
 - Punto de montaje: /
2. Partición de intercambio (swap):
 - Tipo de partición: Primaria o lógica
 - Identificación: /dev/sda2
 - Tipo de File System: swap
 - Punto de montaje: No tiene, se utiliza como memoria virtual.

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

Servidor Web:

- Partición raíz / (ext4).
- Partición /var (ext4) para registros.
- Partición /home (ext4) para archivos de usuario.
- Partición swap.

Estación de trabajo:

- Partición raíz / (ext4).
- Partición /home (ext4) para documentos.
- Partición /data (ext4) para archivos compartidos.
- Partición swap.

Servidor de bases de datos:

- Partición raíz / (ext4).
- Partición /var/lib/mysql (ext4) para datos de MySQL.
- Partición swap.

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

GParted: Interfaz gráfica fácil de usar. Permite crear, redimensionar y mover particiones. Soporta varios sistemas de archivos.

fdisk: Herramienta de línea de comandos para gestionar particiones en sistemas MBR. Menos intuitiva, pero poderosa para usuarios avanzados.

parted: Similar a fdisk, pero con soporte para GPT y una interfaz de línea de comandos más avanzada.

cfdisk: Interfaz de usuario más simple en línea de comandos, adecuada para usuarios que prefieren un enfoque menos técnico.

Comparación:

- Facilidad de uso: GParted > cfdisk > parted > fdisk
- Funciones avanzadas: parted > fdisk > GParted
- Interfaz: GParted (gráfica) > cfdisk > fdisk (línea de comandos)

8. Arranque (bootstrap) de un Sistema Operativo:

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

El BIOS (Basic Input/Output System) es un firmware almacenado en una memoria no volátil que se ejecuta al encender el ordenador. Su principal tarea es realizar la autocomprobación de encendido (POST), que verifica el hardware del sistema, y luego cargar el sistema operativo desde el dispositivo de arranque configurado. Además, el BIOS permite la configuración de parámetros de hardware y gestión de dispositivos, actuando como un puente entre el sistema operativo y el hardware.

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

UEFI (Unified Extensible Firmware Interface) es una especificación de firmware que reemplaza al BIOS. Su función principal es iniciar el hardware del sistema y cargar el sistema operativo, proporcionando un entorno más flexible y moderno. UEFI admite discos duros de más de 2 TB, arranque rápido, interfaces gráficas, y una mayor seguridad a través de características como el arranque seguro (Secure Boot), que ayuda a prevenir la ejecución de software malicioso durante el proceso de arranque.

(c) ¿Qué es el MBR? ¿Qué es el MBC?

MBR (Master Boot Record) es un esquema de particionamiento que se encuentra en el primer sector de un disco duro. Contiene información sobre las particiones del disco y el código de arranque que inicia el sistema operativo.

MBC (Master Boot Code) es el código de arranque que se encuentra dentro del MBR. Es responsable de cargar el gestor de arranque del sistema operativo desde la partición activa.

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

GPT significa GUID Partition Table. Sustituye al MBR (Master Boot Record) como esquema de particionamiento en discos duros. El formato de GPT permite un número mucho mayor de particiones (hasta 128 por defecto) y admite discos de más de 2 TB, además de proporcionar redundancia y mejoras en la integridad de los datos. GPT almacena múltiples copias de la tabla de particiones, lo que la hace más robusta frente a fallos.

(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 (bootloader) es un programa que carga el sistema operativo en la memoria cuando se inicia el ordenador. Su funcionalidad principal es localizar y cargar el núcleo del sistema operativo.

Tipos de gestores de arranque:

1. GRUB (Grand Unified Bootloader): Común en sistemas Linux, admite múltiples sistemas operativos.
2. LILO (Linux Loader): Un gestor de arranque más antiguo y menos flexible que GRUB.
3. Syslinux: Utilizado principalmente para arranques desde medios extraíbles.

Instalación:

Se instalan en el MBR o en la partición EFI en sistemas UEFI.

(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. El proceso de bootstrap en una computadora incluye los siguientes pasos:
2. Encendido: La computadora recibe energía y comienza su inicialización.
3. POST (Power-On Self-Test): El BIOS/UEFI verifica el hardware.

4. Carga del gestor de arranque: Se localiza el gestor de arranque en el MBR o en la partición EFI.
5. Selección del sistema operativo: El gestor de arranque permite elegir entre múltiples sistemas operativos si están disponibles.
6. Cargar el núcleo del sistema operativo: Se carga el núcleo (kernel) en la memoria.
7. Ejecutar procesos del sistema operativo: Se inician procesos y servicios del sistema operativo.

Finalmente, el sistema operativo está listo para interactuar con el usuario.

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

El proceso de arranque en GNU/Linux incluye los siguientes pasos:

1. POST: El BIOS/UEFI realiza una autocomprobación del hardware.
2. Gestor de arranque: Carga el gestor de arranque, como GRUB, que se encuentra en el MBR o en la partición EFI.
3. Selección del kernel: El gestor de arranque presenta opciones de sistemas operativos y selecciona el kernel de Linux.
4. Carga del kernel: Se carga el kernel en la memoria y se inician los módulos necesarios.
5. Init: Se ejecuta el proceso de inicialización del sistema (), que inicia los servicios y procesos del sistema.

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

Pasos en el proceso de parada (shutdown) de GNU/Linux:

1. Notificación a los procesos: Se envía una señal a todos los procesos para que se detengan adecuadamente.
2. Detención de servicios: Los servicios y demonios en ejecución son detenidos.
3. Desmontaje de sistemas de archivos: Se desmontan los sistemas de archivos para asegurar que no haya pérdida de datos.
4. Apagar el hardware: Finalmente, se envía una señal para apagar el hardware.

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

Sí, es posible tener GNU/Linux y otro sistema operativo en la misma PC. Esto se realiza mediante particionamiento del disco y la instalación de un gestor de arranque (como GRUB) que permite seleccionar el sistema operativo al inicio.

9. Archivos:

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

¿Qué idea le da de ls? Ls muestra una lista de los archivos y directorios en el directorio actual

El comando **stat** proporciona información detallada sobre un archivo, como el tamaño, los permisos, la fecha de creación y modificación, etc.

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

El editor **vi** (Visual Editor) es un editor de texto tradicional y muy potente en los sistemas Unix y GNU/Linux. Funciona en la terminal y está disponible prácticamente en todas las distribuciones. Tiene dos modos principales: **modo de comando** y **modo de inserción**

El comando **cat** (concatenate) es uno de los comandos más básicos y utilizados para mostrar el contenido de archivos en la terminal. También se puede usar para concatenar varios archivos.

cat archivo.txt — muestra el archivo

cat archivo.txt archivo2.txt — concatena los archivos y los muestra en consola

cat archivo1.txt archivo2.txt > nuevo_archivo.txt — concatena archivos y los guarda en otro

cat archivo1.txt >> archivo2.txt — agrega lo de un archivo a otro

El comando **more** permite mostrar el contenido de un archivo **página por página**, útil para archivos grandes donde no es práctico usar **cat** porque podría desplazarse rápidamente por la terminal.

more archivo.txt

mcedit: Editor simple de Midnight Commander, con una interfaz más amigable para principiantes.

vi: Editor de texto en terminal, muy potente y configurable, pero con una curva de aprendizaje pronunciada.

(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. ✓

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

El comando **file** dice que tipo de archivo es cada archivo. En lugar de basarse únicamente en la extensión del archivo (como lo hace Windows), el comando **file** inspecciona el **contenido interno del archivo** para determinar qué tipo de archivo es, utilizando una base de datos de firmas conocidas (magic numbers) y otras heurísticas.

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 ✓

Con el **mkdir** **mkdir carpetita**

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

cd carpetita y entras a la carpeta

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

touch iso20

touch iso21

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

ls — muestra el contenido

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

pwd — dice en que ruta estas

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

find iso — dice cuales archivos hay con ese nombre

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

df

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

who — quienes están en el sistema

**(i) Acceder a el archivo iso2017-1 e ingresar Nombre y Apellido
vi iso20**

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

tail iso.txt

11. Investigue su funcionamiento y parámetros más importantes:

(a) shutdown /sbin o /usr/sbin

apaga el sistema de manera segura

(b) reboot /sbin o /usr/sbin

reinicia el sistema.

(c) halt /sbin o /usr/sbin

El comando **halt** detiene todos los procesos del sistema y apaga el hardware, lo que en muchos casos resulta en un apagado inmediato.

(d) locate /usr/bin

El comando **locate** busca archivos en el sistema utilizando una base de datos previamente construida que contiene información sobre los archivos y directorios.

(e) uname /bin

El comando **uname** muestra información sobre el sistema operativo

(f) dmesg /bin

El comando **dmesg** muestra los mensajes del búfer del núcleo (kernel) del sistema, lo cual puede ser útil para la depuración y ver el registro de eventos del núcleo.

(g) lspci /usr/bin

El comando **lspci** lista todos los dispositivos PCI conectados al sistema.

(h) at /usr/bin

El comando **at** se utiliza para programar la ejecución de comandos en el futuro.

(i) netstat /bin o /usr/bin

El comando **netstat** muestra información sobre las conexiones de red, las tablas de enrutamiento, y las interfaces de red.

netstat -a conexiones de red y puertos

netstat -i info de interfaces de red

(j) mount /bin

El comando **mount** se utiliza para montar sistemas de archivos, ya sea de discos, particiones, o unidades removibles.

Montar un sistema de archivos:

sudo mount /dev/sdX1 /mnt/punto_de_montaje

Mostrar los sistemas de archivos montados:

mount

(k) umount /bin

El comando **umount** se utiliza para desmontar sistemas de archivos que han sido montados previamente con el comando **mount**.

Desmontar un sistema de archivos:

sudo umount /mnt/punto_de_montaje

Desmontar un dispositivo específico:

sudo umount /dev/sdX1

(l) head /usr/bin

El comando **head** muestra las primeras líneas de un archivo. Por defecto, muestra las primeras 10 líneas.

head archivo.txt muestra las primeras 10 líneas

Muestra las primeras n líneas

head -n 20 archivo.txt

Mostrar las primeras N bytes de un archivo:

head -c 100 archivo.txt

(m) losetup /bin

El comando **losetup** se utiliza para configurar y controlar dispositivos de bucle (loop devices), que permiten montar imágenes de archivos como si fueran discos físicos.

Asociar un archivo de imagen con un dispositivo de bucle: **sudo losetup /dev/loop0 /ruta/al/archivo.img**

Mostrar los dispositivos de bucle actuales:

losetup -a

Desasociar un archivo de imagen de un dispositivo de bucle:

sudo losetup -d /dev/loop0

(n) write /usr/bin

El comando **write** se utiliza para enviar mensajes a otros usuarios que están conectados al mismo sistema.

Enviar un mensaje a un usuario:

write nombre_usuario

Luego escribe el mensaje y presiona Ctrl+D para enviarlo.

Enviar un mensaje a una terminal específica:

write nombre_usuario pts/N

Donde pts/N es el número de la terminal.

(ñ) mkfs /sbin o /usr/sbin

El comando **mkfs** se utiliza para crear un sistema de archivos en una partición o dispositivo.

Crear un sistema de archivos ext4:

sudo mkfs.ext4 /dev/sdX1

Crear un sistema de archivos FAT32:

sudo mkfs.vfat /dev/sdX1

Crear un sistema de archivos con mkfs:

sudo mkfs -t tipo /dev/sdX1

Donde tipo puede ser ext4, vfat, ntfs, etc., y /dev/sdX1 es la partición.

(o) fdisk (con cuidado) /sbin

El comando **fdisk** se utiliza para gestionar particiones de discos. ¡Ten cuidado al usarlo, ya que puede modificar y borrar particiones, lo cual puede resultar en la pérdida de datos.

Listar las particiones en un disco:

```
sudo fdisk -l
```

Iniciar fdisk para un disco específico:

```
sudo fdisk /dev/sdX
```

Esto abrirá una interfaz interactiva donde puedes crear, eliminar, y modificar particiones. Asegúrate de conocer bien los comandos dentro de la interfaz de fdisk para evitar errores.

p Mostrar la tabla de particiones.

n Crear una nueva partición.

d Eliminar una partición

w escribir los cambios y salir

q Salir sin guardar los cambios.

12. Investigue su funcionamiento y parámetros más importantes:

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