

1. Características de GNU/Linux:

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

- Es un Sistema Operativo tipo Unix (Unix like), pero libre
- S.O. diseñado por miles de programadores
- S.O. gratuito y de libre distribución (se baja desde la Web, CD, etc.)
- Existen diversas distribuciones (customizaciones)
- Es código abierto, lo que nos permite estudiarlo, personalizarlo, auditar, aprovecharnos de la documentación, etc...

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

Windows: Posee una interfaz más amigable para los usuarios y es mucho más popular que otros S.O, sin embargo este necesita de una licencia para poder usar todas sus características, a diferencia de GNU/Linux que es generalmente gratuito y de código abierto

(c) ¿Qué es GNU?

Un proyecto iniciado por Richard Stallman, que reunía una serie de softwares de libre uso. Fue financiado por la FSF y tiempo después se fusionó con el Kernel Linux.

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

En 1985, Richard Stallman fundó la Free Software Foundation (FSF), una organización sin ánimo de lucro para apoyar y coordinar el desarrollo de software libre. Entre sus mayores contribuciones de software libre se encuentran: editor de texto, compilador y depurador para desarrolladores.

Intentaron hacer su propio Kernel (Hurd), sin embargo este se descartó por su complejidad (1990). En 1991 colaboran con el kernel Linux desarrollado por Linus Torvalds, lo que permite la creación de un S.O alternativo libre y funcional a Unix.

En años posteriores se expande y consolida, contando con una gran comunidad que mantiene vivo el movimiento iniciado por Richard Stallman.

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

Si, GNU/Linux es multitarea. Esto significa que es capaz de gestionar los recursos del sistema para ejecutar múltiples programas y/o procesos al mismo tiempo.

(f) ¿Qué es POSIX?

Es un conjunto de estándares definidos por la IEEE que especifica interfaces de programación de aplicaciones (APIs), comandos de línea de comandos y utilidades, así como la gestión de archivos y procesos, con el objetivo de garantizar la portabilidad y compatibilidad entre diferentes sistemas operativos.

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.

Ubuntu (facilitar el uso para principiantes), Fedora (se centra en las últimas innovaciones tecnológicas), Debian (ofrecer estabilidad, es independiente) y Arch Linux (permite a los usuarios crear su S.O desde cero, personalizándolo).

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

Principalmente en que basan sus objetivos, la gestión de paquetes que tengan, el entorno de escritorio, las actualizaciones y su enfoque, el cual varía dependiendo del objetivo que tenga el usuario de estos.

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

Siempre fue proporcionar un sistema operativo estable y libre con una fuerte filosofía de software libre.

3. Estructura de GNU/Linux:

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

Kernel: Es un programa de bajo nivel que se ejecuta en modo privilegiado y tiene acceso directo al hardware de la máquina.

Herramientas GNU: Proporciona un conjunto de herramientas y utilidades desarrolladas por el proyecto GNU.

Gestión de paquetes: son fundamentales para instalar, actualizar, y eliminar software en un entorno GNU/Linux. Administran las dependencias y versiones del software, asegurando la coherencia y estabilidad del sistema.

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

Hardware, Kernel, Interfaz de llamadas al sistema, Bibliotecas del sistema, Shell (intérprete de comandos), Herramientas y utilidades del sistema, Entorno y aplicaciones de usuario, Sistema de gestión de paquetes.

4. Kernel:

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

Es “el corazón del S.O”, es intermediario entre el hardware y el sistema operativo.

(b) ¿Cuáles son sus funciones principales?

Gestión de procesos, gestión de memoria, sistema de archivos, controlador de dispositivos, gestión de redes, control y seguridad de acceso.

(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?

El salto de Linux 2.4 a 2.6 representó un avance importante en términos de rendimiento, escalabilidad, soporte de hardware y características de seguridad. Además aumenta el soporte para múltiples hilos, soporte para múltiples nodos.

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

Si, es posible.

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

Se encuentra en el directorio `/boot`

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

Sí, el kernel de GNU/Linux es considerado un kernel monolítico, esto se debe a que todas las funciones principales del sistema operativo se ejecutan en un solo espacio de memoria.

5. Intérprete de comandos (Shell):

(a) ¿Qué es?

Es el modo en el que el usuario y el sistema operativo se comunican.

(b) ¿Cuáles son sus funciones?

Ejecutar programas a partir del ingreso de comandos y organizar la forma en la que se almacenan los archivos en los dispositivos de almacenamiento.

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

Bourne Shell (sh), Korn Shell (ksh), Bourne Again Shell (bash)(autocompletado, history, alias)

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

Los comandos propios del Shell no están ubicados en el sistema de archivos, ya que se encuentran dentro del propio código de la Shell.

Los comandos externos se encuentran en el directorio `/bin`

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

No es parte del Kernel, debido a su diferencia de objetivos. El kernel es el núcleo del sistema operativo, y se encarga de gestionar todo, incluido el hardware, en cambio el Shell es solo un medio por el cual el usuario puede comunicarse con el sistema operativo, y por ende con el hardware, pero de forma indirecta.

(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. Esto se define en el archivo `/etc/passwd`, este contiene la información sobre las cuentas de usuario en el sistema y es este archivo el que se modifica el intérprete de comandos para cada usuario. (solo se puede siendo root)

6. Sistema de Archivos (File System):

(a) ¿Qué es?

Organiza la forma en que se almacenan los archivos en dispositivos de almacenamiento.

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

fat, ntfs ext2, ext3, reiser.

Hace un tiempo se está debatiendo el reemplazo de ext por Btrfs (B-tree FS) de Oracle.

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

Sí, es posible.

FAT es un sistema de archivos antiguo, pero ampliamente compatible, es usado por memorias USB y tarjetas SD.

NTFS es el sistema de archivos usado por Windows, GNU/Linux puede leer y escribir en este tipo de particiones gracias a la herramienta `ntfs-3g`.

(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?

FHS: hace referencia a la Filesystem Hierarchy Standard, que describe las convenciones para sistemas operativos de tipo Unix.

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

7. Particiones:

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

Es una forma de dividir lógicamente el disco físico, de manera que cada sistema operativo es almacenado únicamente en una partición separada.

Partición primaria: son las particiones primarias que puede tener un disco. Tiene un límite de cuatro particiones primarias.

Ventajas:

- Simples y directas para el arranque del sistema operativo.
- Generalmente necesarias para las particiones de sistema, ya que algunas versiones de BIOS solo pueden arrancar desde una partición primaria.

Desventajas:

- Limitación de solo cuatro particiones primarias en el esquema de partición MBR (Master Boot Record).
- Si necesitas más particiones, es necesario convertir una en partición extendida o cambiar a un esquema más moderno (GPT).

Partición extendida: esta es una partición especial que no contiene datos directamente, sino que actúa como un contenedor para particiones lógicas.

Ventajas:

- Permite superar la limitación de cuatro particiones primarias, ya que se pueden crear muchas particiones lógicas dentro de una partición extendida.
- Flexibilidad para crear múltiples particiones lógicas en un solo disco.

Desventajas:

- Solo puede haber una partición extendida por disco.
- No es posible arrancar directamente desde una partición extendida (aunque esto ha cambiado con esquemas más modernos como GPT y UEFI).

Partición lógica: ocupa la totalidad o parte de la partición extendida y se le define un tipo de FS (Filesystem). Las particiones de este tipo se conectan como una lista enlazada

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

Las particiones se identifican mediante un sistema de nombres que depende del tipo de disco (IDE, SCSI, SATA) y el orden de las particiones en cada disco.

Los discos duros IDE o **Parallel ATA (PATA)** usan el prefijo **hd** para su identificación, seguido de una letra que representa el disco y un número que representa la partición.

Los discos **SCSI**, **SATA**, y **USB** utilizan el prefijo **sd** para identificarse. Este prefijo es más común en los sistemas GNU/Linux actuales debido a la prevalencia de discos SATA y dispositivos USB.

(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.

Como mínimo debe haber una partición (para el /), pero es recomendable tener dos (para el / y SWAP)

1. Partición raíz (/)

- Tipo de partición: Primaria (o lógica si estás en una partición extendida).
- Identificación: Generalmente algo como **/dev/sda1** o **/dev/sdb1**, dependiendo del disco.
- Tipo de File System: Normalmente ext4, aunque también se pueden usar otros como XFS, Btrfs, o ext3.
- Punto de montaje: **/** (directorio raíz).
- Descripción: Es la partición principal donde se instalarán el sistema operativo y todas las aplicaciones. Contiene todo el árbol de directorios, incluyendo **/bin**, **/etc**, **/lib**, **/usr**, y **/var**.

2. Partición de intercambio (Swap)

- Tipo de partición: Primaria o lógica.
- Identificación: Algo como **/dev/sda2** o **/dev/sdb2**, dependiendo del disco.
- Tipo de File System: No tiene un sistema de archivos estándar porque está dedicada a la memoria virtual.
- Punto de montaje: No tiene punto de montaje, ya que se utiliza directamente por el sistema para gestionar la memoria virtual.
- Descripción: La partición de intercambio o swap actúa como memoria virtual y ayuda a gestionar situaciones en las que la memoria RAM se llena. También es utilizada para hibernar el sistema. Si no se tiene suficiente RAM o se desea habilitar la hibernación, esta partición es crucial.

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

1. Particionamiento básico para escritorio.
2. Particionamiento para servidor web o base de datos.
3. Particionamiento para estaciones de trabajo de desarrollo
4. Particionamiento para Sistemas de Cifrado (Seguridad)
5. Particionamiento para Dual Boot (GNU/Linux y Windows)

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

Destructivos: permiten crear y eliminar particiones (fdisk).

No destructivo: permite crear, eliminar y modificar particiones (fips, gparted) ← generalmente las distribuciones permiten hacerlo desde la interfaz de instalación.

8. Arranque (bootstrap) de un Sistema Operativo:

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

Es un proceso de pequeños programas que se graba en un chip, y este se encarga del inicio de una máquina y carga del sistema operativo, y se denomina bootstrap.

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

Define la ubicación del gestor de arranque, la interfaz entre el gestor de arranque y el firmware, y expone información para los gestores de arranque.

(c) ¿Qué es el MBR? ¿Qué es el MBC?

El MBR (Master Boot Record) es el sector de arranque de los discos duros o dispositivos de almacenamiento que usan un esquema de particionamiento tradicional.

El MBC (Master Boot Code) es parte del MBR y corresponde específicamente al código de arranque maestro que se almacena en los primeros bytes del sector de arranque (MBR). Su principal función es leer la tabla de particiones y cargar el sector de arranque de la partición activa.

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

Las siglas GPT hacen referencia a GUID Partition Table (Tabla de Particiones GUID). Es un esquema de particionamiento moderno que viene incluido con UEFI, y reemplaza al esquema de particiones tradicional basado en el MBR.

Características principales de GPT:

- GPT soporta discos de hasta 9.4 zettabytes (ZB), mucho más que los 2 TB que permite el MBR.
- Esto lo hace ideal para discos modernos de gran capacidad.
- GPT permite hasta 128 particiones primarias
- Cada partición en GPT está identificada por un GUID (Identificador Global Único), lo que significa que cada partición tiene un identificador único a nivel mundial. Esto mejora la seguridad y la consistencia en la gestión de las particiones.

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

Cargar el sistema operativo en la memoria y transferir el control del sistema al mismo.

Existen dos tipos:

Bootloader de MBR: En el esquema MBR, el primer sector del disco (el MBR) contiene un pequeño cargador de arranque

Gestores de arranque de segundo nivel: Estos gestores de arranque se cargan después del primer nivel y manejan el proceso de carga del sistema operativo desde la partición activa. Ejemplos son GRUB y LILO. Se instalan en el disco duro, en el SSD o en una partición de arranque separada.

- (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 sistema: llega la energía a la CPU y la placa madre.
 2. La BIOS ejecuta el POST.
 3. Búsqueda del dispositivo de arranque: el firmware BIOS (MBR) o UEFI (EFI/ESP) busca el dispositivo de arranque que contiene el gestor de arranque.
 4. Carga el gestor de arranque: carga el dispositivo de arranque en memoria.
 5. Selección del sistema operativo: Si hay múltiples sistemas operativos instalados, el gestor de arranque (como GRUB, LILO, o Windows Boot Manager) puede mostrar un menú para que el usuario seleccione cuál arrancar.
 6. Carga del kernel del SO: el gestor de arranque carga el núcleo (kernel) del sistema operativo en la memoria RAM
- (g) Analice el proceso de arranque en GNU/Linux y describa sus principales pasos.
- (h) ¿Cuáles son los pasos que se suceden en el proceso de parada (shutdown) de GNU/Linux?
- (i) ¿Es posible tener en una PC GNU/Linux y otro Sistema Operativo instalado? Justifique.

Si.

9. Archivos:

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

En Linux, los archivos pueden ser de diferentes tipos, y estos se identifican mediante un **carácter especial** al comienzo de la lista de permisos cuando se ejecuta el comando `ls -l`:

- `-`: Archivo regular (por ejemplo, un archivo de texto o binario).
- `d`: Directorio.
- `l`: Enlace simbólico (un puntero a otro archivo).
- `b`: Archivo de dispositivo de bloque (dispositivo físico como un disco duro).
- `c`: Archivo de dispositivo de caracteres (como un terminal o impresora).
- `p`: Tubería con nombre (para la comunicación entre procesos).
- `s`: Socket (para la comunicación entre procesos).

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

Comandos básicos de vi:

- Para ingresar al editor: `vi nombre_archivo`
- Para salir del editor: `:q`
- Para cambiar al modo de inserción:
 - `i`: Insertar texto antes del cursor.
 - `a`: Insertar texto después del cursor.
 - `o`: Crear una nueva línea debajo del cursor.
- Una vez en el modo inserción, cada vez que quiera cambiar a otro modo, o borrar el carácter sobre el que estoy escribiendo con la **X**, presionar `:(comando)`
- Para guardar y salir:
 - `:w` para guardar.
 - `:q` para salir.

- `:wq` para guardar y salir.
- `:q!` para salir sin guardar.
- **Para moverse en el texto y editar:**
 - `h`: Moverse a la izquierda.
 - `j`: Moverse hacia abajo.
 - `k`: Moverse hacia arriba.
 - `l`: Moverse a la derecha.
 - `/texto`: Buscar un término en el archivo.
 - `x`: borrar el carácter.
 - `dd`: borrar la línea entera.
 - `3dd`: borrar varias líneas (el número puede cambiar).

`vi` es muy poderoso pero tiene una curva de aprendizaje alta debido a su naturaleza modal.

Comandos básicos de `mcedit`:

- **Para abrir un archivo:** `mcedit nombre_archivo`
- **Para guardar:** Usa `F2`.
- **Para salir:** Usa `F10`.
- **Navegación:** Puedes navegar usando las flechas del teclado, similar a otros editores de texto.

Es ideal para usuarios que prefieren un editor más sencillo y gráfico en entornos de consola.

Funciones principales de `cat`:

- **Mostrar el contenido de un archivo:**
 - `cat archivo.txt`: Muestra el contenido de `archivo.txt` en la terminal.
- **Concatenar archivos:**
 - `cat archivo1.txt archivo2.txt > archivo3.txt`: Combina los contenidos de `archivo1.txt` y `archivo2.txt` en `archivo3.txt`.
- **Crear archivos de texto:**
 - `cat > archivo.txt`: Te permite escribir directamente en `archivo.txt`. Para terminar, presiona `Ctrl + D`.

Funciones principales de `more`:

- **Mostrar el contenido de un archivo paginándolo:**
 - `more archivo.txt`: Muestra el contenido del archivo `archivo.txt` una página a la vez.

Navegación en `more`:

- **Espacio:** Avanza una página completa.
- **Enter:** Avanza una línea.
- `/texto`: Busca un texto dentro del archivo.
- `b`: Retrocede una página.
- `q`: Salir del comando.

(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.

Hecho.

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

El comando **file** (archivo) proporciona una descripción sobre el archivo:

- Tipo de archivo que es
- Si es un texto puede determinar su formato. (ASCII o UTF-8, etc)
- Dependiendo del tipo de archivo, puede ofrecer detalles adicionales como la codificación, arquitectura del sistema (si es un ejecutable), el tipo de imagen (si es una imagen), entre otros.