

Práctica 1

Características de GNU/Linux

a) GNU/Linux es un SO Unix like, gratuito y libre diseñado por miles de programadores, además, posee diversas distribuciones y es open source. Al ser software libre tiene las siguientes características:

- Puede ser usado, copiado, estudiado, redistribuido, etc. libremente.
- Generalmente es de costo nulo aunque puede no ser gratis.
- Comúnmente distribuido con su código fuente.
- Tiene una corrección más rápida ante fallas.
- Garantiza las 4 libertades de GNU:
 - Libertad de usar el software con cualquier propósito.
 - Libertad de estudiar su funcionamiento.
 - Libertad de distribuir sus copias.
 - Libertad de mejorar el software.

También posee ciertas características generales:

- Es multiusuario, multitarea y multiprocesador.
- Su código es altamente portable.
- Posee varios intérpretes de comandos, algunos programables.

- Permite manejo de usuarios y permisos.
- Todo es un archivo.
- Cada directorio puede estar en particiones distintas.
- Es case sensitive y open source.

b) Otros sistemas operativos como Microsoft Windows, macOS, iOS difieren con GNU/Linux en las siguientes características:

- Generalmente tienen un costo asociado.
- Prohibida la libre distribución.
- Prohibida la modificación.
- No son open source o, al menos, generalmente.
- Necesitan menos técnicos especializados.
- La corrección de fallas queda a cargo del propietario.

Esto es así ya que los SO mencionados son Software Propietario.

c) GNU es un proyecto y un movimiento de software libre iniciado por Richard Stallman con el objetivo de crear un sistema operativo completo y libre, Unix like pero sin tener las restricciones del software propietario.

Unix es un sistema operativo multiusuario, originalmente destinado a minicomputadores, pero implementado en una gran variedad de computadoras, además, es multiusuario, no libre y muy popular en la década de los '60 / '70.

d) Historia detrás de GNU / Linux:

- Se inicia la creación del Sistema GNU en 1983 a manos de Richard Stallman con el fin de crear un Unix Libre. Para

esto se necesitó un marco regulatorio que permita la garantía de esta libertad, esto se cumple con la GPL (General Public License de GNU). La financiación del proyecto lo logró Stallman en 1985 creando la FSF (Free Software Foundation). Para 1990, GNU ya tenía editor de textos (Emacs), un compilador (GCC) y varias bibliotecas, solo necesitaba un Kernel. Se intentó con TRIX (en 1988 se abandona por complejidad) y con Mach creando GNU Hurd pero ninguno de estos prosperó. No fue entonces hasta que en 1991 Stallman fusiona su proyecto con Linus Torvalds quien venía desarrollando un Kernel llamado Linux de 1991, naciendo por fin GNU/Linux a partir del software libre.

El Multitarea: Es la capacidad de un sistema operativo o una computadora para ejecutar múltiples tareas o procesos de manera aparentemente simultánea permitiendo que los usuarios realicen varias actividades sin necesidad de esperar que una tarea se complete antes de iniciar otra. GNU/Linux hace uso de esta capacidad.

f) POSIX: "Portable Operating System Interface" es un conjunto de estándares desarrollados por la IEEE para definir una interfaz estandar entre SO y aplicaciones en sistemas Unix-like y SO relacionados, permitiendo la portabilidad y la interoperabilidad entre diferentes plataformas.

Componentes más importantes que cubre:

• API (Interfaz de Programación de Aplicaciones): Define las funciones y lla-

madas al sistema que los programas pueden utilizar para interactuar con el SO.

- **Shell y Utilidades:** Especifica un conjunto de comandos y utilidades básicas necesarias en un SO compatible.
- **Modelo de Conurrencia:** Define como manejar conceptos como hilos o threads y semáforos en un entorno multiusuario y multitarea.
- **Portabilidad:** Los estándares POSIX se diseñan para permitir que las aplicaciones sean portátiles entre SO compatibles.

Distribuciones de GNU/Linux:

a) Una distribución es una customización de GNU/Linux formada por una versión de kernel y determinados programas con sus configuraciones, al tener cada distribución sus propias características, enfoques y objetivos, estas se vuelven adecuadas para diferentes usuarios y casos de uso.

Algunas distribuciones:

- **Ubuntu:** Diseñada para ser fácil de usar y adecuada para todo tipo de usuarios. Basada en Debian. Base de Linux Mint.
 - **Debian:** Conocido por su estabilidad y compromiso con el software libre, es base de otras distribuciones.
 - **Fedora:** Enfocada en la adopción temprana de nuevas tecnologías y características.
 - **Linux Mint:** Diseñada para ser atractiva y fácil de usar para usuarios que vienen de SO más "familiares" como Windows.
- Las diferencias básicas entre estas distribuciones serían su Origen y

Base, sus Enfoques y Estabilidad, el Gestor de Paquetes que utilizan, sus Filosofías y comunidades y sus Entornos de Escritorios predeterminados.

c) Debian es una distribución de GNU/Linux que hace énfasis en la libertad de software, en el compromiso con el software completamente libre y tiene su enfoque en la estabilidad y comunicación. Debian fue una de las primeras distribuciones de GNU/Linux, actualmente, Debian utiliza el Kernel de Linux o de FreeBSD e intenta ofrecer también el uso de el Hurd (colección de servidores que se ejecutan sobre un micronúcleo).

Características o Aspectos de Debian:

- Es open source y ofrece libertad de software.
- Ofrece estabilidad con su rama "Stable"
- Tiene un Ciclo de Lanzamiento muy largo, por lo tanto, las actualizaciones y cambios son introducidos cuidadosamente para garantizar estabilidad.
- Ofrece las ramas "Testing" y "Unstable" para el desarrollo.
- Soporta una gran variedad de Arquitecturas.
- Utiliza el sistema de gestión de paquetes APT (Advanced Package Tool).
- Permite elección de entorno de escritorio (GNOME, KDE, Xfce, LXDE, etc)
- Comunidad activa.
- Muchas distribuciones basadas en Debian

Historia de Debian:

- En 1993 Ian Murdock anuncia el proyecto Debian con el objetivo

de crear una distribución de GNU/Linux completamente abierta y basada en la colaboración y desarrollo comunitario. Se continuó el desarrollo del proyecto y la comunidad y en 1996 surge Debian 1.1 "Buzz" la primera versión estable. En el 2000 se lanzó Debian 2.2 "Potato" introduciendo un nuevo sistema de gestión de paquetes y mejoras infraestructurales. Del 2005 al 2009 siguen surgiendo más versiones con más paquetes, en 2013 nace Debian 7.0 "Wheezy" mejorando el rendimiento y el soporte a Arquitecturas ARM, por último, en el 2019 se produce el lanzamiento de Debian 10 "Buster" mejorando aún más el rendimiento y el soporte de hardware. Aún sigue el desarrollo de Debian ...

Estructura de GNU/Linux

a) Los 3 componentes fundamentales de GNU/Linux son: el Núcleo / Kernel, el Interprete de Comandos o CLI (Command Line Interface) y el Sistema de Archivos.

NOTA

b) La estructura básica de GNU/Linux es el Kernel, elemento encargado de que el software y el hardware puedan relacionarse y sus funciones más importantes son la administración de memoria, CPU y la E/S.

Kernel

a) Un Kernel es la parte central de un SO, encargado de gestionar los recursos del hardware y proporcionar servicios esenciales para que los programas y procesos de un sistema computacional puedan funcionar, este, toma el control de la CPU a partir del Vector de Interrupciones.

Historia del Kernel de GNU/Linux:

- En 1991 Linus Torvalds inicia la programación de un kernel, basado en Minix (clon de Unix) y se anuncia la primera versión de Linux.
- En 1992 se combina su desarrollo con GNU.
- En 1994 aparece la versión 1.0 y se continuó el desarrollo.
- En 1996 se adopta a Tux como mascota, se lanza la versión 2.0 y se define la nomenclatura de versionado.
- En 1999 se lanza la versión 2.2 y en 2001 la 2.4 y cerrando en 2003 la versión 2.6 con soporte de hilos, mejoras de planificación y soporte de nuevo hardware.
- En 2011 se lanza la versión 2.6.39.4 y la 3.0.

b) Las funciones principales de un Kernel son: la Gestión de Memoria,

Procesos, Dispositivos, Sistema de Archivos, Comunicación y Sincronización, y de Interrupciones y Excepciones.

c) La versión actual del kernel es la 6.1 desarrollada por Linus Torvalds. El esquema de versionado del kernel anteriores a la 2.4 definía que los números impares indicaban desarrollo y los pares producción. A partir de la versión 2.6 el esquema de versionado fue el siguiente:

- A: Denota Versión y Cambio con menor frecuencia.
- B: Denota mayor revisión.
- C: Denota menor revisión. Cambio con nuevos drivers o características.
- D: Cambio cuando se corrige un grave error sin agregar nueva funcionalidad.

d) Sí, es posible tener más de un Kernel de GNU/Linux en la misma máquina ya que permite a los usuarios tener flexibilidad al elegir qué versión de Kernel usar.

e) El Kernel se encuentra almacenado en la mayoría de distribuciones de GNU/Linux dentro del File System en el directorio "/boot" pero en otras distribuciones los archivos del Kernel pueden encontrarse en directorios como: "/lib/modules", "/etc" o "/usr/src".

f) El Kernel de GNU/Linux es monolítico debido a que es un único programa donde se ejecutan todas las funciones y servicios esenciales. También es híbrido por tener la capacidad de cargar y descargar funcionalidad a través de módulos/drivers.

Interprete de Comandos (Shell).

a) El Shell es la interfaz principal entre un usuario sentado en su terminal y el sistema operativo, a menos que el usuario esté usando una interfaz gráfica de usuario. Esta interfaz acciona como modo de comunicación y ejecuta programas a partir del ingreso de comandos.

b) Funciones del Interprete de Comandos:

- Ejecución de comandos: Permite a los usuarios ejecutar programas y comandos del SO.
- Gestión de Archivos y directorios: Permite crear, eliminar, mover, copiar y listar Archivos y Directorios.
- Redirección y Tuberías: Permiten redirigir la entrada y salida de los comandos y permite crear pipes/tuberías para conectar la salida de un comando y la entrada de otro.
- Variables y Entorno: Permite crear variables y acceder a las variables de entorno.
- Control de Procesos: Permite ejecutar procesos en primer y segundo plano como también detenerlos, pausarlos, reanudarlos y administrarlos.
- Scripting: Brindan un entorno para escribir scripts.
- Control de Permisos y Seguridad: Permite administrar permisos de Archivos y Directorios, y realizar tareas relacionadas con la seguridad.
- Interacción con herramientas del sistema: Puede interactuar con

herramientas y utilidades del sistema.

- Personalización: Permite personalizar el entorno de shell.

d) Los comandos propios y externos al Shell se ubican en "/bin" (esenciales) y en "/usr/bin"

e) El Shell no es parte del Kernel de GNU/Linux ya que no provee ninguna de las funcionalidades que un Kernel debe ofrecer, el Shell solo es un medio de comunicación entre el SO y el usuario.

c) Interpretadores de Comandos:

● Bash (Bourne Again Shell)

● Popularidad: Es el más comúnmente usado y el predeterminado en varias distribuciones.

● Características: Es versátil y poderoso. Admite expansión de comandos y variables, redirección y pipes, autocompletado, etc.

● Flexibilidad: No es el más rápido en todas las operaciones pero posee amplia disponibilidad.

● Zsh (Z Shell)

● Características Avanzadas: Conocido por sus características avanzadas y su autocompletado inteligente y potente (incluye sugerencias contextuales). También ofrece expansión más sofisticada y funciones de corrección automática.

● Mejoras sobre Bash: Hereda muchas características de Bash pero agrega mejoras, como el autocompletado más inteligente y la sintaxis más avanzada.

● Uso: Especialmente útil para quien valore la eficiencia y la personalización.

● Bourne Shell (sh)

● Características: Sintaxis básica y funcionalidades esenciales para la ejecución de comandos y manipulación de Archivos. No ofrece características avanzadas y es limitado en términos de conveniencia y características de scripting.

● Flexibilidad: Es ligero y adecuado para tareas simples y scripts básicos.

● Uso: No es la mejor opción para usuarios que buscan una experiencia más avanzada en la línea de comandos

Si, es posible definir un intérprete de comandos distinto para cada usuario en GNU/Linux. Esto se puede definir a través de la configuración de varios Archivos alojados en "/home/usuario" que varían según el intérprete que se esté usando ("~.bashrc", ".zshrc", ".fishrc", etc.). Generalmente los usuarios normales pueden definir qué intérprete usar siempre que tengan permisos de escritura en los archivos de configuración relevantes en su directorio de inicio, aunque, hay excepciones según la administración y configuración del sistema.

File System

a) El File System es una estructura y conjunto de reglas que utiliza un SO para administrar y rastrear la información en el

almacenamiento. Define cómo se organizan los archivos, cómo se almacenan en el disco, cómo se accede a ellos, cómo se nombran y cómo se controla el acceso a ellos por parte de usuarios y programas.

b) File Systems soportados por GNU/Linux (Algunos):

- FAT (File Allocation Table)
- NTFS (New Technology File System)
- EXT4 (Fourth Extended File System)
- BTRFS (B-Tree File System)

c) Sí, es posible visualizar particiones del tipo FAT (16 y 32) y NTFS en GNU/Linux ya que este proporciona soporte para estos File Systems a través de controladores y utilidades.

d) Directorios más importantes según FHS (Filesystem Hierarchy Standard):

- /: Tope de la estructura de escritorios.
- /home: Se almacenan archivos de usuarios.
- /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.

Particiones

a) Las Particiones son divisiones lógicas o segmentos en un dispositivo de almacenamiento. Cada partición funciona como una unidad separada con su propio File System y espacio de almacenamiento. Se utilizan para organizar y administrar datos eficientemente.

NOTA

Tipos de Particiones:

Partición Primaria:

- División cruda del disco (puede haber 4 por disco). Se almacena información de la misma en el MBR.
- Ventajas:
 - Compatibilidad de Arranque: Son capaces de contener bootloaders.
 - Independencia: Cada partición es independiente de las demás.
 - Flexibilidad: Permiten tener múltiples SO en el mismo disco.
 - Respaldo y Recuperación: Se pueden crear particiones dedicadas al respaldo de información.
 - Rendimiento optimizado.

Desventajas:

- Límite de 4 particiones.
- Limitaciones en SO antiguos: En algunos SO solo se pueden usar particiones de este tipo.
- Tamaño Limitado: El tamaño de la partición se define al crearse.
- Pueden existir Restricciones d. Hardware.

Partición Extendida:

- Sirven para contener unidades lógicas en su interior. Solo puede existir una partición de este tipo por disco. No se define un File System sobre ella directamente. (Pero si indirectamente).

tamente).

● Ventajas:

- Superar la limitación de 4 particiones primarias: Se puede tener 3 particiones primarias y una partición extendida, con varias particiones lógicas.
- Mayor flexibilidad: Se pueden redimensionar o crear particiones lógicas sin modificar la extendida.
- Organización: Se segmentan y organizan los datos de manera más eficiente y lógica.
- Separación de Datos.

● Desventajas:

- Compatibilidad de Arranque: No pueden contener el bootloader directamente.
- Restricciones en SO Antiguos.
- Complejidad: Mantener un esquema claro y bien organizado de las particiones secundarias es una tarea importante y difícil.

● Partición Secundaria:

- Ocupa la totalidad o parte de la partición extendida y se le define un tipo de File System. Estas particiones se conectan como una lista enlazada. Puede tener bootloader.

b) En GNU/Linux las particiones se identifican combinando el nombre del dispositivo de almacenamiento con un número que representa la partición. Los nombres de los dispositivos varían según la interfaz.

Discos IDE:

- Los dispositivos se nombran hda, hdb, hdc, etc. y las particiones del 1 al 4, ejm: /dev/hda1 ; /dev/hdb2.

Discos SCSI y SATA:

- Los dispositivos se nombran sda, sdb, sdc, etc. y las particiones del 1 al 4, ejm: /dev/sda3 ; /dev/sdc1.

c) Como mínimo es necesario 1 partición (para el '/') pero es recomendable crear una segunda. (para la SWAP).

Partición Raíz (/):

- Tipo: Primaria o Lógica.
- Identificación: /dev/sdX1 (sdX = nombre del dispositivo).
- Tipo de File System: Cualquier compatible con GNU/Linux.
- Punto de montaje: '/'.

Partición de intercambio (SWAP):

- Tipo: Primaria o Lógica.
- Identificación: /dev/sdX2 (ejemplo).
- Tipo de File System: al ser Área de Intercambio no utiliza.
- Punto de Montaje: N/A, no se monta como una carpeta..

e) Existen 2 tipos de software particionador:

Destructivos: Permiten crear y eliminar particiones.

No Destructivos: Permiten crear, eliminar y modificar particiones.

Fdisk (Destructivo): Utilidad de línea de comandos que se utiliza para crear, eliminar y administrar particiones.

Fips (No Destructivo): Herramienta que se utiliza específicamente para cambiar el tamaño de particiones FAT16 en sistemas antiguos.

Gparted (No Destructivo): Herramienta gráfica de administración de particiones que proporciona una interfaz visual y amigable.

Arranque (bootstrap) de un Sistema Operativo

a) BIOS (Basic Input/Output System): Es un firmware (programa que establece la lógica de más bajo nivel de un dispositivo) presente en la motherboard. Tiene como función esencial inicializar y coordinar el hardware de la computadora durante el proceso de arranque, antes de que el SO se cargue en memoria.

b) UEFI (Unified Extensible Firmware Interface): Firmware utilizado como reemplazo del BIOS tradicional. UEFI es una tecnología más avanzada que ofrece varias ventajas como por ejemplo: interfaz gráfica, arranque más rápido, mayor compatibilidad, mejor seguridad, configuración más avanzada, etc. Su función es administrar la inicialización y la carga del sistema operativo, así como facilitar la interacción entre el firmware, el hardware y el software. También define la ubicación del gestor de arranque y la interfaz entre el gestor de arranque y el firmware.

c) MBR (Master Boot Record): Es una estructura crítica ubicada en el cilindro 0, cabeza 0, sector 1 de cada dispositivo de memoria (en este caso disco), si existiese más de un disco en la máquina, sólo uno es designado Primary master Disk (Unidad de disco duro principal en un sistema informático) tiene un tamaño de 512 Bytes (los primeros 446 son para el MBC/Master Boot Code, luego queda una tabla de particiones de 64 bytes y 2 bytes

libres o para firmar el MBR) y es parte fundamental para el proceso de arranque en sistemas que usan el BIOS como firmware. El MBR contiene información para iniciar el SO y ubicar la partición activa desde la cual se cargará el cargador de arranque (bootloader).

MBC (Master Boot Code): Pequeño código que permite arrancar el sistema operativo. La última acción del BIOS es leer el MBC. Lo lleva a memoria y lo ejecuta.

d) GPT (GUID Partition Table) es el sistema que utiliza UEFI para solucionar limitaciones del MBR, como la cantidad de particiones. GPT especifica la ubicación y formato de la tabla de particiones en un disco duro, forma parte de EFI (Extensible Firmware Interface) y se puede ver como sustitución del MBR.

e) Un Bootloader es un programa que se encarga de iniciar el proceso de carga del SO. Su funcionalidad principal es cargar el sistema operativo desde el disco y transferir el control al sistema operativo cargado. También puede permitir elegir al usuario entre múltiples SO instalados.

Existen varios tipos, pero los dos más comunes son GRUB (Grand Unified Bootloader) y LILO (Linux Loader).

Los gestores de arranque se instalan en el sector de arranque de la partición raíz o activa (Volume Boot Record) o en el MBR.

f) El BIOS es el responsable de iniciar la carga del SO a través del MBC, este carga el programa de booteo desde el MBR. El gestor

de arranque lanzado desde el MBC carga el kernel: prueba y hace disponibles los dispositivos y luego pasa el control al proceso init.

g) Proceso de Arranque en GNU/Linux:

- Encendido y POST: Al encender la computadora, la CPU ejecuta instrucciones desde la dirección de arranque. El firmware (BIOS o UEFI) realiza la Prueba de Autodiagnóstico en el Arranque (POST) verificando el hardware.
- Bootloader: Se ejecuta el bootloader por ejm. GRUB y se muestra el menú de arranque que permite al usuario elegir diferentes SO o configuraciones y se carga el Kernel en memoria.
- Carga del Kernel: El Bootloader carga al Kernel en RAM y este empieza a administrar los recursos del sistema, drivers y las interacciones con el hardware.
- Init/Inicio del Sistema: Se inician los servicios esenciales, se configura el entorno del sistema y se realizan tareas de inicialización.
- Montaje del File System: El SO monta el File System, como la partición raíz '/', entre otras necesarias.
- Se activan servicios y procesos en segundo plano necesarios (servicios de red, bases de datos, etc)
- Runlevel o Targets: Dependiendo del sistema de inicio, se ingresará a un runlevel que determinará que servicios se inicien automáticamente. Esto también se puede lograr mediante la activación de targets.
- Logeo al sistema.
- Se carga el Entorno de Usuario (gráfico o de línea de comandos).

h) Proceso de parada o shutdown:

- Se cierran las sesiones de usuario.
- Se detienen los servicios y procesos en segundo plano.
- Se desmonta el File System para que este entre en estado seguro.
- Se envía una señal de terminación a los procesos en ejecución y se espera a que terminen. Si estos no terminan, se envía una señal más fuerte para forzar el cierre.
- Se desactivan dispositivos de hardware y drivers.
- Si es necesario, el sistema desmonta el disco de forma segura.
- El kernel realiza tareas de limpieza y apagado para luego cerrarse.
- Se envía una señal al firmware para apagar el hardware de manera segura.

i) Sí, ya que un disco puede ser particionado y en cada partición se puede tener un sistema operativo diferente, otro File System y un gestor de arranque que nos permita escoger el sistema operativo que queremos usar.

Práctica I

Archivos:

a) Un archivo de GNU/Linux se identifica por su nombre, su ubicación (Ruta Absoluta y Ruta Relativa), su permiso y propietario y su tipo de archivo y sus metadatos

b) Editores de Texto:

- Vi: Editor de texto en línea de comandos que se encuentra en varios sistemas Unix like. Tiene 2 modos principales:

- Modo Comando: Se puede realizar acciones como copiar, pegar, eliminar y buscar.

- Modo Edición: Permite editar texto.

- mcedit (del Paquete Midnight Commander): Editor de texto que forma parte del paquete Midnight Commander, un administrador de archivos en modo de texto. mcedit ofrece una interfaz más amigable y se asemeja más a editores tradicionales con atajos de teclado familiares.

Comandos:

- cat: Se utiliza para mostrar el contenido de uno o varios archivos de texto en la salida estándar. Se puede usar para ver el contenido completo de un archivo.

- more: Es un visualizador de texto paginado. Muestra el contenido de un archivo de manera incremental, una página a la vez.

d) Comando "file": Determina el tipo de archivo de un archivo dado, también se puede ver el tipo de archivo, el formato, la codificación de caracteres, etc.

10.

o) "mkdir" (make directory): La función principal es crear uno o más directorios en la ubicación especificada en tu sistema de archivos.

Parámetros:

- "-p": Permite crear directorios padres si estos no existen.

Por ejm. mkdir -p padre/hijo crea las carpetas "padre" e "hijo" si estas no existen.

- "-m": Permite especificar en notación octal los permisos o modos que se desean asignar al directorio. Ejemplo: mkdir -m 755

carpeta.

- "-v": Sirve para ver una descripción detallada de lo que está haciendo mkdir.

Se pueden crear varios directorios haciendo "mkdir direct1 direct2"

b) "cd": Se usa para cambiar el directorio de trabajo actual en la línea de comandos. Permite navegar por el sistema de archivos y acceder a diferentes directorios.

Parámetros:

- "cd" (sin parámetros): Vuelve al directorio de inicio del usuario actual.

- "cd /ruta/al/directorio": Se puede usar la ruta del directorio al que se desea cambiar.

- "cd ..": Permite moverte un nivel hacia arriba en la estructura de directorios (moverse al padre).

• "cd .": Representa el directorio actual y se usa para referirse al mismo.

• "cd -": Te lleva al directorio anterior al que estabas antes de cambiar.

c) "touch": Se utiliza para crear archivos vacíos o actualizar las marcas de tiempo (timestamp) de archivos existentes.

Parámetros:

• "touch" (sin parámetros): Crea un archivo vacío si no existe, en caso contrario se actualizan las marcas de tiempo a la hora actual.

• "touch -c": No crea ningún archivo.

• "touch -t": Permite especificar una marca de tiempo específica en lugar de la hora actual en un archivo. Se debe proporcionar la marca en un formato específico.

d) "ls": Se utiliza para listar los archivos y directorios en el directorio actual o en una ubicación especificada.

Parámetros:

• "ls -l": Sirve para ver una lista detallada que incluye información adicional sobre cada elemento (permisos, propietario, grupo, tamaño, fecha de modificación y nombre).

• "ls -a": Muestra todos los archivos, incluso los ocultos.

• "ls -h": Combinado con "-l", este parámetro muestra tamaños de archivo más legibles usando KB, MB, GB.

• "ls -t": Ordena la lista por fecha y hora de modificación.

• "ls -r": Invierte el orden de la lista.

el "pwd": Se utiliza para mostrar la ruta completa del directorio actual en la estructura del sistema de archivos.

Parámetros:

- "pwd -P": Muestra la ruta física del directorio actual sin resolver enlaces simbólicos.

- "pwd -L": Muestra la ruta lógica que sigue los enlaces simbólicos en lugar de la ruta física.

el "find": Se utiliza para buscar archivos y directorios en base a criterios necesarios utilizando ciertos patrones o propiedades dentro del sistema de archivos.

Parámetros:

- "find ruta /a /buscar": Se debe especificar desde donde se quiere iniciar la búsqueda, desde la ruta especificada hacia abajo.

- "find ruta -name nombre": Se usa para buscar por nombre, usando asteriscos se pueden realizar búsquedas parciales o múltiples coincidencias. "cadena *" busca todos los que contengan la cadena.

- "find ruta -type": Se usa para especificar el tipo de objeto a buscar. "f" para archivos regulares y "d" para directorios.

- "find ruta -mtime y -atime": Permiten buscar basándose en su tiempo de modificación (-mtime) o su tiempo de acceso (-atime) en días.

- "find ruta -size": Búsqueda por tamaño de archivo.

- "find ruta -name "*.txt" -exec rm {} \": "-exec" permite ejecutar comandos en los archivos encontrados, se debe proporcionar

el comando a ejecutar y usar `{}3` para representar el nombre del archivo.

g) "find ruta -print": Muestra en pantalla los resultados de la búsqueda.

h) "df": Se usa para mostrar información sobre el espacio en disco disponible. Proporciona una vista resumida de la utilización del disco en cada uno de los sistemas de archivos montados, mostrando la lista de todos los sistemas de archivos junto con información sobre la capacidad total, espacio utilizado y porcentaje utilizado.

Parámetros:

• "df -h": Muestra los tamaños en formatos más legibles.

• "df -T": Muestra el tipo de sistema de archivos.

• "df -i": Muestra información sobre inodos utilizados y disponibles en lugar del espacio en el disco.

• "df -x ntfs": Excluye tipos de file systems de lo salido.

h) "who": Muestra información sobre los usuarios conectados al sistema (nicknames, terminales a los que están conectados, direcciones IP de ser disponibles, fecha y hora de inicio de sesión y si están inactivos o no).

Parámetros:

• "who -q": muestra el número total de usuarios conectados al sistema.

j) "tail": Se usa para mostrar las últimas líneas de un archivo de texto o de la salida de otro comando (últimas 10 líneas)

11.

a) "shutdown": Se utiliza para programar el apagado o reinicio del sistema.

Parámetros:

- "-h": Apaga el sistema.
- "-r": Reinicia el sistema.
- "-c": Cancela un comando de apagado o reinicio programado.
- "+m": Establece el tiempo en minutos antes de que se apague o reinicie el equipo. m = cant. minutos
- "now": Apago o reinicio inmediatamente.
- "HH:MM": Establece hora específica de apagado o reinicio.

b) "reboot": Se usa para reiniciar la computadora inmediatamente.

Parámetros:

- "-f": Fuerza el reinicio, no importa si hay procesos ejecutándose.
- "-n": No registra el reinicio en el archivo /var/log/wtmp, que rastrea las actividades de inicio y cierre de sesión en el sistema.
- "-d": Realiza un reinicio en modo depuración.

c) "halt": Apaga la computadora de manera inmediata y abrupta cortando la energía. Usar con precaución.

Parámetros:

• "-p": Apaga el sistema de forma segura. Se asegura de que se realice un cierre ordenado del sistema antes de apagarlo.

• "-f": Fuerza el apagado inmediato.

• "-n": No registra el apagado en el archivo /var/log/wtmp.

• "-d": Apagado en modo depuración.

dlocate: Se utiliza para buscar archivos y directorios en la base de datos del sistema que contiene una lista de todos los archivos y directorios indexados. Es rápido y eficiente pero sus resultados pueden no ser en tiempo real.

Parámetros:

• "locate <patrón>": Búsqueda de archivos/directorios que coincidan con el patrón especificado.

• "-c": Muestra la cantidad de coincidencias encontradas.

• "-i": Búsqueda no Key sensitive.

• "-l <n>": Limita el número de resultados mostrados a "<n>"

• "-r": Realiza una búsqueda usando expresiones regulares en lugar de coincidencias de patrones simples.

• "-S": Actualiza manualmente la base de datos de índices del sistema.

el "uname": Obtiene información sobre el sistema operativo en el que estás trabajando y el hardware en el que se ejecuta.

Parámetros:

• "-a": Muestra toda la información disponible (nombre del SO, versión del Kernel, nombre del Host, tipo de máquina y fecha de construcción).

- "-s": Muestra el nombre del sistema operativo.
- "-r": Muestra la versión del Kernel.
- "-v": Muestra información adicional sobre la versión del Kernel y la fecha de construcción.
- "-n": Muestra el nombre del nodo de red o del host.
- "-m": Muestra la arquitectura del hardware.
- "-p": Muestra el tipo del procesador del sistema.
- "-i": Muestra el nombre de la plataforma de hardware.
- "-o": Muestra el nombre del SO en un formato más legible.

El "dmesg": Muestra mensajes del Kernel que proporcionan información sobre eventos, advertencias y errores en el sistema. Puede ser útil para la resolución de problemas, diagnóstico de hardware y seguimiento de eventos del sistema.

Parámetros:

- "-c": Borra el buffer de mensajes del kernel después de mostrar los mensajes actuales.
- "-n <nivel>": Establece el nivel de mensajes que se mostrarán.
- "=T": Muestra fecha y hora en formato legible junto con los mensajes.
- "-l": Muestra una lista de las propiedades de los niveles de mensaje y sus descripciones.
- "-r": Muestra los mensajes en formato bruto.
- "-s": Limita el tamaño del buffer a un tamaño específico.
- "-w": Muestra mensajes en tiempo real y continua mostrando nuevos a medida que se generan.

g) "lspci": Muestra una lista de todos los dispositivos PCI detectados en el sistema. Proporciona información detallada sobre cada dispositivo, incluyendo su identificador de fabricante, identificador de dispositivo, tipo de dispositivo y otras características.

Parámetros:

"-v": Información detallada de los dispositivos.

"-k": Información sobre los controladores de kernel asociados a los dispositivos PCI.

"-n": Muestra los Identificadores en modo numérico.

"-s": Muestra información sobre un dispositivo específico utilizando su ubicación PCI

"-t": Muestra en forma de árbol la jerarquía de buses PCI en el sistema.

h) "at": Programa tareas para que se ejecuten en una fecha y hora determinadas en el futuro. Permite automatizar tareas periódicas o únicas sin tener que usar Crón (servicio para la programación de tareas).

Puede ser útil para realizar copias de seguridad, informes programados, ejecutar tareas de mantenimiento, etc. Ver at --help.

i) "netstat": Muestra una lista de todas las conexiones de red activas en el sistema, las tablas de enrutamiento utilizadas por el SO para enviar paquetes de datos a través de la red, estadísticas detalladas sobre cada interfaz de red, información sobre grupos multicast y rutas multicast en el sistema, estadísticas generales de protocolo de red y también sobre sockets UNIX. Ver netstat --help

j) "mount": Monta sistemas de archivos en el SO, lo que permite que los archivos y directorios en ese sistema de archivos sean accesibles a través del punto de montaje especificado. El comando mount sin parámetros muestra una lista de todos los file systems montados en el sistema. Ver mount --help.

k) "umount": Se utiliza para desvincular o desmontar un sistema de archivos previamente montado de un punto de montaje en el sistema.

l) "head": Muestra las primeras líneas de un archivo de texto o la salida de un comando en la terminal. Por defecto se muestran 10 líneas pero el número se puede ajustar utilizando un parámetro. Ver head --help

m) "losetup": Configura dispositivos loopback, que pueden estar asociados con archivos regulares. También permite montar imágenes de disco como si fueran dispositivos físicos permitiendo el acceso a el contenido de archivos de imagen de disco sin necesidad de escribirlos en un disco real. Ver losetup --help

n) "write": Permite que un usuario envíe mensajes de texto a otros usuarios que estén conectados en la misma terminal o sesión, los mensajes se entregan de inmediato y pueden ser útiles para la comunicación en tiempo real. Ver write --help.

ñ) "mkfs": Crea un file system en un dispositivo de almacenamiento, permitiendo organización y almacenamiento de datos de forma estructurada. Ver mkfs --help

o) "fdisk": Administra las particiones en dispositivos de almacenamiento y también brinda información detallada sobre las mismas y el disco. Usar con cuidado.

12.

a) Normalmente se encuentran en /bin o /usr/bin, en mi caso, también encontré algunos en /c/WINDOWS/system32. Usar comando which para saber donde están.