**TRABAJO PRACTICO 1**

**Definiciones:**

**Unix:** Fue un sistema operativo usado en los años 70. Era muy estable, multitarea, multiusuario y con la filosofia quetodo es un archivo(Incluso los dispositivos).

**Firmware:** También conocido como soporte lógico inalterable, es un programa informático que establece la lógica de más bajo nivel para controlar los circuitos electrónicos de un dispositivo de cualquier tipo.

**SIGTERM:** Pide que el proceso termine de forma ordenada (le da la chance de cerrar archivos, liberar memoria, etc).

**1\_ Caracteristicas de GNU/Linux**

**A\_ Mencione y explique las características más relevantes de GNU/Linux**

* Es un sistema operativo tipo [Unix](#Unix), pero libre.

**Aclaracion**: Unix fue un sistema operativo usado en los años 70. Era muy estable, multitarea, multiusuario y con la filosofia que todo es un archivo(Incluso los dispositivos).

* Sistema operativo gratuito y de libre distribucion.
* Es multiusuario, multitarea y multiprocesador.
* Es altamente portable(el sistema operativo puede funcionar en muchos tipos de computadoras y dispositivosdiferentes, sin importar la marca o la arquitectura del procesador).
* Permite el manejo de usuarios y permisos.
* Es de codigo abierto, esto nos permite estudiarlo, personalizarlo, auditarlo, etc.

**B\_ Mencione otros sistemas operativos y compárelos con GNU/Linux**

**Windows:**

* Propietario → creado y mantenido por Microsoft. No se puede ver ni modificar el código fuente.
* Enfocado en la facilidad de uso y compatibilidad con software comercial (juegos, programas de oficina, etc.).
* También es multitarea y multiusuario, pero históricamente fue más vulnerable a virus y malware.
* Actualizaciones y soporte dependen exclusivamente de Microsoft.

**Ios:**

* **Propietario y cerrado**, controlado por Apple.
* Basado en Unix (en realidad sobre Darwin, que a su vez viene de BSD, un Unix libre).
* Muy **restringido**: solo se pueden instalar aplicaciones desde la App Store (salvo que lo liberes con jailbreak).
* Está optimizado para funcionar en **dispositivos móviles Apple** → fluidez, consumo de batería, integración con el hardware.

**C\_ ¿Qué es GNU?**

GNU es un sistema operativo libre, compatible con Unix, su objetivo es garantizar que los usuarios tengan la libertad de usar, estudiar, modificar y compartir el software.

**D\_ Indique una breve historia sobre la evolución del proyecto GNU.**

En 1983 Richard Stallman inicio con la idea de crear un Unix libre, para esto necesito crear un marco regulatorio conocido como GLP(General public license de GNU). Para 1990 ya contaba con un editor de texto, un compilador y gran cantidad de bibliotecas que componen un Unix tipico, todavia faltaba el componente principal(Nucleo Kernel). En 1991 Linus Torvalds venía trabajando en un Kernel denominado Linux, el cual iba a ser distribuido bajo la licencia GPL, en 1992, Torvalds y Stallman deciden fusionar ambos proyectos y ahí nace GNU/Linux.

**E\_ Explique qué es la multitarea, e indique si GNU/Linux hace uso de ella.**

Multitarea significa que el sistema operativo permite ejecutar varios procesos a la vez, compartiendo el tiempo de CPU. GNU/Linux implementa multitarea real, asignando de manera controlada los recursos a cada proceso según su prioridad.

**F\_ ¿Qué es POSIX?**

POSIX es un estándar que define una interfaz común para los sistemas operativos tipo Unix, con el objetivo de garantizar portabilidad y compatibilidad de aplicaciones entre distintos sistemas.

Comandos como ls, cd, cat, chmod existen en todos los sistema

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 variante del sistema que combina el kernel de Linux con herramientas GNU y software adicional.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Distribucion** | **Enfoque** | **Dificultad** | **Actualizaciones** | **Uso** |
| **Ubuntu** | Facilidad de uso | Facil | Frecuentes | Principiantes, uso general |
| **Debian** | Estabilidad | Medias | Lentas | Servidores, entornos criticos |
| **Fedora** | Innovación | Medias | Muy rapidas | Desarrolladores |
| **Arch** | Control total | Alta | Constantes | Usuarios avanzados |

**b. ¿En qué se diferencia una distribución de otra?**

La principal diferencia entre las distribuciones GNU/Linux es el público al que están destinadas: algunas priorizan la facilidad de uso (Ubuntu), otras la estabilidad (Debian), la innovación (Fedora) o el control total para usuarios avanzados (Arch Linux).

**c. ¿Qué es Debian? Acceda al sitio https://www.debian.org/ e indique cuáles son los objetivos del proyecto y una breve cronología del mismo.**

Debian es una distribución de GNU/Linux (una de las más antiguas y estables), es software libre, mantenida por la comunidad, y se caracteriza por su estabilidad y confiabilidad.

Su objetivo es crear un sistema operativo libre, disponible para todo el mundo(Libre se refiere al software, no al dinero).

**3. Estructura de GNU/Linux:**

**a. Nombre cuáles son los 3 componentes fundamentales de GNU/Linux.**

**Kernel:** Es la parte central y más crucial del sistema operativo, actuando como interfaz entre el hardware y los procesos del sistema.

**Bibliotecas:** Son fundamentales para el funcionamiento de las aplicaciones y deben ser compatibles con estándares como POSIX.

**Systema de archivos(File System):** Es la forma en la que GNU/Linux organiza y guarda informacion en discos y directorios.

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

Se basa en una gerarquia de archivos organizada en un arbol, con el directorio raíz representada por “/”. Esta organización sigue el estandar FHS(Filesystem Hierarchy Standard).

**4. Kernel:**

**a. ¿Cuáles son sus funciones principales?**

Es el encargado de que el software y hardware puedan trabajar juntos. Su función más importante es la administracions de memoria, cpu y E/S.

**b. ¿Cuál es la versión actual?**

La version actual es la 6.9 que fue lanzada el 13 de mayo de 2024

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

Los numeros versiones que terminaban en numero par eran versiones estables y las de numero impar eran de desarrollo.

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

Se unificó el desarrollo, se abandono la conveción que venian utilizando y se publicaron versiones estables con más frecuencia

**c. ¿Es posible tener más de un Kernel de GNU/Linux instalado en la misma máquina?**

Es posible tener más de un kernel instalado en la misma maquina, pero a la hora de arrancar el sistema solo se permite cargar uno.

**d. ¿Dónde se encuentra ubicado dentro del File System?**

Se encuentra ubicado en “/boot” desde donde el gestor de arranque lo carga al iniciar el sistema.

**5. Intérprete de comandos (Shell):**

**a. ¿Qué es? b. ¿Cuáles son sus funciones?**

Es un modo de comunicación entre el SO y el usuario. Ejecuta programas a partir del ingreso de comandos.

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

**Bourne shell:** Muy basico, pero estable. Ideal para scripts sencillos, gracias a su compatibilidad hoy en día sigue siendo usado como base para scripts.

**Korn shell:** Incluye historial de comandos, edición en linea y variables de entorno más potentes. Hoy en día todavía se utiliza en entornos de Unix clásicos.

**Bourne again shell:** Tienehistorial de comandos, autocompletado, alias, scripting potente, soporte de arrays, funciones, etc. Hoy en día es el más usado en la mayoría de distribuciones GNU/Linux.

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

* **Comandos internos**: están integrados dentro del propio shell (ej: cd, echo), por lo tanto no tienen un path en el sistema de archivos.
* **Comandos externos**: son programas ubicados en directorios como /bin, /usr/bin, /sbin, /usr/sbin, y se ejecutan buscándolos a través de la variable PATH.

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

El Shell no es parte del Kernel porque no administra hardware ni recursos, sino que actúa como intérprete entre el usuario y el kernel, ejecutándose en espacio de usuario.

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

Si, cada usuario puede tener un shell diferente

**¿Desde dónde se define?**

Se define en el archivo /etc/passwd (último campo de cada línea) o usando el comando chsh.

**¿Cualquier usuario puede realizar dicha tarea?**

Cada usuario puede modificar su propio shell, el administrador es el que puede modificar cualquier usuario.

**6. El sistema de Archivos (File System) en Linux:**

**a. ¿Qué es?**

El sistema de archivos de Linux es una estructura jerárquica de árbol de directorio único que comienza en el directorio raíz, representado por una barra diagonal (/), bajo el cual se organizan todos los demás archivos, directorios y dispositivos.

**b. ¿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 significa Filesystem Hierarchy Standard(Sistema de archivos Estándar de jerarquía).

Es una estructura jerarquica de arbol de directorios, los directorios mas importantes son:

* / 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

**c. Mencione sistemas de archivos soportados por GNU/Linux.**

fat, ntfs ext2, ext3, reiser, etc.

**d. ¿Es posible visualizar particiones del tipo FAT y NTFS (que son de Windows) en GNU/Linux?**

Sí, GNU/Linux puede visualizar particiones FAT y NTFS. El soporte para FAT es nativo y completo, mientras que para NTFS se logra mediante controladores como ntfs-3g, lo que permite acceder a los datos de Windows desde Linux.

**7. Particiones:**

**a. Definición. Tipos de particiones. Ventajas y Desventajas.**

Consiste en dividir logicamente el disco fisico. Los tipos de particiones son: Primaria, extendida y logica.

* **Ventajas:** separación de datos, seguridad, posibilidad de múltiples SO, mejor rendimiento y mantenimiento más sencillo.
* **Desventajas:** espacio rígido y difícil de redimensionar, mayor complejidad, riesgo de fragmentar el espacio libre y problemas si se configuran mal.

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

**IDE:** Particiones primarias desde la 1 a la 4, las particiones logicas van de la 5 en adelante.

**SCSI/SATA:** Particiones primarias enumeradas de la 1 a la 4, las particiones extendidas arrancan a partir de la 5.

**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.**

Como mínimo, GNU/Linux requiere una partición para la raíz /. Es recomendable tener 2(/, SWAP).

* Es una particion primaria.
* El punto de montaje es “/” ya que es la raiz del arbol desde donde salen todos los directorios.
* El tipo de file System es ext4(El más usado actualmente)

**d. Dar ejemplos de diversos casos de particionamiento dependiendo del tipo de tarea que se deba realizar en su sistema operativo.**

El esquema de particionamiento depende del uso:

* Escritorio personal: /, swap, /home.
* Servidor: además de lo anterior, separar /var y /tmp.
* Desarrollo: /home grande y opcional /usr/local.
* Cómputo científico: añadir particiones de datos (/data o /scratch).
* Dual boot: incluir partición de Windows y, opcionalmente, una compartida.

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

Existen particionadores destructivos (como fdisk), que eliminan los datos al modificar particiones, y no destructivos (como gparted, parted), que permiten crear o redimensionar particiones manteniendo la información.

**8. Arranque (bootstrap) de un Sistema Operativo:**

**a. ¿Qué es el BIOS? ¿Qué tarea realiza?**

Es un sistema basico de entrada y salida. Es el encargado de iniciar el SO atraves del MBC.

**b. ¿Qué es UEFI? ¿Cuál es su función?**

UEFI significa “Unified Extensible Firmware Interface” (Interfaz de [Firmware](#Firmware) Unificada y Extensible).

Su función es exponer informacion para los gestores de arranque, definir la ubicación delgestor de arranque y definir la interfaz entre el gestor y el firmware.

**c. ¿Qué es el MBR? ¿Qué es el MBC?**

El Registro de Arranque Maestro (**MBR**, por sus siglas en inglés Master Boot Record) es el primer sector físico de un dispositivo de almacenamiento, como un disco duro o una unidad USB, ubicado en la dirección Cilindro 0, Cabeza 0 y Sector 1.

El código de arranque maestro (Master Boot Code, **MBC**) es una parte fundamental del registro de inicio maestro (MBR), que se encuentra en el primer sector del disco duro, ocupando 446 de los 512 bytes totales.

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

Hace referencia a “GUID partition table”. EFI puede verse como una sustitucion del MBR ya que utiliza el sistema GPT para solucionar las limitaciones del MBR, como la cantida de particiones

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

La función del gestor de arranque (bootloader) es cargar una imagen del kernel de alguna particion para su ejecución. Entre los más conocidos están **GRUB, LILO, NTLDR, GAG, YaST, etc**. Se pueden instalar en el **MBR** (en sistemas con BIOS), en el sector de arranque de la partición raíz, o en la partición **EFI** (en sistemas con 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)?**

El bootstrap es el proceso de arranque de la computadora: comienza al encender el equipo con la BIOS/UEFI, que carga el sector de arranque (MBR/EFI), luego el gestor de arranque, el kernel de GNU/Linux y finalmente los procesos de usuario.

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

1. BIOS/UEFI (POST y búsqueda de dispositivo de arranque).
2. MBR o partición EFI.
3. Gestor de arranque (GRUB).
4. Kernel de Linux (se carga desde /boot).
5. Proceso init/systemd (lanza servicios y pocesos ene segundo plano).
6. Inicio de sesión del usuario.

**h. ¿Cuáles son los pasos que se suceden en el proceso de parada? (shutdown)**

El proceso “shutdown” ocurre cuando se apaga la pc y sigue los siguientes pasos:

1. Aviso de apagado
2. Finalizacion de procesos
3. Detencion de sistemas y procesos en segundo plano
4. Sincronizacion de datos
5. Desmontaje de sistemas de archivos
6. Apagado del kernel

**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. Para ello, cada sistema debe instalarse en una partición diferente y el gestor de arranque permite elegir cuál iniciar.

**9. Archivos y editores:**

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

En GNU/Linux los archivos se identifican principalmente por su ruta absoluta dentro del árbol de directorios, y a nivel interno por un número de inodo, que contiene toda la información sobre el archivo. Además, el sistema es *case sensitive*, por lo que distingue mayúsculas de minúsculas en los nombres.

**b. Investigue el funcionamiento de los editores vim, nano y mcedit, y los comandos cat, more y less.**

**Vim:** Dispone de un lenguaje interpretado(o scripting) para programar nuevas funcionalidades. Es un editor de texto de codigo abierto, libre y multiplataforma.

**Nano:** Se basa en una arquitectura de datos innovadora llamada Block Lattice, donde cada cuenta tiene su propia blockchain individual. Esto permite a los usuarios actualizar su cuenta inmediatamente sin tener que esperar a que toda la red confirme la transaccion.

**Mcedit:** Es una aplicación que funciona en modo texto. La pantalla principal cuenta con dos paneles en los cuales se muestra el sistema de ficheros(Uso similar a shell o interfaz de comandos de Unix).

**Comandos**

**Cat:** Imprime por pantalla el contenido del fichero sin necesidad de modificarlo. Lo que hace es concatenar archivos en la salida estandar.

**More:** Su uso principal es facilitar la lectura y la inspección de archivos largos, como archivos de registro, configuraciones del sistema o resultados de comandos que generan una gran cantidad de salida

**Less:** Es una herramienta esencial para visualizar el contenido de archivos de texto de manera interactiva. less carga el archivo de forma eficiente, sin necesidad de cargarlo completamente en la memoria, lo que lo hace ideal para trabajar con archivos grandes, como registros del sistema. Para usarlo, se ejecuta el comando seguido del nombre del archivo, por ejemplo: less archivo.txt.

**c. Cree un archivo llamado “prueba.exe” en su directorio personal usando el vim. 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?**

Es una herramienta esencial para identificar el tipo de archivo de manera rápida y precisa. Puede determinar si un archivo es un texto plano, binario, imagen, ejecutable, comprimido, o cualquier otro tipo de archivo.

**e. Investigue la funcionalidad y parámetros de los siguientes comandos relacionados con el uso de archivos: cd, mkdir, rmdir, ln, tail, locate, ls, pwd, cp, mv, find**

**cd:** Su funcion es moverse entre directorios.

**mkdir:** Crea carpetas nuevas. Parametros:

* -p crea directorios intermedios si no existen.

**rmdir:** Elimina directorios solo si estan vacios(Si tiene archivos, se usa rm -r).

**ln:** Crea enlaces entre archivos. Parametros:

* ln: Enlace duro
* ln -s: Enlace simbolico(como un acceso directo)

**tail:** Muestra las ultimas lineas de un archivo. Parametros:

* -n 10: Muestra las ultimas 10 lineas
* -f: Muestra en tiempo real lo que se agrega al archivo(muy usado en logs)

**locate:** Busca archivos por nombre en una base de datos indexada.

**ls:** Muestra el contenido de un directorio. Parametros:

* -l: formato largo
* -a: incluye archivos ocultos
* -j: tamaños legibles(mb, kb, etc)

**pwd:** Muestra la ruta completa donde estas parado

**cp:** Copia archivos o directorios. Parametros:

* -r: copia directorios de forma recursiva
* -i: pide confirmacion antes de sobreescribir

**mv**: Mueve o renombra archivos

**find:** Busca archivos en tiempo real dentro del sistema de archivos. Parametros comunes:

* -name: busca por nombre
* -type f: busca solo por archivos
* -type d: busca solo por directorios
* -size +10M: busca archivos mayores a 10 mb

**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 ISOCSO**

mkdir ISOCSO

**b. Acceda a la carpeta**

cd ISOCSO

**c. Cree dos archivos con los nombres isocso.txt e isocso.csv**

touch isocso.txt

touch isocso.csv

**d. Liste el contenido del directorio actual**

ls

**e. Visualizar la ruta donde estoy situado**

pwd

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

find . -name ‘iso\*’

**g. Informar la cantidad de espacio libre en disco**

df -h

df = disk free

-h = human readable

**h. Verifique los usuarios conectados al sistema**

**who**: Muestra los usuarios conectados y desde donde se conectan

**w:** Muestra los mismo que who, pero tambien muestra que hace cada usuario

**users:** Muestra solo los nombre de los usuarios conectado

**i. Editar a el archivo isocso.txt e ingresar Nombre y Apellido**

vi isocso.txt

**j. Mostrar en pantalla las últimas líneas de un archivo**

tail isocso.txt(por defecto muestra las ultimas 10 lineas)

**11. Investigue el funcionamiento, parámetros y ubicación (directorio) de los siguientes comandos: man, shutdown, reboot, halt, uname, dmesg, lspci, at, netstat, head, tail**

**Man:** Es una forma de documentacion de software que se encuentra en Unix. El usuario puede invocar una pagina de documentacion con el comando “man” seguido del nombre del elemento del cual quiere la documentacion.

**Shutdown:** Interrumpe sumintros de energia a los componentes centrales del sistema de foema controlada. Generalmente se encuentra en la ruta /sbin/shutdown. Algunos de sus parametros son:

* **-h:** Apagado predeterminado de linux
* **-r:** Reinicia linux
* **-h 0/now:** Apagado inmediato

**Reboot:** Se utiliza para reiniciar el sistema. Generalmente se encuentra en la ruta /usr/sbin/reboot o /sbin/reboot. Parametros:

* **-r:** Reinicia el sistema

**Halt:** Se utiliza para detener el sistema de manera controlada.

**Uname:** Se utiliza para obtener informacion sobre linux. Parametros:

* **-r:** Informa la version del kernel
* **-v:** Informa la fecha en que fue liberada la version del kernel
* -**m:** Informa la version del sistema operativo
* **-o:** Informa el nombre de la distribucion de linux

**Dmesg:** Es una herramienta que muestra mensajes relacionados con el kernel recuperados desde el buffer de anillo del kernel. Parametros:

* -**H:** Habilita una salida mas legible para los humanos.
* -**T:** Muestra las marcas de tiempo en formato humano.

**lspci:** Es un comando para sistema operativos Unix-like que imprime listas con informacion detallada sobre todos los buses y dispositivos del sistema. Parametros:

* **-m:** Genera un formato simple

**at:** Permite programar tareas unicas en un horario especifico

**netstat:** Muestras las conexiones de red, tablas de enrutamiento y una serie de estadisticas de interfaz de red.

**head:** Es una herramienta de linea de comandos utilizada para mostar las primeras lineas de uno o más archivos de texto. Parametros:

* **-n X:** Muestra las primeras X lineas del archivo.
* **-c X:** Muestras los primeros X bytes
* **-q:** Suprime el encabezado

**tail:** Muestra las ultimas lineas de uno o varios archivos.

* **-n 10:** Muestra las ultimas 10 lineas
* **-f:** Muestra en tiempo real lo que se agrega al archivo(muy usado en logs)

**12. Procesos:**

**a. ¿Qué es un proceso?**

Es un programa en ejecución.

**¿A que hacen referencia las siglas PID y PPID?**

**Pid(process identifier):** Es un identificador numerico asignado a cada proceso en ejecucion por el kernel.

**ppid(parent proccess identifier):** Es el responsable de crear el proceso actual. Atraves del proceso principal se crea un proceso secundario

**¿Todos los procesos tienen estos atributos en GNU/Linux? Justifique. Indique qué otros atributos tiene un proceso.**

Sí, todos los procesos tienen un PID (identificador único) y un PPID (identificador de su proceso padre), excepto el proceso inicial del sistema (init/systemd) que tiene PID 1 y PPID 0.

**b. Investigue el funcionamiento, parámetros y ubicación (directorio) de los siguientes comandos relacionados a procesos. En caso de que algún comando no venga por defecto en la distribución que utiliza deberá proceder a instalarlo:**

**Top:** Es utilizado en servidores par ver la ejecucion en tiempo real. Parametros:

* **-b:** Es util para enviar salidas a archivos u otros programas sin interraccion del usuario.
* **-d:** Establece el delay para actualizar la pantalla.

Ej: top -d 5: La pantalla sera refrescada cada 5 segundos

**htop:** Es un monitor de sistema interactivo, visor y admministrador de procesos diseñados para sistemas tipo Unix, Linux, etc.

* **-d:** Establece el delay en decimas de segundo.
* **-p:** Muestra solo los procesos con id de procesos especificados.

**ps:** Es una herramienta para visualizar el estado de los procesos que se estan ejecutando en el sistema.

**Pstree:** Muestra los procesos que estan siendo ejecutados en una jerarquia de formato arbol.

**Kill:** Es una herramienta para enviar señales a procesos en ejecucion utilizada para enviar señales a procesos en ejecucion. Permitiendo su terminacion controlada o modificacion de su comportamiento.

**Pgrep:** Se utiliza para encontarar los identificadores de proceso de programas en ejecución según criterios especificos.

**pkill:** Se utiliza para enviar señales a los procesos en funcion de diversos criterios como su nombre, usuario, sesion u otros atributos.

**killall:** Se utiliza para terminar todos los procesos que coinciden con un nombre especifico.

**renice:** Se utiliza para alternar la propiedad de programacion de uno o más procesos que ya se esten ejecutando modificando su valor

**xkill:** Es usada para forzar la finalizacion de una aplicación grafica, cerrando la aplicación con el servidor

**atop:** Es un monitor de rendimiento de pantalla completa ASCII para Linux diseñado para proporcionar análisis del rendimiento a nivel de proceso y sistema en tiempo real.

**nice:** Permite modificar la propiedad de un proceso frente al resto del sistema, asignando más o menos tiempo de cpu según sea necesario.

**13. Proceso de Arranque SystemV (https://github.com/systeminit/si):**

**a. Enumere los pasos del proceso de inicio de un sistema GNU/Linux, desde que se prende la PC hasta que se logra obtener el login en el sistema.**

1. **Encendido del equipo**

Se inicia el hardware y se ejecuta el firmware (BIOS/UEFI)

1. **Cargar el gestor de arranque(bootloader)**

El BIOS/UEFI busca el gestor de arranque (como GRUB) en el disco.

3. **Carga del kernel de Linux**

GRUB carga el núcleo (kernel) en memoria y lo ejecuta.

4. **Montaje del sistema de archivos raíz**

El kernel monta el sistema de archivos raíz y configura los dispositivos básicos.

5. **Ejecucion del proceso init**

El kernel lanza el primer proceso de usuario: init, con PID 1.

6. **Lectura de /etc/inittab**

Init consulta este archivo para saber qué runlevel iniciar y qué servicios cargar.

1. **Ejecución de scripts RC según el runlevel**

Se ejecutan los servicios de inicio ubicados en /etc/rcX.d/ (donde X es el runlevel)

1. **Carga de servicios y presentación del login**

Se inician servicios como red, interfaz gráfica, etc., y se muestra el prompt de login

**b. Proceso INIT. ¿Quién lo ejecuta?**

Lo ejecuta el kernel como primer proceso de usuario (PID 1).

**¿Cuál es su objetivo?**

Coordinar el arranque del sistema, iniciar servicios, y gestionar el cambio de runlevels.

**c. RunLevels. ¿Qué son?**

Son modos de operación predefinidos que determinan qué servicios se inician

**¿Cuál es su objetivo?**

Permitir configurar el sistema para distintos estados: mantenimiento, multiusuario, gráfico, etc.

**d. ¿A qué hace referencia cada nivel de ejecución según el estándar?**

|  |  |
| --- | --- |
| **Runlevel** | **Descripcion** |
| 0 | Apaga el sistema |
| 1 | Modo monousuario(mantenimiento) |
| 2 | Multiusuario sin red |
| 3 | Multiusuario con red |
| 4 | No utilizado(reservado para personalizacion) |
| 5 | Multiusuario con GUI(Interfza gráfica) |
| 6 | Reinicio del sistema |

**¿Dónde se define qué Runlevel ejecutar al iniciar el sistema operativo?**

En el archivo /etc/inittab, con una línea como:

id:5:initdefault:

**¿Todas las distribuciones respetan estos estándares?**

No todas. Algunas como Debian y Red Hat los respetan, pero otras han migrado a systemd, que usa "targets" en lugar de runlevels.

**e. Archivo /etc/inittab. ¿Cuál es su finalidad?**

Configurar el comportamiento del proceso init

**¿Qué tipo de información se almacena en el?**

* Runlevel por defecto
* Acciones a ejecutar en cada runlevel
* Configuracion de terminales(tty)

**¿Cuál es la estructura de la información que en él se almacena?**

Estructura tipica: **<id>:<runlevel>:<action>:<process>**(formato)

id:5:initdefault:

si::sysinit:/etc/init.d/rcS

l0:0:wait:/etc/init.d/rc 0

l1:1:wait:/etc/init.d/rc 1

**f. Suponga que se encuentra en el runlevel . Indique qué comando(s) deberá ejecutar para cambiar al runlevel .**

**¿Este cambio es permanente? ¿Por qué?**

No es permante. El cambio dura hasta el proximo reinicio. Para hacerlo permanente hay que modificar el runlevel por defecto en /etc/inittab

**g. Scripts RC. ¿Cuál es su finalidad?**

Iniciar po detener servicios según el runlevel.

**¿Dónde se almacenan?**

* **scrip base:** /etc/init.d
* **Enlaces simbolicos a los scrips:** /etc/ecX.d

**Cuando un sistema GNU/Linux arranca o se detiene se ejecutan scripts, indique cómo determina qué script ejecutar ante cada acción.**

Se ejecutan los scripts en que comienzan con:

• (Start) → para iniciar servicios

• (Kill) → para detener servicios

**¿Existe un orden para llamarlos? Justifique.**

Si, los scrips se ejecutan en orden numérico. Esto asegura que los servicios dependientes se ejecuten en el orden correcto.

**14. SystemD (https://github.com/systemd/systemd):**

**a. ¿Qué es systemd?**

Es un conjunto de deamons de administracion de sistemas bibliotecas y herramientas diseñadas como una plataforma de administracion y configuracion central para interactuar con el kernel.

**b. ¿A qué hace referencia el concepto de Unit en SystemD?**

Es la representacion de un recurso o servicio de SystemD que puede manejar. Cada Unit se define mediante un archivo con extension “.sevice”, “.target”, “.mount”, etc, que describe como debe comportarse.

Ejemplo simplificado: Es como una “pieza” del sistema que se administra de manera uniforme: iniciar, detener, reiniciar, monitorear, etc.

**c. ¿Para qué sirve el comando systemctl en SystemD?**

Sirve para administrar y consultar el estado de las *Units* de systemd: servicios, dispositivos, sockets, puntos de montaje, etc.

Con systemd podes:

* Iniciar, detener y reiniciar servicios.
* Habilitar o deshabilitar servicios al arranque.
* Ver el estado de los servicios y de todo el sistema.
* Listar y controlar targets (niveles de ejecución modernos).

**d. ¿A qué hace referencia el concepto de target en SystemD?**

Una Target Unit es un tipo especial de *unit* (archivos terminados en .target).

Sirve para agrupar y organizar otras units con un objetivo en comun.

**e. Ejecutar el comando pstree. ¿Qué es lo que se puede observar a partir de la ejecución de este comando?**

Cuando se ejecuta el comando pstree lo primero que se observa on los procesos en ejecucion en forma de arbol.

**15. Usuarios:**

**a. ¿Qué archivos son utilizados en un sistema GNU/Linux para guardar la información de los usuarios?**

En un sistema GNU/Linux, la información de los usuarios se almacena en varios archivos clave, principalmente ubicados en el directorio /etc. El archivo *passwd* contiene la base de datos del usuario del sistema, compuesta por siete campos separados por dos puntos que proporcionan información sobre cada usuario.

El directorio /home es el lugar donde se crean automáticamente los directorios personales de los usuarios, y cada usuario tiene su propio subdirectorio dentro de /home donde puede almacenar sus archivos personales y sus archivos de configuración.

**b. ¿A qué hacen referencia las siglas UID y GID?**

**UID (User Identifier):** Es el número único que identifica a un usuario dentro del sistema.

**GID (Group Identifier)**: Es el número único que identifica a un grupo.

**Entonces** cada usuario tiene un UID, y pertenece a uno o varios grupos identificados por GID.

**¿Pueden coexistir UIDs iguales en un sistema GNU/Linux? Justifique.**

Es posible que dos usuarios compartan el mismo UID (editando /etc/passwd), pero:

* Ambos usuarios serían prácticamente “el mismo” para el sistema en cuanto a permisos.
* Se pierde el concepto de identidad única.
* Es considerado mala práctica y un riesgo de seguridad.

**c. ¿Qué es el usuario root?**

Es el superusuario en GNU/Linux ya que cuenta con todos los privilegios. Es el equivalente al administrador de windows.

**¿Puede existir más de un usuario con este perfil en GNU/Linux?**

Puede haber varios usuarios con privilegios de root, pero solo uno tiene UID 0, que es el verdadero *root*.

Lo que suele hacerse es agregar usuarios normales al grupo sudo y estos tienen los privilegios de root.

**¿Cuál es la UID de root?**

Siempre es 0, es la forma de que el sistema sabe que se trata del supersuario.

**d. Agregue un nuevo usuario llamado isocso a su instalación de GNU/Linux, especifique que su home sea creada en /home/isocso, y hágalo miembro del grupo informatica (si no existe, deberá crearlo). Luego, sin iniciar sesión como este usuario cree un archivo en su home personal que le pertenezca. Luego de todo esto, borre el usuario y verifique que no queden registros de él en los archivos de información de los usuarios y grupos.**

**Agregra usuario:**

1. Verificar si existe el grupo

* getent group informatica(Si no aparece nada es que no existe)
* sudo groupadd informatica(Crea el grupo)

1. Crear usuario:

**sudo useradd -m -d /home/isocso -s /bin/bash isocso**

* -m: Crear el directorio *home* si no existe
* -d: especifica donde estara el home
* -s: Define la shell por defecto

1. Asignar usuario al grupo informatica

**sudo usermod -aG informatica isocso**

* -aG: agrega el usuario a un grupo secundario sin sacarlo de los que ya pertenece.

1. Asignar contraseña al usuario

Sudo passwd isocso

1. Verificar que se agrego

**Id isocso**(si lo agrego muestra algo como esto)

uid=1001(isocso) gid=1001(isocso) groups=1001(isocso),1002(informatica)()

**Eliminar usuario:**

1. **Crear archivo en home del usuario sin iniciar sesion**

sudo touch /home/isocso/archivo.txt

sudo chown isocso:informatica /home/isocso/archivo.txt

1. **Eliminar usuario**

sudo deluser --remove-home isocso

1. **Verificar que no queden registros**

grep isocso /etc/passwd(Verifica que exista el usuario)

grep isocso /etc/group(Verifica que exista el grupo)

**nota:** Si no devuelve nada, significa que no existen

**e. Investigue la funcionalidad y parámetros de los siguientes comandos:**

**useradd:** Se utiliza para crear usuarios, se debe especificar opciones que definan varios aspectos de la cuenta, como nombre de usuario, UID, GID y directorio único. Parametros:

* **-m** Crea el directorio de inicio del usuario si no existe.
* **-d** Especifica la ruta del directorio de inicio del usuario, por ejemplo, /home/usuario.
* **-g** Asigna al usuario un grupo primario, que debe existir previamente.
* **-U** Crea un grupo con el mismo nombre que el usuario y añade el usuario a ese grupo.

**adduser:** Se utiliza para crear usuarios, automatiza ciertos pasos y ofrece una interaccion mas amigable.Parametros:

* **--home** DIRECTORIO: Permite especificar un directorio personal diferente al predeterminado, creándolo si no existe y copiando los archivos del directorio skel.
* **--shell** CONSOLA: Define la shell de inicio de sesión del usuario, en lugar de la predeterminada (/bin/bash).
* **--uid** ID: Asigna un identificador de usuario (UID) específico al nuevo usuario, aunque el comando fallará si ese UID ya está en uso.
* **--gid** ID o **--ingroup** GRUPO: Añade el nuevo usuario a un grupo específico, ya sea por su ID (--gid) o nombre (--ingroup), afectando así al grupo principal del usuario.

**usermod:** Se utiliza para modificar las propiedades de una cuenta de usuario existente. Parametros:

* **-c** "comentario" para agregar una descripción o comentario sobre el usuario, normalmente almacenado en el campo de comentarios del archivo /etc/passwd.
* **-d** /ruta/nueva para cambiar el directorio de inicio del usuario. Si se usa la opción -m, el contenido del directorio anterior se mueve al nuevo directorio.
* **-g** grupo para cambiar el grupo primario del usuario.

**userdel:** Se utiliza para eliminar una cuenta de usuario del sistema, modificando los archivos del sistema como /etc/passwd y /etc/shadow para eliminar las entradas relacionadas con el nombre de inicio de sesión del usuario. Parametros:

* **-f** Fuerza la eliminación de la cuenta de usuario, incluso si el usuario está conectado.
* **-r** Elimina el directorio personal (home) del usuario junto con su cuenta. Es útil para una limpieza completa del sistema.
* **-h** Muestra un mensaje de ayuda con la sintaxis general y las opciones disponibles.

**su:** Se utiliza para cambiar al usuario especificado, generalmente para obtener privilegios de superusuario (root) o acceder a la cuenta de otro usuario. Al ejecutar su sin ningún parámetro, el sistema solicita la contraseña del usuario root para iniciar una sesión con sus privilegios.

**groupadd:** Se utiliza para crear nuevos grupos de usuarios y ofrece varias opciones para personalizar la creación del grupo. Parametros:

* **-g GID o --gid GID:** permite especificar un ID numérico (GID) para el nuevo grupo.
* **-f o --force:** Hace que el comando finalice con éxito si el grupo ya existe.
* **-o o --non-unique:** Permite crear un grupo con un GID ya utilizado, lo que significa que el mapeo entre el GID y el nombre del grupo no será único.

**who:** Se utiliza para mostrar información sobre los usuarios actualmente conectados al sistema, incluyendo sus nombres de inicio de sesión, la terminal que están utilizando, la hora de conexión y la dirección IP desde la que accedieron. Parametros:

* **-H:** Muestra encabezados de columna en la salida para facilitar la lectura.
* **-q:** Muestra únicamente una lista resumida de los nombres de usuario conectados y el número total de usuarios activos.
* **-b:** Muestra la fecha y hora del último arranque del sistema, lo que permite determinar cuánto tiempo lleva funcionando sin reiniciarse.
* **-m:**Muestra información sobre el usuario actual.

**groupdel:** Se utiliza para eliminar un grupo del sistema.

**passwd:** Es una herramienta fundamental para la gestión de contraseñas de usuarios, permitiendo cambiar contraseñas, establecer políticas de caducidad y bloquear o desbloquear cuentas. Parametros:

* **-S:** Muestra estado de la contraseña
* **-l:** Bloquea la contraseña de un usuario añadiendo un "!" al principio de la entrada en el archivo de contraseñas, lo que impide que el usuario inicie sesión con contraseña.
* **-u:** Desbloquea una contraseña previamente bloqueada, restaurando su valor original.
* **-d:** Elimina la contraseña de un usuario, haciendo que la cuenta sea sin contraseña, lo que puede usarse para desactivar el acceso por contraseña.

**16. FileSystem y permisos:**

**a. ¿Cómo son definidos los permisos sobre archivos en un sistema GNU/Linux?**

Los permisos sobre archivos en un sistema GNU/Linux están definidos mediante un sistema de control de acceso discrecional (DAC) que asigna permisos de lectura (r), escritura (w) y ejecución (x) a tres clases de usuarios: el propietario del archivo (usuario), los miembros del grupo al que pertenece el archivo (grupo) y todos los demás usuarios (otros). Cada clase de usuario puede tener una combinación de estos tres permisos, representados por letras en una notación simbólica de 10 caracteres, donde el primer carácter indica el tipo de archivo (por ejemplo, - para archivo normal, d para directorio). Los permisos se agrupan en tres conjuntos de tres caracteres cada uno: el primero para el propietario, el segundo para el grupo y el tercero para otros.

**b. Investigue la funcionalidad y parámetros de los siguientes comandos relacionados con los permisos en GNU/Linux:**

**chmod:** Se utiliza para cambiar los permisos de acceso a archivos y directorios, permitiendo definir quién puede leer, escribir o ejecutar un archivo. Parametros

* **-R:** Aplica los permisos de forma recursiva a todos los archivos y subdirectorios dentro de un directorio.
* **-v:** Muestra un mensaje de diagnóstico por cada archivo procesado.
* **-f:** Suprime la mayoría de mensajes de error.

**chown:** Se utiliza para cambiar el propietario y/o el grupo de uno o varios archivos o directorios.Parametros:

* **-c:** Muestra solo los cambios realizados.
* **-v:** Muestra un mensaje diagnostico por cada archivo procesado.
* **-R:** Se utiliza para cambiar la propiedad de forma recursiva en todos los archivos y directorios dentro de un directorio especificado.

**chgrp:** Se utiliza para cambiar el grupo propietario de archivos o directorios. Parametros:

* **-R:** Cambia el grupo de forma recursiva en directorios y sus contenidos.
* **-v:** Proporciona información detallada sobre los archivos que se están procesando, mostrando el cambio de grupo.
* **-h:** Actúa sobre el enlace simbólico en sí, no sobre el archivo al que apunta. Esto es útil cuando se desea cambiar el grupo del enlace en lugar del destino.

**Nota:** El comando *chown* se utilza para cambiar el propietario y opcionalemete el grupo. En cambio *chgrp* solo cambia el grupo al que pertenece un archivo o directorio. Aunque *chown* puede realizar la función de *chgrp* al cambiar el grupo, *chgrp* es más específico y se prefiere cuando solo se necesita cambiar el grupo, ya que reduce el riesgo de errores al escribir el comando, especialmente en entornos de producción.

**c. Al utilizar el comando chmod generalmente se utiliza una notación octal asociada para definir permisos. ¿Qué significa esto? ¿A qué hace referencia cada valor?**

La notación octal en chmod es una forma numérica de definir permisos. Cada permiso se representa como un número (lectura = 4, escritura = 2, ejecución = 1), y la suma de estos valores define los permisos para el **usuario, grupo y otros**, en ese orden.

|  |  |  |
| --- | --- | --- |
| **Valor** | **Permisos** | **Significado** |
| 0 | --- | Sin permisos |
| 1 | --x | Solo ejecucion |
| 2 | -w- | Solo escritura |
| 3 | -wx | Escritura + ejecucion |
| 4 | r-- | Solo lectura |
| 5 | r-x | Lectura + ejecucion |
| 6 | rw- | Lectura + escritura |
| 7 | rwx | Lectura + escritura + ejecucion |

**d. ¿Existe la posibilidad de que algún usuario del sistema pueda acceder a determinado archivo para el cual no posee permisos? Indiquelo y realice las pruebas correspondientes.**

En condiciones normales, un usuario solo puede acceder a un archivo si tiene permisos asignados por el dueño o el grupo. Sin embargo, el superusuario root siempre puede acceder a todo.

**e. Explique los conceptos de “full path name” (path absoluto) y “relative path name” (path relativo). De ejemplos claros de cada uno de ellos.**

**Full path name:** Es la ruta complera desde la raíz / hasta el archivo o directorio.

Ejemplo: /home/user/documentos/nota.txt

* **/** raiz del sistema
* **home** carpeta de usuarios
* **user** tu usuario
* **doumentos/nota.txt** la carpeta y el archivo

**Relative path name:** Es la ruta partiendo desde la carpeta en la que te encontras actualmente. Ejemplo:

Ejemplo: /home/user

* cd documentos
* cat nota.txt

Aca uso path relativo porque parto desde el directorio actual y voy bajando.

**f. ¿Con qué comando puede determinar en qué directorio se encuentra actualmente?**

Con el comando **pwd,** muestra la rutaabsoluta del directorio actual

**¿Existe alguna forma de ingresar a su directorio personal sin necesidad de escribir todo el path completo?**

Para ir al directorio personal sin escribir todo el path se puede usar cd o cd ~

**¿Podría utilizar la misma idea para acceder a otros directorios?**

Con el comando **cd** puedo acceder a cualquier directorio del sistema.

**¿Cómo? Explique con un ejemplo.**

Ingresando el comando **cd /etc** puedo ir a /etc sin importar en el directorio en el que estaba antes.

**g. Investigue la funcionalidad y parámetros de los siguientes comandos relacionados con el uso del FileSystem:**

**umount:** Se utiliza para desmontar sistemas de archivos. Parametros:

* **-a o -all**: Desmonta todos los sistemas de archivos descritos en /proc/self/mountinfo (o en el antiguo /etc/mtab), excepto los sistemas de archivos proc, devfs, devpts, sysfs, rpc\_pipefs y nfsd.
* **-A:** Desmonta todos los puntos de montaje en el espacio de nombres de montaje actual para el sistema de archivos especificado, que puede identificarse por un punto de montaje, nombre de dispositivo.
* **-c:** Evita la normalización de rutas, útil para evitar bloqueos en sistemas NFS cuando el servidor no está disponible.
* **-d:** Cuando el dispositivo desmontado es un dispositivo de bucle, también libera el dispositivo de bucle asociado.
* **-f:** Fuerza el desmontaje, especialmente útil en sistemas NFS inaccesibles; sin embargo, no garantiza que el comando no se bloquee.

**du:** Se utiliza para estimar el espacio en disco ocupado por archivos y directorios. Parametros:

* **-a:** Muestra el uso de espacio de cada archivo individual, no solo de los directorios.
* **-h:** Muestra los tamaños en un formato legible por humanos, como KB, MB o GB, facilitando la interpretación de los resultados.
* **-s:** Informa solo el tamaño total del directorio especificado, sin mostrar los subdirectorios.

**df:** Informa acerca del espacio total, ocupado y libre en nuestro sistema. Parametros:

* -**i:** Informa inodes ocupados y libres en cada particion.
* **-h** Muestra los tamaños en un formato legible por humanos.

**mount:**Se utiliza para montar sistemas de archivos, es decir, para hacer accesibles los contenidos de un dispositivo (como un disco duro, USB, DVD, o una imagen ISO) dentro de la jerarquía de archivos del sistema operativo, asociándolos a un directorio específico llamado punto de montaje. Parametros:

* **-t:** Especifica el tipo de sistema de archivos
* **-o:** Permite pasar opciones adicionales, como ro para montar en solo lectura, rw para lectura/escritura, loop para montar una imagen como un dispositivo, etc.
* **-r:** Monta el sistema de archivos en modo solo lectura
* **-w:** Monta el sistema de archivos en modo lectura/escritura(opcion por defecto).

**mkfs:**Se utiliza para formatear un dispositivo de almacenamiento de bloque con un determinado sistema de archivos, creando una nueva estructura de datos que permite al sistema operativo almacenar, acceder, editar y eliminar archivos. Parametros

* -**v:** para producir una salida detallada durante el proceso de formateo.
* -**c:** para comprobar los bloques defectuosos del dispositivo antes de crear el sistema de archivos.
* **-l**: nombre\_de\_archivo: para leer una lista de bloques defectuosos desde un archivo especificado.
* -**m:** para reducir el espacio libre reservado para tareas de recuperación (por defecto es 5%).

**fdisk:** Es una herramienta esencial para la gestión de particiones en sistemas Linux, permitiendo crear, modificar, eliminar y listar particiones en discos duros.

**write:** Permite la comunicación en tiempo real entre usuarios conectados al mismo sistema enviando mensajes directamente a sus sesiones de terminal.

**Losetup:**  Se utiliza para gestionar dispositivos de bucle, que son pseudo-dispositivos que permiten tratar un archivo regular como un dispositivo de bloque, como si fuera un disco duro y CD-ROM. Parametros:

* **-o**: Comienza la lectura desde una posición específica dentro del archivo.
* **-p:** Fuerza el kernel a escanear la tabla de particiones del dispositivo.

**stat:**Se utiliza para mostrar información detallada sobre archivos y sistemas de archivos, proporcionando datos que van más allá de los que ofrece el comando ls -l, como marcas de tiempo, permisos, tamaño, número de inodo y otros atributos extraídos directamente del inodo del archivo. Parametros:

* -**f:** Permite ver detalles como el tamaño total de bloques, los bloques libres y disponibles.
* **-L:** Obtiene los datos del aarchivo al que apunta.
* **-format:** Permite especificar secuencias de formato, para mostrar solo los datos deseados.

**17. Procesos:**

**a. ¿Qué significa que un proceso se está ejecutando en Background?**

Quiere decir que se esta ejecutando en segundo plano, es util para tareas que tardan mucho tiempo o que no requieren interaccion directa

**¿Y en Foreground?**

Se ejecuta directamente en la shell y no se pueden ingresar comando hasta que finalice el proceso.

**b. ¿Cómo puedo hacer para ejecutar un proceso en Background?**

Simplemente hay que agregar el carácter “&” al final del comando que desaes ejecutar.

**¿Como puedo hacer para pasar un proceso de background a foreground y viceversa?**

**Background a foreground:** Utilizando el comando “fg”, este comando recupera el proceso suspendido o en segundo plano y lo lleva al primer plano**.**

**Foreground a background:**Si el proceso ya está en ejecución en primer plano, puedes detenerlo temporalmente y enviarlo al fondo usando la combinación de teclas Ctrl+Z.

**c. Pipe ( | ). ¿Cuál es su finalidad? Cite ejemplos de su utilización.**

Su función principal es redirigir la salida estándar (stdout) de un comando hacia la entrada estándar (stdin) de otro comando, permitiendo así la creación de una conexión directa entre procesos.

Ejemplos: **cat ejemplo.txt | wc -l**

Este comando primero ejecuta cat ejemplo.txt, que imprime el contenido del archivo, y luego ese contenido se envía como entrada al comando wc -l, que devuelve el número total de líneas.

**d. Redirección. ¿Qué tipo de redirecciones existen?**

entrada estándar (STDIN, tipo 0), salida estándar (STDOUT, tipo 1) y error estándar (STDERR, tipo 2).

**¿Cuál es su finalidad? Cite ejemplos de utilización.**

Las redirecciones permiten cambiar el destino de la entrada y salida de los comandos. Se pueden usar para guardar resultados en archivos (ls > lista.txt), agregar datos (>>), leer desde un archivo (sort < datos.txt), capturar errores (2> errores.txt), o encadenar comandos con pipes (ls | grep ".txt").

**18. Otros comandos de Linux (Indique funcionalidad y parámetros): a. ¿A qué hace referencia el concepto de empaquetar archivos en GNU/Linux?**

El concepto de empaquetar archivos en GNU/Linux se refiere a la acción de agrupar varios archivos y/o directorios en un solo fichero contenedor, sin realizar ninguna compresión de su tamaño**.**

**b. Seleccione 4 archivos dentro de algún directorio al que tenga permiso y sume el tamaño de cada uno de estos archivos. Cree un archivo empaquetado conteniendo estos 4 archivos y compare los tamaños de los mismos. ¿Qué característica nota?**

* **Creo los 4 archivos**(opcional)

echo "Archivo 1" > a1.txt

echo "Archivo 2" > a2.txt

echo "Archivo 3" > a3.txt

echo "Archivo 4" > a4.txt

* **Ver tamaño de cada archivo**

du -ch a1.txt a2.txt a3.txt a4.txt

* **Creo archivo empaquetado**

tar -cvf empaquetado.tar a1.txt a2.txt a3.txt a4.txt

* **Ver tamaño de archivo empaquetado**

ls -lh empaquetado.tar

**nota:** La suma de los 4 archivos por separado es de 16 kb, mientras que el archivo empaquetado pesa 10k

**c. ¿Qué acciones debe llevar a cabo para comprimir 4 archivos en uno solo? Indique la secuencia de comandos ejecutados.**

tar -cvzf comprimido.tar.gz a1.txt a2.txt a3.txt a4.txt

**Nota:** Este comando empaqueta y comprime los 4 archivos.

**d. ¿Pueden comprimirse un conjunto de archivos utilizando un único comando?**

Si, para comprimir todo en un solo comando utilizas este comando:

tar -cvzf comprimido.tar.gz a1.txt a2.txt a3.txt a4.txt

**e. Investigue la funcionalidad de los siguientes comandos:**

**tar:** Es una herramienta fundamental para empaquetar, comprimir y descomprimir archivos y directorios.

**grep:** Es una herramienta fundamental para buscar patrones de texto o cadenas específicas dentro de archivos o en la entrada estándar.

**gzip:** Se encargar de de comprimir en formato .gz

**zgrep:** Se utiliza para buscar patrones dentro de archivos sin la necesidad de descomprimirlos.

**wc:** Es fundamental para analizar archivos de texto, contando lineas, palabras, caracteres y bytes.

**19. Indique qué acción realiza cada uno de los comandos indicados a continuación considerando su orden. Suponga que se ejecutan desde un usuario que no es root ni pertenece al grupo de root. (Asuma que se encuentra posicionado en el directorio de trabajo del usuario con el que se logueó). En caso de no poder ejecutarse el comando, indique la razón:**

**l s −l > prueba**

* No se puede ejecutar el comando, porque “ls” debe ir junto

**ps > PRUEBA**

* Ejecuta ps (lista procesos del usuario/terminal por defecto) y guarda la salida en PRUEBA.

**chmod 710 prueba**

**chown root:root PRUEBA**

* Operación no permitida como usuario normal

**chmod 777 PRUEBA**

* Pone permisos rwxrwxrwx en PRUEBA.

**chmod 700 /etc/passwd**

* Operación no permitida como usuario normal

**passwd root**

* Operación no permitida, *passwd* no debe ver o cambiar información de la contraseña para root

**rm PRUEBA**

* borra el archivo PRUEBA. Solo la puede ejecutar el dueño del diectorio o el superusuario.

**man /etc/shadow find / −name ∗ .conf**

* pide la página de manual para "/etc/shadow".

**usermod root −d /home/ newroot −L**

* Operación no permitida como usuario normal

**cd / root**

* Permiso denegado

**rm ∗**

* borra todos los archivos (y con rm -r también directorios) que coincidan con \* en el directorio actual.

**cd / etc**

* cambia al directorio /etc.

**cp ∗ /home −R**

* Permiso denegado

**Shutdown**

* Orden no encotrada

**20. Indique qué comando sería necesario ejecutar para realizar cada una de las siguientes acciones:**

**a. Terminar el proceso con PID 23.**

Kill 23

Esto envía la señal [**SIGTERM**](#sigterm) **(15)** al proceso con PID 23.

Tambien se puede usar: kill -9 23(Fuerza la finalización inmediata).

**b. Terminar el proceso llamado init o systemd. ¿Qué resultados obtuvo?**

Utilizo el comando *killall systemd,* al finalizar este proceso se cierra la sesión.

**c. Buscar todos los archivos de usuarios en los que su nombre contiene la cadena “.conf”**

Comando utilizado *find /home -name "\*.conf"*

**d. Guardar una lista de procesos en ejecución el archivo /home//procesos**

ps aux > /home/<nombre de usuario>/procesos

**e. Cambiar los permisos del archivo /home//xxxx a:**

* **Usuario: Lectura, escritura, ejecución** (7)
* **Grupo: Lectura, ejecución** (5)
* **Otros: ejecución** (1)

**Comando:** chmod751 /home/<nombre de usuario>/xxxx

**Nota:** Primero crear archivo “xxxx”

**f. Cambiar los permisos del archivo /home//yyyy a:**

**i. Usuario: Lectura, escritura.** (6)

**ii. Grupo: Lectura, ejecución** (5)

**iii. Otros: Ninguno** (0)

**Comando:** chmod 650 /home/<nombre de usuario>/yyyy

**Nota:** Primero crear archivo “yyyy”

**g. Borrar todos los archivos del directorio /tmp**

**Comando:** sudo rm -rf /tmp/\*

**Nota:** El usuario debe ser parte de los sudoers

**h. Cambiar el propietario del archivo /opt/isodata al usuario isocso i. Guardar en el archivo /home//donde el directorio donde me encuentro en este momento, en caso de que el archivo exista no se debe eliminar su contenido anterior.**

**21. Indique qué comando sería necesario ejecutar para realizar cada una de las siguientes acciones:**

**a. Ingrese al sistema como usuario “root”**

su -

**b. Cree un usuario. Elija como nombre, por convención, la primera letra de su nombre seguida de su apellido. Asígnele una contraseña de acceso.**

sudo useradd -m gsayago

sudo passwd gsayago

**nota: -**m → crea el usuario y su directorio personal /home/gsayago

**c. ¿Qué archivos fueron modificados luego de crear el usuario y qué directorios se crearon?**

Cuando creás un usuario, el sistema modifica:

* /etc/passwd → agrega la entrada con el nuevo usuario.
* /etc/shadow → guarda la contraseña (encriptada).
* /etc/group → puede actualizarse si el usuario tiene grupo propio.
* /etc/gshadow → similar para contraseñas de grupos.

Y se crea el directorio:

* /home/gsayago/ → el home del nuevo usuario.

**d. Crear un directorio en /tmp llamado miCursada**

mkdir /tmp/miCursada

**e. Copiar todos los archivos de /var/log al directorio antes creado.**

cp -r /var/log/\* /tmp/miCursada/

**Nota:** Debe realizarlo el usuario “root”

**f. Para el directorio antes creado (y los archivos y subdirectorios contenidos en él) cambiar el propietario y grupo al usuario creado y grupo users.**

chown -R gsayago:users /tmp/miCursada

**Nota:** Debe realizarlo el usuario “root”

**g. Agregue permiso total al dueño, de escritura al grupo y escritura y ejecución a todos los demás usuarios para todos los archivos dentro de un directorio en forma recursiva.**

chmod -R 732 /tmp/miCursada

**Nota:** Debe realizarlo el usuario “root”

**h. Acceda a otra terminal para loguearse con el usuario antes creado.**

su - gsayago

**i. Una vez logueado con el usuario antes creado, averigüe cuál es el nombre de su terminal.**

tty

**j. Verifique la cantidad de procesos activos que hay en el sistema.**

ps -e | wc -l

* ps -e → lista todos los procesos.
* wc -l → cuenta cuántas líneas.

**k. Verifiqué la cantidad de usuarios conectados al sistema.**

who

**l. Vuelva a la terminal del usuario root y envíele un mensaje al usuario anteriormente creado enviándole que el sistema va a ser apagado.**

echo "El sistema va a ser apagado" | sudo wall

**Nota:** “sudo wall” manda mensaje a todos los usuarios, para mandar a un usuario usar “write <nombre de usuario> pts/1”

**m. Apague el sistema.**

systemctl poweroff

**22. Indique qué comando sería necesario ejecutar para realizar cada una de las siguientes acciones:**

**n. Cree un directorio cuyo nombre sea su número de legajo e ingrese a él.**

mkdir 227288

**o. Cree un archivo utilizando el editor de textos vi, e introduzca su información personal: Nombre, Apellido, Número de alumno y dirección de correo electrónico. El archivo debe llamarse "LEAME".**

vi LEAME

**p. Cambie los permisos del archivo LEAME, de manera que se puedan ver reflejados los siguientes permisos:**

**➢ Dueño: ningún permiso**

**➢ Grupo: permiso de ejecución**

**➢ Otros: todos los permisos**

chmod 017 LEAME

**q. Vaya al directorio /etc y verifique su contenido. Cree un archivo dentro de su directorio personal cuyo nombre sea leame donde el contenido del mismo sea el listado de todos los archivos y directorios contenidos en /etc.**

ls -l /etc > ~/leame

**Explicación:**

* **ls -l /etc** → lista el contenido de /etc (archivos y directorios con detalles).
* **>** → redirige la salida de ese listado.
* **~/leame** → crea (o sobreescribe) un archivo llamado leame dentro de tu directorio personal (~ es un atajo a /home/<usuario>).

**¿Cuál es la razón por la cuál puede crear este archivo si ya existe un archivo llamado "LEAME” en este directorio?**

Porque en GNU/Linux los nombres de archivo son *case-sensitive* (distinguen mayúsculas de minúsculas).

**r. ¿Qué comando utilizaría y de qué manera si tuviera que localizar un archivo dentro del filesystem?**

find / -name "LEAME" 2>/dev/null

**¿Y si tuviera que localizar varios archivos con características similares?**

find /etc -name "\*.conf"

**Explique el concepto teórico y ejemplifique.**

Para buscar archivos se usan herramientas como find, lo que hace es buscar en tiempo real dentro del sistema de archivos.

**s. Utilizando los conceptos aprendidos en el punto anterior, busque todos los archivos cuya extensión sea .so y almacene el resultado de esta búsqueda en un archivo dentro del directorio creado en el primer inciso. El archivo deberá llamarse ejercicioF.**

find / -name "\*.so" 2>/dev/null > /tmp/miCursada/ejercicioF

**Explicación:**

* **find / -name "\*.so"** → busca todos los archivos con extensión .so (bibliotecas compartidas).
* **2>/dev/null** → descarta los errores de “Permiso denegado”.
* **>** → redirección para guardar el resultado en el archivo /tmp/miCursada/ejercicioF.

**23. Indique qué acción realiza cada uno de los comandos indicados a continuación considerando su orden. Suponga que se ejecutan desde un usuario que no es root ni pertenece al grupo de root. (Asuma que se encuentra posicionado en el directorio de trabajo del usuario con el que se logueó). En caso de no poder ejecutarse el comando indique la razón:**

**Comando para verificar usuarios sudoers:** sudo cat /etc/sudoers

**01. mkdir iso**

Crea el directorio “ISO”

**02. cd . / iso; ps > f0**

Está mal escrito, hay un espacio de más en . / iso.

**Lo correcto sería:** cd ./iso; ps > f0

* **cd ./iso** → entra al directorio iso que está en el directorio actual (.).
* **ps > f0** → lista los procesos en ejecución y guarda la salida en un archivo llamado f0.

**03. ls > f1**

Lista el contenido del directorio actual y guarda el resultado en el archivo f1 (sobrescribiéndolo si ya existe).

**04. cd /**

Cambia al directorio raíz (/).

**05. echo $HOME**

Muestra el valor de la variable de entorno $HOME, que corresponde al directorio personal del usuario (ejemplo: /home/gonzalo).

**06. ls −l $> $HOME/ iso/ls**

* $> no existe como operador válido.
* Hay un espacio en / iso/ls.

**07. cd $HOME; mkdir f2**

* cd $HOME → se mueve al directorio personal del usuario.
* mkdir f2 → crea un directorio llamado f2 ahí dentro.

**08. ls −ld f2**

Muestra los detalles del directorio f2 sin listar su contenido (útil para ver permisos).

**09. chmod 341 f2**

Cambia los permisos de f2 (debería ser un directorio o archivo).

En octal 341 significa:

* Dueño (3): escritura + ejecución (-wx).
* Grupo (4): solo lectura (r--).
* Otros (1): solo ejecución (--x).

**10. touch dir**

Crea el archivo “dir”

**11. cd f2**

Cambia al directorio f2

**12. cd ~/iso**

Entra al directorio iso dentro de home (“~/iso” debería existir)

**13. pwd > f3**

Guarda la ruta completa del directorio actual (pwd) en un archivo llamado f3.

**14. ps | grep 'ps' | wc −l >> ../f2/f3**

**Explicación:**

* **ps** → lista procesos.
* **grep 'ps'** → filtra los que contienen la palabra ps.
* **wc -l** → cuenta cuántas líneas hay (cuántos procesos).
* **>> ../f2/f3** → agrega ese número al archivo f3 dentro de ../f2.

**15. chmod 700 ../f2 ; cd ..**

* **cambia permisos del directorio f2 a rwx------ (solo el dueño puede acceder).**
* **cd .. → sube un nivel en el árbol de directorios.**

**16. find . −name etc/passwd**

* **-name etc/passwd** busca un archivo que se llame exactamente etc/passwd, lo cual no es lo usual.
* **Comando corregido:** find . -name "passwd"

**17. find / −name etc/passwd**

**Comando corregido:** find / -name "passwd" 2>/dev/null

**18. mkdir ejercicio5**

Crea un directorio llamado ejercicio5 en el directorio actual.

**b. Complete los comandos 19 y 20, de manera tal que realicen la siguiente acción:**

**19: Copiar el directorio iso y todo su contenido al directorio creado en 24.a**

ls -l iso > explicacion\_de\_ejercicio

**24. Cree una estructura desde el directorio /home que incluya varios directorios, subdirectorios y archivos, según el esquema siguiente.**

**Gráfico, Gráfico radial

El contenido generado por IA puede ser incorrecto.**

**Asuma que “usuario” indica cuál es su nombre de usuario. Además deberá tener en cuenta que dirX hace referencia a directorios y fX hace referencia a archivos. Utilizando la estructura de directorios anteriormente creada, indique qué comandos son necesarios para realizar las siguientes acciones:**

**a. Mueva el archivo "f3” al directorio de trabajo /home/usuario.**

sudo mv /home/gonzalo/dir1/f3 /home/gonzalo

**b. Copie el archivo "f4” en el directorio "dir11".**

sudo cp /home/gonzalo/dir2/f4 /home/gonzalo/dir1/dir11

**c. Haga los mismo que en el inciso anterior pero el archivo de destino, se debe llamar "f7".**

cp /home/gonzalo/dir2/f4 /home/gonzalo/dir1/dir11/f7

**d. Cree el directorio copia dentro del directorio usuario y copie en él, el contenido de "dir1".**

mkdir copia

sudo cp -r /home/gonzalo/dir1/\* /home/gonzalo/copia

**Nota:** Es necesario hacer una copia recursiva(-r) para mover todo el contenide del directorio.

**e. Renombre el archivo "f1" por el nombre archivo y vea los permisos del mismo.**

sudo mv /home/gonzalo/f1 /home/gonzalo/archivo

ls -l (estando en /home/gonzalo)

**Nota:** mv se utiliza para mover archivos/directorios, en caso de que el archivo destino no exista lo crea.

**f. Cambie los permisos del archivo llamado archivo de manera de reflejar lo siguiente:**

**➢ Usuario: Permisos de lectura y escritura**

**➢ Grupo: Permisos de ejecución**

**➢ Otros: Todos los permisos**

sudo chmod 617 /home/gonzalo/archivo

**g. Renombre los archivos "f3” y “f4" de manera que se llamen "f3.exe” y “f4.exe” respectivamente.**

sudo mv f3 f3.exe

sudo mv /home/gonzalo/dir2/f4 /home/gonzalo/dir2/f4.exe

sudo mv /home/gonzalo/dir1/dir11/f4 /home/gonzalo/dir1/dir11/f4.exe

**h. Utilizando un único comando cambie los permisos de los dos archivos renombrados en el inciso anterior, de manera de reflejar lo siguiente:**

**➢ Usuario: Ningún permiso**

**➢ Grupo: Permisos de escritura**

**➢ Otros: Permisos de escritura y ejecución**

sudo chmod 023 /home/gonzalo/f3.exe /home/gonzalo/dir2/f4.exe /home/gonzalo/dir1/dir11/f4.exe

**25. Indique qué comando/s es necesario para realizar cada una de las acciones de la siguiente secuencia de pasos (considerando su orden de aparición):**

**a. Cree un directorio llamado logs en el directorio /tmp.**

cd /

cd /tmp

mkdir logs

**b. Copie todo el contenido del directorio /var/log en el directorio creado en el punto anterior.**

sudo -cp -r /var/log/\* /tmp/logs

**c. Empaquete el directorio creado en a), el archivo resultante se debe llamar "misLogs.tar".**

sudo tar -cvf misLogs.tar /tmp/logs

**d. Empaquete y comprima el directorio creado en a), el archivo resultante se debe llamar "misLogs.tar.gz".**

sudo tar -cvzf misLogs.tar.gz /tmp/logs

**e. Copie los archivos creados en c) y d) al directorio de trabajo de su usuario.**

sudo cp misLogs.tar misLogs.tar.gz /home/gonzalo

**f. Elimine el directorio creado en a), logs.**

sudo rm -r /tmp/logs

**g. Desempaquete los archivos creados en c y d en 2 directorios diferentes.**

mkdir /home/gonzalo/desempaquetado\_tar

mkdir /home/gonzalo/desempaquetado\_tar\_gz

tar -xvf /home/gonzalo/misLogs.tar -C /home/gonzalo/desempaquetado\_tar

tar -xvf /home/gonzalo/misLogs.tar -C /home/gonzalo/desempaquetado\_tar\_gz