El sistema de archivos en Linux sigue las convenciones de Unix y tiene una estructura específica que es compatible y homogénea con otros sistemas Unix. A diferencia de Windows o MS-DOS, el sistema de archivos en Linux no está directamente ligado a la estructura del hardware, lo que significa que no depende del número de discos duros presentes en el sistema.

El sistema de archivos es fundamental para el funcionamiento de Linux, ya que almacena todos los datos necesarios para el sistema operativo, incluyendo programas, bibliotecas, archivos del sistema y archivos de usuario. Todos estos elementos residen en diferentes directorios que están organizados bajo la raíz del sistema, representada por el directorio **/**.

Algunos directorios importantes del sistema de archivos incluyen:

* **/boot**: Contiene archivos necesarios para el arranque del sistema, como los archivos del gestor de arranque y las imágenes del kernel.
* **/bin**: Almacena ejecutables y comandos del sistema.
* **/home**: Contiene los directorios de trabajo **(personales)** de los usuarios del sistema.
* **/lib**: Contiene las bibliotecas compartidas necesarias para que se ejecuten los programas en **/bin**.
* **/etc**: Reservado para archivos de configuración del sistema.
* **/tmp**: Directorio utilizado por el sistema para la creación de archivos temporales.
* **/dev**: Contiene archivos del sistema que representan dispositivos físicos instalados en el ordenador (discos duros, pendrive).
* **/mnt**: Punto de montaje por defecto para dispositivos de almacenamiento adicionales.
* **/opt**: Directorio opcional para la instalación de aplicaciones del usuario.
* **/var**: Contiene directorios que almacenan información variable, como registros del sistema, trabajos de impresión y mensajes de correo.
* **/usr**: Contiene programas y recursos de uso común para todos los usuarios.

Además, existen varios comandos para obtener ayuda en un sistema Linux, como **--help**, **help**, **info**, **whatis**, **apropos** y **man**, que proporcionan información detallada sobre el uso de comandos, formatos de archivos de configuración, llamadas al sistema, entre otros temas. Los manuales están disponibles en múltiples idiomas y se dividen en secciones para facilitar su consulta.

El comando **su** (Switch User) permite cambiar de usuario sin cerrar la sesión actual. Se utiliza con la sintaxis **su [-] [nombre-usuario]**. Al ejecutar **su** sin argumentos, intentará ingresar como superusuario (root), solicitando su contraseña.

Por otro lado, el comando **sudo** (SuperUser Do) permite a los usuarios ejecutar acciones con los privilegios de seguridad del superusuario (root) de manera segura. En sistemas como Ubuntu, donde la cuenta de root está desactivada por defecto, los usuarios con permisos adecuados pueden utilizar **sudo** para realizar tareas administrativas.

Es importante destacar que **sudo** reemplaza a **su** en muchos casos, permitiendo a los usuarios autorizados ejecutar comandos con privilegios de root temporalmente, sin necesidad de conocer la contraseña de root. La sintaxis básica de **sudo** es **$ sudo opción comando**, solicitando al usuario su propia contraseña para autenticarse.

En resumen, mientras **su** cambia de usuario, **sudo** permite a los usuarios ejecutar comandos con privilegios de root temporalmente y de forma segura, siempre y cuando estén autorizados en el archivo de configuración **/etc/sudoers**.

El usuario root en Ubuntu viene desactivado por defecto para evitar problemas de seguridad, ya que este usuario tiene control total sobre el sistema. En su lugar, se recomienda utilizar cuentas de administrador que tengan privilegios similares a root, pero de manera controlada.

Para iniciar sesión como root, es necesario seguir dos pasos:

1. Cambiar la contraseña de root:
   * Ejecutar el comando **sudo passwd root**.
   * Introducir la contraseña del usuario actual para verificar los permisos.
   * Establecer una nueva contraseña para root cuando se solicite.
2. Iniciar sesión como root:
   * Ejecutar el comando **su root**.
   * Introducir la contraseña de root cuando se solicite.

Es importante recordar que el uso de **sudo** para actuar como root tiene ciertas características:

* La contraseña de root tiene una duración de 15 minutos antes de necesitar ser ingresada nuevamente.
* Las contraseñas no se muestran en pantalla.
* Se puede ejecutar cualquier comando como root agregando **sudo** antes del mismo.

Para desactivar la cuenta de root, se puede utilizar el comando **sudo passwd -l root**, donde la opción "-l" significa "lock" (bloquear). Esto deshabilitará la cuenta de root y evitará que se inicie sesión con ella.

El formato de los comandos shell en Linux sigue una estructura específica, donde cada comando consiste en una secuencia de palabras separadas por uno o más espacios. La primera palabra es el nombre del comando que indica al shell qué acción realizar, seguida de opciones (precedidas por un signo menos "-") y argumentos que son utilizados como entrada del comando o archivos y directorios que se van a manipular.

Los metacaracteres, también conocidos como caracteres comodines, permiten nombrar archivos de forma abreviada y acceder a muchos archivos en un solo comando. El comodín "\*" se refiere a cualquier carácter o cadena de caracteres en el nombre de un archivo, mientras que el "?" expande a un único carácter.

Es importante entender cómo Linux expande la línea de órdenes cuando encuentra metacaracteres. El intérprete de comandos examina los nombres que cumplen la plantilla, los reconstruye y luego ejecuta la orden.

El uso del comodín "\*" puede no coincidir con nombres de archivos que comienzan con un punto ("."), ya que estos archivos son tratados como "ocultos". Por ejemplo, ls -a muestra nombres de archivos ocultos, pero el comodín "\*" no coincidirá con ellos.

Otros metacaracteres incluyen "[ ]" que representa un solo carácter de los incluidos entre corchetes, y el operador de negación "!" que permite la exclusión de caracteres.

En resumen, los metacaracteres son una característica importante de los intérpretes de comandos en Linux que permiten referirse a más de un archivo usando caracteres especiales para facilitar la manipulación de archivos y directorios en la línea de comandos.

Este resumen abarca una variedad de comandos y herramientas básicas en el entorno de Linux. Aquí está un resumen conciso de los puntos clave:

**Comandos Iniciales: Formato**

* **MAN:** Es el equivalente al comando HELP de MS-DOS, pero más detallado. Proporciona ayuda sobre comandos específicos, incluyendo parámetros.
* **PWD:** Muestra la trayectoria completa del directorio actual.
* **CD:** Cambia el directorio de trabajo actual.
* **LS:** Permite ver el contenido de un directorio con varias opciones para personalizar la salida.
* **PASSWD:** Permite al usuario cambiar o asignar una contraseña.
* **UNAME:** Muestra el nombre del sistema en el que se está trabajando.
* **HOSTNAME:** Muestra o establece el nombre del sistema.
* **LOGNAME:** Obtiene el nombre de usuario conectado al ordenador.
* **ID:** Muestra el número de identificación y el grupo al que pertenece el usuario.
* **DATE:** Se utiliza para visualizar la fecha y la hora.
* **WHO:** Muestra información sobre los usuarios conectados al sistema.
* **CAL:** Muestra un calendario.
* **MORE:** Muestra el contenido de un fichero página a página.
* **ECHO:** Envía argumentos al terminal.
* **CLEAR:** Limpia la pantalla.
* **MKDIR:** Crea directorios.
* **RMDIR:** Borra directorios.
* **MV:** Cambia el nombre o mueve directorios.
* **TOUCH:** crea archivos o modificación de un archivo.
* **CAT:** Visualiza y encadena (concatena) ficheros.
* **PAGE:** Muestra el contenido de un archivo de texto página a página.
* **RM :** Borrar archivos

Estos comandos son fundamentales para navegar, administrar y manipular archivos y directorios en un sistema Linux.

1. **MAN:** **Descripción:** Proporciona manuales en línea para comandos específicos.
   * **Opciones:**
     + **-f**: Muestra una breve descripción del comando.
     + **-k**: Busca comandos relacionados con el término especificado.
     + **-w**: Muestra la ubicación del manual en lugar de su contenido.
     + **-h**, **--help**: Muestra un resumen de las opciones de uso.
2. **PWD:** **Descripción:** Muestra la ruta completa del directorio actual.
3. **CD: Descripción:** Cambia el directorio de trabajo actual.
   * **Opciones:**
     + **directorio**: Especifica el directorio al que se quiere cambiar.
4. **LS:** **Descripción:** Lista el contenido de un directorio.
   * **Opciones:**
     + **-a**: Lista todos los archivos, incluyendo los ocultos.
     + **-l**: Muestra los archivos en formato largo.
     + **-R**: Realiza un listado recursivo de todos los archivos.
     + **-r**: Lista los archivos en orden inverso.
     + **-F**: Agrega caracteres especiales para distinguir entre archivos, directorios, etc.
     + **-i**: Muestra el número de i-nodo asociado a cada elemento.
     + **-d**: Proporciona información sobre el directorio en lugar de su contenido.
5. **PASSWD: Descripción:** Permite al usuario cambiar o asignar una contraseña.

* -d : **Deshabilita** la cuenta de usuario eliminando su contraseña

Sudo passwd -d tuprimalacoja

* -l : **Bloquea** la cuenta del usuario, (incluyendo el símbolo (!) delante del campo contraseña de la linea del usuario en el archivo /etc/shadow)

Sudo passwd -l tuprimalacoja

* -u : **Desbloquea** la cuenta del usuario (quitando el símbolo (!))

Sudo passwd -u tuprimalacoja

1. **UNAME:** **Descripción:** Muestra el nombre del sistema en el que se está trabajando.
   * **Opciones:**
     + **-a**: Muestra información adicional sobre el sistema.

**HOSTNAME:** **Descripción:** Muestra o establece el nombre del sistema.

1. **LOGNAME:** **Descripción:** Obtiene el nombre de usuario conectado al ordenador.
2. **ID:** **Descripción:** Muestra el número de identificación y el grupo al que pertenece el usuario.
3. **DATE:** **Descripción:** Se utiliza para visualizar la fecha y la hora.
   * **Opciones:**
     + **+formato**: Especifica el formato de salida de la fecha y hora.
4. **WHO:** **Descripción:** Muestra información sobre los usuarios conectados al sistema.
   * **Opciones:**
     + **-H**: Muestra una cabecera explicando el contenido de cada columna.
     + **-u**: Visualiza información detallada sobre los usuarios actuales.
5. **CAL:** **Descripción:** Muestra un calendario.
   * **Opciones:**
     + **-m**: Muestra el calendario del mes en curso.
     + **-j**: Muestra el calendario en formato juliano.
     + **-y**: Muestra el calendario completo del año en curso.

Estas opciones te permitirán utilizar cada comando de manera más efectiva y personalizada según tus necesidades en el entorno de Linux.

1. **Comando clear**:
   * Este comando se utiliza para limpiar la pantalla de la terminal, lo que hace que la terminal se vuelva a su estado predeterminado sin ningún texto visible. Es útil cuando la pantalla está llena de resultados de comandos anteriores y deseas comenzar con una pantalla limpia.
2. **Comando cd (Cambiar Directorio)**:
   * El comando "cd" se utiliza para cambiar el directorio de trabajo actual en un sistema Unix/Linux.
   * Algunos directorios especiales tienen indicativos especiales, como el directorio raíz "/", el directorio padre "..", el directorio de usuario por defecto "$HOME", y el directorio actual "."
   * Es importante tener en cuenta que "cd" sin argumentos te lleva al directorio por defecto del usuario (HOME).
   * El uso correcto implica dejar un espacio entre el comando "cd" y el nombre del directorio al que deseas cambiar.
   * Este comando es fundamental para la navegación en el sistema de archivos de Unix/Linux.
3. **Variables de entorno**:
   * Las variables de entorno son variables dinámicas que pueden afectar el comportamiento de los programas en un sistema Unix/Linux.
   * El comando "printenv" seguido del nombre de la variable de entorno, como "HOME", muestra su contenido.
   * Las variables de entorno son ampliamente utilizadas para configurar el entorno de trabajo y la ejecución de programas en sistemas Unix/Linux.
4. **Comando mkdir (Crear Directorio)**:
   * "mkdir" se utiliza para crear nuevos directorios en el sistema de archivos.
   * La opción "-p" permite crear directorios de forma recursiva, es decir, crear una jerarquía de directorios si alguno de los directorios padres no existe, lo crea.
   * La opción "-m" se utiliza para establecer permisos específicos para el nuevo directorio creado.
   * Este comando es esencial para organizar y estructurar el sistema de archivos según las necesidades del usuario.
5. **Comando rmdir (Borrar Directorio)**:
   * "rmdir" se utiliza para borrar directorios vacíos del sistema de archivos.
   * La opción "-p" permite borrar directorios padres si están vacíos, lo que facilita la limpieza de estructuras de directorios.
   * La opción "-v" proporciona mensajes verbosos, lo que puede ser útil para rastrear el proceso de eliminación.
   * Es importante tener en cuenta que los directorios deben estar vacíos para que "rmdir" pueda borrarlos.
6. **Comandos de ficheros**:

**Comando touch**: Se utiliza para cambiar las fechas de acceso y modificación de un archivo o crear uno nuevo. Es útil para manipular las marcas de tiempo de un archivo.

**Comando cat**: visualiza los archivos

* Si concatenas varios archivos y solo quieres ver el contenido sin redirigirlo a otro archivo, puedes simplemente usar **cat archivo1.txt archivo2.txt**.
* Si deseas agregar contenido a un archivo existente, es importante usar **>>** en lugar de **>** para no sobrescribir el archivo existente.
* **cat** es especialmente útil cuando necesitas combinar varios archivos de texto en uno solo, como por ejemplo al unir archivos de registro o archivos de código fuente divididos.

**Comandos more y page**: Permiten visualizar el contenido de archivos página por página, lo que facilita la lectura de archivos extensos. Ofrecen opciones para navegar y gestionar la visualización de los datos.

Principio del formulario