1). Editor de textos:

(a) Nombre al menos 3 editores de texto que puede utilizar desde la línea de comandos.

 **vim**: Un editor de texto potente y extensible que es conocido por su eficiencia y modos de operación (modo de comando y modo de inserción).

 **nano**: Un editor de texto más simple y fácil de usar, ideal para quienes buscan una interfaz más amigable en comparación con vim.

 **emacs**: Otro editor de texto muy versátil y potente, que ofrece una amplia gama de funcionalidades y puede ser extensible mediante el uso de Lisp.

(b) ¿En qué se diferencia un editor de texto de los comandos cat, more o less? Enumere los modos de operación que posee el editor de textos vi.

1. **Editores de Texto (como vim, nano, emacs)**:
   * **Propósito**: Permiten crear, modificar y guardar archivos de texto.
   * **Interactividad**: Ofrecen una interfaz interactiva donde puedes mover el cursor, editar texto, buscar y reemplazar contenido, etc.
   * **Funcionalidades Avanzadas**: Generalmente tienen características como resaltado de sintaxis, autocompletado, y soporte para extensiones o plugins.
   * **Guardado**: Los cambios se pueden guardar en el archivo de texto actual.
2. **Comandos cat, more, less**:
   * **Propósito**: Sirven principalmente para visualizar el contenido de archivos de texto.
   * **Interactividad**:
     + **cat**: Muestra el contenido del archivo de una vez y no permite interacciones o modificaciones.
     + **more**: Permite ver el contenido del archivo página por página. Puedes avanzar a la siguiente página, pero no puedes retroceder.
     + **less**: Similar a more, pero con más funcionalidades. Permite navegar hacia adelante y hacia atrás, buscar texto dentro del archivo, y más.
   * **Modificaciones**: No permiten editar el contenido del archivo, solo mostrarlo.

**Modos de Operación en el Editor de Texto vi (o vim)**

vi tiene tres modos principales de operación:

1. **Modo Normal**:
   * Es el modo por defecto cuando entras en vi. En este modo, puedes moverte por el texto, eliminar y copiar texto, y realizar otras operaciones de edición. No puedes insertar texto directamente en este modo.
2. **Modo de Inserción**:
   * Puedes entrar en este modo presionando i (insertar antes del cursor), I (insertar al principio de la línea), a (insertar después del cursor), A (insertar al final de la línea), o (abrir una nueva línea debajo) o O (abrir una nueva línea arriba). En este modo, puedes escribir y modificar el texto directamente.
3. **Modo de Comando**:
   * Accedes a este modo presionando : en el modo normal. En este modo, puedes ingresar comandos para guardar el archivo (:w), salir del editor (:q), buscar texto, reemplazar texto, y otras operaciones administrativas.

(c) Nombre los comandos más comunes que se le pueden enviar al editor de textos vi.

 **Movimiento del Cursor**:

* h: Mover el cursor una posición a la izquierda.
* j: Mover el cursor una línea hacia abajo.
* k: Mover el cursor una línea hacia arriba.
* l: Mover el cursor una posición a la derecha.
* 0: Mover el cursor al inicio de la línea.
* $: Mover el cursor al final de la línea.
* gg: Ir al principio del archivo.
* G: Ir al final del archivo.

 **Edición de Texto**:

* i: Entrar en el modo de inserción antes del cursor.
* I: Entrar en el modo de inserción al inicio de la línea.
* a: Entrar en el modo de inserción después del cursor.
* A: Entrar en el modo de inserción al final de la línea.
* o: Abrir una nueva línea debajo del cursor y entrar en el modo de inserción.
* O: Abrir una nueva línea arriba del cursor y entrar en el modo de inserción.
* x: Eliminar el carácter bajo el cursor.
* dd: Eliminar la línea actual.
* yy: Copiar (yank) la línea actual.
* p: Pegar el contenido copiado o eliminado después del cursor.
* u: Deshacer la última acción.
* Ctrl + r: Rehacer la última acción deshecha.

 **Búsqueda y Reemplazo**:

* /texto: Buscar texto hacia adelante en el archivo.
* ?texto: Buscar texto hacia atrás en el archivo.
* n: Ir a la siguiente ocurrencia de la búsqueda.
* N: Ir a la ocurrencia anterior de la búsqueda.
* :s/antiguo/nuevo/: Reemplazar la primera ocurrencia de antiguo por nuevo en la línea actual.
* :s/antiguo/nuevo/g: Reemplazar todas las ocurrencias de antiguo por nuevo en la línea actual.
* :%s/antiguo/nuevo/g: Reemplazar todas las ocurrencias de antiguo por nuevo en todo el archivo.

 **Guardar y Salir**:

* :w: Guardar los cambios en el archivo.
* :q: Salir del editor.
* :wq o :x: Guardar los cambios y salir del editor.
* :q!: Salir del editor sin guardar los cambios

2. 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 Se empieza a ejecutar el codigo del BIOS

2 El BIOS ejecuta el POST

3 El BIOS lee el sector de arranque (MBR)

4 Se carga el gestor de arranque (MBC)

5 El bootloader carga el kernel y el initrd

6 Se monta el initrd como sistema de archivos raız y se inicializan componentes esenciales (ej.: scheduler)

7 El Kernel ejecuta el proceso init y se desmonta el initrd

8 Se lee el /etc/inittab

9 Se ejecutan los scripts apuntados por el runlevel 1

10 El final del runlevel 1 le indica que vaya al runlevel por defecto

11 Se ejecutan los scripts apuntados por el runlevel por defecto

12 El sistema esta a listo para usarse

(b) Proceso INIT. ¿Quién lo ejecuta? ¿Cuál es su objetivo?

1 Su función es cargar todos los subprocesos necesarios para el correcto funcionamiento del SO

2 El proceso init posee el PID 1 y se encuentra en /sbin/init

3 En SysV se lo configura a traves del archivo /etc/inittab

4 No tiene padre y es el padre de todos los procesos (pstree)

5 Es el encargado de montar los filesystems y de hacer disponible los dem´ as dispositivos

### ¿Quién lo Ejecuta?

* **Ejecución del Proceso init**:
  + **El núcleo (kernel)**: Una vez que el núcleo (kernel) de GNU/Linux ha terminado su inicialización y ha montado el sistema de archivos raíz (root filesystem), el proceso init es el primer proceso que el núcleo ejecuta en modo usuario. En el proceso de arranque, init recibe el PID (Process ID) 1, lo que indica que es el primer proceso y el ancestro de todos los demás procesos en el sistema.

### ¿Cuál es su Objetivo?

* **Objetivo del Proceso init**:
  + **Inicialización del Sistema**: El principal objetivo de init es iniciar el sistema y preparar el entorno para que los usuarios puedan interactuar con él. Esto incluye configurar y lanzar los servicios necesarios, así como preparar el sistema para la operación normal.
  + **Gestión de Servicios y Procesos**: init se encarga de iniciar y gestionar los diferentes servicios y procesos necesarios para el funcionamiento del sistema. Esto se realiza mediante la ejecución de scripts y la activación de servicios especificados en archivos de configuración. Estos servicios pueden incluir redes, servicios de registro, sistemas de archivos, y más.
  + **Ejecución de Scripts de Inicialización**: init ejecuta scripts de inicialización que configuran el sistema y los servicios esenciales. Estos scripts suelen estar ubicados en directorios como /etc/init.d en sistemas basados en SysVinit, o en unidades y objetivos en sistemas basados en systemd.
  + **Control de Niveles de Ejecución (Runlevels)**:
    - **SysVinit**: En sistemas que usan SysVinit, init gestiona diferentes niveles de ejecución (runlevels). Cada runlevel representa un estado del sistema con un conjunto específico de servicios en ejecución. Por ejemplo, el runlevel 3 suele ser para el modo multiusuario con red, mientras que el runlevel 5 puede ser para el modo gráfico.
    - **systemd**: En sistemas que usan systemd, el concepto de runlevels se reemplaza por **targets**. Los targets representan estados del sistema similares a los runlevels, como graphical.target para el modo gráfico y multi-user.target para el modo multiusuario.
  + **Manejo de Señales y Eventos del Sistema**: init también maneja eventos y señales del sistema, tales como el arranque, el apagado o el reinicio. Coordina la transición entre estados del sistema y asegura que todos los servicios y procesos se gestionen correctamente durante estas transiciones.
  + **Limpieza y Apagado**: Durante el apagado del sistema, init se encarga de detener todos los servicios y procesos en un orden específico, asegurando que el sistema se apague de manera ordenada y segura.

(c) RunLevels. ¿Qué son? ¿Cuál es su objetivo?

(d) ¿A qué hace referencia cada nivel de ejecución según el estándar? ¿Dónde se define qué Runlevel ejecutar al iniciar el sistema operativo? ¿Todas las distribuciones respetan estos estándares?

Es el modo en que arranca Linux (3 en Redhat, 2 en Debian)

• El proceso de arranque lo dividimos en niveles

• Cada uno es responsable de levantar (iniciar) o bajar (parar) una serie de servicios

### ¿Qué son los Runlevels?

* **Definición**:
  + Un runlevel es un estado predefinido del sistema que determina qué servicios y procesos deben estar activos. Cada runlevel configura el sistema para un propósito específico, ya sea para el funcionamiento normal, mantenimiento, o apagado.
* **Numeración de Runlevels**:
  + Los runlevels están representados por números, y cada número corresponde a un conjunto específico de servicios que se inician o detienen. Aunque los números y sus significados pueden variar ligeramente entre diferentes distribuciones, los runlevels más comunes son los siguientes:
    - **0**: **Apagado**. El sistema se apaga. Este runlevel detiene todos los servicios y apaga el sistema.
    - **1**: **Modo de Usuario Único (Single-User Mode)**. Utilizado para tareas de mantenimiento del sistema. Solo un usuario (generalmente root) puede acceder, y se inician servicios mínimos necesarios para el mantenimiento.
    - **2**: **Modo Multiusuario sin Red (Multi-User Mode without Networking)**. El sistema permite múltiples usuarios pero no inicia servicios de red.
    - **3**: **Modo Multiusuario con Red (Multi-User Mode with Networking)**. El sistema permite múltiples usuarios y se inician servicios de red, pero no proporciona una interfaz gráfica de usuario (GUI).
    - **4**: **Modo Personalizable (Custom)**. Generalmente reservado para uso personal o personalizado por el administrador del sistema. Puede ser configurado para que tenga características específicas.
    - **5**: **Modo Multiusuario con GUI (Multi-User Mode with Graphical User Interface)**. El sistema permite múltiples usuarios, inicia servicios de red y también inicia el entorno gráfico (X Window System o un gestor de ventanas).
    - **6**: **Reinicio (Reboot)**. El sistema se reinicia. Este runlevel detiene todos los servicios y reinicia el sistema.

### ¿Cuál es su Objetivo?

* **Gestión del Estado del Sistema**:
  + El objetivo principal de los runlevels es proporcionar una forma estandarizada para gestionar el estado del sistema y los servicios que deben estar en funcionamiento en cada estado. Permite a los administradores del sistema configurar y cambiar el entorno del sistema según sea necesario.
* **Flexibilidad en el Arranque y Apagado**:
  + Los runlevels permiten al administrador del sistema controlar qué servicios y procesos se inician o detienen en función del uso previsto del sistema. Por ejemplo, en un entorno de servidor, puede ser útil arrancar en runlevel 3 (modo multiusuario con red) para evitar la sobrecarga del entorno gráfico.
* **Simplificación del Mantenimiento**:
  + En situaciones de mantenimiento o resolución de problemas, el runlevel 1 (modo de usuario único) proporciona un entorno mínimo con acceso limitado, facilitando la reparación o configuración del sistema sin interferencias de servicios no esenciales.
* **Control de Servicios**:
  + Cada runlevel tiene asociados scripts de inicio y detención que configuran los servicios y procesos adecuados para ese nivel. Los scripts se encuentran típicamente en directorios como /etc/rc.d o /etc/init.d, y se ejecutan automáticamente cuando el sistema cambia de un runlevel a otro

### Dónde se Define Qué Runlevel Ejecutar al Iniciar el Sistema Operativo

* **Archivo de Configuración**:
  + En sistemas basados en SysVinit, el runlevel predeterminado para el inicio del sistema se define en el archivo /etc/inittab. Este archivo contiene una línea que especifica el runlevel predeterminado mediante la configuración id

**Sistema de Inicialización Alternativo (Systemd)**:

* En sistemas que usan systemd, el concepto de runlevels ha sido reemplazado por **targets**. El runlevel predeterminado se define mediante los archivos de configuración en /etc/systemd/system/default.target

### ¿Todas las Distribuciones Respetan Estos Estándares?

* **Respeto a los Estándares**:
  + **Distribuciones Basadas en SysVinit**: La mayoría de las distribuciones que usan SysVinit respetan estos niveles de ejecución estándar. Sin embargo, algunos sistemas pueden tener configuraciones personalizadas para el runlevel 4, o pueden agregar sus propios runlevels adicionales para necesidades específicas.
* **Distribuciones Basadas en Systemd**: Las distribuciones modernas como Fedora, Ubuntu (desde 15.04 en adelante), y Debian (desde 8 "Jessie") utilizan systemd, que ha reemplazado los runlevels tradicionales con un sistema basado en targets. Aunque los targets ofrecen una funcionalidad similar, el enfoque y la configuración son diferentes.

(e) Archivo /etc/inittab. ¿Cuál es su finalidad? ¿Qué tipo de información se almacena en él? ¿Cuál es la estructura de la información que en él se almacena?

El archivo /etc/inittab se utiliza para configurar el comportamiento del sistema durante el arranque y el apagado. Define cómo se inicializa el sistema y qué procesos deben ser gestionados por el proceso init, que es el primer proceso que se ejecuta en modo usuario cuando el núcleo (kernel) ha terminado su inicialización.

### Tipo de Información Almacenada en /etc/inittab

1. **Runlevel Predeterminado**:
   * Define el runlevel en el que el sistema debe arrancar por defecto. Este es el nivel de ejecución inicial que se establece cuando el sistema se inicia.
2. **Configuración de Procesos y Servicios**:
   * Especifica qué procesos y servicios deben ser iniciados, detenidos o gestionados en diferentes runlevels. Incluye detalles sobre cómo iniciar y supervisar estos procesos.
3. **Manejo de Tareas de Mantenimiento**:
   * Configura tareas de mantenimiento y procedimientos específicos que deben ejecutarse en ciertos runlevels.
4. **Configuración de Consolas y Terminales**:
   * Puede definir cómo se deben gestionar las consolas y terminales virtuales, y cómo se deben configurar los servicios asociados.

### Estructura de la Información en /etc/inittab

El archivo /etc/inittab sigue una estructura de líneas de configuración con un formato específico. Cada línea suele estar compuesta por varios campos separados por dos puntos (:). La estructura general es la siguiente:

Id: runlevels:action:process

Donde:

* **id**: Un identificador único para la entrada. Debe ser un nombre corto y único que represente la configuración o el proceso.
* **runlevels**: Una lista de runlevels en los que esta entrada debe aplicarse. Los runlevels están separados por comas y pueden incluir rangos (por ejemplo, 2,3,4 o 2-5).
* **action**: La acción que debe llevarse a cabo en los runlevels especificados. Algunas de las acciones comunes son:
  + respawn: Iniciar el proceso en el runlevel especificado y reiniciarlo si se detiene.
  + wait: Iniciar el proceso y esperar a que termine antes de continuar.
  + once: Iniciar el proceso solo una vez durante el arranque.
  + sysinit: Ejecutar el proceso al inicio del sistema (antes de entrar en el primer runlevel).
  + shutdown: Ejecutar el proceso durante el apagado del sistema.
* **process**: El comando o el proceso que debe ser ejecutado. Puede ser una ruta a un script o a un binario que se debe iniciar en el runlevel especificado.

(f) Suponga que se encuentra en el runlevel <X>. Indique qué comando(s) ejecutaría para cambiar al runlevel <Y>. ¿Este cambio es permanente? ¿Por qué?

Para cambiar de un runlevel <X> a un runlevel <Y> en un sistema GNU/Linux que utiliza **SysVinit**, podemos usar el comando init. Este comando permite cambiar el runlevel del sistema de manera temporal durante la sesión actual. Ej: init <Y> (donde y es el número a cambiar).

 **Cambio Temporal**:

* **Sí, el cambio es temporal**. El comando init <Y> cambiará el runlevel en el que se encuentra el sistema y ajustará los servicios y procesos según la configuración definida para el runlevel <Y>. Este cambio afecta solo a la sesión actual. Cuando se reinicie el sistema, se volverá a usar el runlevel predeterminado especificado en /etc/inittab.

 **No Permanente**:

* El cambio no es permanente en el sentido de que no modifica la configuración de arranque del sistema. El runlevel predeterminado que se establece en el archivo /etc/inittab seguirá siendo utilizado en el próximo arranque. Para cambiar el runlevel predeterminado permanentemente, habria que editar el archivo /etc/inittab y ajustar la línea que define el runlevel predeterminado asi: id:<Y>:initdefault:

(g) Scripts RC. ¿Cuál es su finalidad? ¿Dónde se almacenan? Cuando un sistema GNU/Linux arranca o se detiene se ejecutan scripts, indique cómo determina qué script ejecutar ante cada acción. ¿Existe un orden para llamarlos? Justifique.

### Finalidad de los Scripts RC

Los scripts RC son utilizados para iniciar, detener y gestionar servicios y procesos del sistema en diferentes niveles de ejecución (runlevels). Su finalidad es:

* **Inicialización del Sistema**: Al iniciar el sistema, los scripts RC ejecutan los procesos necesarios para poner en marcha servicios y configuraciones según el nivel de ejecución configurado.
* **Apagado del Sistema**: Durante el apagado, los scripts RC aseguran que todos los servicios se detengan de manera ordenada y que se realicen las tareas de limpieza necesarias.
* **Gestión de Servicios**: Permiten la activación y desactivación de servicios y procesos según el runlevel especificado.

### Ubicación de los Scripts RC

* **Ubicación**:
  + Los scripts RC suelen encontrarse en los directorios /etc/init.d/ o /etc/rc.d/. En estas ubicaciones, cada script corresponde a un servicio o proceso específico.
  + Los scripts de nivel de ejecución también se encuentran en subdirectorios bajo /etc/rc.d/ o /etc/rcX.d/ (donde X representa el runlevel). Por ejemplo, los scripts para el runlevel 3 podrían encontrarse en /etc/rc3.d/.

### Determinación de Qué Script Ejecutar

* **Durante el Arranque**:
  + Al iniciar el sistema, el proceso init lee el archivo /etc/inittab para determinar el runlevel predeterminado.
  + Luego, init busca en el directorio /etc/rcX.d/ (donde X es el número del runlevel) para encontrar los scripts que deben ser ejecutados.
  + En cada directorio rcX.d, los scripts de inicio y detención están prefijados por letras como S (Start) o K (Kill), seguidas de un número que determina el orden en que deben ejecutarse. Por ejemplo, S20network se ejecutará antes que S50httpd.
* **Durante el Apagado**:
  + Al apagar el sistema, init cambia al runlevel 0 (o al runlevel de apagado especificado) y busca en el directorio /etc/rc0.d/ (o el directorio correspondiente) para detener los servicios.
  + Los scripts en este directorio están prefijados con la letra K, que indica que deben detener los servicios. Similar a los scripts de inicio, estos scripts tienen números que determinan el orden en que se deben ejecutar.

### Orden de Ejecución

* **Orden de Ejecución**:
  + Los scripts en los directorios /etc/rcX.d/ se ejecutan en orden numérico. Los números prefijados en los nombres de los scripts determinan el orden de ejecución.
  + Por ejemplo, los scripts S20network y S50httpd en el directorio /etc/rc3.d/ se ejecutarán en ese orden, con S20network ejecutándose antes de S50httpd.
  + Durante el apagado, los scripts K también se ejecutan en orden numérico. Los scripts como K20network y K50httpd se ejecutan en el orden numérico inverso al que se ejecutaron durante el arranque.

### Justificación del Orden

* **Justificación del Orden**:
  + El orden de ejecución es crucial para garantizar que los servicios dependientes se inicien y detengan en el orden correcto. Por ejemplo, un servicio de red (network) debe iniciarse antes que un servidor web (httpd), ya que el servidor web depende de la red para funcionar.
  + Al detener los servicios, es igualmente importante que los servicios dependientes se detengan en el orden inverso para evitar problemas de dependencia y asegurar que el sistema se apague de manera ordenada.

3. SystemD(https://github.com/systemd/systemd):

(a) ¿Qué es sytemd?

**Systemd** es un sistema de inicialización y gestor de servicios que organiza el arranque del sistema, la gestión de procesos y la administración de servicios. Está diseñado para ser más eficiente y flexible en comparación con los sistemas de inicialización tradicionales como SysVinit.

(b) ¿A qué hace referencia el concepto de Unit en SystemD?

Es un sistema que centraliza la administración de demonios(¿?) y librerias del sistema

Una **unidad** en systemd es un archivo de configuración que define un servicio, recurso o conjunto de recursos que systemd debe gestionar. Cada unidad está representada por un archivo con una extensión específica y contiene configuraciones sobre cómo se debe iniciar, detener, reiniciar, y gestionar esa unidad.

(c) ¿Para que sirve el comando systemctl en SystemD?

El comando systemctl es una herramienta clave en **systemd** que se utiliza para interactuar con el sistema de inicialización y la gestión de servicios. Permite a los administradores del sistema gestionar y controlar los servicios, sockets, montajes, y otras unidades que systemd gestiona

(d) ¿A qué hace referencia el concepto de target en SystemD?

Un **target** en systemd es una unidad con la extensión .target que agrupa otras unidades. Sirve como un punto de sincronización para alcanzar un estado específico del sistema. Por ejemplo, un target puede representar el estado del sistema durante el arranque, la operación normal o el apagado.

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

4. Usuarios:

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

### 1. ****/etc/passwd****

* **Propósito**: Contiene información básica sobre las cuentas de usuario del sistema.
* **Contenido**: Cada línea del archivo /etc/passwd representa una cuenta de usuario y tiene el siguiente formato:
* username:password:UID:GID:GECOS:home\_directory:shell

 username: Nombre de usuario.

 password: Un marcador de posición para la contraseña (normalmente un x si las contraseñas están almacenadas en /etc/shadow).

 UID: Identificador único del usuario.

 GID: Identificador del grupo principal del usuario.

 GECOS: Información adicional del usuario (como nombre completo).

 home\_directory: Ruta al directorio home del usuario.

 shell: Ruta al intérprete de comandos (shell) del usuario.

### 2. ****/etc/shadow****

* **Propósito**: Contiene información sobre las contraseñas de los usuarios y las políticas de seguridad relacionadas.
* **Contenido**: Cada línea del archivo /etc/shadow corresponde a un usuario y tiene el siguiente formato:
* username:encrypted\_password:last\_changed:min\_age:max\_age:warning:inactive:expire
*  encrypted\_password: Contraseña cifrada del usuario (o un valor especial como ! o \* para indicar que la cuenta está deshabilitada).
*  last\_changed: Número de días desde el 1 de enero de 1970 en que la contraseña fue cambiada por última vez.
*  min\_age: Número mínimo de días que debe transcurrir antes de que el usuario pueda cambiar la contraseña.
*  max\_age: Número máximo de días que una contraseña es válida.
*  warning: Número de días antes de la expiración de la contraseña en que el usuario será advertido.
*  inactive: Número de días después de que la contraseña expire antes de que la cuenta sea deshabilitada.
*  expire: Fecha de expiración de la cuenta en formato de días desde el 1 de enero de 1970.

### 3. ****/etc/group****

* **Propósito**: Contiene información sobre los grupos del sistema.
* **Contenido**: Cada línea del archivo /etc/group representa un grupo y tiene el siguiente formato:
* groupname:password:GID:user\_list
*  groupname: Nombre del grupo.
*  password: Contraseña cifrada del grupo (raramente utilizada).
*  GID: Identificador del grupo.
*  user\_list: Lista de usuarios que son miembros del grupo.

### 4. ****/etc/gshadow****

* **Propósito**: Contiene información adicional sobre los grupos, especialmente las contraseñas de los grupos.
* **Contenido**: Cada línea del archivo /etc/gshadow corresponde a un grupo y tiene el siguiente formato:
* groupname:encrypted\_password:admin\_list:user\_list

 groupname: Nombre del grupo.

 encrypted\_password: Contraseña cifrada del grupo (o un marcador de posición como ! para indicar que no hay contraseña).

 admin\_list: Lista de administradores del grupo.

 user\_list: Lista de miembros del grupo.

### 5. ****/etc/login.defs****

* **Propósito**: Contiene parámetros de configuración para la administración de cuentas de usuario.
* **Contenido**: Define configuraciones globales para el sistema, como el rango de UID y GID, políticas de expiración de contraseñas y otras configuraciones relacionadas con el manejo de cuentas de usuario.

(b) ¿A qué hacen referencia las siglas UID y GID (lo hice en el anterior sin darme cuenta)? ¿Pueden coexistir UIDs iguales en un sistema GNU/Linux? Justifique.

En un sistema GNU/Linux, los **UIDs (User Identifiers)** deben ser únicos para cada cuenta de usuario. La razón principal para esta unicidad es asegurar una identificación precisa y sin ambigüedades de los usuarios y sus permisos en el sistema.

(c) ¿Qué es el usuario root? ¿Puede existir más de un usuario con este perfil en GNU/Linux? ¿Cuál es la UID del root?.

(d) Agregue un nuevo usuario llamado iso2017 a su instalación de GNU/Linux, especifique que su home sea creada en /home/iso\_2017, y hágalo miembro del grupo catedra (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.

### Justificación de la Unicidad de los UIDs

1. **Identificación Única**:
   * Cada UID en un sistema GNU/Linux debe ser único para identificar de manera precisa a un usuario. El UID es un número entero que el sistema operativo usa para distinguir entre diferentes cuentas de usuario y gestionar los permisos de acceso a archivos y recursos.
2. **Permisos y Propietario de Archivos**:
   * Los UIDs se utilizan para asignar la propiedad de los archivos y directorios. Si hubiera UIDs duplicados, el sistema no podría distinguir correctamente entre los propietarios de los archivos, lo que podría resultar en una administración de permisos confusa y errores en la seguridad.
3. **Control de Acceso**:
   * Los sistemas de control de acceso, como los permisos de archivos y las listas de control de acceso (ACL), dependen de la unicidad del UID para determinar quién tiene acceso a qué recursos. Un UID duplicado violaría esta lógica y podría llevar a problemas de seguridad y acceso indebido.
4. **Consistencia del Sistema**:
   * El sistema de archivos, el sistema de control de acceso y otros componentes del sistema operativo se basan en la suposición de que cada UID es único. La duplicación de UIDs podría causar inconsistencias en la configuración del sistema y en la forma en que se gestionan las cuentas de usuario.

(e) Investigue la funcionalidad y parámetros de los siguientes comandos:

### 1. ****useradd / adduser****

* **Funcionalidad**:
  + **useradd**: Crea una nueva cuenta de usuario en el sistema.
  + **adduser**: En muchas distribuciones, adduser es un script que proporciona una interfaz más amigable y de alto nivel para agregar usuarios, y puede ser un enlace simbólico a useradd con algunas configuraciones predeterminadas.
* **Parámetros Comunes**:
  + -m: Crea un directorio home para el usuario.
  + -d DIR: Especifica el directorio home del usuario.
  + -s SHELL: Define el intérprete de comandos (shell) para el usuario.
  + -G GROUPS: Añade el usuario a los grupos adicionales especificados.
  + -u UID: Asigna un UID específico al usuario.
  + -c COMMENT: Añade un comentario o información adicional (como el nombre completo) para el usuario

Ej: useradd -m -s /bin/bash -G sudo -c "John Doe" john

### 2. ****usermod****

* **Funcionalidad**:
  + Modifica una cuenta de usuario existente.
* **Parámetros Comunes**:
  + -aG GROUPS: Añade el usuario a los grupos especificados sin eliminarlo de los grupos existentes.
  + -s SHELL: Cambia el intérprete de comandos (shell) del usuario.
  + -d DIR: Cambia el directorio home del usuario.
  + -e EXPIRE\_DATE: Establece una fecha de expiración para la cuenta en formato AAAA-MM-DD.
  + -l NEW\_NAME: Cambia el nombre de usuario.

Ej: usermod -aG developers john

### 3. ****userdel****

* **Funcionalidad**:
  + Elimina una cuenta de usuario del sistema.
* **Parámetros Comunes**:
  + -r: Elimina el directorio home y los archivos del usuario junto con la cuenta.
  + -f: Fuerza la eliminación de la cuenta incluso si el usuario está actualmente conectado.

Ej: userdel -r john

### . ****su****

* **Funcionalidad**:
  + Permite cambiar de usuario en una sesión de terminal. Por defecto, cambia al usuario root si no se especifica otro usuario.
* **Parámetros Comunes**:
  + -: Inicia una sesión de login como el usuario especificado, cargando su entorno completo.
  + -c COMMAND: Ejecuta un comando específico como el usuario especificado.

Ej: su - john

### ****groupadd****

* **Funcionalidad**:
  + Crea un nuevo grupo en el sistema.
* **Parámetros Comunes**:
  + -g GID: Especifica el GID del nuevo grupo.
  + -r: Crea un grupo del sistema con un GID reservado para grupos del sistema.
* **Ej:** groupadd -g 1001 developers

### ****who****

* **Funcionalidad**:
  + Muestra quién está actualmente conectado al sistema.
* **Parámetros Comunes**:
  + No suele requerir parámetros adicionales, aunque puede usar opciones como -u para mostrar información adicional.

Ej: who

### ****groupdel****

* **Funcionalidad**:
  + Elimina un grupo del sistema.
* **Parámetros Comunes**:
  + No requiere parámetros adicionales. El nombre del grupo a eliminar se especifica directamente.

Ej: groupdel developers

### ****passwd****

* **Funcionalidad**:
  + Cambia la contraseña de un usuario.
* **Parámetros Comunes**:
  + -l: Bloquea la cuenta (desactiva la contraseña).
  + -u: Desbloquea la cuenta (habilita la contraseña).
  + -d: Elimina la contraseña del usuario (deja la cuenta sin contraseña).

Ej: passwd john

5. FileSystem:

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

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

chmod

chown

chgrp

(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?

(d) ¿Existe la posibilidad de que algún usuario del sistema pueda acceder a determinado archivo para el cual no posee permisos? Nombrelo, y realice las pruebas correspondien

tes.

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

(f) ¿Con qué comando puede determinar en qué directorio se encuentra actualmente?

¿Existe alguna forma de ingresar a su directorio personal sin necesidad de escribir todo el path completo? ¿Podría utilizar la misma idea para acceder a otros directorios?

¿Cómo? Explique con un ejemplo.

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

cd

umount

mkdir

du

rmdir

df

mount

ln

ls

pwd

cp

mv

6. Procesos:

(a) ¿Qué es un proceso? ¿A qué hacen referencia las siglas PID y PPID? ¿Todos los procesos tienen estos atributos en GNU/Linux? Justifique. Indique qué otros atributos tiene un proceso.

(b) Indique qué comandos se podrían utilizar para ver qué procesos están en ejecución en un sistema GNU/Linux.

(c) ¿Qué significa que un proceso se está ejecutando en Background? ¿Y en Foreground?

(d) ¿Cómo puedo hacer para ejecutar un proceso en Background? ¿Cómo puedo hacer para pasar un proceso de background a foreground y viceversa?

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

(f) Redirección. ¿Qué tipo de redirecciones existen? ¿Cuál es su finalidad? Cite ejemplos de utilización.

(g) Comando kill. ¿Cuál es su funcionalidad? Cite ejemplos.

(h) Investigue la funcionalidad y parámetros de los siguientes comandos relacionados con el manejo de procesos en GNU/Linux. Además, compárelos entre ellos:

ps

kill

pstree

killall

top

nice

7. Otros comandos de Linux (Indique funcionalidad y parámetros):

(a) ¿A qué hace referencia el concepto de empaquetar archivos en GNU/Linux?

(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?

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

(d) ¿Pueden comprimirse un conjunto de archivos utilizando un único comando?

(e) Investigue la funcionalidad de los siguientes comandos:

tar

grep

gzip

zgrep

wc

8. 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

ps > PRUEBA

chmod 710 prueba

chown root : root PRUEBA

chmod 777 PRUEBA

chmod 700 /etc/passwd

passwd root

rm PRUEBA

man /etc/shadow

f ind / −name ∗.conf

usermod root −d /home/newroot −L

cd /root

rm ∗

cd /etc

cp ∗ /home −R

shutdown

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

(a) Terminar el proceso con PID 23.

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

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

(d) Guardar una lista de procesos en ejecución el archivo /home/<su nombre de usuario>/procesos

(e) Cambiar los permisos del archivo /home/<su nombre de usuario>/xxxx a:

Usuario: Lectura, escritura, ejecución

Grupo: Lectura, ejecución

Otros: ejecución

(f) Cambiar los permisos del archivo /home/<su nombre de usuario>/yyyy a:

Usuario: Lectura, escritura.

Grupo: Lectura, ejecución

Otros: Ninguno

(g) Borrar todos los archivos del directorio /tmp

(h) Cambiar el propietario del archivo /opt/isodata al usuario iso2010

(i) Guardar en el archivo /home/<su nombre de usuario>/donde el directorio donde me encuentro en este momento, en caso de que el archivo exista no se debe eliminar su contenido anterior.

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

(a) Ingrese al sistema como usuario “root”

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

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

(d) Crear un directorio en /tmp llamado cursada2017

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

(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.

(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.

(h) Acceda a otra terminal virtual para loguearse con el usuario antes creado.

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

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

(k) Verifiqué la cantidad de usuarios conectados al sistema.

(l) Vuelva a la terminal del usuario root, y envíele un mensaje al usuario anteriormente creado, avisándole que el sistema va a ser apagado.

(m) Apague el sistema.

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

(a) Cree un directorio cuyo nombre sea su número de legajo e ingrese a él.

(b) 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".

(c) 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

(d) 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. ¿Cuál es la razón por la cuál puede crear este archivo si ya existe un archivo llamado "LEAME.en este directorio?.

(e) ¿Qué comando utilizaría y de qué manera si tuviera que localizar un archivo dentro del filesystem? ¿Y si tuviera que localizar varios archivos con características similares?

Explique el concepto teórico y ejemplifique.

(f) Utilizando los conceptos aprendidos en el punto e), busque todos los archivos cuya extensión sea .so y almacene el resultado de esta búsqueda en un archivo dentro del directorio creado en a). El archivo deberá llamarse .ejercicio\_f".

12. 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:

mkdir iso

cd ./ iso ; ps > f0

l s > f1

cd /

echo $HOME

l s −l $> $HOME/iso/ls

cd $HOME; mkdir f2

l s −ld f2

chmod 341 f2

touch dir

cd f2

cd ~/iso

pwd >f3

ps | grep 'ps' | wc −l >> ../f2/f3

chmod 700 ../ f2 ; cd ..

f ind . −name etc/passwd

f ind / −name etc/passwd

mkdir ejercicio5

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

(a) Inicie 2 sesiones utilizando su nombre de usuario y contraseña. En una sesión vaya siguiendo paso a paso las órdenes que se encuentran escritas en el cuadro superior.

En la otra sesión, cree utilizando algún editor de textos un archivo que se llame .ejercicio10\_explicacion"dentro del directorio creado en el ejercicio 9.a) y, para cada una de las órdenes que ejecute en la otra sesión, realice una breve explicación de los resultados obtenidos.

(b) Complete en el cuadro superior 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 el inciso

9.a).

20: Copiar el resto de los archivos y directorios que se crearon en este ejercicio al directorio creado en el ejercicio 9.a).

(c) Ejecute las órdenes 19 y 20 y comentelas en el archivo creado en el inciso a).

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

(a) Utilizando la estructura de directorios anteriormente creada, indique que comandos son necesarios para realizar las siguientes acciones:

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

Copie el archivo "f4.en el directorio "dir11".

Haga lo mismo que en el inciso anterior pero el archivo de destino, se debe llamar

"f7".

Cree el directorio copia dentro del directorio usuario y copie en él, el contenido de

"dir1".

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

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

Renombre los archivos "f3"f4"de manera que se llamen "f3.exe "f4.exerespectivamente.

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

14. 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.

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

(c) Empaquete el directorio creado en 1, el archivo resultante se debe llamar "misLogs.tar".

(d) Empaquete y comprima el directorio creado en 1, el archivo resultante se debe llamar

"misLogs.tar.gz".

(e) Copie los archivos creados en 3 y 4 al directorio de trabajo de su usuario.

(f) Elimine el directorio creado en 1, logs.

(g) Desempaquete los archivos creados en 3 y 4 en 2 directorios diferentes.