2. Nombre al menos 3 editores de texto que pueda usar desde la terminal.
3. Nano
4. Vi o Vim (Vi improved)
5. Emacs
6. ¿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.

Estos comandos son utilidades que permiten visualizar el contenido de los archivos de texto, pero no son editores de texto.

1. cat (concatenate)

* Muestra el contenido completo de un archivo de texto en la terminal.
* No permite editar el texto, solo visualización.
* Usos comunes: mostrar archivos pequeños, concatenar varios archivos en uno solo

1. more

* Muestra el contenido de un archivo de texto, pero lo hace de a páginas, espera a que el usuario presione una tecla para avanzar.
* No permite editar el texto, solo visualización.
* Se puede salir de este modo presionando 'q'.

1. less

* Muestra el contenido de un archivo de texto, pero lo hace de a páginas.
* Permite moverte hacia arriba y abajo en el archivo, buscar texto y más.

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

* w: escribir cambios
* q o q!: salir del editor
* dd: cortar
* y: copiar
* p: pegar
* u: deshacer
* /fresa: busca "fresa" dentro del archivo

2. 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.
3. Se empieza a ejecutar el código del BIOS
4. El BIOS ejecuta el POST
5. El BIOS lee el MBR (sector de arranque)
6. Se carga el MBC (gestor de arranque)
7. El bootloader carga el kernel y el initrd
8. Se monta el initrd como sistema de archivos raíz y se inicializan componentes esenciales (scheduler)
9. El kernel ejecuta el proceso init y se desmonta el initrd
10. Se lee el /etc/inittab
11. Se ejecutan los scripts apuntados por el runlevel 1
12. El final del runlevel 1 le indica que vaya al runlevel por defecto
13. Se ejecutan los scripts apuntados por el runlevel por defecto
14. El sistema está listo para usarse
15. Proceso **INIT**. ¿Quién lo ejecuta? ¿Cuál es su objetivo?

No tiene padre, es el padre de todos los procesos.

Posee el PID1 y se encuentra en /sbin/init.

Es responsable de varias tareas clave:

* **Arranque del sistema**: carga el SO durante el boot (proceso de arranque).
* **Inicialización de servicios y demonios**: sistema de archivos, administrador de red, administrador de impresoras y otros servicios del sistema.
* **Gestión de cambio de runlevels**.
* **Reinicio y apagado del sistema**: se encarga de que los servicios se detengan adecuadamente antes de apagar o reiniciar.

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

Permite visualizar la jerarquía de procesos en el sistema en forma de un árbol, lo que permite ver que procesos están relacionados entre sí.

Cada proceso es un nodo, y los procesos secundarios (hijos) se muestran debajo de sus procesos principales (padres).

También separa los procesos por sesión de usuario, lo que ayuda a identificar las actividades de usuario en el sistema.

1. RunLevels. ¿Qué son? ¿Cuál es su objetivo?

Permiten iniciar un conjunto de procesos en el **boot** (arranque) o **shutdown** (apagado) del sistema. Cada uno es responsable de **levantar** (iniciar) o **bajar** (parar) una serie de servicios.

Se utilizan para definir diferentes **estados del sistema**, cada uno de los cuales tiene un conjunto específico de servicios y procesos que se ejecutan.

Su principal objetivo es gestionar y controlar la ejecución de servicios y procesos en el sistema, permitiendo cambios en la configuración del sistema en tiempo real sin necesidad de reiniciar.

1. ¿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?
2. halt (parada)
3. single user mode (monousuario)
4. multiuser, without NFS (multiusuario sin soporte de red)
5. full multiuser mode console (multiusuario completo por consola)
6. no se utiliza
7. X11 (multiusuario completo con login grafico basado en X)
8. Reboot

**SysV (System V)**: en sistemas que siguen esta convención, los archivos de configuración para definir qué nivel de ejecución se inicia al arrancar el SO suelen estar en etc/inittab.

**Systemd**: muchas distribuciones modernas siguen este sistema, y la configuración de runlevels se gestiona con el comando 'systemctl'.

No todas las distribuciones siguen la misma convención de niveles de ejecución, algunas pueden tener configuraciones y niveles personalizados. Ubuntu adopto Systemd, por ejemplo.

1. 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?

Es un **archivo de configuración**. Se utiliza para definir la configuración del sistema de inicio (boot) y controlar el proceso de inicialización. Solía ser esencial en sistemas SysV, pero su importancia disminuyó en sistemas más modernos que utilizan Systemd u otros.

**Define los runlevels**: se especifican los runlevels y que acciones tomar al ingresar a uno de ellos. Cada uno tiene asociado una serie de acciones y scripts de inicio y apagado.

**Configura la consola y el proceso de inicio**: se utilizaba para definir configuraciones de consola, como el mensaje de bienvenida, tipo de terminal y el programa de inicio de sesión.

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

SysV: se utiliza el comando init

init <Y>

init 3

Este cambio no es permanente, un cambio realizado mediante el comando init solo es temporal y afecta al estado actual del sistema. Una vez reiniciado el sistema, el SO volverá al runlevel predeterminado definido en la configuración /etc/inittab.

Para que el cambio sea permanente se tendría que editar la configuración del sistema de inicio.