**Practica 2 ISO**

**1. 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 código 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 (initial ram disk).  
6. Se monta el initrd como sistema de archivos raíz y se inicializan componentes esenciales (por ejemplo, el 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 está listo para ser usado

**b. Proceso INIT. ¿Quién lo ejecuta? ¿Cuál es su objetivo?**  
El proceso **init** es ejecutado por el **kernel de Linux** una vez que termina su carga y montaje del sistema de archivos raíz. Es el **primer proceso de espacio de usuario** que se inicia, y por eso se le asigna siempre el **PID 1**.

Su principal objetivo es **inicializar el sistema operativo**, gestionando el inicio de todos los **servicios, demonios y procesos necesarios** para que el sistema funcione correctamente. Esto incluye tareas como montar sistemas de archivos adicionales, configurar la red, iniciar el entorno gráfico (si corresponde) y lanzar el proceso de login para permitir que los usuarios se autentiquen en el sistema.

**c. RunLevels. ¿Qué son? ¿Cuál es su objetivo?**Este es el modo en que arranca GNULinux. Cada runlevel es responsable de iniciar o parar una serie de servicios, ya sea al entrar o al salir de este.  
Existen 7 (numerados del 0 al 6).  
El objetivo principal de los runlevels es **gestionar qué servicios se deben ejecutar en cada etapa del sistema**, permitiendo que el administrador controle fácilmente el comportamiento del arranque y el apagado del sistema

**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?**

• 0 → halt (parada o apagado).

• 1 → single-user mode (modo monousuario).

• 2 → multi-user without network support (multiusuario sin soporte de red).

• 3 → multi-user console mode (modo multiusuario en consola).

• 4 → N/A (no se utiliza).

• 5 → X11 (modo multiusuario con entorno gráfico basado en X.org).

• 6 → reboot (reinicio)

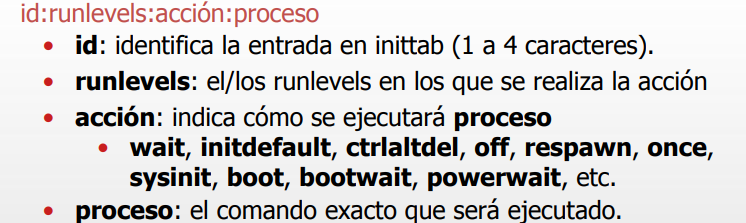
Se encuentran definidos en el archivo /etc/inittab y se inicializan mediante una línea como

id:3:initdefault:

Esto indica que el sistema debe iniciar en el runlevel 3 por defecto.

No todas las distribuciones respetan estrictamente estos valores. Algunas pueden asignarles otros significados o incluso no utilizar SysVinit, ya que muchas distribuciones modernas han migrado a sistemas de inicio más nuevos como systemd, que reemplaza a los runlevels por "targets".

**e. Archivo /etc/inittab. ¿Cuál es su finalidad? ¿Qué tipo de información se almacena en el? ¿Cuál es la estructura de la información que en él se almacena?**Tiene como finalidad configurar el comportamiento del proceso init



**f. Suponga que se encuentra en el runlevel <X>. Indique qué comando(s) deberá ejecutar para cambiar al runlevel <Y>. ¿Este cambio es permanente? ¿Por qué?**Se debe ejecutar el comando init Y  
**No**, el cambio **no es permanente**. Solo afecta a la sesión actual.  
Cuando el sistema se reinicie, volverá a iniciarse en el runlevel definido por defecto en el archivo

**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.**Los **scripts RC** son fundamentales para el inicio y apagado de los servicios en un sistema GNU/Linux. Su finalidad es gestionar los servicios que se deben iniciar o detener cuando el sistema cambia de runlevel. Estos scripts están almacenados principalmente en el directorio **/etc/init.d/**, que contiene los scripts de servicio, y en los directorios **/etc/rcX.d/**, donde X corresponde a cada runlevel del sistema (por ejemplo, rc3.d para el runlevel 3).

Cuando el sistema cambia de runlevel, el sistema verifica el directorio correspondiente al nuevo runlevel y ejecuta los scripts en ese directorio. Los scripts que se ejecutan para iniciar servicios tienen un prefijo S (de *Start*), mientras que los que se ejecutan para detener servicios tienen un prefijo K (de *Kill*). Además, cada script tiene un número al principio de su nombre que define el **orden de ejecución** de los mismos.

Por ejemplo, un script como **S10network** se ejecutará antes que **S20ssh**, lo que asegura que el servicio de red se inicie antes de que el servicio SSH se active. Esto garantiza que los servicios se gestionen en el orden correcto para asegurar que dependencias como la red estén disponibles antes de iniciar otros servicios que dependan de ella.

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

**a. ¿Qué es systemd?**SystemD es un sistema que centraliza la administración de servicios (demonios) y librerías del sistema, mejorando el paralelismo de arranque.  
Puede ser controlado con el comando systemctl.  
El demonio systemd remplaza al proceso init y es el que tiene PID 1  
Los runlevels son reemplazados por targets  
No utiliza el archivo de configuración /etc/inittab

**b. ¿A qué hace referencia el concepto de Unit en SystemD?**En systemd, una Unit (unidad) es un archivo de configuración que describe cómo gestionar un recurso del sistema. Cada Unit representa algo que puede ser iniciado, detenido o supervisado por systemd, como servicios, dispositivos, puntos de montaje, sockets, etc.

**c. ¿Para qué sirve el comando systemctl en SystemD?**El comando **systemctl** es la herramienta principal para **gestionar servicios y unidades en sistemas que usan systemd**. Permite iniciar, detener, reiniciar y consultar el estado de los servicios, así como habilitarlos o deshabilitarlos para que se inicien automáticamente al arrancar el sistema.

**d. ¿A qué hace referencia el concepto de target en SystemD?**  
En **systemd**, un **target** es una unidad especial que **agrupa otras unidades** (como servicios, sockets, mounts, etc.) para facilitar su gestión en conjunto. Los targets **reemplazan a los runlevels** tradicionales de SystemV y permiten definir qué servicios deben estar activos en determinado estado del sistema

**e. Ejecutar el comando pstree. ¿Qué es lo que se puede observar a partir de la ejecución de este comando?**El comando pstree muestra de forma visual y jerárquica todos los procesos que están corriendo en el sistema, organizados como un árbol según quién los creó. En este árbol, el proceso systemd (o init en sistemas más antiguos) aparece en la raíz con PID 1, y de él cuelgan todos los demás procesos.  
  
**3. 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 guarda principalmente en los siguientes archivos:**

* **/etc/passwd**: contiene los datos básicos de cada usuario, como el nombre de usuario, UID, GID, directorio home y el intérprete de comandos (shell).
* **/etc/shadow**: almacena las contraseñas cifradas de los usuarios y datos relacionados con la seguridad, como la fecha del último cambio de contraseña o su vencimiento.
* **/etc/group**: contiene información sobre los grupos del sistema, incluyendo el nombre del grupo, GID y los usuarios que pertenecen a él.

**b. ¿A qué hacen referencia las siglas UID y GID? ¿Pueden coexistir UIDs iguales en un sistema GNU/Linux? Justifique.**  
**UID (User ID)**: es el identificador numérico único que representa a cada usuario en el sistema.

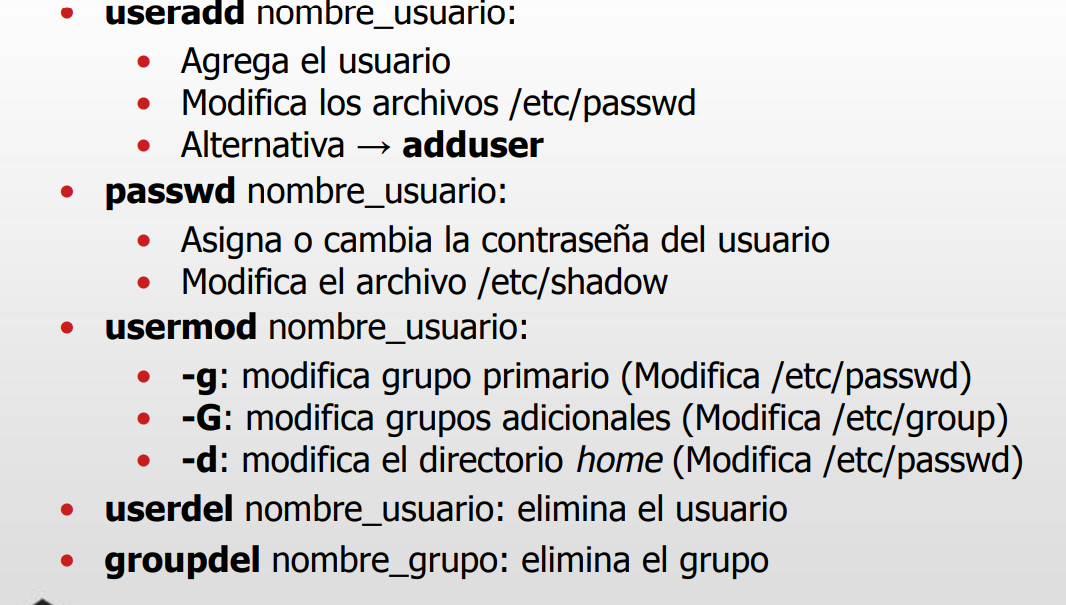
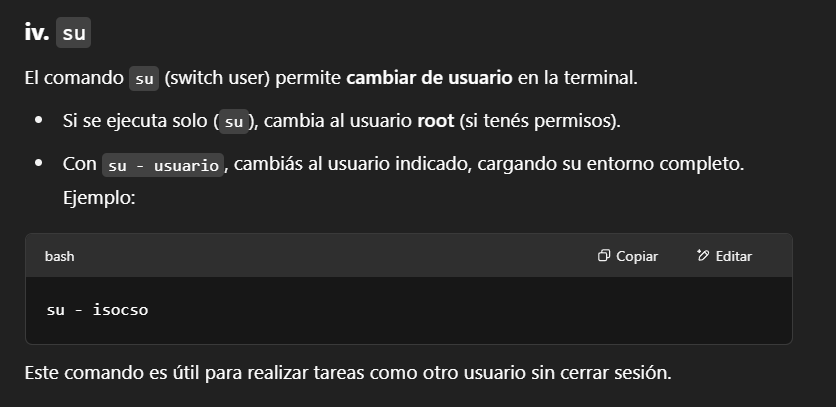
**GID (Group ID)**: es el identificador numérico que representa a un grupo al que pertenece un usuario.

Puede pero no es lo recomendable

**c. ¿Qué es el usuario root? ¿Puede existir más de un usuario con este perfil en GNU/Linux? ¿Cuál es la UID de root?**El **usuario root** es el **administrador del sistema** en GNU/Linux, con **acceso total** a todos los archivos, comandos y configuraciones. Puede ejecutar cualquier acción sin restricciones, por eso se lo conoce como el **superusuario**.

La **UID del root es 0**, y esa es la que el sistema utiliza para identificar al usuario con privilegios absolutos.

**Sí puede haber más de un usuario con perfil de root**, asignándole también la UID 0. Sin embargo, esto **no es recomendable**, ya que complica el control de quién realizó qué acción, afectando la seguridad y la trazabilidad del sistema.

**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.  
1- Crear grupo si no existe**sudo groupadd informática  
**2- agregar al usuario isocso y su directorio personal**sudo useradd -m -d /home/isocso -g infomatica isocso  
**3- Crear un archivo en su home sin iniciar sesión como el**  
sudo touch /home/isocso/bienvenida.txt  
sudo chown isocso:informatica /home/isocso/bienvenida.txt  
**4- Borrar el usuario**sudo userdel -r isocso  
**5- verificar que no quede nada**  
grep isocso /etc/passwd /etc/shadow /etc/group /etc/gshadowe. Investigue la funcionalidad y parámetros de los siguientes comand  


**v. groupadd**crea un nuevo grupo en el sistema  
groupadd nombre

**vi. who**Muestra una lista de los usuarios que están actualmente conectados al sistema. Incluye información como el nombre del usuario, terminal, fecha y hora de inicio de sesión.  
Es útil para monitorear sesiones activas.  
  
***4. FileSystem y permisos:***

**a. ¿Cómo son definidos los permisos sobre archivos en un sistema GNU/Linux?   
Los permisos son definidos mediante una notación octal  
**  
Existen 3 posibles permisos R, W y X  
Los permisos se definen sobre las siguientes agrupaciones de usuarios  
Usuarios, Grupo y otros

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

**i. chmod:** Permite gestionar los permisos de un archivo o directorio  
chmod 755 /tmt/script (asignando RWX a usuarios RX a grupos y otros

**ii. chown:** Se usa para cambiar el dueño y o el grupo de un archivo/directorio  
sudo chown Nahuel documento.txt (asigna a Nahuel el archivo documento.txt)  
sudo chown Nahuel:usuarios documento.txt (cambia dueño y grupo)  
se usa sudo si no sos el dueño del archivo

**iii. chgrp:** Cambia solo el grupo propietario de un archivo o directorio  
chgrp nuevo\_grupo archivo

**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?**   
Significa que se puede definir los permisos con números en lugar de letras.  
Los permisos se dividen en 3 grupos  
-1er numero usuario  
-2do numero grupo  
-3er numero others  
read valor 4  
write valor 2  
excetue valor 1  
chmod 755 archivo.sh (7 (rwx) para el usuario) (5 r-x para el grupo y otros)

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.   
Casos en los que se puede acceder:  
-sos un usuario root  
-tenes permisos de grupo

**e. Explique los conceptos de “full path name” y “relative path name”. De ejemplos claros de cada uno de ellos.**full path name: es la llamada ruta absoluta, es la dirección completa de un archivo o directorio desde la raíz del sistema. Ej: /home/nahu/documentos/tp1.txt  
relative path name: Es la ruta relativa a tu ubicación actual en el sistema, no empieza con /. Ej: documentos/tp1.txt

**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.**Se puede determinar en que directorio te encontras con pwd  
Se puede ingresar a su directorio personal sin necesidad de escribir todo con cd ~prueba (te lleva a /home/prueba si existe en ese usuario)

**g. Investigue la funcionalidad y parámetros de los siguientes comandos relacionados con el uso del FileSystem:**  
**i. umount:** desmonta un sistema de archivos que ya fue montado  
umount /media/usb  
-l desmontaje lento  
-f forzar el desmontaje  
  
**ii. du:** Muestra el uso de espacio de archivos y directorios  
du -h /home/nahu  
-h muestra tamaños en formato legible  
-s resumen total  
  
**iii. df:** muestra el espacio disponible en los sistemas de archivos montados  
df -h  
  
**iv. mount:** monta un dispositivo o sistema de archivos  
mount /dev/sbd1 /mnt/usb  
-t tipo: especifica el tipo  
-o opciones como ro solo lectura etc.  
  
**vi. mfks:** crea un nuevo sistema de archivos (formatea una partición)   
mfks.ext4 /dev/sdb1  
  
**vii fdisk:** herramienta para editar particiones de discos  
sudo fdisk /dev/sdb  
-l lista todas las particiones.  
  
**viii. write:** envia un mensaje a otro usuario logueado en el sistema  
write nahu  
se termina el msj con ctrl+d  
  
**ix: losetup:** asocia archivos a dispositivos de loop  
  
**x. stat:** muestra info detallada de una archivo (mas que ls)  
  
**5. Procesos:  
¿Qué significa que un proceso se está ejecutando en Background? ¿Y en Foreground?**Background: Un proceso se ejecuta cuando no interactua directamente con el usuario. El proceso se ejecuta de manera independiente permitiendo al usuario continuar con otras tareas  
Foreground: Un proceso es aquel que esta directamente en interacción con el usuario. Este proceso toma el control de la terminal y no deja hacer otras tareas hasta que termine su ejecución.

**b. ¿Cómo puedo hacer para ejecutar un proceso en Background? ¿Como puedo hacer para pasar un proceso de background a foreground y viceversa?**   
Background a foregroound:  
se agrega un & al final del comando

**c. Pipe ( | ). ¿Cuál es su finalidad? Cite ejemplos de su utilización.**  
El comando pipe se utiliza para canalizar la salida de un comando como entrada de otro. Ej:  
para listar los archivos y directorios de un directorio y luego buscar una rchivo especifico  
ls | grep “archivo”  
este comando lista los archivos en el directorio y pasa esta salida al grep para buscar los cuales contengan la palabra “archivo”

**d. Redirección. ¿Qué tipo de redirecciones existen? ¿Cuál es su finalidad? Cite ejemplos de utilización.**  
La redirección se utiliza para cambiar el flujo de entrada o salida de los comandos, permitiendo que la salida de un comando se guarde en un archivo o que un archivo se use como comando de entrada  
Tipos de redirección:  
redirección de salida (>): Redirige la salida de un comando a un archivo, sobrescribiendo el contenido del archivo si existe.  
echo “hola” > archivo.txt  
redirección de salida agregada (>>): En lugar de sobrescribir, agrega la salida al final del archivo  
echo ”otro mensaje” >> archivo.txt  
redirección de entrada (<): redirige la entrada de un comando desde un archivo en lugar de la entrada estándar (teclado)  
sort < archivo.txt  
ordena todas las palabras del archivo.txt ya que se le pasa por parámetro”  
  
6. Otros comandos de Linux (Indique funcionalidad y parámetros):

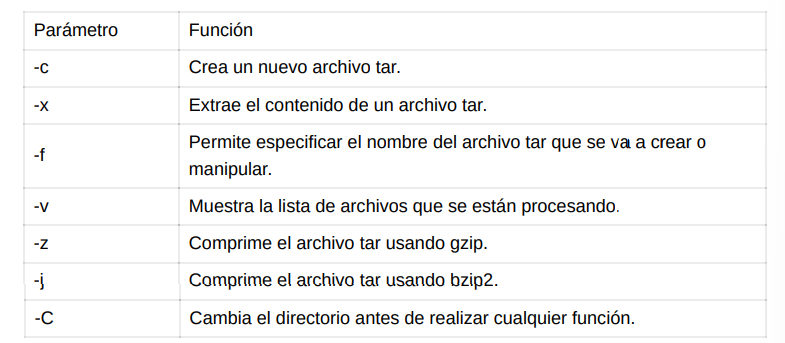
a. ¿A qué hace referencia el concepto de empaquetar archivos en GNU/Linux?  
Empaquetar hace referencia a la acción de crear un archivo que contenga uno o varios archivos o directorios comprimidos en un solo archivo. Hay varios formatos como tar, zip, gzip.

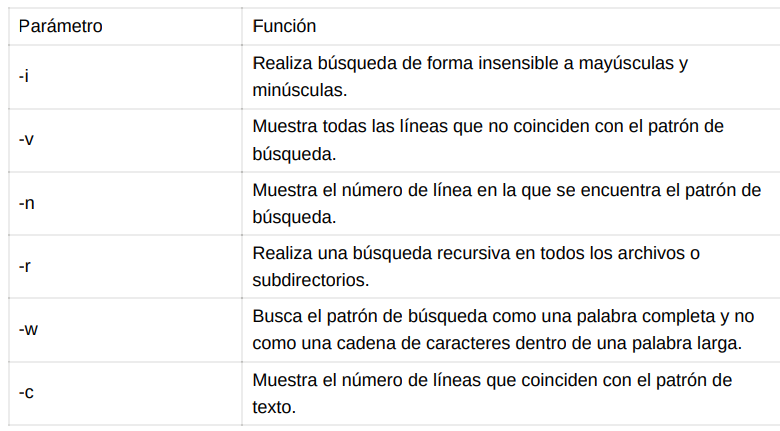
b. Seleccione 4 archivos dentro de algún directorio al que tenga permiso y sume el tamaño de cada uno de estos archivos. Cree un archivo empaquetado conteniendo estos 4 archivos y compare los tamaños de los mismos. ¿Qué característica nota?   
  
du -ch archivo1 archivo2 archivo3 archivo4 #suma tamaños   
tar -czvf archivos.tar.gz archivo1 archivo2 archivo3 archivo4 #empaqueta   
ls -lh archivo1 archivo2 archivo3 archivo4 archivos.tar.gz #compara tamaños

c. ¿Qué acciones debe llevar a cabo para comprimir 4 archivos en uno solo? Indique la secuencia de comandos ejecutados.  
tar -czvf archivo.tar.gz archivo1 archivo2 archivo3 archivo4   
#-c indica que se creará un nuevo archivo  
#-z indica que se utilizará compresión gzip   
#-v indica que se mostrará información detallada del proceso en la terminal   
#-f indica que se especificará el nombre del archivo a crear

d. ¿Pueden comprimirse un conjunto de archivos utilizando un único comando?   
Si, con el inciso c  
(e) Investigue la funcionalidad de los siguientes comandos:

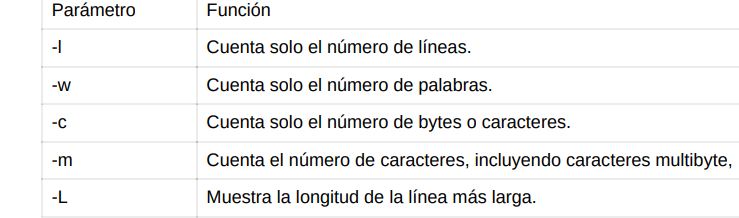
i. tar: crea manipula y desempaqueta archivos empaquetados



ii. grep: busca patrones de texto en uno o varios archivos  


iii. gzip: Comprime archivos en formato GNU.

iv. zgrep: es similar a grep pero está diseñado para buscar patrones en archivos comprimidos con gzip. La funcionalidad es la misma que grep pero en vez de leer un archivo de texto plano, puede leer un archivo comprimido con gzip y buscar un patrón en el contenido del archivo.

v. wc: cuenta el número de líneas, palabras y caracteres de un archivo o entrada de texto.  


7. 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:   
ls −l > prueba lista el contenido de forma detallada y lo redirecciona a prueba  
ps > PRUEBA almacena el estado de los procesos en el archivo PRUEBA  
chmod 710 prueba cambia los permisos de prueba  
chown root:root PRUEBA falla, intenta cambia el propietario de PRUEBA a root   
chmod 777 PRUEBA cambia los permisos  
chmod 700 /etc/passwd falla  
passwd root falla, intenta cambiar contraseña de root  
rm PRUEBA elimina PRUEBA  
man /etc/shadow man abre manual, pero falla ya q no tiene manual  
find / −name ∗ .conf busca archivos con .conf desde la raíz  
usermod root −d /home/ newroot −L   
cd /root intenta entrar al directorio de root pero no tiene permisos   
rm ∗ borra todos los archivos del directorio actual  
cd /etc cambia el directorio   
cp ∗ /home −R copia todo lo del directorio recursivameente  
shutdown apaga el sistema  
  
8. Indique qué comando sería necesario ejecutar para realizar cada una de las siguientes acciones:

a. Terminar el proceso con PID 23.   
kill 23

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

c. Buscar todos los archivos de usuarios en los que su nombre contiene la cadena “.conf”  
find /home -name “\*\.conf!  
  
d- Guardar una lista de procesos en ejecución el archivo /home/<su nombre de usuario>/procesos   
ps aux > /home/nahu/procesos

e. Cambiar los permisos del archivo /home/<su nombre de usuario>/xxxx a:   
i. Usuario: Lectura, escritura, ejecución   
ii.Grupo: Lectura, ejecución   
iii.Otros: ejecución  
 chmod 751 /home/iso/xxxx

f. Cambiar los permisos del archivo /home/<su nombre de usuario>/yyyy a:   
i. Usuario: Lectura, escritura.   
Grupo: Lectura, ejecución   
Otros: Ninguno  
chmod 650 /home/nahu/yyyy

e. Borrar todos los archivos del directorio /tmp   
rm /tmp\*

g. Cambiar el propietario del archivo /opt/isodata al usuario iscocso   
chown isocso /opt/isodata

h. 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.

pwd >> /home/iso/donde  
  
9. Indique qué comando sería necesario ejecutar para realizar cada una de las siguientes acciones:

a. Ingrese al sistema como usuario “root”  
su 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.  
adduser ntranquillini

c. ¿Qué archivos fueron modificados luego de crear el usuario y qué directorios se crearon?   
creado: /home/ntranquillini  
modificado:   
/etc/passwd  
 /etc/shadow   
/etc/group   
/etc/gshadow

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

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 ntranquillini:users /tmp/miCursada

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 764 /tmp/MiCursaa

h. Acceda a otra terminal 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.   
ps -e | wc -l

k. Verifiqué la cantidad de usuarios conectados al sistema.   
who | wc -l  
who -q

l. Vuelva a la terminal del usuario root y envíele un mensaje al usuario anteriormente creado enviándole que el sistema va a ser apagado.  
write ntranquillini “El sistema se apagara”

m. Apague el sistema.  
shutdown  
  
10. 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.  
mkdir 21154-7  
cd 21154-7

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".  
vi 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   
chmod 017 LEAME

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?  
ls -l /etc > ~/leame  
no se puede crear este archivo si ya existe una rchivo llamado LEAME en ese directorio ya que es sensible a mayúsculas y minusculas

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.  
se usaría el comando find  
find[ruta] -name [nombre\_del\_archivo]   
Ej: buscar varios archivos similares (ej todos los .log)  
find / -name “\*.log”

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 ejercicioF.  
find / -name “\*.so” > /home/21154-7/ejerciciof  
  
11. 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 crea el directoiro ISO  
cd . /iso cambia el directorio actual  
ps > f0 guarda en f0 los procesos ejecutandose  
ls > f1 guarda en f1 el listado del directorio actual  
cd / vuelve a la raiz  
echo $HOME muestra el path completo de la raiz  
ls −l $> $HOME/ iso/ls redirige al path %HOME/iso/ls el listado detallado del contenido del directorio  
cd $HOME vuelve al diectorio raiz  
mkdir f2 crea f2  
ls −ld f2 muestra el contenido detallando   
chmod 341 f2 cambia los permisos de f2  
touch dir imprime dir  
cd f2 se ubica en f2  
cd ~/iso se ubica en isoo  
pwd > f3 guarda en f3 el directorio actual  
ps | grep 'ps' | wc −l >> ../f2/f3 en f2/f3 guarda las líneas que no coinciden con el patron de grep  
chmod 700 ../f2   
 cd ..   
find . −name etc/passwd busca en el directorio actual todos los archivos q contengan etc/passwd  
find / −name etc/passwd busca en la raíz todos los archivos que contengan el nombre etc/passwd  
mkdir ejercicio5 crea la carpeta

13. 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.  
mkdir /tmp/logs  
b. Copie todo el contenido del directorio /var/log en el directorio creado en el punto anterior.  
cp -r /var/log/\* /tmp/logs/  
c. Empaquete el directorio creado en 1, el archivo resultante se debe llamar "misLogs.tar".  
tar -cvf misLogs.tar   
d. Empaquete y comprima el directorio creado en 1, el archivo resultante se debe llamar   
"misLogs.tar.gz".  
tar -czvf misLogs.tar.gz /tmp/logs/   
e. Copie los archivos creados en 3 y 4 al directorio de trabajo de su usuario.  
cp misLogs.tar.gz misLogs.tar /home/usuario

f. Elimine el directorio creado en 1, logs.  
rm -r /tmp/logs/

g. Desempaquete los archivos creados en c y d en 2 directorios diferentes.  
tar -xvf misLogs.tar -C /home/usuario/n1   
gzip -d misLogs.tar.gz -C /home/usuario/n2/

**14. ¿Qué es la MMU (Unidad de Gestión de Memoria, por sus siglas en inglés: *Memory Management Unit*)** y qué funciones cumple?  
es un componente del procesador o del sistema que se encarga de **gestionar y traducir las direcciones de memoria virtual a física**  
   
**15. ¿Que representa el espacio de direcciones de un proceso?**El **espacio de direcciones de un proceso** representa el conjunto de direcciones de memoria que ese proceso puede utilizar durante su ejecución. Es una visión lógica de la memoria desde el punto de vista del proceso.  
  
**16. Explique y relacione los siguientes conceptos: Dirección Lógica o Virtual Dirección Física**Direccion Logica (o virtual): Es la dirección de memoria que genera un programa cuando se esta ejecutando. Es manejada por el MMU  
Direccion física: Es la ubicación real en la memoria ram, es la dirección que usa el hardware para acceder a los datos  
Estas direcciones pueden ser traducidas de unos a otros.

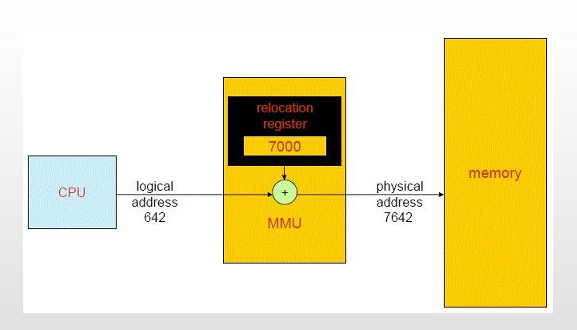
**17. En la técnica de Particiones Múltiples, la memoria es dividida en varias particiones y los espacios de direcciones de los procesos son ubicados en estas, siempre que el tamaño del mismo sea menor o igual que el tamaño de la partición. Al trabajar con particiones se pueden considerar 2 métodos (independientes entre sí): Particiones Fijas   
Particiones Dinámicas  
  
a. Explique cómo trabajan estos 2 métodos. Cite diferencias, ventajas y desventajas.  
Particiones Fijas**

En este método, la memoria se divide en bloques de tamaño fijo cuando se inicia el sistema. Cada una de estas particiones puede contener un solo proceso, siempre y cuando el tamaño del proceso sea menor o igual al de la partición. Si el proceso es más pequeño, el espacio restante dentro de la partición se pierde, generando lo que se llama **desperdicio interno**. Si es más grande, no puede ser cargado en esa partición.

Su principal ventaja es la **simplicidad** de implementación y la **baja sobrecarga** para el sistema operativo. Sin embargo, su desventaja más importante es que **no se adapta al tamaño real** de los procesos, lo que puede generar un uso muy ineficiente de la memoria.

**Particiones Dinámicas**  
A diferencia del anterior, este método no divide la memoria al inicio, sino que las particiones se van creando dinámicamente a medida que se van cargando procesos. Cada proceso recibe una partición del tamaño justo (o lo más cercano posible) según lo que necesita.

Esto reduce el desperdicio interno, ya que se ajusta mejor al tamaño real de cada proceso. Sin embargo, con el tiempo puede generar **fragmentación externa**, es decir, pequeños huecos de memoria dispersos que no pueden ser aprovechados por procesos nuevos. Además, requiere mayor trabajo del sistema operativo, ya que tiene que ir buscando huecos adecuados e incluso a veces compactar la memoria para juntar esos espacios libres.

**b. ¿Qué información debe disponer el Kernel para poder administrar la memoria con estos métodos?**El kernel necesita saber si cada partición está libre u ocupada, la dirección base y el tamaño de cada una. También debe conocer el tamaño del proceso que quiere cargar, para ver si entra en alguna partición (fija o dinámica). En particiones dinámicas, además, debe llevar una lista actualizada de los huecos libres. Por último, tiene que usar un algoritmo de asignación como First Fit o Best Fit para decidir dónde ubicar el proceso.  
  
**c. Realice un gráfico indicando cómo se realiza la transformación de direcciones lógicas a direcciones físicas.**  
  
**18. Al trabajar con particiones fijas, los tamaños de las mismas se pueden considerar: ➢ Particiones de igual tamaño.   
➢ Particiones de diferente tamaño. Cite ventajas y desventajas entre las alternativas**.  
 **Particiones de igual tamaño**

En este caso, todas las particiones tienen el mismo tamaño. Es más fácil de implementar y administrar, ya que el sistema solo necesita saber cuántas particiones hay y si están libres u ocupadas. Sin embargo, puede haber mucho **desperdicio interno** si los procesos son más chicos que la partición, ya que el espacio sobrante no se puede usar.

**Particiones de diferente tamaño**

Acá las particiones tienen distintos tamaños, lo que permite asignar procesos a particiones más ajustadas a su tamaño real. Esto **reduce el desperdicio interno** y mejora el uso de la memoria. Como desventaja, es **más complejo de administrar**, porque el kernel debe decidir en qué partición ubicar cada proceso y puede haber fragmentación si no se elige bien.  
  
**19. Fragmentación. Ambos métodos de particiones presentan el problema de la fragmentación:   
Fragmentación Interna (Para el caso de Particiones Fijas)   
Fragmentación Externa (Para el caso de Particiones Dinámicas)   
a. Explique a qué hacen referencia estos 2 problemas**  
**Fragmentación Interna (Particiones Fijas)**

Este tipo de fragmentación ocurre cuando el proceso que se carga en una partición es más pequeño que el tamaño de la misma. El espacio que sobra dentro de la partición **no puede ser utilizado por ningún otro proceso**, y por eso se desperdicia. Cuanto más grande sea la diferencia entre el tamaño del proceso y el de la partición, mayor será la fragmentación interna.

**Fragmentación Externa (Particiones Dinámicas)**

En las particiones dinámicas, como los procesos van entrando y saliendo, **quedan espacios libres dispersos en la memoria**. Aunque la memoria total libre sea suficiente para un proceso nuevo, si esa memoria está dividida en bloques pequeños no contiguos, el proceso **no puede ser cargado**, generando fragmentación externa.

**b. El problema de la Fragmentación Externa es posible de subsanar. Explique una técnica que permita mitigar este problema.**La fragmentacione xterna es posible de subsanar usando la técnica de COMPACTACION  
Consiste en mover los procesos en memoria para **juntar todos los espacios libres en un solo bloque continuo**, liberando así un área de memoria lo suficientemente grande para cargar procesos nuevos. Aunque es efectiva, tiene un costo en tiempo y procesamiento, por lo que no siempre se hace de forma constante.

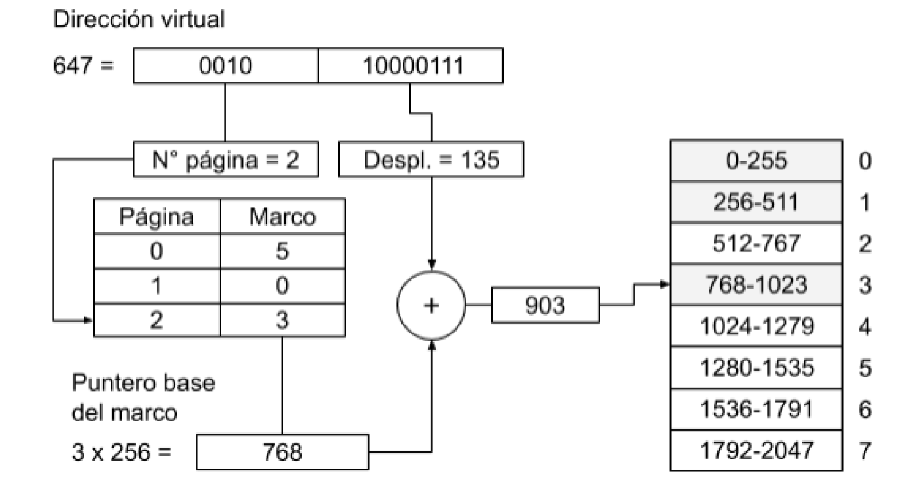
**20. Paginación**

**a. Explique cómo trabaja este método de asignación de memoria.**   
La **paginación** es una técnica que divide la memoria tanto del proceso como de la memoria física en bloques de tamaño fijo, llamados **páginas** (en la memoria lógica) y **marcos de página** (en la memoria física). Cuando un proceso necesita más memoria, se le asignan estas páginas, que no tienen que ser contiguas en la memoria física. El sistema mantiene una tabla llamada **tabla de páginas** que relaciona las páginas lógicas con los marcos de memoria física donde se encuentran.

De esta manera, se evita la necesidad de que los procesos se carguen en bloques contiguos y se facilita la gestión de la memoria. La dirección lógica generada por un proceso se divide en dos partes: la **número de página** y el **desplazamiento dentro de la página**. El número de página se usa para consultar la tabla de páginas y obtener la dirección física correspondiente.

Tamaño de pagina = tamaño de marco 512 bytes

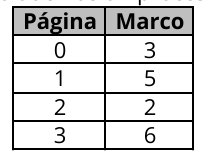
**b. ¿Qué estructuras adicionales debe poseer el Kernel para llevar a cabo su implementación?**   
-Tabla de paginas  
-Directorio de paginas  
-Registros de memoria física  
-Unidad de gestión de memoria MMU

**c. Explique, utilizando gráficos, cómo son resueltas las direcciones lógicas** **en físicas.**  
 

**d. En este esquema: ¿se puede producir fragmentación (interna y/o externa)?**Se puede producir fragmentación interna pero no tan importante como en las particiones fijas.

**21. Cite similitudes y diferencias entre la técnica de paginación y la de particiones fijas.**Similitudes: Pueden generar fragmentación interna, ambas dividen a la memoria en fracciones de tamaño fijo

Diferencias:  
-La paginación divide al proceso en varias particiones de tamaño fijo en donde pueden haber mas de un proceso, en cambio en las particiones fijas dividen la memoria en bloques fijos y cada uno de estos es asociado a un proceso.  
  
**22. Suponga un sistema donde la memoria es administrada mediante la técnica de paginación, y donde:   
El tamaño de la página es de 512 bytes   
Cada dirección de memoria referencia 1 byte.   
Los marcos en memoria principal se encuentran desde la dirección física 0.   
Suponga además un proceso con un tamaño 2000 bytes y con la siguiente tabla de páginas:**



**a. Realice los gráficos necesarios (de la memoria principal, del espacio de direcciones del proceso y de tabla de páginas) en el que refleje el estado descrito.  
  
b. Indicar si las siguientes direcciones lógicas son válidas y en caso afirmativo indicar la dirección física a la que corresponden:**

**i. 35**

**ii. 512**

**iii. 2051**

**iv. 0**

**v. 1325**

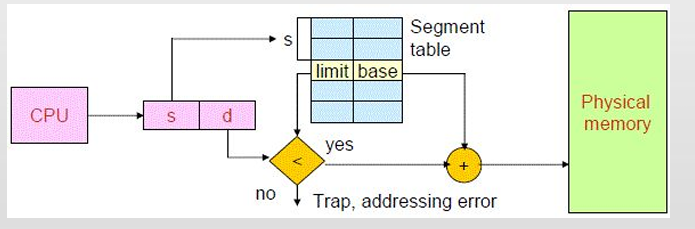
**vi. 602**

**c. Indicar, en caso de ser posible, las direcciones lógicas del proceso que se corresponden a las siguientes direcciones físicas:   
i. 509   
ii. 1500  
iii.0   
iv.3215   
v. 2014  
vi. 2000  
d. Indique, en caso que se produzca, la fragmentación (interna y/o externa)**

**24. Segmentación**

**a. Explique cómo trabaja este método de asignación de memoria.**La segmentación e sun método de asignación de memoria en el que la memoria lógica de un proceso se divide en segmentos que representan unidades lógicas.  
Cada segmento tiene tamaño variable y su propia dirección base en memoria física.

**b. ¿Qué estructuras debe mantener el Kernel para llevar a cabo su implementación?**El proceso tiene una tabla de segmentos donde cada entrada indica:  
-Direccion base (inicio del segmento en memoria física)  
-Tamaño del segmento

**c. Explique, utilizando gráficos, cómo son resueltas las direcciones lógicas a físicas**  
**se accede a la tabla de segmentos usando s y se buscan base y limite  
se verifica q el desplazamiento d sea menor que el limite del segmento, sino ejecuta trap  
si es valido se calcula la dirección física.  
la segmentación protege y traduce direcciones**, garantizando que cada proceso acceda solo a sus propios segmentos, y que no se pase de los límites.

**d. En este esquema: ¿se puede producir fragmentación (interna y/o externa)?**Si, puede ocurrir fragmentación EXTERNA. Ya que los segmentos tienen tamaño variable y se ubican en espacios contiguos, con el tiempo se puede generar huecos entre segmentos al liberar y asignar memoria.

**25. Cite similitudes y diferencias entre la técnica de segmentación y la de particiones dinámicas.**Similitudes:  
-Ambos trabajan con bloques de memoria de tamaño variable y los asignan de manera contigua.  
-En ambos casos pueden aparecer fragmentación externa  
-Ambos necesitan una tabla para llevar un control de que partes de la memoria están ocupadas y cuales están libres

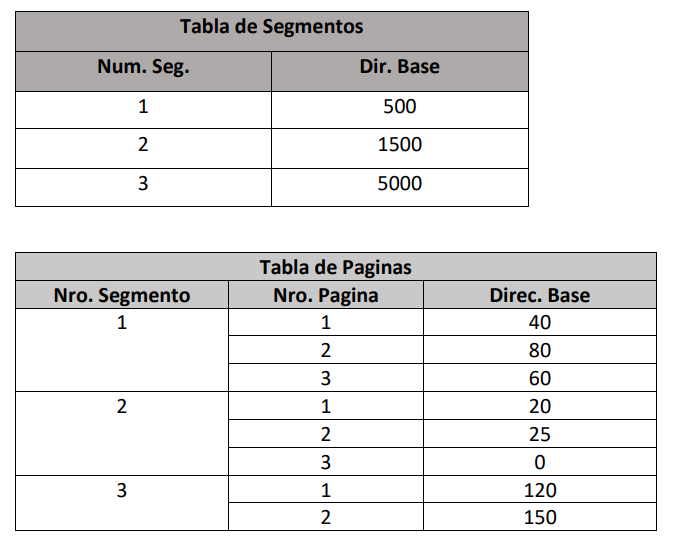
Diferencias:  
-En la segmentación divide un programa en partes lógicas, un solo proceso puede tener varios bloques separados en memoria. En cambio en las dinámicas, se asigna un solo bloque contiguo para todo el proceso completo  
-En la traducción en la segmentación la DL tiene dos componentes, el numero de segmento y el desplazamiento dentro de ese segmento. En particiones dinámicas se usa una única dirección base por proceso y se le suma un desplazamiento  
  
**26. Cite similitudes y diferencias entre la técnica de paginación y segmentación.  
Similitudes:**-Permiten que el proceso no tenga que estar completamente cargado de forma contigua en memoria física.  
-Requieren estructuras especiales  
-Ambas permiten proteger y aislar distintas partes del proceso  
-Ambas traducen direcciones lógicas a físicas  
  
Diferencias:  
-En paginación, la memoria lógica se divide en bloques del mismo tamaño llamado paginas, en cambio en la segmentación se divide en partes lógicas de distinto tamaño  
-Las paginas son de tamaño fijo las cuales pueden generar fragmentación interna. Los segmentos son de tamaño variable y pueden causar fragmentación externa.  
-En paginación las paginas pueden estar ubicadas en cualquier lugar de la memoria física. En segmentación también pero cada segmento debe estar completo y contiguo.

**27. Dada la técnica de administración de memoria por medio de segmentación paginada:**

**a. Explique cómo funciona.**

**b. Cite ventajas y desventajas respecto a las técnicas antes vistas.**

**c. Teniéndose disponibles las siguientes tablas:**



**Indicar las direcciones físicas correspondientes a las siguientes direcciones lógicas (segmento,pagina,desplazamiento):**

i) (2,1,1) = 1500 + 20 + 1 = 1521

ii) (1,3,15) = 500 + 60 + 15 = 575

iii) (3,1,10) = 500 + 120 + 10 = 5130

iv) (2,3,5) = 1500 + 0 + 5 = 1505