# Practica 2 ISO

**1. Editor de textos:**

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

**Vim, Vi, Nano y Emacs.**

**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.**

Un editor de texto es una herramienta interactiva para crear y editar archivos de texto, mientras que los comandos cat, more y less son herramientas para ver y manipular el contenido de archivos de texto en la terminal, pero no te permiten realizar cambios en el archivo directamente desde la terminal.

Modo Insert (Ins o i), Modo Visual (v) y Modo de Ordenes o Normal (Esc).

**c) 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 al portapapeles
* p: pegar desde el portapapeles
* u: deshacer
* /frase: busca \frase" dentro del archivo

**2. Proceso de Arranque *SystemV***

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.

* Se empieza a ejecutar el código del BIOS
* El BIOS ejecuta el POST
* El BIOS lee el sector de arranque (MBR)
* Se carga el gestor de arranque (MBC)
* El bootloader carga el kernel y el initrd
* Se monta el initrd como sistema de archivos raíz y se inicializan componentes esenciales (ej.: scheduler)
* El Kernel ejecuta el proceso init y se desmonta el initrd
* Se lee el /etc/inittab
* Se ejecutan los scripts apuntados por el runlevel 1
* El final del runlevel 1 le indica que vaya al runlevel por defecto
* Se ejecutan los scripts apuntados por el runlevel por defecto
* El sistema está listo para usarse

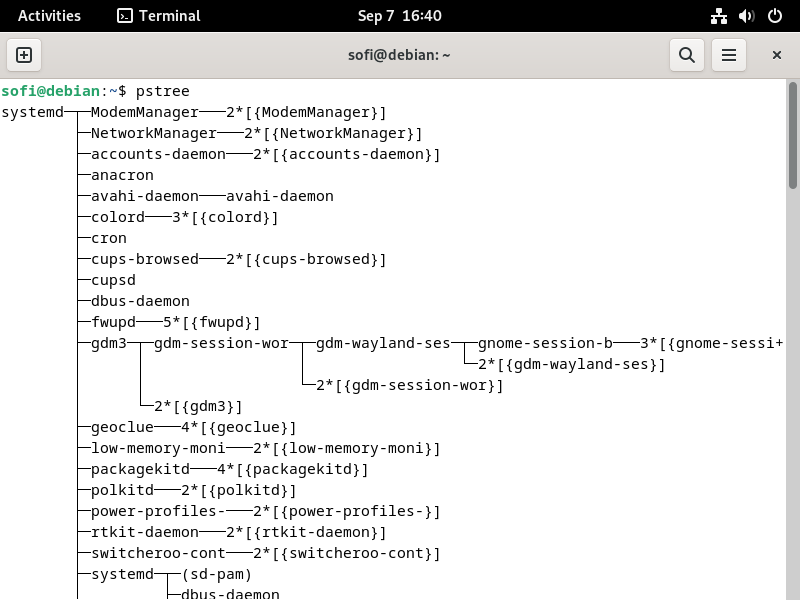
**b) Proceso INIT. ¿Quién lo ejecuta? ¿Cuál es su objetivo?**

Su función es cargar todos los subprocesos necesarios para el correcto funcionamiento del SO. El proceso init posee el PID 1 y se encuentra en /sbin/init.En SysV se lo configura a través del archivo /etc/inittab. No tiene padre y es el padre de todos los procesos (pstree)

**c) Ejecute el comando pstree. ¿Qué es lo que se puede observar a partir de la ejecución**

**de este comando?**

El comando pstree en Linux se utiliza para mostrar la estructura de procesos en forma de un árbol. Proporciona una representación visual de los procesos en ejecución en el sistema y muestra cómo están relacionados entre sí, incluyendo el proceso principal (init o systemd) como raíz del árbol y sus procesos secundarios.



**d) RunLevels. ¿Qué son? ¿Cuál es su objetivo?**

Es un estado que le indica al sistema cómo funcionar. Los niveles se identifican por números. Se pueden usar para definir si el gestor de ventanas X está en marcha, si la red está operativa, etcétera. Cada uno es responsable de levantar (iniciar) o bajar (parar) una serie de servicios

**e) ¿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?**

Existen 7, y permiten iniciar un conjunto de procesos al arranque o apagado del sistema

Según el estándar:

* 0: halt (parada)
* 1: single user mode (monousuario)
* 2: multiuser, without NFS (modo multiusuario sin soperte de red)
* 3: full multiuser mode console (modo multiusuario completo por consola)
* 4: no se utiliza
* 5: X11 (modo multiusuario completo con login gr\_a\_co basado en X)
* 6: reboot

Se encuentran definidos en /etc/inittab (id:nivelesEjecuci\_on:acci\_on:proceso)

id: identica la entrada en inittab (1 a 4 caracteres)

* NivelesEjecucion: el/los niveles de ejecución en los que se realiza la accion

Accion: describe la accion a realizar

wait: inicia cuando entra al runlevel e init espera a que termine

initdefault

ctrlaltdel: se ejecutar\_a cuando init reciba la señal SIGINT

off, respawn, once, sysinit, boot, bootwait, powerwait, etc.

Proceso: el proceso exacto que será ejecutado

**f) 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?**

En SystemV, se configura a través del archivo /etc/inittab. Se almacena información sobre la definición de runlevels, configuración de acciones, procesos y programas. La estructura de su información es id:runlevels:action:process.

**g) 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 runlevel puedo usar sudo init 5 y en sistemas mas actuales que utilizan systemd, para cambiar de runlevel se utiliza sudo systemctl isolate runlevel5.target.

Este cambio no es permanente. Cambiar de runlevel con el comando init es un cambio temporal que afectará la ejecución actual del sistema. La próxima vez que reinicie el sistema, se utilizará el runlevel predeterminado definido en el archivo /etc/inittab.

**h) 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.**

Cuando init entra en un nivel de ejecución, llama al script rc con un argumento numérico que especifica el nivel de ejecución al que ir. rc luego inicia y detiene los servicios en el sistema según sea necesario para llevar el sistema a ese nivel de ejecución. Aunque normalmente se llama al inicio, init puede llamar al script rc para cambiar los niveles de ejecución.

Se almacenan Es compatible con SystemV ! /etc/init/rc-sysinit.conf, runlevels, scripts en /etc/init.d, objetivo start y stop.

El sistema determina qué scripts de inicio o detención ejecutar ante cada acción (inicio o parada) a través de la estructura de directorios y los enlaces simbólicos asociados a los niveles de ejecución (runlevels). El sistema tiene diferentes niveles de ejecución, que representan diferentes estados del sistema, como el nivel de ejecución predeterminado, el nivel de ejecución monousuario, el nivel de ejecución multiusuario, etc.

El sistema ejecuta estos scripts en orden, utilizando los enlaces simbólicos como guía. Esto asegura que los servicios y procesos se inicialicen o detengan de manera adecuada y en el orden correcto.

**i) ¿Qué es insserv? ¿Para qué se utiliza? ¿Qué ventajas provee respecto de un arranque**

**tradicional?**

Insserv es una herramienta que se encarga de gestionar los scripts de inicio de servicios y aplicaciones en los diferentes niveles de ejecución (runlevels) del sistema. Se utiliza para administrar el orden de los enlaces simbólicos del /etc/rcX.d, resolviendo las dependencias de forma automática.

* Ventajas del sistema de inicio tradicional (SysV init):
* Simplicidad: Es fácil de entender y gestionar los scripts de inicio y parada.
* Transparencia: Los scripts están en ubicaciones claras y son fáciles de editar.
* Control granular: Puedes personalizar los scripts según tus necesidades.
* Compatibilidad: Aún se utiliza en algunas distribuciones más antiguas y sistemas Unix.

**j) ¿Cómo maneja Upstart el proceso de arranque del sistema?**

Upstart manejaba el arranque del sistema utilizando eventos, trabajos (jobs), y dependencias. Cada evento desencadenaba la ejecución de trabajos, lo que permitía la concurrencia. Upstart también supervisaba y reiniciaba automáticamente los servicios en caso de fallos, mejorando la confiabilidad. Sin embargo, systemd se convirtió en el sistema de inicio predominante en muchas distribuciones Linux debido a sus capacidades avanzadas y eficiencia, desplazando a Upstart en la mayoría de las distribuciones modernas.

**k) Cite las principales diferencias entre SystemV y Upstart.**

* Permite la ejecución de trabajos en forma asincrónica a través de eventos (event-based) como principal diferencia con sysVinit que es estrictamente sincrónico (dependency-based).
* Las tareas de un job se definen mediante exec o script ... end script
* No más /etc/inittab
* Upstart ofrece capacidades avanzadas para supervisar y reiniciar automáticamente servicios en caso de fallos, lo que mejora la confiabilidad del sistema, en cambio systemV init requiere scripts de inicio y parada en los directorios /etc/init.d/ y utiliza enlaces simbólicos en directorios de ejecución para definir el orden de inicio de los servicios. La gestión de dependencias puede ser más compleja y manual.
* SystemV init carece de características avanzadas de supervisión y reinicio de servicios incorporadas, en cambio, upstart ofrece capacidades avanzadas para supervisar y reiniciar automáticamente servicios en caso de fallos, lo que mejora la confiabilidad del sistema.

**l) Qué reemplaza a los scripts *rc* de *SystemV* en *Upstart*? ¿En que ubicación del filesystem**

**se encuentran?**

Se remplazan por los Jobs, que se encuentran en /etc/init (.conf).

**m) Dado el siguiente *job* de upstart perteneciente al servicio de base de datos del mysql**

**indique a qué hace referencia cada línea del mismo:**

# MySQL S e r v i c e: es un comentario

d e s c r i p t i o n "MySQL Serve r “: descripción del trabajo

author " i n f o autor ": define el autor del trabajo

la siguiente linea especifica las condiciones que deben cumplirse para iniciar la ejecución.

s t a r t on ( net−device−up: Cuando al menos una interfaz de red esté arriba y disponible

and local −filesystems: Cuando los sistemas de archivos locales estén montados y disponibles.

and r u n l e v e l [ 2 3 4 5 ] ): Cuando el sistema esté en uno de los niveles de ejecución 2, 3, 4 o 5.

stop on r u n l e v e l [ 0 1 6 ]: Indica que el trabajo debe detenerse cuando el sistema cambie al nivel de ejecución 0, 1 o 6. El nivel 0 generalmente significa apagar el sistema, el nivel 1 es un modo de usuario único y el nivel 6 es para reiniciar el sistema.

[ . . . ]

exec / usr / sbin /mysqld: indica el comando que se ejecutará cuando se inicie el trabajo.

[ . . . ]

**n) ¿Qué es *sytemd*?**

Es un sistema que centraliza la administración de demonios y librerías del sistema.

**ñ) ¿A qué hace referencia el concepto de activación de socket en systemd?**

* No todos los servicios que se inician en el booteo se utilizan:

. impresora

. servidor en el puerto 80

* ¡Es un mecanismo de iniciación bajo demanda! Podemos ofrecer una variedad de servicios sin que realmente estén iniciados
* Cuando el sockect recibe una conexión spawnea el servicio y le pasa el socket
* ¡No hay necesidad de definir dependencias entre servicios! se inician todos los sockets en primera medida.

**o) ¿A qué hace referencia el concepto de *cgroup*?**

* Permite organizar un grupo de procesos en forma jerárquica.
* Agrupa conjunto de procesos relacionados (por ejemplo, un servidor web Apache con sus dependientes)
* Tareas que realiza:
* Tracking mediante subsistema cgroups ! no se utiliza el PID
* doble fork no funciona para escapar de systemd
* Limitar el uso de recursos

7

**ls -l > prueba**: Este comando lista el contenido del directorio actual en formato largo y redirige la salida a un archivo llamado "prueba". La razón por la que se puede ejecutar es que el usuario tiene permiso para listar el contenido del directorio actual y crear un archivo en ese directorio.

**ps > PRUEBA**: Este comando muestra una lista de los procesos en ejecución y redirige la salida a un archivo llamado "PRUEBA". La razón por la que se puede ejecutar es que el usuario tiene permiso para ver los procesos en ejecución.

**chmod 710 prueba**: Este comando cambia los permisos del archivo "prueba" para que solo el propietario pueda ejecutarlo y leerlo, mientras que los demás no tienen ningún permiso. La razón por la que se puede ejecutar es que el usuario es el propietario del archivo "prueba".

**chown root:root PRUEBA**: Este comando cambia el propietario y el grupo del archivo "PRUEBA" a "root". No se ejecutará con éxito, ya que el usuario no tiene permiso para cambiar el propietario y el grupo de un archivo a "root", a menos que sea el superusuario (root).

**chmod 777 PRUEBA**: Este comando cambia los permisos del archivo "PRUEBA" para que todos los usuarios puedan leer, escribir y ejecutar el archivo. No se ejecutará con éxito, ya que el usuario no tiene permiso para otorgar permisos de escritura y ejecución a otros usuarios en un archivo.

**chmod 700 /etc/passwd**: Este comando cambia los permisos del archivo /etc/passwd para que solo el propietario pueda leer y escribir en él, mientras que los demás no tienen ningún permiso. No se ejecutará con éxito, ya que el usuario no tiene permiso para modificar el archivo /etc/passwd.

**passwd root**: Este comando intenta cambiar la contraseña del usuario "root". No se ejecutará con éxito, ya que el usuario actual no tiene permisos para modificar la contraseña del usuario "root".

**rm PRUEBA:** Este comando elimina el archivo "PRUEBA". Se ejecutará con éxito si el usuario tiene permisos para eliminar archivos en el directorio actual.

**man /etc/shadow**: Este comando intenta ver el manual del archivo /etc/shadow. No se ejecutará con éxito, ya que el usuario no tiene permisos para acceder al archivo /etc/shadow.

**find / -name \*.conf:** Este comando busca archivos con extensión ".conf" en todo el sistema de archivos. Se ejecutará con éxito si el usuario tiene permisos para leer el sistema de archivos, pero puede llevar tiempo y generar una gran cantidad de resultados.

**usermod root -d /home/newroot -L**: Este comando intenta modificar las propiedades del usuario "root" cambiando su directorio de inicio a "/home/newroot" y bloqueando la cuenta. No se ejecutará con éxito, ya que el usuario actual no tiene permisos para modificar las propiedades del usuario "root".

**cd /root:** Este comando intenta cambiar al directorio "/root". No se ejecutará con éxito, ya que el usuario actual no tiene permiso para acceder al directorio "/root".

**rm \*:** Este comando intenta eliminar todos los archivos en el directorio actual. Se ejecutará con éxito si el usuario tiene permisos para eliminar archivos en el directorio actual.

**cd /etc:** Este comando cambia al directorio "/etc". Se ejecutará con éxito si el usuario tiene permisos para acceder al directorio "/etc".

**cp \* /home -R**: Este comando intenta copiar todos los archivos del directorio "/etc" al directorio "/home" de manera recursiva. No se ejecutará con éxito, ya que el usuario actual no tiene permiso para copiar archivos en el directorio "/etc".  
**shutdown**: Este comando intenta apagar el sistema. No se ejecutará con éxito, ya que el usuario actual no tiene permisos para apagar el sistema. Este comando generalmente requiere privilegios de superusuario (root).

8.

**A) Terminar el proceso con *PID* 23.**

kill 23

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

kill init

Su función principal es iniciar y gestionar todos los demás procesos del sistema. Si se termina el proceso init, es muy probable que el sistema se apague abruptamente y no de la manera adecuada, lo que podría resultar en la corrupción de datos y otros problemas.

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

find /home -type f -name "\*\.conf"

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

**procesos**

ps > /home/sofi/procesos

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

**Usuario: Lectura, escritura, ejecución**

**Grupo: Lectura, ejecución**

**Otros: ejecución**

touch /home/sofi/xxxx

chmod u=rwx,g=rw,o=x /home/sofi/xxxx

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

**Usuario: Lectura, escritura.**

**Grupo: Lectura, ejecución**

**Otros: Ninguno**

touch /home/sofi/yyyy

chmod u=rw,g=rw,o= /home/sofi/yyyy

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

rm -i /tmp/\*

**h) Cambiar el propietario del archivo /opt/isodata al usuario iso2021**

chown iso2021:iso2021 /opt/isodata

**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.**

pwd >> /home/sofi/donde

9.

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

su

**b) Cree un usuario. Elija como nombre, por convención, la primera letra de su nombre**

**seguida de su apellido. Asígnele una contraseña de acceso.**

sudo usseradd sofi2

sudo passwd sofi2

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

Archivos Modificados:

* /etc/passwd: El archivo /etc/passwd contiene información sobre los usuarios del sistema.
* /etc/shadow: El archivo /etc/shadow almacena contraseñas encriptadas para los usuarios. Cuando estableces una contraseña para el nuevo usuario, se registra en este archivo de forma segura.
* /etc/group: El archivo /etc/group contiene información sobre los grupos del sistema. Si el usuario se agrega a un grupo específico durante la creación, se reflejará en este archivo.

Directorios Creados:

* Directorio de inicio del usuario: Cuando creas un usuario, se crea un directorio de inicio en el sistema para ese usuario. El directorio de inicio generalmente se encuentra en el directorio /home y lleva el nombre del usuario (por ejemplo, /home/nombre\_de\_usuario).
* Directorio /home/nombre\_de\_usuario: En este directorio, se crean automáticamente subdirectorios como Desktop, Documents, Downloads, Music, Pictures, Public, etc., que proporcionan una estructura de organización predeterminada para los archivos del usuario.
* Directorio de configuración del usuario: En el directorio de inicio del usuario, se crean directorios y archivos de configuración ocultos (que comienzan con un punto, por ejemplo, .config, .bashrc, .profile) que almacenan configuraciones y preferencias específicas del usuario.
* Directorio /var/mail (opcional): Si se habilita el correo electrónico en el sistema, se puede crear un archivo de buzón de correo en /var/mail/nombre\_de\_usuario para que el usuario pueda recibir correos.

**d) Crear un directorio en /tmp llamado cursada2021**

cd /tmp

 mkdir Cursada2023

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

cd /tmp/Cursada2023 (si no estoy en la carpeta)

sudo cp -r /var/log/\* /tmp/Cursada2023/

**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.**

su

chown sofi2:users /tmp/Cursada2023/

**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 u=rwx,go=rx /tmp/Cursada2023/

**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.**

proceso asociado con ese PID e imprima su línea de comando $ ps -p 544

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

ps

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

who

**l) Vuelva a la terminal del usuario root, y envíele un mensaje al usuario anteriormente**

**creado, avisándole que el sistema va a ser apagado.**

su

shutdown -r now

**m) Apague el sistema.**

shutdown -P

10

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

mkdir 193667

**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 u=,g=x,o=rwx /home/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?.**

cd /etc

ls /etc

/etc > /etc/LEAME

Se puede crear este archivo a pesar de que ya existe un archivo llamado "LEAME" en el directorio /etc, ya que Linux distingue entre mayúsculas y minúsculas en los nombres de archivo. "leame" y "LEAME" son nombres de archivo diferentes debido a que una letra está en mayúscula y la otra en minúscula. Por lo tanto, no hay conflicto entre los dos archivos ya que tienen nombres distintos y se encuentran en directorios diferentes (/etc y tu directorio personal).

**(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.**

El comando find busca archivos y directorios en el sistema de archivos utilizando una estructura de árbol. Comienza en el directorio especificado como punto de partida y luego explora recursivamente todos los subdirectorios en busca de archivos que cumplan con los criterios especificados. Es una herramienta poderosa que permite buscar archivos y directorios en función de varios criterios, como nombre, tipo, tamaño, fecha de modificación, etc. Por ejemplo:

* Buscar un archivo por nombre: find /ruta/de/busqueda -name "nombre\_del\_archivo"
* Buscar un archivo llamado "documento.txt" en todo el sistema de archivos: find / name "documento.txt"
* Buscar archivos con características similares: find /ruta/de/busqueda -type f -name "\*.txt" -size +1M

**-type f:** Limita la búsqueda a archivos regulares (excluyendo directorios u otros tipos de archivos).

**-name "\*.txt":** Busca archivos con la extensión ".txt".

**-size +1M:** Busca archivos con un tamaño superior a 1 MB. Puedes ajustar el tamaño según tus necesidades, utilizando "c" para bytes, "k" para kilobytes, "M" para megabytes y "G" para gigabytes.

**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".**

su

find /home -name “\*.so” > /home/193667/ejerf

**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 -cf misLogs.tar -C /tmp logs

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

**"misLogs.tar.gz".**

tar -czf misLogs.tar.gz -C /tmp logs

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

cp /tmp/misLogs.tar ~/ # Para copiar "misLogs.tar" al directorio de inicio del usuario.

cp /tmp/misLogs.tar.gz ~/ # Para copiar "misLogs.tar.gz" al directorio de inicio del usuario.