

## EXAMEN 4

1. Listar el contenido del directorio /etc:
  - Ordenar el listado por fecha de modificación (mostrar primero los archivos más recientes)
  - Ordenar el listado por fecha de modificación (mostrar primero los archivos más antiguos)
  - Mostrar los tamaños de archivo en unidades amigables (KB, MB, GB)
  - Listar primero los directorios y luego los archivos
  - Ordenar por tamaño de archivo (de mayor a menor)
  - Ordenar por tamaño de archivo (de menor a mayor)
2. Determinar en una única línea de comandos cuántos elementos posee un directorio (sin contar los elementos "." y "..").
3. Determinar en una única línea de comandos cuántas palabras (separadas por espacio) contiene un archivo de texto.
4. Volcar por línea de comandos el contenido del archivo /etc/fstab exceptuando las líneas que contengan el caracter '#'.
5. Volcar por línea de comandos el contenido del archivo /etc/fstab exceptuando las 3 primeras líneas.
6. Buscar recursivamente dentro del directorio /etc todos los archivos cuyo nombre comienza con "net" (case insensitive).
7. Determinar cuanto espacio en disco ocupa el directorio /home.
8. Determinar el PID (Process ID) del proceso init.
9. Crear un script bash pepe.sh que se ejecute automáticamente cada vez que se inicia el sistema. En cada ejecución debe crear (y sobrescribir) el archivo /var/log/pepelastrun.txt, el cual debe contener la fecha y hora de la última ejecución del script.
10. Crear el archivo pepe1995.txt dentro del directorio /tmp cuya fecha de modificación sea el día 31 de diciembre de 1995 a la hora 23:59:59.
11. Buscar dentro del directorio \$HOME los archivos que hayan sido modificados el día de ayer.
12. Buscar dentro del directorio /var/log los archivos que hayan sido modificados entre ayer y anteayer.

13. Determinar en una línea de comandos los tipos de filesystems montados utilizando una única columna, por ejemplo debe retornar:

```
ext4
proc
sysfs
devpts
tmpfs
ext4
ext4
ext4
binfmt_misc
rpc_pipefs
```

14. Mantenimiento de usuarios:

- Crear el usuario "pepe" cuyo directorio \$HOME sea /home/pepe
- Asignar un nuevo password al usuario "pepe"
- Agregar al usuario "pepe" al grupo "wheel"
- Cambiar el nombre del usuario "pepe" a "pedro"
- Determinar a qué grupos pertenece el usuario "pedro"
- Permitir el uso del comando "sudo" al usuario "pedro"
- Denegar el uso del comando "sudo" al usuario "pedro"
- Eliminar al usuario "pedro" junto con su directorio \$HOME

15. Listar los últimos comandos ejecutados en la sesión actual.

16. Determinar en una línea de comandos cuánto espacio libre queda en las diferentes particiones del sistema.

17. Manejo de archivos:

- Crear el directorio /tmp/pepe2/
- Copiar todos los archivos de nuestro directorio \$HOME al directorio /tmp/pepe2/ preservando todas las propiedades de los mismos (owners, permisos, fechas, etc.)
- Mover el directorio /tmp/pepe2/ a /tmp/pepe3/
- Dar permiso de ejecución a todos los archivos dentro del directorio /tmp/pepe3/
- Quitar permiso de escritura a todos los archivos dentro del directorio /tmp/pepe3/
- Eliminar el directorio /tmp/pepe3/ junto con todo su contenido

18. Manejo de procesos:

- Determinar el userid efectivo de la sesión actual
- Ejecutar en background (segundo plano) el comando ping 8.8.8.8 > /dev/null
- Ejecutar el comando top, una vez iniciado enviarlo a segundo plano.

- Mostrar todos los procesos que se están ejecutando con el mismo userid efectivo que la sesión actual
- Mostrar un árbol sólo de los procesos que se están ejecutando con el mismo userid efectivo que la sesión actual
- Mostrar todos los procesos ejecutando como usuario "root" (ID efectivo y real) en formato de usuario
- Determinar el PID (Process ID) del proceso ejecutando el comando `ping 8.8.8.8 > /dev/null`
- Bajar al mínimo la prioridad de ejecución del proceso ejecutando el comando `ping 8.8.8.8 > /dev/null`
- Matar el proceso ejecutando el comando `ping 8.8.8.8 > /dev/null`
- Determinar qué proceso está consumiendo más CPU
- Determinar qué proceso está consumiendo más memoria RAM
- Determinar cuanta memoria RAM disponible (libre) posee el sistema
- Traer a primer plano el proceso ejecutando el comando `top`

19. Crear un directorio llamado "A" que contenga los archivos "doc1.txt", "doc3.txt" y "doc77.txt". Crear un directorio llamado "B" que contenga los archivos "doc1.txt", "doc4.txt" y "doc89.txt".

- Listar, utilizando una única línea de comandos, los nombres de archivo que se encuentran en el directorio "A" pero no en el directorio "B"
- Listar, utilizando una única línea de comandos, los nombres de archivo que se encuentran tanto en el directorio "A" como en el directorio "B"

20. Comprimir los directorios "A" y "B" utilizando el algoritmo "gzip" en un único archivo llamado "ab.tar.gz".