

Prácticas

1. Los discos y particiones

Objetivo: crear una partición y que el sistema la reconozca. ¡Cuidado: la operación puede ser destructiva! Estas operaciones pueden efectuarse con el formato GPT, en caso de emplear el comando `gdisk` adapte las acciones en consecuencia.

1. Imaginemos un disco duro de su sistema que dispone de espacio para la creación de una partición. En la máquina de prueba, se trata del primer disco SATA, llamado /dev/sda. Hay tres particiones: /dev/sda1, /dev/sda2 y /dev/sda3. Cree una tercera partición extendida.

Inicie fdisk:

```
# fdisk /dev/sda
```

2. Cree una nueva partición pulsando n y luego [Entrar].

Cree una partición extendida pulsando e y luego [Entrar].

Como número, teclee el 4 y luego [Entrar]: sólo puede haber cuatro particiones primarias.

Finalmente, pulse dos veces [Entrar] para aceptar los valores por defecto.

3. Cree luego una partición lógica, según el mismo principio, en la partición extendida. Llevará el número 5 y ocupará todo el espacio disponible.
4. Guarde la tabla de las particiones pulsando w y luego [Entrar]. Seguramente obtendrá un aviso que le informará de que la nueva tabla no está accesible.
5. Fuerce la actualización de la nueva tabla de las particiones con partprobe:

```
# partprobe /dev/sda
```

6. Verifique en /dev la presencia del archivo /dev/sda5.

```
# ls -l /dev/sda5
```

2. Creación de un sistema de archivos

Objetivo: crear y manejar el sistema de archivos en /dev/sda5. Cuidado: esta operación es destructiva.

1. Cree un sistema de archivos de tipo ext2 en /dev/sda5, sin opciones particulares:

```
# mkfs -t ext2 /dev/sda5
```

2. A decir verdad, había que crearlo en ext4 para aprovechar su transaccionalidad. Modifique el sistema de archivos para que el tipo sea ext4. Primero agregue un diario, que convertirá el sistema de archivos a ext3, luego agregaremos las diversas opciones para cambiar a ext4, y finalmente verificaremos el sistema de archivos:

```
# tune2fs -j /dev/sda5
```

```
# tune2fs -O extents,uninit_bg,dir_index /dev/sda5  
# fsck -pDf /dev/sda5
```

3. La nueva partición va a servir para almacenar varios datos. Asígnele una etiqueta (nombre, label): DATOS.

```
# e2label /dev/sda5 DATOS
```

4. Ahora intente conocer el identificador único de sistema de archivos, el UUID, de su nuevo sistema de archivos. Dispone de dos métodos:

Por udev:

```
# /lib/udev/vol_id --uuid /dev/sda5  
527585d3-1e52-4aba-b7fc-70f18388458d
```

Por dumpe2fs:

```
# dumpe2fs /dev/sda5 | grep -i uuid  
Filesystem UUID:      527585d3-1e52-4aba-b7fc-70f18388458d
```

Por blkid:

```
# blkid -o value -s UUID /dev/sda5  
527585d3-1e52-4aba-b7fc-70f18388458d
```

3. Acceso y montaje del sistema de archivos

Objetivo: acceder al nuevo sistema de archivos creado en la práctica anterior.

1. Cree un directorio que servirá de punto de montaje para nuevo sistema de archivos:

```
# mkdir /DATOS
```

2. Monte el sistema de archivos, por su nombre de periférico, en este directorio:

```
# mount -t ext4 /dev/sda5 /DATOS
```

3. Muévase a este punto de montaje y cree un archivo cualquiera:

```
# cd /DATOS  
# touch pepito
```

4. Salga de este directorio (CD) y desmonte el sistema de archivos:

```
# umount /DATOS
```

5. Añada una línea en /etc/fstab para poder montar este sistema de archivos de manera automática, por su label.

La línea es:

```
LABEL=DATOS /DATOS ext4 defaults 0 0
```

6. Monte el sistema de archivos de manera sencilla desde el nombre de su punto de montaje:

```
# mount /DATOS
```

4. Estadísticas y mantenimiento del sistema de archivos

Objetivo: obtener información sobre la ocupación del sistema de archivos y repararlo si fuese necesario.

1. Mire el estado de ocupación de sus sistemas de archivos de manera legible para un simple "humano":

```
# df -H
```

2. El sistema de archivos que apunta a /home parece ocupado. Se trata de determinar qué puede ocupar tanto espacio. Determine la ocupación de cada archivo y directorio:

```
# du -m /home
```

3. El resultado es demasiado largo. Ordene la salida de manera que aparezcan las mayores ocupaciones en último lugar:

```
# du -m | sort -n
```

4. Un directorio montado en /DATOS (práctica anterior) tiene problemas: parece que el contenido de un directorio se ha corrompido; nombres de archivos y tamaños estrafalarios. Verifique y arregle este sistema de archivos:

```
# cd  
# umount /DATOS  
# fsck /dev/sda5
```

5. Fuerce una verificación de este sistema de archivos en el próximo reinicio. El manual de tune2fs indica que se trata de la opción `-C`:

```
# tune2fs -C 1000 /dev/sda5
```

5. Swap y memoria

Objetivo: gestionar la swap y la memoria.

1. En una máquina dada, el balance de memoria se presenta así, ¿qué deduce de ello?

```
# free
      total    used   free  shared  buffers  cached
Mem:  2060680  2011224   49456     0    170628   958508
-/+ buffers/cache:  882088  1178592
Swap:  2104472    1296  2103176
```

La máquina dispone de 2 GB de memoria física y de 2 GB de swap. Aunque indica unos 48 MB de memoria libre, hay unos 950 MB de memoria caché y 160 MB de memoria intermedia. O sea, 1 GB de memoria potencial para liberar.

2. Verifique el nombre de la partición que contiene el espacio o los espacios de swap. Un `man de swapon` indica que se puede encontrar la información en `/proc/swaps`:

```
# cat /proc/swaps
Filename      Type      Size Used Priority
/dev/sda5     partition 2104472 1336 -1
```

3. A este nivel de carga, posiblemente la swap de esta máquina sea inútil. Desactívela.

```
# swapoff /dev/sda5
```

4. Unos instantes más tarde, debe cargar una aplicación pesada de tratamiento de imagen que va a consumir mucha memoria. Vuelva a cargar la totalidad de las zonas de swap:

```
# swapon -a
```

5. Consulte la información detallada en la memoria:

```
# cat /proc/meminfo
```

6. Los permisos

Objetivo: modificar los permisos y comprender el mecanismo de los permisos y límites de los usuarios.

1. Modifique los propietarios y grupos de /DATOS en seb:users.

```
# chown seb:users /DATOS
```

2. Todo el mundo debe poder escribir en esta carpeta, pero sin suprimir los archivos de los demás. Asimismo, todos los archivos creados en este directorio deben pertenecer al grupo users. Coloque los permisos correctos: hacen falta todos los permisos para todo el mundo, el permiso sticky y el SGID-bit:

```
# chmod 3777 /DATOS
```

3. Cree un directorio de prueba en /tmp con los permisos rwxrwxrwx. Cree un archivo en él y modifique sus permisos retirando el permiso w al grupo y los demás. ¿Quién puede suprimirlo?

```
# mkdir /tmp/test  
# chmod 777 /tmp/test  
# touch /tmp/test/pepito  
# chmod g-w,o-w /tmp/test/pepito
```

Todo el mundo puede suprimir este archivo: sus permisos no tienen importancia. Son los permisos del directorio, en particular el permiso de escritura en el directorio, los que determinan quién puede suprimir los archivos que hay dentro.

4. Cree una máscara restrictiva: usted puede hacer lo que quiera, el grupo sólo tiene acceso a los directorios y puede leer los archivos, pero los demás no pueden hacer nada.

La máscara debe dejar pasar todos los permisos del usuario: 0.

La máscara debe dejar pasar los permisos r y x para los grupos. Sólo w está enmascarado: 2.

La máscara suprime todos los permisos a los demás: 7.

Resultado:

```
# umask 027
```

5. Retire el permiso SUID a /usr/bin/passwd y modifique su contraseña. Intente modificar su contraseña. No funciona: passwd debe ser root para modificar los archivos. Vuelva a poner el permiso s.

```
# chmod u-s /usr/bin/passwd
# passwd =>Error
# chmod u+s /usr/bin/passwd
```