

Prácticas de Sistemas Operativos

Módulo I. Administración de Linux

Sesión 2. Herramientas de administración del sistema de archivos

1. Introducción

Como vimos en el apartado “Organización del Sistema de Archivos (SA) y gestión básica de archivos” (*Sesión 1, apartado 5*), el estándar FHS nos permite conocer que tipo de información se encuentra almacenada en los directorios base de cualquier instalación de Linux que siga este estándar. Además, vimos que el concepto de *tipo de archivo* en un sistema UNIX no tenía tanto que ver con el mundo de las aplicaciones sino con determinados elementos más cercanos al sistema operativo:

- **Archivos regulares.** Archivos de programa y datos.
- **Directorios.** Archivos específicamente diseñados para soportar la estructura jerárquica de organización de la información en un SA. Esta estructura implementa lo que en el mundo UNIX/Linux se conoce como *espacio de nombres de archivo*.
- **Enlaces simbólicos.** Archivos que permiten referenciar a otros archivos desde distintas ubicaciones en el espacio de nombres (distintos directorios).
- **Archivos especiales de dispositivo.** Estos archivos representan dispositivos y permiten al usuario del lenguaje de órdenes del shell y a los programadores de aplicaciones trabajar sobre los dispositivos como si se tratase de un archivo normal. De hecho, estos tipos de archivo implementan la abstracción de dispositivos que proporcionan los SAs en los sistemas UNIX/Linux. Los sistemas UNIX distinguen dos subtipos de archivos especiales de dispositivo: orientados a dispositivos de **bloques** y orientados a dispositivos de **caracteres**.
- Archivos para comunicaciones **FIFO**. Permiten comunicar procesos.
- Archivos para comunicaciones **Socket**. Permiten comunicar procesos.

Al final de la sesión anterior (*Sesión 1, apartado 5.2.1*) vimos los principales archivos que nos permiten conocer la información de como se han montado los SAs accesibles, de forma transparente al usuario, desde nuestro *espacio de nombres*. Estos archivos estaban ubicados en el directorio `/etc`: `/etc/fstab` y `/etc/mtab`, y en el directorio `/proc`: `/proc/mounts` y `/proc/filesystems`.

Esta sesión aborda dos áreas temáticas de la administración de SOs intimamente relacionadas. Por una parte se presentarán los conceptos y órdenes para administración (herramientas del administrador) relativas al trabajo con dispositivos de almacenamiento secundario y creación, personalización y comprobación de inconsistencias de los SAs en sí mismos. Por otra parte, se mostrarán los conceptos, procedimientos y herramientas que se pueden utilizar para administrar el software de nuestro ordenador. La **administración del software** incluye la instalación, eliminación, actualización o reconstrucción de paquetes, lo cual resulta de vital importancia para tener un sistema estable y usuarios satisfechos.

Con respecto a la primera área temática se comentarán los conceptos y procedimientos para: Establecer **particiones** en dispositivos de almacenamiento secundario; asignar un determinado SA a una partición en un dispositivo de este tipo, es decir, lo que se conoce en SOs como **formateo lógico**; establecer **configuraciones personalizadas** de la información que el sistema de archivos mantiene sobre si mismo (metainformación o metadatos o atributos del sistema de archivos); hacer visible la información que almacena un SA en nuestro espacio de nombres de archivo actual, lo que se conoce comúnmente por **montar un SA**; y, finalmente, poder comprobar la **consistencia** de las estructuras de datos que contiene el SA.

2. Objetivos principales

- Comprender qué es una partición de un dispositivo de almacenamiento secundario, conocer la información que se almacena en la tabla de particiones, y saber utilizar una herramienta que permita llevar a cabo la partición de un dispositivo (**fdisk**).
- Comprender el concepto de formateo lógico de particiones y comprender la familia de órdenes **mkfs***, su utilización y sus opciones principales.
- Conocer los metadatos básicos de un SA tipo Linux y saber utilizar las herramientas que proporciona Linux para personalizar dicha información, **tune2fs**, y realizar comprobaciones y reparaciones de las posibles inconsistencias que se puedan producir en dicha información como consecuencia del uso del sistema de archivos, **fsck**.
- Comprender el concepto de espacio de nombres de la estructura jerárquica de directorios y saber ampliar el espacio de nombres con nuevos sistemas de archivos mediante la orden **mount**.
- Conocer los principales aspectos e implicaciones relacionados con llevar a cabo una administración correcta y productiva del software instalado en un ordenador con sistema operativo Linux.
- Ser capaz de tomar decisiones acerca de qué software instalar y cómo hacerlo.
- Adquirir destrezas básicas en el manejo de las principales herramientas disponibles para administrar el software de un sistema Linux.

3. Partición de dispositivos de almacenamiento secundario

Para poder utilizar un dispositivo de almacenamiento secundario (*drive*), como puede ser un disco duro o una memoria flash, en un SO es necesario, en primer lugar, establecer secciones (**particiones**) dentro del dispositivo físico, que sean identificables, y que permitan alojar un SA concreto. Al proceso de establecer e identificar estas particiones en el dispositivo se le denomina comúnmente **partición de disco**, porque los primeros dispositivos de los que dispusimos los informáticos para almacenar información de forma persistente fueron discos (bueno..., fueron *tambores magnéticos* pero eso es otra historia).

En la actualidad podemos encontrar multitud de dispositivos de almacenamiento secundario pero en general todos caben en uno de los tres siguientes tipos:

- Discos duros (SATA, IDE, SCSI,...)
- Memorias Flash (USB *memory sticks*, *Solid State Drives* (SSD's),...)
- Dispositivos RAID (RAID hardware y RAID software en Linux)

Una partición en cualquiera de estos dispositivos está constituida por un conjunto de sectores que van a formar lo que podemos denominar un *disco lógico*. Un *sector* es la mínima unidad de almacenamiento de información en un dispositivo de almacenamiento secundario. El tamaño de un sector varía dependiendo del dispositivo pero los tamaños comunes son: 512, 1024, 2048 y 4096 bytes, aunque puede haber dispositivos con un mayor tamaño de sector.

Cuando creamos una partición es necesario asociarle una etiqueta que indica el tipo de SA que va a alojar cuando posteriormente se *formatee*. Esta información se almacena mediante un código numérico que determina el tipo de partición. Cada SO tiene sus propios códigos numéricos para las particiones pero nosotros solamente nos vamos a centrar en los que se utilizan en Linux. Por ejemplo, el código asociado a una partición que va a alojar un SA de tipo **ext2** es el **0x83**, y el de una partición de intercambio (**swap**) es el **0x82**. Para ver una lista de los tipos de particiones soportados y sus códigos asociados puedes usar la siguiente orden: **/sbin/sfdisk -T**.

El número de particiones que podemos establecer en un dispositivo bajo una arquitectura Intel está limitado debido a la “*compatibilidad hacia atrás*” (algo habitual en informática). La primera **tabla de particiones** (básicamente, es una tabla en la que cada entrada mantiene la información asociada a una partición: donde comienza y finaliza la partición en el disco, tipo de partición, partición de arranque (sí/no), y algo más) se almacenaba como parte del **sector de arranque maestro (master boot record, MBR)** y solamente tenía espacio para almacenar cuatro entradas. Estas cuatro particiones se conocen en la actualidad como **particiones primarias**.

Una partición primaria de disco puede a su vez dividirse. Estas subdivisiones se conocen con el nombre de **particiones lógicas**. De esta forma podemos saltarnos la restricción histórica de poder establecer solamente cuatro particiones por dispositivo de almacenamiento secundario.

La partición primaria que se usa para alojar las particiones lógicas se conoce como partición extendida y tiene su propio tipo de partición: **0x05** (puedes comprobarlo con la orden que ya conoces). A pesar de que *a priori* podríamos pensar que, de esta forma, el número de particiones que podemos establecer en un dispositivo es “ilimitado”, esto no es así. Como casi siempre, los SO imponen límites en el número de recursos disponibles y, por supuesto, el número de particiones no iba a ser una excepción. Por ejemplo, Linux limita el número máximo de particiones que se pueden realizar sobre un disco SCSI a 15, y a un total de 63 sobre un disco IDE.

3.1 ¿Cuántas particiones hago en mi dispositivo?

Para responder a esta pregunta vamos a establecer una clasificación de los dispositivos de almacenamiento secundario, pues según si el dispositivo actuará como **dispositivo de arranque** (*Boot drive*) o no, será necesario establecer una distribución de particiones u otra. Un dispositivo de arranque va a ser el dispositivo que utilice en primer lugar la BIOS de nuestra arquitectura para cargar en memoria el programa SO (¡o un programa cargador de SOs!).

Si queremos que nuestro SO arranque desde el dispositivo sobre el que vamos a realizar las particiones necesitamos establecer la siguiente configuración de particiones:

- Una partición primaria
- Una o más particiones **swap**.
- Ninguna, o las que quieras, partición(es) primaria(s) y lógica(s). Por supuesto que el número de particiones está dentro de los límites que establece la compatibilidad hacia atrás y el SO.

La configuración de particiones de cualquier dispositivo de almacenamiento secundario, que no se vaya a utilizar como dispositivo de arranque sino simplemente para almacenar información, será la siguiente:

- Una o más particiones primarias o lógicas. Por supuesto siempre dentro de los límites comentados anteriormente.
- Ninguna, o las que quieras, partición(es) **swap**.

3.2 ¿Qué directorios de primer nivel del estándar FHS deberían estar soportados por una partición independiente?

Obviamente, toda la estructura de directorios del FHS puede estar soportada por una única partición (la partición de arranque etiquetada como "/"). No obstante, es conveniente en determinadas situaciones establecer particiones independientes que soporten la información que se almacene en determinados directorios de la estructura. A continuación se muestran algunas situaciones de este tipo para algunos de los directorios de primer nivel del FHS.

Tabla 1. Directorios de primer nivel del estándar FHS que pueden ser susceptibles de ser soportados por una partición independiente.

/home	Aquí es donde se almacenan los directorios de inicio de todos los usuarios con cuenta en el sistema. Por tanto, podemos establecer una partición independiente y limitar de esta forma el espacio de almacenamiento permitido para los usuarios (más adelante veremos una forma más flexible). Además, de cara a la instalación de versiones nuevas de Linux, si el /home está soportado en una partición independiente no tendremos problemas con nuestra partición raíz y podremos formatearla e instalar en ella la nueva versión del SO.
/usr	Este directorio almacena la mayoría de los ejecutables binarios del sistema, así como sus demás archivos adicionales, incluyendo los de documentación. Además si instalamos el paquete con el árbol de las fuentes del kernel también va a parar aquí.
/var	Este directorio contiene los directorios de SPOOL como por ejemplo los de impresión y correo electrónico. Normalmente todos los ordenadores en los que Linux actúa como servidor asignan a /var una partición independiente de la partición raíz /.

Actividad 3.1. Partición de un dispositivo: "USB pen drive" o "memory stick".

En esta actividad utilizaremos un dispositivo USB de tipo "pen drive" para establecer particiones. Vamos a crear una tabla de particiones en las que se van a definir dos particiones primarias que se configuraran para albergar dos sistemas de archivos tipo linux, de manera que la primera albergará un SA **ext3** y la segunda un **ext4**. Ambas serán particiones tipo Linux **0x83**. El tamaño de las particiones queda a vuestra libre elección pero por lo menos deberían tener 512 MB.

Como las particiones que vamos a hacer no van a ser excesivamente grandes vamos a utilizar la herramienta **fdisk**¹. Esta orden se puede manejar básicamente como un programa guiado por menús para la creación y manipulación de la tabla de particiones. Para poder actuar sobre el *pen drive* necesitamos saber que existe un archivo especial de dispositivo que actúa como interfaz para

¹ Para tamaños de partición mayores es mejor usar la orden **parted** de GNU.

el dispositivo físico (<dispositivo>), por ejemplo, `/dev/sda2` representa el dispositivo de almacenamiento secundario sobre el que queremos trabajar. La <particion> es un <dispositivo> seguido de un número de partición (comenzando por el 1), p.e. `/dev/sda2`.

A continuación vamos a describir cómo llevaremos a cabo en las aulas el procedimiento para poder realizar la partición de un dispositivo de almacenamiento secundario. Vamos a distinguir dos procedimientos: partición de un dispositivo simulado mediante un archivo especial de dispositivo y partición de un *pen drive*.

A) Preparación previa a la partición de un dispositivo simulado mediante un archivo especial de dispositivo.

Vamos a utilizar un dispositivo simulado mediante un archivo `/dev/loop?`. Estos archivos permiten crear un dispositivo de almacenamiento virtual cuyo espacio de almacenamiento viene soportado por un archivo asociado. A continuación se describen los pasos para construir este dispositivo simulado:

(a) Crea los archivos `/dev/loop0` y `/dev/loop1`, si no se encuentran en el sistema, utilizando las siguientes órdenes:

```
#> mknod /dev/loop0 b 7 0
```

```
#> mknod /dev/loop1 b 7 1
```

(b) Crea un archivo de 20 MB y otro de 30 MB en tu sistema de archivos con las siguientes órdenes:

```
#> dd if=/dev/zero of=/root/archivo_SA20 bs=2k count=10000
```

```
#> dd if=/dev/zero of=/root/archivo_SA30 bs=3k count=10000
```

(c) Ahora vamos a asociar un archivo de dispositivo loop a cada uno de los archivos que acabas de crear. De esta forma el “disco virtual” que representa el archivo pasará a estar asociado al archivo de dispositivo `/dev/loop0` y `/dev/loop1`. Para ello ejecuta las siguientes órdenes:

```
#> losetup /dev/loop0 /root/archivo_SA20
```

```
#> losetup /dev/loop1 /root/archivo_SA30
```

(d) Puedes comprobar la configuración de tus “discos virtuales” mediante la siguiente orden que producirá como salida el siguiente resultado:

```
#> fdisk -l /dev/loop0 /dev/loop1
```

```
#> fdisk -l /dev/loop0 /dev/loop1
Disk /dev/loop0: 20 MB, 20480000 bytes
255 heads, 63 sectors/track, 2 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/loop0 doesn't contain a valid partition table

Disk /dev/loop1: 30 MB, 30720000 bytes
255 heads, 63 sectors/track, 3 cylinders
```

² El nombre de los archivos especiales de dispositivo depende de la distribución concreta.

```
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/loop1 doesn't contain a valid partition table
```

B) Preparación previa a la partición de un *pen drive*

En primer lugar, introduce el *pen drive* en el conector USB y, a continuación podrás ver el archivo especial de dispositivo que se le ha asignado y el directorio que actúa como punto de montaje usando el archivo `/proc/mounts` (también la orden `mount` sin argumentos). Para poder realizar la partición del *pen drive* debemos desmontar el dispositivo: `umount`.

Independientemente de la opción escogida, A) o B), ahora podemos proceder a crear la tabla de particiones mediante `fdisk`, ya sea en el *pen drive* o en los “discos virtuales” `/dev/loop0` Y `/dev/loop1`. Una vez que hayamos finalizado el proceso podemos comprobar si la tabla de particiones es correcta con la orden que ya conoces. El listado de salida debería parecerse a la salida por pantalla que se muestra a continuación.

```
Disk /dev/loop0: 20 MB, 20480000 bytes
125 heads, 58 sectors/track, 5 cylinders
Units = cylinders of 7250 * 512 = 3712000 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x7ba8475a

    Device Boot      Start         End      Blocks   Id  System
/dev/loop0p1          1           6       18976    83  Linux
Partition 1 has different physical/logical beginnings (non-Linux?):
phys=(0, 32, 33) logical=(0, 35, 19)
Partition 1 has different physical/logical endings:
phys=(2, 124, 58) logical=(5, 64, 38)

Disk /dev/loop1: 30 MB, 30720000 bytes
188 heads, 24 sectors/track, 13 cylinders
Units = cylinders of 4512 * 512 = 2310144 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x8eda4aa2

    Device Boot      Start         End      Blocks   Id  System
/dev/loop1p1          1          14       28976    83  Linux
Partition 1 has different physical/logical beginnings (non-Linux?):
phys=(0, 32, 33) logical=(0, 85, 9)
Partition 1 has different physical/logical endings:
phys=(3, 187, 24) logical=(13, 55, 24)
```

A partir de este momento ya disponemos de particiones sobre las que crear sistemas de archivos. Para ahondar más en el proceso de partición de discos sobre Linux podeis consultar el “*Linux Partition HOWTO*” (<http://www.tldp.org/HOWTO/Partition/>).

4. Asignación de un Sistema de Archivos a una partición (formateo lógico)

Una vez que disponemos de las particiones en nuestro dispositivo de almacenamiento secundario debemos proceder a asignar el sistema de archivos adecuado a cada una de las particiones. En Linux, aparte del SA específico para las particiones especialmente dedicadas a intercambio (**swap**), se utilizan normalmente tres sistemas de archivos: **ext2**, **ext3** y **ext4**.

- **ext2**. Es el sistema de archivos de disco de alto rendimiento usado en Linux para discos duros, memorias *flash* y medios extraíbles. El *second extended file system* se diseñó como una extensión del *extended file system* (**ext**). **ext2** ofrece el mejor rendimiento en términos de velocidad de transferencia de E/S y uso de CPU de entre todos los sistemas de archivos que soporta Linux.
- **ext3**. Es una versión de **ext2** que incluye “registro por diario” (*journaling*). El **journaling** es un mecanismo por el cual un sistema informático puede implementar *transacciones*. Se basa en llevar un *journal* o registro de diario en el que se almacena la información necesaria para restablecer los datos afectados por la transacción en caso de que ésta falle. La aplicación del *journaling* en los sistemas de archivos modernos permite evitar la corrupción de las estructuras de datos que soportan la información del sistema de archivos: estructura de directorios, estructura de bloques libres de disco y estructuras que soportan los atributos de los archivos.
- **ext4**. Es el estándar *de facto* actual de las distribuciones Linux. Este SA tiene unas estructuras similares a las del **ext3** pero, además, presenta las siguientes mejoras:
 - **Extensiones**. Las extensiones permiten describir un conjunto de bloques de disco contiguos, mejorando de esta forma el rendimiento de E/S al trabajar con archivos de gran tamaño y reduciendo la fragmentación de disco.
 - **Asignación retardada de espacio en disco (allocate-on-flush)**. Esta técnica permite postergar en el tiempo la asignación de bloques de disco hasta el momento real en el que se va a realizar la escritura.

Actividad 4.1. Creación de sistemas de archivos.

El objetivo es simplemente formatear lógicamente las particiones creadas con anterioridad de forma consistente con el tipo de SA que se estableció que iba a ser alojado. En la primera partición crearemos un SA de tipo **ext3** y en la segunda un **ext4**.

La orden que permite establecer un SA de los reconocidos dentro del sistema Linux sobre una partición de disco es **mke2fs** (consulta el manual en línea para familiarizarte con sus opciones). El resultado de la ejecución de esta orden es el formateo lógico de la partición escogida utilizando el SA que se ha seleccionado.

Utiliza el manual en línea para conocer cómo ejecutar la orden de creación de SA. **mke2fs** es la orden genérica para creación de sistemas de archivos. Como requisito es necesario que establezcas dos etiquetas de volumen para los SAs: **LABEL_ext3** para la primera partición y **LABEL_ext4** para la segunda. Debería aparecer un listado en pantalla similar al siguiente.

```
mke2fs 1.41.11 (14-Mar-2010)
Filesystem label=LABEL_ext3
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
5016 inodes, 20000 blocks
1000 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=20709376
3 block groups
8192 blocks per group, 8192 fragments per group
1672 inodes per group
Superblock backups stored on blocks:
    8193

Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 25 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

5. Ajuste de parámetros configurables de un SA y comprobación de errores

Una vez que tenemos disponibles nuestros nuevos sistemas de archivos podemos utilizar una orden, **tune2fs**, que nos permite ajustar determinados parámetros de los sistemas de archivos **ext2/ext3/ext4**. Puedes utilizar la opción **-l** para obtener un listado por pantalla que nos muestre la información relevante de un determinado SA. A continuación se muestra un ejemplo de listado sobre uno de nuestros “discos virtuales” con su sistema de archivos ya creado.

```
tune2fs 1.41.11 (14-Mar-2010)
Filesystem volume name:   LABEL_ext3
Last mounted on:          <not available>
Filesystem UUID:          d143596b-929f-4339-b393-02d2245943c2
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype
                           sparse_super
Filesystem flags:         signed_directory_hash
Default mount options:    (none)
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              5016
Block count:              20000
Reserved block count:     1000
```



```

Free blocks:          18163
Free inodes:          5005
First block:          1
Block size:           1024
Fragment size:        1024
Reserved GDT blocks:  78
Blocks per group:     8192
Fragments per group:  8192
Inodes per group:     1672
Inode blocks per group: 209
Filesystem created:   Mon Oct 17 00:34:21 2011
Last mount time:       n/a
Last write time:      Mon Oct 17 00:34:21 2011
Mount count:          0
Maximum mount count:  25
Last checked:         Mon Oct 17 00:34:21 2011
Check interval:       15552000 (6 months)
Next check after:     Sat Apr 14 00:34:21 2012
Reserved blocks uid:  0 (user root)
Reserved blocks gid:  0 (group root)
First inode:          11
Inode size:           128
Journal inode:         8
Default directory hash: half_md4
Directory Hash Seed:  2c95350b-8ed7-4706-ae32-e902e16def4c
Journal backup:        inode blocks

```

La tabla siguiente muestra algunas opciones interesantes de **tune2fs**.

Tabla 2. Algunas opciones de la orden **tune2fs**.

-l <dispositivo>	Muestra por pantalla el contenido del <i>superbloque</i> del SA.
-c max-mount-counts <dispositivo>	Establece el número máximo de montajes que se puede llegar a realizar sin que se realice una comprobación de la consistencia del SA.
-L label <dispositivo>	Poner una etiqueta al SA.

Con el tiempo, las estructuras de metainformación del SA pueden llegar a corromperse. Esta situación podría provocar degradación en el rendimiento del sistema e incluso situaciones de pérdida de información. Para solucionarlo Linux automatiza el proceso de comprobación de la consistencia del sistema de archivos en base al número de montajes que se han realizado. No obstante, pueden ocurrir situaciones en las que sea necesario que el administrador ejecute manualmente las comprobaciones y repare las posibles inconsistencias. Linux proporciona la herramienta **fsck** para llevar a cabo esta labor.

Hay que tener muy claro que esta herramienta actúa sobre las estructuras de metainformación del SA, no sobre el contenido de los archivos (datos de archivo). A continuación se muestran algunas de las situaciones de inconsistencia de metadatos del SA que se pueden dar:

- Bloques que están asignados simultáneamente a varios archivos.

- Bloques marcados como libres pero que están asignados a un archivo determinado.
- Bloques marcados como asignados a un archivo determinado pero que realmente están libres.
- Inconsistencia del número de enlaces a un determinado archivo.
- i-nodos marcados como ocupados pero que realmente no están asociados a ningún archivo.

Actividad 5.1. Personalización de los metadatos del SA.

Consultando el manual en línea para la orden **tune2fs** responde a las siguientes preguntas:

- (a) ¿Cómo podrías conseguir que en el siguiente arranque del sistema se ejecutara automáticamente **e2fsck** sin que se haya alcanzado el máximo número de montajes?
- (b) ¿Cómo podrías conseguir reservar para uso exclusivo de un usuario username un número de bloques del sistema de archivos?

6. Montaje y desmontaje de Sistemas de Archivos

Una vez que disponemos de nuestros nuevos sistemas de archivos ya solo nos queda ponerlos a disposición de los usuarios haciendo que puedan ser accesibles dentro de la jerarquía de directorios. Ya se ha comentado con anterioridad que la jerarquía de directorios proporciona un espacio de nombres para los archivos. Hasta este momento solamente disponemos de los sistemas de archivos creados en las particiones correspondientes. Para poder crear archivos y directorios en estos sistemas de archivos es necesario hacerlos visibles en el espacio de nombres. Para ello se utiliza la orden **mount**.

Hablamos de *montaje de un sistema de archivos* en el sentido de añadir una nueva rama al espacio de nombres actualmente en uso, de manera que, tras completarse la orden, toda la información del sistema de archivos montado será accesible en el espacio de nombres. Así mismo, se podrán crear nuevos archivos y directorios en esta nueva rama del espacio de nombres. Para montar un sistema de archivos es necesario solamente indicar en qué directorio se montará (aquí es donde crecerá la nueva rama del espacio de nombres), y cuál es el nombre del archivo especial de dispositivo que representa la partición en donde reside el sistema de archivos. El directorio que se utiliza como punto de partida para el sistema de archivos montado se denomina *punto de montaje*.

Cuando ya no es necesario acceder a la información del sistema de archivos montado se puede llevar a cabo una operación de desmontaje del SA. Esta operación provoca que el sistema de archivos deje de estar disponible en el espacio de nombres y además permite realizar todas las actualizaciones en disco de los metadatos del SA de forma que quede en estado consistente.

NOTA: No es posible desmontar un SA si está siendo utilizado (*busy*).

Actividad 6.1. Montaje de sistemas de archivos.

Utiliza el manual en línea para descubrir la forma de montar nuestros SAs de manera que cumplas los siguientes requisitos:

El SA etiquetado como **LABEL_ext3** debe estar montado en el directorio **/mnt/SA_ext3** y en modo de solo lectura.

El SA etiquetado como **LABEL_ext4** debe estar montado en el directorio **/mnt/LABEL_ext4** y debe tener sincronizadas sus operaciones de E/S de modificación de directorios.

Como vimos en la sesión anterior, `/etc/fstab` es el archivo de configuración que contiene la información sobre todos los sistemas de archivos que se pueden montar y de las zonas de intercambio a activar. El formato del archivo se describe a continuación:

<file system> <mount point> <type> <options> <dump> <pass>

- **<file system>**, es el número que identifica el archivo especial de bloques .
- **<mount point>**, es el directorio que actúa como punto de montaje.
- **<type>**, tipo de sistema de archivos (ext2, ext3, ext4, vfat, iso9660, swap, nfs, etc.)
- **<options>**, son las opciones que se utilizarán en el proceso de montaje. Se especifican como una lista separada por comas y sin espacios.
- **<dump>**, normalmente no se usa, pero si su valor es distinto de 0 indica la frecuencia con la que se realizará una copia de seguridad del SA.
- **<pass>** , durante el arranque del sistema este campo especifica el orden en el que la orden fsck realizará las comprobaciones sobre los SAs.

A continuación se muestran las posibles opciones que pueden especificarse en el campo **<options>**:

- **rw**. Lectura-escritura
- **ro**. Sólo lectura
- **suid/nosuid**. Permitido el acceso en modo SUID, o no permitido
- **auto/noauto**. Montar automáticamente o no montar automáticamente (ni ejecutando mount -a)
- **exec/noexec**. Permitir la ejecución de ficheros, o no permitir
- **usrquota, grpquota**. Cuotas de usuario y de grupo
- **defaults** = *rw,suid,dev,exec,auto,nouser,async*
- **user, users, owner**. Permitir a los usuarios montar un sistema de archivos
- **uid=500, gid=100**. Propietario y grupo propietario de los archivos del SA.
- **umask**. Máscara para aplicar los permisos a los archivos.

Actividad 6.2. Automontaje de Sistemas de archivos

Escribe las dos líneas necesarias en el archivo `/etc/fstab` para que se monten automáticamente nuestros dos SA en el arranque del sistema con los mismos requisitos que se han pedido en la actividad 6.1.

7. Administración de software

7.1 Introducción

Las distribuciones del sistema operativo incluyen habitualmente varios GigaBytes de software. Pero tarde o temprano se requerirá la búsqueda de nuevo software, o actualizaciones del ya previamente instalado. Por ejemplo, entre las responsabilidades de un administrador de un sistema operativo están las tareas de instalar nuevo software de aplicación que los usuarios necesitan ejecutar, o bien mantener la seguridad del sistema mediante actualizaciones. Estas tareas se han de llevar a cabo como administrador del sistema (usuario *root*). Sin embargo, ten en cuenta que otras tareas asociadas como compilación de código fuente y construcción de código ejecutable se pueden llevar a cabo como usuario distinto del administrador. Cuando sea posible, adquirir el hábito de realizar estas tareas bajo un usuario normal del sistema habitualmente puede ayudar a evitar algunos problemas, en el caso que nos ocupa, en cuanto a cambios no deseados en la configuración actual del software del sistema.

Existen multitud de lugares y sitios Web, aparte de los oficiales de las distribuciones Linux: Fedora (<http://fedoraproject.org/>), Ubuntu (<http://www.ubuntu.com/>) y Red Hat (<http://www.redhat.com/>), por nombrar algunas de las más conocidas, donde encontrar gran cantidad de software. Por ejemplo podéis visitar los siguientes sitios Web: <http://freshmeat.net/>, <http://www.tucows.com/linux>, <http://sourceforge.net/>.

El software a instalar en sistemas Linux se estructura en paquetes. Un **paquete software** es un archivo que contiene un conjunto de programas complementarios, y en la mayoría de los casos éstos tienen dependencias con otros paquetes. La especificación de las dependencias (como requeridos o incluso sugeridos) que tiene un paquete software con respecto a otros está incluida en el propio paquete como metadatos. Los metadatos también incluyen guiones (scripts) de ayuda, atributos de archivo, y otra información descriptiva acerca del paquete.

Los paquetes pueden incluir archivos con programas en código binario, o bien código en formato de código fuente junto con las instrucciones para generar el código binario. Tal como se presenta en la siguiente sección, las distribuciones de sistemas operativos incluyen diferentes herramientas gráficas y de línea de orden para la administración de paquetes software, las cuales normalmente se conocen como **gestores de paquetes**.

Actividad 7.1

Accede a los sitios web especializados que ofrecen software para sistemas operativos Linux y enumera las principales características de cada uno de ellos en base, por ejemplo, a si contiene software abierto y/o propietario, tipos de aplicaciones disponibles, tamaño del sitio en cuanto a la cantidad de software que mantiene, y otras características que considere interesantes.

7.2 Gestores de paquetes

La forma más simple de instalar y actualizar software en un sistema es mediante el uso de los gestores de paquetes habitualmente proporcionados en la propia distribución del correspondiente sistema operativo.

Los paquetes software fundamentalmente se identifican dependiendo del sistema operativo y/o distribución concreta dentro de éste. Por ejemplo los paquetes de aquellos sistemas operativos Linux basados en formato Debian, entre ellos el más conocido es Ubuntu, se identifican por tener la extensión “**.deb**” en el nombre del archivo del paquete, la parte anterior a la extensión es el nombre del propio paquete y número de versión. Si la distribución se basa en el el formato del sistema operativo Red Hat (el sistema operativo más conocido es Fedora), entonces el paquete se identifica por la extensión “**.rpm**”. Los paquetes software RPM (Redhat Package Manager) además incluyen en el nombre un identificador acerca del procesador donde fueron compilados los programas incluidos en el paquete, esto es:

- i386: procesador genérico de la familia x86
- i586 e i686: procesadores Pentium o incluso superiores
- sparc: procesador SPARC de Sun Microsystems
- ...

Existe un gran número de gestores de paquetes con características diferentes, y como se ha comentado, su disponibilidad depende del sistema operativo y/o distribución específica. La siguiente tabla muestra algunos de los gestores más usuales, y por tanto conocidos, en relación con los formatos más extendidos: Debian (*.deb*) y Red Hat (*RPM*). En caso de que un mismo gestor de paquetes se encuentre disponible para ambos formatos, el carácter asterisco (*) indica cual fue el formato originario del gestor. En caso de querer usar un gestor de paquetes que no sea el propio de la distribución, es necesario instalarlo previamente. Además, se puede utilizar un conversor de formato, denominado Alien, entre paquetes RPM y debian.

Tabla 3. Clasificación de los principales gestores de paquetes.

	Formato <i>.deb</i> (Debian, Ubuntu)	Formato <i>RPM</i> (Red Hat)
Modo Línea de Órdenes	dpkg; apt-get*; aptitude* (interfaz texto para apt)	rpm; YUM; apt-get; aptitude (interfaz texto para apt)
Modo Gráfico	dselect; Synaptic*; Adept (basado en apt-get); Kpackage (parte de kdeadmin)	Synaptic; pup, pirut y yumex (basados en YUM); Adept (basado en apt-get); gpk-application (parte de gnome-packagekit); ; Kpackage (parte de kdeadmin)

Dentro de los gestores de paquetes en modo línea de órdenes se distinguen dos niveles:

- *Alto nivel* como es el caso de los gestores de paquetes *apt-get* y *YUM*, sobre los que a su vez se suelen proporcionar interfaces gráficas, es por ejemplo el caso de *PackageKit* y *gnome-packagekit* sobre *YUM*.
- *Bajo nivel* tales como *dpkg* y *rpm*.

Los gestores de alto nivel suelen ser más interactivos y pueden realizar búsquedas, actualizaciones automáticas, incluyendo análisis de dependencias y procesamiento de paquetes obsoletos en base a almacenes de metadatos, realización de consultas sobre los paquetes instalados, etc. En cambio, los gestores de bajo nivel pueden llegar a ser más precisos y potentes. Las siguientes subsecciones describen algunos de estos paquetes con más detalle.

7.2.1 APT y YUM

APT (*Advanced Packaging Tool*) fue originalmente desarrollado por Debian Linux y modificado para su uso también con paquetes RPM. Tanto APT, como su interfaz gráfico Synaptic, son fáciles de utilizar. Existen varios proveedores principales de paquetes y repositorios para APT, por ejemplo <http://www.freshrpms.net>, pero no se deben utilizar simultáneamente porque a veces existen conflictos entre versiones.

Algunos desarrolladores piensan que para gestionar paquetes RPM es mejor herramienta YUM (*Yellow dog Updater, Modified* - <http://yum.baseurl.org>) que APT. Además, parece que APT contiene más código innecesario que se utiliza realmente para los paquetes .deb. Las siguientes órdenes son muy útiles al usar YUM:

Tabla 4. Algunas órdenes útiles en YUM.

yum list	Lista los paquetes disponibles en los repositorios para su instalación
yum list installed	Lista los paquetes actualmente instalados
yum list updates	Muestra todos los paquetes con actualizaciones disponibles en los repositorios para su instalación
yum install <nombre-paquete>	Instala el paquete cuyo nombre es especificado
yum update	Se actualizan todos los paquetes instalados
yum remove <nombre-paquete>	Elimina el paquete cuyo nombre es especificado, así como los paquetes que dependen de éste

Existen varios interfaces gráficos que utilizan directamente YUM tales como los mencionados anteriormente: *PackageKit* y *gnome-packagekit*. Aunque YUM sólo proporciona una interfaz para línea de órdenes, resulta muy cómodo y fácil de usar. Puede ejecutar la siguiente orden en el *shell* para obtener un listado más completo de las órdenes y opciones disponibles en YUM; la configuración del repositorio se tiene en cuenta en todas las operaciones de YUM.

```
#> yum --help | more
```

Actividad 7.2

Encuentra los archivos de configuración de YUM y explore las distintas órdenes disponibles en YUM ejecutándolas. En concreto, lista todos los paquetes instalados y disponibles, elimina el paquete instalado que te indique el profesor de prácticas, y a continuación vuelve a instalar el mismo paquete haciendo uso de los paquetes que se encuentran disponibles en */fenix/depar/lsi/so/paquetes*. Para obtener acceso a este directorio del sistema de archivos anfitrión ejecute la siguiente orden de montaje una vez lanzado el sistema operativo *User Mode Linux* (UML):

```
#> mount none /<directorio-punto-montaje> -t hostfs -o /fenix/depar/lsi/so/paquetes
```

Tenga en cuenta que algunas órdenes de YUM pueden no funcionar bien debido a que no pueden acceder a los sitios web que mantienen repositorios de paquetes, ya que no existe conexión a Internet en la configuración actual del sistema operativo UML. Para descargar paquetes de uno de los principales sitios web encargados de mantener repositorios de paquetes en código binario y fuente, con el navegador Web acceda a <https://admin.fedoraproject.org/pkgdb>. Puede realizar

la búsqueda de un paquete binario específico introduciendo parte del nombre y pulsando el botón *BUILD*, o puede acceder al listado completo de paquetes pulsando sobre *Builds* en el menú de navegación que se encuentra a la izquierda.

7.2.2 RPM

El gestor de paquetes RPM descende del primer software de gestión de paquetes Linux. RPM comprueba dependencias e incluye opciones como la verificación de la revisión y las firmas de seguridad de privacidad GNU que permiten que los paquetes se distribuyan con seguridad, esto es, libres de virus.

La bases de datos en nuestra computadora registran las versiones de los paquetes instalados, véase directorio */var/lib/rpm* que es el que utiliza RPM. De éstas, la principal base de datos que guarda los paquetes instalados corresponde al archivo *Packages*.

El formato general para la línea de órdenes se muestra a continuación, acceda al manual de RPM, o ejecute *rpm -help*, para obtener los un descripción detallada de las opciones disponibles:

```
#> rpm <opciones> <nombres-paquetes>
```

El gestor RPM dispone de más de 60 opciones de línea de órdenes que se pueden agrupar en cinco tipos de operaciones sobre el software, cada tipo se caracteriza por las opciones más comunes que se describen en la siguiente tabla. En el manual de RPM podrá obtener detalle de todas las opciones.

Tabla 5. Clasificación de tipos de operaciones en el gestor RPM y órdenes más comunes.

Tipo de Función	Órdenes más comunes	Descripción
Instalación de nuevos paquetes	rpm -i <nombre-archivo-paquete>	Si la operación tiene éxito no se mostrará ningún mensaje
Eliminación de paquetes instalados	rpm -e <nombre-paquete>:	Si la operación tiene éxito no se mostrará ningún mensaje
Actualización de paquetes instalados	rpm -U <nombre-archivo-paquete>	La actualización se consigue descargando el paquete que corresponde a la nueva versión y ejecutando RPM con la opción -U , que además incluye la eliminación automática de la versión del paquete previamente instalada
	rpm -F <nombre-servidor-HTTP/FTP>	Se buscará el paquete en el servidor designado en Internet y se preparará la correspondiente actualización

Obtención de información sobre paquetes software	<code>rpm -qa grep <parte-nombre-paquete-buscado> sort</code>	Esta línea de órdenes puede utilizarse para buscar paquetes instalados por su nombre, o por parte de éste
	<code>rpm -qi <nombre-paquete></code>	Muestra información precisa del paquete instalado que se especifica
Verificación e integridad de la instalación	<code>rpm -V <nombre-paquete></code>	Consulta en la base de datos para verificar la instalación de un paquete recientemente instalado. Si la instalación se ha realizado correctamente, la orden no produce información de salida.

Algunos repositorios en donde puedes encontrar paquetes **.rpm** son:

- <http://pkgs.org>
- <http://www.rpmseek.com/index.html>
- <http://www.rpmfind.net>
- <http://rpm.pbone.net>

Actividad 7.3

En primer lugar deseamos mostrar cierta metainformación acerca de uno o más paquetes ya instalados. Para ello debes utilizar la orden **rpm** con las opciones adecuadas. Utiliza el manual en línea si no sabes ya las opciones que debes utilizar.

1. Muestra la información general (nombre, versión, arquitectura, grupo, descripción, etc.) y lista los archivos que contiene un paquete ya instalado haciendo uso de la orden **rpm** y un único conjunto de opciones.
2. Idem que el anterior pero mostrando únicamente los archivos de configuración que contiene el paquete.
3. Escribe una orden que muestre los paquetes requeridos por un paquete determinado que se encuentre instalado en el sistema. Escribe la orden que devuelva el mismo resultado pero para un paquete no instalado en el sistema.
4. Instala el paquete **quota** que encontrarás en el directorio de software de la asignatura (directorío que ya has montado en la actividad 7.2).
5. Instala y desinstala el paquete **sysstat** mostrando en pantalla también la máxima información posible acerca del propio proceso de eliminación del paquete.

8. Administración de cuotas.

Como se comentó en la tabla 1 del apartado 3, existe una forma más flexible de limitar el uso de disco por parte de los usuarios del sistema. Este control se lleva a cabo mediante un sistema de **cuotas**. Las cuotas de disco permiten limitar el número de recursos de un SA que va a poder utilizar un usuario. Estos recursos son los bloques de disco y los i-nodos. El sistema de cuotas permite aprovechar el concepto de grupo de usuarios para establecer límites sobre el conjunto de usuarios incluidos en un determinado grupo.

Para poder trabajar con el sistema de cuotas en Linux es necesario tener instalado el paquete **quota**, el cual normalmente no se instala por defecto en las distribuciones.

Tradicionalmente el sistema de cuotas establece dos tipos de límites de cara a restringir el uso de bloques e i-nodos. Los límites se pueden establecer para usuarios y/o grupos y para bloques y/o i-nodos.

- Límite *hard*. El usuario no puede sobrepasarlo. Si llegara el caso en el cual lo sobrepasase, el sistema no le permitirá usar más bloques, por lo que no podrá ampliar el tamaño de sus archivos ya creados (salvo que la ampliación de tamaño no requiera usar un nuevo bloque); ni tampoco usar más i-nodos, por lo que no podrá crear nuevos archivos.
- Límite *soft*. Siempre debe configurarse como un número de recursos inferior al límite *hard* y se puede sobrepasar durante cierto tiempo, pero sin llegar a superar al límite *hard*. Transcurrido el tiempo que estipule el administrador para poder estar por encima del límite *soft*, el sistema de cuotas actúa como si se hubiese superado el límite *hard*. Este tiempo durante el cual se puede superar el límite *soft* se conoce con el nombre de *periodo de gracia*.

Actividad 8.1. Establece un sistema de cuotas para el sistema de archivos tipo **ext3**.

En esta actividad se van a presentar los pasos que necesitas llevar a cabo para establecer el sistema de cuotas de disco en Linux. El objetivo será activar el sistema de cuotas sobre el sistema de archivos tipo **ext4** que has creado con anterioridad.

1. Editar el archivo **/etc/fstab** y activar el sistema de cuotas de usuario para el SA tipo **ext3**. Busca cómo se especifica esta opción en el manual en línea. Una ayuda para la búsqueda es que la realices sobre la orden **mount** y recuerdes que las opciones de montaje vienen especificadas en los apartados: **FILESYSTEM INDEPENDENT MOUNT OPTIONS** y **FILESYSTEM SPECIFIC MOUNT OPTIONS**.

2. Montar de nuevo el SA en el espacio de nombres para que se active la opción previamente establecida. Usa la siguiente orden:

```
#> mount -o remount <directorio_punto_de_montaje>
```

3. Crear el archivo que permite llevar el control de cuotas de usuario para el SA. El nombre de este archivo es **aquota.user**. Para ello utiliza la siguiente orden:

```
#> quotacheck -nm <directorio_punto_de_montaje>
```

4. Ahora procedemos a activar el sistema de control de cuotas de usuario. Para ello ejecuta la orden:

```
#> quotaon -a
```

5. Ahora solo falta editar la cuota para cada usuario del sistema mediante la siguiente orden. En este caso, establece los parámetros para cada usuario existente. Puede ser buena idea utilizar el archivo `/etc/passwd` para localizar los nombres.

```
#> edquota username
```

6. Para finalizar estableceremos el periodo de gracia para el límite soft.

```
#> edquota -t
```

En la siguiente tabla se muestran algunas órdenes útiles para el administrador a la hora de trabajar con el sistema de control de cuotas.

Tabla 6. Órdenes útiles para el trabajo con cuotas.

quota username	Asignación de las cuotas para un usuario.
repquota <SA>	Estadística de las cuotas para todos los usuarios .

Actividad 8.2

Establece los límites de bloques e i-nodos para un par de usuarios del sistema UML sobre el que trabajas en el laboratorio.