

Prácticas de Sistemas Operativos

Módulo I. Administración de Linux

Sesión 1. Herramientas de administración básicas

1. Introducción

Esta sesión está pensada, en primer lugar, para que el estudiante se familiarice con el entorno de trabajo sobre el que se desarrollarán la mayoría de las sesiones de prácticas del Módulo I de la asignatura.

Cuando el estudiante consiga disponer de una consola en modo root en el laboratorio de prácticas, abordará el apartado de “**Administración de usuarios y grupos en Linux**” (**apartado 4**). En este apartado el estudiante podrá comprender los siguientes conceptos: Usuario, cuenta de usuario, tipos de usuarios, grupo de usuarios, tipos de grupos. Además, para afianzar estos conceptos utilizará los programas de utilidad básicos para su gestión.

Finalmente, el estudiante abordará el apartado de “**Organización del Sistema de Archivos (SA) y gestión básica de archivos**” (**apartado 5**). Este apartado permitirá al alumno enlazar los conocimientos adquiridos previamente en la asignatura Fundamentos del Software, relativos a archivos, y extenderlos con la comprensión y utilización de programas de utilidad para la gestión de archivos. Además, le permitirá conocer la propuesta para estandarizar la estructura jerárquica de los sistemas de archivos Linux (**FSH**, del inglés **Filesystem Hierarchy Standard**).

2. Objetivos principales

- Conocer y saber usar el entorno de trabajo donde se desarrollará el módulo de prácticas de administración de SO Linux.
- Conocer los tipos de usuarios de un sistema operativo Linux y sus funciones.
- Saber crear cuentas de usuarios y grupos.
- Conocer la organización del estándar FHS.
- Saber utilizar las órdenes básicas para gestionar un sistema de archivos Linux.

3. Obtención de privilegios de root en el laboratorio

Para realizar las prácticas sobre administración del sistema operativo linux es necesario disponer de acceso al sistema como usuario **root**. En las aulas de prácticas podemos ejecutar linux pero solamente como un usuario normal, sin más privilegios. Por ello, vamos a ejecutar un kernel que nos permitirá disponer de privilegios modo root. Para ello realizaremos una secuencia de pasos. Como utilizaremos este kernel durante el desarrollo de las prácticas del módulo de administración será conveniente automatizar mediante un script dichos pasos.

Los pasos son:

- Tras arrancar el ordenador del laboratorio, elegiremos la opción del sistema operativo que indique el profesor
- Copiaremos los archivos de la ruta especificada a un directorio temporal:

```
cp /fenix/depar/lsi/UML/*.gz /tmp
```

- Descomprimos los archivos copiados en el directorio `/tmp` ya que tienen la extensión `.gz`. Para ello podemos usar la orden `gunzip`.
- Ejecutamos en `/tmp`:

```
./kernel132-3.0.4 ubda=./Fedora14-x86-root_fs mem=1024m
```

- Arranca una ejecución de un sistema linux Fedora con un kernel 3.0.4 e introducimos el nombre de usuario: **root** y, como contraseña, ninguna.

A continuación vamos a enumerar una serie de observaciones útiles para el desarrollo de las prácticas de administración de sistemas Linux.

- No existe modo gráfico en el sistema operativo sobre el que trabajamos en modo root, ya que no disponemos de espacio suficiente para que arranque, por lo que trabajaremos durante todas las prácticas con una consola de terminal (*terminal console*).
- Recordad que lo que hay en un directorio temporal, `/tmp`, se borrará cuando salgamos de la sesión y apaguemos el sistema. Los archivos son muy grandes para copiarlos en la cuenta (os quedaríais pronto sin espacio) por lo que el primer ejercicio que vais a hacer es automatizar este proceso.
- Cualquier sistema Linux arranca automáticamente seis procesos para autenticación en el sistema que muestran un *prompt* de *login* en modo texto (consola de terminal) y un proceso para autenticación en modo gráfico. Podemos acceder al sistema desde cualquiera de estos procesos y podemos alternar la visualización en pantalla de cada uno de ellos utilizando la siguiente combinación de teclas: **Ctrl+Alt+{F1-F7}**. Concretamente, la combinación permite acceder a la consola gráfica **Ctrl+Alt+F7**, y el resto de combinaciones de teclas permiten acceder a consolas modo texto.
- La característica mostrada en el apartado anterior nos puede servir para utilizar determinados programas o utilidades que estén instaladas en el SO anfitrión pero no en el sistema de archivos, **Fedora14-x86-root_fs**, que es el que utiliza el programa **kernel132-3.0.4**. Por ejemplo, como el programa **man** para acceder al manual en línea no está instalado, podemos utilizar el **man** accediendo desde otro terminal de entrada (p.e. **Ctrl+Alt+F3**) al sistema operativo, que es el sistema anfitrión desde donde arrancamos inicialmente, y ejecutar el **man** ya que aquí si que está instalado.

- Puesto que no vamos a usar archivos muy grandes, podemos usar el **vi** como editor de textos.

Actividad 3.1

Crea un *script de bash* que automatice todos los pasos vistos en este punto y que guardarás preferiblemente en tu directorio home. Al entrar de nuevo en el sistema sólo tendrás que ejecutar el script para empezar a trabajar en modo root.

4. Administración de usuarios y grupos en Linux

4.1 Usuario administrador del sistema en Linux: root

El usuario administrador del sistema, o también llamado **superusuario**, es el usuario que tiene siempre todos los privilegios sobre cualquier archivo, instrucción u orden del sistema. En Linux, y en cualquier sistema UNIX, este usuario se identifica con el nombre de usuario **root** que pertenece al grupo **root** y cuyo directorio home es **/root**.

El administrador del sistema debe tener amplios conocimientos de todo el sistema (hardware, software, datos, usuarios,...), una buena capacidad para tomar decisiones, ser eficaz y responsable ya que trabaja con datos muy importantes.

Las tareas asignadas a un administración del sistema suelen ser:

- Añadir nuevos usuarios. Recordemos que Linux es un sistema operativo multiusuario.
- Controlar el rendimiento del sistema
- Realizar las copias de seguridad (y restaurarlas...)
- Añadir/eliminar elementos hardware
- Instalar/actualizar/desinstalar software
- Controlar la seguridad del sistema
- Controlar el correcto arranque del sistema
- Monitorización del sistema
- Localizar y resolver problemas del sistema
- Resolver dudas de los usuarios
- etc.

Convertirse en **administrador** implica entrar al sistema como usuario **root**, o si ya se ha iniciado un shell con otro usuario, podemos usar la orden **su** que pedirá la contraseña del **root** y lanzará un proceso nuevo que ejecutará el mismo programa shell pero como usuario **root**.

```
$> whoami          # pido a la shell el nombre de usuario actual
patricia

$> su              # solicitud de cambio a usuario root
```

```

Password:          # el sistema solicita la contraseña de root

#> whoami          # suele cambiar el prompt del sistema
root

```

Nota: Cuando estemos trabajando con un sistema Linux/UNIX, por seguridad, deberíamos hacerlo siempre bajo una cuenta de usuario normal y no usar la cuenta del administrador, del root, a no ser que queramos realizar tareas de administración. Debemos tener en cuenta que el administrador es el usuario con el mayor privilegio.

4.2 Gestión de usuarios

Un usuario (**user**) es una persona que trabaja en el sistema mediante una cuenta de usuario a la que accede mediante una identificación. En Linux, un usuario se caracteriza por:

- Su nombre de usuario, también conocido como *username*.
- Su identificador de usuario (**UID**, del inglés User IDentifier) que es un número entero que le asigna internamente el sistema y que lo representa (el sistema operativo no trabaja con su nombre sino con su UID). El **UID** del root es el 0.
- El grupo o grupos a los que pertenece (**GID**, del inglés Group IDentifier). Un usuario tiene asignado un **grupo principal** (*primary group*) que es el grupo que aparece especificado en el archivo `/etc/passwd`, pero puede pertenecer a más de un grupo. Los grupos adicionales a los que puede pertenecer un usuario se denominan **grupos suplementarios** (*supplementary groups*). Todos los grupos y los usuarios que pertenecen a cada grupo están especificados en el archivo `/etc/group`. El GID principal del superusuario es el 0.

Como acabamos de ver, la información relativa a usuarios y grupos es almacenada por el SO en varios archivos. A continuación se muestra una tabla que incluye el nombre de estos archivos y una breve descripción de su contenido. Muestra estos archivos por pantalla para familiarizarte con el formato y contenido que almacenan.

Tabla 1. Archivos que especifican los usuarios, grupos y contraseñas (passwords) del sistema.

| | |
|--------------------------|--|
| <code>/etc/passwd</code> | Almacena información de las cuentas de usuarios |
| <code>/etc/shadow</code> | Guarda los password encriptados e información de “envejecimiento” de las cuentas |
| <code>/etc/group</code> | Definición de los grupos y usuarios miembros |

4.2.1. Creación de cuentas de usuario

Para añadir un nuevo usuario al sistema se han de realizar una serie de pasos que enumeramos a continuación y que se llevan a cabo utilizando las órdenes que veremos más adelante:

1. Decidir el nombre de usuario (*username*) y los grupos a los que va a pertenecer el usuario que utilizará esa cuenta (grupo principal y grupos suplementarios).
2. Introducir los datos en los archivos `/etc/passwd` y `/etc/group`, rellenando el campo correspondiente a la contraseña (*password*) con el carácter “x”.

3. Asignar un password a la nueva cuenta de usuario.
4. Establecer los parámetros de envejecimiento de la cuenta.
5. Crear el *directorio de inicio* del nuevo usuario (HOME) normalmente en el directorio del sistema **/home**, establecer el propietario y grupo correspondiente y los permisos adecuados.
6. Copiar los archivos de inicialización (**.bash_profile**, **.bashrc**,...)
7. Establecer otras facilidades: cuotas, mail, permisos para imprimir, etc.
8. Ejecutar cualquier tarea de inicialización propia del sistema.
9. Probar la nueva cuenta.

Las herramientas automáticas para la creación de cuentas de usuario suelen realizar todas las tareas básicas del proceso anterior, a excepción de las específicas (cuotas o impresión, etc.). Las órdenes para la creación de cuentas de usuario son:

```
$> useradd # ó...
$> adduser
```

Se creará el usuario y su grupo principal, así como las entradas correspondientes en **/etc/passwd**, **/etc/shadow** y **/etc/group**. También se creará el directorio de inicio, normalmente en **/home**, y los *archivos de configuración* particulares para cada usuario que se ubican dentro de este directorio y que se detallan más adelante. Si ejecutamos dichas órdenes sin argumentos, nos muestran la lista de opciones por *stderr* (salida de error estándar).

Esta orden toma los valores por defecto que se le van a asignar al usuario (a su cuenta) a partir de la información especificada en los archivos **/etc/default/useradd** y **/etc/login.defs**.

Actividad 4.1

Visualiza el contenido de los dos archivos anteriores y comprueba cuáles son las opciones por defecto que tendría un usuario que se creara en nuestro sistema. A continuación, crea una cuenta de usuario y visualiza el contenido de los archivos **/etc/passwd** y **/etc/group**, y el directorio **/home** para comprobar que los nuevos datos se han rellenado conforme a la especificación tomada de **/etc/default/useradd** y **/etc/login.defs**.

Para modificar los valores asociados a una cuenta, disponemos de las siguientes órdenes:

Tabla 2. Órdenes para gestión de cuentas de usuario.

| | |
|-----------------|---|
| usermod | modifica una cuenta de usuario ya existente |
| userdel | elimina una cuenta de usuario (por defecto no borra el directorio HOME) |
| newusers | crea cuentas de usuarios utilizando la información introducida en un archivo de texto, que ha de tener el formato del archivo |

| | |
|----------------------------------|-----------------------------|
| | <code>/etc/passwd</code> |
| <code>system-config-users</code> | herramienta en modo gráfico |

En el directorio `/etc/skel` se guardan unos archivos de configuración del shell, los cuales se copian al directorio HOME asignado cuando se crea una cuenta de usuario. Posteriormente, cada usuario podrá personalizar su copia. Estos archivos son guiones shell que realizan determinadas tareas como inicializar variables, ejecutar funciones específicas, establecer los alias, etc. Estos archivos dependen del intérprete de órdenes seleccionado y en el caso del bash son:

Tabla 3. Archivos de configuración para el shell Bash.

| | |
|----------------------------|--|
| <code>.bash_profile</code> | se ejecuta al hacer el login (conectarnos al sistema) y en él podremos indicar alias, variables, configuración del entorno, etc. que deseamos iniciar al principio de la sesión. |
| <code>.bashrc</code> | su contenido se ejecuta cada vez que se ejecuta una shell, tradicionalmente en este archivo se indican los programas o scripts a ejecutar. |
| <code>.bash_logout</code> | se ejecuta al salir el usuario del sistema y en él podremos indicar acciones, programas, scripts, etc., que deseamos ejecutar al salirnos de la sesión. |

Actividad 4.2

1. Utiliza el manual en línea para leer la sintaxis completa de la utilidad para creación de cuentas y crea dos o tres usuarios en tu sistema cambiando alguno de los valores por defecto.
2. Elimina alguno de ellos y comprueba que “rastros” ha dejado la cuenta recién eliminada en el sistema.
3. Entra (orden `su`) en el sistema como uno de estos usuarios que has creado y mira qué archivos tiene en su directorio home. La orden `sudo` permite cambiar el modo de trabajo a modo root específicamente para ejecutar una orden con privilegios de supervisor y tras su ejecución continuar con los privilegios del usuario que abrió la sesión.

Actividad 4.3

Visualiza el archivo `/etc/passwd` e indica cual es el formato de cada línea de dicho archivo. Para ello también puedes consultar el `man` o `info` de Linux. ¿Quién es el propietario de este archivo y cuáles son sus permisos?

4.2.2. Cambio de contraseña (*password*)

Para asignar una contraseña a un usuario se puede usar la orden `passwd`. Esta orden también se utiliza para cambiar la contraseña del usuario que la ejecuta si no se le indica ningún argumento. El usuario administrador puede cambiar las contraseñas de todos los usuarios del sistema,

incluyendo la suya propia, pero, por razones obvias, un usuario normal solamente podrá cambiar la contraseña asociada a su propia cuenta. A continuación se muestra la orden:

```
$> passwd <nombre_usuario>
```

Podemos asignar o cambiar el intérprete de órdenes por defecto que usará el usuario cuando se conecte al sistema. En el último campo del archivo `/etc/passwd` se establece para cada usuario el intérprete de órdenes que se ejecuta por defecto cada vez que entra al sistema. En el archivo `/etc/shells` se indican los shells permitidos, es decir los que están instalados en el sistema. Con la orden `chsh` el usuario puede cambiar su shell (el nuevo ha de estar entre los permitidos). Si un usuario no tiene asignado ningún intérprete de órdenes, se usará el shell por defecto, representado por el archivo `/bin/sh`. Si se desea que el usuario no pueda entrar al sistema se le puede asignar al campo que indica el shell los nombres de archivo `/bin/false` o `/sbin/nologin`. También se puede establecer como shell un *archivo ejecutable*, de forma que cuando el usuario entre al sistema se ejecutará dicho archivo y al finalizar su ejecución se acabará la sesión de trabajo del usuario.

Actividad 4.4

Visualiza el archivo `/etc/shadow` desde un usuario distinto al `root` ¿Te da algún problema? ¿Sabes por qué? Intenta averiguarlo.

4.2.3. Parámetros de configuración de una cuenta.

Para las cuentas de los usuarios se pueden establecer restricciones de tiempo, también llamadas de envejecimiento, respecto a la validez de la cuenta o de la contraseña. Los valores se guardan en el archivo `/etc/shadow`. La siguiente tabla muestra los valores posibles.

Tabla 4. Valores para controlar el envejecimiento de contraseñas y cuotas.

| | |
|-----------------|--|
| changed | fecha del último cambio de contraseña |
| minlife | número de días que han de pasar para poder cambiar la contraseña |
| maxlife | número de días máximo que puede estar con la misma contraseña sin cambiarla |
| warn | cuántos días antes de que la contraseña expire (maxlife) será informado sobre ello, indicándole que tiene que cambiarla |
| inactive | número de días después de que la contraseña expire que la cuenta se deshabilitará de forma automática si no ha sido cambiada |
| expired | fecha en la que la cuenta expira y se deshabilita de forma automática |

Los valores los establece el administrador con las órdenes **chage** o con **passwd**. Recordemos que el archivo `/etc/login.defs` tiene los valores por defecto. La siguiente tabla muestra algunas opciones y argumentos útiles para la orden **chage**.

Tabla 5. Posibles escenarios de uso de la orden **chage**.

| | |
|-----------------------------------|--|
| chage -d ult_día usuario | fecha del último cambio de password |
| chage -m min_días usuario | nº de días que han de pasar para poder cambiar la contraseña |
| chage -M max_días usuario | nº de días máximo que puede estar con la misma contraseña sin cambiarla |
| chage -W warn_días usuario | cuántos días antes de que la contraseña expire (maxlife) será avisado de ello, indicándole que tiene que cambiarla |
| chage -I inac_días usuario | nº de días después de que la contraseña expire que la cuenta se deshabilitará de forma automática si la contraseña no ha sido cambiada |
| chage -E exp_días usuario | fecha en la que la cuenta expira y se deshabilita de forma automática |

4.3 Gestión de grupos

Un grupo es un conjunto de usuarios que comparten recursos o archivos del sistema. Con los grupos se pueden garantizar permisos concretos para un conjunto de usuarios, sin tener que repetirlos cada vez que se desee aplicarlos.

Un grupo se caracteriza por:

- Nombre del grupo, o **groupname**
- Identificador del grupo (**GID**, del inglés Group Identifier) que es un número que permite al sistema identificar al grupo (ver sección 4.2).
- Archivo de configuración **/etc/group**. Cada línea de este archivo presenta el siguiente formato: **nombre:x:gid:lista de usuarios**

Tabla 6. Órdenes relacionadas con la gestión de grupos.

| | |
|-----------------------|---|
| groupadd grupo | crea un nuevo grupo |
| groupmod grupo | modifica un grupo existente |
| groupdel grupo | elimina un grupo |
| newgrp grupo | cambia de grupo activo (lanza un shell con ese grupo) |

| | |
|------------------------------------|---|
| <code>gpasswd grupo</code> | asigna una contraseña a un grupo |
| <code>gpasswd -a user grupo</code> | añade un usuario a un grupo |
| <code>groups [usuario]</code> | informa de los grupos a los que pertenece un usuario |
| <code>id [usuario]</code> | lista el identificador del usuario y los grupos a los que pertenece |
| <code>grpck</code> | comprueba la consistencia del archivo de grupos |

Actividad 4.5

1. Crea un par de grupos y asignáselos a algunos de los usuarios de tu sistema.
2. ¿Qué información devuelve la orden `id` si estás conectado como root?

4.4 Usuarios y grupos especiales

Los usuarios especiales son aquellos que no están asociados a personas físicas. A continuación se muestran dos tablas, la primera incluye los nombres de usuario de algunos usuarios especiales comunes en los sistemas UNIX y la segunda, incluye grupos estándar que se encuentran en la mayoría de los sistemas UNIX.

Tabla 7. Usuarios especiales del sistema.

| | |
|--|---|
| <code>root</code> | Usuario administrador del sistema |
| <code>bin, daemon, lp, sync, shutdown,...</code> | Tradicionalmente usados para poseer archivos o ejecutar servicios |
| <code>mail, news, ftp,...</code> | Asociados con herramientas o utilidades |
| <code>postgres, mysql, xfs,...</code> | Creados por herramientas instaladas en el sistema para administrar y ejecutar sus servicios |
| <code>nobody ó nfsnobody</code> | Usada por NFS y otras utilidades |

Tabla 8. Grupos estándar del sistema.

| | |
|---|---|
| <code>root, sys, bin, daemon, adm, lp, disk, mail, ftp, nobody</code> | Algunos de los nombres de grupo <i>preconfigurados</i> por los sistemas UNIX. Los GIDs inferiores a 500 están reservados para estos grupos. |
| <code>tty, dialout, disk, cdrom, audio, video</code> | Nombres de grupo específicos para dispositivos. |
| <code>kernel</code> | Grupo propietario de los programas para leer |

| | |
|--------------|---|
| | la memoria del kernel |
| users | Puede usarse como grupo por defecto para todos los usuarios normales del sistema. |

5. Organización del sistema de archivos y gestión básica de archivos

Como se vió en la primera sesión de prácticas de la asignatura Fundamentos del Software, la organización de los archivos en un sistema de archivos presenta una estructura jerárquica en forma de árbol en donde los nodos interiores (internos) están representados por los directorios y los nodos finales (nodos hoja) están representados normalmente por los archivos asociados con un directorio. Podemos tener un directorio que no contenga ningún archivo, por lo que sería un nodo hoja, pero entonces, ¿para qué lo queremos si no contiene nada!

En esta sesión, se explicó el concepto de archivo y el concepto de nombres de archivo, ya que un archivo, en la estructura jerárquica de directorios/archivos puede ser referenciado (nombrado) de dos formas distintas: de manera absoluta (su nombre empieza por "/" en sistemas UNIX), o de manera relativa (su nombre no empieza por "/").

Con respecto a la estructura jerárquica se enumeraron algunos de los directorios que comúnmente se encuentran en cualquier SO tipo UNIX/Linux y se describió brevemente la información que deberían contener. Esta descripción está basada en el **FHS** (del inglés, Filesystem Hierarchy Standard) que veremos en más profundidad al final de esta sección.

Como colofón a los conceptos de archivos introducidos en la primera sesión de prácticas de Fundamentos del Software se introdujo el estándar de Internet **MIME** (del inglés, Multipurpose Internet Mail Extensions) que se utiliza en los SO para establecer asociaciones de tipo de archivo con la aplicación que es capaz de "entender" (o que generó) los datos que almacena dicho archivo.

Sin embargo, el concepto de *tipo de archivo* asociado a los sistemas UNIX es distinto. Este concepto no está asociado a los programas de aplicación y sus archivos de datos, aunque por supuesto estos también son archivos, sino más bien a todos los elementos de información y procesamiento que pone a nuestra disposición el SO. A continuación se describe de manera genérica parte de la información que se almacena en el sistema de archivos.

- El programa que contiene el *kernel* del SO, el cual se carga en el arranque del sistema, junto con todos los programas de utilidad disponibles normalmente en el sistema de archivos raíz (identificado por el símbolo "/" cuando se configuran las particiones). En Linux, el archivo "**vmlinux***" o "**vmlinuz***"¹ es un archivo ejecutable, enlazado estáticamente, que contiene el *kernel* de Linux en uno de los formatos de archivo objeto soportados por Linux: **ELF**, **COFF** o **a.out**. Con respecto a los programas de utilidad, ya se vieron algunos ejemplos en la asignatura Fundamentos del Software.
- Una de las características de los sistemas de archivos de los SOs tipo UNIX es que actúan como interfaz de alto nivel para el acceso a los dispositivos de E/S. De esta forma, se pueden utilizar las órdenes para trabajar con archivos normales (**regulares**, en terminología UNIX/Linux) para, a través del interfaz que proporcionan los **archivos especiales de dispositivo**, operar con los dispositivos de E/S. Todos estos archivos de dispositivo suelen estar ubicados en el directorio **/dev**, sobre todo si nuestra distribución de

¹ El "*" significa que aquí van el resto de símbolos que indican la versión del *kernel* que contiene ese archivo.

Linux sigue el estándar **FHS**, que es lo más común. Algunos ejemplos de estos tipos de archivo son: `/dev/sda`, `/dev/sda1` y `/dev/tty`.

- Archivos para establecer comunicaciones para transferencia de información entre procesos. En Linux disponemos de dos formas de establecer comunicación entre procesos mediante archivos: archivos de **tipo Socket** y archivos de **tipo FIFO** (*pipes*, que se suele traducir como cauces o tuberías).
- Los archivos de **tipo directorio** están encargados de soportar la estructura jerárquica del sistema de archivos. Su estructura lógica de almacenamiento de información es mediante registros en la que cada registro almacena el nombre de un archivo (de cualquier tipo incluido directorios) y, en los sistemas UNIX/Linux, una referencia a los *atributos* del archivo. Claramente, uno de estos atributos será el tipo de archivo.
- Los sistemas de archivos tipo UNIX permiten establecer nombres de archivo distintos para un mismo archivo. De esta forma un archivo se puede referenciar desde diferentes directorios dentro de la estructura de directorios. Estos tipos de archivos son muy útiles desde el punto de vista de la administración del sistema en aspectos como la portabilidad de aplicaciones y el ocultamiento de detalles de versión. Este tipo de archivos se conoce con el nombre de archivos de **tipo enlace**. Los sistemas UNIX permiten establecer dos tipos de enlaces sobre archivos: **enlace duro** (*hard link*) y **enlace simbólico** (*soft link*).

Actividad 5.1

- Utilizando la orden (**find**) que ya conoces para la búsqueda de archivos en el sistema de archivos, anota el nombre absoluto del archivo del kernel de Linux que se ha cargado en el sistema operativo que estás usando en el laboratorio de prácticas para acceso modo root.

5.1 Organización común en sistemas de archivos tipo Linux. Filesystem Hierarchy Standard (FHS)

El FHS es un estándar que propone una forma sistemática de organizar toda la información que almacena un sistema operativo tipo Linux. La Fundación Linux (Linux Foundation) es la encargada de mantener este estándar. Esta organización integra entre sus miembros a muchas de las empresas líderes en el sector de la informática, tanto hardware como software. Para más información puedes visitar la página: <http://www.linuxfoundation.org/>

En el primer apartado de prácticas ya has podido conocer uno de los directorios importantes recogidos en el estándar FHS, el directorio `/etc`. Básicamente en este directorio podemos encontrar todos los archivos de configuración del sistema. Obviamente, para un administrador de sistema conocer y comprender el contenido de este directorio es fundamental para el correcto desempeño de su labor. A continuación se muestra una tabla con algunos de los directorios que es interesante conocer y que recoge este estándar. Para más información puedes visitar la página: http://en.wikipedia.org/wiki/Filesystem_Hierarchy_Standard

Tabla 9. Nombres de directorio y tipo de información que almacenan en el estándar FHS.

| | |
|--------------------|--|
| <code>/bin</code> | Programas de utilidad fundamentales para ser utilizados por cualquier usuario del sistema. |
| <code>/sbin</code> | Programas de utilidad fundamentales para ser utilizados por el usuario root . |
| <code>/boot</code> | Archivos fundamentales para el programa Boot Loader . |

| | |
|---------------|--|
| /dev | Todos los archivos especiales de dispositivo. |
| /etc | Archivos de configuración del sistema. |
| /home | Los directorios de inicio de todos los usuarios que disfrutan de una cuenta en el sistema, excepto, el directorio de inicio del root : /root |
| /lib | Bibliotecas sin las que no pueden funcionar los programas ubicados en /bin y /sbin . |
| /media | Este directorio actúa como <i>punto de montaje</i> para dispositivos extraíbles: DVD-ROM, dispositivos USB, etc. |
| /mnt | Este directorio actúa como <i>punto de montaje</i> para sistemas de archivos montados temporalmente. |
| /opt | Normalmente aquí se ubican los programas que no forman parte de la distribución instalada en el sistema. |
| /proc | Sistema de archivos virtual que hace de interfaz con el núcleo y los procesos. |
| /tmp | Archivos temporales que normalmente no se mantienen una vez se apaga el sistema. |
| /usr | Archivos ejecutables, archivos de código fuente, bibliotecas, documentación y, en general, todos los programas y utilidades. |
| /var | Los archivos cuyo contenido se espera que cambie durante el funcionamiento normal del sistema. |

Actividad 5.2

Un programa que se ejecuta en modo root, ¿dónde podría guardar la información temporal de forma que ésta se mantuviese entre arranques del sistema?

5.2 Órdenes básicas para gestión del sistema de archivos.

En la asignatura Fundamentos del Software se utilizaron diversas órdenes para explorar la estructura jerárquica de directorios/archivos: **pwd**, **ls** y **cd**; para crear y borrar directorios y archivos: **mkdir**, **rmdir**, **cat** y **rm**; y para copiar y mover archivos y directorios a distintas ubicaciones dentro de la estructura jerárquica: **cp** y **mv**. También se vieron algunas órdenes para modificar los atributos de los archivos: **chmod** y **touch**, así como para consultar algunos de sus atributos: **file** y **ls**.

En nuestras prácticas el objetivo principal es el trabajo con el sistema de archivos, es decir, no nos centraremos en órdenes que realizan su funcionalidad sobre archivos sino sobre todo el sistema de archivos en su conjunto. La funcionalidad que requiere un administrador del sistema en este ámbito incluye, pero no está reducida a:

- Acceso a información del SO relativa a sistemas de archivos.
- Ampliación de la estructura jerárquica de directorios mediante la orden `mount`.
- Instalación y configuración de nuevos dispositivos de almacenamiento y creación de sistemas de archivos sobre estos.
- Comprobación del estado del sistema de archivos.
- Cuotas de disco.

En esta sesión abordaremos exclusivamente algunas órdenes que nos permitirán acceder a la información relativa a qué sistemas de archivos tenemos disponibles actualmente en nuestra estructura de directorios y los atributos con los que fueron montados estos sistemas de archivos.

5.2.1 Acceso a información del SO relativa a sistemas de archivos.

Si pensamos en acceso a información del sistema entonces sabemos que, según el estándar FHS, algo deberíamos encontrar en el directorio `/etc`. Efectivamente, aquí disponemos de dos archivos fundamentales para obtener información de los sistemas de archivos: `/etc/fstab` y `/etc/mtab`.

Actividad 5.3

Los archivos `/etc/fstab` y `/etc/mtab` muestran información sobre los sistemas de archivos que se encuentran montados en el sistema. ¿Cuál es la diferencia entre la información que muestra cada uno de ellos?

La información que muestra el archivo `/etc/fstab` es muy útil para comprender las opciones de montaje que se han realizado para cada uno de los sistemas de archivos que tenemos accesibles en nuestro sistema. A continuación se muestran algunas de las opciones que refleja este archivo:

- Modo de acceso a los archivos del sistema de archivos: `{rw|ro}`, lectura/escritura o sólo lectura.
- Modo de acceso SUID: `{suid|nosuid}`, si/no.
- Montaje automático: `{auto|noauto}`, se permite o no el montaje automático. En el caso de no permitirlo no se realizará el montaje ni utilizando la orden `mount -a` (la veremos en la siguiente sesión).
- Ejecución de archivos: `{exec|noexec}`, si/no.
- Cuotas de usuario y de grupo: `usrquota`, `grpquota`.
- Valores por defecto de montaje (`defaults`): `rw`, `suid`, `dev`, `exec`, `auto`, `nouser`, `async`
- Permitir a los usuarios montar un sistema de ficheros : `user`, `users`, `owner`.
- Propietario y grupo propietario de los ficheros del SA : `uid=500`, `gid=100`
- Máscara a aplicar en los permisos de los archivos de nueva creación: `umask=022`

Actividad 5.4

Edita el archivo `/etc/fstab` del sistema de archivos que estás utilizando en modo root y anota y describe la información que tiene registrada. Si no conoces alguna opción puedes consultar el manual en línea: `man fstab`.

Otro directorio del FHS que nos va a ser muy útil para obtener información es el `/proc`, el cual soporta el sistema de archivos virtual `proc`. Este directorio contiene archivos de texto que permiten acceder a información de estado del sistema. En este apartado solamente utilizaremos los archivos que tienen información relativa los sistemas de archivos pero es conveniente familiarizarse con él porque se utilizará en varios apartados del módulo de administración del SO Linux.

Tabla 10. Archivos con información de sistema de archivos que proporciona `/proc`.

| | |
|--------------------------------|---|
| <code>/proc/filesystems</code> | Enumera, uno por línea, todos los tipos de sistemas de archivos disponibles. |
| <code>/proc/mounts</code> | Sistemas de archivos montados actualmente, incluyendo los que se hayan montado manual o automáticamente tras el arranque del sistema. |

Actividad 5.5

Compara la información que contienen los cuatro archivos de texto que se han presentado en este apartado (`/etc/fstab`, `/etc/mtab`, `/proc/filesystems` y `/proc/mounts`). Describe en un párrafo para qué te sirve la información que registra cada archivo.