Ediciones ENI

LINUX

Preparación para la certificación LPIC-1

(exámenes LPI 101 y LPI 102)

4ª edición

Colección
Certificaciones

Extracto del Libro

Requisitos previos

☒Saber utilizar el shell y sus comandos.

☒Disponer de un editor de texto.

☒Estar en posesión de la contraseña root.

☒Disponer de un almacenamiento físico o virtual modificable.

☒Conocer los principios del funcionamiento del hardaware de un ordenador.

Objetivos

Al final de este capítulo, será capaz de:

☒Reconocer los discos y las particiones.

☒Manejar los parámetros de los periféricos.

☒Elegir entre los diferentes sistemas de archivos.

☒Crear particiones.

☒Crear un sistema de archivos.

☒Crear puntos de montaje y modificar el archivo fstab.

☒Controlar, modificar y reparar un sistema de archivos.

☒Crear espacios de swap.

☒Instalar cuotas.

☒Gestionar los permisos y los propietarios de los archivos.

☒Conocer el principio de los hard links.

A. Representación de los discos

Nota previa: las unidades de medida de almacenamiento usadas en este capítulo y en todo el libro usan la representación tradicional, según la regla 1KB = 1024 bytes (2^{10}), a no ser que se indique lo contrario.

1. Nomenclatura

Este apartado realiza un repaso a los puntos ya vistos en el capítulo Presentación de Linux. En función del tipo de controlador e interfaz en los cuales se conectan los discos, Linux da diferentes nombres a los archivos especiales que representan discos duros.

Cada disco y cada partición está representado por un archivo especial de tipo bloque.

a. IDE

Los discos con controladores IDE (también llamados PATA, *Parallel Ata* o ATAPI) se llaman hdX:

- hda: IDE0, Master
- hdb: IDE0, Slave
- hdc: IDE1, Master
- hdd: IDE1, Slave
- etc.

Contrariamente a lo que se cree, no hay límite al número de controladores IDE, más allá del número de los puertos de extensión de la máquina (slots PCI). Hay muchas tarjetas adicionales y placas base que disponen de cuatro, seis, ocho conectores. En estos casos, los archivos especiales reciben los nombres hde, hdf, hdg, etc.

Linux entiende que los lectores de CD-Rom, DVD y grabadores de tipo IDE/ATAPI son discos IDE y respetan la nomenclatura citada.

Los últimos núcleos de Linux utilizan por defecto un API llamado libata para acceder al conjunto de los discos IDE, SCSI, USB, Firewire, etc. En este caso (puede comprobarlo dirigiéndose a las notas de versión de la distribución), la nomenclatura sigue la de los discos SCSI, que tratamos en el punto siguiente.

☞*De hecho, la representación de los discos con nomenclatura hdx se ha vuelto rara. Sin embargo tenga cuidado. Si actualiza su antigua distribución que mantiene ese formato, el paso a una nueva versión más reciente probablemente cambiará el nombre a sdx, la nomenclatura SCSI. Así que puede experimentar problemas si olvida modificar el archivo /etc/fstab o la configuración de grub.*

b. SCSI, SATA, USB, FIREWIRE, etc.

Los discos con controladores SCSI, SCA, SAS, FiberChannel, USB, Firewire, thunderbolt (y probablemente otras interfaces exóticas, como los lectores ZIP en puerto paralelo) se llaman sdX. La enumeración de los discos sigue el orden de detección de las tarjetas SCSI y de los adaptadores (hosts) asociados, más la adición o supresión manual de otras interfaces de discos duros mediante hotplug o udev.

- sda: primer disco SCSI
- sdb: segundo disco SCSI
- sdc: tercer disco SCSI

- etc.

La norma SCSI marca una diferencia entre los diversos soportes. Así, los lectores de CD-Rom, DVD, HD-DVD, BlueRay y los grabadores asociados no llevan el mismo nombre. Los lectores y grabadores están en srX (sr0, sr1, etc.). También puede encontrar scd0, scd1, etc. Pero suelen ser vínculos simbólicos hacia sr0, sr1, etc.

El comando **lsscsi** permite enumerar los periféricos SCSI. Observe que los discos son sdX, mientras que el lector dvd es srX.

```
$ lsscsi
[4:0:0:0]    disk    ATA      ST380011A        8.01  /dev/sda
[5:0:0:0]    cd/dvd  LITE-ON  COMBO SOHC-4836V S9C1  /dev/sr0
[31:0:0:0]   disk    USB2.0   Mobile Disk      1.00  /dev/sdb
```

2. Casos especiales

a. Controladores específicos

Algunos controladores no siguen esta nomenclatura. Por ejemplo, es el caso de algunos controladores RAID físicos. Hay que verlo caso por caso. Un controlador Smart Array en un servidor HP, que utilice el controlador cciss, coloca sus archivos de periféricos en /dev/cciss con los nombres cXdYpZ, donde X es el slot, Y el disco y Z la partición...

b. Virtualización

La representación de discos de sistemas invitados (*guests*) virtualizados depende del tipo de controlador simulado. La mayoría son de tipo IDE o SCSI, y en ambos casos muy a menudo con libata son vistos como SCSI. Sin embargo, algunos sistemas, como por ejemplo KVM o XEN que ofrece paravirtualización, disponen de un controlador específico que presenta los discos con el nombre vd_x (virtual disk x o xvd_x):

- vda: primer disco virtualizado, o vxda.
- vdb: segundo disco virtualizado, o vxbd.
- etc.

c. SAN, iSCSI, multipathing

Los discos conectados a través de una SAN (*Storage Area Network*, generalmente con fibra óptica) o mediante iSCSI se ven como discos SCSI y conservan esta nomenclatura. Sin embargo, los sistemas de gestión de rutas múltiples (*multipathing*) se ubican por debajo, proporcionando otros nombres. Powerpath llamará a los discos emcpowerx (emcpowera, emcpowerb, etc.) mientras que el sistema por defecto de Linux llamado multipath los llamará mpathx (mpath0, mpath1, etc.) o de cualquier otro modo elegido por el administrador.

B. Operaciones de bajo nivel

1. Información

El comando **hdparm** permite efectuar un gran número de operaciones directamente en los discos duros gestionados por la librería libata, o sea todos los discos SATA, ATA (IDE) y SAS. El comando **sdparm** puede hacer más o menos lo mismo para los discos SCSI. Observe que, a pesar de que los nombres de periféricos de la libata sean idénticos a los del SCSI, es más que probable que muchas opciones de configuración de hdparm no funcionen en discos SCSI. Lo mismo vale para sdparm con los discos SATA o IDE. Los ejemplos que damos a continuación se basan en hdparm.

Para obtener información completa relativa a un disco, utilice los parámetros `-i` o `-I`. El primero recupera la información, desde el núcleo, que se obtiene en el momento del arranque. El segundo interroga directamente al disco. Es preferible `-I` porque da una información muy detallada.

```
# hdparm -I /dev/sda

/dev/sda:

ATA device, with non-removable media
    Model Number:       VBOX HARDDISK
    Serial Number:      VB91a2e953-933cdc65
    Firmware Revision:  1.0
Standards:
    Used: ATA/ATAPI-6 published, ANSI INCITS 361-2002
    Supported: 6 5 4
Configuration:
    Logical         max    current
    cylinders    16383    16383
    heads        16    16
    sectors/track    63    63
    --
    CHS current addressable sectors:   16514064
    LBA    user addressable sectors:   63152320
    LBA48  user addressable sectors:   63152320
    Logical/Physical Sector size:           512 bytes
    device size with M = 1024*1024:       30836 MBytes
    device size with M = 1000*1000:       32333 MBytes (32 GB)
    cache/buffer size  = 256 KBytes (type=DualPortCache)
Capabilities:
    LBA, IORDY(cannot be disabled)
    Queue depth: 32
    Standby timer values: spec'd by Vendor, no device specific minimum
    R/W multiple sector transfer: Max = 128        Current = 128
    DMA: mdma0 mdma1 mdma2 udma0 udma1 udma2 udma3 udma4 udma5 *udma6
         Cycle time: min=120ns recommended=120ns
    PIO: pio0 pio1 pio2 pio3 pio4
         Cycle time: no flow control=120ns  IORDY flow control=120ns
```

```
Commands/features:
    Enabled    Supported:
       *    Power Management feature set
       *    Write cache
       *    Look-ahead
       *    48-bit Address feature set
       *    Mandatory FLUSH_CACHE
       *    FLUSH_CACHE_EXT
       *    Gen2 signaling speed (3.0Gb/s)
       *    Native Command Queueing (NCQ)
Checksum: correct
```

2. Modificación de los valores

Se puede modificar varios parámetros de los discos. Sin embargo, ¡cuidado! Algunas opciones de **hdparm** pueden resultar peligrosas tanto para los datos contenidos en el disco como para el propio disco. La mayoría de los parámetros son de lectura y escritura. Si no se especifica ningún valor, **hdparm** muestra el estado del disco (o del bus) para este comando. A continuación le presentamos algunos ejemplos de opciones interesantes.

- **-c**: anchura del bus de transferencia EIDE en 16 o 32 bits. 0=16, 1=32, 3=32 compatible.
- **-d**: utilización del DMA. 0=no DMA, 1=DMA activado.
- **-X**: modifica el modo DMA (mdma0 mdma1 mdma2 udma0 udma1 udma2 udma3 udma4 udma5). Puede utilizar cualquiera de los modos anteriores o valores numéricos: 32+n para los modos mdma (n varía de 0 a 2) y 64+n para los modos udma.
- **-C**: modo de ahorro de energía en el disco (unknown, active/idle, standby, sleeping). Se puede modificar el estado con -S, -y, -Y y -Z.
- **-g**: muestra la geometría del disco.
- **-M**: indica o modifica el estado del Automatic Acoustic Management (AAM). 0=off, 128=quiet y 254=fast. No todos los discos lo soportan.
- **-r**: pasa el disco en sólo lectura.
- **-T**: bench de lectura de la caché del disco, ideal para probar la eficacia de transferencia entre Linux y la caché del disco. Hay que volver a ejecutar el comando dos o tres veces.
- **-t**: bench de lectura del disco, fuera de la caché. Mismas observaciones que para la opción anterior.

Así, el comando siguiente pasa el bus de transferencia a 32 bits, activa el modo DMA en modo Ultra DMA 5 para el disco sda:

```
# hdparm -c1 -d3 -X udma5 /dev/sda
```

Le mostramos a continuación otros ejemplos:

```
# hdparm -c /dev/sda

/dev/sda:
 IO_support    =  0 (default 16-bit)

# hdparm -C /dev/sda
```

```
/dev/sda:
 drive state is:  active/idle

# hdparm -g /dev/sda

/dev/sda:
 geometry      = 3931/255/63, sectors = 63152320, start = 0

# hdparm -T /dev/sda

/dev/sda:
 Timing cached reads:   23868 MB in  2.00 seconds = 11950.45 MB/sec
# hdparm -t /dev/sda

/dev/sda:
 Timing buffered disk reads: 308 MB in  3.02 seconds = 101.87 MB/sec
```

C. Elegir un sistema de archivos

1. Fundamentos

a. Definición de sistema de archivos

La acción de “formatear” un disco, un pendrive o cualquier soporte de datos consiste únicamente en crear en un soporte de memoria secundaria (volumen de almacenamiento) la organización lógica que permite colocar datos en él. La palabra “formateo” en Linux se utiliza para describir la creación de un sistema de archivos. Hablamos de un sistema de archivos que representa a la vez la organización lógica de los soportes tanto a un nivel inferior como a un nivel de usuario.

No se escribe la información en los discos de cualquier manera. Se requiere una mínima organización para colocar en ellos tanto la información relativa a los archivos como los datos almacenados. El sistema de archivos (y los controladores asociados) es el que define esta organización. Si bien los fundamentos organizativos suelen ser los mismos en los diferentes sistemas de archivos presentes soportados por Linux, las implementaciones y organizaciones lógicas de los datos en el disco varían bastante de uno a otro. De esta manera, no hay un único tipo de sistema de archivos, sino varios, puestos a disposición del usuario, el administrador o el ingeniero.

Todos los sistemas de archivos de Linux deben respetar las normas POSIX. Como POSIX define un conjunto de reglas básico, un sistema de archivos puede ir más lejos de esta norma ofreciendo extensiones. La mayoría de estas conciernen a elementos de seguridad, como las ACL o selinux.

El principio básico es asociar un nombre de archivo con su contenido y autorizar su acceso: creación, modificación, supresión, desplazamiento, apertura, lectura, escritura, cierre. Conforme a este principio, el sistema de archivos debe gestionar lo que deriva de ello: mecanismos de protección de los accesos (permisos, propietarios), accesos concurrentes, etc.

Ediciones ENI

LINUX

Preparación para la certificación LPIC-2

(exámenes LPI 201 y LPI 202)

3ª edición

Colección
Certificaciones

Extracto del Libro

Capítulo 5

Requisitos

Los conocimientos adquiridos con la certificación LPI nivel 1, especialmente:

☒Conocer la estructura del archivo /etc/passwd.

☒Conocer la existencia y los fundamentos del archivo hosts.

Objetivos

Al final de este capítulo, usted será capaz de:

☒Interpretar una configuración NSS.

☒Comprender la autentificación modular PAM.

☒Conocer los principales módulos PAM.

☒Modificar la configuración PAM para permitir cambios en la forma de autentificarse.

☒Conocer el formato de los archivos LDIF.

☒Consultar un directorio LDAP.

☒Administrar las contraseñas en un directorio OpenLDAP.

☒Añadir o modificar elementos de un directorio OpenLDAP.

☒Configurar la autentificación de un sistema Linux en un directorio OpenLDAP.

A. Evolución de la autentificación

1. Los primeros sistemas Unix y el archivo passwd

a. Contraseñas en el archivo /etc/passwd

Desde su aparición, los sistemas Unix utilizan el archivo **/etc/passwd** como base de datos de cuentas de usuarios. Este archivo se utiliza de forma natural para abrir sesiones en el sistema. Como su nombre todavía indica, albergaba, además de los identificadores de los usuarios, sus contraseñas cifradas. Si algún otro elemento distinto del de apertura de sesión necesita información sobre las cuentas (conexión ftp, apertura de sesión remota, etc.), también consultará este archivo. En esta sencilla situación inicial, hay una única base de datos de cuentas de usuario y múltiples aplicaciones que la usan. Todas las aplicaciones tienen que reconocer el formato de esta base de datos.

b. Contraseñas en el archivo /etc/shadow

Con la evolución de las técnicas de ataque de contraseñas, se hizo necesario mover las contraseñas a un archivo no accesible a usuarios normales. Para ello, se almacenaron en un nuevo archivo: **/etc/shadow** cerrado a los usuarios. Los parámetros de autentificación con shadow se administran mediante el archivo **/etc/login.defs**. Los parámetros almacenados en este archivo son en general adecuados.

Gestión de errores de autentificación en el archivo login.defs

De la gran cantidad de parámetros del archivo login.defs, los que están relacionados con el login son los que se modifican con mayor frecuencia.

```
usuario@ubuntu:~$ grep LOGIN /etc/login.defs
LOGIN_RETRIES 5
LOGIN_TIMEOUT 60
usuario@ubuntu:~$
```

2. Otras bases de datos

En la consulta de datos de identificación, la situación se complicó cuando aparecieron otras bases de datos de cuentas diferentes del archivo **passwd** y, sobre todo, más complejas. A menudo, estas bases de datos de identidad están centralizadas, como es el caso de NIS (*Network Information Server*) o LDAP (*Lightweight Directory Access Protocol*). La primera solución propuesta fue, por supuesto, volver a escribir los programas que originalmente operaban con el archivo **/etc/passwd** para que fueran capaces de consultar las bases de datos centralizadas en red. Este método carecía claramente de flexibilidad debido a que obligaría a rehacer una gran cantidad de programas en profundidad cada vez que apareciese un cambio o una nueva forma de almacenamiento en las bases centralizadas.

3. NSS

NSS (*Name Service Switch*) es una primera respuesta a la multiplicidad de bases de datos locales o centralizadas. NSS tiene como objetivo normalizar la resolución de nombres en un sistema. NSS permite resolver un nombre obteniendo la información asociada, como por ejemplo un nombre de usuario y su uid, un nombre de grupo y su gid o incluso un nombre de host y su dirección IP.

En el funcionamiento de NSS, el archivo **/etc/nsswitch.conf** determina para distintos tipos de resoluciones la fuente de información que se debe consultar. Las aplicaciones que necesiten esta información consultarán las fuentes en el orden impuesto por el archivo **nsswitch.conf**. De este modo, la resolución se apoya en librerías NSS (**libnss_*X*.so** donde *X* representa el servicio de resolución empleado) y las aplicaciones no necesitan conocer directamente el método de resolución empleado.

Formato del archivo nsswitch.conf

resolución: fuente_1 fuente_n

nsswitch.conf: formato del archivo	
resolución	El tipo de resolución que se realizará.
fuente_1	Obligatorio. La primera fuente de resolución que se usará.
fuente_n	Opcional. La fuente o las otras fuentes de resolución posibles que se utilizarán después de la primera.

Ejemplo de archivo nsswitch.conf

En este ejemplo se puede ver que las resoluciones de tipo password, group y shadow se realizarán mediante la librería libnss_compat.so y que la resolución de nombres de host se realizará mediante las librerías libnss_files.so y libnss_dns.so. Esto significa que los elementos de identificación de los usuarios se encontrarán en los archivos locales de /etc y que la resolución de nombres de host se realizará en primer lugar mediante el archivo local (/etc/hosts) antes de utilizar el servicio dns.

```
passwd:         compat
group:          compat
shadow:         compat

hosts:          files dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

☞*En un sistema Linux moderno, NSS ya solo se usa para operaciones de identificación, es decir, encontrar información de una entidad. Todo lo relativo a la autentificación se realiza en un mecanismo más elaborado: PAM.*

4. Módulos de autentificación

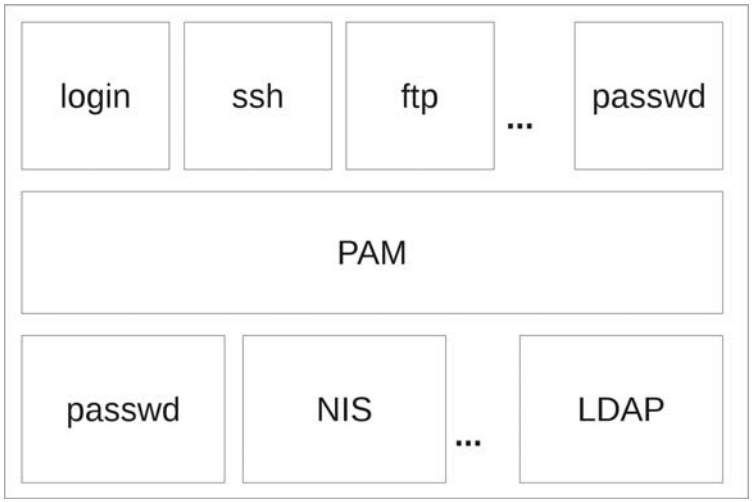
Si NSS ya representa un progreso en relación a los archivos estáticos usados en los primeros años, la revolución llega con PAM (*Pluggable Authentication Module*). PAM es un mecanismo complementario de NSS que proporciona una autentificación a medida mediante la ejecución de módulos a la elección del administrador.

Cuando se abre una sesión en Linux, el usuario tiene que presentar su identificador y una contraseña. Gracias a la resolución NSS, se deducirán los identificadores uid/gid, así como el resto de los parámetros necesarios (fecha de expiración, etc.). En lo que respecta a PAM, ejecutará según su configuración un módulo u otro para proporcionar la autentificación, pero también puede realizar ciertas tareas ligadas a la apertura de sesión, como por ejemplo la definición de variables.

B. PAM

1. El principio

PAM se sitúa como un intermediario entre las aplicaciones y los métodos de autentificación.



El objetivo principal de PAM es proporcionar una capa de abstracción entre las aplicaciones y los métodos de autentificación. De este modo, una aplicación que quiera ser flexible y evolutiva en cuanto a los métodos de autentificación que emplea solo deberá ser compatible con PAM. Esto significa que deberá ser capaz de dirigirse a la capa de autentificación PAM, sin importarle todo lo que hay detrás. Paralelamente, los procedimientos de autentificación, sean cuales sean, deberán ser accesibles y utilizables por el mecanismo PAM.

Una aplicación solicita a PAM si un usuario se puede conectar. PAM, en función de su configuración, invocará los módulos que utilizarán un método de autentificación. Si el resultado es positivo (el usuario ha proporcionado los elementos correctos de autentificación), PAM devuelve la autorización de conexión a la aplicación.

PAM tiene otra ventaja. Acabamos de ver que la solicitud de autentificación entrañaba la carga de módulos. Pues bien, el número de módulos no tiene límites y estos se pueden acumular. Por lo tanto, se puede solicitar una doble autentificación siguiendo dos métodos de autentificación distintos. Además, se puede sacar provecho de la autentificación con PAM para provocar la carga de librerías sin relación con la autentificación. Por lo tanto, es posible desencadenar muchas acciones una vez que se ha realizado con éxito la autentificación.

En resumen: cuando se solicita a un usuario que se autentifique, los módulos PAM se cargan en función de un archivo de configuración y estos módulos provocan ciertas acciones, que pueden ser la propia autentificación u otras acciones.

2. Los módulos PAM

a. Los módulos PAM principales

Los módulos PAM, invocados cuando se producen operaciones de autentificación, son muchos y están enfocados a distintos usos. Algunos de ellos, sin embargo, se utilizan con mucha frecuencia y hay que conocerlos. Otros son menos frecuentes y dependen de la distribución que se esté usando. No obstante, conocer su funcionamiento y su finalidad permite comprender mejor la mecánica y la filosofía de PAM.

Estos módulos están en archivos cuya ubicación estándar es **/lib/security**.

Módulos PAM principales	
pam_securetty.so	Prohíbe el login para la cuenta root excepto en los terminales listados en /etc/securetty.
pam_nologin.so	Si el archivo /etc/nologin existe, muestra su contenido ante cualquier intento de apertura de sesión y prohíbe el login ante cualquier usuario que no sea root.
pam_env.so	Declara las variables de entorno que se leen en /etc/environment o en el archivo al que se hace referencia con el parámetro «envfile=».
pam_unix.so	Permite la autentificación mediante el método tradicional de los archivos /etc/passwd y /etc/shadow.
pam_deny.so	Vía muerta. Generalmente se ejecuta si ningún otro módulo se ha ejecutado con éxito.
pam_permit.so	Devuelve un resultado positivo incondicionalmente.
pam_limits.so	Asigna ciertas limitaciones funcionales a usuarios o grupos en función de los datos del archivo /etc/security/limits.conf.
pam_cracklib.so	Se asegura que la contraseña empleada presenta un nivel de seguridad suficiente.
pam_selinux.so	Si selinux está activo en el sistema, este módulo va a asegurar que el shell se ejecuta en el contexto de seguridad adecuado.
pam_lastlog.so	Muestra la información de la última apertura de sesión con éxito.
pam_mail.so	Comprueba la presencia de nuevos correos para un usuario (mensajería interna).