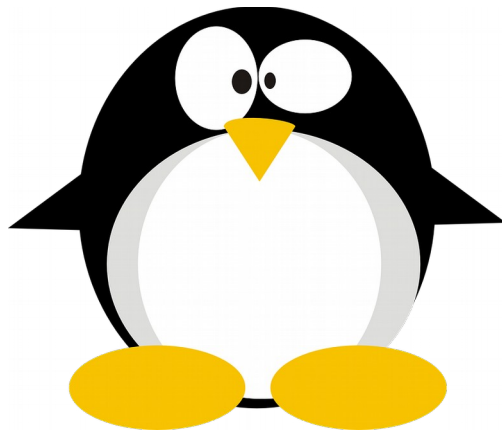


LINUX ESSENTIALS



CURSO RÁPIDO

NOTAS DEL AUTOR

Este manual ha sido creado por **Ramón Lozano Juan** para el curso “**Linux Essentials**” impartido en el acuartelamiento Capitán Sevillano con el propósito de formar técnicamente al personal del regimiento de transmisiones 22 y ha sido puesto a disposición del alumnado para ser usado como una guía de referencia rápida.

El temario se ajusta al contenido impartido por el docente en el curso presencial de dos semanas en fecha de Enero de 2017.

La información contenida en este documento se centra en el aprendizaje y el uso de la interfaz de línea de comandos.

Descripciones y ejemplos han sido simplificados para proporcionar una visión genérica.

Para obtener más información sobre el uso de las herramientas descritas se recomienda consultar las páginas de manual integradas en el sistema operativo.

Se permite la libre distribución y/o modificación del documento.

ÍNDICE DE TEMAS

1 – INTRODUCCIÓN A BASH

2.1 – INTRODUCCIÓN AL SISTEMA DE FICHEROS

2.2 – USUARIOS Y GRUPOS

2.3 – PROPIEDAD Y PERMISOS

3.1 – ACCESO A INFORMACIÓN DEL HARDWARE

3.2 – ALMACENAMIENTO LOCAL

3.3 – OPERACIONES EN SISTEMAS DE FICHEROS

4.1 – PROCESOS

4.2 – INICIO DEL SISTEMA

5.1 – SOFTWARE EN GNU/LINUX

5.2 – LIBRERÍAS

5.3 – GESTIÓN DE SOFTWARE EN DEBIAN Y DERIVADOS

5.4 – GESTIÓN DE SOFTWARE EN RED HAT Y DERIVADOS

6 – KERNEL LINUX

7.1 – GESTIÓN DE REDES

7.2 – RESOLUCIÓN DE NOMBRES

7.3 – REDES INALÁMBRICAS

8 – INTRODUCCIÓN AL FILTRADO DE PAQUETES

ANEXOS

A. EVOLUCIÓN HISTÓRICA DE UNIX Y GNU/LINUX

B. COMPARATIVA GNU/LINUX – WINDOWS

C. ENLACES DE INTERÉS

D. BIBLIOGRAFÍA

TEMA 1 – INTRODUCCIÓN A BASH

CLI

Interfaz de línea de comandos

Tipos

- Bourne shell → **bsh** , **bash**
- C shell → **cs**h , **tc**sh
- Korn shell → **k**sh , **z**sh

Visualizar las shells instaladas

```
$cat /etc/shells
```

Bash (*Bourne Again Shell*)

Intérprete de comandos que ejecuta órdenes leídas desde la entrada estándar o desde un fichero

Tipos de comandos

- **Internos**
 - Comandos integrados en el shell
 - No generan un nuevo proceso al ejecutarse
- **Externos**
 - Ficheros ejecutables en el sistema de ficheros
 - Generan un proceso hijo del shell al ser ejecutados

Comandos internos básicos

\$pwd		Muestra directorio actual
\$cd		Cambiar directorio
\$echo		Imprime sus argumentos
\$time [comando]		Cronometra un comando
\$exit	\$logout	Cierra el shell
\$help		Lista los comandos internos
\$alias		Lista los alias
\$history		Lista el historial
\$type		Proporciona información sobre un comando
\$source	\$.	Ejecuta un fichero en una subshell y exporta las variables
\$export [VAR]		Exporta una variable al entorno de sistema
\$unset [VAR]		Elimina una variable o función

Sintaxis CLI

```
$comando [opciones] [argumento/s]
```

Agrupación de comandos

```
$comando1 ; comando2 ; comando3 ...
```

El shell espera que cada comando termine por orden.

Códigos de retorno

Testifican el desarrollo de la última ejecución

\$echo \$?

- = 0 ejecución correcta
- $\neq 0$ ejecución incorrecta (error)

Comandos internos de test

\$true	Devuelve el código de retorno 0
\$false	Devuelve el código de retorno 1

Ejecuciones condicionales

\$comando1 && comando2	(AND)	\$?=0
\$comando1 comando2	(OR)	\$?\neq 0

Canales de comunicación

Linux trata las entradas y salidas como flujos de datos

- **stdin** → Entrada de datos (Teclado) → **Descriptor de fichero 0**
- **stdout** → Salida de datos (Monitor) → **Descriptor de fichero 1**
- **stderr** → Mensajes de error (Monitor) → **Descriptor de fichero 2**

Operadores de redirección

1>	Redirecciona la salida a un nuevo fichero
1>>	Redirecciona la salida a un fichero
2>	Redirecciona los mensajes de error a un nuevo fichero
2>>	Redirecciona los mensajes de error a un fichero
&>	Redirecciona la salida y los mensajes de error a un nuevo fichero
&>>	Redirecciona la salida y los mensajes de error a un fichero
 	Redirige la salida del comando con la entrada de otro

Localiza ejecutables del comando

\$whereis [comando]

Muestra un resumen de la función de un comando

\$whatis [comando]

Variables de entorno

Datos en memoria identificados con un nombre

- Sistema
- Locales

Se identifican con **\$NOMBRE**

- Podemos definir una cadena de caracteres se rodea de comillas dobles (“”)
- Para intepretar literalmente debe ir rodeado de comillas simples (")
- Podemos definir un comando, debe ir rodeado de acentos graves (`)

Asignación

\$VAR=valor	asigna variable local
\$export VAR	exporta al sistema
\$export VAR=valor	asigna y exporta
\$echo \$VAR	visualiza variable
\$unset VAR	borra variable
\$export	lista variables exportadas
\$set / declare / typeset	lista todas las variables

Lee y ejecuta el fichero exportando las variables del fichero al entorno

\$. [fichero]
\$source [fichero]

Variables de entorno comunes

USER → Nombre de usuario actual

HOSTNAME → Nombre de equipo

SHELL → Ruta al shell

TERM → Tipo de terminal

BASH → Ruta al ejecutable de bash

PWD → Directorio de trabajo actual

OLDPWD → Directorio de trabajo anterior

HOME → Directorio home del usuario actual

PATH → Ruta de sesión (contiene todos los comandos ejecutables)

PS1 → Prompt por defecto

LANG → Idioma (ajuste local)

EDITOR → Editor por defecto

HISTFILE → Ruta al fichero de historial del usuario

HISTFILESIZE → Número máximo de entradas guardadas en HISTFILE

Añadir un directorio a la variable PATH

\$PATH=\$PATH:[directorio]

Metacaracteres

Caracteres especiales del shell

\$	Dólar	Hace referencia a una variable
\	Barra invertida	Deshabilita el siguiente caracter especial
“ ”	Comillas dobles	Interpreta literalmente, expande variables
' '	Comillas simples	Interpreta literalmente, evita cualquier expansión
` `	Comillas invertidas	Expande un comando
.	Punto	Indica el directorio actual
..	Dos puntos	Indica el directorio padre
;	Punto y coma	Separa comandos
&	Ampersand	Ejecuta un comando en segundo plano

Expresiones regulares

Caracteres para expresar patrones de texto.

[ab]	Expresiones entre corchetes	Coinciden con cualquiera de los contenidos dentro
[a-b]	Expresiones de rango	Listan los puntos de inicio y final separados por un guión
*	Asterisco	Denota una o más coincidencias
?	Interrogación	Denota una o ninguna coincidencia

Combinaciones de teclado útiles en el shell

Desplazamiento

CTRL+A	Desplaza al inicio
CTRL+E	Desplaza al final
CTRL+(← , →)	Desplaza entre palabras

Borrado de texto

CTRL+U	Borra hasta el inicio
CTRL+K	Borra hasta el final
CTRL+W	Borra una palabra hacia atrás
CTRL+Y	Pega texto borrado

Otros

CTRL+L	Limpia la pantalla
CTRL+C	Finalizar programa SIGTERM
CTRL+Z	Detener el comando actual (a segundo plano) SIGINT
CTRL+X+E	Ejecutar el editor por defecto

Definir alias

\$alias [nombre]='comando'	Define alias
\$alias	Muestra lista de alias
\$unalias [nombre]	Borra el alias
\$unalias -a	Borra todos los alias

Ejecutar un comando sin hacer uso de alias

\$\ [comando]

Historial de comandos

\$history	Lista el historial
#!n	Ejecuta el comando número n
!!	Ejecuta el comando anterior
\$history -c	Borra el historial

Scripts de configuración de bash

Se ejecutan en orden al iniciar sesión en un shell bash

Globales

1. **/etc/profile**
2. **/etc/profile.d/***

Usuario

1. **~/.bash_profile**
2. **~/.bashrc**

Adicionales de usuario

- **~/.bash_login** → Configuración de comandos ejecutados durante el acceso al bash
- **~/.bash_logout** → Configuración de comandos antes de cerrar bash
- **~/.bash_history** → Contiene el historial de comandos

Bash Scripts

Programas escritos en un **lenguaje interpretado** integrado en el shell.
Son ficheros de texto que presentan una serie de órdenes y estructuras organizadas.
Se ejecutan de forma secuencial orden tras orden.

Características básicas

- Un shell script contiene comandos del shell y estructuras de control.
- El código **#!** al inicio del fichero indica que se trata de un script y utiliza el resto de la línea como ruta al programa que va a interpretar el script.
- El carácter **#** al inicio de una línea se utiliza para insertar comentarios.
- El carácter **&** al final de la línea indica al shell que pase a la siguiente línea sin esperar a que finalice el comando.
- El script utiliza las rutas completas a cada comando para mejorar la seguridad.

Ejecución

Ejecutar un programa hace que el shell realice una llamada **fork** al sistema que crea un nuevo proceso duplicado del shell. El subshell trata de ejecutar el comando realizando una llamada **exec**.

Añadir permisos de ejecución a un script

```
#chmod +x [fichero]
```

Nueva subshell (fork)

La ruta existe en PATH	\$script
La ruta no existe en PATH	\$/ruta/script
Directorio actual	\$/./script
Llamar a un nuevo shell	\$sh script

Reemplaza el shell actual (exec)

Comando interno exec	\$exec script
-----------------------------	----------------------

Añadir permisos y ejecutar

```
$chmod +x [script] && ./[script]
```

TEMA 2.1 – INTRODUCCIÓN AL SISTEMA DE FICHEROS

Fichero

Colección de bytes relacionados con un nombre.

(Límite de caracteres en un nombre de fichero → 255)

El kernel Linux y las aplicaciones lo gestionan todo a través de ficheros

Tipos de ficheros

-	Fichero de datos	
d	Directorio	Estructura lógica que agrupa y relaciona ficheros.
l	Enlace simbólico	Apunta al fichero enlazado
p	Pipe	Permite la comunicación entre procesos
s	Socket	Permite la comunicación entre procesos y enlaces de red
b	Dispositivo de bloque	Corresponde a un dispositivo de hardware de bloque
c	Dispositivo de caracteres	Corresponde a un dispositivo de hardware de caracteres

FHS (*Filesystem Hierarchy Standard*)

Se emplea para definir la estructura de ficheros en los SO tipo Unix.

/	Directorio raíz. Todos los directorios se ramifican de este
/bin	Contiene ficheros ejecutables accesibles para todos los usuarios
/boot	Contiene ficheros del kernel y relacionados con el arranque
/dev	Sistema de ficheros virtual dinámico que aloja los ficheros de los dispositivos
/etc	Contiene archivos de configuración específicos del sistema
/home	Contiene los datos de los usuarios.
/lib	Contiene librerías de código compartido.
/media	Aloja las estructuras de directorio de los dispositivos
/mnt	Directorio usado como punto de montaje
/opt	Pensado para los paquetes que no vienen con el SO
/proc	Sistema de ficheros virtual dinámico que da acceso a información del hardware
/root	Directorio home del usuario root
/run	Contiene ficheros variables de programas en ejecución
/sbin	Contiene programas que solo ejecuta el usuario root
/srv	Contiene ficheros utilizados por servicios
/sys	Sistema de ficheros virtual dinámico que da acceso a información del hardware
/tmp	Aloja ficheros temporales creados por los programas en ejecución
/usr	Aloja el grueso de los programas.
/var	Contiene ficheros variables de programas

Comandos de visualización y manipulación de ficheros y directorios

Listar contenido de un directorio

```
$ls [opciones] [directorio/s]
```

Opciones

-a	--all	Muestra todos los ficheros ocultos
-l	--long	Genera un listado largo con información detallada
-R	--recursive	Genera un listado recursivo
-r	--reverse	Invierte el orden de visualización
-S	--size	Ordena según el tamaño de fichero
-h	--human-readable	Ayuda en la lectura del tamaño de los ficheros

Crea un fichero vacío

```
$touch [fichero/s]
```

Si el fichero ya existe, se cambia la fecha de modificación por la actual.

Crear directorios

```
$mkdir [opciones] [directorio/s]
```

Opciones

-p	--parents	Crea los directorios padre necesarios
-----------	------------------	---------------------------------------

Borrar directorios vacíos

```
$rmdir [opciones] [directorio/s]
```

Opciones

-p	--parents	Borra un árbol de directorio vacío completamente
-----------	------------------	--

Crear enlaces simbólicos

```
$ln -s [fichero enlazado] [nombre de enlace]
```

Copiar ficheros

```
$cp [opciones] [fichero origen] [fichero destino]
```

Opciones

-v	--verbose	Modo detallado
-i	--interactive	Pregunta antes de sobrescribir
-p	--preserve	Preserva la propiedad y permisos del original
-r	--recursive	Copia directorios completos
-f	--force	Fuerza a sobrescribir sin preguntar

Mover o renombrar ficheros

```
$mv [opciones] [fichero origen] [fichero destino]
```

Opciones

-v	--verbose	Modo detallado
-i	--interactive	Pregunta antes de sobrescribir
-f	--force	Fuerza a mover

Borrar ficheros

```
$rm [opciones] fichero/s
```

Opciones

-v	--verbose	Modo detallado
-i	--interactive	Pregunta antes de borrar
-r	--recursive	Borra directorios completos
-f	--force	Fuerza el borrado

Localizar ficheros

```
$locate [nombre de fichero]
```

Locate trabaja con una base de datos que “cron” actualiza a menudo

Archivar ficheros y directorios

```
$tar [opciones] [fichero/s]
```

Opciones

-c	--create	Crea un archivo
-f	--file [fichero]	Especifica el fichero como archivo
-x	--extract	Extrae ficheros de un archivo
-v	--verbose	Proporciona información adicional
-t	--list	Lista el contenido de un archivo
-r	--append	Añade ficheros a un archivo
-u	--update	Añade ficheros sobrescribiendo si existen
-p	--same-permissions	Preserva la información de permisos
-1	--one-file-system	Copia o restaura un sistema de ficheros completo
-z	--gzip	Procesa un archivo mediante gzip
-j	--bzip2	Procesa un archivo mediante bzip2

Compresión GZIP/BZIP2

```
$gzip/bzip2 [fichero]
```

Opciones

-1/9		Grado de compresión
-d	--decompress	Descomprimir (gunzip)
-c	--stdout	Visualiza un fichero comprimido (gzcat)
-l	--list	Muestra información del archivo comprimido
-r	--recursive	Comprime recursivamente por directorios
-t	--test	Comprueba la integridad de un archivo comprimido
-k	--keep	Mantiene el fichero de entrada
-v	--verbose	Muestra información detallada del proceso

Algunas opciones pueden variar en función al comando que se utilice.

Buscar ficheros y directorios

\$find [ruta/s] [expresión/es] [acción]
--

Expresiones

-name [nombre]	Buscar por nombre de fichero
-iname [nombre]	Busca por nombre e ignora entre mayúsculas
-perm [modo]	Buscar por modo de permiso (simbólico u octal)
-type [tipo]	Busca por tipo de fichero (d, f, l)
-uid [UID]	Buscar por ID de usuario
-user [nombre]	Buscar por nombre de usuario
-gid [GID]	Buscar por ID de grupo
-group [nombre]	Buscar por nombre de grupo
-maxdepth [n]	Limita el número de subdirectorios en los que buscar

Acciones

-print	Opción por defecto. Muestra el resultado en pantalla
-exec [comando] {} \;	Ejecuta un comando con cada coincidencia

TEMA 2.3 – USUARIOS Y GRUPOS

Usuarios

Se utilizan para identificar a usuarios concretos.

Cada fichero de grupo está asociado a un usuario, propietario del mismo.

Los nombres de usuario deben comenzar por una letra y pueden constar de hasta 32 caracteres.

Grupos

Linux utiliza los grupos como medios para organizar los usuarios.

Cada fichero de Linux está asociado con un grupo específico.

Identificadores del sistema

- **UID** : ID de usuario
- **GID** : ID de grupo

Numeración IDs

0 → UID / GID de Root

1 - 999 → Cuentas y grupos asociadas con utilidades y funciones del sistema

+1000 → Cuentas de usuario y grupos normales

Mostrar información de UID/GID

```
$id
```

Encontrar ficheros y directorios de usuarios y grupos específicos

```
$find [directorio] -uid [UID]
```

```
$find [directorio] -gid [GID]
```

Ficheros de configuración de usuarios

- **[/etc/passwd](#)**
 - Define las cuentas y el grupo primario de cada usuario
- **[/etc/shadow](#)**
 - Define las contraseñas de los usuarios
- **[/etc/group](#)**
 - Define los grupos y sus miembros

Configuración de las cuentas → [/etc/passwd](#)

Campos

- Nombre de usuario
- Contraseña
 - (*) : usuario bloqueado
 - (x) : contraseña encriptada
- UID
- GID primario
- Comentario
- Directorio home
- Shell por defecto

Este fichero suele tener permisos 644 para ser legible por usuarios normales

Configuración de contraseñas → [/etc/shadow](#)

Campos

- Nombre de usuario
- Contraseña encriptada
 - (*) : ausencia de contraseña
 - (!) : contraseña bloqueada
- Último cambio de contraseña. (Número de días desde 1/1/1970)
- Días hasta permitir un cambio
- Días hasta requerir un cambio
- Días de aviso previos a la expiración de contraseña
- Días entre la expiración y la desactivación
- Fecha de expiración
- Indicados especial (vacío)

Este fichero se suele almacenar con permisos restrictivos (640) para evitar que los usuarios lean el fichero y obtengan la lista de contraseñas

Configuración de grupos → [/etc/group](#)

Campos

- Nombre del grupo
- Contraseña
- GID
- Lista de usuarios

Añadir usuarios

```
#useradd [opciones] [usuario]
```

Opciones

-D	Muestra la configuración predeterminada
-u [UID]	Crea una cuenta con el UID especificado
-g [grupo]	Especifica el grupo por defecto
-d [/ruta]	Especifica la ruta del directorio home
-k [/ruta]	Especifica la plantilla del directorio home (por defecto: /etc/skel)
-s [/ruta]	Especifica la ruta al shell por defecto
-c [comentario]	Añade el campo comentario
-M	NO crea el directorio home
-n	Desactiva la creación del grupo de usuario
-r	Especifica la creación de una cuenta de sistema
-e [YYYY-MM-DD]	Especifica la fecha de expiración de la cuenta

Es posible modificar la configuración predeterminada en **/etc/default/useradd**

Existe otro comando llamado **adduser** que nos presentará una salida interactiva.

Definir una contraseña

```
#passwd [usuario]
```

Modificar una cuenta existente

```
#usermod [opciones] [usuario]
```

Opciones

-d [/ruta]	Cambia el directorio home de usuario
-dm [/ruta]	Cambia el directorio home y mueve los ficheros al nuevo sitio
-l [usuario]	Cambia el nombre de acceso por el especificado
-aG	Añade un usuario a un grupo sin eliminar la pertenencia a otros
-L	Bloquea la contraseña
-U	Desbloquea la contraseña
-s [shell]	Modifica el shell por defecto

Eliminar cuentas

```
#userdel [opciones] [usuario]
```

Opciones

-f	Elimina todos los ficheros del usuario
-----------	--

Mostrar grupos de un usuario

```
#groups [usuario]
```

Añadir grupos

```
#groupadd [opciones] [grupo]
```

Opciones

-g [GID]	Especifica la GID del grupo
-r	Crea una GID de sistema

Eliminar grupos

```
#groupdel [grupo]
```

Modificar los parámetros del grupo

```
#groupmod [opciones] [grupo]
```

Opciones

-g [GID]	Especifica una nueva GID
-n [grupo]	Especifica un nuevo nombre para el grupo

Su

Se utiliza para cambiar la identidad aparente del usuario

```
$su [opciones][usuario]
```

Opciones

-	Cambia al entorno de root (login shell)
----------	---

Es necesario conocer el password del usuario al que queramos cambiar.

Sudo

Permite delegar autorizaciones a grupos o usuarios para ejecutar comandos como root.

Otorga privilegios temporales para un unico comando.

Se basa en la lista de control.

Se otorgan 5 minutos antes de volver a pedir la contraseña de usuario.

```
$sudo [opciones][comando]
```

Opciones

-v	Valida otros 5 minutos
-l	Muestra una lista de comandos permitidos por el usuario actual
-e [fichero]	Invoca al editor por defecto para editar el fichero
-i	Ejecuta un terminal root

Editar fichero de configuración de sudo (/etc/sudoers)

```
#visudo [opciones]
```

Opciones

-c	Comprueba que el fichero sudoers esta bien configurado
-f [fichero]	Edita otro fichero con la sintaxis de sudo

TEMA 2.4 – PROPIEDAD Y PERMISOS DE UN FICHERO

Cada fichero tiene un propietario individual y está asociado a un grupo.

Cadena de permisos

1. Propietario (**u**)
2. Grupo (**g**)
3. Permisos globales (**o**)

Tipos de permisos

- (**r**)ead → lectura
- (**w**)rite → escritura
- e(**x**)ecution → ejecución
- (-) → ausencia de permisos

Linux codifica esta información en formato binario

Los permisos se suelen expresar en formato octal (base 8)

- **1** → ejecución
- **2** → escritura
- **4** → lectura

rw-rw-rwx : 777	rw-r-xr-x : 755	rw-r-x--- : 750
rw-x----- : 700	rw-rw-rw- : 666	rw-rw---- : 660
rw-r----- : 640	rw----- : 600	r----- : 400

El bit de ejecución de directorio significa que se puede acceder al mismo.

Root puede leer o escribir en cualquier fichero independientemente de los permisos del fichero.

Cambiar máscara de permisos

La máscara del usuario define un valor octal que representa los bits a eliminar de los permisos cuando se crea un nuevo fichero

\$umask [opciones][máscara nueva]
--

Opciones

Sin opciones	Muestra la máscara de permisos del usuario (Sin argumentos)
-S	Muestra la máscara en notación simbólica

Cambiar los permisos de un fichero

<code>\$chmod [opciones] [modo simbólico] [fichero/s]</code>
--

<code>\$chmod [opciones] [modo octal] [fichero/s]</code>
--

- Modo simbólico → [**u** | **g** | **o** | **a**] [+ | - | =] [**r** | **w** | **x** | **s** | **t**]
- Modo octal → [**xxx**] → [Propietario, Grupo, Global]

Opciones

-R	--recursive	Implementa el cambio en todo un árbol de directorios
-----------	--------------------	--

Solo el propietario del fichero y root pueden ajustar los permisos del fichero

Cambiar el propietario de un fichero

<code>#chown [opciones] [nuevo propietario][fichero/s]</code>

Opciones

-R	--recursive	implementa el cambio en todo un árbol de directorios
-----------	--------------------	--

Cambiar el grupo de un fichero

<code>#chgrp [opciones] [nuevo grupo] [fichero/s]</code>
--

Opciones

-R	--recursive	implementa el cambio en todo un árbol de directorios
-----------	--------------------	--

TEMA 3.1 - ACCESO A INFORMACIÓN DEL HARDWARE

Parámetros de CPU

```
$cat /proc/cpuinfo
```

Parámetros de memoria

```
$cat /proc/meminfo
```

Comprobar información de la batería (laptops)

```
#acpi -V
```

Mostrar la información sobre los buses y dispositivos PCI

```
#lspci [opciones]
```

Opciones

-v / -vv	Incrementa el nivel de detalle de la salida
-----------------	---

Mostrar información sobre los dispositivos USB

```
#lsusb [opciones]
```

Opciones

-v	Muestra información ampliada
-t	Vista en árbol

Bases de datos de asociación de ID de fabricante y vendedores

- PCI → **/usr/share/misc/pci.ids**
- USB → **/var/lib/usbutils/usb.ids.gz**

Udev

Sistema de ficheros virtual montado en **/dev** que crea ficheros de dispositivos dinámicos conforme se cargan y descargan.

Estos ficheros proporcionan al kernel acceso al hardware.

El subsistema udev crea entradas para la mayoría de dispositivos de hardware al iniciar el sistema

Los ficheros en **/dev** tienen propietarios, grupos y permisos

TEMA 3.2 - ALMACENAMIENTO LOCAL

Esquema de partición x86

El tamaño de sector por defecto es de 512 bytes

Puede manejar discos de hasta 2 TB

Aumentando el tamaño de sector a 4096 bytes el límite se eleva a 16 TB

Tipos de particiones:

- Particiones primarias (máximo 4)
- Partición extendida
 - Contiene particiones lógicas

Linux puede arrancar desde cualquier tipo de partición.

Los ficheros asociados a los dispositivos se encuentran en **/dev**

La nomenclatura de los discos duros IDE es **/dev/hd[xy]**

La nomenclatura de los discos duros SATA, SCSI y externos USB y FW es **/dev/sd[xy]**

- **x** representa el número de disco
- **y** representa el número de partición

Tabla de partición

Almacena información sobre los formatos de las particiones y la geometría del disco.

Puntos de montaje

Directorios que se utilizan como medio para montar y acceder al sistema de ficheros de la partición.

Fichero /etc/fstab

Se especifican los sistemas de ficheros que se montaran automáticamente durante el arranque

Lista los dispositivos de disco disponibles

```
$lsblk
```

Lista e informa sobre los dispositivos montados

```
$findmnt
```

Particionar discos

```
#fdisk [opciones] [/dev/partición]
```

Opciones

-b [tamaño]	especifica el tamaño de los sectores (512, 1024, 2048, 4096)
-l	muestra la tabla de particiones

El resultado es un **prompt** de fdisk

Comandos

- m** → Muestra la ayuda
- p** → Muestra la tabla de partición actual (fdisk -l /dev/disco)
- n** → Crear una partición
- d** → Eliminar una partición
- t** → Cambiar el tipo de partición (mediante el código de partición)
- l** → Lista los códigos de partición
- a** → Marca una partición como arrancable (necesario para algunos SO)
- q** → Salir sin guardar cambios
- w** → Salir guardando los cambios

Proporcionar acceso al sistema de ficheros de una partición de disco (Montaje)

```
#mount [opciones] [/dev/partición] [/punto de montaje]
```

Opciones

-a	Monta todos los sistemas de ficheros listados en el fichero /etc/fstab
-v	Muestra una salida detallada
-r	Monta en modo sólo lectura
-w	Monta en modo lectura/escritura (-o rw)
-t [fstype]	Especifica el sistema de ficheros. Si se omite, Linux intentará detectarlo
-o [opciones]	Especifica opciones adicionales

Opciones adicionales

loop	Permite montar un fichero como si se tratase de una partición de disco
auto / noauto	Monta o no monta el sistema de ficheros en tiempo de arranque
user / nouser	Permite o impide a los usuarios normales montar el sistema de ficheros
ro	Especifica el montado en modo sólo lectura
rw	Especifica el montado en modo lectura/escritura
exec / noexec	Permite o no ejecutar binarios
defaults	Se utilizan las opciones por defecto : auto nouser ro exec
uid=[n]	Define el propietario de todos los ficheros por UID
gid=[n]	Define el grupo de todos los ficheros por GID

Para montar un sistema de ficheros de un tipo concreto, el kernel debe soportarlo.

Desmontar sistema de ficheros

```
#umount [opciones] [/dev/partición o /punto de montaje]
```

Opciones

-a	Desmonta todas las particiones listadas en el fichero /etc/fstab
-f	Fuerza el desmontado
-r	Pasa a modo sólo lectura
-t [fstype]	Especifica el sistema de ficheros

Mostrar UUID (Identificador único) de partición

```
#blkid -p [/dev/partición]
```

Swap

Espacio de intercambio, partición que Linux trata como una extensión de memoria.

1.Preparar la partición

```
#mkswap [opciones] [/dev/partición]
```

Opciones

-L [etiqueta]	Establece etiqueta de la partición
----------------------	------------------------------------

2.Activar swap

```
#swapon [opciones] [/dev/partición]
```

Opciones

-a	Activa todos los dispositivos swap listados en /etc/fstab
-s	Muestra información resumida de la utilización del espacio
-L [etiqueta]	Monta la swap por su etiqueta

Desactivar swap

```
#swapoff [/ev/partición]
```

Crear fichero de intercambio (Ejemplo de 512 Megas)

```
#dd if=/dev/zero of=/ruta/fichero bs=1024 count=512000
```

```
#mkswap /ruta/fichero
```

```
#swapon /ruta/fichero
```

Mostrar información de la swap

```
$cat /proc/swaps
```

Encriptar dispositivos/particiones (LUKS)

```
#cryptsetup [acción]
```

Acciones

LuksFormat [/dev/partición]	Encripta una partición
LuksOpen [/dev/partición]	Abre una partición encriptada
LuksClose [/dev/partición]	Cierra una partición encriptada

TEMA 3.3 - OPERACIONES EN SISTEMAS DE FICHEROS

Sistemas de ficheros

Estructuras de datos de bajo nivel que permiten el acceso por nombre a los archivos y ficheros almacenados.

Tipos

- **Ext2fs**
 - Nativo de Linux
 - Buen resultado para particiones pequeñas
- **Ext3fs**
 - Consiste en ext2 con un respaldo de transacciones añadido
- **Ext4fs**
 - Ext3 incorporando la capacidad de trabajar con discos y ficheros muy grandes
- **Reiserfs**
 - Sistema de ficheros con respaldo de transacciones
 - Buen resultado gestionando sistemas con un gran número de ficheros pequeños
- **JFS**
 - Sistema de ficheros con respaldo de transacciones (AIX y OS/2)
 - Buen resultado para discos y ficheros grandes
- **FAT**
 - Sistema de ficheros de DOS y Windows 9x
- **NTFS**
 - Sistema de ficheros de Windows NT
- **HFS+**
 - Sistema utilizado por Apple (Mac OS)
- **ISO-9660**
 - Sistema estándar para CD-ROM
- **UDF**
 - Se utiliza para DVD-ROM y discos ópticos regrabables

Respaldo de transacciones

Registro de datos que describe las operaciones efectuadas en el sistema de ficheros (transacciones).

En caso de corrupción de los datos, Linux puede examinar el registro y revisar las estructuras de datos descritas en él. El sistema puede revertir los cambios devolviendo el disco a un estado consistente.

I-nodes

Punteros utilizados para localizar los datos almacenados. Se almacenan dentro de un superbloque.

El **superbloque** es un registro que contiene información sobre el sistema de archivos.

El superbloque se almacena en varias copias para proporcionar tolerancia a fallos.

Región

Grupo de bloques contiguos que pertenecen a un fichero

Fragmentación

- Interna
 - Espacios vacíos de los ficheros que no llenan los bloques
- Externa
 - Bloques de un fichero que están repartidos en lugar de contiguos

Crear sistema de ficheros

```
#mkfs [opciones] [/dispositivo]
```

Opciones

-t [tipo]	Especifica el sistema de ficheros
-c	Realiza una comprobación de bloques defectuosos (mediante badblocks)
-m [n]	Define el porcentaje de espacio reservado para el respaldo de transacciones

Comprobar uso de espacio en disco

```
#df [opciones] [/dispositivo]
```

Opciones

sin opciones	Muestra un resumen del espacio utilizado en bloques
-h	Cambia la escala a bytes
-i	Obtiene información sobre los i-nodes
-T	Muestra el tipo de sistema de ficheros

Comprobar uso de espacio por directorios

```
#du [opciones] [directorio/s]
```

Opciones

sin opciones	Realiza una búsqueda recursiva desde el directorio actual
-a	Muestra todos los ficheros de los directorios analizados
-c	Añade una suma total del espacio ocupado
-h	Cambia la escala a Bytes
-s	Resume la salida
-d [n]	Limita el informe a n niveles de subdirectorios

Obtener información del sistema de ficheros

```
#dumpe2fs -h [/dispositivo]
```

Revisar el sistema de ficheros en busca de errores

```
#fsck [opciones] [/dispositivo]
```

Opciones

-f	Fuerza la revisión
-A	Revisar todos los ficheros marcados en /etc/fstab
-C	Indicar el progreso
-V	Muestra una salida detallada
-c	Busca bloques de disco dañados
-p	Resuelve errores

Ejecutarlo en sistemas de archivos desmontados o solo lectura.

Forzar la revisión en el inicio de sistema

```
#touch /forcefsk
```

TEMA 4.1 – PROCESOS

Proceso

Forma en que un sistema operativo trata una secuencia de instrucciones durante su ejecución. La CPU administra múltiples procesos asignando slots temporales para cada uno.

La shell inicia un proceso nuevo por cada comando, permaneciendo inactiva hasta que acabe. La estructura de los procesos es jerárquica. Las jerarquias forman familias de procesos

Cuando se crea un nuevo proceso se ejecuta una llamada al sistema **fork**

Mientras el proceso hijo se ejecuta, el proceso padre permanece inactivo por la llamada del sistema **sleep**

Estados de un proceso

- **Runnable** → En ejecución
- **Sleeping** → En ejecución sin actividad
- **sTopped** → Detenido
- **Zombie** → Terminado incorrectamente
- **Xdead** → Terminado
- **Device** → Procesos asociados a acciones de E/S

Mostrar lista y estado de los procesos

```
$ps [opciones]
```

Soporta un gran número de opciones

Varia entre formato System V (precedidos de guión) y formato BSD (sin uso de guión)

sin argumentos	Muestra los procesos del terminal actual
-e	Muestra todos los procesos
-f	Muestra información adicional
-H	Muestra la jerarquía de procesos

Parámetros

- **UID** → ID de usuario
- **PID** → ID de proceso
- **PPID** → ID del proceso padre
- **TTY** → Terminal asociado al proceso
- **STIME** → Hora de inicio
- **TIME** → Tiempo de CPU
- **CMD** → Nombre de ejecutable

Mostrar lista de procesos en forma de árbol jerárquico

```
$pstree -p
```

Ejecuta un programa en segundo plano

\$ [comando] &

Control de los procesos en segundo plano

\$jobs	Comprobar procesos y lista de tareas asociados a la sesión
\$fg [número tarea]	Reanuda la tarea en primer plano
\$bg [número tarea]	Reanuda la tarea en segundo plano

Control+Z → Detiene un proceso en ejecución del terminal

Enviar señales a los procesos

#kill [opciones] [señal] [PID]

Opciones

-l	Muestra lista de señales
-s	Especifica la señal por su nombre

Señal por defecto 15 (SIGTERM)

Señales

2	INT	Interrupción de programa	(CTRL+Z)
15	TERM	Finalizar programa	(CTRL+C)

Envía señales a todos los procesos que coincidan con el nombre especificado

\$killall [opciones] [nombre]

Opciones

-s [señal]	Envía una señal por su nombre
-i	Pide confirmación antes de enviar la señal

TOP

Herramienta en modo texto que muestra lista de procesos y ordena las entradas por el uso de CPU o memoria, actualizando los datos cada pocos segundos.

```
#top [opciones]
```

Opciones

-d [retardo]	Ajusta el tiempo entre actualizaciones (s)
-p [pid,...]	Monitoriza el estado de proceso/s específicos
-u [usuario]	Especifica procesos de usuario

Comandos interactivos

- **f** → Agregar o quitar columnas
- **k** → Destruir proceso por PID
- **s** → Cambia la velocidad de actualización (3s)
- **P** → Ordena las entradas por uso de CPU
- **M** → Ordena las entradas por uso de memoria
- **u** → Ordena las entradas por usuario
- **q** → Sale de top

Comprobar el uso de memoria

```
#free [opciones]
```

Opciones

-h	Muestra en un formato más adaptado
-s [n]	Genera un refresco cada n segundos

Por defecto lo muestra en kilobytes

TEMA 4.2 – INICIO DEL SISTEMA

Proceso de arranque

1. Se enciende el sistema. La CPU ejecuta la **BIOS**.
2. El código de la BIOS realiza la comprobación y configuración del hardware (**POST**). Busca el cargador de arranque primario en el **MBR** del disco.
3. El control pasa de la BIOS al **cargador de arranque**.
Algunos cargadores de arranque van directamente a un cargador secundario de una partición activa del disco duro.
4. El cargador de arranque encuentra un kernel, lo carga en memoria y lo ejecuta.
5. El kernel de Linux toma el control.
Detecta el hardware
Inicializa dispositivos.
Monta la partición raíz.
Carga y ejecuta el proceso inicial de sistema **/sbin/init** (PID=1)
6. Dependiendo del sistema de inicio init determina que otros programas ejecutar.
(scripts de inicio , gettys, login, interfaz gráfica ...)

Extraer información del arranque

```
$dmesg
```

Fichero de almacenamiento:

- **/var/log/dmesg**

Información contenida:

- Identifica el kernel
 - Describe detalles técnicos sobre el uso de memoria del kernel
- Clasifica la CPU y sus parámetros
- Identifica los buses de la máquina
- Identifica subsistemas de hardware
 - Controladores de disco SCSI
 - Controlador de arquitectura de sonido
 - Inicialización de protocolos de red
- Identifica controladores de hardware específicos
 - Dispositivos HID
 - Dispositivos PATA/SATA
 - Dispositivos USB
 - Interfaces de red
- Identifica sistemas de ficheros para montar
- Añadidos y errores de memoria

Otras fuentes de información del arranque:

- **/var/log/messages**
- **/var/log/syslog**
- **/var/log/boot**

Modos de ejecución

Cada modo tiene asignado un conjunto de servicios que deberían estar activos.

- **0** : cierra y apaga el sistema
- **1, S** : modo monousuario
- **2, 3, 4, 5** : varían según la distribución
- **6** : reinicia el sistema

Determinar el modo de ejecución anterior y el actual

```
#runlevel
```

Cambiar al modo de ejecución “n”

```
#init [n]
```

Gestión de SysV Init

Sistema de inicio clásico de UNIX

Fichero /etc/inittab

Formato de las líneas → 4 campos delimitados por dos puntos

```
id:runlevels:action:process
```

1. **id** : Secuencia de identificación de función
2. **runlevels** : Lista de modos de ejecución para los que se aplica la entrada
3. **action** : Le indica a init como tratar el proceso
4. **process** : Proceso a ejecutar para la entrada, incluye opciones y argumentos necesarios

Gestionar accesos

La mayoría de sistemas estándar definen seis terminales en modo texto (**tty**).

Se definen en **/etc/inittab** indicando que ejecute un programa **getty** por cada terminal virtual.

Los programas getty gestionan el programa **login** que acepta accesos de usuarios.

Scripts de inicio principales

Ubicaciones usuales:

- **/etc/init.d**
- **/etc/rc.d**

Scripts asociados a un modo de ejecución

Enlaces simbólicos en los directorios **/etc/rc?.d** a los scripts principales

Donde (n) es el numero del modo de ejecución

Inician o detienen los servicios dependiendo del parámetro que se les pase

- **Iniciar** : scripts que empiezan por la letra **S**
- **Detener** : scripts que empiezan por la letra **K**

El programa **rc** ejecuta los scripts en orden numérico definido en el nombre de enlace.

Las distribuciones Linux incluyen un script de inicio ubicado en el fichero **/etc/rc.local** que se ejecuta después de los otros scripts SysV

Iniciar y detener scripts manualmente

```
#!/etc/init.d/script [start|stop|restart|reset|status]
```

Ejecuta o detiene un script de inicio

```
#service [script] [start|stop|restart|status]
```

Gestión de Upstart

Sistema de inicio moderno adoptado por algunas distribuciones GNU/Linux.

Se deshace de /etc/inittab proporcionando en su lugar un grupo integrado de scripts de inicio.

La ubicación de los scripts sigue estando en **/etc/init.d**.

Ubicación de ficheros de configuración de servicios:

- **/etc/init**
- **/etc/event.d**

Para cambiar los modos de ejecución en que se ejecuta un servicio hay que editar las líneas de su archivo de configuración:

`start on [n]`

`stop on [n]`

Upstart posee un modo de compatibilidad con SysV para ejecutar scripts que se encuentran enlazados en las ubicaciones habituales.

Herramienta de control de servicios de Upstart

```
#initctl [comando] [servicio]
```

Comandos

List	Lista los módulos y su estado
Reload	Recarga los ficheros de configuración
Start	Inicia un servicio
Stop	Detiene un servicio

Apagado del sistema

```
#shutdown [opciones] [acciones]
```

Opciones

-r	Reinicia el sistema
-h	Apaga el sistema
-c	Cancela la operación
-F	Fuerza fsck durante el reinicio
-t [s]	Retraso en segundos

Acciones

0	Al instante
hh:mm	Formato de tiempo de 24 horas
+m	Espera [m] minutos

Es posible añadir un mensaje entre comillas al final del comando para avisar a los usuarios logueados.

TEMA 5.1 – SOFTWARE EN LINUX

La mayoría del software de Linux está disponible como:

- **Código fuente**
- **Código binario**

El software se convierte de código fuente a binario a través de un **compilador**.

El software de código fuente puede ser compilado para ejecutarse en cualquier computadora que contenga el compilador y las herramientas necesarias.

El código binario es personalizado por un SO y una familia de hardware concreta durante la compilación.

Los gestores de paquetes generalmente trabajan con software precompilado en una arquitectura determinada.

Compilar código fuente

Requerimientos:

- Herramientas de desarrollo
 - **GCC** (*CNU Compiler Collection*)
- Librerías de desarrollo
 - **Ficheros de cabecera**
- Herramientas genéricas de apoyo
 - **Make ...** (Generalmente distribuidas en el paquete **build-essential**)

Proceso de instalación del código fuente:

1) Obtener el código fuente

- Sitios Web: (por ejemplo)
 - www.sourceforge.net
 - www.code.google.com
- A través del gestor de paquetes de la distribución:
 - APT
 - YUM

Computar md5

```
#md5sum [fichero md5]
```

Importar claves y verificar firmas

```
#gpg --import [clave pública]
```

```
#gpg --verify [fichero de firma]
```

2) Desempaquetar el código fuente

```
$tar -zxvf [archivo.tgz]
```

```
$tar -jxvf [archivo.tbz2]
```

Cuando son extraídos crean un subdirectorio con el mismo nombre del tarball.

El directorio **/usr/src/** almacena código fuente de programas que se compilan de forma local.

3) Leer la documentación

README

- Documentación general

INSTALL

- Instrucciones de instalación

4)Configurar la compilación

Utilizar un script llamado **configure** ubicado en el directorio principal del código fuente

```
$./configure [parámetros]
```

Parámetros

--help	Muestra un resumen de opciones
--quiet	Muestra menos mensajes durante la compilación
--prefix=[/directorio]	Establece el directorio de instalación de arquitectura independiente
--enable-[función]	Habilita una función del programa
--disable-[función]	Deshabilita una función del programa
--disable-debug	Inhabilita la salida de depuración

El script escaneará el sistema y ajustará el archivo **Makefile** que controla el proceso de compilación

Makefile es un archivo de configuración para el programa make

El software que se compila localmente se instala normalmente en **/usr/local/**

5)Compilación del código fuente

```
$make
```

Organiza la compilación y el enlazado de librerías.

El proceso genera líneas con diferentes secuencias.

Si la salida incluye la palabra error indica un problema que evita que GCC compile el software.

Arreglar el primero de los errores puede eliminar todos los posteriores.

6)Instalación del software

```
#make install [opciones]
```

Realiza la instalación y ajustará automáticamente la propiedad y los permisos.

El programa binario se copiará en **/usr/local/bin**

Las páginas man y los archivos de soporte se copiarán en las ubicaciones adecuadas.

Eliminar ficheros generados durante la compilación

```
#make clean
```

Ejecuta el comando en el directorio del código fuente.

Desinstalar el software

```
#make uninstall
```

Ejecuta el comando en el directorio del código fuente.

→ Instalación manual

Algunos programas no proporcionan el parámetro install para make.

Se debe copiar el archivo de programa de forma manual. Consulta la documentación.

- Código fuente → **/usr/local/src**
- Ejecutables → **/usr/local/bin**
- Librerías → **/usr/local/lib**
- Ficheros de configuración → **/usr/local/etc**

TEMA 5.2 – LIBRERÍAS

Librerías

Fragmentos de código utilizados por los programas con más frecuencia.

- Estáticas (.a)
- Dinámicas/Compartidas (.so)

El ejecutable incluye referencias a los ficheros de librería compartida, que se pueden cargar junto con el fichero del programa principal.

Los programas deben poder localizar las librerías compartidas ajustando los ficheros de configuración y las variables de entorno.

Directorios de librerías habituales

- **/lib** - **/lib64**
- **/usr/lib** - **/usr/lib64**

Fichero de configuración de librerías

- **/etc/ld.so.conf**

Cada línea lista un directorio en el que se pueden encontrar ficheros de librería compartida.

Añadir directorios de ficheros de librería

```
#export LD_LIBRARY_PATH=[/ruta1:/ruta2]
```

Los directorios definidos en la variable tendrán preferencia frente a otros directorios.

A veces el fichero binario indica la ruta de la librería, si es así es necesario crear un enlace simbólico desde la ubicación que espera el programa al directorio correcto.

Los cambios en los números de versión también se resuelven creando un enlace simbólico.

Comprobar las librerías que utiliza un fichero de programa

```
$ldd [opciones] [/ruta/fichero]
```

Opciones

```
-v Muestra una salida más detallada
```

TEMA 5.3 – GESTIÓN DE SOFTWARE EN DEBIAN Y DERIVADOS

Versiones de paquetes de Debian

- Stable → Libres de errores
- Testing → En revisión
- Unstable → En desarrollo

DPKG

Herramienta para instalar, administrar y generar paquetes localmente

```
#dpkg [accion] [paquete/s]
```

Acciones generales

-i	--install	Instala paquete
-r	--remove	Elimina paquete
-P	--purge	Elimina paquete y ficheros de configuración
-I	--info	Información del fichero de un paquete desinstalado
-c	--contents	Muestra los contenidos del archivo
-l	--list	Lista todos los paquetes instalados
-L	--listfiles	Lista los ficheros asociados del paquete
-S	--search	Localiza el paquete al cual pertenece el fichero especificado

Suite APT (Advance Package Tool)

Proporciona un sistema lo más libre de errores posible, intenta seguir los principios del software de código libre.

Ubicaciones

Configuraciones generales

- /etc/apt/apt.conf.d/*

Configuración de repositorios

- /etc/apt/sources.list

Caché de paquetes

- /var/cache/apt/*

Añade un CD o DVD a la lista de fuentes disponibles

```
#apt-cdrom -m add [punto de montaje]
```

Interfaz de tipo texto para APT

```
#aptitude
```

APT-CACHE

Proporciona información sobre la base de datos de los paquetes Debian (caché binaria de APT)

```
$apt-cache [acción] [paquete/s]
```

Acciones

search	Busca un paquete
show	Muestra descripción del paquete
showpkg	Muestra información de dependencias de paquete
unmet	Muestra información sobre dependencias no satisfechas
dump	Lista todos los paquetes de la caché

APT-GET

Herramienta para descargar, desempaquetar, instalar y configurar paquetes.

Opera obteniendo información de las fuentes listadas en **/etc/apt/sources.list**

```
#apt-get [opciones] [comando] [paquete/s]
```

Comandos

update	Sincroniza información actualizada sobre paquetes disponibles
upgrade	Actualiza todos los paquetes sin eliminar dependencias ni otros paquetes
dist-upgrade	Actualiza todos los paquetes eliminando otros paquetes si es necesario
install	Instala un paquete y sus dependencias
remove	Elimina un paquete dejando los ficheros de configuración
purge	Elimina un paquete y sus ficheros de configuración
source	Obtiene el fichero de código fuente más reciente disponible del paquete
check	Revisa la base de datos y las dependencias rotas
clean	Realiza una limpieza total de la caché de paquetes
autoclean	Realiza una limpieza de versiones anteriores de paquetes de la caché
autoremove	Elimina las dependencias sin utilizar
build-dep	Instala o elimina las dependencias de compilación de un paquete fuente
moo	Podemos de supervaca

Opciones

-d	--download-only	Descarga los ficheros del paquete sin instalarlos
-m	--fix-missing	Ignora los ficheros de paquetes que no se pueden obtener
-u	--show-upgraded	Muestra la lista de paquetes a actualizar
-q	--quiet	Modo silencioso
-s	--simulate	Realiza una simulación
-y	--force-yes	Genera «yes» a cualquier pregunta de los scripts de instalación
-b	--build	Compila un paquete fuente tras obtenerlo (source)

TEMA 5.4 – GESTIÓN DE SOFTWARE EN RED HAT Y DERIVADOS

RPM

Herramienta para gestionar los paquetes de Red Hat

\$rpm [operación] [opciones] [paquete/s]

Operaciones

-i	--install	Instala un paquete
-e	--erase	Desinstala un paquete
-v	--verbose	Salida descriptiva
-h	--hash	Muestra una barra de estado
-U	--upgrade	Actualiza un paquete
-q	--query	Consulta información del paquete
-V	--verify	Verifica los ficheros de un paquete

Opciones

--root [/ruta]	Modifica el sistema para que tenga el directorio raíz en otro lugar
--force	Fuerza la instalación o actualización del paquete
--requires	Verifica las dependencias del paquete
--nodeps	Especifica que no se realice comprobación de dependencias
--noscripts	Deshabilita la ejecución de los scripts de pre y postinstalación
--test	Verifica las dependencias y conflictos sin instalar el paquete
--prefix [/ruta]	Asigna la ruta como directorio de instalación
--changelog	Muestra el historial de cambios de un paquete
--import [key]	Importa una firma digital para la verificación del paquete

Opciones de consulta

-qa	Consulta o verifica todos los paquetes instalados
-qp [rpm]	Especifica un paquete rpm para la consulta [p]
-qi	Muestra información general del paquete
-qR	Muestra los paquetes y ficheros de los que depende el especificado
-ql	Muestra ficheros que pertenecen a un paquete especificado
-qls	Muestra ficheros con información de estado
-qf [file]	Consulta a que paquete pertenece un fichero especificado
-qv	Muestra nombre, versión y si está instalado el paquete especificado
-qvl	Salida de -qv más detallada incluyendo los ficheros instalados del paquete
-qc	Lista los ficheros de configuración de un paquete instalado

YUM

Gestor de paquetes RPM

Ficheros de configuración

- /etc/yum.conf
- /etc/yum.repos.d/*

#yum [opciones] [comando] [paquete/s]

Opciones

-C	Se ejecuta completamente desde la caché local
-y	Contesta “yes” a todas las preguntas

Comandos

install	Instala paquetes y dependencias
groupinstall	Instala grupos de paquetes y sus dependencias
check-update	Comprueba si hay actualizaciones disponibles
update	Actualiza el paquete especificado o todos los paquetes (sin argumentos)
remove	Borra un paquete y sus dependencias
groupremove	Borra un grupo de paquetes y sus dependencias
list	Muestra versión y actualizaciones de un paquete
list installed	Muestra todos los paquetes instalados
list updates	Muestra los paquetes disponibles para actualizar
grouplist	Genera una lista de grupos instalados y disponibles
repolist	Genera una lista de repositorios configurados
search	Busca paquetes por palabra clave
info	Muestra información de versión y descripción de paquete
groupinfo	Muestra información sobre grupos de paquetes
provides	Muestra información de los paquetes proporcionados por un programa
deplist	Muestra las dependencias del paquete especificado
resolvedep	Lista paquetes que coincidan con la dependencia especificada
clean all	Limpia todo el directorio de caché
clean dbcache	Limpia la caché de base de datos sqlite
clean headers	Elimina ficheros de cabecera
clean metadata	Elimina ficheros de metadatos (XML)
clean packages	Elimina los paquetes almacenados en caché
makecache	Genera una nueva caché
localinstall [*rpm]	Instala rpm locales utilizando el repositorio para resolver dependencias
help	Muestra ayuda del comando especificado

TEMA 6 – KERNEL LINUX

El kernel Linux es el responsable de los accesos al hardware y de la gestión de los recursos a través de los servicios de llamadas al sistema.

Escrito en lenguaje C menos el manejo de IRQs, escrito en ensamblador

Funciones principales:

- Asignación de recursos.
- Creación de procesos, asignación de tiempos y sincronización.
- Administración el sistema de ficheros.
- Supervisión de la transmisión de datos entre la memoria y los periféricos.

Durante el arranque el kernel es cargado en RAM y es responsable de inicializar las interfaces de hardware.

→ Sitio web del kernel Linux : <http://www.kernel.org>

Nomenclatura de versiones → [A.B.xx-yy]

- A → Versión del núcleo
- B → Actualización
 - Par → Estable
 - Impar → Inestable (En desarrollo)
- xx → Revisión de funciones
- yy → Arreglos de errores

Mostrar información del kernel

```
$uname -r
```

Devuelve la información de varios ficheros del directorio **/proc/sys/kernel/**

Modificar las opciones del kernel en tiempo real

Algunos ficheros del árbol **/proc** controlan funciones del kernel

El programa **sysctl** permite ver y establecer opciones de **/proc** utilizando una clave y un valor.

```
#sysctl [opciones] [clave]
```

Opciones

-a	Muestra todas las claves y valores asociados
-w [clave=valor]	Cambiar parámetro

El fichero **/etc/sysctl.conf** contiene las opciones de inicio del kernel.

Se utiliza para realizar cambios permanentes.

Código fuente

Reside generalmente en `/usr/src/linux-[version]`

Incluye un directorio de documentación.

El enlace simbólico `/usr/src/linux` debe apuntar al directorio del kernel fuente actual

```
#ln -s /usr/src/linux-[version] /usr/src/linux
```

Algunos programas dependen de ficheros de cabecera contenidos en el árbol fuente del kernel para su compilación.

Binarios

El archivo principal se ubica en `/boot`

- **vmlinux**
 - Versión descomprimida (no puede arrancarse directamente)
- **vmlinuz**
 - Versión comprimida con funciones de arranque
- **bzImage**
 - Versión comprimida. Generalmente compilados localmente

Módulos

Extensiones del kernel que se cargan en memoria cuando se les necesita. Se ubican en `/lib/modules` (corresponden con distintas versiones del kernel).

Se pueden cargar y descargar para proporcionar acceso al hardware.

Comprovar los módulos cargados en el sistema

```
#lsmod
```

Obtener información sobre los módulos

```
#modinfo [nombre del módulo]
```

Carga o descarga un módulo y sus dependencias

```
#modprobe [opciones] [nombre del módulo]
```

Opciones

-v	--verbose	Muestra información adicional sobre las operaciones
-nv	--dry-run	Realiza una simulación
-r	--remove	Descarga un módulo y sus dependencias
-f	--force	Fuerza la carga o descarga del módulo

Proceso de compilación e instalación de un kernel Linux

1) Descargar kernel, desempaquetar en /usr/src y leer la documentación:

- README
 - Visión general del kernel e instrucciones de compilación
- 00-INDEX
 - Contiene una visión general del árbol /usr/src/linux/Documentation

2) Configurar la compilación

Se debe conocer el hardware del sistema para saber los módulos a instalar y compilar el soporte necesario para el nuevo kernel:

- Identificar el chipset.
- Comprobar dispositivos PCI.
- Comprobar dispositivos USB.
- Comprobar los módulos cargados.
- Identificar soporte para otros controladores.

Utilizar **make** para la configuración.

Acepta entre otros, los siguientes argumentos:

help	Muestra una lista de objetivos make y su descripción
clean	Elimina temporales de compilación
mrproper	Elimina temporales y ficheros de configuración
distclean	Limpieza completa
defconfig	Crea un fichero de configuración utilizando valores por defecto
allmodconfig	Crea un fichero de configuración utilizando valores modulares
config	Configura elementos del kernel en interfaz de texto
menuconfig	Configura elementos del kernel en interfaz de texto basado en menus (ncurses)
xconfig	La configuración se realiza en entorno GUI basado en Qt
gconfig	La configuración se realiza en entorno GUI basado en GTK

El comando se ejecuta make en el árbol fuente del kernel.

Abrir menu de configuración

<code>\$make menuconfig</code>

Las opciones de configuración del kernel se muestran de forma jerárquica

La marca de verificación en la casilla compilará la opción directamente en el kernel (*)

La M en la casilla compilará la opción como módulo (M)

Categorías principales

- Configuración general
- Capa de bloque
- Tipo y funciones del procesador
- Opciones de administración de energía y ACPI
- Opciones de bus
- Formatos de archivos ejecutables
- Soporte de red
- Drivers de dispositivos
- Soporte de sistemas de archivos
- Hacks del kernel
- Opciones de seguridad
- Módulos criptográficos
- Virtualización
- Rutinas de bibliotecas

La configuración generará un archivo **.config** que guía a make en la compilación.

3)Compilación

Utiliza **make** en el directorio fuente del kernel

<code>\$make</code>	Compila el fichero principal del kernel
<code>\$make modules</code>	Compila los módulos
<code>\$make all</code>	Compila kernel y módulos
<code>\$make grep -iw "error"</code>	Muestra solamente los errores en la compilación

4) Instalación

Ejecutar make con el parámetro install

```
#make install
```

Copia de la ubicación del árbol fuente **arch/x86/boot**, el fichero del kernel al directorio **/boot**

El archivo **System.map** contiene indicadores a funciones en el kernel y se utiliza para depurar problemas.

Es aconsejable copiarlo en **/boot**.

El comando **make install** realiza automáticamente la copia del fichero del kernel, de System.map, la creación del disco RAM inicial para controladores adicionales y la modificación de GRUB.

(También es aconsejable hacerlo a mano)

Este comando depende del script **installkernel**.

Instalación de los módulos compilados

```
#make modules_install
```

Utilizar el comando **make modules_install** creando un subdirectorio en **/lib/modules** de la versión del kernel instalada y copiando los módulos

Linux requiere información de las dependencias entre los módulos.

Es necesario regenerar el fichero **modules.dep** con el comando **depmod**.

Por lo general, make modules_install llama a la utilidad depmod para hacerlo automáticamente.

Resumen para compilar un kernel a la “Debian”

Descargamos el kernel y lo ubicamos en /usr/src

<pre>#cd /usr/src/linux-source</pre>	Entramos en el árbol del código fuente
<pre>#make menuconfig</pre>	Configuramos opciones
<pre>#make deb-pkg</pre>	Compilamos creando un paquete “deb” del kernel
<pre>#make clean</pre>	Limpiamos árbol del código fuente
<pre>#dpkg -i [paquete.deb]</pre>	Instalamos el paquete creado

TEMA 7.1 – GESTIÓN DE REDES

Linux es un SO orientado a redes.

Los dispositivos de red en Linux no tienen entradas en **/dev**.

En su lugar las utilidades de red de bajo nivel toman los nombres de los dispositivos y se trabaja con ellos directamente.

La mayoría de las distribuciones vienen con scripts de inicio que detectan automáticamente la tarjeta de red, cargan el módulo correcto y realizan las configuraciones.

El kernel Linux puede funcionar como un router, puede conectar dos o más redes dirigiendo el tráfico en base a su tabla de enrutamiento.

Configuración rápida 1 de interfaces

```
#ifconfig [interfaz] up [IP] netmask [máscara]
#route add default gw [IP gateway]
```

Configuración rápida 2 de interfaces

```
#ip addr add [IP] dev [interfaz]
#ip route add default via [IP gateway] dev [interfaz]
```

Definir el nombre de host (/etc/hostname)

```
#hostname [nombre de host]
```

La mayoría de las distribuciones crean una interfaz virtual en RAM reservada para el tráfico interno:

```
#ifconfig lo 127.0.0.1
#route add 127.0.0.1 lo
$ping -c 4 localhost
```

Ficheros de configuración permanentes

- Red Hat → **/etc/sysconfig/network-scripts/ifcfg-eth***
- Debian → **/etc/network/interfaces**

Configurar interfaces

```
#ifconfig [interfaz] [parámetros]
```

Parámetros

up / down	Activa o desactiva la interfaz
address [IP]	Asigna la dirección IP especificada
netmask [mask]	Especifica la máscara de subred
broadcast [IP]	Define la dirección de broadcast
hw ether [MAC]	Cambia la dirección física de la interfaz
mtu [n]	Asigna la máxima unidad de transferencia (por defecto 1500)
multicast	Activa la transmisión multicast
arp / -arp	Activa o desactiva el uso del protocolo ARP
promisc / -promisc	Activa o desactiva el modo promiscuo

Modificar la tabla de enrutamiento

```
#route [opciones] [comando] [tipo] [dirección] [modificadores]
```

Opciones

-n	Muestra direcciones numéricas
-v	Detalla el proceso

Comandos

add	Añadir una ruta
del	Eliminar una ruta

Tipos de ruta

-net	Se trata como una ruta a una red
-host	Se trata como una ruta simple a un host

Modificadores

dev [interfaz]	Asocia una ruta con la interfaz especificada
netmask [máscara]	Utiliza la máscara indicada
gw [gateway]	Configura la puerta de acceso
metric [n]	Asigna la métrica de enrutamiento
reject	Instala una ruta de bloqueo

Diseño de la tabla de enrutamiento

Destination	Dirección destino
Genmask	Máscara de red de la dirección destino
Gateway	Dirección donde enviar los paquetes
Flags	U Activo G Gateway
Metrica	Métrica
Iface	Interfaz de salida

Activar el reenvío de paquetes entre interfaces

En el fichero `/etc/sysctl.conf`:

- `net.ipv4.ip_forward=1`

```
#echo "1" > /proc/sys/net/ipv4/ip_forward
```

Herramientas de gestión de red

Mostrar y modificar la tabla ARP

```
$arp [opciones]
```

Opciones

-a	Utiliza una salida simplificada
-n	Muestra información de forma numérica
-s [host/IP] [MAC]	Añade una entrada permanente (netmask [máscara])
-d [host/IP]	Elimina la entrada del host especificado
-v	Modo descriptivo adicional

Linux mantiene las entradas de la caché ARP durante un tiempo limitado

Ping

Envía paquetes ICMP ECHO_REQUEST al host de la red especificado

```
$ping [opciones] [IP/nombre de host]
```

Opciones

-c [n]	Manda un numero definido de paquetes
-i [n]	Especifica un intervalo en segundos entre el envio de cada paquete (por defecto 1)
-t [n]	Modifica el TTL del paquete

Traceroute

Traza la ruta a un host modificando el campo **TTL** de la cabecera del paquete enviado

```
$traceroute [opciones] [IP/nombre de host]
```

Opciones

-T	Utiliza TCP
-U	Utiliza UDP
-I	Utiliza ICMP

Cliente DHCP

El cliente DHCP envía una multidifusión en busca de un servidor DHCP.

El cliente autoconfigura la red con los parámetros ofrecidos por el servidor DHCP.

```
#dhclient [opciones][interfaz]
```

Opciones

-v	Modo detallado
-d	Ejecutar como proceso en primer plano
-x	Detener el cliente
-s [srv]	Especifica el servidor a utilizar
-p [port]	Especifica el puerto a utilizar

Buscar información de un dominio o dirección IP

```
#whois [IP/nombre de host]
```

Utiliza una base de datos pública de Internet

Puertos de red

Número añadido tras la dirección IP para dirigir el tráfico a un programa específico (TCP/UDP)

Los sistemas Unix restringen el acceso a los puertos con privilegios (inferiores a 1024)

Los usuarios pueden acceder a los números de puertos superiores al 1024

Asociación de puertos con nombres

- `/etc/services`

Netstat

Imprime información sobre los sockets activos, las tablas de enrutamiento, las interfaces, las conexiones enmascaradas (NAT) y los grupos multicast.

```
#netstat [opciones]
```

Opciones

	sin opciones	
		Devuelve información sobre los puertos abiertos y sockets internos
-i	--interface [interfaz]	Devuelve información acerca de la interfaz
-r	--route	Devuelve información acerca de la tabla de enrutamiento
-M	--masquerade	Devuelve información del enmascaramiento NAT
-g	--groups	Muestra los grupos multicast
-s	--statistics	Muestra las estadísticas para cada protocolo
-a	--all	Muestra todas las entradas (puertos y sockets)
-t	--tcp	Limita la información a los sockets TCP
-u	--udp	Limita la información a los sockets UDP
-l	--listening	Muestra sólo los sockets que escuchan
-n	--numeric	No resuelve nombres (recomendado)
-c	--continuous	Muestra información refrescando una vez por segundo
-e	--extend	Amplia el nivel de detalle
-v	--verbose	Modo detallado

Tcpdump

Intercepta y registra las cabeceras y los paquetes que circulan por la red que coinciden con la expresión.

```
#tcpdump [opciones][expresión]
```

Opciones

-w [fichero]	Escribe la información en un fichero
-r [fichero]	Lee un fichero
-i [interfaz]	Escucha en la interfaz especificada
-c [n]	Finaliza tras recibir el número especificado de paquetes
-n	No realiza resolución de nombres
-A	Imprime el contenido los paquetes en texto ASCII
-x	Imprime el contenido delos paquetes en hexadecimal
-t	Omite el la marca de tiempo
-v / -vv / -vvv	Detalla la salida
-q	Abrevia la salida

Expresiones (se pueden conectar mediante “and”)

Dirección	src [IP] dst [IP]
Protocolo	ether fddi tr wlan ip ip6 arp rarp tcp udp
Tipo	Host [nombre] net [IP] port [n]

Netcat

Inicia y supervisa conexiones TCP y UDP

```
#netcat / nc [opciones] [host] [puerto]
```

Opciones

-v	Utiliza una salida más descriptiva
-u	Utiliza el protocolo UDP (por defecto utiliza TCP)
-n	No resuelve nombres
-w [n]	Especifica en segundos el tiempo límite de cierre de una conexión
-p [n]	Abre y lee del puerto especificado
-l [n]	Especifica un puerto para que se comporte como servidor (estado de escucha)
-z [n-n]	Realiza un escaneo de puertos

TEMA 7.2 – RESOLUCIÓN DE NOMBRES

DNS

Base de datos distribuida que realiza conversiones entre direcciones IP y nombres de host.

Especificación de los servidores DNS

/etc/resolv.conf

nameserver [IP DNS]

*Se pueden incluir hasta tres servidores

Asociaciones de IP con nombres de host (local)

/etc/hosts

[IP] [nombre de host]

Configuración de la resolución de nombres

- **/etc/nsswitch.conf**
 - files → resuelve localmente los nombres estáticos
 - dns → resuelve empleando unicast DNS
 - mdns4 → resuelve empleando multicast DNS

Uso del cliente DNS

#dig [opciones][opciones de consulta] [registro]

Opciones

@[servidor]	Servidor a consultar (si no se especifica se comprueba en resolv.conf)
-q [nombre de dominio]	Especifica el nombre del dominio a consultar
-p [n]	Especifica el número de puerto (por defecto el 53)
-x [IP]	Realiza una búsqueda inversa

Opciones de consulta

+tcp, +notcp	Utiliza o no, el protocolo TCP en lugar de UDP
+search, +nosearch	Utiliza o no, el fichero /etc/resolv.conf
+time=[n]	Especifica el tiempo límite para las consultas (5s)
+tries=[n]	Especifica el número de intentos para las consultas UDP (3s)

TEMA 7.3 – REDES INALÁMBRICAS

CRDA (*Central Regulatory Domain Agent*)

Utilizado por udev para establecer el dominio regulatorio de la comunicación

IWLIST

Proporciona información sobre la interfaz de red inalámbrica y escanea las redes del entorno

```
#iwlist [interfaz] [comando]
```

Comandos

scan	Muestra la lista de redes disponibles
freq	Muestra la lista de canales disponibles para el dispositivo
rate	Muestra lista de velocidades soportadas por el dispositivo
txpower	Muestra lista de parámetros de energía de transmisión disponibles

IWCONFIG

Determina los parámetros de las interfaces inalámbricas

```
#iwconfig [interfaz] [parámetros]
```

Parámetros

sin opciones	Muestra los parámetros actuales
essid ["ssid" any]	Ajusta el nombre de red
channel [n auto]	Ajusta el canal
txpower [n nmW auto off]	Ajusta la potencia de transmisión en dBm
key [clave hex s:clave ASCII]	Ajusta la clave de encriptación

Conectar a una red abierta o con seguridad WEP

```
#iwconfig [int] essid ["SSID"] channel [n] key [clave]
```

Cambiar potencia de transmisión de antena

```
#iw reg set B0
```

```
#iwconfig [interfaz] txpower [dbm]
```

Equivalencias

20 dbm	100 mw
27 dbm	500 mw
30 dbm	1 W

IWGETID

Proporciona parámetros de la red y la interfaz en uso

```
#iwgetid [interfaz] [opciones]
```

Opciones

-a	--ap	Muestra la MAC del punto de acceso
-f	--freq	Muestra la frecuencia en uso
-c	--channel	Muestra el canal en uso
-m	--mode	Muestra el modo en uso
-p	--protocol	Muestra el protocolo en uso
-r	--raw	Muestra el ESSID

Cliente WPA y de autenticación 802.11X

```
#wpa_supplicant [opciones]
```

Opciones

-i [interfaz]	Especifica la interfaz
-D [driver]	Especifica el driver
-c [fichero]	Especifica el fichero de configuración
-B	Lanza el programa en segundo plano
-d	Salida detallada

Interfaz de comando del cliente WPA

```
#wpa_cli [opciones] [comando]
```

Opciones

-i [interfaz]	Especifica la interfaz a configurar
----------------------	-------------------------------------

Comandos

status	Muestra el estado de la autenticación
reassociate	Fuerza la reasociación
reconfigure	Fuerza a releer el fichero de configuración
terminate	Termina el demonio wpa supplicant

Conectar a una red con seguridad WPA/WPA2

```
#wpa_passphrase [essid] [clave] > wpa.conf
```

```
#wpa_supplicant -D [driver] -i [interfaz] -c/ruta/wpa.conf -B
```

TEMA 8 – INTRODUCCIÓN AL FILTRADO DE PAQUETES

El kernel Linux utiliza una serie de reglas para determinar que hacer con los paquetes que recibe o transmite por los procesos locales. El kernel lee las cabeceras y filtra según reglas.

Las reglas se disponen en cadenas (chains) y proporcionan una serie de acciones que deben llevarse a cabo si un paquete se ajusta a la regla.

→ Cuando un paquete encuentra una regla que lo define, la aplica y no lee las siguientes.

Las cadenas se organizan en **tablas**:

- **FILTER** (Filtrado de paquetes)
- **NAT** (Enmascaramiento de direcciones)
- **MANGLE** (Modifica paquetes de forma especial)

Para crear un cortafuegos de filtrado de paquetes se debe diseñar un serie de reglas para tablas y cadenas específicas.

La tabla por defecto es **FILTER**, contiene las **cadenas**:

- **INPUT** → Procesa paquetes recibidos localmente.
- **OUTPUT** → Procesa paquetes originados localmente transmitidos a sistemas externos.
- **FORWARD** → Procesa paquetes de ruta/pasarela

Cada paquete pasa solo por una de estas cadenas.

Políticas usuales

ACCEPT	Acepte el paquete y se lo pase a la siguiente cadena
DROP	Ignore el paquete, lo descarta
REJECT	Rechaza el paquete, enviando al emisor un código reject (ICMP)

Tipos de implementaciones

Permisiva → Política por defecto ACCEPT

- Se acepta todo el tráfico y solo se deniega explícitamente.

Restrictiva → Política por defecto REJECT/DROP

- Todo el tráfico es denegado o descartado y solo se permite explícitamente.

Manipular las reglas del cortafuegos

```
#iptables [opciones]
```

Opciones

-t [tabla]	--table	Especifica la tabla sobre la que operar
-L [cadena]	--list	Muestra la configuración las cadenas o de la especificada
-S [cadena]	--list-rules	Muestra las reglas de en formato corto
-F [cadena]	--flush	Elimina las reglas de todas las cadenas o de la especificada
-P [cadena] [política]	--policy	Establece la política de la cadena especificada
-A [cadena] [regla]	--append	Añade una nueva regla a la cadena
-D [cadena] [regla/n]	--delete	Elimina una regla de la cadena
-v		Informa sobre los paquetes que atraviesan las cadenas
-n		Muestra los resultados en forma numérica

Añadir reglas

```
#iptables -A [cadena] [criterios] -j [política]
```

Criterios de coincidencia

-m [nombre]	--match	Añade reglas ampliadas por el módulo nombrado
-p [prot]	--protocol	Establece el protocolo [all,tcp,udp,icmp,esp,ah,sctp...]
-s [IP/mask]	--source	Especifica la dirección de origen
-d [IP/mask]	--destination	Especifica la dirección de destino
-i [int]	--in-interface	Especifica la interfaz de entrada (INPUT)
-o [int]	--out-interface	Especifica la interfaz de salida (OUTPUT)
	--dport [puerto]	Especifica el puerto destino, es posible definir un rango
	--sport [puerto]	Especifica el puerto origen, es posible definir un rango
	--icmp-type [tipo]	Especifica el código del tipo de paquete ICMP (8)
--to [IP]	--to-source	Cambia la dirección origen por la especifica (SNAT)
	--to-destination	Cambia la dirección destino por la específica (DNAT)
	--state [estado]	Especifica el estado de la conexión
		INVALID,NEW,ESTABLISHED,RELATED
-j [política]	--jump	Indica que hacer si el paquete coincide con la regla
-g [cadena]	--goto	Indica que continúe procesando en una nueva cadena

Algunos ejemplos de uso de IPTABLES

Ver estado y reglas

```
#iptables -L -n -v
```

Establecer política por defecto

```
#iptables -P [cadena] [política]
```

Aceptar conexiones internas (loopback)

```
#iptables -A INPUT -i lo -j ACCEPT
```

Aceptar/Descartar conexión al puerto/s determinados

```
#iptables -A INPUT -p [prot] -dport [n] -j [ACCEPT/DROP]
#iptables -A INPUT -s 0.0.0.0/0 -p tcp -dport [1:1024] -j [ACCEPT/DROP]
#iptables -A INPUT -s 0.0.0.0/0 -p udp -dport [1:1024] -j [ACCEPT/DROP]
```

Permitir/Denegar conexión a determinadas IPs

```
#iptables -A INPUT -s [IP origen] -p [prot] -dport [n] \
-j [ACCEPT/DROP]
```

Obtener acceso al firewall desde una red local

```
#iptables -A INPUT -s [dirección de red/máscara] \
-i [interfaz local] -j ACCEPT
```

Filtrar acceso de la red local al exterior

```
#iptables -A FORWARD -s [dirección de red/máscara] \
-i [interfaz local] -p [prot] -dport [n] -j [ACCEPT/DROP]
```

ANEXOS

1.EVOLUCIÓN HISTÓRICA DE UNIX Y GNU/LINUX

Década de los 60

- Un consorcio industrial formado por Bell Labs, AT&T y General Electric desarrollan **MULTICS** (*Multiplexed Information Computing Service*) para uso en mainframes de procesamiento por lotes.
- Ken Thompson (Bell Labs) desarrolla **UNICS** (*Uniplexed Information Computing Service*), versión simplificada de MULTICS para su uso en minicomputadoras PDP.
- Llega la portabilidad del código de la mano de Ken Thompson y Dennis Ritchie (Bell Labs) con la creación del lenguaje **C** (sucesor del lenguaje B).
- Reescriben UNICS en C y ensamblador, dando lugar al sistema operativo **UNIX**.

Década de los 70

- El código fuente de UNIX es distribuido por universidades con propósito educativo y con intención de ser analizado y mejorado.
- Surgen varias versiones de UNIX, a destacar el **System V** de AT&T y el **BSD** de la universidad de Berkeley.

Década de los 80

- El IEEE (Instituto de ingenieros y electrónicos) normaliza el **POSIX** (1003.1), conjunto de procedimientos y estándares entre sistemas UNIX, mejorando la compatibilidad del software entre diferentes versiones.
- Algunas empresas empiezan a licenciar y a comercializar sus versiones de UNIX.
- Andrew Tanenbaum, profesor de una universidad holandesa, desarrolla una versión simplificada de UNIX conocida como **MINIX** con propósito educativo.
- Richard Stallman, investigador del MIT, diseña el sistema operativo **GNU** (*GNU's Not UNIX*)
A parte también inventa:
 - La licencia **GPL** (*GNU General Public License*) que permite el libre uso, distribución y modificación de un software.
 - La **FSF** (*Free Software Foundation*) que defiende el uso del software libre.
 - El “**Copyleft**” que evita que el software se proteja con “Copyright”
- IBM lanza el **IBM PC** con MSDOS/BASIC dominando el mercado y acelerando la expansión de computadoras personales.

Década de los 90

- Linus Torvalds, estudiante finlandés, modifica MINIX para uso en su computadora personal creando así el núcleo **LINUX**.
Publica bajo GPL su sistema Linux en Internet, generando gran aceptación y atención de desarrolladores de todo el mundo.
- El GNU de Stallman no era todavía funcional y Linux era un kernel libre. Juntaron ambos proyectos creando la unión **GNU/LINUX**, un sistema operativo totalmente libre.
- Aparecen las primeras distribuciones basadas en GNU/LINUX (Debian, Red Hat, Slackware) para facilitar el uso a las comunidades de usuarios.

2. COMPARATIVA GNU/LINUX – WINDOWS

LINUX	WINDOWS
<p>Código abierto.</p> <p>Distribuye el código y muestra todo lo que ocurre en el sistema.</p> <p>Dificulta la existencia de programas que pongan en riesgo el sistema.</p>	<p>Código cerrado.</p> <p>No muestra el código y oculta casi todo lo que ocurre en el sistema.</p> <p>Facilita la transmisión de programas que ponen en riesgo al sistema.</p>
<p>Gratuito.</p> <p>Es propiedad de la comunidad de desarrolladores y de los propios usuarios.</p>	<p>De pago.</p> <p>Es propiedad de una corporación privada.</p> <p>Suele ser caro.</p>
<p>Legal.</p> <p>Gracias a su licencia GPL.</p>	<p>Legal si pagas la licencia.</p> <p>En otro caso, pirata.</p>
<p>Limpio.</p> <p>Instalaciones personalizadas.</p> <p>Todo el software ha sido revisado por la comunidad.</p>	<p>Sucio.</p> <p>Incorpora spyware por defecto y permite la posibilidad de que los programas instalen adware y ejecuten código malicioso (virus, troyanos ...)</p>
<p>Kernel monolítico con opción a compilarse modularmente.</p> <p>Los controladores y otros subsistemas pueden no formar parte del núcleo.</p> <p>Es más estable y solo es necesario reiniciar cuando se instala un nuevo núcleo.</p>	<p>Microkernel híbrido.</p> <p>Los controladores y otros subsistemas vienen integrados en el núcleo.</p> <p>Es más rápido pero más propenso a fallos que afecten al núcleo, el resultado son cuelgues y es necesario reiniciar para casi cualquier actualización en el sistema.</p>
<p>Núcleo ligero</p> <p>Unas 8 millones de líneas de código.</p> <p>Unas 300 llamadas al sistema.</p>	<p>Núcleo pesado</p> <p>Unas 25 millones de líneas de código.</p> <p>Unas 1000 llamadas al sistema.</p>
<p>Portable.</p> <p>El kernel Linux puede correr en multitud de arquitecturas de hardware (Alpha, ARM, RISC, x86, x86_64, IA-64, MIPS, PowerPC, SPARC ...)</p>	<p>Definido.</p> <p>Solo corre en arquitecturas de hardware x86, x86_64 y ARM.</p>
<p>Complicado.</p> <p>Diseñado para ser flexible, no para ser fácil.</p>	<p>Fácil</p> <p>Diseñado para utilizarlo sin conocimientos.</p>
<p>Falta compatibilidad con nuevo hardware.</p> <p>La industria no se ha esforzado demasiado (salvo excepciones) en proporcionar controladores para Linux</p>	<p>Compatible con nuevo hardware.</p> <p>Viene preinstalado gracias al modelo de negocio OEM que Microsoft pacta con los fabricantes.</p>

3 – ENLACES DE INTERÉS

Distrowatch → <http://distrowatch.com/>

Proporciona información sobre multitud de distribuciones, establece un ranking basado en descargas.

The Linux Documentation Project → <http://tldp.org/>

Documentación mantenida por la comunidad. Traducida a varios idiomas.

Linux Tracker → <http://linuxtracker.org/>

Repositorio de torrents de imágenes ISO GNU/Linux

DISTRIBUCIONES PRINCIPALES

Debian → <https://www.debian.org/index.es.html>

Distribución multipropósito que sigue decididamente los principios del software libre.

Una gran variedad de distribuciones actuales derivan de Debian.

Ubuntu → <https://www.ubuntu.com/>

Popular distribución de escritorio basada en Debian.

Disponible en varias categorías y en varios entornos gráficos.

Red Hat Enterprise Linux → <https://www.redhat.com/es/technologies/linux-platforms/enterprise-linux>

Distribución enfocada a entornos corporativos registrada por Red Hat Inc.

Restringe la redistribución pero sigue siendo de código abierto.

CentOS → <https://www.centos.org/>

Copia binaria de Red Hat sin restricciones y mantenida por la comunidad. Enfocada a servidores.

Fedora → <https://getfedora.org/es/>

Distribución multipropósito basada en Red Hat.

Disponible en las categorías de escritorio, servidor y host de virtualización.

Slackware Linux → <http://www.slackware.com/>

Distribución multipropósito más antigua todavía mantenida.

Diseñada para la estabilidad y simplicidad.

OpenSUSE → <https://es.opensuse.org/>

Distribución multipropósito desarrollada por Novell basada en Slackware.

Arch Linux → <https://www.archlinux.org/>

Distribución minimalista muy personalizable.

Basada en el principio KISS (Keep It Simple, Stupid). Enfocada a usuarios avanzados.

Gentoo Linux → <https://www.gentoo.org/>

Distribución independiente basada en código fuente.

Todo el software se compila localmente. Enfocada a usuarios avanzados.

4. FUENTES DE REFERENCIA

Bibliografía

Sistemas operativos modernos. Andrew S. Tanenbaum. Pearson Educación, 2009

Guia de estudio LPIC-1. Roderick W. Smith. Ediciones Anaya Multimedia, 2010.

Guia de estudio LPIC-2. Roderick W. Smith. Ediciones Anaya Multimedia, 2011.

Linux kernel in a nutshell. Greg Kroah Hartman. O'Reilly Media, 2006

Webgrafía

Debian wiki: <https://wiki.debian.org/>

Arch wiki: <https://wiki.archlinux.org/>

The Linux Documentation Project: <http://tldp.org/>

-----EOF