



LINUX

Instalación y administración básica

Versión: 1.0.1

Alfredo Barrainkua Zallo

Mayo del 2007



SOME RIGHTS RESERVED CreativeCommons - ShareAlike Lizentzia laburpena: English Castellano

Índice

1. Introducción	4
1.1. GPL (General Public License) o Copyleft	4
1.2. Linux	5
1.3. FLOSS (Free Libre Open Source Software)	5
1.4. GNU / Linux	
1.5. Distribuciones Linux	
1.6. El administrador	
1.7. Particionado del disco duro	
2. Instalación	8
2.1. GRUB	
3. Trabajando con Linux	
3.1. Ficheros y directorios	
3.2. El árbol de directorios	
3.3. Consolas virtuales	
3.4. Directorio de trabajo de los usuarios (home)	
3.5. Directorio padre e hijo	
3.6. Listados	
3.7. Sistema de permisos	
3.8. Enlaces simbólicos	
4. Comandos básicos	15
4.1. Comodines	
4.2. Práctica	
4.3. Ejecutando como root	
5. Gestión de usuarios	19
6. Programación de tareas	21
7. Arranque y parada del sistema y niveles de ejecución	
7.1. Proceso de arranque	
7.2. Niveles de ejecución	23
7.3. Sysinit	
7.4. Scrips de inicio y parada de servicios	
7.5. Modo monousuario	
7.6. Parada del sistema	
8. Procesos	
8.1. Genealogía de procesos	
8.2. Control de procesos	
8.3. Matando procesos	
8.4. Control de tareas en Linux	
8.5. Redirección de E/S	
8.6. Intercomunicación de procesos: PIPES (tuberías)	
9. Sistema de registro (log)	
9.1. Configuración de Logs	
9.2. Ficheros de logs	

9.3. Visualización de los ficheros.	
9.4. Rotación de los ficheros de log	
10. La red	
11. Actualización del sistema e instalación de paquetes	
11.1. Instalación de paquetes	43
12. Tareas importantes a realizar	45
12.1. El nombre de la máquina	45
12.2. La resolución de nombres	45
12.3. Deshabilitar el reinicio por teclado	46
12.4. La hora, la hora, la hora.	46
12.5. Acceso por la red : SSH	47
13. Más comandos	
14. Ficheros importantes del sistema.	49
15. Referencias	53
16. Autor	54

1. Introducción

En esta introducción vamos a conocer unos conceptos básicos.

1.1. GPL (General Public License) o Copyleft

A diferencia de Copyright, que se basa en los derechos de los autores, la licencia GPL, hace hincapié en los derechos de los usuarios,

La licencia GPL permite a las personas vender el software libre (free) y ganar dinero con el, pero no permite restringir a otros los derechos de distribuir el software de la misma manera.

Hay que diferenciar el concepto de software libre (free) con el de dominio público (public domain). El software de dominio público no tiene copyright y es propiedad de todo el mundo. El software GPL es copyright del autor o autores.

La licencia GPL permite a la gente modificar y distribuir el software pero cualquier trabajo derivado del soft GPL debe de estar cubierto por la licencia GPL posteriormente. Es decir, alguien podría modificar Linux pero no podría venderlo bajo una licencia restrictiva.

Las personas y las organizaciones pueden distribuir el software GPL para regalarlo o ganar dinero con él. El distribuidor del software debe dejar claro a los usuarios que el software está cubierto por GPL. Los distribuidores deben proporcionar, libre de cargo, el código fuente completo del software que está siendo distribuido o deben indicar desde donde se puede bajar ese software por Internet de manera gratuita.

Las 4 libertades

0 - libertad para utilizar el programa para cualquier propósito.

1 - libertad para poder estudiar cómo funciona el programa. Implica acceso al código fuente del mismo.

2 - libertad para redistribuir el programa.

3 - libertad para hacer modificaciones y distribuir las mejoras. Implica también acceso al código fuente del mismo.

Para más información sobre la licencia GPL: http://www.gnu.org/copyleft/gpl.html

1.2. Linux

Linux es un Sistema Operativo compatible con Unix. Es de libre distribución y fue originalmente desarrollado por Linus Torvalds quien empezó a trabajar en el proyecto en 1991 como estudiante de la Universidad de Helsinki. Linus dejó que el código se distribuyera a través de Internet. Actualmente Linux está mantenido por un grupo de miles de personas colaborando a través de Internet aunque los kernel son controlados por un grupo de personas más reducido (entre ellos Linus Torvalds).

Linux es solamente el núcleo del sistema operativo (KERNEL).

1.3. FLOSS (Free Libre Open Source Software)

Este término hace referencia a software libre en general. Este software puede estar licenciado bajo GPL o cualquiera de otras licencias compatibles con ella.

Cabe destacar que se utiliza el término "Libre", para evitar el doble sentido que el término "Free" tiene.

El término correcto es "Free as free speech" (Libre), y no "Free as free beer" (Gratis).

1.4. GNU / Linux

Este término hace referencia a todo el sistema operativo. El núcleo Linux, y todas las utilidades tanto de espacio del kernel como de espacio de usuario.

1.5. Distribuciones Linux

Una distribución es un núcleo Linus determinado, con las herramientas herramientas de espacio de usuario y programas adicionales, organizada de una manera, y generalmente con un sistema de empaquetamiento de los diferentes programas.

Existen infinidad de distribuciones Linux, pero podemos destacar las siguientes:

- Debian
- Red Hat
- SUSE Linux
- Ubuntu
- Linspire
- Gentoo

Ubuntu y Debian utilizan el sistema de paquetes **deb** de Debian.

SUSE y Red Hat utilizan los paquetes RPM, desarrollados por Red Hat.

1.6. El administrador

Un sistema operativo UNIX (y cualquier otro sistema servidor) debe tener al menos una persona encargada de su mantenimiento. Esta persona es el Administrador del Sistema. La labor del administrador es asegurar el correcto funcionamiento del sistema y de los servicios que éste ofrece a los usuarios procurando un óptimo rendimiento.

Para ello el administrador debe realizar una serie de tareas que requieren un conocimiento del sistema en profundidad. Entre otras:

- Arranque y parada del sistema
- Gestión y soporte de usuarios
- Instalación, configuración y mantenimiento de dispositivos
- Instalación de software
- Comprobación de los niveles de utilización rendimiento (CPU, disco, memoria)
- Mantenimiento de la seguridad del sistema
- Realización de copias de seguridad
- Mantenimiento de la historia del sistema

La mayoría de estas labores requieren ser realizadas por un usuario con privilegios especiales. En todo sistema UNIX existe una cuenta de usuario conocido como "superusuario" que tiene todos los privilegios del sistema. El nombre de esta cuenta es **root**.

1.7. Particionado del disco duro

El disco duro de un ordenador, ha de ser particionado para crear los sistemas de ficheros. El sistema de ficheros consta de una estructura (tabla, árbol, ...) que contiene los metadatos del sistema de ficheros, y los ficheros en sí. En este curso no vamos a entrar en el estudio de los sistemas de ficheros.

A diferencia de Windows donde es habitual que un disco duro tenga una única partición o dos en el caso de un servidor, en Linux es habitual crear tres o cuatro particiones. El número de ellas y su tamaño varía ampliamente según la aplicación a que vaya destinada la máquina. Además hay que mencionar que aunque en Windows se utiliza un fichero para el intercambio de páginas entre la memoria RAM y el disco duro (memoria virtual), en Linux se utiliza una partición separada y específica para este fin. Puede haber hasta 16 particiones de esas.

Una aplicación típica de estación de trabajo o servidor genérico podría tener las siguientes particiones:

/boot	500MB	Imágenes de arranque y otros ficheros afines
-------	-------	--

swap	2 – 4 GB	partición para el intercambio de páginas
/		Directorio raíz

Una aplicación típica de servidor de correo (**maildir**) o servidor de homes de usuario será:

/boot	500MB	Imágenes de arranque y otros ficheros afines
swap	2 – 4 GB	partición para el intercambio de páginas
/	8 GB	Directorio raíz
/home		Datos de usuarios en local o unidad de almacenamiento externa

Una aplicación típica de servidor web o ftp anónimo será:

/boot	500MB	Imágenes de arranque y otros ficheros afines
swap	2 – 4 GB	partición para el intercambio de páginas
/	8 GB	Directorio raíz
/var		Bases de datos, páginas web, ficheros de descarga en local o unidad de almacenamiento externa

Obviamente, en grandes sistemas, habrá una organización más compleja. Aquí hemos realizado una división basada en la cantidad de almacenamiento, pero hay otros considerandos, como el rendimiento y la seguridad.

Es muy común realizar particiones o discos enteros para ser exportados para su lectura en la red. Estas particiones se montan como de sólo lectura.

Para una aplicación genérica, nos basta con el primer ejemplo. Hemos de tener en cuenta también si compartimos el disco con otro sistema operativo. En el caso de un servidor no tiene sentido, pero lo tendremos que tener en cuenta.

IMPORTANTE: Las especificaciones de discos ATA limitan la cantidad de particiones primarias a 4. Además la suma de primarias y extendidas ha de ser también 4. En una unidad extendida podemos tener varias unidades lógicas, pero hay algunos sistemas operativos (Windows) que no pueden arrancar de unidades que no sean primarias y estén marcadas como activas.

2. Instalación

Aunque la instalación de sistemas Linux tiene fama de difícil y peligrosa, nada más lejos de la realidad. El mayor problema a la hora de una instalación es que haya un sistema operativo previamente instalad. Esto hace que debamos de redimensionar la partición ocupada por el mismo para hacer sitio, o que debamos particionar a mano. Lo haremos de esta forma, que es la mejor para aprender.

Vamos a utilizar un sistema Debian. Concretamente **Debian 4.0r0** para arquitecturas i386. Insertamos y arrancamos el sistema desde un CD de arranque.

Seleccionamos:

Idioma	-> Spanish		
País	-> España		
Teclado	-> Español		
Detección	del hardware		
Configura	ción de la red con DE	ICP Cancelar.	
Configura	r a mano		
Dire	cción IP:		10.22.1.xx
Más	cara de red:		255.255.255.0
Pasa	rela (gateway):		10.22.1.1
Dire	cción del servidor de	nombres (DNS):	10.22.1.8
Nombre de	e la máguina:	I2DBxx	
Nombre de	el dominio:	iurreta-institutua.	net
Método de	particionado: Manu	al	
En el ESPA	ACIO LIBRE:		
Crea	ar partición nueva.		
Tam	año de la partición:		500 MB
Tipo	de partición:		Primaria
Ubio	cación de la nueva pa	rtición:	Principio
Pun	to de montaje:		/boot
Se h	a terminado de defin	ir la partición.	
En el ESPA	ACIO LIBRE:	_	
Crea	ar partición nueva.		
Tam	año de la partición:		1 GB
Tipo de partición: Primaria			

Ubicación de la nueva partición:	Principio		
Utilizado como:	Área de intercambio		
Se ha terminado de definir la partición.			
En el ESPACIO LIBRE:			
Crear partición nueva.			
Tamaño de la partición:	10 GB		
Tipo de partición:	Lógica		
Ubicación de la nueva partición:	Principio		
Punto de montaje:	/		
Se ha terminado de definir la partición.			
Finalizar el particionamiento y escribir los camb	ios en el disc	0.	
Seleccionar la ubicación de sus zona horaria:	Península		
Clave del superusuario (root):	na103	(2 veces)	
Configurar usuario y contraseña			
Nombre completo:	Usuario inio	cial	
Nombre de usuario:	iurreta		
Contraseña: iurreta (2			
Configuración del gestor de paquetes			
Desea utilizar una réplica en red:	Si		
Protocolo para descargar ficheros:	http		
Ubicación:	España		
Repositorio:	ftp.es.debia	n.org	
Si tiene que usar un proxy http:	NO	-	
Desea participar en la encuesta sobre el uso de p	aquetes:	NO	
Elegir los prog ramas a instalar:			
Entorno de escritorio: X			
Sistema estándar: X			

2.1. GRUB

GRUB es el sistema de arranque utilizado en la actualidad por prácticamente todas las distribuciones Linux. Nos presenta en un menú las opciones que tenemos al arrancar. Normalmente serán los sistemas operativos que hay instalados y alguna utilidad de diagnóstico. Es típico, que al actualizar el kernel del sistema, se ponga el nuevo primero y lo arranque por defecto, pero nos permita arrancar alguno antiguo, en caso de que tengamos problemas con el nuevo. En el sector 0 de la pista 0 (el MBR) se pone un puntero al manejador de arranque (GRUB) situado en /boot.

3. Trabajando con Linux

3.1. Ficheros y directorios

El sistema de ficheros de UNIX es jerárquico o de tipo piramidal. Existe un directorio raíz llamado "/" del cual cuelga el resto de los directorios o ficheros. De estos directorios cuelgan a su vez otros ficheros y directorios.

Para Unix un fichero es una colección de bytes referenciados por un nombre y sin ninguna estructura interna o asignación de significado a su contenido.

Existen varios tipos de fichero:

Ordinarios	Documentos, programas fuente, ejecutables y ficheros de datos.
Directorios	Ficheros que contiene referencias a otros ficheros. Es decir, contienen los nombres de todos los ficheros que están colgados de él, así como información sobre la localización en el disco de cada uno de los ficheros y subdirectorios que contiene
Dispositivos	Son ficheros especiales que no contienen información. Se trata de canales de acceso a los dispositivos de E/S del sistema y así nos encontramos con que por cada periférico existe alguno de estos ficheros especiales asociado. Cuando se quiere realizar una operación de E/S sobre el periférico el trabajo se hace a través de su fichero asociado. Todos estos ficheros se encuentran en un directorio especial llamado /dev .

En el directorio /dev nos podemos encontrar con dispositivos como:

/dev/fd0	Floppy
/dev/hda1	Disco duro IDE 1
/dev/hda2	Disco duro IDE 2
/dev/sda1	Disco duro SCSI 1
/dev/cdrom	CD-ROM
/dev/lp	Impresora

3.2. El árbol de directorios

En los sistemas Unix, a diferencia del DOS o Windows, el árbol de directorios es único. Es decir, no existe un árbol por cada volumen o unidad de disco, sino que todos los volúmenes de datos se cuelgan de una estructura única. Al acto de unir un volumen a nuestro árbol, se le llama "montar" un volumen o unidad. Estas unidades pueden ser físicas (discos duros) o unidades y directorios de otros sistemas, exportados a través de la red.

La estructura del directorio raíz, en la cual no deberíamos añadir ningún directorio o carpeta, tiene los siguientes directorios entre otros:

/	Directorio raíz. Donde comienza el árbol de directorios
/bin	Archivos binarios, ejecutables esenciales
/boot	Ficheros de arranque (kernel), y otros ficheros de carga
/sbin	Archivos binarios, sólo ejecutables por el superusuario (root)
/dev	Ficheros controladores de dispositivos: "devices"
/etc	Archivos de configuración del sistema. Puede haber directorios dependiendo del programa que contenga los ficheros de configuración. Algunos programas guardarán sus ficheros de configuración en /etc o /usr/etc. También puede haber enlaces a estos ficheros
/etc/skel	Esqueleto, que recibe todo nuevo usuario que se cree.
/tmp	Archivos temporales
/etc/X11	Ficheros de configuración del sistema X11
/home	Directorio donde se situarán las carpetas de todos los usuarios
/usr	Directorio con ejecutables /usr/bin, /usr/sbin/ y con programas instalados por los usuarios o por el administrador /usr/local/
/var	Archivos de log, spool, correo,
/proc	Información acerca del estado del núcleo (kernel), dispositivos,
/lib	Bibliotecas compartidas (shared) o librerías dinámicas
/mnt	Directorio sobre el que se montarán los sistemas de ficheros.
/media	Directorio para montrar sistemas de ficheros de medios removibles
/opt	Aplicaciones opcionales (StarOffice, Zimbra)
/usr/X11R6	sistema X-Window, versión 11 distribución 6
/usr/include	Ficheros de cabecera para el compilador de C
/usr/lib	Bibliotecas de programas (librerías) y ejecutables que se necesitan para el funcionamiento de algunos programas
/usr/local	Programas que son locales a la máquina
/usr/share/man	Fichero con los manuales de ayuda

/usr/local/src	Código fuente para los programas instalados localmente
/usr/src/linux	Código fuente del núcleo
/var/lock	Directorio donde se crean los ficheros de bloqueo de los procesos. El formato utilizado es el de los ficheros de bloqueo HDU UUCP. Esto es, ficheros de bloqueo que contienen un PID como un número decimal ASCII
/var/log	Ficheros de registro "log" de todo tipo
/var/run	Ficheros de variables de ejecución, como los ficheros que contienen los identificadores de proceso (PIDs)
/var/spool	Ficheros en cola para varios programas
/var/spool/cups	Trabajos en cola para su impresión
/var/spool/mail	Buzones de correo de los usuarios

Alguien podría preguntar: Pero, qué sentido tiene esta maraña de directorios?. Primero, organizar el contenido del sistema por funciones afines. Segundo, poder controlar mejor el acceso a las diferentes partes del sistema dependiendo de los permisos del usuario o la aplicación. Además hay otras consideraciones como el rendimiento, y que las diferentes partes pueden estar montadas a través de la red o en particiones diferentes o...

Por ejemplo, podríamos tener el contenido de /usr/lib en una máquina, y que todas las máquinas de la red lo monten al inicio. Esta unidad estaría como de sólo lectura. Ahorraría espacio en todos los discos de la red, y el administrador solamente tendría que realizar una actualización para actualizar todas las máquinas.

3.3. Consolas virtuales

Por defecto tenemos 6 consolas virtuales. De la 1 a la 6 son en modo texto, y la 7 es la salida de display principal del sistema X. Para cambiar de una a otro pulsamos CTRL+ALT+Fx, siendo x el número de consola deseado.

3.4. Directorio de trabajo de los usuarios (home)

También llamado directorio home, es el directorio en el que normalmente trabaja un usuario, y en que se sitúa al entrar al sistema. En Linux se encuentra en /home/NOMBREDEUSUARIO Se crea automáticamente cuando el administrador crea una nueva cuenta de usuario. Sobre él tenemos todos los permisos del sistema y podremos crear, borrar y modificar ficheros y directorios. El directorio home del usuario se indica de modo resumido con el carácter tilde "~".

3.5. Directorio padre e hijo

El directorio padre es el directorio del que cuelga el directorio en el que estamos. Su

representación es "..". Un directorio hijo es el directorio que cuelga o pertenece a uno dado. Se representa como ".". Estando en cualquier directorio si hacemos referencia a . se trata del directorio actual.

3.6. Listados

Veamos un listado de un directorio: (ls -l)

```
drwxr-xr-x 2 johnka ikasleak 4096 2007-05-13 19:22 .
drwxr-xr-x 4 root root 4096 2007-05-13 19:22 ..
-rw-r--r-- 1 johnka ikasleak 220 2007-05-13 19:22 .bash_logout
-rw-r--r-- 1 johnka ikasleak 2298 2007-05-13 19:22 .bashrc
drwxr-xr-x 3 johnka ikasleak 4096 2007-05-13 19:22 Desktop
lrwxrwxrwx 1 johnka ikasleak 26 2007-05-13 19:22 Examples -> /usr/share/example-
content
-rw-r--r-- 1 johnka ikasleak 566 2007-05-13 19:22 .profile
```

Cada línea comienza con los permisos. Después tenemos un número que indica el tipo. Después están el propietario y el grupo al que pertenece el fichero o directorio. El tamaño en bytes. La fecha de última modificación y el nombre. Si el nombre comienza con un punto, es un fichero oculto.

Cuando los permisos comienzan con la letra "d", se trata de un directorio. Si es una "l" es un enlace. El guión "-" indica un fichero normal.

3.7. Sistema de permisos

Los permisos básicos en Linux (y Unix en general) son tres. Lectura, escritura y ejecución. Estos permisos se repiten para el dueño del fichero, el grupo al que pertenece el fichero, y al resto de los usuarios. Por lo tanto hay tres grupos de 3 permisos.

La estructura de la codificación de los permisos es la siguiente:

r	Fichero - Permiso de lectura sobre el fichero
	Directorio - Se puede ver los contenidos del directorio
w	Fichero - Permiso de escritura sobre el fichero o directorio
	Directorio - Se pueden añadir o borrar ficheros del directorio
x	Fichero - Permiso de ejecución sobre el fichero o directorio
	Directorio - Se puede listar información sobre los ficheros en ese directorio

Los permisos también se representan numéricamente. La representación es en hexadecimal, correspondiendo un carácter a cada grupo de permisos.

La lectura tiene valor 4, la escritura 2 y la ejecución 1.

rwxrwxrwx	->	0777
rw-rw	->	0660

3.7.1. Permisos especiales: SUID, SGID SB

En algunas ocasiones los usuarios que no tienen permisos especiales en el S.O. deben ejecutar programas que requieren ciertos privilegios. Un ejemplo de esto es el programa **passwd**. Dicho programa permite cambiar la contraseña de un usuario que está en el fichero **/etc/passwd**. En cambio la solución ideal no es dar permisos de escritura a los usuarios contra ese fichero ya que podrían modificar las contraseñas de todos los usuarios (no solo la suya).

De la misma forma, el programa **mail** tiene que poder escribir en la cola de mensajes de cualquier usuario y un usuario tampoco tiene acceso a dichos ficheros en modo escritura.

Para solucionar este tipo de problemas Unix permite a los programas que puedan tener algunos privilegios. Los programas que ejecutan estos otros programas pueden asumir otro UID o GID cuando se están ejecutando. A todo programa que cambia su UID se le llama SUID o set-UID. Un programa puede ser SUID y SGID al mismo tiempo.

Cuando un programa SUID está corriendo, su UID efectivo será el del propietario del fichero en lugar que el del usuario que lo está ejecutando. Si un programa es SUID o SGID, la salida del comando ls –l muestra una s en lugar de una x.

El bit SGID en un directorio controla la forma en que los grupos son asignados para los ficheros creados en el directorio.

Si el programa es **sticky** (válido sólo para directorios), los ficheros dentro del directorio pueden borrarse o renombrarse sólo por el propietario del fichero, el propietario del directorio, o por root. Esta opción se creo para que un usuario normal del sistema no pueda borrar ficheros de otros usuarios del directorio **/tmp**.

Los permisos para SUID, SGID y Sticky son los siguientes:

4000	SUID ó u=rws
2000	SGID ó g=rws
1000	Sticky ó o=rwt

3.8. Enlaces simbólicos

Los enlaces simbólicos son ficheros que apuntan a otros ficheros. Son similares (y muy diferentes) a los accesos directos de Windows. Hay dos tipos, pero el que más nos interesa a nivel de usuario es el software.

4. Comandos básicos

Cuan do se habla de los comandos de un sistema, hay que tener en cuenta que hay muchos que son para uso diario por parte de los usuarios, otros para el administrador, y muchos otros los podrán usar los usuarios normales, pero de una manera limitada. Esto es: Sólo pueden ver las configuraciones, pero no modificarlas. Además siempre hay que tener en cuenta los permisos de los recursos sobre los que se realizan los comandos.

ls	listar el contenido de un directorio
	-l contenido largo
cd xxx	cambiar al directorio xxx
ср хх уу	copiar el fichero xx en yy
	-r de forma recursiva
mv	mover o renombrar un fichero o directorio
	-r de forma recursiva
rm	borra un fichero
mkdir xx	crear un directorio con nombre xx
rmdir	borra un directorio
chmod	cambiar los permisos a un fichero o directorio
	ugo+x-> añade el permiso de ejecución para todoso-x-> quita el permiso de ejecución para el restog+w-> da el permiso de escritura al grupo0777-> da todos los permisos a todos0600-> sólo el dueño puede leer y escribir-R-> de forma recursiva (en directorios)
touch	crear o modificar la fecha de última modificación de un fichero
pwd	indica el directorio donde nos encontramos
man	páginas de manual. Todos los comandos, y la gran mayoría de ficheros de configuración tienen su página de manual. Por ejemplo man (man man).

4.1. Comodines

El asterisco "*" es un comodín. Este, sustituye a cualquier número de caracteres. El carácter de interrogación "?" sustituye a un carácter (el que sea).

4.2. Práctica

Hagamos un poco de práctica. Vamos a crear esta estructura de directorios:

```
~/|-Enero|-uno
| |-dos
| |-tres
|-Febrero|-uno
|-dos
|-tres
mkdir ~/Enero
mkdir ~/Enero/uno
mkdir ~/Enero/dos
mkdir ~/Enero/tres
mkdir -p ~/Febrero/uno
mkdir ~/Febrero/{dos,tres}
```

Ahora vamos a ir al directorio ~/Enero/tres

cd ~/Enero/tres

Y al ~/Febrero/dos

cd ../../Febrero/dos

miramos donde estamos

pwd

Creamos un fichero llamado **en-tres-1** en **Enero/tres/**

touch ~/Enero/tres/en-tres-1

Creamos otro llamado **en-tres-2** en el mismo directorio

cd ~/Enero/tres touch en-tres-2

Copiamos todo el directorio **Enero/tres** y su contenido en **Febrero/tres**

cp -r ~/Enero/tres ~/Febrero/tres/

Listamos el contenido de Febrero/tres

```
ls -l ~/Febrero/tres
```

Listamos el contenido de **Febrero/tres/tres/**

ls -l ~/Febrero/tres/tres

Cambiamos los permisos a Febrero/tres y todo su contenido a 0770

chmod -R 0770 ~/Febrero/tres

Cambiamos los permisos a todo el contenido de **Enero/tres** a 0660

chmod -R 0660 ~/Enero/tres/*

Borramos Enero/tres/en-tres-1

rm ~/Enero/tres/en-tres-1

Borramos todo el contenido de Febrero/tres/tres

rm ~/Febrero/tres/tres/*

Vamos a mirar lo que dice el manual de ls

man ls

4.3. Ejecutando como root

Hay labores que no puede realizar un usuario normal. Se requieren derechos de administrador. Hay dos formas de poder realizar estas labores. Una es poniéndonos como usuario root y a partir de ahí somos root, y la otra es ejecutar comandos como root.

El primer caso requiere del conocimiento de la clave de acceso de root. El segundo en cambio, nos permite ejecutar como root, programas a los que previamente root nos ha concedido el derecho. En este último caso, solamente necesitamos introducir nuestra clave (o no).

su

Nos ponemos como un usuario determinado. Normalmente será root, aunque a veces también nos interesan probar cosas como otros usuarios. Si es root quien ejecuta el comando para ponerse como otro usuario, no pide la contraseña. En el resto de los casos pide la contraseña del usuario que queremos ser. Si no ponemos el usuario, se supone que queremos ser root.

su xxxx	Ponernos como el usuario xxxx (Set User)
	- además obtenemos el entorno del usuario

Ejemplos:

su su johnka su — johnka

sudo

Ejecutamos como root los comandos permitidos en el fichero **/etc/sudoers**. El contenido del fichero puede ser:

root ALL=(ALL) ALL %admin ALL=(ALL) ALL

Nos indica que root y los miembros del grupo **admin**, pueden ejecutar todos los comandos en todas las máquinas como cualquier usuario. A los miembros del grupo **admin** les pedirá su clave de usuario.

sudo xxxx	Ejecutar como root, el comando xxxx
	•

5. Gestión de usuarios

Los usuarios están representados en el sistema, individualmente y por grupos. Los usuarios pueden ser a su vez usuarios físicos (personas) o usuarios que ejecutan ciertos procesos en el sistema. Lo mismo ocurre con los grupos.

La información de los usuarios están contenidos en el fichero **/etc/passwd**. Antiguamente, las claves de acceso también se encontraban en este fichero. Debido a que este fichero ha de ser legible por todo el mundo, las claves se ponen en la actualidad en el fichero **/etc/shadow.** En caso de necesidad, el comando **pwconv** convierte las entradas del fichero **passwd** al **shadow**. El fichero que contiene los grupos es **/etc/group**. Veamos el aspecto de las entradas del fichero **/etc/passwd**.

username	Nombre de la cuenta del usuario
password	Contraseña cifrada del usuario
	"*" es primer carácter, cuenta deshabilitada.
	"x", Contraseña en /etc/shadow
uid	Identificador de usuario
gid	Identificador del grupo primario
gecos	Datos del usuario (nombre completo, dirección,)
homedir	Directorio home del usuario
shell	Programa de consola que utiliza tras entrar al sistema

username:password:uid:gid:gecos:homedir:shell

El fichero de grupos nos permite organizar a los usuarios en grupos para facilitar la administración de permisos en el sistema. Un usuario tiene un grupo primario, pero puede a su vez pertenecer a otros grupos. Este es el contenido de una entrada en el fichero **/etc/group**. Al igual que el fichero de usuarios tiene otro para guardar las contraseñas, en algunos sistemas, el de grupos también dispone de un fichero para las contraseñas de grupo. Dicho fichero es **/etc/gshadow**.

groupname:password:gid:members

groupname	Nombre del grupo
password	Contraseña del grupo (apenas utilizado). Permite acceder al grupo con el
	comando newgrp

gid	Identificador del grupo
members	Lista de nombres de usuario que también son miembros del grupo. Se
	listan separados por comas, sin espacios, y sin coma al final.

Los comandos que utilizaremos para la gestión de usuarios y grupos:

useradd	Crea cuentas de usuario
	 -m -> Crea el directorio home -s -> Indica el shell a utilizar -c -> Comentario (gecos) -g -> Grupo primario -G -> Grupos adicionales -p -> Password (cifrado)
adduser	Crea cuentas de usuario
userdel	Borra cuentas de usuario
	-r -> Borra su directorio home
deluser	Borra cuentas de usuario
addgroup groupadd	Crea grupos
delgroup groupdel	Borra grupos
passwd	Cambia el password del usuario actual Añadiendo detrás un nombre de usuario , cambia su password (sólo root)

Ejemplos:

```
groupadd ikasleak
useradd -c "Jonh Karrantza" -g ikasleak -m -s /bin/bash johnka
userdel -r johnka
groupdel ikasleak
```

6. Programación de tareas

La ejecución de tares programadas en un sistema la realiza un demonio llamado **cron**. Las tareas a realizar por cron se encuentran en un fichero que se edita con el comando **crontab**. El contenido de un fichero de cron consta de distintas líneas correspondientes a distintas tareas. Cada línea tiene el siguiente aspecto:

minutos	Especifica los minutos (0 - 59)
hora	Especifica la hora (0 - 23)
día	Especifica el día del mes (1 - 31)
mes	Especifica el mes (1 - 12)
dia-semana	Especifica el día de la semana (0 – 6). El 0 es el domingo.
comando	El comando a ejecutar

minutos hora día mes día-semana comando

Ejemplos:

30 23 * * * /root/bin/hartu-antibirusa
07 02 * * 1,2,3,4,5 /root/bin/sys-segkopi-diff

Para editar el fichero de cron usaremos el comando crontab.

crontab -u root -e

Tenemos que tener en cuenta que este comando lanza (por defecto) el editor vi. Para pasar al modo edición debemos de pulsar "i". Para ponernos en modo comando, "ESC". Para guardar, estando en modo comando, "!w". Para salir, estando en modo comando, "!x".

7. Arranque y parada del sistema y niveles de ejecución

El arranque se realiza automáticamente al iniciar el ordenador, pero podemos hacerlo de diversas formas, en función de varios parámetros. Para ello tenemos unos procesos, y unos ficheros asociados en los que configuraremos su comportamiento

7.1. Proceso de arranque

Después de que los drivers se han cargado en el arranque del sistema, el kernel ejecuta el programa **init** (**/sbin/init**). **init** es un programa de propósito general que lanza procesos y reinicia ciertos programas cuando estos finalizan su ejecución. Por ejemplo, cada consola virtual tiene un proceso mingetty asociado y que ha sido lanzado por init. Cuando un usuario finaliza su sesión de una consola, el proceso mingetty finaliza e init arranca uno nuevo.

init también es el responsable de ejecutar varios programas y scripts cuando arranca el sistema. Todo lo que init hace está controlado por el fichero **/etc/inittab**. Cada línea de este fichero tiene el siguiente formato:

Etiqueta de uno o dos caracteres que se usan para identificar esta entrada en el fichero
Lista de runlevels en donde una entrada se ejecuta. Un runlevel es un numero que indica el nivel de ejecución del sistema. Por ejemplo, cuando el runlevel es de 3, todas las entradas del fichero /etc/inittab de ese nivel se ejecutan. En el runlevel 1 se ejecutan sólo los scripts de configuración mínima del sistema, en el runlevel 2 se ejecuta cada cosa del runlevel 1 más la configuración de red, en el runlevel 3 se ejecuta todo lo del 1 y 2 y el proceso de login.
Le indica a init si tiene que ejecutar un mandato una sola vez o reiniciarlo una vez que finalice su ejecución
 wait -> El proceso será iniciado cuando se entra en el runlevel e init espera a que termine respam -> El proceso se reinicia de nuevo cuando termina

${\bf code:} {\bf runlevels:} {\bf action:} {\bf command}$

command El comando a ejecutar cuando sucede la acción

El runlevel por defecto en Debian es el 2 En RH y SUSE, con sistema gráfico, el runlevel por defecto es el 5. Sin sistema gráfico instalado, es el 3. El runlevel 5 arranca automáticamente el sistema gráfico X-Window.

En Debian, el runlevel por defecto está indicado en el fichero /**etc/inittab** en una línea tal como la siguiente:

id:2:initdefault

7.2. Niveles de ejecución

Nivel	Debian	RH y SUSE
0	Parada del sistema	Parada del sistema
1	Modo monousuario y sin red	Modo monousuario y sin red
2	Modo multiusuario con red y X11	Modo multiusuario
3	Modo multiusuario con red y X11	Modo multiusuario con red
4	Modo multiusuario con red y X11	No se usa
5	Modo multiusuario con red y X11	Modo multiusuario con red y X11
6	Reinicio	Reinicio

El siguiente mandato arranca el sistema en modo monousuario (runlevel 1).

init 1

Si salimos de la sesión con exit o CONTROL D volvemos al modo multiusuario o modo 2 (Debian).

7.3. Sysinit

Esta entrada en el inittab: **si::sysinit:/etc/init.d/rcS** le indica a init que ejecute el script **/etc/ init.d/rcS** cada vez que el sistema arranca. Este script ejecuta **/etc/init.d/rc** con el parámetro **S**, lo que le indica que active todos los servicios en ese runlevel.

Además contiene comandos que manejan el inicio del sistema como la activación del swap, monta los sistemas de archivos, activa el reloj del sistema según la CMOS,

7.4. Scrips de inicio y parada de servicios

Linux almacena los comandos de arranque en ficheros con el nombre de **rc**. En Debian, el

principal script es el **/etc/init.d/rc**. Cada runlevel del fichero inittab llama al fichero rc y este ejecuta según el runlevel diferentes scripts. Para cada runlevel los scripts están en el directorio **/etc/rcN.d** (En RH /etc/rc.d/rcN.d y en SUSE /etc/init.d/rcN.d) en donde **N** es el **runlevel** arrancado. Así en el modo multiusuario o modo normal, los scripts que se ejecutan están el el directorio /etc/rc2.d. Si miramos en este directorio veremos que hay diferentes ficheros con el nombre Snnxxx y Knnxxx en donde nn es un número de 00 a 99 y xxx es el nombre de algún servicio. Aquí vemos unos de estos ficheros.

```
15 2007-04-23 23:28 S11klogd -> ../init.d/klogd
lrwxrwxrwx
            1 root root
                          14 2007-04-23 23:28 S12dbus -> ../init.d/dbus
lrwxrwxrwx
            1 root root
                          13 2007-04-23 23:28 S13gdm -> ../init.d/gdm
lrwxrwxrwx
            1 root root
lrwxrwxrwx 1 root root
                          16 2007-04-23 23:28 S19cupsys -> ../init.d/cupsys
lrwxrwxrwx
            1 root root
                          15 2007-04-23 23:28 S19hplip -> ../init.d/hplip
                          14 2007-04-23 23:28 S20apmd -> ../init.d/apmd
lrwxrwxrwx 1 root root
                          16 2007-04-23 23:28 S20apport -> ../init.d/apport
            1 root root
lrwxrwxrwx
                          22 2007-04-23 23:28 S20hotkey-setup -> ../init.d/hotkey-
            1 root root
lrwxrwxrwx
setup
                          17 2007-04-23 23:28 S20makedev -> ../init.d/makedev
lrwxrwxrwx
            1 root root
                          23 2007-04-23 23:28 S20nvidia-kernel ->
lrwxrwxrwx
            1 root root
../init.d/nvidia-kernel
lrwxrwxrwx
            1 root root
                          19 2007-04-23 23:28 S20powernowd -> ../init.d/powernowd
                          15 2007-04-23 23:28 S20rsync -> ../init.d/rsync
lrwxrwxrwx
            1 root root
                          13 2007-04-23 22:59 S23ntp -> ../init.d/ntp
lrwxrwxrwx 1 root root
                          19 2007-04-23 23:28 S25bluetooth -> ../init.d/bluetooth
lrwxrwxrwx 1 root root
lrwxrwxrwx 1 root root
                          18 2007-04-29 22:35 S40arpalert -> ../init.d/arpalert
            1 root root
                          17 2007-04-23 23:28 S89anacron -> ../init.d/anacron
lrwxrwxrwx
                          13 2007-04-23 23:28 S89atd -> ../init.d/atd
lrwxrwxrwx 1 root root
                          14 2007-04-23 23:28 S89cron -> ../init.d/cron
lrwxrwxrwx 1 root root
                          24 2007-04-23 23:28 S90binfmt-support ->
lrwxrwxrwx
            1 root root
../init.d/binfmt-support
            1 root root
lrwxrwxrwx
                          17 2007-04-23 23:28 S98usplash -> ../init.d/usplash
```

Los scripts cuyo nombre comienza por **K** se ejecutan por **/etc/init.d/rc** para "matar" (**kill**) los servicios que están corriendo en el sistema y los scripts cuyo nombre comienza por **S** son los servicios que **/etc/init.d/rc** arrancará (**start**). El orden de cómo se paran o arrancar tales scripts los impone el número nn del nombre de los ficheros.

Debido a que los mismos servicios se arrancan y detienen en diferentes runlevels, se usan enlaces simbólicos (soft links) para referenciar a los scripts desde diferentes lugares.

Como veremos del listado anterior, los scripts de parada y arranque de los diferentes servicios instalados están en /etc/init.d. También podemos arrancar, parar y reiniciar a mano estos servicios:

Para arrancar el servicio de servidor de nombres:

/etc/init.d/named start

Para parar el servicio de servidor de nombres:

/etc/rc.d/init.d/named stop

Para reiniciar el servicio:

/etc/rc.d/init.d/named restart

7.5. Modo monousuario

Se trata de un modo especial en el que no existe un prompt de login. Según arranca el sistema pasamos a ser root. Este modo es importante para ciertas tareas de administración como puede ser el chequeo de un sistema de archivos corrupto. En modo monousuario los sistemas de archivos no se montan. Podemos ejecutar varios programas en multitarea pero no se puede ejecutar X-Window.

Para activar el modo monousuario lo podemos activar así:

- Con el sistema corriendo ejecutando el mandato : init 1
- Cuando arranca el sistema en el prompt de GRUB: single
- Parando el sistema con el mandato **shutdown** sin indicar ni la opción –h o –r.

7.6. Parada del sistema

Para realizar la parada, tenemos el comando **shutdown**.

Este comando soporta varios parámetros.

-h	detener
-r	reiniciar
now	ahora
+5	dentro de 5 minutos
14:00	a las 14:00 horas
"Abajo el sistema"	Envía el mensaje a todos los usuarios conectados. Ojo, "conectados" significa logueados en el sistema (es un sistema multiusuario). No utilizando servicios de sus servidores.

Enlaces simbólicos a shutdown:

halt /sbin/shtdown -h now reboot /sbin/shtdown -r now

También podemos, directamente, ir al runlevel 0.

init O

8. Procesos

Un proceso es un programa en ejecución, no el programa en si mismo. Siendo LINUX un sistema operativo multiproceso, está claro que entre sus capacidades debe estar necesariamente la de gestionar procesos.

En un sistema multiusuario, éste conjunto de procesos son gestionados por el núcleo del S.O. basándose en una tabla con informaciones variadas que abarcan, desde el estado o prioridad de proceso hasta su estado (existen varios), de forma que se procuren evitar situaciones de conflicto entre varios procesos por su localización en RAM o por su acceso simultaneo a algún periférico.

8.1. Genealogía de procesos

La terminología hace referencia al hecho de que un proceso (padre) que está ocupando tiempo CPU, puede lanzar otro proceso (su hijo) que va a estar coexistiendo en la ocupación de la CPU con él, existiendo por ende la posibilidad de que un proceso determinado no lance un solo proceso hijo, sino varios procesos hijo que a su vez pueden dar lugar a otros procesos (nietos del padre).

El SHELL del usuario (el que interpreta las órdenes del usuario) es un proceso asociado con el terminal en que se ha conectado el sistema, los comandos que el usuario ejecuta durante su sesión son procesos hijo del SHELL.

Antes de que un usuario se conecte vía login, el terminal es controlado por un proceso llamado getty o mingetty (Linux). Después de que nos conectamos, el proceso getty muere (otro nuevo será lanzado por el kernel cuando nos desconectemos de nuestra sesión) y nuestro terminal es gestionado por el shell, el cuál es un proceso diferente. El shell crea entonces un proceso nuevo cada vez que introducimos un comando. La creación de un nuevo proceso se denomina forking, porque un proceso se parte en dos.

Si por ejemplo lanzamos un editor, este nuevo proceso es hijo del SHELL y si desde el editor llamamos a un comando SHELL, tenemos un proceso SHELL nieto del SHELL original.

El sistema por otra parte tiene permanentemente una tabla de procesos con una información referente a cada uno de los que se están ejecutando en ese momento dado, se trata de información referente tanto el estado como al identificador del proceso (número con que el S.O. identifica al citado proceso). Estas informaciones son interesantes conocerlas, dado que en ocasiones pudiera interesarnos disponer de ellas de cara a neutralizar procesos o cambiarles el estado o prioridad de ejecución, en caso de que ello fuera necesario, a fin de optimizar el rendimiento del Sistema.

8.2. Control de procesos

Vamos a tomar contacto con todos aquellos comandos que nos pueden permitir como usuarios un control relativo de los procesos en Linux.

ps	Ver el estado de los procesos del terminal actual		
	 a -> Los procesos de todos los terminales ax -> Todos los procesos de todos los terminales 		
kill [-n] yyyy	Envia la señal n al proceso con pid yyyy		
	 -1 -> Reiniciar el proceso -9 -> Matar el proceso -15 -> Terminar el proceso Por defecto, -15 		
killall xxxx	Terminar todos los procesos de nombre xxxx		
	Acepta las señales de kill		

Ejemplo: (En una máquina con Ubuntu)

```
alfredo@SAZE-UBPC01:~$ ps ax
```

PID	TTY	STAT	TIME	COMMAND
1	?	Ss	0:02	/sbin/init
2	?	SN	0:00	[ksoftirqd/0]
3	?	S	0:00	[watchdog/0]
4	?	S<	0:00	[events/0]
5	?	S<	0:00	[khelper]
• • • • •				
132	?	S<	0:00	[aio/0]
1999	?	S<	0:01	[ata/0]
2000	?	S<	0:00	[ata_aux]
2001	?	S<	0:00	[scsi_eh_0]
2002	?	S<	0:08	[scsi_eh_1]
2004	?	S<	0:00	[ksuspend_usbd]
2005	?	S<	0:00	[khubd]
2046	?	S	0:00	/bin/sh /usr/lib/openoffice/program/soffice -writer f
2061	?	Sl	0:15	/usr/lib/openoffice/program/soffice.bin -writer file:
2250	?	Sl	0:00	gnome-terminal
2252	?	S	0:00	gnome-pty-helper
2253	pts/0	Ss	0:00	bash
2275	pts/0	R+	0:00	ps ax
2385	?	D<	0:05	[kjournald]
2592	?	S <s< td=""><td>0:00</td><td>/sbin/udevddaemon</td></s<>	0:00	/sbin/udevddaemon
3439	?	S<	0:00	[kpsmoused]

Linux: Instalación y administración básica Iurreta GLHB Institutua – Olaburu 19 IURRETA (BIZKAIA) – Tel: 944 66 88 00

3504	?	S<	0:00	[kgameportd]
4114	?	S<	0:00	[kjournald]
4124	?	S<	0:00	[kjournald]
4349	?	Ss	0:00	/sbin/portmap
4489	tty4	Ss+	0:00	/sbin/getty 38400 tty4
4490	tty5	Ss+	0:00	/sbin/getty 38400 tty5
4493	tty2	Ss+	0:00	/sbin/getty 38400 tty2
4495	tty3	Ss+	0:00	/sbin/getty 38400 tty3
4496	tty1	Ss+	0:00	/sbin/getty 38400 tty1
• • • •	•			
4969	?	S	0:00	hald-addon-keyboard: listening on /dev/input/event4
4972	?	S	0:00	hald-addon-keyboard: listening on /dev/input/event5
4975	?	S	0:00	hald-addon-keyboard: listening on /dev/input/event6
4995	?	S	0:10	hald-addon-storage: polling /dev/scd0
4999	?	S	0:00	hald-addon-storage: polling /dev/scd1
5016	?	Ss	0:00	/usr/sbin/dhcdbdsystem
5031	?	Ssl	0:00	/usr/sbin/NetworkManagerpid-file /var/run/NetworkM
5049	?	Ss	0:00	avahi-daemon: registering [SAZE-UBPC01.local]
5050	?	Ss	0:00	avahi-daemon: chroot helper
5066	?	Ss	0:00	/usr/sbin/NetworkManagerDispatcherpid-file /var/ru
5079	?	Ss	0:00	/usr/bin/system-tools-backends
5080	• ?	S	0:00	dbus-daemonsessionprint-addressnofork
5126	• ?	Ss	0:00	/usr/sbin/gdm
5127	• ?	S	0:00	/usr/sbin/gdm
5132	• ++•7	Set	2.17	/usr/X11R6/bin/X ·0 _br _audit 0 _auth /war/lib/gdm/.
5137	2017	S	0.00	/sbin/dbclient =1 =1f /var/lib/dbcp3/dbclient.eth0.le
5231	•	5	6.45	/usr/sbin/hpiod
5230	•	23 C	0.10	nython /usr/sbin/hpssd
5255	·	5	0.00	pychon /usi/sbin/hpssu
5486	• ?	Ss	0.00	/usr/shin/ntnd _n /war/run/ntnd.nid _u 112:119 _g
5503	•	55 59	0.00	/usr/sbin/heid _y _s
5518	•	55	0.00	[krfcommd]
5534	•	5 c	0.00	<pre>/shin/mdadmmonitornid_file /war/run/mdadm/monit</pre>
55/0	•	25 C	0.00	/usr/sbin/arpalert_d_f_/etg/arpalert/arpalert_conf
5590	•	5	0.00	/usr/sbin/arparett -u -1 /etc/arparett/arparett.com
5500	: 2	22	0.00	/usr/sbin/acu
5021	•	55 C	0.00	/usr/sbin/cioh
2021	: 2	с с	0.00	/usr/sbin/winbindd
5032	•	G	0:00	/ USI / SDIII / WIIDIIIUU
5655	2	C	0.00	(ucr / chin / winhindd
EODE	?	S	0:00	/usr/sbin/winbindd
5835	? ?	S Ss	0:00	<pre>/usr/sbin/winbindd /usr/bin/smbmount //zir014950a/alfredo\$ /home/IURRETA [ambied]</pre>
5835 5844	? ? ?	S Ss S<	0:00 0:00 0:20	<pre>/usr/sbin/winbindd /usr/bin/smbmount //zir014950a/alfredo\$ /home/IURRETA [smbiod] /usr/bin/smbmount //zir014050a/taldaab\$ /home/IURDETA</pre>
5835 5844 5847	? ? ?	S Ss S< Ss	0:00 0:00 0:20 0:00	<pre>/usr/sbin/winbindd /usr/bin/smbmount //zir014950a/alfredo\$ /home/IURRETA [smbiod] /usr/bin/smbmount //zir014950a/taldeak\$ /home/IURRETA /usr/bin/smbmount //zir014950a/taldeak\$ /home/IURRETA</pre>
5835 5844 5847 5850	? ? ? ?	S Ss S< Ss Ss	0:00 0:00 0:20 0:00 0:00	<pre>/usr/sbin/winbindd /usr/bin/smbmount //zir014950a/alfredo\$ /home/IURRETA [smbiod] /usr/bin/smbmount //zir014950a/taldeak\$ /home/IURRETA /usr/bin/smbmount //zir014950a/eskola\$ /home/IURRETA/</pre>
5835 5844 5847 5850	? ? ?	S Ss Ss Ss Ss	0:00 0:00 0:20 0:00 0:00	<pre>/usr/sbin/winbindd /usr/bin/smbmount //zir014950a/alfredo\$ /home/IURRETA [smbiod] /usr/bin/smbmount //zir014950a/taldeak\$ /home/IURRETA /usr/bin/smbmount //zir014950a/eskola\$ /home/IURRETA/</pre>
5835 5844 5847 5850 5868	? ? ? ?	S SS SS SS SS	0:00 0:00 0:20 0:00 0:00 0:00	<pre>/usr/sbin/winbindd /usr/bin/smbmount //zir014950a/alfredo\$ /home/IURRETA [smbiod] /usr/bin/smbmount //zir014950a/taldeak\$ /home/IURRETA /usr/bin/smbmount //zir014950a/eskola\$ /home/IURRETA/ /usr/bin/smbmount //oiz/soft /home/IURRETA/alfredo/OI</pre>
5835 5844 5847 5850 5868 5871	? ? ? ?	S Ss Ss Ss Ss Ssl	0:00 0:00 0:20 0:00 0:00 0:00 0:00	<pre>/usr/sbin/winbindd /usr/bin/smbmount //zir014950a/alfredo\$ /home/IURRETA [smbiod] /usr/bin/smbmount //zir014950a/taldeak\$ /home/IURRETA /usr/bin/smbmount //zir014950a/eskola\$ /home/IURRETA/ /usr/bin/smbmount //oiz/soft /home/IURRETA/alfredo/OI x-session-manager /usr/chin/anasha2 k start</pre>
5835 5844 5847 5850 5868 5871 5913	? ? ? ? ? ? ? ?	S Ss Ss Ss Ss Ssl SNS	0:00 0:00 0:20 0:00 0:00 0:00 0:00 0:00	<pre>/usr/sbin/winbindd /usr/bin/smbmount //zir014950a/alfredo\$ /home/IURRETA [smbiod] /usr/bin/smbmount //zir014950a/taldeak\$ /home/IURRETA /usr/bin/smbmount //zir014950a/eskola\$ /home/IURRETA/ /usr/bin/smbmount //oiz/soft /home/IURRETA/alfredo/OI x-session-manager /usr/sbin/apache2 -k start /usr/chin/smbmouh2 -k start</pre>
5835 5844 5847 5850 5868 5871 5913 5916	? ? ? ? ? ? ? ? ? ? ?	S Ss Ss Ss Ss Ssl SNS SN	0:00 0:00 0:20 0:00 0:00 0:00 0:00 0:00	<pre>/usr/sbin/winbindd /usr/bin/smbmount //zir014950a/alfredo\$ /home/IURRETA [smbiod] /usr/bin/smbmount //zir014950a/taldeak\$ /home/IURRETA /usr/bin/smbmount //zir014950a/eskola\$ /home/IURRETA/ /usr/bin/smbmount //oiz/soft /home/IURRETA/alfredo/OI x-session-manager /usr/sbin/apache2 -k start /usr/sbin/apache2 -k start /usr/sbin/apache2 -k start</pre>
5835 5844 5847 5850 5868 5871 5913 5916 5917	? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?	S Ss Ss Ss Ss Ssl SNS SN SN	0:00 0:00 0:20 0:00 0:00 0:00 0:00 0:00	<pre>/usr/sbin/winbindd /usr/bin/smbmount //zir014950a/alfredo\$ /home/IURRETA [smbiod] /usr/bin/smbmount //zir014950a/taldeak\$ /home/IURRETA /usr/bin/smbmount //zir014950a/eskola\$ /home/IURRETA/ /usr/bin/smbmount //oiz/soft /home/IURRETA/alfredo/OI x-session-manager /usr/sbin/apache2 -k start /usr/sbin/apache2 -k start /usr/sbin/apache2 -k start /usr/sbin/apache2 -k start</pre>
5835 5844 5847 5850 5868 5871 5913 5916 5917 5918	· · · · · · · · · · · · · · · · · · ·	S Ss Ss Ss Ss Ss Ss Sn Sn Sn Sn Sn	0:00 0:00 0:20 0:00 0:00 0:00 0:00 0:00	<pre>/usr/sbin/winbindd /usr/bin/smbmount //zir014950a/alfredo\$ /home/IURRETA [smbiod] /usr/bin/smbmount //zir014950a/taldeak\$ /home/IURRETA /usr/bin/smbmount //zir014950a/eskola\$ /home/IURRETA/ /usr/bin/smbmount //oiz/soft /home/IURRETA/alfredo/OI x-session-manager /usr/sbin/apache2 -k start /usr/sbin/apache2 -k start /usr/sbin/apache2 -k start /usr/sbin/apache2 -k start /usr/sbin/apache2 -k start</pre>

5921	?	SN	0:00	/usr/sbin/apache2 -k start
5923	?	Ss	0:00	/usr/bin/ssh-agent /usr/bin/dbus-launchexit-with-s
5934	?	S	0:00	/usr/bin/dbus-launchexit-with-session x-session-ma
5951	?	Ss	0:00	/usr/bin/dbus-daemonforkprint-pid 4print-add
5958	?	SNs	0:00	/usr/sbin/cupsd
5960	?	S	0:01	/usr/lib/libgconf2-4/gconfd-2 8
5972	?	S	0:00	/usr/bin/gnome-keyring-daemon
5974	?	Sl	0:03	/usr/lib/control-center/gnome-settings-daemon
5981	?	Ss	0:00	/bin/sh -c /usr/bin/esd -terminate -nobeeps -as 1 -sp
5982	?	S	0:00	/usr/bin/esd -terminate -nobeeps -as 1 -spawnfd 25
5986	?	S	0:10	/usr/bin/metacitysm-client-id=default0
5989	?	S	0:09	gnome-panelsm-client-id default1
5995	?	S	0:36	nautilusno-default-windowsm-client-id default2
5998	?	Ssl	0:00	/usr/lib/bonobo-activation/bonobo-activation-server -
6000	?	Ss	0:00	gnome-volume-managersm-client-id default4
6002	?	S	0:00	/usr/lib/gnome-vfs-2.0/gnome-vfs-daemon
6007	?	Sl	6 : 07	<pre>beagled /usr/lib/beagle/BeagleDaemon.exereplace</pre>
6009	?	Sl	0:02	<pre>beagle-searchdebug /usr/lib/beagle/Search.exeic</pre>
6017	?	Sl	0:00	/usr/lib/evolution/2.10/evolution-alarm-notify
6019	?	S	0:00	nm-appletsm-disable
6020	?	S	0:13	<pre>gnome-cups-iconsm-client-id default3</pre>
6021	?	Ss	0:00	gnome-power-manager
6081	?	Sl	0:00	/usr/lib/evolution/2.10/evolution-exchange-storage
6086	?	Sl	0:00	/usr/lib/evolution/evolution-data-server-1.10oaf-a
6091	?	S	0:00	/usr/lib/gnome-applets/trashappletoaf-activate-iid
6132	?	S	0:00	/usr/lib/nautilus-cd-burner/mapping-daemon
6157	?	S	0:00	<pre>/usr/lib/gnome-applets/mixer_applet2oaf-activate-i</pre>
6165	?	Ss	0:09	gnome-screensaver
6176	?	Sl	0:26	/usr/lib/firefox/firefox-bin
6305	?	S<	0:00	[scsi_eh_2]
6306	?	S<	0:00	[usb-storage]
6331	?	S	0:00	hald-addon-storage: polling /dev/sdc
20348	?	SNl	8:28	<pre>beagled-helper /usr/lib/beagle/IndexHelper.exe</pre>
alfred	do@SAZE-UI	BPC01:~\$	5	

Donde la información que aparece se desglosa de la siguiente manera:

PID	Identificador de proceso o número con el que el sistema identifica a un proceso en la tabla de procesos
TTY	Indica el terminal desde el cual ha sido lanzado el proceso
STAT	Estatus del proceso
TIME	Es el tiempo de ocupación de la CPU que ocupa un proceso. Suele estar especificado en minutos y segundos
CMD	Es la orden que se escribe para lanzar el proceso

Se pueden especificar los parámetros al estilo BSD o Linux. Producen resultados diferentes. Probarlo.

8.3. Matando procesos

Una cuestión que surge es la de cómo pode eliminar un determinado proceso una vez lanzado; bien porque produce problemas o bien por que se demora demasiado tiempo su ejecución. En tal sentido LINUX proporciona el comando **kill**. Su sintaxis es la siguiente:

kill identificador de proceso (PID)

Aún así, un determinado proceso lanzado puede haber lanzado procesos secundarios y cuesta un poco neutralizarlo, en este sentido se puede mejorar la actuación de **kill** especificando a continuación un número como por ejemplo:

kill -9 "identificador del comando/proceso"

Ese número constituye una de las señales con las que LINUX "indica" a los procesos para que finalicen. Usualmente estas señales van desde la 1 a la 31, siendo la de servicio más frecuente la número 9 y además no evitable, es decir, el proceso se parará de forma garantizada. Si por alguna razón un proceso no responde a nuestros requerimientos, esta señal nos garantiza que el proceso terminará.

Hay otras señales que se usan habitualmente como el -1 (o HUP). Esta señal le indica al proceso que se reinicie.

No vamos a ver aquí la lista completa de las señales de Unix que se pueden enviar a un proceso. Muchas de ellas sólo tienen sentido si es el sistema quien las envía. Vamos ver las que tienen utilidad de cara a un usuario o administrador.

HUP	1	HangUp. reinicia el proceso, leyendo nuevamente la configuración.
KILL	9	Kill with extreme prejudice. Mata el proceso.
TERM	15	Terminate. Indicamos al proceso de que termine.

Indicar que podemos indicar el número de la señal y también su nombre:

kill -HUP idenntificadordelproceso killall -HUP nombredelproceso

8.4. Control de tareas en Linux

Ya hemos visto como trabajar a un nivel elemental con procesos Sin embargo, hemos observado que en algunas situaciones el resultado, se envía por defecto a ciertos ficheros, lo cual puede ser indeseable. En tal sentido pudiera ser interesante que el resultado se desviara a otro fichero. Esto puede conseguirse con el conocimiento de lo que se conoce como redirecciones de E/S estándar. Una capacidad que suministra el SHELL de LINUX y que es especialmente potente.

&	Ejecuta en comando en background (segundo plano)	Program &
^ _Z	Parar el proceso en primer plano (pasa a	^z
	segundo plano)	[1]+ Stopped man killall
jobs	Lista procesos en segundo plano	jobs
		[1]- Stopped man killall
		[2]+ Stopped man kill
%n	Pasa a primer plano el proceso en bg número	%1
	n	Pasa el proceso 1 en bg, a fg
%? xxxx	Pasa a primer plano el proceso en bg xxxx	%? man
bg	Arranca un proceso en segundo plano	bg man kill
fg	Pasa a primer plano, un proceso en segundo plano	fg o fg %n
kill %n	Termina un proceso en bg (queda en bg)	kill %1

Esta capacidad está relacionada con el control del flujo de datos creado por los procesos. Podemos variar el comportamiento de los procesos con una serie de comandos.

En Linux podemos ejecutar varios comandos de forma secuencial separando con ; (punto y coma) los distintos comandos:

man man; cd /; ls -l; cd ~

También podemos ejecutar las órdenes de manera condicional. Si separamos las órdenes con el carácter && (condición AND), la segunda orden se ejecutará sólo si la primera se ha ejecutado correctamente:

ls -l file.txt && cat file.txt

Usando el carácter || (condición OR) la orden de la derecha se ejecutará sólo si la primera no se ha ejecutado correctamente:

ls docs || mkdir docs

8.5. Redirección de E/S

En LINUX, como en casi todos los Sistemas Operativos actuales, existen tres ficheros de especial importancia. Se trata de los ficheros **Entrada Estándar**, **Salida Estándar** y **Salida de error Estándar**. Se trata de ficheros asociados a determinados dispositivos con el fin de que la información (datos) suministrados por la ejecución de un proceso se envíen por defecto a un determinado dispositivo.

El fichero de salida estándar suele estar asociado con el dispositivo pantalla o terminal, mientras que el fichero entrada estándar está asociado con el dispositivo físico teclado y

el fichero de salida de errores también suele coincidir con la pantalla / terminal.

Pues bien, el Shell dota de la capacidad de reorientar ese flujo de datos que de forma habitual van hacia uno de los ficheros de E/S estándar Para tal fin existen los tres caracteres especiales siguientes: "<", ">" y ">>".

Metacaracteres que permiten cualquier posibilidad de reorientación del flujo de datos generados por los procesos lanzados. No solo en el sentido estricto de redirección de E/S estándar sino en su visión más amplia de redirección de/hacia cualquier fichero o dispositivo como veremos.

El carácter ">" redirecciona la salida de un comando a un fichero sobreescribiendo su contenido si existe el fichero. Si no existe, lo crea.

```
echo "Kaixo" > proba.txt
echo " muneka">> proba.txt
cat proba.txt
Kaixo
muneka
```

El metacarácter ">>" redirecciona la salida a un fichero, añadiendo la información al final del fichero.

El carácter "<" permite redireccionar la entrada a un comando.

comando < fichero

Creamos un fichero llamado entrada con el siguiente contenido:

ls —l

Después ejecutamos el siguiente comando:

ls < entrada

Veremos que se ejecuta el comando ls -l.

8.5.1. Redirección de la salida y de los errores

Si ejecutamos el siguiente comando:

```
date > log 2>&1
```

La salida del comando date (la fecha y la hora del sistema) se redirecciona al fichero log y la salida de errores (2) se redirecciona a la salida estándar (&1).

Si hacemos un cat del fichero log veremos la salida en el fichero log. Comprobamos la diferencia si ejecutamos este mandato:

date > log

El fichero log sigue teniendo la salida del mandato date. Sin embargo, si provocamos un error:

date popopo > log

En el fichero vemos la salida del error al dispositivo de salida standard (la pantalla):

date: invalid date `popopo`

Sin embargo, si ejecutamos este mandato:

date rarara > log 2>&1

Veremos que en el fichero log aparece también la salida de errores que se envía a la salida Estándar: 2>&1

Si ponemos 2>1 el 1 lo toma como el nombre del fichero de salida (por eso se pone el carácter &).

Existe un dispositivo para la redirección de la salida de los mandatos: el dispositivo nulo o **null (/dev/null**).

Supongamos que quiero ejecutar un mandato pero que la salida de mensajes no aparezca en la pantalla:

ls -l > /dev/null

Este comando no tiene sentido ya que el listado de ficheros lo redirecciona al dispositivo null. Es decir, a ninguna parte.

Pero puede ser útil para algunos mandatos. Supongamos que quiero estar ejecutando un mandato ping. Simplemente quiero ejecutar la orden en segundo plano pero no quiero que sus mensajes aparezcan en pantalla:

ping 212.55.31.3 > /dev/null &

Si hacemos un ps –eax veremos que está ejecutándose el comando pero no vemos nada en pantalla.

8.6. Intercomunicación de procesos: PIPES (tuberías)

De entre todos los métodos de intercomunicación de procesas en un sistema Linux, hay uno especialmente interesante para usuarios no programadores. El sistema de PIPES (tuberías). La línea de comunicación (pipeline) se representa por el metacarácter "|". El nombre, cuya traducción es tubería, hace merecida referencia a su significado. De una manera más simple, podríamos explicar la tubería como el hecho de hacer que un proceso envíe los resultados de su ejecución a otro para que este actué sobre esos resultados; es decir, la actuación del segundo proceso es sobre los resultados del primero. El fenómeno lleva implícita una redirección de E/S. Probar con el siguiente ejemplo:

ls -l /etc/ | more

9. Sistema de registro (log)

Los logs del sistema son uno de los principales puntos a tener en cuenta por todo administrador de un sistema. Ellos nos dicen los mensajes del kernel sobre avisos (**warnings**), errores, alertas, y de otros demonios (**daemons**) que corren en nuestro sistema (correo, telnet, ftp, web,...).

El daemon syslogd es el encargado de registrar todas las entradas de logs del sistema. Es lanzado por el script **/etc/init.d/sysklogd** al arrancar el sistema.

9.1. Configuración de Logs

El fichero de configuración es el /etc/syslog.conf.

1 Su formato es el siguiente:

facility.severity fichero-de-log

La primera parte de cada línea de este fichero es la clase de mensajes que queremos registrar en los ficheros de log. El segundo campo indica en qué fichero registraremos el log.

El campo facility puede contener las siguientes cadenas:

mail	Avisos del sistema de correo
kern	Mensajes del kernel
user	Mensajes de programas de usuario
auth	Mensajes del sistema de autentificación con programas como login o su
lpr	Subsistema de impresión
daemon	Procesos servidor del sistema
*	Indica todos los tipos

El campo severity puede contener las siguientes cadenas por orden de severidad:

debug	Información de depuración de programas
info	Mensajes informativos

Linux: Instalación y administración básica Iurreta GLHB Institutua – Olaburu 19 IURRETA (BIZKAIA) – Tel: 944 66 88 00

notice	Mensajes no críticos
warning	Avisos
err	Errores
crit	Errores críticos como fallos de disco duro
alert	Error serio requiriendo atención inmediata
emerg	Panic del sistema

Veamos el aspecto fichero de configuración de logs (/etc/syslog.conf):

```
auth, authpriv.*
                                 /var/log/auth.log
*.*;auth,authpriv.none
                                 -/var/log/syslog
#cron.*
                                 /var/log/cron.log
daemon.*
                                 -/var/log/daemon.log
kern.*
                                 -/var/log/kern.log
lpr.*
                                 -/var/log/lpr.log
mail.*
                                 -/var/log/mail.log
user.*
                                 -/var/log/user.log
uucp.*
                                 /var/log/uucp.log
# Logging for the mail system. Split it up so that
# it is easy to write scripts to parse these files.
#
mail.info
                                 -/var/log/mail.info
mail.warn
                                 -/var/log/mail.warn
mail.err
                                 /var/log/mail.err
```

Para arrancar el daemon que controla los logs:

/etc/init.d/sysklogd start

Para reiniciarlo si hacemos algún cambio en el fichero de configuración :

/etc/init.d/sysklogd restart

Podemos también activar la salida de una determinada facilidad y un determinado nivel a un terminal en lugar de a un fichero o a las dos cosas simultáneamente. Por ejemplo, para ver la salida de los logs generales del sistema en una de las siete consolas podremos esta línea en el fichero /etc/syslog.conf :

<pre>*.info;mail.none;authpriv.none</pre>	/var/log/messages
<pre>*.info;mail.none;authpriv.none</pre>	/dev/tty6

Después, debemos reiniciar el demonio:

```
/etc/init.d/syslog restart
```

9.2. Ficheros de logs

Los ficheros de registro del sistema se encuentran en el directorio **/var/log**. Estos son algunos de los ficheros de registro de Debian.

/var/log/messages	Fichero de log principal
/var/log/mail.*	Ficheros de log del correo (POP y SMTP) . Clasificados según la severidad.
/var/log/wtmp	Fichero binario. Indica la fecha y hora y el tiempo de login de los usuarios. A él accede el comando last.
/var/run/utmp	Fichero binario. Indica los usuarios conectados actualmente. Lo usan mandatos como who, w y finger.
/var/log/lastlog	Fichero binario. Determina la fecha y hora y el tiempo de login de los usuarios. A él accede el comando finger. Es parecido a wtmp.
/var/log/kern.log	Mensajes del kernel
/var/log/daemon.log	Demonios en el sistema

9.3. Visualización de los ficheros

Para poder visualizar los ficheros de log disponemos del comando **tail**. Si queremos ver un fichero del log con el comando cat o more esto puede convertirse en una tarea tediosa.

Comandos más importantes:

tail	Muestra las últimas líneas del fichero de texto que le indiquemos
	 -f -> Sigue sacando las líneas finales, según se escriben en el fichero -n xx -> Saca las últimas xx líneas del fichero
head	Muestra las 10 líneas iniciales del fichero de texto que le indiquemos -n xx -> Muestra las xx líneas iniciales

Ejemplos:

Muestra las últimas líneas del fichero /var/log/messages

tail /var/log/messages

Muestra las 50 últimas líneas del fichero /var/log/messages

tail -n 50 /var/log/messages

Muestra las continuamente el final del fichero /var/log/messages

tail _f /var/log/messages

Muestra las 10 primeras líneas del fichero /var/log/messages

head /var/log/messages

9.4. Rotación de los ficheros de log

El problema de los ficheros de log es que pueden alcanzar tamaños muy grandes y volverse inmanejables. Debido a ello, se les cambia de nombre añadiendo un numero al final, y se crean nuevos.

El control de estos y más ficheros (depende de los servicios instalados) se realiza a través del fichero **/etc/logrotate.conf**. Este fichero nos permite indicarle como se realizará la rotación de los ficheros de log. Cada cuanto tiempo se rotarán, como crear los nuevos, cuantos mantener, etc. Veamos su contenido.

```
# see "man logrotate" for details
# rotate log files weekly
weekly
# keep 4 weeks worth of backlogs
rotate 4
# create new (empty) log files after rotating old ones
create
# uncomment this if you want your log files compressed
#compress
# packages drop log rotation information into this directory
include /etc/logrotate.d
# no packages own wtmp, or btmp -- we'll rotate them here
/var/log/wtmp {
    missingok
    monthly
    create 0664 root utmp
    rotate 1
}
/var/log/btmp {
    missingok
    monthly
    create 0664 root utmp
    rotate 1
```

system-specific logs may be configured here

Además de este fichero, se utilizan también los ficheros del directorio /etc/logrotate.d. En este directorio ponemos los ficheros de control de rotación de las aplicaciones que vayamos instalando. Veamos como ejemplo, el fichero correspondiente al servidor web apache. Los ficheros de log de la versión 2 de apache se encuentran en /var/log/apache2.

```
/var/log/apache2/*.log {
        weekly
       missingok
        rotate 52
        compress
        delaycompress
        notifempty
        create 640 root adm
        sharedscripts
        postrotate
                if [ -f /var/run/apache2.pid ]; then
                        /etc/init.d/apache2 restart > /dev/null
                fi
        endscript
```

}

}

10. La red

La red es parte fundamental en todo sistema tipo Unix. La red no sólo como sistema de comunicación entre hosts, sino sistema de comunicación entre procesos. Esto es: Aunque no exista una red entre ordenadores, pues solamente tenemos uno, y ni siquiera tenga tarjeta de red física, necesitamos de una red IP para que muchos de los procesos se comuniquen y funcionen.

La familia de protocolos nativa en los sistema Unix es el TCP/IP. TCP/IP nació en sistemas Unix, y ambos están muy ligados.

No vamos a entrar en la teoría de redes TCP/IP, ni en el estudio de los protocolos que la componen. Es nuestro objetivo, saber configurar en Linux, una red IP, y saber dónde mirar en caso de problemas.

Debemos indicar que el término interfase, aquí hace referencia a un interfase de red lógico. Este interfase puede estar ligado a un interfase de red físico (una tarjeta de red o NIC), o no. Además un NIC puede soportar varios interfaces. Esto es: Una tarjeta de red puede soportar varias direcciones IP con sus conexiones correspondientes.

Los comandos a utilizar son los siguientes:

ifconfig	Muestra / mo	difica la configuración la red
	interface - address - broadcast - netmask - up - down -	 > Interfase de red (eth0, eth1,) > Dirección ip > Dirección broadcast > Máscara de subred > Activa un interfase de red > Desactiva un interfase de red
route	Muestra / modifica la tabla de rutas	
	add - del - default - gw - net - host -	 > Añade una ruta > Borra una ruta > Ruta por defecto > gateway > red > host
netstat	Muestra el es	stado de la red
arp	Muestra / mo	difica la caché de resolución arp

ping	Envía un mensaje de solicitud de eco a un host (ICMP ECHO REQUEST)	
	 -c -> Número de mensajes -n -> No resolver los nombres de host 	
traceroute	Muestra la ruta que toman los paquetes a un host	
	-n -> No resolver los nombres de host	
ipcalc	Calculadora de parámetros de red	

Ejemplos:

Asignar la dirección 10.22.1.33/24 al interfase eth0

ifconfig etho 10.22.1.33 netmask 255.255.255.0

Desactivar el interfacer eth0

ifconfig eth0 down

Añadir una rupa de red:

route add -net 10.22.1 gw 10.22.1.33

Borra una ruta a un host:

route del -host 10.22.3.44

Añadir una ruta por defecto:

route add default gw 10.22.1.1

Ver tablas de rutas

route netstat -nr

Ver caché de resolución de direcciones ethernet a IP.

arp

Mirar si se puede acceder (o está despierto) el host 10.22.3.8

ping www.google.com
ping -nc 12 10.22.3.8

Mostrar la ruta hasta ww.google.com:

traceroute www.google.com

Calcular los parámetros de red correspondientes al interfase 10.22.3.8/24

ipcalc 10.22.3.8/24

11. Actualización del sistema e instalación de paquetes

En un sistema Debian, la instalación de paquetes y la actualización se realizan mayormente con la herramienta **apt**. Los comandos habituales son:

apt-get update	Actualiza el listado de paquetes disponibles en los repositorios
apt-get upgrade	Actualiza todos los paquetes instalados, ala versión más nueva
apt-get install xxxxx	Instala el paquete xxxxxx de un repositorio
apt-get remove xxxx	Desinstala el paquete xxxx del sistema
apt-cache search xxx	Busca em los repositorios, un paquete que contenga xxx
apt-cache showpkg xx	Muestra toda la información del paquete xx
apt-get dist-upgrade	Actualiza la versión de la distribución

Para tratar paquetes individuales se puede utilizar también **dpkg**. Por ejemplo, si disponemos de un paquete llamadlo "miprograma-1.0.3.i386.deb" en el directorio home, podemos instalarlo con el siguiente comando:

dpkg -i ~/miprograma-1.0.3.i386.deb

El comando dpkg es de más bajo nivel, y tiene infinidad de opciones.

11.1. Instalación de paquetes

La instalación de paquetes se realiza normalmente desde repositorios de la distribución. Un repositorio pude estar en CD.ROM o Internet o la red local de una empresa. Los repositorios que utiliza un sistema están listados en el fichero **/etc/apt/sources.list**. El contenido de este fichero será algo similar a lo siguiente:

```
deb http://ftp.es.debian.org/debian/ etch main contrib non-free
deb-src http://ftp.es.debian.org/debian/ etch main contrib non-free
deb http://security.debian.org/ etch/updates main contrib
```

El significado de las distintas partes:

deb	indica paquetes binarios Debian
deb-src	indica paquetes de código fuente
http://	indica la dirección del repositorio y el protocolo a utilizar
etch	indica la distribución (en este caso Etch)
updates	actualizaciones de seguridad
main	paquetes soportados, propios de la distribución
contrib	paquetes contribuidos por empaquetadores Debian
non-free	paquetes con restricciones de distribución por motivos legales en algunos países o código no libre

Ejemplos:

Para instalar el servidor web **apache2** y la versión 5.0 del servidor de bases de datos **mysql**:

apt-get install apache2 mysql-server-5.0

En sistemas Red Hat o SUSE, que utilizan paquetes RPM, se utiliza la aplicación del mismo nombre **rpm**. Por ejemplo, en el caso del programa anterior, ejecutaríamos:

rpm -ivh ~/miprograma-1.0.3.i386.rpm

Para las actualizaciones tenemos **yum** para RH (es de Yellow Dog) o **you** de SUSE.

12. Tareas importantes a realizar

12.1. El nombre de la máquina

Debemos verificar que el nombre de la máquina está correctamente puesto en el fichero **/etc/hostname,** en el sistema, y en el fichero **/etc/hosts**.

El contenido del fichero /etc/hostname será algo así:

I2DBPC01

Este puede ser el contenido del fichero /etc/hosts:

```
127.0.0.1 localhost
127.0.0.1 I2DBPC01.iurreta-institutua.net I2DBPC01
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

El sistema lee el fichero **/etc/hostname** al inicio. Es por ello que si lo modificamos, debemos de reiniciar el sistema, o decirle al sistema que lo cambie (internamente). Para ello utilizaremos el comando **hostname**.

hostname	Muestra el nombre del host
	-F Lee el nombre del host (hostname -F << /etc/hostname)

12.2. La resolución de nombres

En el fichero **/etc/nsswitch.conf** tenemos la secuencia que sigue la resolución de nombres.

hosts: files dns wins

En caso de que no dispongamos servidor **DNS**, tendremos que poner los nombres de las máquinas de nuestra red, en el fichero **/etc/hosts**. En caso de estar en una red Windows antigua, y no disponer de servidor DNS, nos convendrá poner **wins** antes que **dns**.

12.3. Deshabilitar el reinicio por teclado

La distribución Debian, por defecto tiene activado el reinicio del sistema cuando detecta la combinación de teclado **CTRL+ALT+DEL**. Debemos desactivarlo. De este modo, se deberá introducir el comando oportuno para reiniciar el sistema.

El el fichero **/etc/inittab**, debemos comentar (poner "#" delante) la siguiente línea, que quedará así:

#ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -a now

12.4. La hora, la hora, la hora

Poner la hora correctamente es importante. Téngase en cuenta que estamos en una red. Al registrar eventos debemos fecharlos cuando de verdad suceden. Además algunos protocolos de red (entre ellos Kerberos) requieren que la hora de distintos ordenadores esté sincronizada. Además, mucho hardware de servidores no tiene un reloj muy preciso. Por ello, debemos sincronizar el reloj de nuestro servidor con una fuente horaria externa. Cualquier distribución Linux viene con software tanto servidora como cliente de hora. La hora se actualiza de dos maneras. Al encender el ordenador de una forma brusca, y durante su funcionamiento, realizando actualizaciones dependiendo del deslizamiento horario de nuestro sistema.

Primero vamos a instalar el software:

apt-get install ntp ntpdate

Ahora vamos a configurar nuestra referencia. Si tenemos conexión a Internet, utilizaremos un servidor de Internet. Recomendable es el **pool.ntp.org**. Si tenemos una red grande en nuestra organización, se recomienda utilizar un servidor de la red. Si tenemos un dominio ActiveDirectory, nuestro controlador de dominio es la mejor referencia.

El fichero de configuración se encuentra en **/etc/ntp.conf**. En él, vamos a poner el (los) servidores a utilizar. En el caso de **pool.ntp.org**, debemos sustituir las líneas que empiezan por server por lo siguiente:

```
server 0.pool.ntp.org
server 1.pool.ntp.org
server 2.pool.ntp.org
```

El fichero de configuración de ntpdate (/etc/default/ntpdate) no hace falta modificarlo.

Por defecto, se indica que tome la referencia a utilizar del fichero de configuración de ntp.

Tras esto, reiniciamos ntp.

/etc/init.d/ntp restart

12.5. Acceso por la red : SSH

En un servidor es común acceder al mismo por la red. Esto nos permite acceder al mismo desde lugares remotos, o aunque estemos en la misma localización, no tener que instalar periféricos para el mismo. el acceso se puede realizar tanto en modo consola o gráfico.

Las comunicaciones conviene realizarlas cifradas. De este modo, no se transmitirán por la red nuestros datos (entre otros, contraseñas de acceso) en testo plano. Los datos se cifran con algún método criptográfico. No vamos a entrar aquí en teoría ni sistemas de certificados, ni cosas de esas. Vamos a ver simplemente un acceso de clave compartida.

El sistema de acceso que vamos a utilizar el **ssh**. Hay que mencionar que **ssh** también permite la copia de ficheros entre máquinas, de forma cifrada con **scp**. Vamos a instalarlo.

apt-get install ssh

Vamos a configurar el paquete. El fichero de configuración es **/etc/ssh/sshd_config**. Vamos a poner las siguientes líneas de esta forma:

Protocol 2 PermitRootLogin no

Si queremos securizar más el sistema, podemos evitar el acceso con password y que se usen certificados, limitarlo a unas máquinas, etc.

Para acceder al sistema como el usuario elusuario:

ssh elusuario@lamaquina.eldominio.net

Para copiar el fichero **mifile** de **/home/mihome** al directorio home del usuario **elusuario** en el sistema 10.22.1.59:

scp /home/mihome/mifile elusuario@10.22.1.59:/home/elusuario/

13. Más comandos

Aquí tenemos un montón de comandos más que son de utilidad en la administración básica de un sistema.

top	Muestra en estado de las tareas y la memoria y CPU utilizadas
free	Muestra la cantidad de memoria libre y en uso
uptime	Tiempo desde el último reinicio del sistema
clear	Borra la pantalla
id	Identidad y grupos del usuario
who	Quién está conectado al sistema y en que terminal
hostname	Nombre del host
date	Fecha y hora del sistema
write	Enviar un mensaje a otro usuario conectado al sistema
	Debe estar activada la opción (mesg y)
mail	Debe estar activada la opción (mesg y) Enviar correo a otro usuario
mail more	Debe estar activada la opción (mesg y) Enviar correo a otro usuario Produce una salida de pantalla por páginas
mail more tar	Debe estar activada la opción (mesg y) Enviar correo a otro usuario Produce una salida de pantalla por páginas Crea / extrae archivos de ficheros (comprimidos o no)
mail more tar grep	Debe estar activada la opción (mesg y) Enviar correo a otro usuario Produce una salida de pantalla por páginas Crea / extrae archivos de ficheros (comprimidos o no) Filtra la entrada y en su salida se obtienen las líneas que cumplen el filtro
mail more tar grep chown	Debe estar activada la opción (mesg y) Enviar correo a otro usuario Produce una salida de pantalla por páginas Crea / extrae archivos de ficheros (comprimidos o no) Filtra la entrada y en su salida se obtienen las líneas que cumplen el filtro Cambiar el propietario de un fichero o directorio
mail more tar grep chown mount	Debe estar activada la opción (mesg y) Enviar correo a otro usuario Produce una salida de pantalla por páginas Crea / extrae archivos de ficheros (comprimidos o no) Filtra la entrada y en su salida se obtienen las líneas que cumplen el filtro Cambiar el propietario de un fichero o directorio Monta una partición o volumen
mail more tar grep chown mount	Debe estar activada la opción (mesg y) Enviar correo a otro usuario Produce una salida de pantalla por páginas Crea / extrae archivos de ficheros (comprimidos o no) Filtra la entrada y en su salida se obtienen las líneas que cumplen el filtro Cambiar el propietario de un fichero o directorio Monta una partición o volumen -t tipo de sistema de ficheros

14. Ficheros importantes del sistema

/boot/grub/menu.lst	Fichero donde se indica el menú de arranque y algunas
	opciones.
	default -> Opción que está resaltada por defecto
	timeout -> tiempo de espera del menu
/etc/hostname	Nombre del host
	IZDB01
/etc/hosts	Fichero para resolver nombres de host y sus direcciones IP
	127.0.0.1 IOCAINOSt
loteleorricoe	Listado do los sorvicios TCP y LIDP por número do nuerto
	Les interferences de mediciels refry ODF, por indificio de puerto
/etc/network/interfaces	Las interfaces de red del sistema y su configuración
	auto lo
	iface lo inet loophack
	auto eth0 eth0:1
	iface eth0 inet static
	address 10.22.1.17
	netmask 255.255.255.0
	network 10.22.1.0
	broadcast 10.22.1.255
	gateway 10.22.1.1
	dns-nameservers 10.22.1.8
	dns-search iurreta-institutua.net
	iface eth0:1 inet static
	address 10.22.2.17
	netmask 255.255.255.0
	network 10.22.2.0
	broadcast 10.22.2.255

	gateway 10.22.2.1
/etc/sudoers	
/etc/fstab	<pre># /etc/fstab: static file system information. #</pre>
	<pre># <file system=""> <mount point=""> <type> <options></options></type></mount></file></pre>
	<dump> <pass></pass></dump>
	proc /proc proc defaults 0 0
	/dev/sda3 / ext3 defaults,errors=remount-ro 0 1
	dev/sda2 /boot ext3 defaults 0 2
	/dev/sda1 none swap sw 0 0
	/dev/hda /media/cdrom0 udf,iso9660 user,noauto 0 0
	/dev/fd0 /media/floppy0 auto rw,user,noauto 0 0
/etc/inittab	# The default runlevel. id:2:initdefault:
	# Boot-time system configuration/initialization script.# This is run first except when booting in emergency (-b) mode.si::sysinit:/etc/init.d/rcS
	<pre># What to do in single-user mode. ~~:S:wait:/sbin/sulogin</pre>
	# /etc/init.d executes the S and K scripts upon change
	# of fullevel.
	# Runlevel 0 is halt.
	# Runlevel 1 is single-user.
	# Runlevels 2-5 are multi-user.
	# Runlevel 6 is reboot.
	10:0:wait:/etc/init.d/rc 0
	l1:1:wait:/etc/init.d/rc 1
	l2:2:wait:/etc/init.d/rc 2
	13:3:wait:/etc/init.d/rc 3
	l4:4:wait:/etc/init.d/rc 4
	15:5:wait:/etc/init.d/rc 5
	# Normally not reached, but fallthrough in case of omorganov
	z6:6:respawn:/sbin/sulogin
	# What to do when CTDL ALT DEL is presend
	# what to do when CTRL-ALI-DEL is pressed. # ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
	# Action on special keypress (ALT-UpArrow).

	#kb::kbrequest:/bin/echo "Keyboard Requestedit /etc/inittab to let this work."
	# What to do when the power fails/returns. pf::powerwait:/etc/init.d/powerfail start pn::powerfailnow:/etc/init.d/powerfail now po::powerokwait:/etc/init.d/powerfail stop
	# /sbin/getty invocations for the runlevels. #
	# The "id" field MUST be the same as the last # characters of the device (after "tty"). #
	<pre># Format: # <id>:<runlevels>:<action>:<process></process></action></runlevels></id></pre>
	# # Note that on most Debian systems tty7 is used by the X Window System
	# so if you want to add more getty's go ahead but skip tty7 if you run X.
	# 1:2345:respawn:/sbin/getty 38400 tty1 2:23:respawn:/sbin/getty 38400 tty2
	3:23:respawn:/sbin/getty 38400 tty3 4:23:respawn:/sbin/getty 38400 tty4 5:23:respawn:/sbin/getty 38400 tty5
	6:23:respawn:/sbin/getty 38400 tty6
	# Example how to put a getty on a serial line (for a terminal)#
	#T0:23:respawn:/sbin/getty -L ttyS0 9600 vt100 #T1:23:respawn:/sbin/getty -L ttyS1 9600 vt100
	# Example how to put a getty on a modem line.
	#T3:23:respawn:/sbin/mgetty -x0 -s 57600 ttyS3
/etc/mtab	/dev/sda3 / ext3 rw,errors=remount-ro 0 0
	proc /proc proc rw.noexec.nosuid.nodev 0 0
	svsfs /svs svsfs rw.noexec.nosuid.nodev 0 0
	prochususb /proc/bus/usb usbfs rw 0 0
	udev /dev tmpfs rw,mode=0755 0 0
	tmpfs /dev/shm tmpfs rw,nosuid,nodev 0 0
	devpts /dev/pts devpts rw,noexec,nosuid,gid=5,mode=620 0 0
	/dev/sda2 /boot ext3 rw 0 0
	/dev/sda6 /var ext3 rw 0 0

/etc/resolv.conf	Fichero de dominio por defecto y servidores DNS	
	search iurreta-institutua.net nameserver 10.22.3.8	
/etc/passwd	Fichero de usuarios del sistema	
/etc/shadow	Fichero de contraseñas de los usuarios	
/etc/group	Fichero de grupos del sistema	
/etc/nsswitch	Orden de resolución de nombres y fuentes de usuarios para la autentificación	
	passwd: #passwd: group: #group: shadow: #shadow: hosts: #hosts: networks: protocols: services: ethers: rpc:	compat files ldap compat files ldap compat files ldap files dns files dns files dns ldap files db files db files db files db files
	netgroup:	nis

15. Referencias

Lista de comandos de GNU/Linux I. II y III

http://www.esdebian.org/staticpages/index.php?page=20050930063719706 http://www.esdebian.org/staticpages/index.php?page=200601112133535 http://www.esdebian.org/staticpages/index.php?page=20060330140303942

Aprendiendo la shell

http://www.linuxcommand.org/learning_the_shell.php

16. Autor

Alfredo Barrainkua Zallo

Iurreta Institutuko Sare Administraria

alfredobz @iurreta-institutua.net