Linux Administración nivel 1



Oscar

Índice

•	Estructura del sistema de archivos en Linux	4-20
	- Introduccion e historia	
	- FRS Todo on Linux os un archivo	
	- Todo en Linux es un archivo Organización, del Sistema de Archives según EHS	7
	- Organización del Sistema de Archivos segun FHS	7
	 Contonido del directorio raíz 	י ד 10
		7-12
	- /biii	
	- /modia	
	- /media	
	- /mit	
	- //oor	
	- /tmn	
	\sim El Directorio /usr	13-16
		10-10
	 /usr/local 	
	\sim El Directorio /var	17-19
		17 18
	 /var/lock 	
	 /var/log 	
	• /var/run	
	 /var/spool 	
	 /var/tmp 	

	 El Directorio /lost+found. 	20
	 El directorio /proc 	20
•	Instalación de Linux	21-29
	 Instalación paso a paso : Ubuntu 	21-29
•	Administración de Usuarios Y Grupos	30-34
	- Creación de Usuarios	
	- Modificación de Usuarios	
	- Eliminación de Usuarios	
	- Creación de Grupos	
	- Modificación de Grupos	
	- Eliminacion de Grupos	
	- Anadir Usuarios do un Grupo	
•	Herramienta Gráfica de Administración de Usuarios	34
	Maneio de Archivos Y directorios a través de Comandos	34
•	- Comandos Linux para la Gestión de archivos y directorios.	00-40
	o CD	
	o mv	
	o rm	
	o mkdir	
	o rmdir	
	○ IS	
	o pwu	
	o find	
	o grep	
	- Otros Comandos de Linux	45-46
	o man	
	∘ mesg	
	o lpr	
	o tree	
	o liy	
•	Configuración de Red	47-49
•	 Dirección IP Estática 	47 40
	 Dirección IP Dinámica 	
•	Sistemas de Permisos	50-60
	 Notación Simbólica 	
	 Notación Octal 	
	 Mascara de Usuario 	
	Control de Procesos	
	pstree	



- kill
- killall
- nice
- renice
- Instalación y Actualización de Software.
 - Debian: Paquetes DEB
 - o apt-get
 - Red Hat: Paquetes RPM
 - Filtros y Redirecciones
 - sort
 - shuf
 - more
 - less
 - tail
 - head
 - grep
 - cut
 - uniq
 - ∎ tr
 - tee
 - Redirecciones
 - stdin
 - stdout
 - stderr
 - Redireccionamiento de Entrada
 - Redireccionamiento de Salida
 - Redireccinamiento de Adicción
 - Redireccionamiento de Errores
 - Redireccionamiento Completo
 - Redireccionaiento por Parámetros
 - o Tuberías, PIPES, Encadenamiento de Comandos
- Proceso de Arranque y niveles de Ejecución
 - Descripción general de un proceso típico de arranque en gnu/Linux
 - Niveles de Ejecución

• X-Windows / Entornos gráficos

- Que es X-Windows
- Como configurar las X-windows
- Como cambiar la resolución por defecto
- Como cambiar es escritorio virtual
- Como cambiar el número de colores por defecto
- Como arrancar directamente en X-Windows
- Que es un gestor de Ventanas
- o Como elegir el Gestor de Ventanas que se desea arrancar
- o Donde conseguir un nuevo Gestor de Ventanas

61-72

76-85

73-75



La estructura del sistema de archivos en Linux

1.- Introducción e historia

Un sistema Linux reside bajo un árbol jerárquico de directorios muy similar a la estructura del sistema de archivos de plataformas Unix.

Originariamente, en los ini cios de Linux, este árbol de directorios no seguía un estándar cien por cien, es decir, podíamos encontrar diferencias en él de una distribución a otra.

Todo esto hizo pensar a cierta gente^{*} que, posteriormente, desarrollarían el proyecto FHS (Filesystem Hierarchy Standard, o lo que es lo mismo: Estándar de Jerarquía de Sistema de Ficheros) en otoño de 1993.

* Rusty Russell, Daniel Quinlan y Christopher Yeoh, creadores del estándar FHS entre otras personas.

2.- FHS

FHS se define como un estándar que detalla los nombres, ubicaciones, contenidos y permisos de los archivos y directorios, es decir, un conjunto de reglas que especifican una distribución común de los directorios y archivos en sistemas Linux.

Como se ha mencionado, se creó inicialmente para estandarizar la estructura del sistema de archivos para sistemas GNU/Linux y más tarde, en torno al año 1995, también para su aplicación en sistemas Unix.

FHS no es más que un documento guía, es decir, cualquier fabricante de software independiente o cualquier persona que decida crear una nueva distribución GNU/Linux, podrá aplicarlo o no a la estructura del sistema de archivos, con la ventaja de que si lo integra en el sistema, el entorno de éste será mucho más compatible con la mayoría de las distribuciones.

Es importante saber que el estándar FHS es en cierto modo flexible, es decir, existe cierta libertad en el momento de aplicar las normas. De ahí que existan en la actualidad leves diferencias entre distribuciones GNU/Linux.

Objetivos principales de FHS:

- Presentar un sistema de archivos coherente y estandarizado.
- Facilidad para que el software prediga la localización de archivos y directorios instalados.
- Facilidad para que los usuarios prediga la localización de archivos y directorios instalados.

une@eb

Instituto de nuevas Tecnologías

- Especificar los archivos y directorios mínimos requeridos.
- El estándar FHS está enfocado a Fabricantes de software independiente y creadores de sistemas operativos, para que establezcan una estructura de ficheros lo más compatible posible.
- Usuarios comunes, para que entiendan el significado y el contendido de cada uno de los elementos del sistema de archivos.

Además, FHS manifiesta algunas diferencias entre varios tipos de archivos que puede haber en el sistema:

- Archivos compartibles y no compartibles.
- Ficheros que son propios de un host determinado y, archivos que pueden compartirse entre diferentes host.

Ejemplo:

- Archivos compartibles: los contenidos en /var/www/html (que es el DocumentRoot por defecto del servidor Web Apache. Donde se almacena inicialmente el index.html de bienvenida).
- Archivos no compartibles: los contenidos en /boot/grub/ (Subdirectorio donde se ubican los ficheros del gestor de arranque GRUB).
- Archivos estáticos y variables.
 Ficheros que no cambian sin la interacción de un administrador del sistema y, archivos que cambian sin la interacción de un administrador del sistema.

Para comprender mejor estos dos tipos, imaginemos los ficheros log (archivos de bitácora) del sistema. Estos cambian sin la intervención del administrador; en consecuencia estos son del tipo variables.

Los demás archivos son estáticos. No cambian su contenido ni tamaño a menos que lo autorice el administrador del sistema (o sea el propio quien lo modifique, por supuesto).

- Archivos estáticos: /etc/password, /etc/shadow.
- Archivos variables: /var/log/messages (log de mensajes generados por el kernel del sistema).

3.- Todo en Linux es un archivo

Cierto, todo en un sistema Linux es un archivo, tanto el Software como el Hardware. Desde el ratón, pasando por la impresora, el reproductor de DVD, el monitor, un directorio, un subdirectorio y un fichero de texto.

De ahí vienen los conceptos de montar y desmontar por ejemplo un CDROM.

El CDROM se monta como un subdirectorio en el sistema de archivos. En ese subdirectorio se ubicará el contenido del disco compacto cuando esté montado y, nada cuando esté desmontado.

Para ver que tenemos montado en nuestra distribución GNU/Linux, podemos ejecutar el comando mount.

Este concepto es muy importante para conocer cómo funciona Linux.

En apartados posteriores, veremos donde ubica Linux los elementos Hardware del PC en el sistema de ficheros.

NOTA: podemos acceder a los dispositivos Hardware como si fueran archivos. Realmente son ficheros para Linux; pero no son archivos normales, son archivos binarios (o .exe para los que vengan de Windows).

Hay que saber que si editamos, por ejemplo, un fichero vinculado a un elemento Hardware, seguramente sea totalmente ilegible y posiblemente quedará inutilizable y bloqueada nuestra línea de comandos (shell). Es más, corremos el riesgo de corromper los datos y dejar el sistema inestable.

En definitiva, no es aconsejable leer o abrir y mucho menos modificar archivos vinculados a elementos Hardware y/o dispositivos, a menos que sepamos con toda seguridad lo que estemos haciendo.



4.- Organización del sistema de archivos según FHS

4.1.- El directorio raíz

Todo surge a partir del directorio raíz (/).

El contenido de este directorio debe ser el adecuado para reiniciar, restaurar, recuperar y/o reparar el sistema, es decir, debe proporcionar métodos, herramientas y utilidades necesarias para cumplir estas especificaciones.

Además, es deseable que se mantenga los más razonablemente pequeño como sea posible por cuestión de funcionamiento y de seguridad.

Por último, este debe que ser el único directorio en el nivel superior del árbol jerárquico de archivos y, tiene que ser imposible moverse más allá del mismo.

Es el último origen.

a		alex@ale	xDebian: /			
Archivo	<u>E</u> ditar ⊻er <u>T</u> ermina	<u>S</u> olapas Ayud	ła			
alex@alex	Debian:/\$ ls -F		-			
bin/	downloads/	jorge/	mnt/	STV/	Whitnuz.nld@	
boot/	etc/	Lib/	opt/	sys/		
cdro @	home/	lib64/	proc/	tmp/		
ctr2/	initrd/	lost+found/	publico/	usr/		
Desastre/	initrd.img@	marfil/	root/	var/		
dev/	Debian:/S	media/	sbin/	wmlinuz@		

Vemos, por ejemplo, un listado de su estructura:

Contenido del directorio raíz

• /bin

En este directorio se ubica el código binario o compilado de los programas y comandos que pueden utilizar todos los usuarios del sistema.

La denominación es clara, bin de BINARY (binario en castellano).

No debe haber subdirectorios en /bin.

Estos son, por ejemplo, algunos comandos contenidos en /bin.

unelleb

Instituto de nuevas Tecnologías

3		alex@alexDebian:	/bin		
Archivo Editar	Ver Terminal So	lapas Ayuda			
alex@alexDebia	n:/bin\$ ls -F				
arch*	csh@	kill*	netstat*	tempfile*	
bash*	date*	Lo*	pldot@	touch*	
bunzip2*	dd*	loadkeys*	ping*	true*	
busybox*	df*	login*	ping6*	umount*	
bzcat*	dir*	15*	ps*	uname*	
bzcmp*	dnesg*	lsb_release*	pwd*	uncompress*	
bzd1ff*	dnsdomainname*	Lsmod*	rbash@	vdir*	
bzegrep*	echo*	lsmod.modutils@	readlink*	which*	
bzexe*	ed*	lspc1@	ra*	zcat*	
bzfgrep*	egrep*	mkdir*	rmd1r*	zcmp*	
bzgrep*	false*	mknod*	rnanc@	zdiff*	
bzip2*	fgconsole*	mktemp*	run-parts*	zegrep*	
bzip2recover*	fgrep*	more*	sed*	zfgrep*	
bzless*	fuser*	mount*	setpci@	zforce*	
bzmore*	grep*	mountpoint*	she	zgrep*	
cat*	gunzip*	t@	sleep*	zless*	
chgrp*	gzexe*	mt-gnu*	stty*	zmore*	
chmod*	gzip*	mv*	su	znew*	
chown*	hostname*	nano*	sync*		
cp*	10*	nc*	tar*		
cpio*	kernelversion@	netcat@	tcsh@		
alex@alexDebia	in:/bin\$		6		•

Símbolo @ al lado del nombre de un fichero representa un enlace simbólico

• /boot

Este directorio contiene todo lo necesario para que funcione el proceso de arranque del sistema, /boot almacena los datos que se utilizan antes de que el kernel comience a ejecutar programas en modo usuario.

El núcleo del sistema operativo (normalmente se guarda en el disco duro como un fichero imagen llamado *vmlinuz-versión _ núcleo*) se debe situar en este directorio o, en el directorio raíz.



El núcleo tiene la capacidad de crear dos entornos o modos de ejecución totalmente separados, uno de ellos está reservado para el propio kernel, denominado el "modo núcleo"; y el otro está reservado para el resto de programas, llamado el "modo usuario". Realmente se crean dos entornos totalmente separados, es decir, cada uno tiene su propia zona de memoria y procesos independientes.

Démonos cuenta que esta técnica ofrece mucha seguridad y estabilidad al sistema. Cuando un proceso del "modo usuario" necesita recursos del "modo kernel" (por ejemplo, acceder a la memoria USB) se hacen uso de las "famosas" llamadas al sistema (interface que ofrece el núcleo para la comunicación del modo usuario con el modo kernel).

a	alex@alexDebian: /boot	HI
Archivo Editar Ver Termi	nal <u>S</u> olapas Ayuda	
alex@alexDebian:/boot\$ l config-2.6.16-2-686 config-2.6.17-2-686 grub/ initrd.img-2.6.16-2-686 alex@alexDebian:/boot\$	s -F initrd.img-2.6.17-2-686 vmlinuz-2.6.16-2-6 lost+found/ vmlinuz-2.6.17-2-6 System.map-2.6.16-2-686 System.map-2.6.17-2-686	86 86

/dev

Este directorio almacena las definiciones de todos los dispositivos. Como se ha mencionado, cada dispositivo tiene asociado un archivo especial.

Por ejemplo, el contenido de la sexta partición del disco duro será /dev/hda5.

El fichero asociado al ratón tipo PS/2 será /dev/psaux.

Además, es importante saber que los dispositivos pueden ser de bloque o de carácter. Normalmente los dispositivos de bloque son los que almacenan datos y, los de carácter los que transfieren datos.

Én definitiva, la estructura de este directorio es algo complejo. Podríamos dedicar otro artículo entero para poder explicar en profundidad el contenido y funcionamiento especifico del directorio /dev. De momento nos basta con saber que Linux lo utiliza para asociar dispositivos (devices) con ficheros.

NOTA: El subdirectorio /dev/null es como un "agujero negro". Esto es así, puesto que cualquier dato que se almacena aquí, desaparece. Es muy útil para re direccionar los errores

/etc

El directorio /etc contiene archivos necesarios para configuración del sistema.



Archivos que son propios del ordenador y que se utilizan para controlar el funcionamiento diversos programas.

Deben ser ficheros estáticos y nunca pueden ser archivos binarios y/o ejecutables. Algunos subdirectorios contenidos en /etc:

- X11. Subdirectorio para la configuración del sistema de ventanas. Es opcional.
- SGML. Subdirectorio para la configuración de SGML. Es opcional.
- Xml. Subdirectorio para la configuración de XML. Es opcional.

X11 – Sistema de ventanas graficas originario de UNIX en su versión 11. Este sistema tiene la peculiaridad de ser totalmente independiente del sistema operativo. Es una estructura cliente-servidor.

XML - eXtensible Markup Language (lenguaje de marcas extensible). Es un metalenguaje de etiquetas.

SGML - Standard Generalized Markup Language (Lenguaje de Marcación Generalizado). Sistema para la organización y etiquetado de documentos.

En definitiva, /etc mantiene los archivos de configuración del sistema para un ordenador específico.

Vemos parte de su contenido:

a	alex@alexDebian: /etc				
Archivo Editar Ver I	erminal <u>S</u> olapas Ayuda				
alex@alexDebian:/etcs	sls-F			*	
acpi/	environment	ld.so.conf	profile		
adduser.conf	esound/	ld.so.conf.d/	protocols		
adjtime	exim4/	ld.so.hwcappkgs	python/		
aliases	firefox/	libao.conf	python2.3/		
alsa/	fonts/	libgda/	python2.4/		
alternatives/	foomatic/	libpaper.d/	qt3/		
apache2/	fstab	lisp-config.lisp	rc0.d/		
apm/	gaim/	locale.gen	rcl.d/		
apt/	gconf/	localtime	rc2.d/		
at.deny	gdm/	logcheck/	rc3.d/		
autopackage/	gftp/	login.defs	rc4.d/		
bash.bashrc	gimp/	logrotate.conf	rc5.d/		
bash_completion	gnome/	logrotate.d/	rc6.d/		
bash completion.d/	gnome-vfs-2.0/	lsb-base/	rc.local*		
bonobo-activation/	gnome-vfs-mime-magic	magic	rcS.d/		
ca-certificates.conf	gre.d/	mailcap	reportbug.conf		
calendar/	groff/	mailcap.order	resolv.conf		
chatscripts/	group	mailname	rmt*		
common-lisp/	group-	mail.rc	rpc		
complete.tcsh	gshadow	manpath.config	samba/		
console/	gshadow-	menu/	sane.d/		
console-tools/	gssapi mech.conf	menu-methods/	scrollkeeper.conf		
cron.d/	gtk/	mime.types	scsi id.config	-	

/home

Directorio que contiene los subdirectorios que son directorios origen para cada uno de los usuarios del sistema.

Cada subdirectorio /home/user de cada usuario proporciona el lugar para almacenar sus ficheros, así como los archivos de configuración propios de cada uno.

Es importante saber que también algunos servicios, y no solo usuarios, crean aquí su directorio origen, por ejemplo: el servicio de transferencia de ficheros (FTP).

El administrador tiene su propio directorio home, que es /root.

• /lib

El directorio /lib contiene librerías compartidas (similar a las dll's para los usuarios de Windows) necesarias para arrancar el sistema y para los ficheros ejecutables contenidos en, por ejemplo, /bin.

Normalmente las librerías son ficheros binarios escritos en lenguaje C .

También contiene módulos del kernel esenciales que permiten el funcionamiento de muchos elementos Hardware. Se ubicarán normalmente en /lib/modules/versión-del-kernel/.

En el apartado "/usr/include" se explica que son y cómo funcionan las "librerías" y los archivos cabecera.

Aquí vemos parte de su contenido:

alex@a	alexDebian: /lib	Hill
Archivo Editar Ver Terminal Solapas	Ayuda	- 1-
alex@alexDebian:/lib\$ ls -F		
alsa/	libnsl-2.3.6.so	
cpp@	Libnsl.so.10	
discover/	libnss_compat-2.3.6.so	
firmware/	libnss_compat.so.2@	
1386-11nux-gni@	libnss_dns-2.3.6.so	
1486-linux-gnu/	Libnss_dns.so.2@	
init/	libnss_files-2.3.6.so	
iptables/	libnss_files.so.2@	
klibc-AJgIVoLaaW905YIHQQt104N964c.se	* libnss_hesiod-2.3.6.so	
ld-2.3.6.50*	libes_hesiod.so.2@	
1d-linux.so.20	libnss_nis-2.3.6.so	
Ltbacl.so.l@	libnss_nisplus-2.3.6.so	
libacl.so.l.l.0	libnss_misplus.so.2@	
libanl-2.3.6.so	Libnas_nis.so.2@	
libanl.so.10	libpamc.so.00	
libata.so.l@	libpamc.so.0.79	
libatm.so.l.0.0	libpan_misc.so.00	
libattr.so.1@	libpam_misc.so.0.79	
libattr.so.1.1.0	libpan, so.60	
libblkid.so.1@	libpam.so.0.79	
libblkid.so.1.0	libparted-1.7.so.1@	
libBrokenLocale-2.3.6.so	libparted-1.7.so.1.0.0	
libBrokenLocalé.so.1@	libpcprofile.so	-

• /media

Este directorio contiene los subdirectorios que se utilizan como puntos del montaje para los medios de almacenamiento, tales como disquetes, CD-ROM y memorias USB's.

• /mnt

Este directorio contiene sistemas de archivos externos que hayan sido montados. Las entidades que aparecen dentro de /mnt representan recursos externos a los que se puede acceder a través de este directorio.

opt

En este directorio (/opt de options, u opciones en castellano) se suelen instalar complementos o add-ons de los programas.

Las aplicaciones crean un subdirectorio dentro de /opt denominado con el mismo nombre del programa.

/root

Este directorio es el directorio /home del administrador del sistema (root).

/sbin

Los programas y comandos que se utilizan para la administración del sistema se almacenan en /sbin, /usr/sbin y /usr/local/sbin.

/sbin únicamente contiene los ejecutables esenciales para el arranque, recuperación y reparación del sistema.

Todos estos directorios (/sbin, /usr/sbin y /usr/local/sbin) se utilizan con fines administrativos, por tanto, sólo puede ejecutar su contenido el administrador.

Vemos parte de su contenido:

a	alex@alexDebia	an: /sbin	_#i=]
<u>Archivo</u> <u>E</u> ditar <u>V</u> er	<u>T</u> erminal <u>S</u> olapas Ayuda		
alex@alexDebian:/sb	in\$ ls -F		
acpi_available*	iptables-restore*	pivot_root*	
apm_available*	<pre>iptables-save*</pre>	plipconfig*	
badblocks*	iptunnel*	pmap_dump*	
blkid*	1sosize*	pmap_set*	
blockdev*	iwconfig*	portmap*	
bootlogd*	iwevent*	poweroff@	
cfdisk*	iwgetid*	rarp*	
debugfs*	iwlist*	T 200	
depmod*	impriv*	reboot@	
depmod.modutils*	iwspy*	resizeZfs*	
dhclient*	kallsyms@	ramod*	
discover*	kallsyms.modufils@	rmmod.Lmodutils@	
dosfsck*	kbdrate*	Finnod. modulils@	
dumpe2fs*	kernelversion*	route*	
e2fsck*	killall5*	rpc.lockd*	
e2image*	klogd*	rpc.statd*	
e2label*	ksyms@	rtacct*	
fd1sk*	ksyms.modulils@	rtmon*	
findfs*	ldconfig*	runlevel*	2223
fsck*	logsave*	scsi ide	
fsck.cramfs*	losetup*	sfdisk*	
fsck.ext2*	Lsmod@	shadowconfig*	
fsck.ext3*	Ismod. Lmodul 11.50	showmount*	-

• /srv

Contiene los archivos de datos específicos para cada servicio instalado en el sistema.

/tmp

En este directorio se guardan los archivos temporales.

4.2.- El directorio /usr

Es la segunda sección más grande o estructura jerárquica (después del directorio raíz) del sistema de ficheros.

Este directorio está pensado para almacenar datos que se puedan compartir con otros hosts. Estos datos además deben ser inalterables, es decir, sólo de lectura.

Normalmente, este directorio tiene su partición propia.

Comúnmente, se almacena aquí el software instalado en el sistema. Vemos su contenido:

a	alex@alexDebian: /usr						
Archivo	<u>E</u> ditar <u>V</u> er	Termina	<u>S</u> olapas	Ayuda			
alex@ale	xDebian:/u	sr\$ ls -	F				
applnk/	games/	lib/	libexec/	lost+found/	share/	X11R6/	
bin/	include/	lib64/	local/	sbin/	src/		111
alex@ale	exDebian:/u	sr\$		1.1.1			•

Estructura de /usr

• /usr/bin

Éste es el directorio primario de comandos ejecutables del sistema. /usr/bin alberga los archivos ejecutables vinculados al software instalado en el sistema.

• /usr/include

Linux está escrito en lenguaje C.

En C es posible utilizar funciones que ya estén predefinidas (como otros muchos lenguajes de programación) para incluirlas en el programa que estemos haciendo. Esta técnica se denomina programación modular.

Estas funciones se llaman comúnmente archivos cabecera (.h de header) y contienen las declaraciones externas de una librería.

La manera de incluir estos archivos cabecera en nuestro programa, es haciendo uso de la directiva include; de ahí la denominación del subdirectorio.



Ejemplo programa C:

```
#include <libreria >
main()
{
unsigned int num,masc;
char resp;
clrscr();
do{
```

Todos estos ficheros cabecera (que necesite el software instalado en el sistema) se almacenan en este subdirectorio.

Una librería no es más que un programa compilado, donde originariamente se implementó el código fuente de las funciones que la componen.

La declaración pública del conjunto de funciones de la librería reside en los archivos cabecera.

Vemos parte de su contenido:

a	ale	x@alexDebian	: /usr/include	t.	_40
Archivo Editar	Ver Terminal S	olapas Ayuda			
alex@alexDebia	n:/usr/include\$	ls -F			
aio.h	elf.h	grp.h	netatalk/	regexp.h	tar.h
aliases.h	endian.h	iconv.h	netax25/	resolv.h	termio.h
alloca.h	envz.h	ieee754.h	netdb.h	rpc/	termios.h
a.out.h	err.h	ifaddrs.h	neteconet/	rpcsvc/	tgmath.h
apache2/	errno.h	initreq.h	netinet/	sched.h	thread_db.h
argp.h	error.h	inttypes.h	netipx/	scs1/	time.h
argz.h	execinfo.h	kde/	netpacket/	search.h	tls.h
ar.h	fcntl.h	langinfo.h	netrom/	semaphore.h	ttyent.h
arpa/	features.h	lastlog.h	netrose/	setjmp.h	ucontext.h
asm/	fenv.h	libgen.h	nfs/	sgtty.h	ulimit.h
asm-generic/	FlexLexer.h	libintl.h	nl_types.h	shadow.h	unistd.h
asm-1486/	fmtmsg.h	libio.h	nptl/	signal.h	ustat.h
asm-x86_64/	fnmatch.h	limits.h	nss.h	spawn.h	utime.h
assert.h	fpu_control.h	link.h	obstack.h	stab.h	utmp.h
autosprintf.h	fstab.h	linux/	paths.h	stdint.h	utmpx.h
bits/	fts.h	locale.h	poll.h	stdio_ext.h	values.h
byteswap.h	ftw.h	malloc.h	printf.h	stdio.h	wait.h
C++/	_G_config.h	math.h	protocols/	stdlib.h	wchar.h
complex.h	gconv.h	mcheck.h	pthread.h	string.h	wctype.h
cpio.h	gdict-1.0/	memory.h	pty.h	strings.h	wordexp.h
crypt.h	getopt.h	mntent.h	pwd.h	stropts.h	X11/
ctype.h	gettext-po.h	monetary.h	python2.3/	sys/	xlocale.h
dirent.h	glob.h	mqueue.h	python2.4/	syscall.h	-



• /usr/lib

Este directorio incluye librerías compartidas y ficheros binarios pensados para no ser ejecutados directamente por los usuarios del sistema.

• /usr/local/

/usr/local/ es para uso del administrador del sistema cuando instala software localmente. Puede usarse para programas y datos que son compartibles entre un grupo de máquinas Este subdirectorio tiene una estructura similar a la del directorio /usr. Vemos su contenido:

alex@alexDebian: /usr/local							_AI		
Archivo	Editar	⊻er	Terminal	<u>S</u> olapas	Ayuda				
alex@ale Adobe/ alex@ale	exDebia bin/ exDebia	n:/us games n:/us	r/local\$ / inclu r/local\$	ls -F	/ <u>ear@</u>	sbin/	share/	src/	

• usr/sbin

Este directorio contiene comandos y programas no esenciales usados exclusivamente por el administrador de sistema.

Como se ha comentado, los comandos necesarios para la reparación, recuperación y otras funciones esenciales del sistema, se almacenan en /sbin. Veamos su contenido:

3				alex@al	exDebian: /usr/sbin	HU
Archivo	Editar	⊻er	Terminal	<u>S</u> olapas	Ayuda	
alex@alex	Debia	n:/us	r/sbin\$	ls -F		
a2dismod*					lpinfo*	
a2dissite	*				tpmove*	
a2enmod*					Lsusb*	
a2ensite*					mailer*	
ab*					make-ssl+cert*	111
ab.7@					mkboot*	810
accept*					mkinitramfs*	_
accessdb*					mkinitramfs-kpkg*	
acp1d*					mklost+found*	
addgnupgh	one*				mksmbpasswd*	
addgroup@					modhandler.py*	
add-shell	*				mount unidades red csic*	
adduser*					newusers*	
alsaconf*					nfsstat*	
alsactl*					nmbd*	
apache2*					no icmp*	
apache2ct	1=				nologin*	
apache2-s	sl-ce	rtifi	cate*		ownership*	
arp*					pac*	
arpd*					pam getenv*	
aspell-au	tobul	ldhas	h*		pam_tally*	
atd*					pppd	
blosdecod	e*				pppdump*	-

• /usr/src

Por lo general, en /usr/src (src de source o, fuente en castellano) se guarda el código fuente del Kernel del sistema.

Para comprobar si tenemos en nuestra distribución los fuentes del kernel instalados, deberíamos ver un enlace simbólico llamado linux.

4.3.- El directorio /var

Este directorio va a contener ficheros de datos variables y temporales, así como archivos spool (ficheros almacenados en "fila"en espera a ejecutarse, como por ejemplo colas de impresión).

Todos los log del sistema y los generados por los servicios instalados, se ubican dentro de la estructura jerárquica de /var. Esto quiere decir que el tamaño global de este directorio va ha crecer constantemente.

La utilidad de /var radica en poder detectar problemas para prevenirlos y solucionarlos. Es aconsejable montar en una nueva partición este directorio. Si no se pudiera, es preferible ubicar /var fuera de la partición raíz y de la partición /usr.

Distribución de algunos subdirectorios de /var

Veamos Su contenido:



/var/cache

Subdirectorio pensado para albergar datos de aplicaciones en cache (usados en un espacio breve de tiempo).

El sistema de paquetes de Debian (apt-get), mantiene y almacena todos los paquetes que nos hemos instalado con el gestor de paquetes Apt-get. Por ejemplo, si ejecutamos:

alex@alexDebian:# apt-get install nmap

Debian se bajará de algún repositorio especificado en /etc/apt/sources.list el archivo nmap_version.deb, lo almacenará en /var/cache/apt y lo instalará desde esta ruta. Posteriormente lo podemos borrar. Por defecto Debian almacena aquí todo los paquetes que nos hemos instalado con su gestor de paquetes Apt-get.

/var/lib

Encontramos aquí información sobre el estado variable de las aplicaciones.

/var/lock

Aquí se almacenan los ficheros que están bloqueados por el sistema.

/var/log

En /var/log se guardan los mensajes de registro generados por el sistema operativo y por diversos servicios.



Por ejemplo:

En /var/log/messages son los logs generados por el kernel, en /var/log/httpd/access_log encontramos quien (desde que ip) está accediendo a nuestro servidor Web y, en /var/log/wtmp encontraremos todos los accesos y salidas en el sistema.

Vemos su contenido:

a		alex@alexDebian: /var/log									
Archivo Edit	ar ⊻er <u>T</u> erminal	<u>S</u> olapas Ayud	la								
alex@alexDet	pian:/var/log\$ ls	- F									
acpid	dmesg	fsck/	mail.err	syslog							
apache2/	dmesg.0	gdm/	mail.info	user.log							
aptitude	dmesg.1.gz	installer/	mail.log	uucp.log							
auth.log	dmesg.2.gz	iptraf/	mail.warn	wtmp							
boot	dmesg.3.gz	kern.log	messages	Xorg.0.log							
btmp	dmesg.4.gz	ksymoops/	news/	Xorg.0.log.old							
cups/	dpkg.log	lastlog	pycentral.log	Xorg.20.log							
daemon.log	exim4/	lp-acct	samba/	Xorg.20.log.old							
debug	faillog	lp-errs	scrollkeeper.log		(even)						
dirmngr.log	fontconfig.log	lpr.log	sshd.log		1220						
alex@alexDet	bian:/var/log\$	91 - 48-55	85.52		-						

• /var/mail

Linux enviará aquí los archivos de correos de cada usuario del sistema.

• /var/run

/var/run contiene archivos con información del sistema que lo describen desde que se arrancó. Generalmente, se borrará todos los archivos que cuelgan de este subdirectorio al comenzar el proceso de arranque.

Estos archivos con información del sistema son los llamados "archivos identificados de procesos" ó PID, que guardan el identificador del proceso (Process ID).

Podemos ver aquí los archivos PID en un instante determinado en mi máquina:

unelleb

Instituto de nuevas Tecnologías

a		alex@alexDebi	ti 🗆		
Archivo Editar	<u>∨</u> er <u>T</u> erminal	<u>S</u> olapas Ayuda			
alex@alexDebia	an:/var/run\$ ls	- F			
acpid.socket=	crond.reboot	gdm.pid	klogd.pid	sshd/	
alsa/	cups/	hal/	lpd.pid	sshd.pid	111
apache2/	dbus/	hotkey-setup	motd	sudo/	
apache2.pid	dirmngr/	identd/	network/	syslogd.pid	
atd.pid	dirmngr.pid	iptraf/	rpc.statd.pid	utmp	
crond.pid	exim4/	john/	samba/		
alex@alexDebia	an:/var/run\$	12			-

/var/spool

/var/spool contiene ficheros almacenados en forma de "fila de trabajos", para un procesamiento posterior.

3	_41					
Archivo	Editar	⊻er	Terminal	Solapas	Ayuda	
alex@ale cron/ c alex@ale	xDebia ups/ xDebia	n:/va <mark>exim4</mark> n:/va	r/spool\$ / <mark>lpd/</mark> r/spool\$	ls -F	openoffice/ samba/	

Un ejemplo claro puede ser los trabajos que guarda la impresora para, posteriormente, ejecutarlos por un orden de llegada y/o prioridad.

/var/tmp

Algunos datos temporales se almacenan aquí y que, posiblemente, pueden aparecer en nuestra distribución GNU/Linux para no "saturar" el directorio /tmp. Existen otra serie de directorios que no especifica el estándar FSH, pero que son importantes. Los vemos

4.4.- Directorio /lost+found. "Perdidos y encontrados"

Las herramientas y utilidades para restaurar y/o reparar el sistema de archivos almacenan los datos en este directorio.

Es un espacio temporal donde se guardan los datos que se recuperan después de una caída del sistema.

Fijémonos que, normalmente en cada partición que creemos existirá un /lost+found en el nivel superior.

Por último, decir que este directorio existe sólo en distribuciones que tengan como sistemas de archivos ext2 o ext3.

4.5.- Directorio /proc

/proc es un sistema de archivos virtual. Se genera y actualiza dinámicamente, es decir, no se mantiene en el disco duro, se mantiene el la memoria RAM. Es el sistema quien lo crea y lo destruye.

Este directorio contiene información sobre los procesos, el núcleo e información relativa al sistema.

Vemos su contenido:

3				alex@	alexDebian: /	proc		4-E
Archivo	Editar	⊻er	Terminal	Solapas	Ayuda			
alex@al	exDebia	an:/pr	oc\$ ls -	F				
1/	3954/	4268/	4452/	4527/	6/	fb	mpt/	
101/	3960/	4269/	4454/	4529/	66/	filesystems	mtrr	
102/	4/	4270/	4457/	4536/	8/	fs/	net/	
103/	4020/	4272/	4470/	4537/	9/	ide/	partitions	
104/	4038/	4343/	4473/	4542/	9494/	interrupts	scsi/	
1478/	4845/	4344/	4478/	4567/	9519/	iomem	self@	
1579/	4053/	4353/	4482/	4626/	acp1/	ioports	slabinfo	
1850/	4054/	4370/	4485/	4632/	asound/	irq/	stat	
2/	4060/	4371/	4486/	4686/	buddyinfo	kallsyms	swaps	
2004/	4064/	4372/	4488/	4730/	bus/	kcore	sys/	
2770/	4077/	4373/	4503/	4732/	cmdline	key-users	sysrq-trigger	
2811/	4094/	4374/	4508/	4733/	cpuinfo	kmsg	sysvipc/	
2835/	4140/	4375/	4510/	5/	crypto	loadavg	tty/	
3/	4154/	4397/	4512/	5663/	devices	locks	uptime	
3202/	4155/	4441/	4514/	5664/	diskstats	meminfo	version	
3427/	4195/	4445/	4516/	5776/	dma	misc	vmstat	1.00
3429/	4245/	4447/	4518/	5798/	driver/	modules	zoneinfo	100
3431/	4252/	4450/	4524/	5799/	execdomains	mounts@		
alex@al	exDebia	an:/pr	ocș					-

Subdirectorios identificados por un número, corresponden a los PID de los procesos.



INSTALACION DE LINUX

La instalación paso a paso: Ubuntu

Una vez tenemos nuestro live-CD o live-USB con Ubuntu arrancamos con él del modo ya conocido, pero en esta ocasión seleccionamos Instalar Ubuntu o bien si estamos ante el escritorio, por haber arrancado en el modo de prueba, hacemos doble clic en el icono Instalar.



Se inicia entonces el asistente de instalación. Aunque no es estrictamente necesario es muy conveniente que durante la instalación tenga acceso a Internet. En ese caso marcamos las casillas "Descargar actualizaciones mientras se instala" e "Instalar software de terceros". Una vez hecho esto pulsamos en Continuar.

Instalar			
Preparando la instalación de Ubun	tu		
Para obtener los mejores resultados, asegúrese que es	te equipo:		
🎻 tiene al menos 4.4 GB de espacio en disco disponible	2		
🞻 está conectado a Internet		3	
S Descargar actualizaciones mientras se instala		A	
Ubuntu usa software de terceros para mostrar Flash, MP3 y algunos dispositivos inalámbricos. Parte de este software e los términos de la licencia incluida con la documentación de	otros archivos m es de código cerra el mismo.	nultimedia y p ado. El softwa	ara trabajar con ire está sujeto a
Minstalar este software de terceros			
El complemento Fluendo MP3 incluye la tecnologia de decodificacio Technicolor SA.	ón MPEG Layer-3 lice	nciada por Fraur	nhofer IIS y
	Salir	Atrás	Continuar

Si su ordenador dispone de una conexion Wifi y todavía no ha configurado la red, el asistente le dará ahora la oportunidad de hacerlo. Marque la casilla *Conectarse a esta red*, seleccione la red y escriba la contraseña de acceso.

Veremos entonces una pantalla que nos permite elegir dónde se instalará Ubuntu. Seleccionamos "Instalar Ubuntu junto a Microsoft Windows...". Si queremos que Ubuntu sea el único sistema operativo en nuestro disco duro marcamos "Reemplazar Microsoft Windows...". Una vez hecha la elección pulsamos en *Continuar*.

Este equ	ipo tiene instalado actualmente Microsoft Windows XP Professional. ¿Qué desea hacer
•	Instalar Ubuntu junto a Microsoft Windows XP Professional Se conservarán todos los documentos, música y demás archivos. Podrá elegir qué sistema operativo iniciar cada vez que arranque su equipo.
•	Reemplazar Microsoft Windows XP Professional con Ubuntu Aviso: Esto eliminarà todos sus programas, documentos, fotos, música y demás archivos de Microsoft Windows XP Professional.
• 1	Algo más Puede crear particiones, redimensionarlas, o elegir varias particiones para Ubuntu.

Importante: ¡Cuidado!

Si seleccionas la opción "Reemplazar Microsoft Windows...", como puedes ver en la imagen anterior, perderás tu instalación original de windows y todos los datos, documentos, etc que tengas en ese entorno. Esta acción no se puede deshacer, ya que se formatea el disco duro, por lo que los cambios serán irreversibles y no podrás recuperar la información de Windows. Recomendamos la primera opción "Instalar Ubuntu junto a Microsoft Windows...", para establecer la posibilidad de compartir ambos sistemas operativos y establecer un arranque dual. Piensa bien que es lo que quieres hacer.



En la parte superior de la pantalla que veremos a continuación se nos muestra la capacidad de nuestro disco. En la parte inferior veremos una posible división del disco. Si queremos cambiar la asignación de espacio situamos el cursor entre las dos mitades y arrastramos hacia la derecha o izquierda según queramos dar más espacio a Windows (cuadro de la izquierda) o a Ubuntu (cuadro de la derecha). En todo caso ambos cuadros tienen un límite hasta el que podemos reducirlos representado por el espacio que actualmente ocupa Windows, y el espacio mínimo que Ubuntu necesita para instalarse. Llegados a este límite no podremos seguir arrastrando. Una vez hecho esto pulsamos en *Instalar ahora*.

Se nos advierte ahora que se van a llevar a cabo las operaciones de cambio de asignación de espacio en el disco. Pulsamos en *Continuar*.



unelleb

Instituto de nuevas Tecnologías

Instalar

Inst	talar Ubuntu junto a Microsoft Windows XP Professiona cione la unidad: SCSI1 (0.0.0) (sda) - 120.6 GB ATA VBOX HARDDISK 117.3 © ¿Desea guardar los cambios previos a disco y continuar?	l SB V
Asign	Debe guardar los cambios previos al disco antes de q <mark>u</mark> e pueda seleccionar un nuevo tamaño de la partición. No puede deshacer esta operación. Por favor, tenga en cuenta que la operación de ajuste de tamaño puede tardar mucho tiempo.	
▶ Inst	Volver: Continuar	юга

Comienza ahora un proceso de trabajo simultáneo. Mientras Ubuntu cambia el tamaño de asignación en el disco, le da formato y copia los archivos necesarios se nos irán preguntando los datos necesarios para personalizar la instalación. En primer lugar debemos seleccionar nuestra zona horaria. Para ello hacemos clic en el lugar adecuado del mapa o bien escribimos "Madrid" en la caja de texto al pie de la ventana.





Continuación seleccionamos nuestra distribución de teclado.

Instalar	
Distribución del teclado	
Elija la distribución del teclado: Dhivehi Dzongkha Eslovaco Esloveno Español Español (latinoamericano) Esperanto Estonio Faroés	Español Español - Asturiano (español, con H de medio punto Español - Catalán (español, con L de medio punto) Español - Español (Dvorak) Español - Español (Macintosh) Español - Español (eliminar teclas muertas) Español - Español (incluir tilde muerta) Español - Español (teclas muertas de Sun)
Escriba aquí para probar su teclado Detectar distribución del teclado	Atrás Continuar

Tenemos ahora que facilitar una serie de datos para identificarnos en el sistema. Los datos que se nos piden son bastante evidentes. Vale la pena detenerse en algunos a los que quizá no esté habituado:

Iniciar sesión automáticamente. Si marcamos esta casilla Ubuntu arrancará directamente sin pedirnos nuestro nombre de usuario y contraseña. Si no desea tener que poner su nombre de usuario y contraseña cada vez que arranque Ubuntu marque la casilla "Iniciar sesión automáticamente".

			1000
Sunombre:	Antonio Saorín		-
El nombre de su equipo:	MiUbuntu	1	
	El nombre que usa cuan	do habla con otros equipos.	
Introduzca un nombre de usuario:	saorin	1	
Introduzca una contraseña:	000000	Contraseña débil	
Confirme su contraseña:			
	🔿 Iniciar sesión aul	tomáticamente	
	Solicitar mi cont	raseña para iniciar sesió	n
	🗌 Cifrar mi carp	oeta personal	
		Atrás	Continuar

A partir de ahora y hasta el final de la instalación se nos irán mostrando una serie de diapositivas con algunas de las características más sobresalientes de Ubuntu. Si así lo indicó se descargarán las actualizaciones disponibles con lo que el proceso de instalación se alargará considerablemente, dependiendo de la velocidad de su conexión a Internet.

Instalar

Escriba y haga presentaciones gratis LibreOffice es un paquete de oficina . Master Deck 27 Sep 2011.odp - LibreOffice Impress libre con todo lo que necesita para crear impresionantes documentos, hojas de 🥦 • 🊘 🗷 🚅 📝 🖺 🖷 🦈 🌃 🕺 🖄 🛍 🎍 🐟 • 🖉 • 💣 💷 • 🚳



Si todo va bien verá la pantalla final de instalación pidiendo que reinicie el sistema.

🙁 Instalación terminada	
La instalación se la nueva instalac	e ha completado. Necesita reiniciar el equipo para poder usar ción. Reiniciar ahora

Después de la instalación **ARRANQUE CON GRUB**

Después del reinicio, y a partir de ahora siempre que encendamos el ordenador, veremos el cargador de arrangue GRUB que nos permitirá elegir qué sistema operativo ejecutar. Si no hacemos nada se cargará por defecto Ubuntu (un poco más abajo veremos cómo cambiarlo). Para seleccionar cualquiera de las otras opciones usamos las teclas del cursor.

unelleb

Instituto de nuevas Tecnologías

GNU GRUB versión 1.99–21ubuntu3

Ubuntu, con Linux 3.2.0-23-generic-pae Ubuntu, con Linux 3.2.0-23-generic-pae (modo recuperación) Memory test (memtest86+) Memory test (memtest86+, serial console 115200) Microsoft Windows XP Professional (on /dev/sda1) Use las teclas * y + para seleccionar qué entrada se resalta. Pulse Entrar para arrancar el S.O. resaltado, «e» para editar órdenes antes de arrancar o «c» para línea de órdenes. La entrada resaltada se ejecutará automáticamente en 9s.

Una vez concluida la carga veremos la pantalla de registro de Ubuntu. Escribimos aquí la contraseña que pusimos en el momento de la instalación y pulsamos [Intro]

Mit	Jount													0	iii es	- (M)	21:39	Ċ
		Ant	onio	Sao	rín		0											
		Ko	ntra	seña							A							
		Saci	ón c		litar	to												
		Jesi	onic	Jenn	vica													
				- 12														
	Ľ	IDUr	icu.	12.0	04 L1	rs												

Administración de usuarios y grupos

En nuestro centro tanto el profesorado como el alumnado acostumbran a personalizar su escritorio, incluyendo accesos directos, carpetas y demás vínculos que considera importantes y cómodos en su tarea diaria. Pero todo esto incomoda, en muchas ocasiones, a otros usuarios que también utilizan el mismo equipo. Esta situación es especialmente patente en las aulas de informática, donde parece que el alumnado compite en crear ámbitos de trabajo extravagantes e inútiles, con la pérdida de tiempo en la carga del sistema operativo y que posteriormente repercute en nuestro trabajo al tener que estar reinstalando equipos, eliminando programas, etc.

Para resolver el problema anterior es necesario disponer de una base de datos de usuarios y grupos donde poder asignar o denegar permisos de acceso a los recursos autenticados de nuestro centro. Ello permitirá personalizar los entornos de trabajo de cada usuario, dando solución al problema planteado tanto por el profesorado como por el alumnado.

La administración de usuarios y grupos solamente puede realizarlas el usuario root utilizando los comandos de gestión de usuarios. Las tareas y los comandos para realizarlas son:

- Creación de usuarios / useradd
- Modificación de usuarios / usermod
- Eliminación de usuarios / userdel
- Creación de grupos / groupadd
- Modificación de grupos / groupmod
- Eliminación de grupos / groupdel
- Añadir usuarios a un grupo / adduser
- Quitar usuarios de un grupo / deluser

Creación de usuarios

El comando useradd permite añadir un usuario indicando como parámetros la información particular para crear el usuario en la misma líne de comandos. La sintaxis es:

//Crear

usuario

useradd [opciones] nombre-usuario

Entre las opciones más destacables tenemos:

- -g: **Grupo principal** que queremos tenga el usuario (debe existir previamente)
- -d: Carpeta home del usuario. Suele ser /home/nombre-usuario
- -m: Crear carpeta home si es que no existe.
- -s: Intérprete de comandos (shell) del usuario. Suele ser /bin/bash



Ejemplo, si deseamos crear un usuario llamado 'pedro' cuyo grupo principal sea 'profesores', cuya carpeta home sea /home/pedro y su intérprete de comandos sea /bin/bash, ejecutaremos el siguiente comando:

// Crear un usuario

sudo useradd -g profesores -d /home/pedro -m -s /bin/bash pedro

De esta manera habremos creado al usuario pedro y su carpeta home. Si no utilizamos la opción -m, no se creará la carpeta home del usuario; en tal caso tendríamos que crearla manualmente. Tan solo nos quedará establecer su contraseña con el comando passwd:

// Establecer la contraseña del usuario

sudo passwd pedro

Entonces el sistema nos preguntará dos veces la contraseña que queremos asignar a pedro.

El comando useradd permite crear muchos usuarios automáticamente mediante archivos de comandos (scripts).

Se recomienda que el nombre de usuario sea en minúsculas y además de letras también puede contener números y algún signo como guiones normales y guiones bajos. Debemos recordar que unix distingue entre mayúsculas y minúsculas, es decir, Pepe es distinto de pepe.

Modificación de usuarios

Se utiliza el comando **usermod** y permite cambiar el nombre del usuario, su carpeta home, su intérprete de comandos, los grupos a los que pertenece y algunos otros parámetros.

// Cambiar el home de un usuario

sudo usermod -d /home/carpeta_pedro pedro



Eliminación de usuarios

Se realiza con el comando **userdel** seguido del nombre del usuario. Con la opción -r eliminará también su carpeta home, ejemplo:

// Eliminación de un usuario

sudo userdel -r pedro

Eliminaría el usuario pedro y su carpeta home.

Creación de grupos

El comando **groupadd** permite añadir un grupo indicando como parámetro el nombre del grupo. Ejemplo, si deseamos crear un grupo llamado 'alumnos' ejecutaremos:

// Añadir un grupo

sudo groupadd alumnos

Modificación de grupos

El comando **groupmod** permite modificar el nombre de un grupo o el gid del mismo. La sintaxis es: sudo groupmod [-g nuevo-gid] [-n nuevo-nombre] nombre-grupo, ejemplo:

// Cambiar el gid del grupo profesores

sudo groupmod -g 2000 profesores



Eliminación de grupos

Se realiza con el comando groupdel seguido del nombre del grupo, ejemplo:

// Eliminación de un grupo

sudo groupdel profesores

Eliminaría el grupo profesores. Si algún usuario tuviera dicho grupo como grupo primario, el comando groupdel no eliminará el grupo.

Añadir usuarios a un grupo

Se utiliza el comando **adduser** seguido del nombre del usuario y del nombre del grupo al que queremos añadirle, ejemplo:

// Añadir a 'juan' al grupo 'profesores'

sudo adduser juan profesores

Quitar usuarios de un grupo

Se utiliza el comando deluser seguido del nombre del usuario y del nombre del grupo del que queremos quitarle, ejemplo:

// Quitar a 'juan' del grupo 'profesores'

sudo deluser juan profesores



Para más información de todos estos comandos se puede consultar la ayuda del manual ejecutando man seguido del nombre del comando, ejemplo man adduser

Herramienta gráfica de administración de usuarios

Ubuntu dispone de una herramienta gráfica de administración de usuarios que es '**users-admin**'. Para ejecutarla podemos abrir una consola de root y ejecutar users-admin o si hemos iniciado sesión como root, podemos pulsar Alt+F2 y ejecutar users-admin.



MANEJO DE ARCHIVOS Y DIRECTORIOS A TRAVES DE COMANDOS

Más allá de interfaces gráficas como KFM, Linux dispone de un numeroso conjunto de utilidades para manipular archivos (individualmente o en grupos) tipeando comandos en una terminal. A través de esta modalidad de trabajo es posible alcanzar un alto nivel de flexibilidad y versatilidad para realizar operaciones de cualquier grado de complejidad. Los comandos utilizados más frecuentemente se describen brevemente a continuación. Para un mayor nivel de detalle, se sugiere consultar el manual en línea de Linux (ver más adelante el comando **man**) y especialmente la documentación del intérprete de comandos utilizado

(usualmente, el bash).

COMANDOS LINUX PARA LA GESTION DE ARCHIVOS Y DIRECTORIOS

ср

Propósito

El comando cp es un abreviatura de copy (copiar); permite copiar archivos y directorios. Para copiar un archivo se usa el siguiente mandato: *Sintaxis:*

cp [Opciones] archivo_fuente directorio_destino cp [Opciones] archivo_fuente archivo_destino

Opciones:

- -a conserva todos los atributos de los archivos.
- -b hace un backup antes de proceder a la copia.
unetteb

Instituto de nuevas Tecnologías

- -d copia un vínculo pero no el fichero al que se hace referencia.
- -i pide confirmación antes de sobreescribir archivos.
- -p conserva los sellos de propiedad, permisos y fecha.
- -R copia los archivos y subdirectorios.
- -s crea enlaces en vez de copiar los ficheros.
- -u únicamente procede a la copia si la fecha del archivo origen es posterior a la del destino.
- -v muestra mensajes relacionados con el proceso de copia de los archivos.

Descripción

El comando cp copia un archivo a otro. También puede copiar varios ficheros en un directorio determinado.

Ej.

cp manual_linux_v1 ../../../doc/linux

En este ejemplo copia el archivo manual_linux en un directorio dos niveles más arriba del actual, en el directorio doc/linux

mv

Propósito

Modifica el nombre de los archivos y directorios moviéndolos de una ubicación a otra.

Sintaxis

mv [Opciones] fuente destino

Opciones

- -d hace una copia de seguridad de los archivos que se van a mover o renombrar.
- -f elimina los archivos sin solicitar confirmación.
- -v pregunta antes de sobreescribir los archivos existentes.

Descripción



El comando mv se puede utilizar para modificar el nombre o mover un archivo de un directorio a otro. Trabaja tanto con archivos como con los directorios. *Ej.*

mv manual_linux_v1 manuales/linux

mv manual_linux_v1 manual_linux_v1_doc

mv manual_linux_cap1 manual_linux_cap2 manual_linux_cap2 /manual/linux

rm

Propósito

Elimina uno más archivos (puede eliminar un directorio completo con la opción -r).

Sintaxis

rm [Opciones] archivos

Opciones

- -f elimina todos los archivos sin preguntar.
- -i pregunta antes de eliminar un archivo.
- -r elimina todos los archivos que se encuentran en un subdirectorio y por último borra el propio subdirectorio.
- -v muestra el nombre de cada archivo antes de eliminarlo.

Descripción

El comando rm se utiliza para borrar los archivos que se le especifiquen. Para eliminar un fichero ha de tener permiso de escritura en el directorio en el que se encuentra.

Ej.

rm manual_linux_v1 rm –r documentos/

mkdir

Propósito



crear directorios.

Sintaxis

mkdir [Opciones] nombre_directorio

Opciones

- -m modo, asigna la configuración de permisos especificada al nuevo directorio.
- -p crea directorios emparentados (en caso de que no existan).

Descripción

El comando mkdir se utiliza para crear un directorio específico. *Ej.* mkdir manuales

rmdir

Propósito

Elimina un directorio (siempre y cuando esté vacío).

Sintaxis

rmdir [Opciones] directorio

Opciones

• -p elimina cualquier directorio emparentado que este vacío.

Descripción

El comando rmdir elimina los directorios vacíos. Si tiene algún contenido, tendrá que utilizar el comando rm –r para eliminar el directorio y sus contenidos.



Ej. rmdir manual

ls

Propósito

Listar el contenido de un directorio.

Sintaxis Is [Opciones] [nombre_directorio o archivo]

Opciones

- -a muestra todos los archivos. Incluyendo a los ocultos.
- -b muestra los caracteres no imprimibles de los nombres de los ficheros utilizando un código octal.
- -c ordena los archivos de acuerdo con la fecha de creación.
- -d muestra una lista en la que aparecen los directorios como si fuesen archivos (en vez de mostrar su contenido).
- -f muestra el contenido del directorio sin ordenar.
- -i muestra información de i-node.
- -l muestra la lista de archivos con formato largo y con información detallada (tamaño, usuario, grupo, permisos etc.).
- -p añade un carácter al nombre del archivo para indicar a que tipo pertenece.
- -r coloca la lista en orden alfabético inverso.
- -s muestra el tamaño (kb) de cada archivo próximo al solicitado.
- -t ordena la lista de acuerdo con la fecha de cada fichero.
- -R muestra una lista con el contenido del directorio actual y de todos sus subdirectorios.

Descripción

El comando ls muestra el contenido de un directorio determinado. Si se omite el nombre del directorio, mostrará el contenido del directorio en el que se encuentre. Por defecto, ls no muestra el nombre de los archivos cuyo nombre comience con un punto; para verlos tendrá que utilizar la opción –a.

Ej.

- ls –a
- ls –l
- ls –la

cd

Propósito

Cambiar de directorio.

Sintaxis

cd [directorio]

Opciones

Ninguna

Descripción

Si escribe cd sin ningún nombre de directorio como argumento, se cambiará al directorio home del usuario. En cualquier otro caso se moverá al directorio indicado, si existe.

pwd

Propósito

Mostrar la ruta del directorio de trabajo actual.

Sintaxis.

pwd

Opciones



Ninguna

Descripción

El comando pwd imprime el directorio de trabajo (aquel en el que actualmente se está trabajando).

chmod

Propósito

Modifica los permisos de uno o más archivos o directorios.

Sintaxis

chmod [Opciones] [permiso_descripción] archivo

Opciones

- -c muestra los archivos a los que se les han modificado los permisos.
- -f hace que no aparezca en pantalla ningún mensaje de error.
- -v muestra los cambios efectuados en los permisos de archivos.
- -R cambia los permisos de los archivos de todos los subdirectorios.

Permisos_descripción

Quien Acción Permiso

Quien	Acción	Permiso
u: usuario g: grupo o: otros a: todos	+: agregar -: quitar =: asignar	r: lectura w: escritura x: ejecutar s: ajustar con el ID del usuario.



Ej.

chmod u+xr manual_linux

El usuario tendrá los permisos de lectura y ejecución sobre el archivo manual_linux

Descripción

Para utilizar eficazmente el comando chmod, debe especificarse la configuración de los permisos de acuerdo a la tabla de permisos_descripción.

Por ejemplo para que todos tengan permiso de lectura en un determinado archivo se tipea, *chmod a+r nombre_archivo*. También se podría haber tipeado chmod u=r,g=r,o=r nombre_archivo.

Otra forma de modificar los permisos es a través de un número octal de 3 cifras una cifra por cada grupo de permisos, este número surge de realizar la suma de los permisos que se les quiere asignar de acuerdo a los siguientes valores:

- Permiso de lectura r = 4
- Permiso de escritura w = 2
- Permiso de ejecución x = 1

Y si no se le concede cualquier permiso el valor asignado es 0.

El formato para utilizar chmod especificando los permisos por medio de números es el siguiente.

chmod permiso_usuario permiso_grupo permiso_otros

Ejemplo, supongamos que creamos el archivo permiso.txt y queremos que el usuario tenga todos los permisos, el grupo los permisos de lectura y ejecución y finalmente que el resto de los usuarios tenga sólo el permiso de ejecución.

Para el usuario: lectura r = 4, escritura w = 2, ejecución x = 1; sumados = 7

Para el grupo: lectura r = 4, escritura w = 0, ejecución x = 1; sumados = 5

Para el resto de los usuarios: lectura r = 0; escritura w = 0, ejecución x = 1; sumados = 1 Entonces el comando seria: chmod 751 permisos.txt

En la lista detallada de los archivos de un directorio (usando el comando ls), los permisos de lectura escritura y ejecución del usuario, grupo y otros se mostrarán a través de la secuencia rwxrwxrwx, cuando algún permiso no está activado aparece un guión en su reemplazo.

cat

Propósito

Muestra el contenido de un archivo utilizando la salida estándar (pantalla).

```
Sintaxis
```

cat [-benstvA] archivos

Opciones

- -b números de líneas que no estén en blanco.
- -e muestra el final de una línea (como \$) y todos los caracteres no imprimibles.
- -n numera todas las líneas de salida, comenzando por el 1.
- -s sustituye varias líneas en blanco por una sola.
- -t muestra las tabulaciones como ^A.
- -v muestra los caracteres no imprimibles.
- -A muestra todos los caracteres (incluidos los no imprimibles).

Descripción

Normalmente, cat se utiliza para mostrar el contenido de un archivo o para *concatenar* varios dentro de un mismo fichero. Por ejemplo,

```
cat archivo1, archivo2, archivo3 > todo
```

Combina los tres archivos dentro de uno solo llamado todo.

find

Propósito

Muestra una lista con los archivos que coinciden con un criterio especifico.

Sintaxis

find [ruta] [opciones]

Opciones

- -depth procesa, en primer lugar, el directorio en el que se encuentra y luego sus subdirectorios.
- -maxdepyh n restringe la búsqueda a n niveles de directorios.

unelleb

Instituto de nuevas Tecnologías

- -follow procesa los directorios que se incluyen dentro de los enlaces simbólicos.
- -name modelo localiza los nombres de los archivos que coinciden con el modelo propuesto.
- -ctime n localiza los nombres de los archivos creados n días atrás.
- -user nombre_usuario nombre_usuario localiza los archivos pertenecientes al usuario especifico.
- -group nombre_grupo localiza los archivos pertenecientes al grupo específico.
- -path ruta localiza a los archivos cuya ruta coincide con el modelo propuesto.
- -perm modo localiza los archivos con los permisos especificados.
- -size +nK localiza los archivos cuyo tamaño (en kilobytes) es mayor de especificado.
- -print imprime el nombre de los archivos que encuentra.
- -exec comando [opciones] {} \; ejecuta el comando especificado analizando el nombre del archivo localizado.

Descripción

El comando find es de gran utilidad cuando se quiere localizar todos los archivos que coinciden con algún criterio. Si escribe find sin ningún argumento, la salida mostrará un listado en el que aparecen los archivos de todos los subdirectorios de la carpeta en la que se encuentre.

Para ver todos los archivos cuyo nombre termine con .gz, tendrá que escribir:

find . -name "*.gz ".

Para buscar a partir del directorio /usr/doc todos los archivos con extensión *bak* y eliminarlos, utilizar el comando:

find /usr/doc -name "*.bak" -exec rm -f {} \;

en donde la secuencia {} se substituirá por el nombre completo de cada archivo encontrado.

Grep

Propósito

Busca en uno o más archivos las líneas que coincidan con una expresión regular (modelo de búsqueda).

Sintaxis

grep [opciones] modelo archivos



Opciones

- -N muestra N líneas que contienen el modelo de búsqueda señalado.
- -c muestra el número de líneas que contienen el modelo de búsqueda.
- -f archivo lee las opciones del archivo especificado.
- -i ignora letras
- -l muestra los nombres de los archivos que contienen un modelo.
- -q devuelve el número de línea siguiente a aquellas en las que se encuentra el modelo de búsqueda.
- -v muestra las líneas que no contienen el modelo de búsqueda.

Descripción

El comando localiza el modelo de búsqueda en los archivos especificados. El modelo es una expresión regular en los archivos especificados que tienen sus propias reglas. Generalmente se utiliza para buscar una secuencia de caracteres en uno o más archivos de texto. *Ejemplo*

grep Juan ListadoDeAlumnos.txt

OTROS COMANDOS DE LINUX

man: Muestra por pantalla secciones del manual del usuario.

Formato: man Nombre del comando.

Ej: man ls.

mesg: Habilita o deshabilita la comunicación entre usuarios por medio de *write*.

Formato : mesg [n/y].

LINUX ADMINISTRACIÓN Nivel 1



Ipr: Imprime el contenido de un archivo.

Formato: Ipr [Opción] Archivo

Se consideran las principales opciones:

- -P cola Indica la cola de impresión a utilizar.
- -n<número>: Indica la cantidad de copias a imprimir, por defecto siempre es 1.
- -R: Remueve el archivo después de realizada la impresión.

tree: Lista todos los directorios a partir del directorio actual o del directorio indicado. *Formato*: tree [Directorio].

tty: Muestra el número de la terminal donde está trabajando el usuario. *Formato*: tty

WhO: Visualiza los usuarios que están activos en el sistema, sin ningún tipo de argumento éste comando muestra los nombres de usuario, número de terminal y horario de conexión por cada usuario activo del sistema. Utilizando los argumentos who am i el comando muestra con que nombre de usuario está usted conectado.

Formato: who [Opción]

Write: Envía mensajes a otros usuarios hasta que se digite "Control D". La recepción de estos mensajes puede ser deshabilitada por el usuario utilizando el comando MESG. *Formato:* write Usuario Terminal



Configuración de RED

Para empezar veremos que interfaces tenemos y su nomenclatura eth (Ejecutamos en una terminal)

ivan@	CanBarça: ~
Archivo	Editar Ver Buscar Terminal Ayuda
ivan@Can eth0	<pre>Barça:~\$ ifconfig -a Link encap:Ethernet direcciónHW 00:17:31:47:67:b2 ACTIVO DIFUSIÓN MULTICAST MTU:1500 Métrica:1 Paquetes RX:0 errores:0 perdidos:0 overruns:0 frame:0 Paquetes TX:0 errores:0 perdidos:0 overruns:0 carrier:0 colisiones:0 long.colaTX:1000 Bytes RX:0 (0.0 B) TX bytes:0 (0.0 B) Interrupción:23 Dirección base: 0x4000</pre>
lo	Link encap:Bucle local

ifconfig -a

Una vez sepamos que interfaz vamos a configurar pasamos a editar lo siguiente

sudo nano /etc/network/interfaces





Justo unas líneas debajo de iface lo inet loopback Agregamos lo siguiente para los siguientes casos

Direccion IP Estatica

Agregamos estos valores de acuerdo a lo como quieran ustedes configurar su interfaz, la gateway no es obligatorio configurarla además aquí también pueden configurar la dirección de broadcast

auto eth0 iface eth0 inet static address 192.168.100.10 netmask 255.255.255.0 gateway 192.168.100.10 broadcast 192.168.100.255

Reiniciamos la interfaz de red con el siguiente comando

sudo /etc/init.d/networking restart

Si quieren agregar un servidor DNS lo agregamos en **/etc/resolv.conf** asi: nameserver 200.80.30.30





Direccion IP Dinamca

Si tenemos un servidor DHCP que nos proporciona la IP automáticamente escribiremos lo siguiente (repetimos el paso de ifconfig –a para saber la interfaz)

auto eth0

iface eth0 inet dhcp

	ivan@	CanBarç	a: ~				N	
Α	rchivo	Editar	Ver	Buscar	Terminal	Ayuda	*	
	GNU na	no 2.2.	4	Arc	hivo: /et	c/networ	k/interfaces	
# #	This f and how	ile des w to ac	cribe tivat	s the n e them.	etwork in For more	terfaces informa	available or tion, see int	n yo terf
# au if	The lo to lo ace lo	opback inet l	netwo oopba	rk inte ck	rface			
au if	to eth ace et	0 h0 inet	dhcp					

Reiniciamos la interfaz de red con el siguiente comando

sudo /etc/init.d/networking restart

Por ultimo para comprobar que la dirección ip se agregó correctamente comprobamos con el comando ifconfig

Nota: Para cada interfaz solo puede tener un tipo de configuración o estática o dinámica, no pueden estas las dos



SISTEMA DE PERMISOS

En los sistemas operativos tipo **POSIX** cada elemento del sistema de archivos, como archivos, directorios, enlaces simbólicos, etc., tiene la característica de poseer permisos que lo ubican dentro del mismo. Éstos sirven como uno más de los niveles de seguridad del sistema operativo al impedir que cualquier usuario pueda leer, escribir, ejecutar o acceder a dichos archivos y directorios de manera arbitraria. Estos permisos vistos de manera básica son: **lectura** (**r**, *read*), **escritura** (**w**, *write*) y **ejecución**(**x**, *execution*) y se agrupan en bloques (**rwx**) para 3 diferentes clases (usuario, grupo y otros).

Los permisos de acceso de cada archivo y directorio del sistema son mostrados por un conjunto de 10 caracteres, los cuales proporcionan información acerca del tipo de elemento, junto con permisos para el usuario y grupo propietario para leer, escribir y ejecutar, como se muestra en el siguiente ejemplo:

-rwxr-xr-x 1 fulano fulano 0 jul 31 18:11 algo.txt

La asignación de permisos de acceso (de lectura, escritura y ejecución) pueden gestionarse a través de modos, los cuales consisten de combinaciones de números de tres dígitos — usuario, grupo y otros— que son manipulados a través de **chmod** y**setfacl**.

Notación simbólica.

El esquema de notación simbólica se compone de 10 caracteres, donde el primer carácter indica el tipo de archivo:

Valor	Descripción
-	Archivo regular.
d	Directorio.
b	Archivo especial como dispositivo de bloque.
с	Archivo de carácter especial
1	Enlace simbólico.
р	Tubería nombrada (FIFO)
S	Zócalo de dominio (socket)

Como se mencionó anteriormente, cada clase de permisos es representada por un conjunto de tres caracteres. El primer conjunto de caracteres representa la clase del usuario, el



segundo conjunto de tres caracteres representa la clase del grupo y el tercer conjunto representa la clase de «otros» (resto del mundo). Cada uno de los tres caracteres representa permisos de lectura, escritura y ejecución, respectivamente y en ese orden.

Ejemplos:

Permisos	Descripción
drwx r-x r-x	Directorio con permiso 755
crw-rw-r	Archivo de carácter especial con permiso 664.
srwx rwx r-x	Zócalo con permiso 775.
prw- rw- r	Tubería (FIFO) con permiso 664.
-rw- r r	Archivo regular con permiso 644.

Notación octal.

La notación octal consiste de valores de tres a cuatro dígitos en base-8. Con la notación octal de tres dígitos cada número representa un componente diferente de permisos a establecer: clase de usuario, clase de grupo y clase de otros (resto del mundo) respectivamente. Cada uno de estos dígitos es la suma de sus bits que lo componen (en el sistema numeral binario). Como resultado, bits específicos se añaden a la suma conforme son representados por un numeral:

- El Bit de ejecución (acceso en el caso de directorios) añade 1 a la suma.
- El bit de escritura añade 2 a la suma
- El bit de lectura añade 4 a la suma.

Estos valores nunca producen combinaciones ambiguas y cada una representa un conjunto de permisos específicos. De modo tal puede considerarse la siguiente tabla:

Valor Permiso Descripción

		L L L L L L L L L L L L L L L L L L L
0		Nada
1	x	Sólo ejecución de archivos o acceso a directorios
2	-w-	Sólo escritura
3	-wx	Escritura y ejecución de archivos o acceso a directorios
4	r	Sólo lectura
5	r-x	Lectura y ejecución de archivos o acceso a directorios
6	rw-	Lectura y escritura
7	rwx	Lectura, escritura y ejecución de archivos o acceso a directorios

Cabe señalar que el permiso 3 (wx) es el resultado de 1+2 (w+x), que el permiso 5 (rx) es el resultado de 4+1 (r+x), que el permiso 6 (rw) es el resultado de 4+2 (r+w) y que el permiso 7 (rwx) es el resultado de 4+2+1 (r+w+x).



Máscara de usuario.

La máscara de usuario (*umask*, abreviatura de *user mask*) es una función que establece los permisos predeterminados para los nuevos archivos y directorios creados en el sistema. Puede establecerse en notación octal de tres o cuatro dígitos o bien en notación simbólica. Puede establecerse cualquier valor para *umask*, pero debe tomarse en consideración que ésta jamás permitirá crear nuevos archivos ejecutables.

Cuando se utiliza la notación octal de cuatro dígitos, el primer dígito siempre corresponde a los permisos especiales, pero el valor de éste siempre será 0; el segundo dígito corresponde a la máscara de la clase otros; el tercer dígito corresponde a la máscara para la clase de grupo; y el cuarto dígito corresponde a la máscara para la clase de usuario.

```
----- Permisos especiales (siempre es 0 en umask)
| .----- Clase de otros
| | .----- Clase de grupo
| | | .---- Clase de usuario
| | | ↓
↓
↓
0 0 2 2
```

El valor de la máscara de usuario, que se asigna ejecutando **umask**, corresponde a los bits contrarios del permiso predeterminado que se quiera asignar. Es decir, si por ejemplo se quiere asignar una máscara de usuario equivalente a 0775 (rwxrwxr-x), el valor de la máscara de usuario corresponderá a 0002 (el resultado de la operación 777 menos 775), que será lo mismo que definir u=rwx,g=rwx,o=rx.

Si por ejemplo se quiere asignar una máscara de usuario equivalente a 0744 (rwxr--r--), el valor de la máscara de usuario corresponderá a 0033 (el resultado de la operación 777 menos 744), que será lo mismo que definir u=rwx,g=r,o=r.

Los valores nunca producen combinaciones ambiguas y cada una representa un conjunto de permisos específicos. De modo tal puede considerarse la siguiente tabla:

Valor octal	Valor simbólico	Descripción
0	rwx	Lectura, escritura y acceso a directorios
1	rw-	Lectura y escritura
2	r-x	Lectura y acceso a directorios
3	r	Sólo lectura
4	-wx	Escritura y acceso a directorios
5	-w-	Sólo escritura
6	x	Sólo acceso a directorios
7		Nada

El valor predeterminado de la máscara de usuario del sistema en CentOS, Fedora™, Red Hat™ Enterprise Linux, openSUSE™ y SUSE™ Linux Enterprise es **0022**, es decir se asigna permiso 0755 (-rwxr-xr-x) para nuevos directorios y 0644 (-rw-r--r--) para nuevos archivos. El



sistema jamás permite crear nuevos archivos con atributos de ejecución. El valor predeterminado se define en una variable de entorno del archivo **/etc/profile** y puede ser cambiado por el que el administrador del sistema considere pertinente. El valor también puede establecerse por usuario en el archivo **~/.bash_profile** (CentOS, Fedora™ y Red Hat™ Enterprise Linux) o bien en el archivo **~/.profile** (openSUSE™ y SUSE™ Linux Enterprise).

El valor predeterminado de la máscara de usuario utilizado por **useradd**, para la creación de directorios de inicio de usuarios, se define en el archivo **/etc/login.defs**.

En CentOS, Fedora[™] y Red Hat[™] Enterprise Linux el valor predeterminado de la máscara de usuario utilizada por **useradd** es 0077, es decir que los directorios de inicio de cada usuario que sea creado en el sistema tendrá un permiso 0700 (rwx-----).

En openSUSE[™] y SUSE[™] Linux Enterprise el valor predeterminado de la máscara de usuario utilizada por **useradd** es 0022, es decir 0755 (rwxr-xr-x), debido a que la variable **UMASK** está deshabilitada con una almohadilla en el archivo **/etc/login.defs**, pues se recomienda se defina ésta variable en el archivo **/etc/default/useradd**. Ejemplo:

```
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
UMASK=0077
```

Para determinar el valor en notación octal para la máscara de usuario predeterminada del sistema, ejecute **umask** sin opciones ni argumentos.

umask

Para determinar el valor en notación simbólica para la máscara de usuario predeterminada del sistema, ejecute **umask** con la opción **-S** (mayúscula), sin argumentos.

umask -S

Para cambiar la mascara de usuario en la sesión activa y procesos hijos, se requiere ejecutar **umask** con el valor octal deseado. En el siguiente ejemplo, se definirá 0002 (0775, rwxrwxr-x) como máscara de usuario:

umask 0002

Lo anterior también se puede hacer utilizando notación simbólica: umask u=rwx,g=rwx,o=rx

Control de procesos

La más simple definición de un proceso podría ser que es una instancia de un programa en ejecución (corriendo). A los procesos frecuentemente se les refiere como tareas. El contexto de un programa que esta en ejecución es lo que se llama un proceso. Este contexto puede ser mas procesos hijos que se hayan generado del principal (proceso padre), los recursos del sistema que este consumiendo, sus atributos de seguridad (tales como su propietario y permisos de archivos asi como roles y demás de SELinux), etc.

Linux, como se sabe, es un sistema operativo multitarea y multiusuario. Esto quiere decir que múltiples procesos pueden operar simultáneamente sin interferirse unos con los otros. Cada proceso tiene la "ilusión" que es el único proceso en el sistema y que tiene acceso exclusivo a todos los servicios del sistema operativo.

Programas y procesos son entidades distintas. En un sistema operativo multitarea, múltiples instancias de un programa pueden ejecutarse sumultáneamente. Cada instancia es un proceso separado. Por ejemplo, si cinco usuarios desde equipos diferentes, ejecutan el mismo programa al mismo tiempo, habría cinco instancias del mismo programa, es decir, cinco procesos distintos.

Cada proceso que se inicia es referenciado con un número de identificación único conocido como Process ID PID, que es siempre un entero positivo. Prácticamente todo lo que se está ejecutando en el sistema en cualquier momento es un proceso, incluyendo el shell, el ambiente gráfico que puede tener múltiples procesos, etc. La excepción a lo anterior es el kernel en si, el cual es un conjunto de rutinas que residen en memoria y a los cuales los procesos a través de llamadas al sistema pueden tener acceso.

El comando es el que permite informar sobre el estado de los procesos. esta basado en el sistema de archivos /proc, es decir, lee directamente la información de los archivos que se encuentran en este directorio. Tiene una gran cantidad de opciones, incluso estas opciones varían dependiendo del estilo en que se use el comando. Estas variaciones sobre el uso de son las siguientes:

- Estilo UNIX, donde las opciones van precedidas por un guión -
- Estilo BSD, donde las opciones no llevan guión
- Estilo GNU, donde se utilizan nombres de opciones largas y van precedidas por doble guión --



Sea cual sea el estilo utilizado, dependiendo de las opciones indicadas, varias columnas se mostrarán en el listado de procesos que resulte, estas columnas pueden ser entre muchas otras, las siguientes (y principales):

p o PID	Process ID, número único o de identificación del proceso.										
P o PPID	Parent Process ID, padre del proceso										
U o UID	ser ID, usuario propietario del proceso										
t o TT o TTY	Terminal asociada al proceso, si no hay terminal aparece entonces un '?'										
T o TIME	empo de uso de cpu acumulado por el proceso										
c o CMD	l nombre del programa o camndo que inició el proceso										
RSS	Resident Sise, tamaño de la parte residente en memoria en kilobytes										
SZ o SIZE	Tamaño virtual de la imagen del proceso										
NI	Nice, valor nice (prioridad) del proceso, un número positivo significa menos tiempo de procesador y negativo más tiempo (-19 a 19)										
C o PCPU	Porcentaje de cpu utilizado por el proceso										
STIME	Starting Time, hora de inicio del proceso										
S o STAT	Status del proceso, estos pueden ser los siguientes										
	• R runnable, en ejecución, corriendo o ejecutándose										
	• S sleeping, proceso en ejecución pero sin actividad por el momento, o esperando por algún evento para continuar										
	• T sTopped, proceso detenido totalmente, pero puede ser reiniciado										
	• Z zombie, difunto, proceso que por alguna razón no terminó de manera correcta, no debe haber procesos zombies										
	• D uninterruptible sleep, son procesos generalmente asociados a acciones de IO del sistema										
	• X dead, muerto, proceso terminado pero que sigue apareciendo, igual que los Z no deberían verse nunca										

Las opciones completas de las encuentras en las páginas del manual (), o escribiendo en la terminal, y para ver un resumen de sus opciones más comunes usa :

#>	pshelp	
***	****** simple selection ********	******** selection by list ********
-A	all processes	-C by command name
-N	negate selection	-G by real group ID (supports names)
-a	all w/ tty except session leaders	-U by real user ID (supports names)
-d	all except session leaders	-g by session OR by effective group name
-e	all processes	-p by process ID
Т	all processes on this terminal	-s processes in the sessions given
а	all w/ tty, including other users	-t by tty
g	OBSOLETE DO NOT USE	-u by effective user ID (supports names)
r	only running processes	U processes for specified users
Х	processes w/o controlling ttys	t by tty

LINUX ADMINISTRACIÓN Nivel 1



******	⊧* output fo	rmat :	******	******	***	<pre> long options ***********</pre>		
-o,o user-defined -f				GroupUserpidcolsppid				
−j,j job	control s	sign	al	group	ı	ısersidrowsinfo		
-0,0 prel	oaded -o v	virt	ual memory	cumula	ıtiv	veformatdeselect		
-I, I long	u	user	-oriented	sort -	-tt	yforestversion		
-F extr	afull X	regi	sters	headir	ng -	no-headingcontext		
		****	**** misc	options *	***>	*****		
-V, V sho	v version	L	list format	t codes	f	ASCII art forest		
-m, m, -L, -	「,H threads	S	children in	n sum	-у	change -I format		
-M,Z sec	urity data	С	true comman	nd name	-c	scheduling class		
-w,w wid	e output	n	numeric WC	HAN, UID	-H	process hierarchy		

A continuación algunos cuantos ejemplos de con la salida recortada.

≻# ps	-e	(-е	e muest	ra	todos	los p	roce	sos)			
PID	TTY		TI	ME	CMD						
1	?		00:00:	01	init						
2	?		00:00:	00	kthre	add					
3	?		00:00:	00	migra	tion/0)				
4	?		00:00:	00	ksoft	irqd/0)				
#≻ ps	-ef	(-	f mues	tra	a opcio	ones c	omplo	etas)			
UID		PID	PPID	С	STIME	TTY		TIM	E CMD		
root		1	0	0	10:12	?	(0:00:00	1 init	[5]	
root		2	0	0	10:12	?	(0:00:00	0 [kth	readd]	
 root		6130	5662	0	10:24	pts/0	(0:00:00	0 su -		
root		6134	6130	0	10:24	pts/0	(0:00:00	0 -basł	h	
sergon	ı	6343	5604	0	10:28	?	(0:00:00	0 kio 1	file [kdein	it] file /home/sergon/tmp/ksocket-sergon/kl
root		6475	6134	0	10:38	pts/0	(0:00:00	0 ps -6	ef	
#> ps	-eF	(-	F mues	tra	a opcio	ones c	lamo	etas ex	tra)		
UID		PID	PPID	С	SZ	RSS	PSR	STIME	TTY	TIME	CMD
root		1	0	0	412	556	1	16:59	?	00:00:01	init [5]
root		2	0	0	0	0	1	16:59	?	00:00:00	[kthreadd]
sergon	ı	8326	8321	0	902	1272	0	17:07	?	00:00:00	/bin/sh /usr/lib/firefox-2.0.0.8/run-
mozill	a. sł	n /usr	/lib/f								
sergon	ı	8331	8326	4	53856	62604	0	17:07	?	00:00:50	/usr/lib/firefox-2.0.0.8/mozilla-firefox-bin
sergon	1	8570	7726	2	15211	37948	0	17:17	?	00:00:10	quanta
#> ps	ax	(f	ormato	BS	SD sin	guión	aı	nuestra	todos,	, x sin mos [.]	trar tty)

PID TTY	ST	AT TI	ME COMMAI	ND							
1 ?	Ss	0:	01 init	[5]							
2 ?	S<	0:	00 [kthr	eadd]							
3 ?	S<	0:	00 [migr	ation/0]							
4 ?	S<	0:	00 [ksof	tirqd/0]							
				• • -							
#> ps aux	(formation)	ato BSD	sin guid	ón, u muest	ra usuar	ios y d	emás co	olumnas)			
USER	PID %C	PU %MEM	VSZ	RSS TTY	STAT	START	TIME	COMMAND			
root	1 0.	0 0.0	1648	556 ?	Ss	16:59	0:01	init [5]			
root	2 0.	0 0.0	0	0 ?	S<	16:59	0:00	[kthreadd]			
root	30.	0 0.0	0	0 ?	S<	16:59	0:00	[migration/0]			
root	4 0.	0 0.0	0	0 ?	S<	16:59	0:00	[ksoftirqd/0]			
root	50.	0 0.0	0	0 ?	S<	16:59	0:00	[migration/1]			
#> ps -eo	user,pi	d, tty	(-o outp	out persona	lizado,	se indi	can los	s campos separados p	or coma,	ver ps	help o
ps L)			-	-							
USER	PID TT										
root	1 ?										
root	2 ?										
sergon	8570 tt	y 1									
root	8876 pt	s/1									
#> ps −eH	(muest	ra árbo	l de proc	cesos)							
#> ps axf	(lo mi	smo en 1	formato E	BSD)							
#> ps -ec	(el com	nando d	que se es	sta ejecuta	ndo, sin	la rut	a, solo	o el nombre real)			
#> ps −el	(muest	ra forma	ato largo	o de varias	columna	s, muy	práctic	co)			
#> ps L	(No mu	estra pi	rocesos,	lista todo	s los có	digos d	e forma	atos)			
-			,			-					

pstree

Muestra los procesos en forma de árbol, te da las opciones más comunes. Recomiendo uses lo uses con la opción -A y -G para que te un árbol con líneas con líneas estilo ASCII y de terminal VT100 respectivamente, puedes añadir también -u para mostrar entre paréntesis al usuario propietario del proceso:



```
|-gam_server(sergio)
|-gconfd-2(sergio)
|-hald(haldaemon)----hald-runner(root)----hald-addon-acpi(haldaemon)
| -hald-addon-cpuf
| -hald-addon-inpu
| -hald-addon-stor
|-httpd---8*[httpd(apache)]
|-2*[ifplugd]
|-ipw3945d
|-kaccess(sergio)
```

kill

El comando, que literalmente quiere decir matar, sirve no solo para matar o terminar procesos sino principalmente para enviar señales (signals) a los procesos. La señal por default (cuando no se indica ninguna es terminar o matar el proceso), y la sintaxis es, siendo PID el número de ID del proceso. Asi por ejemplo, es posible enviar una señal de STOP al proceso y se detendrá su ejecución, después cuando se quiera mandar una señal de Continuar y el proceso continuara desde donde se quedó.

#> kill -l (lista todas las posibles señales que pueden enviarse a un proceso)

```
1) SIGHUP
             2) SIGINT
                        3) SIGQUIT
                                     4) SIGILL
 5) SIGTRAP
             6) SIGABRT
                          7) SIGBUS
                                      8) SIGFPE
 9) SIGKILL
            10) SIGUSR1
                          11) SIGSEGV
                                      12) SIGUSR2
13) SIGPIPE 14) SIGALRM
                         15) SIGTERM
                                       16) SIGSTKFLT
17) SIGCHLD
            18) SIGCONT
                          19) SIGSTOP
                                        20) SIGTSTP
21) SIGTTIN
            22) SIGTTOU
                          23) SIGURG
                                       24) SIGXCPU
25) SIGXFSZ
            26) SIGVTALRM 27) SIGPROF
                                         28) SIGWINCH
29) SIGIO
           30) SIGPWR
                        31) SIGSYS
                                    34) SIGRTMIN
35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3 38) SIGRTMIN+4
39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12
47) SIGRTMIN+13 48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14
51) SIGRTMAX-13 52) SIGRTMAX-12 53) SIGRTMAX-11 54) SIGRTMAX-10
55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7 58) SIGRTMAX-6
59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX
```

La lista previa presenta una lista de todas las posibles señales que pueden mandarse a un proceso y estas pueden ser invocadas a través del número de la señal o de su código, por ejemplo:

```
#> kill -9 11428 (termina, mata un proceso completamente)
```



#> kill -SIGKILL 11428 (Lo mismo que lo anterior)

Las señales más comunes son la 19 y 20 que detienen momentáneamente la ejecución de un proceso o programa, 18 la continua, 1 que es la señal de hang up que obliga al proceso a releer sus archivos de configuración estando en ejecución y 9 que termina rotundamente un proceso.

Killall

El comando, que funciona de manera similar a kill, pero con la diferencia de en vez de indicar un PID se indica el nombre del programa, lo que afectará a todos los procesos que tengan ese nombre. Asi por ejemplo si se tienen varias instancias ejecutándose del proxy server squid, con eliminará todos los procesos que se esten ejecutando con el nombre 'squid'

#> killall -l (lista de posibles señales)

#> killall -HUP httpd (manda una señal de "colgar", detenerse releer sus archivos de configuración y reiniciar)

#> killall -KILL -i squid (manda señal de matar a todos los procesos squid pero pide confirmación en cada uno)

Nice

Permite cambiar la prioridad de un proceso. Por defecto, todos los procesos tienen una prioridad igual ante el CPU que es de 0. Con es posible iniciar un programa (proceso) con la prioridad modificada, más alta o más baja según se requiera. Las prioridades van de -20 (la más alta) a 19 la más baja. Solo root o el superusuario puede establecer prioridades negativas que son más altas. Con la opción -l de es posible observar la columna NI que muestra este valor.

#> nice (sin argumentos, devuelve la prioridad por defecto)

#> nice -n -5 *comando* (inicia *comando* con una prioridad de -5, lo que le da más tiempo de cpu)

Renice

0



Así como establece la prioridad de un proceso cuando se inicia su ejecución, permite alterarla en tiempo real, sin necesidad de detener el proceso.

#> nice -n -5 yes (se ejecuta el programa 'yes' con prioridad -5) (dejar ejecutando 'yes' y en otra terminal se analiza con 'ps') #> ps -el

F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD 4 S 0 12826 12208 4 75 -5 - 708 write_pts/2 00:00:00 yes

#> renice 7 12826
12826: prioridad antigua -5, nueva prioridad 7
#> ps -el
F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
4 S 0 12826 12208 4 87 7 - 708 write_pts/2 00:00:15 yes

(Obsérvese el campo NI en el primer caso en -5, y en el segundo con renice quedó en 7, en tiempo real)

Instalación y actualización de software

Debian: paquetes DEB

Debian tiene herramientas interactivas como tasksel, que permiten escoger unos subconjuntos de paquetes agrupados por tipo de tareas: paquetes para X, para desarrollo, para documentación, etc., o como dselect que nos permite navegar por toda la lista de paquetes disponible (hay miles), y escoger aquellos que queramos instalar o desinstalar.

En el nivel de línea de comandos dispone de dpkg, que es el comando de más bajo nivel, para gestionar directamente los paquetes DEB de software [Deb02], típicamente dpkg i paquete.deb para realizar

la instalación. Pueden realizarse todo tipo de tareas, de información, instalación, borrado o cambios internos a los paquetes de software.

Otro nivel intermedio lo presentan las herramientas APT (la mayoría son comandos apt-xxx), donde se gestionan los paquetes a través de una lista de paquetes actuales y disponibles a partir de varias fuentes

de software, ya sea desde los propios CD de la instalación, sitios ftp o web (HTTP). Esta gestión se hace de forma transparente, de manera que el sistema es independiente de las fuentes de software.

La configuración del sistema APT se efectúa desde los archivos disponibles en /etc/apt, donde /etc/apt/sources.list es la lista de fuentes

disponibles; un ejemplo podría ser:

deb http://http.us.debian.org/debian stable main contrib non-free

deb http://non-us.debian.org/debian-non-US stable/non-US main contrib non-free

deb http://security.debian.org stable/updates main contrib non-free

debsrc http://http.us.debian.org/debian stable main contrib non-free

debsrc http://nonus.debian.org/debian-non-US stable non-US

#Sources Oficiales de Debian STABLE "Woody"



deb http://ftp.debian.org/debian/ woody main non-free contrib debsrc http://ftp.debian.org/debian/ woody main non-free contrib deb http://non-us.debian.org/debian-non-US woody/non-US main contrib non-free debsrc http://non-us.debian.org/debian-non-US woody/non-US main contrib non-free

NOTA: Los paquetes DEB de Debianes quizás el sistema de instalación más potente existente en GNU/Linux. Una prestación destacable es la independencia del sistema de las fuentes de los paquetes (mediante APT).

En que hay recopiladas algunas de las fuentes "oficiales" para una software, así como las actualizaciones que estén disponibles. Básicamente, se especifica el tipo de fuente (web en este caso), el sitio, la Debian Woody, desde donde se pueden obtener los paquetes de versión de la distribución (stable o woody), y categorías del software Nota que se buscará (libre, o contribuciones de terceros o comercial).

Los paquetes de software están disponibles para las diferentes distribuciones Debian, existen paquetes para las versiones stable, testing, y unstable. El uso de unos u otros determina el tipo de distribución.

Pueden tenerse fuentes de paquetes mezcladas, pero no es muy recomendable, ya que se podrían dar conflictos entre las versiones de las diferentes distribuciones.

Una vez tenemos las fuentes de software configuradas, la principal herramienta para manejarlas en nuestro sistema es apt-get, que nos permite instalar, actualizar o borrar desde el paquete individual, hasta actualizar la distribución entera. Existe también un reemplazo a

apt-get, llamado aptitude, cuya interfaz de opciones es prácticamente igual; como ventaja aporta una mejor gestión de dependencias de los paquetes y permite una interfaz interactiva.

Algunas funciones básicas de apt-get:

• Instalación de un paquete particular:

apt-get install paquete

LINUX ADMINISTRACIÓN Nivel 1



• Borrado de un paquete:

apt-get remove paquete

• Actualización de la lista de paquetes disponibles:

apt-get update

• Actualización de la distribución, podríamos efectuar los pasos:

apt-get update

apt-get upgrade

apt-get dist-upgrade

Mediante el último proceso, podemos mantener nuestra distribución actualizada permanentemente, actualizando los paquetes instalados y verificando las dependencias con los nuevos. Una herramienta útil para construir esta lista es apt-spy, que intenta buscar los sitios oficiales más rápidos. Por otro lado, podemos buscar las fuentes oficiales (podemos configurarlas con apt-setup), o bien copiar algún fichero de fuentes disponible. Software adicional (de terceros) puede necesitar añadir otras fuentes más, se pueden obtener listas de sitios de fuentes disponibles

(por ejemplo: http://www.apt-get.org)

Red Hat: paquetes RPM

El sistema de paquetes RPM creado por Red Hat supone un paso adelante, ya que incluye la gestión de dependencias y tareas de configuración del software.

Además, el sistema guarda una pequeña base de datos con los paquetes ya instalados, que puede consultarse y se actualiza con las nuevas instalaciones.



Los paquetes RPM, por convención, suelen usar un nombre como: paquete-version-rev.arq.rpm

donde paquete es el nombre del software, version es la versión del software, rev suele ser la revisión del paquete RPM, que indica las veces que se ha construido, y arq, la arquitectura a la que va destinado el paquete, ya sea Intel (i386, i586, i686) u otras como Alpha, Sparc,PPC. La arquitectura Noarch suele usarse cuando es independiente, por ejemplo, un conjunto de scripts, y src en el caso de que se trate de paquetes de código fuente. La ejecución típica incluye la ejecución de rpm, las opciones de la operación a realizar, junto con uno o más nombres de paquetes por procesar juntos.

Ejemplo

El paquete: ces que se ha construido, y arq, la arquitectura a la que va destinado apache-1.3.19-23.i686.rpm el paquete, ya sea Intel (i386, i586, i686) u otras como Alpha, Sparc, indicaría que se trata del software Apache (el servidor web), en su versión 1.3.19, revisión del paquete RPM 23, para arquitecturas Pentium II

Las operaciones típicas con los paquetes RPM incluyen: Información del paquete: se consulta sobre el paquete una información determinada, se usa la opción -qp acompañada del nombre del paquete instalado. Si el paquete no ha sido instalado todavía la opción sería - q acompañada de la información que se quiera, y si se quiere preguntar a todos los paquetes a la vez, la opción sería -qa. Por ejemplo, preguntas a un paquete instalado.

Instalación: simplemente rpm -i paquete.rpm. Esto podrá realizarse siempre que se estén cumpliendo las dependencias del paquete, ya sea software previo o bibliotecas que deberían estar instaladas. En caso de no cumplirlo, se nos listará qué software falta, y el nombre del paquete que lo proporciona. Puede forzarse la instalación (a riesgo de que no funcione) con las opciones -- force o --nodeps, o simplemente se ignora la información de las dependencias.

Actualización: equivalente a la instalación pero comprobando primero que el software ya existe rpm -u paquete.rpm. Se encargará de borrar la instalación previa.

• Verificación: durante el funcionamiento normal del sistema, muchos de los archivos instalados cambian. En este sentido, RPM permite verificar los archivos para detectar las



modificaciones, bien por proceso normal, bien por algún error que podría indicar datos corrompidos. Mediante rpm -V paquete verificamos un paquete concreto, y mediante rpm -Va los verificará todos.

• Eliminación: borrar el paquete del sistema RPM de paquete; si hay dependencias, puede ser necesario eliminar otros primero. El sistema RPM también ofrece la posibilidad de trabajar con paquetes remotos, que pueden descargarse desde servidores FTP o web, sólo hay que utilizar la sintaxis ftp:// o http:// para dar la localización del paquete.

Por ejemplo:

rpm -i ftp://sitio/directorio/paquete.rpm

nos permitiría descargar el paquete desde el sitio ftp o web proporcionado, con su localización de directorios, y proceder en este caso a la instalación del paquete.

Hay que vigilar con la procedencia de los paquetes, y sólo utilizar fuentes de paquetes conocidas y fiables, ya sea del propio fabricante de la distribución, o sitios en los que confiemos. Normalmente se nos ofrecen junto con los paquetes, alguna "firma" digital de estos para que podamos comprobar la autenticidad, suelen utilizarse las firmas md5, u otros sistemas como GPG. Hay también en Internet diferentes almacenes de paquetes RPM, donde están disponibles para diferentes distribuciones que usen o permitan el formato RPM.

En cuanto a las distribuciones, en Red Hat (y también en Fedora), RPM es el formato por defecto de paquetes y el que usa ampliamente la distribución para las actualizaciones, y la instalación de software.

En Debian se utiliza el formato DEB, hay soporte para RPM (existe el comando rpm), pero sólo para consulta o información de paquetes. En el caso de que sea imprescindible instalar un paquete, se recomienda utilizar la utilidad alien, que permite convertir formatos de paquetes, en este caso de RPM a DEB, y proceder a la instalación con el paquete convertido.

Filtros y Redirecciones

Filtros

Los filtros nos permiten personalizar o modificar la salida de la ejecución de un comando de



diversas maneras.

sort

Muestra por la salida estándar el contenido de un fichero ordenado de manera alfabética. Se pueden utilizar los siguientes parámetros:

- -n: emplear orden numérico.
- -d: emplear orden alfabético (por defecto).
- -f: ignorar mayúsculas y minúsculas.
- -r: ordenación inversa.
- -k: ordenar por un determinado campo.
- -t: definir el carácter de separación de campo (espacio por defecto).

sort doc1

sort -r doc1

shuf

Muestra por la salida estándar las líneas del contenido de un fichero de manera aleatoria:

shuf doc1

more

Muestra por la salida estándar el contenido de un fichero pero paginado. Similar a la orden cat pero página a página y no entero de una sola vez.

more doc1

less

Es un comando más completo que more, ya que muestra la salida paginada pero añadiendo más funcionalidad, por ejemplo, avance y retroceso de páginas. Está basado en los comandos del editor vi, al pulsar cualquier tecla aparecerán dos puntos tras los cuales escribiremos la orden. La tecla q es para salir.

unelleb

Instituto de nuevas Tecnologías

less doc1

tail

Muestra la parte final de un documento. Con el parámetro –n le indicamos cuántas líneas finales deseamos visualizar.

tail -n 10 doc1

tail -10 doc1

Si al número le ponemos el signo + indica que se debe mostrar desde ese número de línea hasta el final, es decir, de la línea 10 al final en nuestro caso.

tail -n +10 doc1

head

Muestra la parte inicial de un documento. Con el parámetro –n le indicamos cuántas líneas iniciales deseamos visualizar.

head -n 10 doc1

head -10 doc1

Podemos mostrar todas las líneas menos el número que digamos utilizando un número negativo.

head -n -10 doc1

grep

Busca en un fichero la cadena de texto indicada y la muestra por la salida estándar.

- [...] : conjunto de caracteres.
- [a-z]: con el guíon, rangos en conjuntos de caracteres.
- [^ ...] : complemente del conjunto de caracteres.

- . : cualquier carácter.
- : la expresión se repite 0 o más veces.
- + : la expresión se repite 1 o más veces.
- { n, m } : la expresión se repite entre m y n veces.
- ^ : comienzo de la línea.
- \$: final de la línea.

grep "A" doc1		
grep [ahj] doc1		
grep [a-z] doc1		
grep [a-zA-Z0-9] doc1		
grep ^aeiou doc1		
grep aeiou\$ doc1		

Además se pueden utilizar los siguientes parámetros:

- -i: ignorar mayúsculas y minúsculas.
- -n: numerar las líneas.
- -c: no mostrar la salida, sólo contar el número de coincidencias.
- -h: suprimir nombre de fichero coincidente (sólo mostar la línea).
- -l: suprimir la línea coincidente (sólo mostrar nombres de ficheros).
- -v: invertir la salida (buscar líneas NO coincidentes).

cut

Muestra, de un fichero ordenado por campos separados por un determinado carácter, sólo las columnas que nosotros le indiquemos. Como ejemplo utilizaremos el fichero /etc/passwd en el que cada fila posee unos campos separados por dos puntos ":". Indicaremos los campos que queramos mostrar mediante su número separado por comas (pero sin espacios en blanco), sabiendo que el primer campo es el 1. También podemos indicar un intervalo con el guión "-".

cut -d ":" -f1 /etc/passwd

cut -d ":" -f1,2,6 /etc/passwd

LINUX ADMINISTRACIÓN Nivel 1

cut -d ":" -f1-4 /etc/passwd

Para cortar de un número de carácter específico a otro.

cut -c 1-5 /etc/passwd

WC

Muestra el número de líneas -I, palabras -w, caracteres -m o bytes -c de un documento.

wc –l doc1			
wc –w doc1			
wc –m doc1			

wc –c doc1

uniq

Elimina las líneas repetidas de un archivo.

unia	doc1		
~···~			

tr

Cambia, borra o comprime caracteres.

- "set1" "set2": va cambiando cada caracter de set1 por el correspondiente de set2.
- -d: borra caracteres indicados.
- -s: comprime a uno sólo secuencias repetidas de los caracteres indicados.
- -c: complementa el conjunto de caracteres indicados.

tr "set1" "set2"

tr "a-z" "A-Z"



tee

Este filtro recoge datos de la entrada estándar y los redirige a la salida estándar y a un fichero especificado, tanto en operaciones de escritura como de añadidura. Esta orden suele usarse en procesos largos para observar y registrar la evolución de los resultados.

configure 2>&1| tee salida.txt

configure 2>&1| tee -a salida.txt

REDIRECCIONES

El redireccionamiento se utiliza para modificar la salida o entrada estándar, normalmente tras la ejecución de un comando en modo consola. La entrada estándar, la salida estándar y la salida de error se asocian a los programas mediante tres ficheros con los cuales se comunican con otros procesos y con el usuario. Estos tres ficheros son:

- stdin (entrada estándar): A través de este descriptor de fichero los programas reciben datos de entrada. Normalmente stdin está asociado a la entrada del terminal en la que está corriendo el programa, es decir, al teclado. Cada descriptor de fichero tiene asignado un número con el cual podemos referirnos a él dentro de un script, en el caso de stdin es el 0.
- stdout (salida estándar): es el descriptor de fichero en el que se escriben los mensajes que imprime el programa. Normalmente estos mensajes aparecen en la pantalla para que los lea el usuario. Su descriptor de fichero es el número 1.
- stderr (salida de error): es el descriptor de fichero en el que se escriben los mensajes de error que imprime el programa. Normalmente coincide con stdout. Tiene como descriptor de fichero el número 2.



Redireccionamiento de entrada: Con él podremos introducir los datos que nos solicite el comando mediante un fichero o la salida de otro comando. Por ejemplo, el comando siguiente

cat < fichero

Redireccionamiento de salida: Con él podremos enviar la salida de un comando hacia un fichero, otro comando o un dispositivo.

ls –l > fichero

ls –l > /dev/null

Redireccionamiento de adición: Es un redireccionamiento de salida con la peculiaridad de que la salida la añade al destino indicado, normalmente un fichero, y por tanto no borra su contenido.

man ls >> fichero

ls /etc >> /dev/tty4

Redireccionamiento de errores: De esta manera sólo redirigimos los errores que produzca el comando hacia otro dispositivo de salida que no sea la pantalla. Puede ser de adición también.

cp doc1 doc2 2> /dev/null

cp doc1 doc2 2>> fichero_error


Se pueden combinar la salida estándar con la de errores.

cp –v /boot/* /root/copia/ >> boot.log 2>> error.log

Redireccionamiento completo: De esta manera redirigimos con un solo comando tanto la salida normal como la salida de errores al mismo destino, puede ser de adición también. Los siguientes ejemplos son similares en su resultado.

cp –v /boot/* /root/copia/ >> boot.log 2>> boot.log

cp -v /boot/* /root/copia/ &>> boot.log

Redireccionamiento por parámetros: De esta manera redirigimos a un destino que ya se ha nombrado anteriormente en el mismo comando, el número indica la posición de ese destino. Vemos como el ejemplo anterior también se podría escribir de la siguiente manera.

cp –v /boot/* /root/copia/ >> boot.log 2>> &1

TUBERÍAS, PIPES O ENCADENAMIENTO DE COMANDOS

Es posible introducir en una misma línea dos o más órdenes separadas por el carácter tubería o pipe (|). De esta forma podemos ejecutar un comando y hacer que el resultado de éste sea la entrada para otro, que a su vez puede servir de entrada a un tercero.

ls –l | wc –l

man Is | more

echo -e "password\npassword" | passwd usuario



Observamos cómo podemos utilizar los filtros en la salida de otros comandos. Introduciendo el carácter \n simulamos el intro que el comando pudiera solicitar por teclado al usuario.

Proceso de Arranque y Niveles de Ejecución

Descripción general del proceso típico de arranque en gnu/Linux

En Linux, el flujo de control durante el arranque es desde la **BIOS**, al gestor de arranque y al núcleo (*kernel*). El núcleo inicia el planificador (para permitir la multitarea) y ejecuta el primer espacio de usuario (es decir, fuera del espacio del núcleo) y el programa de inicialización (que establece el entorno de usuario y permite la interacción del usuario y el inicio de sesión), momento en el que el núcleo se inactiva hasta que sea llamado externamente.

La etapa del cargador de arranque no es totalmente necesaria. Determinadas BIOS pueden cargar y pasar el control a Linux sin hacer uso del cargador. Cada proceso de arranque será diferente dependiendo de la arquitectura del procesador y el *BIOS*.

1. El BIOS realiza las tareas de inicio específicas de la plataforma de hardware.

2. Una vez que el hardware es reconocido y se inicia correctamente, el BIOS carga y ejecuta el código de la partición de arranque del dispositivo de arranque designado, que contiene la fase 1 de un gestor de arranque Linux. La fase 1 carga la fase 2 (la mayor parte del código del gestor de arranque). Algunos cargadores pueden utilizar una fase intermedia (conocida como la fase 1.5) para lograr esto, ya que los modernos discos de gran tamaño no pueden ser totalmente leídos sin código adicional.

3. El gestor de arranque a menudo presenta al usuario un menú de opciones posibles de arranque. A continuación, carga el sistema operativo, que descomprime en la memoria, y establece las funciones del sistema como del hardware esencial y la paginación de memoria, antes de llamar a la función start_kernel().

4. La función start_kernel() a continuación realiza la mayor parte de la configuración del sistema (interrupciones, el resto de la gestión de memoria, la inicialización del dispositivo, controladores, etc), antes de continuar por separado el proceso inactivo y planificador, y el proceso de *Init* (que se ejecuta en el espacio de usuario).

5. El planificador toma control efectivo de la gestión del sistema, y el núcleo queda dormido (inactivo).

6. El proceso Init ejecuta secuencias de comandos (*Scripts*) necesarios para configurar



todos los servicios y estructuras que no sean del sistema operativo, a fin de permitir que el entorno de usuario sea creado y pueda presentarse al usuario con una pantalla de inicio de sesión.

En el apagado, *Init* es llamado a cerrar toda las funcionalidades del espacio de usuario de una manera controlada, de nuevo a través de secuencias de comandos, tras lo cual el *Init* termina y el núcleo ejecuta el apagado.

• niveles de ejecución

El sistema operativo GNU/Linux puede aprovechar los **niveles de ejecución** a través de los programas del proyecto sysvinit. Después de que el núcleo Linux haarrancado, el programa init lee el archivo /etc/inittab para determinar el comportamiento para cada **nivel de ejecución**. A no ser que el usuario especifique otro valor como un parámetro de autoarranque del núcleo, el sistema intentará entrar (iniciar) al **nivel de ejecución** por defecto.

La mayor parte de usuarios de sistemas puede comprobar el **nivel de ejecución** actual con cualquiera de los comandos siguientes:

# runlevel	# como usuario root
\$ who -r	# como cualquier usuario



Niveles de ejecución típicos en Linux

la mayoría de las distribuciones Linux, definen los siguientes **niveles de** ejecución adicionales:

Los 7 niveles de ejecución (runlevels) estándars

Nivel de ejecución	Nombre o denominación	Descripción
0	Alto	Alto o cierre del sistema (Apagado).
1	Modo de usuario único (Monousuario)	No configura la interfaz de red o los demonios de inicio, ni permite que ingresen otro usuarios que no sean el usuario root, sin contraseña. Este nivel de ejecución permite reparar problemas, o hacer pruebas en el sistema.
2	Multiusuario	Multiusuario sin soporte de red.
3	Multiusuario con soporte de red.	Inicia el sistema normalmente.
4	Multiusuario con soporte de red.	Igual que el 3.
5	Multiusuario gráfico (X11)	Similar al nivel de ejecución 3 + display manager.
6	Reinicio	Se reinicia el sistema.



X-windows / Entornos gráficos

¿Qué es X-windows?

El protocolo X-windows fue desarrollado a mediados de los años 80 como respuesta a la necesidad de un interfaz gráfico transparente sobre todo para sistemas Unix.

X-windows es el encargado de visualizar la información de manera gráfica y es totalmente independiente del sistema operativo (los sistemas Unix/Linux no necesitan de X-windows para funcionar, pudiendo trabajar en modo texto). La gran diferencia entre X-windows y la interfaz gráfica de otros sistemas operativos es que X-windows distribuye el procesamiento de aplicaciones, especificando un enlace cliente-servidor. El cliente X especificara "Que hacer" al servidor X, que se encargara de "Como hacerlo".

Pero dejémonos de teoría y pasemos a un ejemplo práctico. La gran ventaja de X-windows es que el servidor X de una aplicación y el cliente X donde trabajamos no tienen por qué estar en la misma máquina. Podemos estar utilizando X-windows en nuestra máquina, conectarnos a otra remota, ejecutar un programa en esta máquina remota y utilizar/ver este programa en nuestra maquina local. Todo esto independientemente de la plataforma/sistema operativo que el ordenador remoto utilice. Como veis las posibilidades son muchas y potentes.



¿Cómo configuro las X-windows?

Para utilizar el sistema en modo grafico es necesario instalar X-windows en tu sistema. Existen servidores comerciales y bajo licencia GNU, en estos apartados nos basaremos en el Xwindows bajo licencia GNU, XFree, ya que es el que se incluye en todas las distribuciones de Linux.

Una vez instaladas las X-windows tendremos que configurarlas para que funcionen en nuestro sistema. El fichero de configuración de las X-windows se llama XF86Config y generalmente se encuentra en el directorio /etc/X11/.

Unos datos que tienes que saber antes de configurar X-windows son:



- Datos técnicos de la tarjeta gráfica que utilizas, o por lo menos el nombre de la tarjeta.
 El programa /usr/X11R6/bin/SuperProbe os puede ayudar a obtener información sobre vuestra tarjeta.
- Datos técnicos del monitor que tienes, resoluciones/frecuencias de refresco verticales/horizontales a la que puede trabajar. Este punto es importante para evitar problemas posteriores con vuestro monitor.

Antes de lanzar X-windows mediante el comando startx debemos asegurarnos de la correcta configuración de los diferentes parámetros del mismo (teclado, ratón, tarjeta de vídeo, etc.). Hay diferentes métodos para hacer esto.

En primer lugar tenemos los programas de configuración semi-automática. Según la distribución de Linux que utilicemos tendremos a nuestra disposición una serie de programas que nos permitirán realizar la configuración de XWindow de una manera más o menos sencilla:

- /usr/X11R6/bin/xf86config Todas las distribuciones: Programa en modo texto.
- /usr/X11R6/bin/XF86Setup SuSe, Redhat, Slackware: Programa en modo gráfico, tiene que arrancarse en una consola texto.
- /usr/X11R6/bin/Xconfigurator Redhat: Programa semi-gráfico con menús.

Estos programas nos preguntarán cosas básicas sobre el teclado, el monitor, la video memoria o la tarjeta de vídeo instalada (lo harán cuando no sean capaces de detectarlas por sí mismos). Debemos responder siempre la opción más acertada y en cuanto a monitores, nunca especificar características superiores de las que disponemos. Un caso muy habitual en tarjetas muy nuevas es no encontrar un soporte exacto para los últimos chipsets, problema que se suele corregir acudiendo al soporte de framebuffer del kernel o esperando a que quienes trabajan en XFree86 obtengan por parte de los fabricantes (poco colaboradores) la información sobre cómo programar sus tarjetas.

En ciertos casos también se nos puede preguntar si deseamos que el sistema arranque directamente en XWindow o incluso testear los diferentes modos de vídeo para ver que si son correctamente soportados.

Además para que estos programas realicen una correcta configuración el sistema debe estar ya correctamente configurado en cuanto a modo texto se refiere. Por ejemplo, Xconfigurator no activará el teclado en castellano para XWindow si no hemos especificado que el idioma en que queremos usar nuestro Linux es este mismo, mediante la orden

export LANG=es_ES

Bien en la línea de comandos, o en el fichero /etc/profile. Cuando Xconfigurator detecta esta variable de entorno ya sabe qué lenguaje utiliza el usuario y por tanto configurará XWindow para el correcto uso del teclado en este lenguaje. Otras variables indican el tipo de ratón, etc. y suelen ser correctamente especificadas por el programa de instalación. En el caso de XF86Setup la configuración permite una mayor participación por parte del usuario, ya que en lugar de realizar una configuración automática proporciona una lista de opciones para cada apartado, de modo que si conocemos nuestro hardware es la mejor manera de



configurar el sistema. Además, a la hora de elegir resoluciones de vídeo, este programa permite especificar cuál es la resolución por defecto deseada mientras que Xconfigurator deja marcada la menor de ellas como "por defecto" (o la primera que seleccionemos).

El programa de configuración xf86config es el más antiguo de todos, y realiza la configuración mediante preguntas que se responderán con teclado. Nos preguntará la tarjeta de vídeo, monitor, rango de frecuencias del monitor, videomemoria y resoluciones de trabajo deseadas, así como el mapa del teclado a utilizar (es 102 = España 102 teclas).

Como siempre, el método más eficaz de configurar XWindow es directamente editando los ficheros de configuración. Y es que otro método de configuración básico es sencillamente editar el fichero/etc/X11/XF86Config, que es en realidad lo que hacen todos los programas comentados anteriormente. En este fichero la configuración está dividida en secciones y mediante cualquier editor de texto llano es posible cambiar cualquier parámetro de XWindow.

Este fichero se divide en varias categorías o secciones (todas ellas con más subopciones muy bien comentadas en dicho fichero de configuración):

- Sección Files: Indica los caminos hacia las fuentes o ficheros de colores.
- RgbPath "/usr/X11R6/lib/X11/rgb"
- FontPath "/usr/X11R6/lib/X11/fonts/75dpi:unscaled"
- (etc.)
- Sección ServerFlags: Opciones varias sobre el servidor.
- Sección KeyBoard: Configuración del teclado. Para teclados en castellano las opciones son similares a las siguientes:
- Protocol "Standard"
- LeftAlt Meta
- RightAlt Meta
- ScrollLock Compose
- RightCtl Control
- XkbKeycodes "xfree86"
- XkbTypes "default"
- XkbCompat "default"
- XkbSymbols "us(pc101)"
- XkbGeometry "pc"
- XkbRules "xfree86"
- XkbModel "pc101"
- XkbLayout "es"
- Sección Pointer: Configuración del ratón:
- # Para un ratón microsoft mouse o un genius:
- Protocol "Microsoft"
- Device "/dev/mouse"
- Sección Monitor: Datos sobre el monitor (frecuencias).
- Sección Device: Indica el servidor X a utilizar: Por ejemplo para una tarjeta con chipset Mach64:

Instituto de nuevas Tecnologías

- Section "Device"
- Identifier "Mach64 GB"
- VendorName "ATI"
- BoardName "ATI 3d Rage II"
- VideoRam 8192
- EndSection
- Sección Screen: Modos de vídeo seleccionados: Por ejemplo para activar 320x200, 640x480, 800x600 y 1024x768 tanto a 8 como a 16 bpp de color, siendo usado por defecto el modo 1024x768x16 (el primero que aparezca en la lista):
- Section "Screen"
- Driver "accel"
- Device "Mach64 GB"
- Monitor "My Monitor"
- DefaultColorDepth 16
- Subsection "Display"
- Depth 16
- Modes "1024x768" "800x600" "640x480" "320x200"
- ViewPort 00
- Virtual 1024 768
- EndSubsection
- Subsection "Display"

8

- Depth
- Modes "1024x768" "800x600" "640x480" "320x200"
- ViewPort 00
- Virtual 1024 768
- EndSubsection
- EndSection

Una vez creado este fichero, entramos en XWindow mediante el comando startx.

Del entorno XWindow se puede salir en cualquier momento mediante la combinación de teclas Ctrl+Alt+Borrar, o bien seleccionando LOGOUT en el menú del gestor de ventanas que utilicemos.

Para volver a las consolas de texto bastará con utilizar la combinación de teclas Ctrl+Alt+F1, +F2, etc (hasta F6 en las instalaciones por defecto). Para volver al terminal gráfico se utiliza de F7 en adelante (Ctrl+Alt+F7, +F8, etc.).

Si durante la instalación (o en la línea apropiada del fichero XF86Config) seleccionamos varias resoluciones de vídeo, es posible cambiar entre ellas mediante las teclas Ctrl+Alt+'+' y Ctrl+Alt+'-' (los signos del teclado numérico).

Además, si por defecto marcamos 8bpp (o 16 bpp) en las opciones, nada nos impide lanzar posteriormente XWindow en la profundidad de color deseada:

[sromero@localhost]# startx -- -bpp 8 [sromero@localhost]# startx -- -bpp 16



etc.

También es posible establecer la profundidad de color por defecto mediante una línea DefaultColorDepth = 16 en el fichero de configuración XF86Config, sección Screen.

Por último, saber que podemos lanzar más de una sesión de XWindow simultáneamente (en las consolas accesibles desde Ctrl+Alt+F8, +F9, etc.), mediante:

```
[sromero@localhost]# startx -- :1
[sromero@localhost]# startx -- :2
etc.
```

¿Cómo cambio la resolución por defecto?

En el fichero /etc/X11/XF86Config en la sección screen correspondiente al servidor que se está utilizando actualmente, en la subsección Display correspondiente a la Depth en que estemos trabajando, buscamos la línea:

Modes "resolución_1" "resolución_2"...

Un ejemplo:

Modes "1024x768" "800x600" "640x480" "320x200"

Entonces ponemos la resolución que queramos que sea por defecto de primera. Estas resoluciones puestas aquí son entre las que podremos cambiar si queremos al pulsar Ctrl+ Alt+ '+' ó Ctrl+Alt+'-', una vez arrancado X-windows..

¿Cómo cambio el escritorio virtual?

En el fichero /etc/X11/XF86Config, en la sección screen correspondiente al servidor que se está utilizando actualmente, en la subsection Display correspondiente a la Depth en que estemos trabajando, añadimos la línea:

Virtual tamaño_x tamaño_y #en píxeles

Un ejemplo:

Virtual 1024 768

Hay que puntualizar que si existe una resolución definida de Modes mayor que el escritorio virtual, esta no se podrá utilizar. Resumiendo, no se puede tener una resolución de trabajo mayor que el escritorio virtual.

¿Cómo cambio el número de colores por defecto?

Hay dos maneras:

Mediante la línea de comandos:

[user@localhost]# startx --bpp resolución (dónde resolución es 8, 16,24 ó 32) Mediante el fichero de configuración:



En el fichero /etc/X11/XF86Config, en la sección screen correspondiente al servidor que se está utilizando actualmente, añadimos la línea:

DefaultColorDepth depth_que_queramos_utilizar Un ejemplo:

DefaultColorDepth 16

Un Depth con valor 8 significa que trabajaremos a 256 colores, uno con valor 32 truecolor. No todas las tarjetas gráficas podrán utilizar todos los Depth disponibles en todas las resoluciones, todo dependerá de la tarjeta gráfica y de la memoria que esta tenga.

¿Cómo arranco directamente en X-windows?

En Linux es perfectamente posible pedir que el arranque del sistema se haga en modo gráfico, y que el login y password se introduzcan directamente en una ventana XWindow para la posterior carga del gestor de ventanas habitual que use dicho usuario.

Es decir, podremos identificarnos y aparecer directamente bajo X sin necesidad de ejecutar startx.

Para arrancar directamente en X Window (o no hacerlo) todo el proceso de configuración gira en torno a cambiar el runlevel (o nivel de ejecución en que arranca Linux).

El runlevel es, dicho de una manera sencilla, el modo en que arranca Linux. Por defecto el runlevel suele ser el 2 ó el 3, es decir, arranque en modo texto o consola. Para cada distribución suele haber una lista de runleves y sus significados, aunque casi se puede decir que son similares para todas ellas. Para Redhat, por ejemplo, la lista es la siguiente:

- # Porción del fichero /etc/inittab
- # Default runlevel. The runlevels used by RHS are:
- # 0 halt (Do NOT set initdefault to this)
- # 1 Single user mode
- # 2 Multiuser, without NFS (The same as 3)
- # 3 Full multiuser mode
- # 4 unused
- # 5 X11
- # 6 reboot (Do NOT set initdefault to this)

Como puede verse, el arranque normal por defecto para que puedan acceder usuarios a Linux es 2 (si no tenemos red) o 3 si queremos usar servicios de red, siendo normalmente este el runlevel por defecto. Como puede verse, X11 tiene asignado el runlevel al 5. Si cambiamos el runlevel por defecto para que arranque en el 5, nos aparecerán directamente X Window.

Para cambiar este runlevel por defecto hay que editar el fichero /etc/inittab, y acudir a la siguiente línea:

id:3:initdefault:



El valor numérico antes de initdefault indica el runlevel de arranque por defecto. Si lo cambiamos de 3 a 5, la próxima vez que arranquemos Linux arrancará en X Window: id:5:initdefault:

Para que el proceso se ejecute correctamente debe tener instalado un gestor de arranque en runlevel 5, que es el programa encargado de pedir el login+passwd y arrancar el gestor de ventanas adecuado. Puede elegir entre xdm (el que viene por defecto con XWindow), kdm (version de kde del mismo) y gdm (versión aportada por gnome). Cada una de ellas dispone de ciertos ficheros de configuración o ejecutables para cambiar el fondo del login, el lenguaje, gestor de ventanas, etc. Consulte los ficheros en los paquetes relacionados.

Haga "man runlevel" para más información sobre runlevels.

En las últimas versiones de Redhat, el programa Xconfigurator le permite elegir si desea o no arrancar directamente en XWindow (él mismo modifica el runlevel por defecto) mediante una simple pregunta a la que usted debe responder SI (Si desea arrancar en runlevel 5) o no (si desea arrancar en runlevel 3). Esta pregunta se le suele realizar al finalizar la selección de resoluciones y antes de salir de Xconfigurator.

Una vez arranque en XWindow, puede volver a cualquier consola de texto mediante las teclas Ctrl+Alt+F1, Ctrl+Alt+F2, etc. (por defecto entre F1 y F6 tendrás 6 consolas de texto), y volver a XWindow en cualquier momento mediante Ctrl+Alt+F7 en adelante.

En algunas distribuciones puede ser necesario indicarle a inittab dónde está el gestor de runlevel 5 que debe arrancar:

Para xdm:

#Run xdm in runlevel 5 x:5:respawn:/usr/X11R6/bin/wdm -nodaemon Para gdm: x:5:respawn:/etc/X11/prefdm -nodaemon

Estas líneas suelen ser automáticamente incluidas en el fichero por los rpm/deb instaladores de xdm, kdm y gdm. Consulte en la ayuda de dichos programas para más información.

¿Qué es un gestor de ventanas?

Un gestor de ventanas no es otra cosa que el conjunto de programas, ventanas, funcionalidades,.... Que hacen posible que el usuario pueda interactuar con el sistema de forma gráfica y no en modo texto.

Para usar un gestor de ventanas, hay que tener configurado un *servidor X*. También hay que decir que el *gestor de ventanas* utilizado es totalmente independiente del *servidor X* utilizado.



Al contrario que en otros sistemas operativos, en Linux no es necesario utilizar un *servidor X* - *gestor de ventanas* para usar el sistema. El sistema operativo y el conjunto *servidor X* - *gestor de ventanas* usado, son cosas totalmente diferentes, independientes entre sí. Es más, existen usuarios que trabajan en modo texto sin ningún problema y sin usar un interfaz gráfico.

Existen numerosos y variados gestores de ventanas para Linux, unos mejores y otros más desarrollados y estables. Es el usuario el que tiene que decidir que gestor satisface mejor sus necesidades, pudiendo incluso tener más de uno instalado. Para aclarar un poco las cosas, podríamos decir que, si un ordenador es usado por varios usuarios, todos utilizaran el mismo *servidor X* pero no necesariamente el mismo *gestor de ventanas*.

¿Cómo elijo el Gestor de ventanas que quiero arrancar?

Una vez configurado X se hace necesario decirle a XWindow qué gestor de ventanas debe arrancar. Dicho de una manera sencilla, XWindow es la conjunción de 3 componentes:

- El entorno de ventanas X Window: Es quien implementa las llamadas internas de XWindow, el protocolo X, etc., es decir, el sistema de comunicaciones entre aplicaciones.
- El servidor gráfico X. Es un programa (un fichero ejecutable) encargado de responder a las órdenes gráficas encargadas por el entorno de ventanas. Es el encargado de dibujar rectángulos, puntos de color o pixels, rellenar zonas, dibujar imágenes, etc. La manera de decirle a XWindow qué ejecutable es el que deseamos usar es crear un enlace simbólico llamado /usr/X11R6/bin/X que apunte al servidor que usemos:
- Ej: In -s /usr/X11R6/bin/XF86-S3 /usr/X11R6/bin/X
- (le diría que utilice el servidor de las S3).

Este enlace es creado por el programa de configuración cuando seleccionamos la tarjeta en la lista que nos presenta.



• El gestor de ventanas es el módulo encargado de decidir qué aspecto, forma, funcionalidades, atajos de teclado, opciones y menús debe tener XWindow.

Así, al dividir XWindow en estos 3 módulos se nos permite cambiar cualquiera de ellos para dotar a XWindow del aspecto o funcionalidades que deseemos. El entorno de ventanas no suele cambiarse nunca (es XFree86), y el servidor X tampoco, ya que éste último es siempre el ejecutable apropiado para manejar nuestra tarjeta gráfica (XFree86-SVGA, XFree86-S3, etc.), pero el último es totalmente seleccionable entre multitud de opciones y es lo que nos permite cambiar radicalmente el aspecto de XWindow. Cambiando el gestor de ventanas (o Window Manager) podemos cambiar prácticamente el funcionamiento completo de XWindow.

Gestores de ventanas hay muchos: kde, afterstep, gnome, WindowMaker, icewm, etc, y todos tienen sus respectivos ejecutables: startkde, afterstep, gnome-session, wmaker, icewm, etc. Toda esta explicación se ha dado por un motivo muy sencillo: la posibilidad de elegir el Gestor de Ventanas deseado es parte de la configuración activa de XWindow, y vamos a ver la manera de decirle a XWindow qué gestor deseamos utilizar.

Para ello, simplemente debemos ir al directorio HOME del usuario en cuestión (/home/usuario) o de root (/root) y editar (creándolo si no existe) el fichero de texto .xinitrc . En este fichero pondremos una línea que indicará el Gestor de Ventanas que deseamos utilizar, en el formato:

exec ejecutable_del_gestor_de_ventanas Además podemos utilizar líneas de comentarios que comiencen por el carácter '#' y que XWindow ignorará:

Ejemplos de ficheros .xinitrc:

utilizar afterstep
exec afterstep
otro ejemplo podría ser lanzar kde:
utilizar KDE
exec startkde
Así, dependiendo del ejecutable que lancemos (startkde, gnome-session, icewm, wmaker, ctwm, fvwm2, wmaker, blackbox, etc.) podremos arrancar el WM deseado.



¿Dónde consigo un nuevo gestor de ventanas?

Muchas distribuciones incluyen en sus CDes una serie de gestores de ventanas. También lo puedes bajar de sus respectivos servidores web:

- AfterStep: <u>http://www.afterstep.org</u>
- Gnome: <u>http://www.gnome.org/</u>
- KDE: <u>http://www.kde.org</u>
- FVWM: <u>http://www.hpc.uh.edu/fvwm/</u>
- Enlightenment: http://www.rasterman.com/
- WindowMaker: <u>http://www.windowmaker.org/</u>
- mlvwm: http://www.bioele.nuee.nagoya-u.ac.jp/member/tak/mlvwm.html
- TkDesk: http://people.mainz.netsurf.de/~bolik/tkdesk/
- eXode: http://www.simplicity.net/exode/
- wm2: http://www.all-day-breakfast.com/wm2/
- icewm: http://www.kiss.uni-lj.si/~k4fr0235/icewm/