

There are no translations available.

Descubre en este artículo la estructura del sistema de archivos en Linux..

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 inicios 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 creo 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.
- 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 contenido 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).

- o Archivos estáticos: /etc/password, /etc/shadow.
- o 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 como funciona Linux.

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

NOTA	: podemos acceder a los d
-------------	---------------------------

Hay que saber que si editamos, por ejemplo, un fichero vinculado a un elemento Hardware, seguramen
--

En definitiva, no es aconsejable leer o abrir y mucho menos modificar archivos vinculados a elementos

4.- Organización de 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

La estructura del sistema de archivos en Linux

Sagrario Peralta-k idatzia

Ostirala, 2007(e)ko abuztua(r)en 31-(e)an 13:10etan

para cumplir estas especificaciones.

Además, es deseable que se mantenga lo 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.

Vemos, por ejemplo, un listado de su estructura:



The screenshot shows a terminal window titled "alex@alexDebian: /". The window has a menu bar with "Archivo", "Editar", "Ver", "Terminal", "Solapas", and "Ayuda". The terminal displays the command "alex@alexDebian:/\$ ls -F" and its output, which lists the root directory structure in a grid-like format. The output is as follows:

```
alex@alexDebian:/$ ls -F
bin/      downloads/  jorge/     mnt/       srv/        vmlinuz.old@
boot/     etc/        lib/        opt/        sys/
cdrom@    home/       lib64/      proc/       tmp/
ctr2/     initrd/     lost+found/ publico/     usr/
Desastre/ initrd.img@ marfil/     root/       var/
dev/      initrd.img.old@ media/      sbin/       vmlinuz@
alex@alexDebian:/$
```

Contenido del directorio raíz

o **/bin**

En este directorio se ubica el código binario o compilado de los programas y comandos que

La estructura del sistema de archivos en Linux

Sagrario Peralta-k idatzia

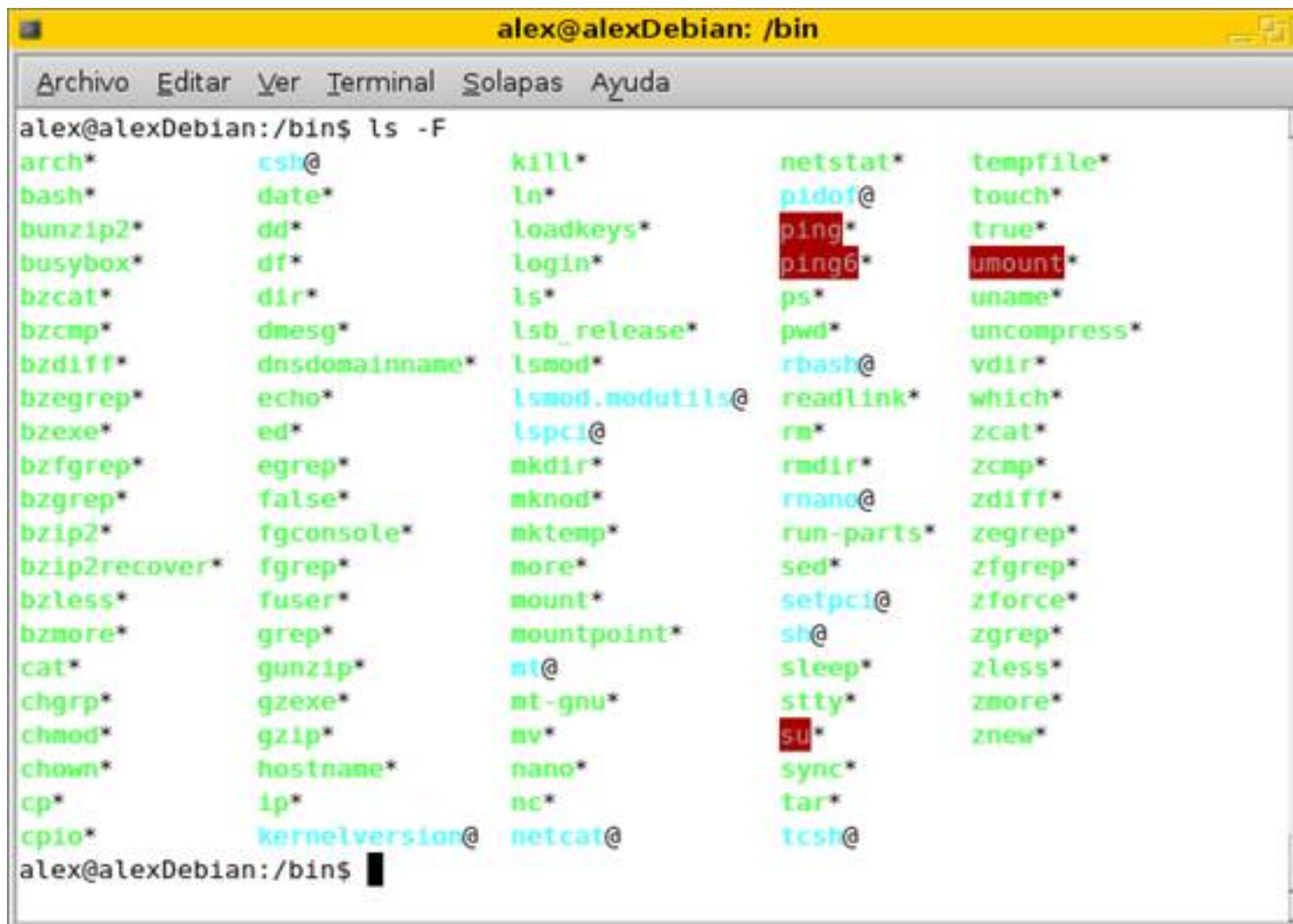
Ostirala, 2007(e)ko abuztua(r)en 31-(e)an 13:10etan

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.



The screenshot shows a terminal window titled "alex@alexDebian: /bin". The window has a menu bar with "Archivo", "Editar", "Ver", "Terminal", "Solapas", and "Ayuda". The command "alex@alexDebian:/bin\$ ls -F" has been executed, displaying a list of files and directories in the /bin directory. The output is as follows:

```
alex@alexDebian:/bin$ ls -F
arch*      csh@      kill*     netstat*  tempfile*
bash*      date*     ln*       pidof@    touch*
bunzip2*   dd*       loadkeys* ping*      true*
busybox*   df*       login*    ping6*    umount*
bzipcat*   dir*      ls*       ps*       uname*
bzipcmp*   dmesg*    lsb_release* pwd*       uncompress*
bzipdiff*  dnsdomainname* lsmod*    rbash@    vdir*
bzipgrep*  echo*     lsmod.modutils@ readlink*  which*
bzipexe*   ed*       lspci@    rm*       zcat*
bzipfgrep* egrep*    mkdir*    rmdir*    zcmp*
bzgrep*    false*    mknod*    rnano@    zdiff*
bzip2*     fgconsole* mktemp*   run-parts* zegrep*
bzip2recover* fgrep*    more*     sed*      zfgrep*
bzless*    fuser*    mount*    setpci@   zforce*
bzmore*    grep*     mountpoint* sh@        zgrep*
cat*       gunzip*   nt@       sleep*    zless*
chgrp*     gzexe*    nt-gnu*   stty*     zmore*
chmod*     gzip*     nv*       su*        znew*
chown*     hostname* nano*      sync*
cp*        ip*       nc*       tar*
cpio*      kernelversion@ netcat@    tcsh@
alex@alexDebian:/bin$
```

La estructura del sistema de archivos en Linux

Sagrario Peralta-k idatzia

Ostirala, 2007(e)ko abuztua(r)en 31-(e)an 13:10etan

* La @ al lado del nombre de un fichero representa un enlace simbólico

o **/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 usuario.

Realmente se crean dos entornos totalmente separados, es decir, cada uno tiene su propia zona de memoria.

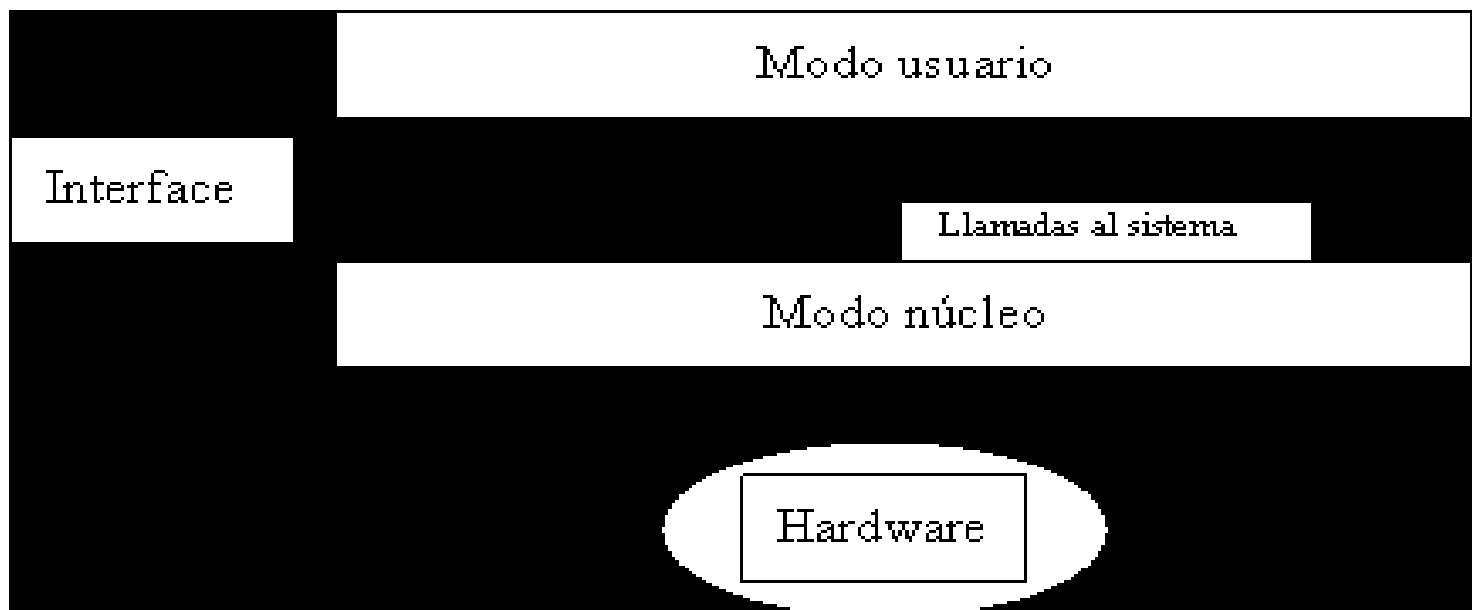
Démonos cuenta que esta técnica ofrece mucha seguridad y estabilidad al sistema.

Cuando un proceso del "modo usuario" necesita recursos del "modo kernel" (por ejemplo, acceder a la memoria física), se debe pasar al modo kernel.

La estructura del sistema de archivos en Linux

Sagrario Peralta-k idatzia

Ostirala, 2007(e)ko abuztua(r)en 31-(e)an 13:10etan



```
alex@alexDebian: /boot

Archivo  Editar  Ver  Terminal  Solapas  Ayuda

alex@alexDebian:/boot$ ls -F
config-2.6.16-2-686      initrd.img-2.6.17-2-686  vmlinuz-2.6.16-2-686
config-2.6.17-2-686     lost+found/              vmlinuz-2.6.17-2-686
grub/                  System.map-2.6.16-2-686
initrd.img-2.6.16-2-686 System.map-2.6.17-2-686
alex@alexDebian:/boot$
```


La estructura del sistema de archivos en Linux

Sagrario Peralta-k idatzia

Ostirala, 2007(e)ko abuztua(r)en 31-(e)an 13:10etan

```
alex@alexDebian: /etc
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
alex@alexDebian:/etc$ ls -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     rc1.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
```

Veremos parte de su contenido. Se explica que son y cómo funcionan las librerías y los archivos cabecera.

La estructura del sistema de archivos en Linux

Sagrario Peralta-k idatzia

Ostirala, 2007(e)ko abuztua(r)en 31-(e)an 13:10etan

```
alex@alexDebian: /lib
Archivo  Editor  Ver  Terminal  Solapas  Ayuda
alex@alexDebian:/lib$ ls -F
alsa/                                libnsl-2.3.6.so
cpp@                                  libnsl.so.1@
discover/                           libnss_compat-2.3.6.so
firmware/                           libnss_compat.so.2@
i386-linux-gnu@                     libnss_dns-2.3.6.so
i486-linux-gnu/                     libnss_dns.so.2@
init/                               libnss_files-2.3.6.so
iptables/                           libnss_files.so.2@
klibc-AJglVoLaaW90SYIH0Qt104N9G4e.so* libnss_hesiod-2.3.6.so
ld-2.3.6.so*                         libnss_hesiod.so.2@
ld-linux.so.2@                       libnss_nis-2.3.6.so
libacl.so.1@                         libnss_nisplus-2.3.6.so
libacl.so.1.1.0                     libnss_nisplus.so.2@
libanl-2.3.6.so                     libnss_nis.so.2@
libanl.so.1@                         libpamc.so.0@
libatm.so.1@                         libpamc.so.0.79
libatm.so.1.0.0                     libpam_misc.so.0@
libattr.so.1@                       libpam_misc.so.0.79
libattr.so.1.1.0                     libpam.so.0@
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
libBrokenLocale.so.1@               libpcprofile.so
```

El directorio /lib contiene los archivos de biblioteca compartida que se utilizan para ejecutar los programas. Este directorio también contiene algunos archivos de configuración y de datos. Los archivos de biblioteca compartida se encuentran en el directorio /lib, los archivos de configuración en /etc y los archivos de datos en /var.

```
alex@alexDebian: /sbin
Archivo  Editor  Ver  Terminal  Solapas  Ayuda
alex@alexDebian:/sbin$ ls -F
acpi_available*      iptables-restore*   pivot_root*
apm_available*       iptables-save*      plipconfig*
badblocks*           iptunnel*           pmap_dump*
blkid*               isosize*            pmap_set*
blockdev*            iwconfig*           portmap*
bootlogd*            iwevent*            poweroff@
cfdisk*              iwgetid*            rarp*
debugfs*             iwlist*             raw*
depmod*              iwpriv*             reboot@
depmod.modutils*     iwspy*              resize2fs*
dhclient*            kallsyms@           rmod*
discover*            kallsyms.modutils@ rmod.Lmodutils@
dosfsck*             kbdrate*            rmod.modutils@
dumpe2fs*            kernelversion*      route*
e2fsck*              killall5*           rpc.lockd*
e2image*             klogd*              rpc.statd*
e2label*            ksyms@              rtacct*
fdisk*               ksyms.modutils@     rtmmon*
findfs*              ldconfig*            runlevel*
fsck*                logsave*             scsi_id@
fsck.cramfs*         losetup*             sfdisk*
fsck.ext2*           lsmod@              shadowconfig*
fsck.ext3*           lsmod.Lmodutils@    showmount*
```

En este directorio se guardan los archivos temporales de los servicios instalados en el sistema.

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.

La estructura del sistema de archivos en Linux

Sagrario Peralta-k idatzia

Ostirala, 2007(e)ko abuztua(r)en 31-(e)an 13:10etan

Comúnmente, se almacena aquí el software instalado en el sistema.

Vemos su contenido:



```
alex@alexDebian: /usr
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
alex@alexDebian:/usr$ ls -F
applnk/  games/  lib/    libexec/  lost+found/  share/  X11R6/
bin/     include/  lib64/  local/    sbin/        src/
alex@alexDebian:/usr$
```

Estructura de /usr

- o **/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.

- o **/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.

La estructura del sistema de archivos en Linux

Sagrario Peralta-k idatzia

Ostirala, 2007(e)ko abuztua(r)en 31-(e)an 13:10etan

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
```

```
main()
```

{	
unsigned	int
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 implemento el código fu

La estructura del sistema de archivos en Linux

Sagrario Peralta-k idatzia

Ostirala, 2007(e)ko abuztua(r)en 31-(e)an 13:10etan

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

Vemos parte de su contenido:

```
alex@alexDebian: /usr/include
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
alex@alexDebian:/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/        scsi/         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      nls_types.h   shadow.h      unistd.h
asm-i486/      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        locale.h       paths.h        stdint.h      utmpx.h
bits/          fts.h          malloc.h       printf.h       stdio_ext.h   values.h
byteswap.h     ftw.h          math.h         pthread.h      stdio.h       wait.h
c++/           _G_config.h   mcheck.h      pty.h          stdlib.h      wchar.h
complex.h      gconv.h        memory.h       pwd.h          strings.h     wctype.h
cpio.h         getopt.h       mntent.h      python2.3/     stropts.h     wordexp.h
crypt.h        gettext-po.h   monetary.h    python2.4/     sys/          xlocale.h
ctype.h        glob.h         mqueue.h      syscall.h
```

o **/usr/lib**

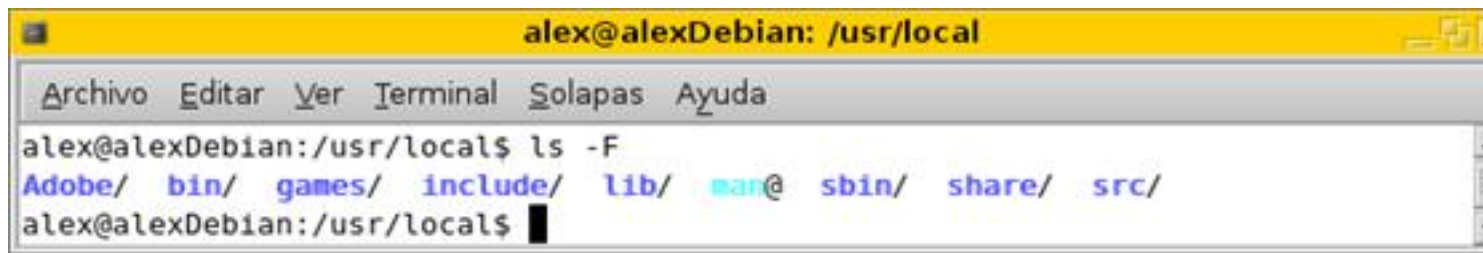
Este directorio incluye librerías compartidas y ficheros binarios pensados para no ser ejecutados directamente por los usuarios del sistema.

o **/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:



The screenshot shows a terminal window titled "alex@alexDebian: /usr/local". The window has a menu bar with "Archivo", "Editar", "Ver", "Terminal", "Solapas", and "Ayuda". The terminal text shows the command "alex@alexDebian:/usr/local\$ ls -F" and its output: "Adobe/ bin/ games/ include/ lib/ man@ sbin/ share/ src/". The prompt "alex@alexDebian:/usr/local\$" is followed by a cursor.

o **/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.

Vemos parte de su contenido:

```
alex@alexDebian: /usr/sbin
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
alex@alexDebian:/usr/sbin$ ls -F
a2dismod*          lpinfo*
a2dissite*         lpmove*
a2enmod*           lsusb*
a2ensite*          mailer*
ab*               make-ssl-cert*
ab2@              mkboot*
accept*           mkinitramfs*
accessdb*         mkinitramfs-kpkg*
acpid*            mklost+found*
addgnupghome*     mksmbpasswd*
addgroup@         modhandler.py*
add-shell*        mount_unidades_red_csic*
adduser*          newusers*
alsaconf*         nfsstat*
alsactl*          nmbd*
apache2*          no_icmp*
apache2ctl*       nologin*
apache2-ssl-certificate*
arp*              ownership*
arpd*             pac*
aspell-autobuildhash*
atd*              pam_getenv*
biosdecode*       pam_tally*
                  pppd*
                  pppdump*
```

Repositorio de código fuente de la estructura de archivos de 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 a crecer constantemente.

La utilidad de /var radica en poder detectar problemas para prevenirlos y solucionarlos.

La estructura del sistema de archivos en Linux

Sagrario Peralta-k idatzia

Ostirala, 2007(e)ko abuztua(r)en 31-(e)an 13:10etan

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.

Vemos su contenido:

A screenshot of a terminal window titled 'alex@alexDebian: /var'. The window has a menu bar with 'Archivo', 'Editar', 'Ver', 'Terminal', 'Solapas', and 'Ayuda'. The terminal shows the command 'alex@alexDebian:/var\$ ls -F' and its output: 'backups/ games/ local/ log/ opt/ run/ tmp/' on the first line and 'cache/ lib/ lock/ mail/ packages/ spool/ www/' on the second line. The prompt 'alex@alexDebian:/var\$' is visible at the bottom.

```
alex@alexDebian:/var$ ls -F
backups/  games/  local/  log/    opt/    run/    tmp/
cache/    lib/    lock/   mail/   packages/ spool/  www/
alex@alexDebian:/var$
```

Distribución de algunos subdirectorios de /var

o /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.

- o **/var/lib**

Encontramos aquí información sobre el estado variable de las aplicaciones.

- o **/var/lock**

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

- o **/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:

```
alex@alexDebian: /var/log
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
alex@alexDebian:/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
btmpt          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
dirmngr.log    fontconfig.log lpr.log        sshd.log
alex@alexDebian:/var/log$

alex@alexDebian: /var/run
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
alex@alexDebian:/var/run$ ls -F
acpid.socket=  crond.reboot   gdm.pid        klogd.pid     sshd/
alsa/          cups/          hal/           lpd.pid       sshd.pid
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@alexDebian:/var/run$

alex@alexDebian: /var/spool
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
alex@alexDebian:/var/spool$ ls -F
cron/  cups/  exim4/  lpd/  mail@  openoffice/  samba/
alex@alexDebian:/var/spool$
```

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 en 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:

```
alex@alexDebian: /proc
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
alex@alexDebian:/proc$ 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/   4045/  4344/  4478/  4567/  9519/   iomem   self@
1579/   4053/  4353/  4482/  4626/  acpi/   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
3429/   4245/  4447/  4518/  5798/  driver/ modules  zoneinfo
3431/   4252/  4450/  4524/  5799/  execdomains mounts@
alex@alexDebian:/proc$
```

* Los subdirectorios identificados por un numero, corresponden a los PID de los procesos.

5.- Referencias

- o <http://www.wikipedia.es/>
- o <http://www.redhat.com/>
- o <http://www.pathname.com/fhs/>
- o http://www.microtecnologias.cl/linux_fhs.html
- o <http://www.FreeUOC.org/> / Proyecto LPI

La estructura del sistema de archivos en Linux

Sagrario Peralta-k idatzia

Ostirala, 2007(e)ko abuztua(r)en 31-(e)an 13:10etan

- o <http://es.tldp.org/>
- o <http://www.hispafuentes.com/>
- o <http://www-128.ibm.com/developerworks/linux/lpi/index.html>
- o Administración básica de GNU/Linux. Autor: José Ángel de Bustos Pérez
- o G.L.U.P. Guía de Linux Para El Usuario. Larry Greenfield
- o GNU/Linux: Instalación y primeros pasos. Autores: Matt Welsh, Phil Hughes. Julio de 2002
- o Sistema Operativo GNU/Linux básico. UOC. Autores: Roger Borg Viñas. Francesc Aulí Llinas
- o Apuntes □ Sistema Operativo Linux□ Universidad Politécnica de Madrid. Noviembre 2004
- o Guía de referencia Debian. Autor: Osamu Auki. Febrero 2005
- o Linux. 6ª Edición. Autores: David Bandel y Robert Napier. Editorial: Prentice Hall