

Estándar de Jerarquía del Sistema de Archivos

Grupo de trabajo LSB, La Fundación Linux

Estándar de Jerarquía del Sistema de Archivos

Grupo de trabajo LSB, The Linux Foundation

versión 3.0

Fecha de publicación 19 de marzo de 2015

Copyright © 2015 The Linux Foundation

Copyright © 1994-2004 Daniel Quinlan

Copyright © 2001-2004 Paul 'Rusty' Russell

Copyright © 2003-2004 Christopher Yeoh

Abstracto

Este estándar consta de un conjunto de requisitos y pautas para la ubicación de archivos y directorios en sistemas operativos similares a UNIX. Las pautas están destinadas a respaldar la interoperabilidad de aplicaciones, herramientas de administración de sistemas, herramientas de desarrollo y scripts, así como conseguir una mayor uniformidad de documentación para estos sistemas.

Todas las marcas comerciales y derechos de autor pertenecen a sus propietarios, a menos que se indique específicamente lo contrario. No se debe considerar que el uso de un término en este documento afecte la validez de ninguna marca comercial o de servicio.

Se otorga permiso para hacer y distribuir copias textuales de este estándar siempre que los derechos de autor y este aviso de permiso se conserven en todas las copias.

Se concede permiso para copiar y distribuir versiones modificadas de este estándar bajo las condiciones para la copia literal, siempre que la página de título esté etiquetada como modificada incluyendo una referencia al estándar original, siempre que se incluya y proporcione información sobre cómo recuperar el estándar original. que todo el trabajo derivado resultante se distribuye bajo los términos de un aviso de permiso idéntico a éste.

Se concede permiso para copiar y distribuir traducciones de este estándar a otro idioma, bajo las condiciones anteriores para versiones modificadas, excepto que este aviso de permiso puede establecerse en una traducción aprobada por el titular de los derechos de autor.

Dedicatoria

Este comunicado está dedicado a la memoria de Christopher Yeoh, un viejo amigo y colega. Uno de los editores originales de la FHS. Sin su dedicación este trabajo no hubiera sido posible.

Tabla de Contenido

1.	Introducción.....	1
1.1.	Propósito.....	1
1.2.	Convenciones.....	1
2.	El Sistema de Archivos.....	2
3.	El Sistema de Archivos Raíz.....	3
3.1.	Propósito.....	3
3.2.	Requisitos.....	4
3.3.	Opciones Específicas.....	4
3.4.	/bin : Binarios de comandos de usuario esenciales (para todos los usuarios).....	5
3.4.1.	Propósito.....	5
3.4.2.	Requisitos.....	5
3.4.3.	Opciones Específicas.....	6
3.5.	/boot : Archivos estáticos del cargador de arranque.....	7
3.5.1.	Propósito.....	7
3.5.2.	Opciones Específicas.....	7
3.6.	/dev : Archivos de Dispositivo.....	7
3.6.1.	Propósito.....	7
3.6.2.	Opciones Específicas.....	7
3.7.	/etc : Configuración del Sistema Especifico del host.....	7
3.7.1.	Propósito.....	7
3.7.2.	Requisitos.....	7
3.7.3.	Opciones Específicas.....	8
3.7.4.	/etc/opt : Archivos de Configuración para /opt.....	9
3.7.5.	/etc/X11 : Configuración para el sistema X Window (opcional)	9
3.7.6.	/etc/sgml : Archivos de Configuración para SGML (opcional)	10
3.7.7.	/etc/xml : Archivos de Configuración para XML (opcional)	10
3.8.	/home : Directorios de inicio de Usuario (opcional)	10
3.8.1.	Propósito.....	10
3.8.2.	Requisitos.....	10
3.8.3.	Especificaciones y convenciones del directorio de inicio.....	11
3.9.	/lib : Módulos de kernel y bibliotecas compartidas esenciales.....	11
3.9.1.	Propósito.....	11
3.9.2.	Requisitos.....	11
3.9.3.	Opciones Específicas.....	11
3.10.	/lib <qual> : Bibliotecas compartidas esenciales de formato alternativo (optional).....	11
3.10.1.	Propósito.....	11
3.10.2.	Requisitos.....	12
3.11.	/media : Punto de Montaje para Medios Extraíbles.....	12
3.11.1.	Propósito.....	12
3.11.2.	Opciones Específicas.....	12
3.12.	/mnt : Punto de Montaje temporal para un sistema de archivos.....	12
3.12.1.	Propósito.....	12
3.13.	/opt : Paquetes de Software de aplicación complementaria.....	13
3.13.1.	Propósito.....	13
3.13.2.	Requisitos.....	13
3.14.	/root : Directorio de Inicio para el usuario Raíz (opcional)	14
3.14.1.	Propósito.....	14
3.15.	/run : Datos variables en Tiempo de Ejecución.....	14
3.15.1.	Propósito.....	14
3.15.2.	Requisitos.....	14
3.16.	/sbin: Binarios del Sistema.....	15

3.16.1.	Propósito.....	15
3.16.2.	Requisitos.....	15
3.16.3.	Opciones específicas.....	15
3.17.	/srv: Datos de los servicios proporcionados por este sistema.....	16
3.17.1.	Propósito.....	16
3.18.	/tmp: Archivos temporales.....	17
3.18.1.	Propósito.....	17
4.	La Jerarquía /usr.....	18
4.1.	Propósito.....	18
4.2.	Requisitos.....	18
4.3.	Opciones específicas.....	18
4.4.	/usr/bin: La mayoría de los comandos de usuario.....	18
4.4.1.	Propósito.....	18
4.4.2.	Requisitos.....	19
4.4.3.	Opciones específicas.....	19
4.5.	/usr/include: Directorio para archivos de inclusión estándar.....	19
4.5.1.	Propósito.....	19
4.5.2.	Opciones específicas.....	19
4.6.	/usr/lib: Bibliotecas para programación y paquetes.....	20
4.6.1.	Propósito.....	20
4.6.2.	Opciones específicas.....	20
4.7.	/usr/libexec: Binarios ejecutados por otros programas (opcional)	20
4.7.1.	Propósito.....	20
4.8.	/usr/lib<qual>: Bibliotecas de formato alternativo (opcional)	21
4.8.1.	Propósito.....	21
4.9.	/usr/local: Jerarquía local.....	21
4.9.1.	Propósito.....	21
4.9.2.	Requisitos.....	21
4.9.3.	Opciones específicas.....	21
4.9.4.	/usr/local/share: Jerarquía local independiente de la arquitectura.....	22
4.10.	/usr/sbin: Binarios estándar del sistema no esenciales.....	22
4.10.1.	Propósito.....	22
4.10.2.	Requisitos.....	22
4.11.	/usr/share: Datos independientes de la arquitectura.....	22
4.11.1.	Propósito.....	22
4.11.2.	Requisitos.....	23
4.11.3.	Opciones específicas.....	23
4.11.4.	/usr/share/color: Información de gestión del color.....	24
4.11.5.	/usr/share/dict: Listas de word.....	24
4.11.6.	/usr/share/man: Páginas de manual.....	24
4.11.7.	/usr/share/misc: Datos misceláneos independientes de la arquitectura.....	27
4.11.8.	/usr/share/ppd: Definiciones de impresora (opcional)	28
4.11.9.	/usr/share/sgml: Datos SGML (opcional)	28
4.11.10.	/usr/share/xml: Datos XML (opcional)	28
4.12.	/usr/src: Código Fuente (opcional)	29
4.12.1.	Propósito.....	29
5.	La Jerarquía /var.....	30
5.1.	Propósito.....	30
5.2.	Requisitos.....	30
5.3.	Opciones específicas.....	31
5.4.	/var/account: Registros contables de procesos.....	31
5.4.1.	Propósito.....	31
5.5.	/var/cache: Datos cache de aplicación.....	31
5.5.1.	Propósito.....	31
5.5.2.	Opciones específicas.....	31
5.5.3.	/var/cache/fonts: Fuentes generadas localmente (opcional)	32
5.5.4.	/var/cache/man: Páginas manual formateadas localmente (opcional).....	32
5.6.	/var/crash: Volcados por caída del sistema (opcional)	33

5.6.1.	Propósito.....	33
5.7.	/var/games: Datos variables de juego (opcional).....	33
5.7.1.	Propósito.....	33
5.8.	/var/lib: Información de estado variable.....	33
5.8.1.	Propósito.....	33
5.8.2.	Requisitos.....	34
5.8.3.	Opciones específicas.....	34
5.8.4.	/var/lib/<editor>: Estado y archivos de copia de seguridad del editor (opcional).....	34
5.8.5.	/var/lib/color: Información de gestión del color (opcional)	35
5.8.6.	/var/lib/hwclock: Directorio de estado para hwclock(opcional)	35
5.8.7.	/var/lib/misc: Datos variables misceláneos.....	35
5.9.	/var/lock: Archivos de bloqueo.....	35
5.9.1.	Propósito.....	35
5.10.	/var/log: Archivos de registro y directorios.....	36
5.10.1.	Propósito.....	36
5.10.2.	Opciones específicas.....	36
5.11.	/var/mail: Archivos del buzón de usuario (opcional)	36
5.11.1.	Propósito.....	36
5.12.	/var/opt: Datos variables para /opt.....	36
5.12.1.	Propósito.....	36
5.13.	/var/run: Datos variables en tiempo de ejecución.....	37
5.13.1.	Propósito.....	37
5.13.2.	Requisitos.....	37
5.14.	/var/spool: Datos del carrete de aplicación.....	37
5.14.1.	Propósito.....	37
5.14.2.	Opciones específicas.....	37
5.14.3.	/var/spool/lpd: Colas de impresión daemon de impresoras línea (opcional).....	37
5.14.4.	/var/spool/rwho: Archivos Rwho (opcional)	38
5.15.	/var/tmp: Archivos temporales preservados entre reinicios.....	38
5.15.1.	Propósito.....	38
5.16.	/var/yp: Archivos de base de datos del Servicio de Información de Red (NIS) (opcional).....	38
5.16.1.	Propósito.....	38
6.	Anexo específico del sistema operativo.....	39
6.1.	Linux.....	39
6.1.1.	/: Directorio Raíz.....	39
6.1.2.	/bin : Binarios de comando de usuarios esenciales (uso para todos los usuarios).....	39
6.1.3.	/dev: Dispositivos y archivos especiales.....	39
6.1.4.	/etc: Configuración del sistema específico del host.....	39
6.1.5.	/proc: Sistema de archivos virtual informativo del kernel y procesos.....	40
6.1.6.	/sbin: Binarios esenciales del sistema.....	40
6.1.7.	/sys: Sistema de archivos virtual informativo del kernel y del sistema.....	41
6.1.8.	/usr/include: Archivos de encabezado incluidos por programas C.....	41
6.1.9.	/usr/src: Código fuente.....	41
6.1.10.	/var/spool/cron: Trabajos cron y at.....	41
7.	Apéndice.....	42
7.1.	La lista de correo de FHS.....	42
7.2.	Antecedentes de la FHS.....	42
7.3.	Reglas generales.....	42
7.4.	Alcance.....	42
7.5.	Agradecimientos.....	43
7.6.	Colaboradores.....	43

Capítulo 1. Introducción

1.1. Propósito

Este estándar permite:

- Al Software predecir la ubicación de los archivos y directorios instalados.
- A los Usuarios predecir la ubicación de los archivos y directorios instalados.

Hacemos esto para:

- Especificar los principios rectores para cada área del sistema de archivos.
- Especificar los archivos y directorios mínimos requeridos.
- Enumerar las excepciones a los principios.
- Enumerar casos concretos en los que se ha producido un conflicto histórico.

El documento FHS es utilizado por:

- Proveedores de software independientes para crear aplicaciones que cumplan con FHS y trabajen con distribuciones que cumplan con FHS.
- Creadores de SO para proporcionar sistemas que cumplan con FHS.
- Usuarios, para comprender y mantener el cumplimiento de FHS de un sistema.

El documento FHS tiene un alcance limitado:

- La ubicación local de archivos locales es un problema local, por lo que FHS no intenta usurpar a los administradores del sistema.
- FHS aborda problemas en los que la ubicación de archivos debe coordinarse entre varias partes, como sitios locales, distribuciones, aplicaciones, documentación, etc.

1.2. Convenciones

Le recomendamos que lea una versión tipográfica de este documento en lugar de la versión en texto sin formato. En la versión de composición tipográfica, los nombres de archivos y directorios se muestran en una fuente de ancho constante.

Los componentes de los nombres de archivo que varían se representan mediante una descripción del contenido incluido en caracteres "<" y ">", *<así>*. Las direcciones de correo electrónico también se incluyen entre "<" y ">", pero se muestran con el tipo de letra habitual.

Los componentes opcionales de los nombres de archivo se incluyen entre "/" y "]" y puede combinarse con la convención "<" y ">". Por ejemplo, si se permite que un nombre de archivo aparezca con o sin extensión, podría estar representado por *<nombre de archivo> [.<extensión>]*.

Las subcadenas variables de nombres de directorio y nombres de archivo se indican mediante "*".

Las secciones del texto marcadas como *Razones Fundamentales* son explicativas y no normativas.

Capítulo 2. El Sistema de Archivos

Este estándar asume que el sistema operativo subyacente a un sistema de archivos compatible con FHS tiene las mismas características de seguridad básicas que se encuentran en la mayoría de los sistemas de archivos UNIX.

Es posible definir dos distinciones independientes entre archivos: compartible versus no compartible y variable versus estático. En general, los archivos que difieren en cualquiera de estos aspectos deben ubicarse en directorios diferentes. Esto facilita el almacenamiento de archivos con diferentes características de uso en diferentes sistemas de archivos.

Los archivos "compartibles" son aquellos que pueden almacenarse en un host y usarse en otros. Los archivos "no compartibles" son aquellos que no se pueden compartir. Por ejemplo, los archivos de los directorios *home* de los usuarios se pueden compartir, mientras que los archivos de bloqueo del dispositivo no.

Los archivos "estáticos" incluyen binarios, bibliotecas, archivos de documentación y otros archivos que no cambian sin la intervención del administrador del sistema. Los archivos "variables" son archivos que no son estáticos.

Razón Fundamental

Los archivos que se pueden compartir pueden almacenarse en un host y usarse en varios otros. Sin embargo, por lo general, no todos los archivos de la jerarquía del sistema de archivos se pueden compartir, por lo que cada sistema tiene un almacenamiento local que contiene al menos los archivos que no se pueden compartir. Es conveniente si todos los archivos que requiere un sistema y que están almacenados en un host externo pueden estar disponibles montando uno o varios directorios del host externo.

Los archivos estáticos y variables deben separarse porque los archivos estáticos, a diferencia de los archivos variables, se pueden almacenar en medios de solo lectura y no es necesario hacer copias de seguridad en el mismo procedimiento que copie los archivos variables.

Las jerarquías de archivos de sistemas históricos similares a UNIX contenían archivos estáticos y variables en ambos */usr* y */etc*. Para conseguir las ventajas mencionadas anteriormente se creó la jerarquía */var* y todos los archivos de variables se transfirieron desde */usr* a */var*. Como consecuencia, */usr* ahora se puede montar como solo lectura (si es un sistema de archivos separado). Se han transferido archivos variables desde */etc* a */var* durante un período más largo, según lo ha permitido la tecnología.

A continuación, se muestra un ejemplo de un sistema compatible con FHS. (Son posibles otros diseños compatibles con FHS).

	compartible	imposible de compartir
Estático	<i>/usr</i>	<i>/etc</i>
	<i>/opt</i>	<i>/boot</i>
Variable	<i>/var/mail</i>	<i>/var/run</i>
	<i>/var/spool/news</i>	<i>/var/lock</i>

Capítulo 3. El Sistema de Archivos Raíz

3.1. Propósito

El contenido del sistema de archivos raíz debe ser adecuado para iniciar, restaurar, recuperar y/o reparar el sistema.

- Para arrancar un sistema, debe haber suficiente software y datos en la partición raíz para montar otros sistemas de archivos. Esto incluye utilidades, configuración, información del cargador de arranque (boot loader) y otros datos de inicio esenciales. `/usr`, `/opt`, y `/var` están diseñados de manera que puedan estar ubicados en otras particiones o sistemas de archivos.
- Para permitir la recuperación y/o reparación de un sistema, las utilidades que necesita un mantenedor experimentado para diagnosticar y reconstruir un sistema dañado deben estar presentes en el sistema de archivos raíz.
- Para restaurar un sistema, las utilidades necesarias para restaurar desde copias de seguridad del sistema (en disquete, cinta, etc.) deben estar presentes en el sistema de archivos raíz.

Razón Fundamental

Los requisitos mínimos para el sistema de archivos raíz deben ser tan pequeños como sea razonablemente posible, pero no más pequeños. Si bien es posible que muchos usuarios no deseen la complejidad adicional de un sistema particionado, la opción de mantener la raíz pequeña debe conservarse por varias razones:

- En ocasiones se monta desde soportes muy pequeños.
- El sistema de archivos raíz contiene muchos archivos de configuración específicos del sistema. Como ejemplo podemos mencionar un kernel, que es específico del sistema, un nombre de host específico, etc. Esto significa que el sistema de archivos raíz no siempre se puede compartir entre sistemas en red. Mantenerlo pequeño en servidores de sistemas en red minimiza la cantidad de espacio perdido para áreas de archivos que no se pueden compartir. También permite estaciones de trabajo con discos duros locales más pequeños.
- Si bien es posible tener el sistema de archivos raíz en una partición grande y llenarla al máximo, habrá otros usuarios con particiones más pequeñas. Si tienes más archivos instalados, puedes encontrar incompatibilidades con otros sistemas que utilicen sistemas de archivos raíz en particiones más pequeñas. Si eres desarrollador, es posible que estés convirtiendo tu opción en un problema para una gran cantidad de usuarios.
- Los errores de disco que corrompen los datos en el sistema de archivos raíz son un problema mayor que los errores en cualquier otra partición. Un sistema de archivos raíz pequeño es menos propenso a la corrupción como resultado de un bloqueo del sistema.

Estas consideraciones deben equilibrarse con la necesidad de un entorno operativo mínimamente útil, por el bien del proceso de arranque, así como en situaciones de recuperación de fallas.

Las aplicaciones nunca deben crear ni requerir archivos o subdirectorios especiales en el directorio raíz. Otras ubicaciones en la jerarquía de FHS brindan flexibilidad más que suficiente para cualquier paquete.

Razón Fundamental

Hay varias razones por las que está prohibido crear un nuevo subdirectorio del sistema de archivos raíz:

- Requiere espacio en una partición raíz que el administrador del sistema puede querer que se mantenga pequeña y simple por razones de rendimiento o de seguridad.

- Evita cualquier disciplina que el administrador del sistema pueda haber establecido para distribuir jerarquías de archivos estándar en los volúmenes montables.

Las distribuciones no deben crear nuevos directorios en la jerarquía raíz sin una consideración extremadamente cuidadosa de las consecuencias, incluida la portabilidad de la aplicación.

3.2. Requisitos

Los siguientes directorios, o enlaces simbólicos a directorios, son obligatorios en /.

Directorio	Descripción
bin	Binarios de comandos esenciales
boot	Archivos estáticos del cargador de arranque
dev	Archivos de dispositivo
etc	Configuración del sistema específico del host
lib	Bibliotecas compartidas y módulos del kernel esenciales
media	Punto de montaje para medios extraíbles
mnt	Punto de montaje para un sistema de archivos temporal
opt	Paquetes de software de aplicaciones complementarias
run	Datos relevantes para los procesos en ejecución
sbin	Binarios esenciales del sistema
srv	Datos de los servicios prestados por este sistema
tmp	Archivos temporales
usr	Jerarquía secundaria
var	Datos variables

Cada directorio enumerado anteriormente se especifica en detalle en subsecciones separadas a continuación. `/usr` y `/var` cada uno tiene una sección completa en este documento debido a la complejidad de esos directorios.

3.3. Opciones específicas

Los siguientes directorios, o enlaces simbólicos a directorios, deben estar en /, si el subsistema correspondiente está instalado:

Directorio	Descripción
home	Directorios de inicio de usuario (opcional)
lib<qual>	Bibliotecas compartidas esenciales de formato alternativo (opcional)
root	Directorio de inicio para el usuario raíz (opcional)

Cada directorio enumerado anteriormente se especifica en detalle en subsecciones separadas a continuación.

3.4. /bin : Binarios de comandos de usuario esenciales (para todos los usuarios)

3.4.1. Propósito

/bin contiene comandos que pueden ser utilizados tanto por el administrador del sistema como por los usuarios, pero que son necesarios cuando no hay otros sistemas de archivos montados (por ejemplo, en modo de usuario único). También puede contener comandos que los scripts utilizan indirectamente.¹

3.4.2. Requisitos

No debe haber subdirectorios en /bin.

Los siguientes comandos, o enlaces simbólicos a comandos, son necesarios en /bin:

Comando	Descripción
cat	Utilidad para concatenar archivos a la salida estándar
chgrp	Utilidad para cambiar la propiedad del grupo de archivos
chmod	Utilidad para cambiar los permisos de acceso a archivos
chown	Utilidad para cambiar el propietario y el grupo del archivo
cp	Utilidad para copiar archivos y directorios
date	Utilidad para imprimir o configurar los datos y la hora del sistema
dd	Utilidad para convertir y copiar un archivo
df	Utilidad para informar sobre el uso del espacio en disco del sistema de archivos
dmesg	Utilidad para imprimir o controlar el búfer de mensajes del kernel
echo	Utilidad para mostrar una línea de texto
false	Utilidad para no hacer nada, sin éxito
hostname	Utilidad para mostrar o establecer el nombre de host del sistema
kill	Utilidad para enviar señales a procesos
ln	Utilidad para hacer enlaces entre archivos
login	Utilidad para iniciar una sesión en el sistema
ls	Utilidad para enumerar el contenido del directorio
mkdir	Utilidad para hacer directorios
mknod	Utilidad para hacer archivos especiales de bloques o personajes
more	Utilidad para hojear el texto
mount	Utilidad para montar un sistema de archivos
mv	Utilidad para mover / renombrar archivos
ps	Utilidad para informar el estado del proceso
pwd	Utilidad para imprimir el nombre del directorio de trabajo actual
rm	Utilidad para eliminar archivos o directorios

¹ Los archivos binarios de comandos que no son lo suficientemente esenciales como para colocarlos en /bin deben colocarse en /usr/bin. Elementos que solo son necesarios para usuarios no raíz (el sistema X Window, chsh, etc.) generalmente no son lo suficientemente esenciales como para colocarse en la partición raíz.

Comando	Descripción
rmdir	Utilidad para eliminar directorios vacíos
sed	El editor de flujo 'sed'
sh	Shell de comando compatible con POSIX
stty	Utilidad para cambiar e imprimir la configuración de la línea del terminal
su	Utilidad para cambiar la identificación de usuario
sync	Utilidad para vaciar los búferes del sistema de archivos
true	Utilidad para no hacer nada, con éxito
umount	Utilidad para desmontar sistemas de archivos
uname	Utilidad para imprimir información del sistema

Si **/bin/sh** no es el comando de shell compatible con POSIX en sí, debe ser un enlace duro o simbólico al comando de shell real.

Los comandos **|** y **test** deben colocarse juntos en **/bin** o en **/usr/bin**.

Razón Fundamental

Varios shells se comportan de manera diferente cuando se llaman como **sh**, a fin de preservar la compatibilidad POSIX y al mismo tiempo permitir cambios o extensiones a POSIX cuando se desee.

El requisito para el **|** y los comandos **test** que se incluirán como binarios (incluso si están implementados internamente por el shell) se comparten con el estándar POSIX.1-2008.

3.4.3. Opciones específicas

Los siguientes programas, o enlaces simbólicos a programas, deben estar en **/bin** si el subsistema correspondiente está instalado:

Comando	
csh	El shell de C (opcional)
ed	El editor ed (opcional)
tar	La utilidad de empaquetado tar (opcional)
cpio	La utilidad de empaquetado cpio (opcional)
gzip	La utilidad de compresión de GNU (opcional)
gunzip	La utilidad de descompresión de GNU (opcional)
zcat	La utilidad de descompresión de GNU (opcional)
netstat	La utilidad de estadísticas de red (opcional)
ping	La utilidad de verificación de red ICMP (opcional)

/bin/csh puede ser un enlace simbólico a **/bin/tcsh** o **/usr/bin/tcsh**.

Razón Fundamental

Los comandos **tar**, **gzip** y **cpio** se han agregado para hacer posible la restauración de un sistema (si el / ofrecido está intacto).

Por el contrario, si nunca se espera una restauración desde la partición raíz, entonces estos binarios podrían omitirse (por ejemplo, un chip ROM raíz, montado **/usr** a través de NFS). Si la restauración de un sistema se planifica a través de la red, entonces, **ftp** o **tftp** (junto con todo lo necesario para obtener una conexión ftp) debe estar disponible en la partición raíz.

3.5. /boot : Archivos estáticos del cargador de arranque

3.5.1. Propósito

Este directorio contiene todo lo necesario para el proceso de inicio, excepto los archivos de configuración que no se necesitan en el momento del inicio y el instalador del mapa. Por lo tanto `/boot` almacena datos que se utilizan antes de que el kernel comience a ejecutar programas en modo usuario. Esto puede incluir sectores de arranque maestro guardados y archivos de mapa de sector.

Los programas necesarios para que el cargador de arranque pueda arrancar un archivo deben colocarse en `/sbin`. Los archivos de configuración para cargadores de arranque que no son necesarios en el momento del arranque deben colocarse en `/etc`.

3.5.2. Opciones Específicas

El kernel del sistema operativo debe estar ubicado en `/` o `/boot`.

Algunas arquitecturas pueden tener otros requisitos para `/boot` relacionados con limitaciones o expectativas específicas de esa arquitectura. Estos requisitos no se enumeran aquí; Las distribuciones pueden agregar requisitos según sea necesario para permitir el inicio del sistema en estas arquitecturas.

3.6. /dev : Archivos de dispositivo

3.6.1. Propósito

El directorio `/dev` es la ubicación de archivos especiales o de dispositivo.

3.6.2. Opciones Específicas

Si existe la posibilidad de que los dispositivos en `/dev` deban crearse manualmente, `/dev` debe contener un comando llamado `MAKEDEV`, que puede crear dispositivos según sea necesario. También puede contener un `MAKEDEV.local` para cualquier dispositivo local.

Si es requerido, `MAKEDEV` debe tener provisiones para crear cualquier dispositivo que se pueda encontrar en el sistema, no solo aquellos que instala una distribución en particular.

3.7. /etc : Configuración del sistema específico del host

3.7.1. Propósito

La jerarquía `/etc` contiene archivos de configuración. Un "archivo de configuración" es un archivo local que se utiliza para controlar el funcionamiento de un programa; debe ser estático y no puede ser un binario ejecutable.²

Se recomienda que los archivos se almacenen en subdirectorios de `/etc` en lugar de directamente en `/etc`.

3.7.2. Requisitos

No se pueden ubicar binarios en `/etc`.

² Para ser claro, `/etc` puede contener scripts ejecutables, como los scripts de comando comúnmente llamados `init` para iniciar y apagar el sistema e iniciar procesos `daemon`. "Binario ejecutable" en este contexto se refiere a código de máquina directo o pseudocódigo que no está en un formato legible por humanos, como los nativos ejecutables ELF.

Los siguientes directorios o enlaces simbólicos a directorios son obligatorios en `/etc`:

Directorio	Descripción
opt	Configuración para <code>/opt</code>

3.7.3. Opciones Específicas

Los siguientes directorios o enlaces simbólicos a directorios deben estar en `/etc`, si el subsistema correspondiente está instalado:

Directorio	Descripción
X11	Configuración para el sistema X Window (opcional)
sgml	Configuración para SGML (opcional)
xml	Configuración para XML (opcional)

Los siguientes archivos, o enlaces simbólicos a archivos, deben estar en `/etc` si el subsistema correspondiente está instalado:

Archivo	Descripción
csch.login	Archivo de inicialización de todo el sistema para inicios de sesión de Shell C (opcional)
exports	Lista de control de acceso al sistema de archivos NFS (opcional)
fstab	Información estática sobre sistemas de archivos (opcional)
ftputers	Lista de control de acceso de usuarios de daemon FTP (opcional)
gateways	Archivo que enumera las puertas de enlace para enrutar (opcional)
gettydefs	Configuración de velocidad y terminal utilizada por <code>getty</code> (opcional)
group	Archivo de grupo de usuarios (opcional)
host.conf	Archivo de configuración del solucionador (opcional)
hosts	Información estática sobre nombres de host (opcional)
hosts.allow	Archivo de acceso de host para envoltorios TCP (opcional)
hosts.deny	Archivo de acceso de host para envoltorios TCP (opcional)
hosts.equiv	Lista de hosts confiables para <code>rlogin</code> , <code>rsh</code> , <code>rcp</code> (opcional)
hosts.lpd	Lista de hosts confiables para <code>lpd</code> (opcional)
inetd.conf	Archivo de configuración para <code>inetd</code> (opcional)
inittab	Archivo de configuración para <code>init</code> (opcional)
issue	Mensaje previo al inicio de sesión y archivo de identificación (opcional)
ld.so.conf	Lista de directorios adicionales para buscar bibliotecas compartidas (opcional)
motd	Mensaje posterior al inicio de sesión del archivo del día (opcional)
mtab	Información dinámica sobre sistemas de archivos (opcional)
mttools.conf	Archivo de configuración para <code>mttools</code> (opcional)

³ Los sistemas que utilizan el paquete de contraseñas ocultas tendrán archivos de configuración adicionales en `/etc (/etc/shadow` y otros) y programas en `/usr/sbin (useradd, usermod, y otros)`.

Archivo	Descripción
networks	Información estática sobre nombres de redes (opcional)
passwd	El archivo de contraseña (opcional)
printcap	La base de datos de capacidad de la impresora lpd (opcional)
profile	Archivo de inicialización de todo el sistema para inicios de sesión de Shell sh (opcional)
protocols	Listado de protocolos IP (opcional)
resolv.conf	Archivo de configuración del solucionador (opcional)
rpc	Listado de protocolos RPC (opcional)
securetty	Control de acceso TTY para inicio de sesión root (opcional)
services	Nombres de puertos para servicios de red (opcional)
shells	Nombres de ruta de shells de inicio de sesión válidos (opcional)
syslog.conf	Archivo de configuración para syslogd (opcional)

`mtab` no se ajusta a la naturaleza estática de `/etc`: se exceptúa por razones históricas. ⁴

3.7.4. `/etc/opt` : Archivos de configuración para `/opt`

3.7.4.1. Propósito

Los archivos de configuración específicos del host para los paquetes de software de la aplicación complementaria deben instalarse dentro del directorio `/etc/opt/<subdir>`, donde `<subdir>` es el nombre del subárbol en `/opt` donde se almacenan los datos estáticos de ese paquete.

3.7.4.2. Requisitos

No se impone ninguna estructura a la disposición interna de `/etc/opt/<subdir>`.

Si un archivo de configuración debe residir en una ubicación diferente para que el paquete o sistema funcione correctamente, puede colocarse en una ubicación distinta de `/etc/opt/<subdir>`.

Razón Fundamental

Consulte las razones de `/opt`.

3.7.5. `/etc/X11` : Configuración para el sistema Window X (opcional)

3.7.5.1. Propósito

`/etc/X11` es la ubicación para todas las configuraciones específicas del host X11. Este directorio es necesario para permitir el control local si `/usr` está montado para solo lectura.

3.7.5.2. Opciones Específicas

Los siguientes archivos, o enlaces simbólicos a archivos, deben estar en `/etc/X11` si el subsistema correspondiente está instalado:

⁴ En algunos sistemas Linux, esto puede ser un enlace simbólico a `/proc/mounts`, en cuyo caso esta excepción no es necesaria.

Archivo	Descripción
xorg.conf	El archivo de configuración para las versiones 7 y posteriores de X.org (opcional)
Xmodmap	Archivo de modificación de teclado global X11 (opcional)

Subdirectorios de `/etc/X11` puede incluir los de `xdm` y para cualquier otro programa (algunos gestores de ventanas, por ejemplo) que los necesite.

3.7.6. `/etc/sgml` : Archivos de configuración para SGML

3.7.6.1. Propósito

Los archivos de configuración genéricos que definen los parámetros de alto nivel de los sistemas SGML se instalan aquí. Archivos con nombres `*.conf` indican archivos de configuración genéricos. Archivo con nombres `*.cat` son los catálogos centralizados específicos de DTD, que contienen referencias a todos los demás catálogos necesarios para utilizar la DTD determinada. El archivo de súper catálogo `catalog` hace referencia a todos los catálogos centralizados.

3.7.7. `/etc/xml` : Archivos de configuración para XML

3.7.7.1. Propósito

Aquí se instalan los archivos de configuración genéricos que definen los parámetros de alto nivel de los sistemas XML. Archivos con nombres `*.conf` indican archivos de configuración genéricos. El archivo de catálogo super `catalog` hace referencia a todos los catálogos centralizados.

3.8. `/home` : Directorios de inicio de usuario (opcional)

3.8.1. Propósito

`/home` es un concepto bastante estándar, pero claramente es un sistema de archivos específico del sitio.⁶ La configuración será diferente de un host a otro. Por lo tanto, ningún programa debe asumir una ubicación específica para un directorio de inicio, sino que debe consultarlo.⁷

3.8.2. Requisitos

Los archivos de configuración específicos del usuario para aplicaciones se almacenan en el directorio de inicio del usuario en un archivo que comienza con el carácter `'.'` (un "archivo punto"). Si una aplicación necesita crear más de un archivo punto, entonces deben colocarse en un subdirectorio con un nombre que comience con el carácter `'.'` (un "directorio punto"). En este caso, los archivos de configuración no deberían comenzar con el carácter `'.'`⁸

⁵ `/etc/X11/xdm` contiene los archivos de configuración para `xdm`. Estos son la mayoría de los archivos encontrados anteriormente en `/usr/lib/X11/xdm`. Algunos datos de variables locales para `xdm` se almacena en `/var/lib/xdm`.

⁶ Diferentes personas prefieren colocar cuentas de usuario en una variedad de lugares. Esta sección describe solo una ubicación sugerida para los directorios de inicio de los usuarios; no obstante, recomendamos que todas las distribuciones compatibles con FHS utilicen esta ubicación como la ubicación predeterminada para los directorios de inicio de los usuarios. Las cuentas que no son de inicio de sesión creadas con fines administrativos a menudo tienen sus directorios de inicio en otro lugar.

En sistemas más pequeños, el directorio de inicio de cada usuario se implementa normalmente como un subdirectorio directamente debajo de `/home`, por ejemplo `/home/smith`, `/home/torvalds`, `/home/operator`, etc. En sistemas grandes (especialmente cuando los directorios `/home` se comparten entre muchos hosts que utilizan NFS) es útil subdividir los directorios de inicio de los usuarios. La subdivisión se puede lograr mediante el uso de subdirectorios como `/home/staff`, `/home/guests`, `/home/students`, etc.

⁷ Para encontrar el directorio de inicio de un usuario, usa una función de biblioteca como `getpwent`, `getpwent_r` de `fgetpwent` en lugar de depender de `/etc/passwd` porque la información del usuario puede almacenarse de forma remota utilizando sistemas como NIS.

⁸ Se recomienda que, además de guardar automáticamente y bloquear archivos, los programas se abstengan de crear archivos o directorios que no sean dot en un directorio de inicio sin el consentimiento del usuario.

3.8.3. Especificaciones y convenciones del directorio de inicio

Se han realizado varios esfuerzos en el pasado para estandarizar el diseño de los directorios de inicio, incluida la especificación XDG Base Directories ⁹ y las convenciones GLib sobre el contenido del directorio de usuarios. ¹⁰ Es posible realizar esfuerzos adicionales en esta dirección en el futuro. Para adaptarse al software que hace uso de estas especificaciones y convenciones, las distribuciones pueden crear jerarquías de directorios que sigan las especificaciones y convenciones. Esas jerarquías de directorios pueden estar ubicadas debajo de los directorios de inicio.

3.9. /lib : Módulos de kernel y bibliotecas compartidas esenciales

3.9.1. Propósito

El directorio /lib contiene las imágenes de la biblioteca compartida necesarias para iniciar el sistema y ejecutar los comandos en el sistema de archivos raíz, es decir, por binarios en /bin y /sbin. ¹¹

3.9.2. Requisitos

Se requiere al menos uno de cada uno de los siguientes patrones de nombre de archivo (pueden ser archivos o enlaces simbólicos):

Archivo	Descripción
libc.so. *	La biblioteca C vinculada dinámicamente (opcional)
ld *	El vinculador / cargador de tiempo de ejecución (opcional)

Si se instala un preprocesador C, /lib/cpp debe ser una referencia a él, por razones históricas. ¹²

3.9.3. Opciones específicas

Los siguientes directorios, o enlaces simbólicos a directorios, deben estar en /lib, si el subsistema correspondiente está instalado:

Directorio	Descripción
Modules	Módulos cargables de kernel (opcional)

3.10. /lib<qual> : Bibliotecas compartidas esenciales de formato alternativo (opcional)

3.10.1. Propósito

Puede haber una o más variantes del directorio /lib en sistemas que admiten más de un formato binario el cual requiere bibliotecas independientes. ¹³

⁹ Se encuentra en <http://standards.freedesktop.org/basedir-spec/basedir-spec-latest.html> y <http://www.freedesktop.org/wiki/Software/xdg-user-dirs>.

¹⁰ Se puede encontrar una descripción de las convenciones de GLib en la documentación de GUserDirectory, en <http://developer.gnome.org/glib/unstable/glib-Misiverse-Utility-Functions.html#GUserDirectory>.

¹¹ Bibliotecas compartidas que solo son necesarias para binarios en /usr (como cualquier binario de X Window) no debe estar en /lib. Solo las bibliotecas compartidas necesarias para ejecutar binarios en /bin y /sbin tal vez aquí. En particular, la biblioteca libm.so. * también se puede colocar en /usr/lib si no es requerido por nada en /bin o /sbin.

¹² La ubicación habitual de este binario es /usr/bin/cpp.¹³ Esto se usa comúnmente para soporte de 64 bits o 32 bits en sistemas que admiten múltiples formatos binarios, pero requieren bibliotecas con el mismo nombre.

En este caso, /lib32 y /lib64 podrían ser los directorios de la biblioteca, y /lib un enlace simbólico a uno de ellos.

3.10.2. Requisitos

Si existe uno o más de estos directorios, los requisitos para su contenido son los mismos que los del directorio normal `/lib`, excepto que `/lib<qual>/cpp` no es requerido.¹⁴

3.11. `/media` : Punto de montaje para media extraíble

3.11.1. Propósito

Este directorio contiene subdirectorios que se utilizan como puntos de montaje para medios extraíbles como discos, cdroms y discos zip.

Razón Fundamental

Históricamente, se han utilizado otros lugares diferentes para montar medios extraíbles como `/cdrom`, `/mnt` o `/mnt/cdrom`. Colocar los puntos de montaje para todos los medios extraíbles directamente en el directorio raíz podría resultar en una gran cantidad de directorios adicionales en `/`. Aunque el uso de subdirectorios en `/mnt` como un punto de montaje ha sido común recientemente, entra en conflicto con una tradición mucho más antigua de usar `/mnt` directamente como un punto de montaje temporal.

3.11.2. Opciones específicas

Los siguientes directorios, o enlaces simbólicos a directorios, deben estar en `/media`, si el subsistema correspondiente está instalado:

Directorio	Descripción
<code>floppy</code>	Unidad de disquete (opcional)
<code>cdrom</code>	Unidad de CD-ROM (opcional)
<code>cdrecorder</code>	Grabadora de CD (opcional)
<code>zip</code>	Unidad zip (opcional)

En sistemas donde existe más de un dispositivo para montar un cierto tipo de medio, los directorios de montaje se pueden crear agregando un dígito al nombre de los disponibles arriba comenzando con '0', pero el nombre no calificado también debe existir.¹⁵

3.12. `/mnt`: Punto de montaje para un sistema de archivos montado temporalmente

3.12.1. Propósito

Este directorio se proporciona para que el administrador del sistema pueda montar temporalmente un sistema de archivos según sea necesario. El contenido de este directorio es un problema local y no debería afectar la forma en que se ejecuta ningún programa.

Este directorio no debe ser utilizado por programas de instalación: en su lugar, debe utilizarse un directorio temporal adecuado que no esté siendo utilizado por el sistema.

¹⁴ `/lib<qual>/cpp` todavía está permitido: esto permite el caso donde `/lib` y `/lib<qual>` son iguales (uno es un vínculo simbólico con el otro).

¹⁵ Una distribución compatible con dos unidades de CDROM puede tener `/media/cdrom0` y `/media/cdrom1` con `/media/cdrom` un enlace simbólico a cualquiera de estos.

3.13. /opt : Paquetes de software de aplicaciones complementarias

3.13.1. Propósito

/opt está reservado para la instalación de paquetes de software de aplicación complementarios.

Un paquete para ser instalado en /opt debe ubicar sus archivos estáticos en un /opt /<package> o un árbol de directorios /opt /<provider> donde <package> es un nombre que describe el paquete de software y <provider> es el nombre registrado LANANA del proveedor.

3.13.2. Requisitos

Directorio	Descripción
<package>	Objetos de paquete estático
<provider>	Nombre del proveedor registrado LANANA

Los directorios /opt/bin, /opt/doc, /opt/include, /opt/info, /opt/lib, y /opt/man están reservados para uso del administrador del sistema local. Los paquetes pueden proporcionar archivos "front-end" destinados a ser colocados en (enlazando o copiando) estos directorios reservados por el administrador del sistema local, pero deben funcionar normalmente en ausencia de estos directorios reservados.

Los programas que van a invocar los usuarios deben estar ubicados en el directorio /opt/<package>/bin o bajo la jerarquía /opt/<provider>. Si el paquete incluye páginas de manual de UNIX, deben estar ubicadas en /opt/<package>/share/man o bajo la jerarquía /opt/<provider>, y la misma subestructura que /usr/share/man debe ser usada.

Los archivos de paquete que son variables (cambios en el funcionamiento normal) deben instalarse en /var/opt. Consulte la sección sobre /var/opt para más información.

Los archivos de configuración específicos del host deben instalarse en /etc/opt. Consulte la sección sobre /etc para más información.

No pueden existir otros archivos de paquete fuera de jerarquías /opt, /var/opt, y /etc/opt a excepción de los archivos de paquetes que deben residir en ubicaciones específicas dentro del árbol del sistema de archivos para que funcionen correctamente. Por ejemplo, los archivos de bloqueo del dispositivo deben colocarse en /var/lock y los dispositivos deben estar ubicados en /dev.

Las distribuciones pueden instalar y administrar software en /opt bajo un subdirectorio debidamente registrado.

Razón Fundamental

El uso de /opt para software complementario es una práctica bien establecida en la comunidad UNIX. La interfaz binaria de la aplicación System V [AT&T 1990], basada en la definición de interfaz. System V (tercera edición), proporciona una estructura /opt muy similar a la definida aquí.

El estándar de compatibilidad binaria de Intel v. 2 (iBCS2) también proporciona una estructura similar para /opt.

Por lo general, todos los datos necesarios para admitir un paquete en un sistema deben estar presentes en `/opt /<package>`, incluidos los archivos destinados a ser copiados en `/etc/opt/<package>` y `/var/opt/<package>` así como directorios reservados en `/opt`.

Las restricciones menores sobre distribuciones usando `/opt` son necesarias porque es posible que haya conflictos entre el software instalado por distribución y el instalado localmente, especialmente en el caso de rutas de acceso fijas que se encuentran en algún software binario.

La estructura de los directorios a continuación `/opt/<provider>` se deja en manos del empaquetador del software, aunque se recomienda que los paquetes se instalen en `/opt/<provider>/<package>` y seguir una estructura similar a las pautas para `/opt/<package>`. Una razón válida para divergir de esta estructura es para los paquetes de soporte que pueden tener archivos instalados en `/opt/<provider>/lib` o `/opt/<provider>/bin`.

3.14. `/root` : Directorio de inicio para el usuario raíz (opcional)

3.14.1. Propósito

El directorio de inicio de la cuenta raíz puede ser determinado por el desarrollador o la preferencia local, pero esta es la ubicación predeterminada recomendada.¹⁶

3.15. `/run` : datos variables en tiempo de ejecución

3.15.1. Propósito

Este directorio contiene datos de información del sistema que describen el sistema desde que se inició. Los archivos de este directorio se deben borrar (eliminar o trunca según corresponda) al comienzo del proceso de inicio.

Los propósitos de este directorio fueron servidos una vez por `/var/run`. En general, los programas pueden seguir usando `/var/run` para cumplir los requisitos establecidos para `/run` a efectos de compatibilidad con versiones anteriores. Programas que han migrado para usar `/run` deberían dejar de usar `/var/run`, excepto como se indica en la sección sobre `/var/run`.

Los programas pueden tener un subdirectorio de `/run`; esto se recomienda para programas que utilizan más de un archivo de tiempo de ejecución. Los usuarios también pueden tener un subdirectorio de `/run`, aunque se debe tener cuidado de limitar adecuadamente los derechos de acceso para evitar el uso no autorizado de `/run` por sí mismo y otros subdirectorios.¹⁷

3.15.2. Requisitos

Archivos de identificador de proceso (PID), que se colocaron originalmente en `/etc`, deben colocarse en `/run`. La convención de nomenclatura para archivos PID es `<program-name>.pid`. Por ejemplo, el archivo `crond` PID se llama `/run/crond.pid`.

¹⁶ Si el directorio de inicio de la cuenta raíz no está almacenado en la partición raíz, será necesario asegurarse de que estará predeterminado en / si no se puede ubicar.

Recomendamos no usar la cuenta raíz para tareas que se pueden realizar como un usuario sin privilegios y que se use únicamente para la administración del sistema. Por esta razón, recomendamos que los subdirectorios de correo y otras aplicaciones no aparezcan en el directorio de inicio de la cuenta raíz, y que el correo para funciones de administración como root, postmaster y webmaster se reenvíe a un usuario apropiado.

¹⁷ `/run` no debe tener permisos de escritura para usuarios sin privilegios; es un problema de seguridad importante si cualquier usuario puede escribir en este directorio. Los subdirectorios específicos del usuario solo deben ser modificables por el propietario de cada directorio.

El formato interno de los archivos PID permanece sin cambios. El archivo debe constar del identificador de proceso en decimal codificado en ASCII, seguido de un carácter de nueva línea. Por ejemplo, si **crond** era el proceso número 25, `/run/crond.pid` contendría tres caracteres: dos, cinco y nueva línea.

Los programas que leen archivos PID deberían ser algo flexibles en lo que aceptan; es decir, deben ignorar los espacios en blanco adicionales, los ceros iniciales, la ausencia de una nueva línea al final o líneas adicionales en el archivo PID. Los programas que crean archivos PID deben usar la especificación simple ubicada en el párrafo anterior.

Los programas del sistema que mantienen sockets de dominio UNIX transitorios deben colocarlos en este directorio o en un subdirectorio apropiado como se describe anteriormente.

3.16. /sbin : Binarios del sistema

3.16.1. Propósito

Las utilidades utilizadas para la administración del sistema (y otros comandos exclusivos de raíz) se almacenan en `/sbin`, `/usr/sbin`, y `/usr/local/sbin`. `/sbin` contiene binarios esenciales para arrancar, restaurar, recuperar y/o reparar el sistema además de los binarios en `/bin`.¹⁸ Programas ejecutados después de `/usr` se sabe que se montan (cuando no hay problemas) generalmente se colocan en `/usr/sbin`. Los programas de administración del sistema instalados localmente deben colocarse en `/usr/local/sbin`.¹⁹

3.16.2. Requisitos

No debe haber subdirectorios en `/sbin`.

Los siguientes comandos, o enlaces simbólicos a comandos, son necesarios en `/sbin`:

Comando	Descripción
shutdown	Comando para apagar el sistema

3.16.3. Opciones específicas

Los siguientes archivos, o enlaces simbólicos a archivos, deben estar en `/sbin` si el subsistema correspondiente está instalado:

Comando	Descripción
fastboot	Reiniciar el sistema sin verificar los discos
fasthalt	Detener el sistema sin verificar los discos

¹⁸ Originalmente, `/sbin` los binarios se mantuvieron en `/etc`.

¹⁹ Decidir en que van las cosas en los directorios "`sbin`" es simple: si un usuario normal (no un administrador del sistema) lo ejecutará directamente, entonces debe colocarse en uno de los directorios "`bin`". Los usuarios habituales no deberían tener que colocar ninguno de los directorios `sbin` en su ruta.

Por ejemplo, archivos como **chfn** que los usuarios solo usan ocasionalmente deben colocarse en `/usr/bin`. **ping**, aunque es absolutamente necesario para root (recuperación y diagnóstico de red), los usuarios suelen utilizarlo y deben vivir en `/bin` por esta razón.

Recomendamos que los usuarios tengan permiso de lectura y ejecución para todo en `/sbin` excepto, quizás, ciertos programas `setuid` y `setgid`. La división entre `/bin` y `/sbin` no fue creada por razones de seguridad o para evitar que los usuarios vean el sistema operativo, sino para proporcionar una buena partición entre los binarios que todos usan y los que se usan principalmente para tareas de administración. No existe una ventaja de seguridad inherente al hacer `/sbin` fuera de los límites para los usuarios.

Comando	Descripción
fdisk	Manipulador de tabla de particiones (opcional)
fsck	Utilidad de verificación y reparación del sistema de archivos (opcional)
fsck.*	Utilidad de verificación y reparación del sistema de archivos para un sistema específico de archivos
getty	El programa getty (opcional)
halt	Comando para detener el sistema (opcional)
ifconfig	Configurar una interfaz de red (opcional)
init	Proceso inicial (opcional)
mkfs	Comando para construir un sistema de archivos (opcional)
mkfs.*	Comando para construir un sistema de archivos específico (opcional)
mkswap	Comando para configurar un área de intercambio (opcional)
reboot	Comando para reiniciar el sistema (opcional)
route	Utilidad de tabla de enrutamiento IP (opcional)
swapon	Habilitar paginación e intercambio (opcional)
swapoff	Deshabilitar la paginación y el intercambio (opcional)
update	Daemon para vaciar periódicamente los búferes del sistema de archivos

3.17. /srv : Datos de los servicios proporcionados por este sistema

3.17.1. Propósito

/srv contiene datos específicos del sitio que son proporcionados por este sistema.

Razón Fundamental

El propósito principal de especificar esto es para que los usuarios puedan encontrar la ubicación de los archivos de datos para un servicio en particular y para que los servicios que requieren un solo árbol para datos de solo lectura, datos de escritura y scripts (como scripts cgi) puedan ubicarse razonablemente. Los datos que solo son de interés para un usuario específico deben ir en el directorio de inicio de ese usuario. Si el directorio y la estructura de archivos de los datos no están expuestos a los consumidores, debe ir en /var/lib.

La metodología utilizada para nombrar subdirectorios de /srv no está especificada ya que actualmente no hay consenso sobre cómo se debe hacer esto. Un método para estructurar datos en /srv es por protocolo, por ejemplo, ftp, rsync, www y cvs. En sistemas grandes puede resultar útil estructurar /srv por contexto administrativo, como /srv/physics/www, /srv/compsci/cvs, etc. Esta configuración será diferente de un host a otro. Por lo tanto, ningún programa debería depender de una estructura de subdirectorios específica de /srv datos existentes o que necesariamente se almacenan en /srv. Sin embargo /srv siempre debe existir en sistemas compatibles con FHS y debe usarse como la ubicación predeterminada para dichos datos.

Las distribuciones deben tener cuidado de no eliminar archivos ubicados localmente en estos directorios sin el permiso del administrador.²⁰

²⁰ Esto es particularmente importante ya que estas áreas a menudo contendrán tanto los archivos instalados inicialmente por el distribuidor como los agregados por el administrador.

3.18. /tmp: Archivos temporales

3.18.1. Propósito

El directorio /tmp debe estar disponible para programas que requieran archivos temporales.

Los programas no deben asumir que se conservan archivo o directorios en /tmp entre invocaciones del programa.

Razón Fundamental

El estándar POSIX.1-2008 de IEEE enumera requisitos similares a los de la sección anterior.

Aunque los datos almacenados en /tmp pueden eliminarse de una manera específica del sitio, se recomienda que los archivos y directorios ubicados en /tmp se borren siempre que se inicie el sistema.

FHS agregó esta recomendación sobre la base de un precedente histórico y una práctica común, pero no la convirtió en un requisito porque la administración del sistema no está dentro del alcance de esta norma

Capítulo 4. La jerarquía /usr

4.1. Propósito

/usr es la segunda sección principal del sistema de archivos. /usr son datos de solo lectura que se pueden compartir. Eso significa que /usr debe poder compartirse entre varios hosts compatibles con FHS y no se le debe de escribir. Cualquier información que sea específica del host o que varíe con el tiempo se almacena en otro lugar.

Los paquetes de software grandes no deben utilizar un subdirectorio directo debajo de la jerarquía /usr.

4.2. Requisitos

Los siguientes directorios, o enlaces simbólicos a directorios, son obligatorios en /usr.

Directorio	Descripción
bin	La mayoría de los comandos de usuario
lib	Bibliotecas
local	Jerarquía local (vacía después de la instalación principal)
sbin	Binarios del sistema no vitales
share	Datos independientes de la arquitectura

4.3. Opciones específicas

Directorio	Descripción
games	Juegos y binarios educativos (opcional)
include	Archivos de encabezado incluidos por programas C
libexec	Binarios ejecutados por otros programas (opcional)
lib<qual>	Bibliotecas de formato alternativo (opcional)
src	Código fuente (opcional)

Se hace una excepción para el sistema X Window debido a un precedente considerable y una práctica ampliamente aceptada.

Pueden estar presentes los siguientes enlaces simbólicos a directorios. Esta posibilidad se basa en la necesidad de preservar la compatibilidad con sistemas más antiguos hasta que se pueda suponer que todas las distribuciones utilizan la jerarquía /var.

```
/usr/spool -> /var/spool
/usr/tmp -> /var/tmp
/usr/spool/locks -> /var/lock
```

Una vez que un sistema ya no requiere ninguno de los enlaces simbólicos anteriores, el enlace puede eliminarse, si lo desea.

4.4. /usr/bin : La mayoría de los comandos de usuario

4.4.1. Propósito

Este es el directorio principal de comandos ejecutables en el sistema.

4.4.2. Requisitos

No debe haber subdirectorios en /usr/bin.

4.4.3. Opciones específicas

Los siguientes archivos, o enlaces simbólicos a archivos, deben estar en /usr/bin, si el subsistema correspondiente está instalado:

Comando	Descripción
perl	El lenguaje práctico de extracción e informe (opcional)
python	El lenguaje interpretado de Python (opcional)
tclsh	Shell simple que contiene el intérprete de Tcl (opcional)
wish	Carcasa de ventana simple Tcl/Tk (opcional)
expect	Programa para diálogo interactivo (opcional)

Razón Fundamental

En muchos scripts ejecutables, el intérprete que se invocará para ejecutar el script se especifica mediante `#!path_to_interpreter` en la primera línea de un script. Para que dichos scripts sean portables entre diferentes sistemas, es ventajoso estandarizar las ubicaciones de los intérpretes. El intérprete de shell ya está fijo en /bin según esta especificación, pero los intérpretes para Perl, Python, Tcl y esperan se pueden instalar en varios lugares. Las ubicaciones especificadas aquí pueden implementarse como enlaces simbólicos a la ubicación física de los intérpretes.

4.5. /usr/include : Directorio para archivos de inclusión estándar.

4.5.1. Propósito

Aquí es donde se deben colocar todos los archivos de inclusión de uso general del sistema para el lenguaje de programación C.

4.5.2. Opciones específicas

Los siguientes directorios, o enlaces simbólicos a directorios, deben estar en /usr/include, si el subsistema correspondiente está instalado:

Directorio	Descripción
bsd	La compatibilidad con BSD incluye archivos (opcional)

4.6. /usr/lib : Bibliotecas para programación y paquetes

4.6.1. Propósito

/usr/lib incluye archivos de objetos y bibliotecas. ¹ En algunos sistemas, también puede incluir binarios internos que no están destinados a ser ejecutados directamente por usuarios o scripts de shell. ²

Las aplicaciones pueden usar un solo subdirectorio en /usr/lib. Si una aplicación utiliza un subdirectorio, todos los datos dependientes de la arquitectura utilizados exclusivamente por la aplicación deben colocarse dentro de ese subdirectorio. ³

4.6.2. Opciones específicas

Por razones históricas, /usr/lib/sendmail debe ser un enlace simbólico que resuelve el comando compatible *sendmail*- proporcionado por el agente de transferencia de correo del sistema, si este último existe. ^{4 5}

4.7. /usr/libexec : Binarios ejecutados por otros programas (opcional)

4.7.1. Propósito

/usr/libexec incluye binarios internos que no están destinados a ser ejecutados directamente por usuarios o scripts de shell. Las aplicaciones pueden usar un solo subdirectorio en /usr/libexec.

Aplicaciones que utilizan /usr/libexec de esta manera no deben usar también /usr/lib para almacenar binarios internos, aunque pueden usar /usr/lib para los demás propósitos documentados aquí.

Razón Fundamental

Algunas versiones anteriores de este documento no admitían /usr/libexec, a pesar de ser una práctica estándar en varios entornos. ⁶ Para adaptarse a esta restricción, se convirtió en una práctica común utilizar /usr/lib en lugar de. Cualquiera de las dos prácticas es ahora aceptable, pero cada aplicación debe elegir una forma u otra para organizarse.

¹ Varios archivos estáticos y subdirectorios específicos de la aplicación independientes de la arquitectura deben colocarse en /usr/ share.

² Ve a continuación, en la sección /usr/libexec, para una discusión de /usr/lib vs. /usr/libexec para binarios ejecutables.

³ Por ejemplo, el subdirectorio perl5 para módulos y bibliotecas de Perl ⁵.

⁴ Algunos comandos ejecutables como **makewhatis** y **sendmail** también se han colocado tradicionalmente en /usr/lib. **makewhatis** es un binario interno y debe colocarse en un directorio binario; los usuarios solo acceden a **catman**. Los binarios **sendmail** nuevos ahora se colocan por defecto en /usr/sbin. Además, los sistemas que utilizan un agente *sendmail*-compatible deben proporcionar /usr/sbin/sendmail como el comando **sendmail**, ya sea como el propio ejecutable o como un enlace simbólico al ejecutable apropiado.

⁵ Los datos específicos del host para el sistema X Window no deben almacenarse en /usr/lib/X11. Archivos de configuración específicos del host, como xorg.conf debe almacenarse en /etc/X11. Esto incluye datos de configuración como system.twmrc incluso si solo se hace un enlace simbólico a un archivo de configuración más global (probablemente en /usr/lib/X11).

⁶ Consulte, por ejemplo, los "Estándares de codificación GNU" de la Free Software Foundation.

4.8. /usr/lib<qual> : Bibliotecas de formato alternativo (opcional)

4.8.1. Propósito

/usr/lib<qual> realiza el mismo papel que /usr/lib para un formato binario alternativo, excepto que los enlaces simbólicos /usr/lib<qual>/sendmail y /usr/lib<qual>/X11 no son necesarios.⁷

4.9. /usr/local : Jerarquía local

4.9.1. Propósito

La jerarquía /usr/local es para que la utilice el administrador del sistema al instalar software localmente. Debe estar a salvo de que se sobrescriba cuando se actualice el software del sistema. Puede usarse para programas y datos que se pueden compartir entre un grupo de hosts, pero que no se encuentran en /usr.

El software instalado localmente debe colocarse dentro de /usr/local en vez de /usr a menos que se esté instalando para reemplazar o actualizar el software en /usr.⁸

4.9.2. Requisitos

Los siguientes directorios, o enlaces simbólicos a directorios, deben estar en /usr/local

Directorio	Descripción
bin	Binarios locales
etc	Configuración del sistema específico del host para binarios locales
games	Binarios de juegos locales
include	Archivos de encabezado C locales
lib	Bibliotecas locales
man	Manuales en línea locales
sbin	Binarios del sistema local
share	Jerarquía independiente de la arquitectura local
src	Código fuente local

Ningún otro directorio, excepto los que se enumeran a continuación, pueden estar en /usr/local después de instalar por primera vez un sistema compatible con FHS.

4.9.3. Opciones específicas

Si los directorios /lib<qual> o /usr/lib<qual> existen, los directorios equivalentes también deben existir en /usr/local.

/usr/local/etc puede ser un enlace simbólico a /etc/local.

⁷ El caso donde /usr/lib y /usr/lib<qual> son los mismos (uno es un enlace simbólico al otro) estos archivos y los subdirectorios por aplicación existirán.

⁸ Software colocado en / o /usr las actualizaciones del sistema pueden sobrescribirlos (aunque recomendamos que las distribuciones no sobrescriban los datos en /etc bajo estas circunstancias). Por esta razón, el software local no debe colocarse fuera de /usr/local sin una buena razón.

Razón Fundamental

La consistencia de `/usr/local/etc` es beneficioso para los instaladores y ya se utiliza en otros sistemas. Como todos `/usr/local` necesita una copia de seguridad para reproducir un sistema, no introduce gastos de mantenimiento adicionales, sino un enlace simbólico a `/etc/local` es adecuado si los sistemas quieren toda su configuración bajo una jerarquía.

Ten en cuenta que `/usr/etc` todavía no está permitido: programas en `/usr` deben colocar los archivos de configuración en `/etc`.

Si el directorio `/usr/share/color` existe como se especifica en este documento, entonces el directorio `/usr/local/share/color` también debe existir, regido por las mismas reglas que `/usr/share/color`.

Razón Fundamental

Este uso permite al administrador del sistema un lugar para instalar perfiles de color manualmente cuando sea necesario.

4.9.4. `/usr/local/share` : Jerarquía independiente de la arquitectura local

Los requisitos para el contenido de este directorio son los mismos que para `/usr/share`.

4.10. `/usr/sbin` : Binarios del sistema estándar no esenciales.

4.10.1. Propósito

Este directorio contiene los archivos binarios no esenciales utilizados exclusivamente por el administrador del sistema. Programas de administración del sistema necesarios para la reparación del sistema, la recuperación del sistema, el montaje/`usr`, u otras funciones esenciales deben colocarse en `/sbin` en lugar de.⁹

4.10.2. Requisitos

No debe haber subdirectorios en `/usr/sbin`.

4.11. `/usr/share`: datos independientes de la arquitectura

4.11.1. Propósito

La jerarquía `/usr/share` es para todos los archivos de datos independientes de la arquitectura de solo lectura.

⁹ Los programas de administración del sistema instalados localmente deben colocarse en `/usr/local/sbin`.

¹⁰ Gran parte de estos datos vivían originalmente en `/usr(man,doc)` o `/usr/lib(dict,terminfo,zoneinfo)`.

Se pretende que esta jerarquía se pueda compartir entre todas las plataformas de arquitectura de un sistema operativo determinado; así, por ejemplo, un sitio con plataformas i386, Alpha y PPC podría mantener un solo directorio `/usr/share` que está montado de forma centralizada. Sin embargo, tenga en cuenta que `/usr/share` generalmente no está destinado a ser compartido por diferentes sistemas operativos o por diferentes versiones del mismo sistema operativo.

Cualquier programa o paquete que contenga o requiera datos que no necesiten ser modificados debe almacenar esos datos en `/usr/share` (o `/usr/local/share`, si está instalado localmente). Se recomienda que se utilice un subdirectorio en `/usr/share` para este propósito. Las aplicaciones que usan un solo archivo pueden usar `/usr/share/misc`.

Datos del juego almacenados en `/usr/share/games` deben ser datos puramente estáticos. Cualquier archivo modificable, como archivos de puntuación, registros de juego, etc, debe colocarse en `/var/games`.

4.11.2. Requisitos

Los siguientes directorios, o enlaces simbólicos a directorios, deben estar en `/usr/share`

Directorio	Descripción
<code>man</code>	Manuales en línea
<code>misc</code>	Varios datos independientes de la arquitectura

4.11.3. Opciones específicas

Los siguientes directorios, o enlaces simbólicos a directorios, deben estar en `/usr/share`, si el subsistema correspondiente está instalado:

Directorio	Descripción
<code>color</code>	Información de gestión del color (opcional)
<code>dict</code>	Listas de palabras (opcional)
<code>Doc</code>	Documentación miscelánea (opcional)
<code>games</code>	Archivos de datos estáticos para <code>/usr/games</code> (opcional)
<code>info</code>	Directorio principal para el sistema de información GNU (opcional)
<code>locale</code>	Información de configuración regional (opcional)
<code>nls</code>	Catálogos de mensajes para compatibilidad con idiomas nativos (opcional)
<code>ppd</code>	Definiciones de impresora (opcional)
<code>sgml</code>	Datos SGML (opcional)
<code>terminfo</code>	Directorios para la base de datos terminfo (opcional)
<code>tmac</code>	macros de troff no distribuidas con groff (opcional)
<code>xml</code>	Datos XML (opcional)
<code>zoneinfo</code>	Información y configuración de la zona horaria (opcional)

Se recomienda que los directorios independientes de la arquitectura y específicos de la aplicación se coloquen aquí. Dichos directorios incluyen **groff**, **perl**, **ghostscript**, **texmf**, y **kbd** (Linux) o **syscons** (BSD). Sin embargo, pueden colocarse en `/usr/lib` para compatibilidad con versiones anteriores, a discreción del distribuidor. Del mismo modo, una jerarquía `/usr/lib/games` puede ser usada junto con una jerarquía `/usr/share/games` si el distribuidor desea colocar algunos datos de juego allí.

4.11.4. /usr/share/color : Información de gestión del color (opcional)

4.11.4.1. Propósito

Este directorio es el hogar de los archivos de gestión del color ICC instalados por el sistema.

4.11.4.2. Opciones específicas

Los siguientes directorios deben estar en /usr/share/color, si el subsistema correspondiente está instalado:

Directorio	Descripción
icc	Perfiles de color ICC (opcional)

El directorio de nivel superior /usr/share/color no debe contener ningún archivo; todos los archivos deben estar en subdirectorios de /usr/share/color.

4.11.5. /usr/share/dict : Listas de palabras (opcional)

4.11.5.1. Propósito

Este directorio es el hogar de listas de palabras en el sistema; Tradicionalmente, este directorio contiene solo el archivo de words en inglés, que es utilizado por **look(1)** y varios programas de ortografía. words puede utilizar la ortografía estadounidense o británica.

Razón Fundamental

La razón por la que se encuentran listas de words es que son los únicos archivos comunes a todos los correctores ortográficos.

4.11.5.2. Opciones específicas

Los siguientes archivos, o enlaces simbólicos a archivos, deben estar en /usr/share/dict, si el subsistema correspondiente está instalado:

Archivo	Descripción
words	Lista de palabras en inglés (opcional)

Los sitios que requieren ortografía tanto estadounidense como británica pueden enlazar words a - /usr/share/dict/american-english o - /usr/share/dict/british-english.

Se pueden agregar listas de palabras para otros idiomas usando el nombre en inglés para ese idioma, por ejemplo, /usr/share/dict/french, /usr/share/dict/danish, etc. Estos deberían, si es posible, usar un juego de caracteres basado en Unicode, siendo el juego de caracteres UTF-8 la opción preferida.

Aquí se deben incluir otras listas de palabras, si están presentes.

4.11.6. /usr/share/man : Páginas de manual

4.11.6.1. Propósito

Esta sección detalla la organización de las páginas manual en todo el sistema, incluyendo /usr/share/man.

Consulte también la sección sobre /var/cache/man.

El primario del sistema <mandir> es /usr/share/man. /usr/share/man contiene información manual para comandos y datos en / y en sistemas de archivo /usr.¹¹

Las páginas de manual se almacenan en <mandir>/<locale>/man<section>/<arch>. Una explicación de <mandir>, <locale>, <section>, y <arch> se da a continuación.

A continuación, se incluye una descripción de cada sección:

- **man1:** Páginas. Manual de programas de usuario que describen comandos de acceso público se encuentran en este capítulo. La mayor parte de la documentación del programa que un usuario necesitará utilizar se encuentra aquí.
- **man2:** Llamadas al Sistema. Esta sección describe todas las llamadas al sistema (solicitudes para que el kernel realice operaciones)
- **man3:** Funciones y subrutinas de la biblioteca. La Sección 3 describe las rutinas de la biblioteca de programas que no son llamadas directas a los servicios del kernel. Esto y el capítulo 2 solo interesan realmente a los programadores
- **man4:** Archivos especiales. La Sección 4 describe los archivos especiales, las funciones del controlador relacionadas y el soporte de red disponible en el sistema. Normalmente, esto incluye los archivos de dispositivo que se encuentran en /dev y la interfaz del kernel para el soporte del protocolo de red
- **man5:** Formatos de archivo. Los formatos de muchos archivos de datos se documentan en la sección 5. Esto incluye varios archivos de inclusión, archivos de salida del programa y archivos del sistema
- **man6:** Juegos. Este capítulo documenta juegos, demostraciones y, en general, programas triviales. Diferentes personas tienen varias nociones sobre cuán esencial es esto
- **man7:** Miscelánea. Las páginas Manual que son difíciles de clasificar se designan como sección 7. Aquí se encuentran troff y otros paquetes de macros de procesamiento de texto
- **man8:** Administración del sistema. Los programas utilizados por los administradores del sistema para la operación y el mantenimiento del sistema se documentan aquí. Algunos de estos programas también son útiles ocasionalmente para usuarios normales

4.11.6.2. Opciones específicas

Los siguientes directorios, o enlaces simbólicos a directorios, deben estar en /usr/share/<mandir>/<locale>, a menos que estén vacíos:¹²

Directorio	Descripción
man1	Programas de usuario (opcional)
man2	Llamadas al sistema (opcional)
man3	Llamadas a la biblioteca (opcional)
man4	Archivos especiales (opcional)
man5	Formatos de archivo (opcional)
man6	Juegos (opcional)
man7	Diversas (opcional)
man8	Administración del sistema (opcional)

El componente <section> describe la sección del manual.

¹¹ Obviamente, no hay páginas manual en / porque no se requieren en el momento del arranque ni en emergencias.

¹² Por ejemplo, si /usr/share/man no tiene páginas manual en la sección 4 (Dispositivos), entonces /usr/share/man/man4 puede omitirse.

Deben tomarse provisiones en la estructura de `/usr/share/man` para apoyar páginas de manual que están escritas en diferentes (o múltiples) idiomas. Estas provisiones deben tener en cuenta el almacenamiento y la referencia de estas páginas manual. Los factores relevantes incluyen el idioma (incluidas las diferencias geográficas) y el conjunto de códigos de caracteres.

Este nombramiento de subdirectorios de idiomas de `/usr/share/man` se basa en el Apéndice E del estándar POSIX 1003.1 que describe la cadena de identificación local, el método más aceptado para describir un entorno cultural. La cadena `<locale>` es:

`<language>[_<territory>][.<character-set>][,<version>]`

El campo `<language>` debe tomarse de ISO 639 (un código para la representación de nombres de idiomas). Debe tener dos caracteres de ancho y especificarse solo con letras minúsculas.

El campo `<territory>` debe ser el código de dos letras de ISO 3166 (una especificación de representaciones de países), si es posible. (La mayoría de las personas están familiarizadas con los códigos de dos letras que se utilizan para los códigos de país en las direcciones de correo electrónico). Debe tener dos caracteres de ancho y especificarse solo con letras mayúsculas.¹³

El campo `<character-set>` debe representar el estándar que describe el juego de caracteres. Si el campo `<character-set>` es solo una especificación numérica, el número representa el número del estándar internacional que describe el juego de caracteres. Se recomienda que sea una representación numérica si es posible (normas ISO, especialmente), que no incluya símbolos de puntuación adicionales y que las letras estén en minúsculas.

Un parámetro que especifica una `<version>` del perfil se puede colocar después del campo `<character-set>`, delimitado por una coma. Esto puede usarse para discriminar entre diferentes necesidades culturales; por ejemplo, el orden del diccionario frente a un orden de clasificación más orientado a los sistemas. Esta norma recomienda no utilizar el campo `<version>`, a menos que sea necesario.

Los sistemas que utilizan un idioma y un conjunto de códigos únicos para todas las páginas manual pueden omitir la subcadena `<locale>` y almacenar todas las páginas de manual en `<mandir>`. Por ejemplo, los sistemas que solo tienen páginas de manual en inglés codificadas con ASCII, pueden almacenar páginas de manual (los directorios de `<section>` `man`) directamente en `/usr/share/man`. (Esa es la circunstancia y el arreglo tradicionales, de hecho).

Los países para los que existe un conjunto de códigos de caracteres estándar bien aceptado pueden omitir el campo `<character-set>`, pero se recomienda encarecidamente que se incluya, especialmente para países con varios estándares en competencia.

Varios ejemplos:

Idioma	Territorio	Conjunto de caracteres	Directorio
inglés	-	ASCII	<code>/usr/share/man/es</code>
inglés	Reino Unido	Unicode UTF-8	<code>/usr/share/man/ en_GB.10646</code>
inglés	Estados Unidos	ASCII	<code>/usr/share/man/en_US</code>
francés	Canadá	ISO 8859-1	<code>/usr/share/man/ fr_CA.88591</code>
francés	Francia	ISO 8859-1	<code>/usr/share/man/ fr_FR.88591</code>
alemán	Alemania	ISO 646	<code>/usr/share/man/ de_DE.646</code>

¹³ Una excepción importante a esta regla es el Reino Unido, que es "GB" en la norma ISO 3166, pero "RU" o "UK" para la mayoría de las direcciones de correo electrónico.

Idioma	Territorio	Conjunto de caracteres	Directorio
Alemán	Alemania	ISO 639-1	/usr/share/man/de_DE.6391
Alemán	Alemania	ISO 8859-1	/usr/share/man/de_DE.88591
Alemán	Suiza	ISO 646	/usr/share/man/de_CH.646
Japonés	Japón	JIS	/usr/share/man/ja_JP.jis
Japonés	Japón	SJIS	/usr/share/man/ja_JP.sjis
Japonés	Japón	UJIS (o EUC-J)	/usr/share/man/ja_JP.ujis
Japonés	Japón	Unicode UTF-16	/usr/share/man/ja_JP.10646

Del mismo modo, se deben prever las páginas de manual que dependen de la arquitectura, como la documentación sobre los controladores de dispositivos o los comandos de administración del sistema de bajo nivel. Estos deben colocarse debajo de un directorio `<arch>` en el directorio `man <section>` apropiado; por ejemplo, una página manual para el comando `i386 ctrlaltdel (8)` podría colocarse en `/usr/share/man/<locale>/man8/i386/ctrlaltdel.8`.

Páginas de manual para comandos y datos en `/usr/local` se almacenan en `/usr/local/man` o `/usr/local/share/man`. Todas las jerarquías de páginas manual del sistema deben tener la misma estructura que `/usr/share/man`, ya que esta estructura es esperada por comandos que consumen contenido de página manual.¹⁴

Las secciones de las páginas `cat (<section> cat)` que contienen entradas de página manual formateadas también se encuentran dentro de los subdirectorios de `<mandir>/<locale>`, pero no son obligatorios ni pueden distribuirse en lugar de las páginas manual fuente `nroff`.

Las secciones numeradas del "1" al "8" se definen tradicionalmente. En general, el nombre de archivo de las páginas de manual, ubicadas dentro de una sección en particular terminan con `<section>`.

Además, algunos conjuntos grandes de páginas manual específicas de la aplicación tienen un sufijo adicional adjunto al nombre de archivo de la página manual. Por ejemplo, las páginas de manual del sistema de manejo de correo MH deben tener `mh` adjunto a todos los manuales de MH. Todas las páginas de manual del sistema X Window deben tener una `X` agregado al nombre del archivo.

La práctica de colocar páginas de manual en varios idiomas en subdirectorios apropiados de `/usr/share/man` también se aplica a las otras jerarquías de páginas manual, como `/usr/local/man`. (Esta parte del estándar también se aplica más adelante en la sección sobre la estructura opcional `/var/cache/man`)

4.11.7. /usr/share/misc : Datos independientes de arquitectura miscelánea

Este directorio contiene varios archivos independientes de la arquitectura que no requieren un subdirectorio separado en `/usr/share`.

4.11.7.1. Opciones específicas

Los siguientes archivos, o enlaces simbólicos a archivos, deben estar en `/usr/share/misc`, si el subsistema correspondiente está instalado:

¹⁴ `/usr/local/man` está en desuso y puede descartarse en una versión futura de esta especificación.

Archivo	Descripción
ascii	Tabla de juego de caracteres ASCII (opcional)
termcap	Base de datos de capacidad de terminal (opcional)
termcap.db	Base de datos de capacidad de terminal (opcional)

Otros archivos (específicos de la aplicación) pueden aparecer aquí, pero un distribuidor puede colocarlos en `/usr/lib` a su discreción.^{15 16}

4.11.8. `/usr/share/ppd` : Definiciones de impresora (opcional)

4.11.8.1. Propósito

`/usr/share/ppd` contiene archivos de definición de impresora PostScript (PPD), que muchos sistemas de impresión utilizan como descripciones de controladores de impresora. Los archivos PPD se pueden colocar en este directorio o en un subdirectorio.

4.11.9. `/usr/share/sgml`: datos SGML (opcional)

4.11.9.1. Propósito

`/usr/share/sgml` contiene archivos independientes de la arquitectura utilizados por las aplicaciones SGML, como catálogos ordinarios (no los centralizados, consulte `/etc/sgml`), DTD, entidades u hojas de estilo.

4.11.9.2. Opciones específicas

Los siguientes directorios, o enlaces simbólicos a directorios, deben estar en `/usr/share/sgml`, si el subsistema correspondiente está instalado:

Directorio	Descripción
docbook	docbook DTD (opcional)
tei	tei DTD (opcional)
html	html DTD (opcional)
mathml	mathml DTD (opcional)

Otros archivos que no son específicos de una DTD determinada pueden residir en su propio subdirectorio.

4.11.10. `/usr/share/xml` : datos XML (opcional)

4.11.10.1. Propósito

`/usr/share/xml` contiene archivos independientes de la arquitectura utilizados por aplicaciones XML, como catálogos ordinarios (no los centralizados, consulte `/etc/sgml`), DTD, entidades u hojas de estilo.

¹⁵ Algunos de estos archivos incluyen: `aeropuerto`, `ficha de nacimiento`, `eqnchar`, `getopt`, `gprof.callg`, `gprof.flat`, `inter.phone`, `ipfw.samp.filters`, `ipfw.samp.scripts`, `keycap.pcvr`, `mail.help`, `mail.tildehelp`, `man.template`, `map3270`, `mdoc.template`, `more.help`, `na.phone`, `nslookup.help`, `operator`, `scsi_modes`, `sendmail.hf`, `style`, `units.lib`, `vgrindefs`, `vgrindefs.db`, `zipcodes`.

¹⁶ Históricamente, el archivo `magic` se colocó en `/usr/share/misc`, pero las variantes modernas del comando de archivo usan varios archivos y los colocan en `/usr/share/file`. Por compatibilidad, la distribución puede crear un enlace simbólico en `/usr/share/misc/magic`, apuntando a `/usr/share/file/magic`.

4.11.10.2. Opciones específicas

Los siguientes directorios, o enlaces simbólicos a directorios, deben estar en `/usr/share/xml`, si el subsistema correspondiente está instalado:

Directorio	Descripción
docbook	docbook XML DTD (opcional)
xhtml	XHTML DTD (opcional)
mathml	MathML DTD (opcional)

4.12. `/usr/src` : Código fuente (opcional)

4.12.1. Propósito

El código fuente se puede colocar en este subdirectorio, solo con fines de referencia. ¹⁷

¹⁷ Generalmente, la fuente no debe construirse dentro de esta jerarquía.

Capítulo 5. La jerarquía / var

5.1. Propósito

/var contiene archivos de datos variables. Esto incluye archivos y directorios de spool, datos administrativos y de registro y archivos temporales y transitorios.

Algunas porciones de /var no se pueden compartir entre diferentes sistemas. Por ejemplo, /var/log, /var/lock, y /var/run. Se pueden compartir otras porciones, notablemente /var/mail, /var/cache/man, /var/cache/fonts, y /var/spool/news.

/var se especifica aquí para que sea posible montar /usr de solo lectura. Todo lo que una vez entró en /usr que se escribe durante el funcionamiento del sistema (a diferencia de la instalación y el mantenimiento del software) debe estar en /var.

Si /var no se puede volver una partición separada, a menudo es preferible mover /var fuera de la partición raíz y dentro de la partición /usr. (Esto a veces se hace para reducir el tamaño de la partición raíz o cuando el espacio se agota en la partición raíz). Sin embargo, /var no debe estar vinculado a /usr porque esto hace la separación de /usr y /var más difícil y es probable que cree un conflicto de nombres. En su lugar, enlace /var a /usr/var.

Las aplicaciones generalmente no deben agregar directorios al nivel superior de /var. Dichos directorios solo deben agregarse si tienen alguna implicación en todo el sistema y en consulta con la lista de correo de FHS.

5.2. Requisitos

Los siguientes directorios, o enlaces simbólicos a directorios, son obligatorios en /var:

Directorio	Descripción
cache	Datos de la caché de la aplicación
lib	Información de estado variable
local	Datos variables para /usr/local
lock	Bloquear archivos
log	Archivos de registro y directorios
opt	Datos variables para / opt
run	Datos relevantes para los procesos en ejecución
spool	Datos de la cola de aplicaciones
tmp	Archivos temporales conservados entre reinicios del sistema

Varios directorios están "reservados" en el sentido de que no deben ser utilizados arbitrariamente por una nueva aplicación, ya que entrarían en conflicto con la práctica histórica y/o local. Ellos son:

- /var/backups
- /var/cron
- /var/msgs
- /var/preserve

5.3. Opciones específicas

Los siguientes directorios, o enlaces simbólicos a directorios, deben estar en /var, si el subsistema correspondiente está instalado:

Directorio	Descripción
account	Procesar registros de contabilidad (opcional)
crash	Volcados por caída del sistema (opcional)
games	Datos de juego variables (opcional)
mail	Archivos de buzón de usuario (opcional)
yp	Archivos de base de datos del Servicio de información de red (NIS) (opcional)

5.4. /var/account : Procesar registros de contabilidad (opcional)

5.4.1. Propósito

Este directorio contiene el registro de contabilidad del proceso activo actual y los datos de uso del proceso compuesto (como se usa en algunos sistemas similares a UNIX por lastcomm y sa).

5.5. /var/cache: Datos de la caché de la aplicación

5.5.1. Propósito

/var/cache está destinado a los datos almacenados en caché de las aplicaciones. Dichos datos se generan localmente como resultado de operaciones de E/S o cálculos que requieren mucho tiempo. La aplicación debe poder regenerar o restaurar los datos. A diferencia de /var/spool, los archivos en caché se pueden eliminar sin pérdida de datos. Los datos deben permanecer válidos entre las invocaciones de la aplicación y el reinicio del sistema.

Archivos ubicados en /var/cache pueden estar vencidos de una manera específica de la aplicación, por el administrador del sistema, o ambos. La aplicación siempre debe poder recuperarse de la eliminación manual de estos archivos (generalmente debido a la escasez de espacio en el disco). No se establecen otros requisitos sobre el formato de datos de los directorios de la caché.

Razón Fundamental

La existencia de un directorio separado para los datos almacenados en caché permite a los administradores del sistema establecer diferentes políticas de disco y respaldo de otros directorios en /var.

5.5.2. Opciones específicas

Directorio	Descripción
fonts	Fuentes generadas localmente (opcional)
man	Páginas de manual con formato local (opcional)
www	Proxy WWW o datos de caché (opcional)

Directorio	Descripción
<package>	Paquete de datos de caché específicos (opcional)

5.5.3. /var/cache/fonts: Fuentes generadas localmente (opcional)

5.5.3.1. Propósito

El directorio `/var/cache/fonts` debe usarse para almacenar cualquier fuente creada dinámicamente. En particular, todas las fuentes generadas automáticamente por **mktexpk** deben estar ubicadas en subdirectorios de `/var/cache/fonts`.¹

5.5.3.2. Opciones específicas

Otras fuentes creadas dinámicamente también se pueden colocar en este árbol, bajo subdirectorios de `/var/cache/fonts`.

5.5.4. /var/cache/man : Páginas manual formateadas localmente (opcional)

5.5.4.1. Propósito

Este directorio proporciona una ubicación estándar para los sitios que proporcionan una partición `/usr`, pero desea permitir el almacenamiento en caché de páginas manual formateadas localmente. Sitios que montan `/usr` como escribible (por ejemplo, instalaciones de un solo usuario) puede optar por no usar `/var/cache/man` y puede escribir páginas de manual formateadas en los directorios de `cat<section>` en `/usr/share/man` directamente. Recomendamos que la mayoría de los sitios utilicen una de las siguientes opciones en su lugar:

- Formatee previamente todas las páginas de manual junto con las versiones sin formato.
- No permita el almacenamiento en caché de las páginas manual formateadas y requiera que se realice el formato cada vez que se abre una página manual.
- Permitir el almacenamiento en caché local de páginas manual formateadas en `/var/cache/man`.

La estructura de `/var/cache/man` debe reflejar tanto el hecho de múltiples jerarquías de páginas manual como la posibilidad de compatibilidad con múltiples idiomas.

Dada una página manual sin formato que normalmente aparece en `<path>/man/<locale>/man<section>`, el directorio para colocar las páginas de manual formateadas es `/var/cache/man/<catpath>/<locale>/cat<section>`, donde `<catpath>` se deriva de `<path>` quitando cualquier interlineado `usr` y/o arrastrando componentes de nombre de ruta `share`. (Tenga en cuenta que el componente `<locale>` Puede que falte.)²

Páginas man escritas en `/var/cache/man` eventualmente pueden ser transferidas a los directorios caducados o a los preformateados apropiados en la jerarquía fuente `man`; páginas manual igualmente formateadas en la jerarquía fuente `man`, pueden expirar si no se accede a ellas durante un período de tiempo.

¹ Este estándar no incorpora actualmente la Estructura de directorios de TeX (un documento que describe el diseño de los archivos y directorios de TeX), pero puede ser útil para leer. Se encuentra en <http://ctan.tug.org/tex/>

² Por ejemplo, `/usr/share/man/man1/lis.1` está formateado en `/var/cache/man/cat1/lis.1`, y `/usr/X11R6/man/<locale>/man3/XtClass.3x` en `/var/cache/man/X11R6/<locale>/cat3/XtClass.3x`

Si las páginas de manual preformateadas vienen con un sistema en un medio de solo lectura (un CD-ROM, por ejemplo), deben instalarse en la jerarquía fuente `man` (por ejemplo. `/usr/share/man/cat <sección>`). `/var/cache/man` está reservada como una caché grabable para páginas manual formateadas.

Razón fundamental

Liberación 1.2 de esta norma especificada `/var/catman` para esta jerarquía. La ruta se ha movido debajo de `/var/cache` para reflejar mejor la naturaleza dinámica de las páginas manual formateadas. El nombre del directorio se ha cambiado a `man` permitiendo mejorar la jerarquía para incluir formatos posprocesados distintos de "cat", como PostScript, HTML o DVI.

5.6. `/var/crash` : Volcados de memoria del sistema (opcional)

5.6.1. Propósito

Este directorio contiene volcados de memoria del sistema. A la fecha de esta publicación del estándar, los volcados del sistema no eran compatibles con Linux, pero pueden ser compatibles con otros sistemas que pueden cumplir con la FHS.

5.7. `/var/games`: Datos de juego variables (opcional)

5.7.1. Propósito

Cualquier dato variable relacionado con juegos en `/usr` debe colocarse aquí. `/var/games` debe contener los datos variables previamente encontrados en `/usr`; Los datos estáticos, como el texto de ayuda, las descripciones de nivel, etc, deben permanecer en otro lugar, como `/usr/share/games`.

Razón fundamental

`/var/games` ha recibido una jerarquía propia, en lugar de dejarla debajo de `/var/lib` como en la versión 1.2 de esta norma. La separación permite el control local de las estrategias de copia de seguridad, los permisos y el uso del disco, así como también permite compartir entre hosts y reducir el desorden en `/var/lib`. Adicionalmente, `/var/games` es la ruta tradicionalmente utilizada por BSD.

5.8. `/var/lib` : Información de estado variable

5.8.1. Propósito

Esta jerarquía contiene información de estado perteneciente a una aplicación o al sistema. La información de estado son datos que los programas modifican mientras se ejecutan y que pertenecen a un host específico. Los usuarios nunca deberían necesitar modificar archivos en `/var/lib` para configurar el funcionamiento de un paquete, la jerarquía de archivos específica utilizada para almacenar los datos no debe exponerse a los usuarios habituales.³

La información de estado se usa generalmente para preservar la condición de una aplicación (o un grupo de aplicaciones interrelacionadas) entre invocaciones y entre diferentes instancias de la misma aplicación. La información de estado generalmente debe seguir siendo válida después de un reinicio, no debe ser un resultado de registro y no deben ser datos en cola.

Una aplicación (o un grupo de aplicaciones interrelacionadas) debe utilizar un subdirectorio de `/var/lib` por sus datos. Hay un subdirectorio obligatorio, `/var/lib/misc`, que está destinado a archivos de estado que no necesitan un subdirectorio; los otros subdirectorios solo deben estar presentes si la aplicación en cuestión está incluida en la distribución.⁴

³ Los datos con la estructura del sistema de archivos expuesta deben almacenarse en `/srv`.

`/var/lib/<nombre>` es la ubicación que se debe utilizar para todo el soporte de empaquetado de distribución. Diferentes distribuciones pueden usar diferentes nombres, por supuesto.

3

5.8.2. Requisitos

Los siguientes directorios, o enlaces simbólicos a directorios, son obligatorios en `/var/lib`:

Directorio	Descripción
<code>misc</code>	Datos de estado misceláneos

5.8.3. Opciones específicas

Los siguientes directorios, o enlaces simbólicos a directorios, deben estar en `/var/lib`, si el subsistema correspondiente está instalado:

Directorio	Descripción
<code><editor></code>	Estado y archivos de copia de seguridad del editor (opcional)
<code><pkgtool></code>	Empaquetar archivos de soporte (opcional)
<code><paquete></code>	Datos de estado para paquetes y subsistemas (opcional)
<code>color</code>	Información de gestión del color (opcional)
<code>hwclock</code>	Directorio de estado para <code>hwclock</code> (opcional)
<code>xdm</code>	X Display Manager datos variables (opcional)

5.8.4. `/var/lib/<editor>` : Estado y archivos de copia de seguridad del editor (opcional)

5.8.4.1. Propósito

Estos directorios contienen archivos guardados generados por cualquier terminación inesperada de un editor (por ejemplo, **elvis**, **Júpter**, **nvi**).

Es posible que otros editores no requieran un directorio para los archivos de recuperación de fallos, pero pueden requerir un lugar bien definido para almacenar otra información mientras se ejecuta el editor. Esta información debe almacenarse en un subdirectorio en `/var/lib` (por ejemplo, GNU Emacs colocaría archivos de bloqueo en `/var/lib/emacs/lock`).

Los editores en el futuro pueden requerir información de estado adicional más allá de los archivos de recuperación de fallos y los archivos de bloqueo; esta información también debe colocarse en `/var/lib/<editor>`.

Razón fundamental

Las versiones anteriores de Linux, así como todos los proveedores comerciales, utilizan `/var/preserve` para `vi` o sus clones. Sin embargo, cada editor usa su propio formato para estos archivos de recuperación de fallos, por lo que se necesita un directorio separado para cada editor.

4 Una diferencia importante entre esta versión de este estándar y las anteriores es que ahora se requiere que las aplicaciones utilicen un subdirectorio de `/var/lib`.

Los archivos de bloqueo específicos del editor suelen ser bastante diferentes del dispositivo o de los archivos de bloqueo de recursos que se almacenan en `/var/lock` y por tanto, se almacenan en `/var/lib`.

5.8.5. `/var/lib/color` : Información de gestión del color (opcional)

5.8.5.1. Propósito

Este directorio es el hogar de los archivos de gestión del color ICC instalados dinámicamente. Este directorio se distribuirá utilizando las mismas reglas que el directorio `/usr/share/color`.

5.8.6. `/var/lib/hwclock` : Directorio de estado para hwclock (opcional)

5.8.6.1. Propósito

Este directorio contiene el archivo `/var/lib/hwclock/adjtime`.

Razón fundamental

En FHS 2.1, este archivo era `/etc/adjtime`, pero como **hwclock** lo actualiza, eso obviamente era incorrecto.

5.8.7. `/var/lib/misc` : Datos variables misceláneos

5.8.7.1. Propósito

Este directorio contiene datos variables que no se encuentran en un subdirectorio en `/var/lib`. Se debe intentar utilizar nombres relativamente únicos en este directorio para evitar conflictos de espacio de nombres.⁵

5.9. `/var/lock` : Archivos de bloqueo

5.9.1. Propósito

Los archivos de bloqueo deben almacenarse dentro de la estructura de directorios `/var/lock`.

Los archivos de bloqueo para dispositivos y otros recursos compartidos por varias aplicaciones, como los archivos de bloqueo de dispositivos en serie que se encontraron originalmente en `/usr/spool/locks` o `/usr/spool/uucp`, ahora deben almacenarse en `/var/lock`. La convención de nomenclatura que debe utilizarse es "LCK .." seguida del nombre base del dispositivo. Por ejemplo, para bloquear `/dev/ttyS0` se crearía el archivo "LCK..ttyS0".⁶

El formato utilizado para el contenido de dichos archivos de bloqueo debe ser el formato de archivo de bloqueo HDB UUCP. El formato HDB es para almacenar el identificador de proceso (PID) como un número decimal ASCII de diez bytes, con una nueva línea al final. Por ejemplo, si el proceso 1230 contiene un archivo de bloqueo, contendrá los once caracteres: espacio, espacio, espacio, espacio, espacio, espacio, espacio, uno, dos, tres, cero y una nueva línea.

⁵ Esta jerarquía debe contener archivos almacenados en `/var/db` en las versiones actuales de BSD. Éstas incluyen localizar base de datos y mountdtab, y la (s) base de datos (s) de símbolos del núcleo.

⁶ Entonces, cualquier cosa que desee usar `/dev/ttyS0` puede leer el archivo de bloqueo y actuar en consecuencia (todos los bloqueos en `/var/lock` deben ser legibles en todo el mundo).

5.10. /var/log : Archivos de registro y directorios

5.10.1. Propósito

Este directorio contiene varios archivos de registro. La mayoría de los registros se deben escribir en este directorio o en un subdirectorio apropiado.

5.10.2. Opciones específicas

Los siguientes archivos, o enlaces simbólicos a archivos, deben estar en /var/log, si el subsistema correspondiente está instalado:

Archivo	Descripción
lastlog	registro del último inicio de sesión de cada usuario
messages	mensajes del sistema de syslogd
wtmp	registro de todos los inicios y cierres de sesión

5.11. /var/mail : Archivos de buzón de usuario (opcional)

5.11.1. Propósito

El spool de correo debe ser accesible a través de /var/mail y los archivos de cola de correo deben tener la forma <nombre de usuario>. ⁷

Los archivos de buzón de correo del usuario en esta ubicación deben almacenarse en el formato de buzón de correo estándar de UNIX.

Razón fundamental

La ubicación lógica de este directorio se cambió de /var/spool/mail para poner FHS en línea con casi todas las distribuciones de UNIX. Este cambio es importante para la interoperabilidad ya que un solo /var/mail a menudo se comparte entre varios hosts y varias distribuciones de UNIX (a pesar de los problemas de bloqueo de NFS).

Es importante tener en cuenta que no es necesario mover físicamente la bobina de correo a esta ubicación. Sin embargo, los programas y los archivos de encabezado deben cambiarse para usar /var/mail.

5.12. /var/opt : Datos variables para /opt

5.12.1. Propósito

Los datos variables de los paquetes en /opt deben instalarse en /var/opt/<subdir>, donde <subdir> es el nombre del subárbol en /opt donde se almacenan los datos estáticos de un paquete de software complementario, excepto donde sea reemplazado por otro archivo en /etc. No se impone ninguna estructura a la disposición interna de /var/opt/<subdir>.

Razón fundamental

Consulte la justificación de /opt.

⁷ Tenga en cuenta que /var/mail puede ser un enlace simbólico a otro directorio.

5.13. /var/run : Datos variables en tiempo de ejecución

5.13.1. Propósito

Este directorio alguna vez estuvo destinado a los datos de información del sistema que describen el sistema desde que se inició. Estas funciones se han movido a /run; este directorio existe para garantizar la compatibilidad con los sistemas y el software que utilizan una versión anterior de esta especificación.

5.13.2. Requisitos

En general, los requisitos para /run también se aplicará a /var/run. Es válido implementar /var/run como enlace simbólico a /run.

El utmp El archivo, que almacena información sobre quién está utilizando actualmente el sistema, se encuentra en este directorio.

Los programas no deben acceder a ambos /var/run y /run directamente, excepto para acceder /var/run/utmp.⁸

5.14. / var/spool : Datos de bobina de aplicaciones

5.14.1. Propósito

/var/spool contiene datos que están esperando algún tipo de procesamiento posterior. Los datos en /var/spool representan el trabajo a realizar en el futuro (por un programa, usuario o administrador); a menudo, los datos se eliminan una vez procesados.⁹

5.14.2. Opciones específicas

Los siguientes directorios, o enlaces simbólicos a directorios, deben estar en /var/spool, si el subsistema correspondiente está instalado:

Directorio	Descripción
lpd	Directorio bobina de impresión (opcional)
mqueue	Cola de correo saliente (opcional)
news	Directorio bobina de noticias (opcional)
rwho	Archivos Rwhod (opcional)
uucp	Directorio bobina para UUCP (opcional)

5.14.3. /var/spool/lpd : Colas de impresión daemon de impresora de línea (opcional)

5.14.3.1. Propósito

El archivo de bloqueo para **lpd**, **lpd.lock**, debe colocarse en /var/spool/lpd. Se sugiere que el archivo de bloqueo de cada impresora se coloque en el directorio bobina de esa impresora específica y se denomine lock.

⁸ Esto es para evitar confusiones sobre la ubicación de los archivos transitorios. En general, un programa debe usar /var/run o /run para acceder a estos archivos, no a ambos.

⁹ Los archivos de bloqueo UUCP deben colocarse en /var/lock. Consulte la sección anterior sobre /var/lock.

5.14.3.2. Opciones específicas

Directorio	Descripción
impresora	Carretes para una impresora específica (opcional)

5.14.4. /var/spool/rwho : Archivos Rwho (opcional)

5.14.4.1. Propósito

Este directorio contiene el rwho información para otros sistemas en la red local.

Razón fundamental

Algunas versiones de BSD usan /var / rwho para estos datos; dada su ubicación histórica en /var / spool en otros sistemas y su ajuste aproximado a la definición de datos "en cola", esta ubicación se consideró más apropiada.

5.15. /var/tmp: Archivos temporales conservados entre reinicios del sistema

5.15.1. Propósito

El /var / tmp El directorio está disponible para programas que requieren archivos temporales o directorios que se conservan entre reinicios del sistema. Por lo tanto, los datos almacenados en /var / tmp es más persistente que los datos en /tmp.

Archivos y directorios ubicados en /var / tmp no debe eliminarse cuando se inicia el sistema. Aunque los datos almacenados en / var / tmp generalmente se elimina de una manera específica del sitio, se recomienda que las eliminaciones se produzcan en un intervalo menos frecuente que /tmp.

5.16. /var/yp : Archivos de base de datos del Servicio de información de red (NIS) (opcional)

5.16.1. Propósito

Los datos variables para el Servicio de información de red (NIS), anteriormente conocido como Sun Yellow Pages (YP), deben colocarse en este directorio.

Razón fundamental

/var/yp es el directorio estándar para datos NIS (YP) y se utiliza casi exclusivamente en la documentación y los sistemas NIS.¹⁰

¹⁰ NIS no debe confundirse con Sun NIS +, que usa un directorio diferente, /var/nis.

Capítulo 6. Anexo específico del sistema operativo

Esta sección es para requisitos y recomendaciones adicionales que solo se aplican a un sistema operativo específico. El material de esta sección nunca debe entrar en conflicto con el estándar básico.

6.1. Linux

Este es el anexo del sistema operativo Linux.

6.1.1. / : Directorio raíz

En sistemas Linux, si el kernel está ubicado en /, recomendamos usar los nombres vmlinux o vmlinuz, que se han utilizado en los paquetes fuente del kernel de Linux recientes.

6.1.2. /bin : Binarios de comandos de usuario esenciales (para uso de todos los usuarios)

Los sistemas Linux que los requieren colocan estos archivos adicionales en /bin:

- setserial

6.1.3. /dev : Dispositivos y archivos especiales

Los siguientes dispositivos deben existir en /dev.

/dev/null	Todos los datos escritos en este dispositivo se descartan. Una lectura de este dispositivo devolverá una condición EOF.
/dev/zero	Este dispositivo ofrece una fuente de datos a cero. Todos los datos escritos en este dispositivo se descartan. Una lectura de este dispositivo devolverá tantos bytes con el valor cero como se solicitó en su momento.
/dev/tty	Este dispositivo es sinónimo de terminal de control de un proceso. Una vez que se abre este dispositivo, todas las lecturas y escrituras se comportarán como si se hubiera abierto el dispositivo terminal de control real.

Razón fundamental

Las versiones anteriores de la FHS tenían requisitos más estrictos para /dev. También pueden existir otros dispositivos en /dev. Los nombres de dispositivos pueden existir como enlaces simbólicos a otros nodos de dispositivos ubicados en /dev o subdirectorios de /dev. No hay ningún requisito con respecto a los valores numéricos mayores/menores.

6.1.4. /etc : Configuración del sistema específico del host

Los sistemas Linux que los requieren colocan estos archivos adicionales en /etc.

- lilo.conf

6.1.5. /proc : Sistema de archivos virtual de información de proceso y kernel

El sistema de archivos `proc` es el método estándar de-facto de Linux para manejar la información del proceso y del sistema, en lugar de `/dev/kmem` y otros métodos similares. Recomendamos encarecidamente esto para el almacenamiento y la recuperación de información de procesos, así como otra información del kernel y de la memoria.

6.1.6. /sbin : Binarios esenciales del sistema

Los sistemas Linux colocan comandos relacionados con el mantenimiento del sistema de archivos y la administración del cargador de arranque en `/sbin`.

Archivos opcionales para `/sbin`

- Binarios estáticos:

- **ldconfig**
- **sln**
- **ssync**

ln estático (**sln**) y **sync** (**ssync**) estático son útiles cuando las cosas van mal. El uso principal de **sln** (para reparar enlaces simbólicos incorrectos en `/lib` después de una actualización mal orquestada) ya no es una preocupación importante ahora que el programa **ldconfig** (generalmente ubicado en `/usr/sbin`) existe y puede actuar como guía en la actualización de las bibliotecas dinámicas. **Sync** estático es útil en algunas situaciones de emergencia. Tenga en cuenta que no es necesario que sean versiones vinculadas estáticamente del estándar **ln** y de **sync**, pero tal vez.

El binario **ldconfig** es opcional para `/sbin` ya que un sitio puede optar por ejecutar **ldconfig** en el momento del arranque, en lugar de solo al actualizar las bibliotecas compartidas. (No está claro si es ventajoso ejecutar **ldconfig** en cada arranque) Aun así, a algunas personas les gusta **ldconfig** alrededor para la siguiente situación (demasiado común):

1. Acabo de eliminar `/lib/<archivo>`.
2. No puedo encontrar el nombre de la biblioteca porque **ls** está vinculado dinámicamente, estoy usando un shell que no tiene **ls** integrado, y no sé cómo usar "**echo ***" como un reemplazo.
3. Tengo una **sln** estática, pero no sé cómo llamar al enlace.

- Varios:

- **ctrlaltdel**
- **kbdrate**

Para hacer frente al hecho de que algunos teclados tienen una tasa de repetición tan alta que no se pueden utilizar, **kbdrate** se puede instalar en `/sbin` en algunos sistemas.

Dado que la acción predeterminada en el kernel para la combinación de teclas Ctrl-Alt-Supr es un reinicio completo instantáneo, generalmente es recomendable deshabilitar el comportamiento antes de montar el sistema de archivos raíz en modo lectura-escritura. Algunos suites **init** pueden deshabilitar Ctrl-Alt-Del, pero otras pueden requerir el programa **ctrlaltdel**, que se puede instalar en `/sbin` en esos sistemas.

6.1.7. /sys : Sistema de archivos virtual de información del sistema y del kernel

El sistema de archivos `sys` es la ubicación donde se expone la información sobre dispositivos, controladores y algunas características del kernel. Su estructura subyacente está determinada por el kernel de Linux particular que se esté utilizando en este momento y, por lo demás, no se especifica.

6.1.8. /usr/include : Archivos de encabezado incluidos por programas C

Estos enlaces simbólicos son necesarios si se instala un compilador C o C++ y solo para sistemas no basados en glibc.

```
/usr/include/asm -> /usr/src/linux/include/asm-<arch>  
/usr/include/linux -> /usr/src/linux/include/linux
```

6.1.9. /usr/src: Código fuente

Para sistemas basados en glibc, no hay pautas específicas para este directorio. Para sistemas basados en Linux las correcciones libc antes de glibc, se aplican las siguientes pautas y fundamentos:

El único código fuente que debe colocarse en una ubicación específica es el código fuente del kernel de Linux. Está localizado en `/usr/src/linux`.

Si se instala un compilador C o C++, pero el código fuente completo del kernel de Linux no está instalado, entonces los archivos de inclusión del código fuente del kernel deben ubicarse en estos directorios:

```
/usr/src/linux/include/asm- <arch>  
/usr/src/linux/include/linux
```

<arch> es el nombre de la arquitectura del sistema.

Nota

`/usr/src/linux` puede ser un enlace simbólico a un árbol de código fuente del kernel.

Razón fundamental

Es importante que los archivos de inclusión del kernel estén ubicados en `/usr/src/linux` y no en `/usr/` incluir por lo que no hay problemas cuando los administradores del sistema actualizan su versión del kernel por primera vez.

6.1.10. /var/spool/cron : Trabajos cron y at

Este directorio contiene los datos variables para los programas `cron` y `at`.

Capítulo 7. Apéndice

7.1. La lista de correo de FHS

La lista de correo de FHS se encuentra en <fhs-discuss@lists.linuxfoundation.org> (suscripción requerida como medida de limitación de spam). La información de suscripción a la lista de correo, los archivos, etc. se encuentran en <https://lists.linux-foundation.org/mailman/listinfo/fhs-discuss> [<https://lists.linux-foundation.org/mailman/listinfo/fhs-discuss>]

7.2. Antecedentes de la FHS

El proceso de desarrollo de una jerarquía de sistema de archivos estándar comenzó en agosto de 1993 con un esfuerzo por reestructurar la estructura de archivos y directorios de Linux. El FSSTND, un estándar de jerarquía del sistema de archivos específico del sistema operativo Linux, se publicó el 14 de febrero de 1994. Las revisiones posteriores se publicaron el 9 de octubre de 1994 y el 28 de marzo de 1995.

A principios de 1995, se adoptó el objetivo de desarrollar una versión más completa de FSSTND para abordar no solo Linux, sino otros sistemas similares a UNIX con la ayuda de miembros de la comunidad de desarrollo BSD. Como resultado, se hizo un esfuerzo concertado para centrarse en cuestiones que eran generales en los sistemas similares a UNIX. En reconocimiento de esta ampliación del alcance, el nombre del estándar se cambió a Estándar de jerarquía del sistema de archivos o FHS para abreviar.

Los voluntarios que han contribuido ampliamente a este estándar se enumeran al final de este documento. Esta norma representa una opinión de consenso de esos y otros contribuyentes.

Gracias a Network Operations en la Universidad de California en San Diego, y más tarde a SourceForge, quienes nos permitieron usar sus excelentes servidores de listas de correo durante las primeras fases de desarrollo.

7.3. Reglas generales

Estas son algunas de las pautas que se han utilizado en el desarrollo de este estándar:

- Resolver problemas técnicos limitando las dificultades de transición.
- Hacer que la especificación sea razonablemente estable.
- Obtener la aprobación de los distribuidores, desarrolladores y otros tomadores de decisiones en los grupos de desarrollo relevantes y fomentar su participación.
- Proporcionar un estándar que sea atractivo para los implementadores de diferentes sistemas similares a UNIX.

7.4. Alcance

Este documento especifica una jerarquía de sistemas de archivos estándar para los sistemas de archivos FHS especificando la ubicación de los archivos y directorios, y el contenido de algunos archivos del sistema.

Este estándar ha sido diseñado para ser utilizado por integradores de sistemas, desarrolladores de paquetes y administradores de sistemas en la construcción y mantenimiento de sistemas de archivos compatibles con FHS. Su objetivo principal es ser una referencia y no un tutorial sobre cómo administrar una jerarquía de sistema de archivos conforme.

El FHS surgió de un trabajo anterior en FSSTND, un estándar de organización de sistemas de archivos para el sistema operativo Linux. Se basa en FSSTND para abordar problemas de interoperabilidad no solo en la comunidad de Linux sino en un ámbito más amplio que incluye sistemas operativos basados en 4.4BSD. Incorpora lecciones aprendidas en el mundo BSD y en otros lugares sobre el soporte de múltiples arquitecturas y las demandas de redes heterogéneas.

Aunque este estándar es más completo que los intentos anteriores de estandarización de la jerarquía del sistema de archivos, las actualizaciones periódicas pueden ser necesarias a medida que cambian los requisitos en relación con la tecnología emergente. También es posible que se descubran mejores soluciones a los problemas aquí abordados, de modo que nuestras soluciones ya no sean las mejores soluciones posibles. Es posible que se publiquen borradores complementarios además de las actualizaciones periódicas de este documento. Sin embargo, un objetivo específico es la compatibilidad con versiones anteriores de una versión de este documento a la siguiente.

Los comentarios relacionados con esta norma son bienvenidos. Cualquier comentario o sugerencia de cambios puede dirigirse a la lista de correo de FHS, o archivarse como errores, o ambos. Los comentarios tipográficos o gramaticales deben archivarse como errores. El rastreador de errores está en <http://bugs.linuxfoundation.org>; use la categoría FHS.

Antes de enviar correo a la lista de correo, se solicita que primero eche un vistazo a los archivos de la lista de correo para evitar una discusión excesiva de temas antiguos.

En ocasiones, pueden surgir preguntas sobre cómo interpretar los elementos de este documento. Si necesita una aclaración, comuníquese con la lista de correo de FHS. Dado que esta norma representa un consenso de muchos participantes, es importante asegurarse de que cualquier interpretación también represente su opinión colectiva. Por esta razón, es posible que no sea posible dar una respuesta inmediata a menos que la consulta haya sido tema de discusión previa.

7.5. Expresiones de gratitud

Los desarrolladores de FHS desean agradecer a los desarrolladores, administradores de sistemas y usuarios cuya aportación fue esencial para este estándar. Deseamos agradecer a cada uno de los colaboradores que ayudaron a escribir, compilar y componer este estándar.

El Grupo FHS también desea agradecer a los desarrolladores de Linux que apoyaron el FSSTND, el predecesor de este estándar. Si no hubieran demostrado que el FSSTND era beneficioso, el FHS nunca podría haber evolucionado.

7.6. Colaboradores

Brandon S. Allbery	John A. Martin	Mike Sangrey
Keith Bostic	Ian McCloghrie	David H. Silber
Drew Eckhardt	Chris Metcalf	Thomas Sippel-Dau
Rik Faith	Ian Murdock	Theodore Ts'o
Karl Goetz	David C. Niemi	Stephen Tweedie
Stephen Harris	Lennart Poettering	Fred N. van Kempen
Ian Jackson	Daniel Quinlan	Bernd Warken
Andreas Jaeger	Eric S. Raymond	Mats Wichmann
Jeff Licquia	Russell Oxford	Christopher Yeoh