

Estructura del Sistema de Archivos de **Linux**

Grupo de Standard de el Sistema de Archivos [**FSSTND**]

Daniel Quinlan

Traductor: Israel Barrientos *mailto:jbarrien@ccr.dsi.uanl.mx*

Maquetador Linuxdoc-SGML: Antonio Ismael Olea González,

mailto:olea@poboxes.com 2:345/108.9@fidonet.org

r1.2, abril de 1996

El proceso abierto y distribuido en el cual el sistema operativo **Linux** se ha desarrollado propicia un rápido crecimiento, tanto del sistema operativo, como de aplicaciones, y distribuciones integradas. Por tanto, existe la necesidad de la estandarización de la estructura del sistema de archivos de **Linux**. Este documento intenta especificar la localización estándar de archivos y directorios en sistemas **Linux**. Una estructura del sistema de archivos estandarizada permite a usuarios, desarrolladores, y distribuidores, el obtener componentes del sistema de varias fuentes que trabajarán juntas tan bien como si hubiesen sido desarrolladas bajo un proceso de desarrollo centralizado. Ésto también facilita la administración del sistema, así como el desarrollo de paquetes de segundas y terceras personas, y la escritura de documentación que no depende de la implementación.

Índice General

1	Advertencia legal	4
2	Prefacio	4
2.1	Estatus del Estándar	4
3	Organización del Estándar	4
3.1	Convenciones Tipográficas	5
4	General	5
4.1	Enfoque	5
4.2	Problemas específicos.	6
4.3	Objetivos	7
4.4	Historia y Progreso	7
4.5	Conformidad con este Documento	8
5	El sistema de Archivos	9
5.1	Tabla con ejemplos.	10
5.2	El Directorio Raíz	11
5.3	/bin Binarios de comandos esenciales de usuarios (disponibles para todos los usuarios).	11
5.3.1	Comandos generales:	11
5.3.2	Comandos para restauración.	12
5.3.3	Comandos de red.	12
5.4	/boot: Archivos estáticos del cargador de arranque (boot loader).	12

5.5	/dev Archivos de dispositivos.	12
5.6	/etc : Configuración del sistema local a la máquina.	13
5.6.1	Archivos generales:	13
5.6.2	Archivos de Red:	14
5.7	/home: Directorios hogar de los usuarios (opcional)	14
5.8	/lib: Librerías compartidas y módulos de kernel esenciales	14
5.9	/mnt: Punto de montaje para sistemas de archivos montados temporalmente.	15
5.10	/proc: Sistema de archivos virtual de informacion de procesos y del kernel.	15
5.11	/root: Directorio hogar de root (opcional)	15
5.12	/sbin: Binarios del Sistema (Alguna vez mantenidos en /etc)	15
5.12.1	Comandos Generales.	16
5.12.2	Comandos de Apagado.	16
5.12.3	Comandos de manejo de sistemas de archivos.	16
5.12.4	Comandos del sistema de archivos ext2 (opcional)	16
5.12.5	Comandos de Red.	16
5.12.6	Binarios estáticos. (compilados estáticamente)	16
5.12.7	Misceláneos	17
5.13	/tmp: Archivos temporales.	17
6	La jerarquía /usr	17
6.1	/usr/X11R6: El sistema X Window, Version 11 release 6	18
6.2	/usr/X386: El sistema X Window, Version 11 release 5 en plataformas X 86	18
6.2.1	/usr/bin: La mayoría de los comandos del usuario.	19
6.3	/usr/dict: Listas de palabras	19
6.4	/usr/etc: Configuración del sistema (todo el site)	19
6.5	/usr/include: Directorio para archivos include estándar.	20
6.6	/usr/lib: Librerías para programas y paquetes.	20
6.7	/usr/local: Jerarquía local	21
6.8	/usr/man : Páginas del manual.	22
6.8.1	man1: Programas de usuario.	24
6.8.2	man2: Llamadas al sistema.	24
6.8.3	man3: Subrutinas y funciones de librería.	24
6.8.4	man4: Archivos especiales.	24
6.8.5	man5: Formatos de archivos.	24
6.8.6	man6: Juegos.(Binarios Educativos)	24
6.8.7	man7: Misceláneos	24
6.8.8	man8: Administración del Sistema	24

6.8.9	man9: Funciones y variables internas del kernel	24
6.9	/usr/sbin: Binarios de sistema estándar no-esenciales.	25
6.10	/usr/share : Información Independiente de Arquitectura.	25
6.11	/usr/src : Código fuente.	25
7	La Jerarquía /var	25
7.1	/var/adm : Bitácora del sistema y archivos contables (obsoleto)	26
7.2	/var/catman : Páginas del Manual Formateadas localmente (opcional)	26
7.3	/var/lib : Información de Estado de Aplicaciones.	27
7.3.1	/var/lib/emacs	27
7.3.2	/var/lib/games	27
7.3.3	/var/lib/news	27
7.3.4	/var/lib/texmf	28
7.3.5	/var/lib/xdm	28
7.4	/var/local : Información variable del software que está en /usr/local	28
7.5	/var/lock : Archivos de Bloqueo	28
7.6	/var/log : Archivos bitácora y directorios	29
7.7	/var/named : Archivos DNS	29
7.8	/var/nis : Archivos de bases de datos del servicio de información de red (NIS)	29
7.9	/var/preserve:Archivos guardados después de una colisión o una terminación inesperada de ex ó vi	29
7.10	/var/run : Archivos variables de tiempo de ejecución	29
7.11	/var/spool : Directorios de fila de trabajos para procesamiento posterior	30
7.11.1	/var/spool/lpd	30
7.12	/var/tmp : Archivos temporales, utilizando para mantener /tmp pequeño.	30
8	Razonamientos adicionales y asuntos sin resolver	30
8.1	¿Qué es esencial?	31
8.2	Red	31
8.2.1	/bin :Cualquier cosa que algún usuario querrá utilizar y que es también considerado vital.	31
8.2.2	/sbin :Cualquier cosa que sólo root necesita y se considere vital.	31
8.2.3	/usr/bin : Algúnos binarios que algún usuario querrá utilizar y que no son vitales.	31
8.2.4	/usr/sbin : Algúnos binarios sólo para el administrador que no son vitales.	31
8.3	Estructuras independientes de la arquitectura	31
8.4	Enlaces Simbólicos.	32
8.5	Binarios Compilados Estáticamente.	32
9	La lista del correo del FSSTND	32

1 Advertencia legal

Linux no es una Marca Registrada y no tiene conexión con **UNIX**

UNIX es una marca registrada de **XOpen Company, Ltd/.**

HP-UX es una marca registrada de **Hewlett-Packard.**

Novell y **Novell NetWare** son marcas registradas de **Novell.**

SunOS Sun Microsystems, Sun NIS, Sun RPC, y NFS son marcas registradas de **Sun Microsystems.**

System V y SVR4 son marcas registradas de **AT&T.**

X Windows System es una marca registrada de el **X Consortium, Inc.**

Todos los otros copyrights son de los propietarios, a menos que se especifique otra cosa .

El uso de cualquier término en este documento no debería ser tomado como que afecte la validez de una marca registrada o servicio registrado.

Copyright 1994 Daniel Quinlan

Se permite la distribución y copia de este estándar siempre que se preserve este copyright y este mensaje en todas las copias.

Se permite a los participantes y contribuyentes de la **FSSTND** copiar y distribuir versiones modificadas de este estándar para propósitos de las actividades de estandarización del sistema de archivos solamente, bajo las condiciones abajo especificadas. Todas las copias o porciones de este documento deben identificar el título del documento y la sección, y deben acompañar este aviso en un lugar prominente.

Ninguna porción de este documento puede ser redistribuida en una forma modificada o recortada sin la aprobación previa de el coordinador del **FSSTND**. Cualquier entidad que busque permiso para distribuir cualquier material derivado de este documento (Que no sean copias idénticas) debe de contactar al coordinador del **FSSTND** para la licencia apropiada.

2 Prefacio

2.1 Estatus del Estándar

Ésta es la versión 1.2 de la Estructura del Sistema de Archivos de **Linux (Linux Filesystems Structure) FSSTND**

Las guías de este estándar están sujetas a cambio. El uso de la información contenida en este documento es bajo su propio riesgo.

3 Organización del Estándar

Este estándar está dividido en 6 partes:

- General: incluyendo un enunciado de enfoque, problemas, objetivos, y requerimientos de conformidad. (4 (Sección 1)).
- El Sistema de Archivos: Un enunciado de algunos principios guías. (5 (Sección 2)).

- El Directorio raíz `/`: (5.2 (Sección 3)).
- La jerarquía `/usr`: (6 (Sección 4)).
- La jerarquía `/var`: (7 (Sección 5)).
- Razonamientos adicionales y asuntos sin resolver (8 (Sección 6)).

3.1 Convenciones Tipográficas

La fuente tipo courier se usa para los nombres de archivos y directorios.

Los componentes de nombres de archivos que varían son representados por una descripción del contenido encerrada entre los caracteres "`<`" y "`>`". Las direcciones de correo electrónico están también encerradas entre "`<`" y "`>`" pero se muestran en la fuente usual.

Los componentes adicionales de los nombres de los archivos se encierran entre los caracteres "`[`" y "`]`" y pueden ser combinados con la convención "`<`" y "`>`". Por ejemplo, si existe un archivo que puede ser encontrado con o sin extensión, éste podría ser representado por `<nombre>[.extensión]`.

Las subcadenas variables de los nombres de archivos y directorios se indican con "`*`".

4 General

4.1 Enfoque

Este documento especifica una estructura estándar del sistema de archivos para los sistemas **Linux**, incluyendo la localización de archivos y directorios, y el contenido de algunos archivos de sistema.

El estándar de sistema de archivos ha sido diseñado para ser usado por desarrolladores de distribuciones, desarrolladores de paquetes e implementadores de sistemas. De cualquier forma está hecho para ser una referencia y no es un tutorial de como manejar un sistema de archivos **Linux** ó jerarquía de directorios.

Éstos son algunos de los problemas fundamentales que motivaron originalmente el esfuerzo de estandarización. No había una estructura única, bien aceptada estructura de directorios **Linux**, en su lugar había muchas estructuras cada una incompatible con las demás. Las jerarquías más ampliamente usadas no estaban bien estructuradas y diferían bastante de las estructuras de directorios modernas "estándares" (tales como System V, BSD, SunOS, y otras). El sistema de archivos era poco familiar e incómodo para los usuarios de **UNIX** con experiencia y los administradores que habían tenido experiencia con otros sistemas operativos similares a **UNIX**. La falta de regularidad también confundía a los recién-iniciados en **Linux**, especialmente aquellos que no tenían un conocimiento previo de **UNIX**. Cualquier incompatibilidad entre las distribuciones primarias de **Linux** y los paquetes de aplicación se resolvían por métodos de una naturaleza poco elegante. Los enlaces simbólicos(symbolic links) eran usados demasiado frecuentemente para arreglar los problemas. (De todas maneras, hay veces en las que los enlaces se usan para asegurar compatibilidad hacia atrás o para permitir a algunos sistemas específicos que tengan un sistema de archivos individual y muy particular).

En cualquier esfuerzo de estandarización surgen diferencias de opinión. La necesidad del consenso y la práctica común dentro de la comunidad de **Linux** deben opacar estas diferencias.

Este estándar del sistema de archivos fue primeramente desarrollado dentro de la lista de correo **FSSTND** y previamente, en el canal **FSSTND** de la lista de correo de los **LINUX-ACTIVISTS**. Los comentarios y recomendaciones fueron recibidos de un gran numero de desarrolladores de **Linux**, notables programadores de **Linux**, administradores de sistemas y usuarios. Estos voluntarios quienes han contribuido extensivamente al estándar están listados en el final de este documento. Este estándar representa la visión en consenso de éstos y otros contribuyentes.

Este estándar busca atacar los problemas estos problemas con una estructura de sistema de archivos bien diseñada que esperamos que la comunidad **Linux** seguirá voluntariamente. Aunque este estándar comprende más y es más completo que cualquier anterior intento de estandarización, lo más seguro es que nunca esté verdaderamente terminado. Las necesidades de la comunidad **Linux** cambiarán continuamente debido a las tecnologías emergentes. Es también muy posible que se descubran mejores soluciones a los problemas que nosotros atacamos o que las soluciones que nosotros ahora proponemos dejen de ser las mejores es por eso que el **FSSTND** planea publicar suplementos y actualizaciones periódicas a este documento.

Los comentarios relacionados con este documento se agradecen y son bienvenidos por el grupo **FSSTND**. Cualquier comentario o sugerencia de cambio deberá ser dirigida a al coordinador del **FSSTND** o si lo prefiere, cualquier coordinador listado en este documento. Los comentarios tipográficos o gramáticos deberán ser dirigidos a el coordinador **FSSTND**.

Existe también una FAQ mantenida por Ian McCloaghrie, que responde algunas de las preguntas más frecuentemente hechas acerca de este estándar. Si desea implementar el **FSSTND** o si tiene alguna pregunta por favor lea antes el **FSSTND FAQ**. Ésta está disponible vía ftp anonymous en tsx-11.mit.edu en el directorio /pub/linux/docs/linux-standards/fsstnd/FSSTND-FAQ.

Por favor no mande correo a la lista de correo sin antes contactar a el coordinador del **FSSTND** o a un colaborador listado. Los mensajes impropios no van a ser bien recibidos en la lista.

Preguntas de como interpretar ciertos artículos en este documento pueden ocasionalmente hacerse, si tiene necesidad de aclarar algún punto, por favor contacte al coordinador. Dado que este estándar representa el consenso de muchos participantes es importante asegurarse que cierta interpretación es también el consenso de su opinión colectiva. Por esa razón puede no ser posible una respuesta inmediata a menos que la pregunta halla sido objeto de discusión previa.

El coordinador del **FSSTND** es Daniel Quinlan <Daniel.Quinlan@linux.org>

4.2 Problemas específicos.

Naturalmente, existían ciertos problemas específicos que tratábamos de corregir cuando estandarizamos la estructura del sistema de archivos de **Linux**, estos son algunos de los más obvios y más grandes. Los directorios binarios principales /bin y /usr/bin no tienen divisiones bien definidas entre ellos. Como resultado la distribución de los binarios entre estos dos directorios varía grandemente entre en cada distribución de **Linux**. Al incluir ambos, los archivos binarios y los de configuración en /etc hace más confuso este directorio y más difícil de mantener para ambos, el administrador de sistema y el usuario inexperto (especialmente aquellos con sistemas grandes) La división entre lo que debe ser un archivo de configuración específico de un site, y lo que es un archivo de configuración local de una máquina es difícil de establecer. Muchas implementaciones comunes de /usr no pueden ser montadas sólo-lectura debido a que contienen archivos variables y directorios a los cuales se necesita escribir En un ambiente en red es deseable servir software a las estaciones de trabajo vía NFS. Tales sistemas de archivos pueden necesitar ser montados sólo-lectura, para que los accidentes o malicia deliberada desde un estación de trabajo no puedan dañar los archivos en el servidor. Ésto requiere la identificación y la separación de los archivos a los que una máquina debe escribir y los que son específicos de una máquina. Las estructuras de los sistemas de archivos **Linux** tradicionales no eran muy adecuadas para instalaciones en red, las cuales podrían requerir que componentes sólo-lectura dentro del sistema de archivos (principalmente en la jerarquía /usr) o involucrar estaciones sin discos.

Mientras que algunos de los mayores problemas fueron atacados, surgieron numerosos y adicionales que necesitan ser resueltos. Este estándar intenta atacar muchos de estos problemas, pero puede haber algo que fue pasado por alto. Si desea traer algo a nuestra atención, por favor note que hubo asuntos que se han discutido largamente y que no fueron incluidos en este estándar.

4.3 Objetivos

Al tratar de resolver los problemas arriba mencionados, se identificaron varios objetivos que necesitaban ser alcanzados en adición a los problemas más técnicos. Estas metas comprenden la corrección de problemas sobresalientes así como la validación de este estándar.

Resolver los problemas listados antes, y al mismo tiempo limitar las dificultades transicionales mientras se traslada desde los antiguos estandartes de facto. Ganar la aprobación de los distribuidores, desarrolladores y otra gente importante en la comunidad **Linux**, así como alentarlos a que compartan con nosotros sus sugerencias. Proveer un estándar que la comunidad **Linux** escoja seguir por que resuelve los problemas anteriores y provee la más sensata estructura de sistemas de archivos de las instalaciones **Linux**.

Algunos de estos objetivos se han completado total o parcialmente debido a la distribución limitada de este documento (en borrador) al distribuidor o desarrollador que solicitó alguno.

4.4 Historia y Progreso

El mensaje original que motivó este esfuerzo para reestructurar el sistema de archivos **Linux** fue escrito por Olaf Kirsh <okir@monad.swb.de> el 02 de Agosto de 1993 en el entonces canal NORMAL de la lista de correo de los LINUX-activists.

En corto tiempo se decidió que la mejor manera para acometer la necesaria reestructuración de el sistema de archivos de **Linux** sería la creación de una lista de correo separada con el fin de desarrollar un standard de consenso.

Después de una discusión comprensiva y con muy pocas discordias un borrador preliminar fue emitido, con la ayuda de algunas personas dedicadas, el borrador fue terminado y el borrador resultante sometido a consideración en el canal **FSSTND** para mayor discusión.

El primer borrador fue emitido al canal el 18 de Septiembre de 1993 por Daniel Quinlan.

Al tiempo que la discusión continuaba y los borradores de las recomendaciones de el **FSSTND** se desarrollaban más, se establecieron contactos con los desarrolladores más accesibles quienes entonces ofrecieron su apoyo y comentarios a nuestro esfuerzo.

Muchos desarrolladores de **Linux** estuvieron de acuerdo en que este esfuerzo de estandarización valía la pena y lo apoyaron.

A continuación estan algunos de los desarrolladores que intentan seguir el estándar **FSSTND**, parcial o completamente, listados en orden alfabético:

ATIM LINUX

PRO/ Fred N van Kempen et al. <waltje@infomagic.com >

BOGUS LINUX

Rik Faith, Kevin E. Martin, y Doug L. Hoffman <linux-bogus@cs.unc.edu >

DEBIAN LINUX

Ian A. Murdock <imurdock@debian.org >

LILO boot loader

Werner Almesberger <almesber@nessie.cs.id.ethz.ch >

MCC Interim LINUX

Owen LeBlanc <LeBlanc@mcc.ac.uk >

Red Hat Software LINUX (RHS LINUX)

Marc Ewing <marc@redhat.com>

Slackware LINUX

Patrick J. Volkerding <volkerdi@mhd1.moorehead.msus.edu>

TAMU LINUX

Dave Safford <dave.safford@net.tamu.edu>

util-linux (paquete)

Rik Faith <faith@cs.unc.edu>

Yggdrasil Plug-and-Play LINUX

Adam J. Richter <adam@yggdrasil.com>

4.5 Conformidad con este Documento

Esta sección define los términos "conforme" y "compatible" con respecto a este estándar, y el de "parcialmente" conforme y compatible.

Una "implementación" aquí se refiere a una distribución, un sistema instalado, un programa, un paquete (o alguna pedazo similar de software o datos), o algún componente de ellos.

Una implementación es totalmente conforme con este estándar si cada requerimiento en este estándar es cubierto. Cada archivo o directorio que sea parte de la implementación debe estar localizado como se especifica en este documento. Si el contenido de un archivo es descrito aquí, el contenido actual debe corresponder con el de la descripción. La implementación también debe intentar encontrar a los archivos o directorios (externos a sí mismo) primeramente o exclusivamente en el lugar especificado en este estándar.

Una implementación es totalmente compatible con este estándar si cada archivo o directorio que contiene puede ser encontrado viendo en el lugar especificado aquí, aún cuando este no sea el lugar primario o el lugar físico del archivo o directorio en cuestión. La implementación debe, cuando intente encontrar cualquier archivo o directorio que no sea parte de sí, en el lugar especificado en este estándar, aunque también puede intentar encontrarlos en otros lugares no-estándar.

Una implementación es parcialmente conforme o compatible si cumple con o es compatible con un subconjunto significativo de este documento. La conformidad o compatibilidad parcial se aplica solamente a distribuciones y no a programas separados. La frase subconjunto significativo es deliberadamente subjetiva y en casos marginales, la parte interesada deberá contactar al coordinador del **FSSTND**. Se anticipa que cierta variación será tolerada en casos marginales.

Para calificar como parcialmente conforme con **FSSTND** o parcialmente compatible con **FSSTND**, una implementación debe proporcionar una lista de lugares en los cuales ella y el **FSSTND** difieren, junto con una nota breve explicando la razón de la discrepancia. Esta lista deberá de ser provista con la distribución en cuestión y deberá estar disponible para la lista **FSSTND** o el coordinador del **FSSTND**.

Los términos "tiene que", "debe", "contiene", "es" y demás deben ser leídos como requerimientos para la conformidad o compatibilidad.

Note que una implementación no necesita contener todos los archivos y directorios especificados en este estándar para ser conforme o compatible. Sólo es necesario que aquellos archivos y directorios que sí contiene, que estén localizados apropiadamente. Por ejemplo si el sistema de archivos ext2 no está soportado por una distribución, las herramientas ext2 no necesitan estar incluidas, aunque se mencionen explícitamente en la sección sobre /sbin.

Más aún, ciertas porciones de este documento son opcionales. En este caso se enunciará explícitamente, o se indicará con el uso de una o más de las palabras " puede", "se recomienda" o "se sugiere" . Los artículos marcados como opcionales no tienen influencia en la conformidad o compatibilidad de una implementación, sólo son sugerencias pensadas para motivar la práctica común, pero pueden estar localizados en cualquier lugar a juicio del implementador.

5 El sistema de Archivos

El sistema de archivos **UNIX** está caracterizado por:

Una estructura jerárquica.

Un tratamiento consistente de la información de los archivos.

Protección de los archivos.

Este estándar del sistema de archivos **Linux** sigue el mismo principio básico que la mayoría de los sistemas de archivos **UNIX** siguen. Note, sin embargo que este estándar no intenta concordar en cada aspecto posible con alguna implementación particular del sistema **UNIX**. De cualquier forma, muchos de los aspectos de este estándar están basados en ideas encontradas en **UNIX** y sistemas similares a **UNIX**.

Es posible después de cuidadosa consideración de otros factores, incluyendo:

Prácticas comunes en la comunidad **Linux**.

La implementación de otras estructuras de sistemas de archivos.

Los estándares aplicables.

Definir dos categorizaciones ortogonales de archivos: Compartibles vs. no compartibles, y variables vs. estáticos.

La información compartible es aquella que puede ser compartida entre varias máquinas diferentes; la no compartible es aquella que debe ser local a una máquina particular. Por ejemplo. Los directorios hogar de los usuarios son compartibles, pero los archivos de bloqueo de dispositivo (lock files) son no compartibles.

La información estática incluye binarios, librerías, documentación y todo aquello que no cambia sin la intervención del administrador del sistema. La información variable es todo lo que cambia sin la intervención del administrador.

El entendimiento de estos principios básicos ayudará a guiar la estructura, a lo largo de este documento, y en cualquier sistema de archivos bien planeado, esto brindará consistencia adicional.

La distinción entre información compartible y no compartible es necesaria por varias razones:

En un ambiente de red (i.e. más de un host en un site), existe una buena cantidad de información que se puede compartir entre diferentes máquinas para ahorrar espacio y facilitar la tarea de administración.

En un ambiente de red, ciertos archivos contienen información específica a una sola máquina, por tanto, estos sistemas de archivos no pueden ser compartidos (sin tomar medidas especiales).

Las implementaciones de facto del sistema de archivos no permitían que la jerarquía `/usr` fuera montada sólo-lectura, porque contenía archivos y directorios que necesitaban ser escritos muy frecuentemente. Éste es un factor que debe atacarse cuando algunas partes de `/usr` se comparten en una red, o se montan sólo-lectura debido a otras consideraciones tales como la seguridad.

La distinción "compartible" puede ser usada para soportar, por ejemplo:

- Una partición `/usr` (o componentes de `/usr`) montada (sólo-lectura) a través de la red (usando NFS).
- Una partición `/usr` (o componentes de `/usr`) montada desde medios de sólo-lectura. Un cd-rom puede ser considerado como un sistema de archivos sólo-lectura compartido con otros sistemas **Linux** utilizando

el sistema de correo como una red.

La distinción "estática" contra "variable" afecta el sistema de archivos de dos maneras principales:

Dado que / contiene ambos tipos de información, variable y estática necesita montarse lectura-escritura.

Dado que el /usr tradicional contiene ambos tipos de información variable y estática y dado que podríamos desear montarlo sólo-lectura (vea arriba), es necesario proporcionar un método para hacer que /usr se monte sólo-lectura. Ésto se logra con la creación de una jerarquía /var que se monta lectura-escritura (o es parte de una partición lectura-escritura tal como /), que toma mucho de la funcionalidad tradicional de la partición /usr.

5.1 Tabla con ejemplos.

CompatibleNo-CompatibleEstática/usr/etc/home/bootVariable/var/spool/mail/var/run/var/spool/news/var/lock3.
El directorio raíz /. Esta sección describe la estructura del directorio raíz. El contenido del sistema de archivos raíz será el adecuado para arrancar, bootear, restaurar, recuperar y/o reparar el sistema:

Para arrancar el sistema, debe estar presente lo suficiente como para montar /usr y otras partes no-esenciales del sistema de archivos. Ésto incluye herramientas, información de configuración y del cargador de arranque (boot loader) y alguna otra información esencial al arrancar.

Para habilitar la recuperación y/o la reparación del sistema, estará presente en el sistema de archivos raíz aquellas herramientas que un administrador experimentado necesitaría para diagnosticar y reconstruir un sistema dañado.

Para restaurar un sistema, estarán presentes en el sistema de archivos raíz aquellas herramientas necesarias para restaurar el sistema desde respaldos (en floppy, cintas, etc).

La principal preocupación que se usa para balancear las anteriores consideraciones, que favorecen el colocar muchas cosas en el sistema de archivos raíz, es la meta de mantener / tan pequeño como sea razonablemente posible. Por varias razones es deseable mantener el sistema de archivos /

Es frecuentemente montado desde media muy pequeña. Por ejemplo muchos usuarios de **Linux** instalan y recuperan sistemas montando / como un disco ram, que es copiado de un disco de 1.44Mb único.

El sistema de archivos / tiene muchos archivos de configuración específicos de un sistema. Posibles ejemplos son un kernel que es específico al sistema, un hostname diferente, etc. Ésto significa que el sistema de archivos / no es siempre compatible entre sistemas en red. Manteniéndolo pequeño en sistemas en red, se minimiza el espacio perdido en los servidores por archivos no-compatibles. También permite estaciones de trabajo con discos duros locales más pequeños.

Aunque usted podría tener el sistema de archivos / en una partición grande, y ser capaz de llenarla según sus deseos, siempre habrá gente con particiones más pequeñas. Si usted tiene más archivos instalados, podría encontrar incompatibilidades con otros sistemas que utilizan un sistema de archivos / en particiones más pequeñas. Si usted es un desarrollador entonces estaría volviendo su suposición en un problema para un gran número de usuarios.

Los errores del disco, que corrompen la información en el sistema de archivos / son un problema mayor que los errores en cualquier otra partición. Un sistema de archivos / pequeño es menos propenso a corromperse como resultado de una falla del sistema.

En este documento, actualmente se requiere un sistema de archivos / escribible (debido principalmente a /etc/mstab). De cualquier forma, no se necesita que el sistema de archivos / esté totalmente almacenado localmente. La partición / no tiene porque estar almacenada localmente para ser específica del sistema por ejemplo, podría estar montada de un servidor NFS.

El software no deberá crear o requerir archivos o subdirectorios especiales en el directorio /. La estructura del sistema de archivos **Linux** proporciona más que suficiente flexibilidad para cualquier paquete. Cualquier paquete que ocupe un directorio bajo la raíz / del sistema de archivos sufre de bastante arrogancia.

5.2 El Directorio Raíz

bin	Binarios de comandos esenciales
boot	Archivos estáticos de cargador de arranque(boot-loader)
dev	Archivos de dispositivos
etc	Configuración del sistema local-máquina
home	Directorios home de los usuarios
lib	Librerías compartidas
mnt	Punto de montaje de particiones temporales
root	Directorio hogar del usuario root
sbin	Binarios del sistema esenciales
tmp	Archivos temporales
usr	Segunda jerarquía mayor
var	Información variable

Cada directorio listado será discutido en detalle en una subsección separada más adelante. /usr y /var, cada uno tiene en su propia sección en este documento.

El kernel de **Linux** estaría localizado en, ya sea / ó en /boot. Si está localizado en / recomendamos usar el nombre VMLINUX o VMLINUZ, nombres que han sido usados en paquetes fuentes del kernel de **Linux** recientes. Más información de la localización del kernel se puede encontrar en la sección acerca de / más adelante.

5.3 /bin Binarios de comandos esenciales de usuarios (disponibles para todos los usuarios).

bin contiene comandos que pueden ser utilizados por ambos los usuarios y /el administrador del sistema, pero que son requeridos en el modo /mono-usuario (single-user mode) puede también contener comandos que son /utilizados indirectamente por algunos scripts.

Todos los binarios utilizables sólo por root, tales como daemons,init,getty, update, etc. Estarían localizados en /sbin ó /usr/sbin dependiendo si son o no esenciales. Para una mayor discusión de la definición de que es esencial en el sistema de archivos /, lea por favor la sección 6, "Razonamientos adicionales y asuntos sin resolver".

No habrá subdirectorios dentro de /bin.

Los binarios de los comandos que no son suficientemente esenciales para estar en /bin estarán localizados en /usr/bin, los elementos que son utilizados por usuarios solamente (pero no por root) (mail,chsh, etc) no son suficientemente esenciales para estar dentro de la partición /.

Archivos requeridos en /bin:

5.3.1 Comandos generales:

Los siguientes comandos han sido incluidos porque son esenciales. algunos están presentes debido a que tradicionalmente han estado en /bin.

arch, cat, chgrp, chmod, chown, cp, date, dd, df, dmesg, echo, ed, false,kill, in, login, mkdir, mknod, mo

Si /bin/sh es Bash, entonces /bin/sh sería en enlace simbólico o duro a /bin/bash dado que bash se comporta diferente cuando es llamado como sh ó bash. La pdksh que puede ser la /bin/sh en los discos de instalación y sería igualmente arreglada a que /bin/sh sea un enlace simbólico a /bin/ksh. El uso de enlaces simbólicos en estos casos permite que los usuarios vean fácilmente que /bin/sh no es una shell estilo bourne.

Dado que la localización estándar de facto de shell estilo c es `/bin/csh`, si y sólo si está disponible en el sistema una shell estilo c ó equivalente (tal como `/bin/tcsh`, esta, estaría disponible con el nombre `/bin/csh`. `/bin/csh` puede ser un enlace simbólico a `/bin/tcsh` ó `/usr/bin/tcsh`).

Los comandos `[` y `test` están interconstruidos en `bash`, `pksh`, `zsh`, y las shell `korn` recientes, esencialmente cada remplazo de las shell tipo `bourne` que hay para **Linux**. Estos comandos estarían localizados dentro de `/usr/bin`. (se deben incluir como binarios separados con cualquier sistema **Linux** que intente cumplir con el estándar `POSIX`).

`bin/arch` produciría el mismo resultado que `uname-m`, específicamente; `386 /o`; `486` para sistemas `intel` y compatibles.

5.3.2 Comandos para restauración.

Estos comandos se han incluido para hacer posible el restaurar el sistema (siempre que `/` este intacto).

`tar`, `gzip`, `gunzip` (enlace hacia `gzip`), `zcat` (enlace hacia `gzip`).

Si se hacen respaldos de sistemas utilizando otros programas, entonces la partición `/` contendrá los componentes mínimos necesarios. Por ejemplo, muchos sistemas incluirían `cpio` como la segunda utilidad más usada para respaldos después de `tar`. Pero si jamás se espera restaurar el sistema desde la partición `/`, entonces estos binarios se pueden omitir (i.e., montar `/` en chip `ROM`, montar `/usr` desde `NFS`). Si la restauración del sistema se planea a través de la red, entonces `FTP` ó `TFTP` (junto con todo lo necesario para obtener una conexión `FTP`) estarían disponibles en la partición `/`.

Los comandos de restauración pueden aparecer en, ya sea `/bin` ó `/usr/bin` en sistemas **Linux** diferentes.

5.3.3 Comandos de red.

Éstos son únicamente los binarios de red que los usuarios y `root` querrán o necesitarán ejecutar que no sean los que están en `/usr/bin` ó `/usr/local/bin`

`domainname`, `hostname`, `netstat`, `ping`.

5.4 `/boot`: Archivos estáticos del cargador de arranque (boot loader).

Este directorio contiene todo para arrancar excepto los archivos de configuración y el instalador de mapas. En su sentido más sencillo `/boot` es para cualquier cosa que se utiliza antes de que el kernel ejecute `/sbin/init`. Ésto incluye sectores maestros de arranque (master boot sectors) guardados, archivos de mapeo de sectores y cualquier otra cosa que no es editada directamente a mano. Los programas necesarios para arreglar que el cargador de arranque sea capaz de arrancar un archivo (tal como el instalador de mapas [`lilo`]) estarán localizados en `/sbin`. Los archivos de configuración para cargadores de arranque podrían estar localizados en `/etc`.

Como se expuso arriba, el kernel de **Linux** puede estar localizado en `/` ó en `/boot`, si se localiza en `/boot`, recomendamos que se le dé un nombre más descriptivo.

5.5 `/dev` Archivos de dispositivos.

Éste es el directorio de los dispositivos. Contendría un archivo por cada dispositivo que el kernel de **Linux** puede soportar.

dev también contiene un script llamado MAKEDEV el cual puede crear /dispositivos cuando se necesiten. Puede contener un MAKEDEV local para /dispositivos sólo-local.

MAKEDEV debe hacer previsión para crear cualquier archivo de dispositivo especial listado en la lista de numeros mayores/menores, no sólo aquellos de una distribución particular.

Los enlaces simbólicos no se deben distribuir en sistemas **Linux**, sino sólo como se prevé en la lista de dispositivos de **Linux**. Ésto es porque las instalaciones locales seguro diferirán de aquellas de la máquina del desarrollador. Además si un script de instalación configura enlaces simbólicos en la instalación, estos enlaces seguramente no se actualizarán si se hacen cambios locales en el hardware. Cuando se usan responsablemente, como sea, son de buen uso.

Este documento incorpora como referencia la lista de dispositivos de **Linux**, mantenida por: Peter.Anvin@linux.org: El encargado de los dispositivos **Linux**. Todos los archivos especiales de dispositivo seguirán el estándar en ese documento, que está disponible en ftp.yggdrasil.com en /pub/device-list.

5.6 /etc : Configuración del sistema local a la máquina.

etc contiene archivos y directorios que son locales al sistema actual.

Ningún binario debe ir directamente dentro de /etc. Los binarios que en el pasado se encontraban en /etc, irán en /sbin ó /usr/sbin. Ésto incluye archivos tales como init, getty y update. Los binarios tales como hostname que son utilizados por usuarios ordinarios y por root no irán en /sbin sino en /bin.

```
/etc --- Configuración de sistemas locales de máquina.
```

```
X11          Archivos de configuración para el x11
skel         Esqueletos de configuración de usuarios
```

etc/skel es la localidad para los llamados archivos esqueletos de /usuarios, que le son dados por defecto cuando un nuevo usuario recibe una /cuenta, este directorio puede contener subdirectorios para diferentes /grupos de usuarios (i.e. /etc/skel/apoyo, /etc/skel/usuarios).

etc/X11 es el lugar recomendado para todos los archivos de configuración /de X11 locales a la máquina. Este directorio es necesario para permitir el /control local si /usr se monta sólo-lectura. Los archivos que deben ir en /este directorio incluyen Xconfig (y/o XF86Config) y Xmodmap.

Los subdirectorios de /etc/X11 pueden incluir aquellos para xdm y para cualesquier otros programas (como algunos manejadores de ventanas por ejemplo) que lo necesiten. Recomendamos que los manejadores de ventanas con un solo archivo de configuración que es un archivo *.wmrc por defecto, que lo llamen system.*wmrc (a menos que exista una alternativa ampliamente aceptada) y que no utilice un subdirectorio. Cualquier subdirectorio de un manejador de ventanas se llamaría idéntico al binario del manejador de ventanas.

etc/X11/xdm retiene los archivos de configuración de xdm. Ésto es la /mayoría de los archivos normalmente hallados en /usr/lib/X11/xdm; Vea la /sección 5, /var/lib/xdm, para mayor información.

La siguiente sección intenta parcialmente examinar la descripción del contenido de /etc con algunos ejemplos: Definitivamente ésta no es una lista exhaustiva.

Archivos requeridos en /etc:

5.6.1 Archivos generales:

Estos archivos son necesarios en la mayoría de los sistemas **Linux**.

adjtime, csh.login, disktab, fdprm, fstab, gettydefs, group, inittab, issue, ld.so.conf, lilo.conf, magic,

5.6.2 Archivos de Red:

Estos archivos estarían instalados en la mayoría de los sistemas **Linux**.

```
exports, ftpusers, gateways, hosts, host.conf, host.equiv, host.lpd,
inetd.conf, networks, printcap, protocols, resolv.conf.rpc, services
```

Hay dos modelos para la instalación de los scripts de comandos "rc" los cuales son invocados por `init(8)` al momento de arrancar, el modelo `/etc/rc.d/*` estilo SystemV. Cualquiera puede ser utilizado o una mezcla de los dos.

Los sistemas con la suite de passwords sombreadas (shadow password) tendrán archivos de configuración adicionales, en `/etc` (`/etc/shadow` y otros) y `/usr/bin` (`useradd`, `usermod`, y otros).

5.7 `/home`: Directorios hogar de los usuarios (opcional)

`home` es un concepto algo estándar, pero es claramente un sistema de `/archivos` específico de un site. El arreglo diferirá de máquina a máquina. Esta sección describe una localización sugerida para los directorios hogar `/de` los usuarios, aun así, recomendamos que todas las distribuciones **Linux** usen este lugar como la localización por defecto de los `/directorios` hogar.

En sistemas pequeños, cada directorio de usuario es uno de los subdirectorios debajo de `/home`, p.ej. `/home/smith`, `/home/torvalds`, `/home/operador`, etc.

En sistemas grandes (especialmente cuando los directorios `/home` son compartidos entre varias máquinas usando NFS) es útil subdividir los directorios hogar. La subdivisión puede ser llevada a cabo utilizando subdirectorios tales como `/home/apoyo`, `/home/huéspedes`, `/home/estudiantes`, etc.

Muchas personas prefieren poner las cuentas de los usuarios en una variedad de lugares. Por tanto, ningún programa deberá confiar en esta localización. Si usted desea encontrar el directorio hogar de cualquier usuario, debería usar la función de librería `getpwent(3)` en vez de contar con `/etc/passwd`, por que la información puede estar almacenada remotamente usando usando sistemas como NIS.

5.8 `/lib`: Librerías compartidas y módulos de kernel esenciales

El directorio `/lib` contiene aquellas imágenes de las librerías compartidas que se necesitan para arrancar el sistema y ejecutar los comandos en el sistema de archivos raíz.

`lib` — librerías compartidas y modulos de kernel esenciales.
`modules` Modulos de kernel cargables.

Esto incluye `/lib/libc.so.*`, `/lib/libm.so.*`, el enlazador dinámico compartido `/lib/ld.so.*`, y otras librerías compartidas requeridas por binarios en `/bin` y `/sbin`.

Las librerías que son necesitadas sólo por los binarios en `/usr` (como cualquier binario de X Window) no pertenecen a `/lib`. Sólo las librerías compartidas requeridas para ejecutar los binarios dentro de `/bin` y `/sbin` deben estar aquí. La librería `libm.so.*` podría estar localizada en `/usr/lib` si no es requerida por nada en `/bin` ó `/sbin`.

Por razones de compatibilidad, `/lib/cpp` necesita existir como una referencia al pre-procesador C instalado en el sistema. La localización usual del binario es `/usr/lib/gcc-lib/<target>/<version>/cpp`. Puede existir un enlace `/lib/cpp` apuntando a este binario o a cualquier otra referencia a este binario que exista en el sistema de archivos. (Por ejemplo, `/usr/bin/cpp` se usa frecuentemente).

La especificación para `/lib/modules` está aún por aparecer.

5.9 `/mnt`: Punto de montaje para sistemas de archivos montados temporalmente.

Este directorio se ha provisto para que el administrador pueda montar temporalmente sistemas de archivos cuando lo necesite. El contenido de este directorio es un asunto local y no debe afectar la manera en la cual se ejecuta ningún programa.

Recomendamos la no utilización de este directorio por programas de instalación, y sugerimos utilizar un directorio temporal adecuado que no este en uso por el sistema.

5.10 `/proc`: Sistema de archivos virtual de informacion de procesos y del kernel.

El sistema de archivos `proc` se está convirtiendo en el estándar de facto para el manejo de informacion de procesos y de sistema en vez de `/dev/kmem` y otros metodos similares. Recomendamos fuertemente esto para el almacenamiento y obtención de información de procesos asi como otra información del kernel y de memoria.

5.11 `/root`: Directorio hogar de root (opcional)

El directorio `/` es tradicionalmente el directorio hogar del usuario `root` en los sistemas **UNIX**. `/root` se usa en muchos sistemas **Linux** y en algunos sistemas **UNIX**. El directorio hogar de la cuenta de el usuario `root` puede ser determinada por el desarrollador o por preferencias locales. Las posibilidades obvias incluyen `/`, `/root`, y `/home/root`.

Si el directorio hogar de `root` no está almacenado en la partición raíz, será necesario asegurarse que tome `/` por defecto si no puede ser localizado.

NOTA: Recomendamos contra el uso de la cuenta `root` para cosa mundanas tales como leer el correo y ver las noticias (`mail & news`) sino que se use solamente para la administración del sistema. Por esta razón recomendamos que no aparezcan subdirectorios como `Mail` y `News` en el directorio hogar de la cuenta del usuario `root`. Recomendamos que el `Mail` para `root` y `postmaster` sean redirigidos a un usuario más adecuado.

5.12 `/sbin`: Binarios del Sistema (Alguna vez mantenidos en `/etc`)

Los útiles usados por la administración del sistema (y otros comandos que sólo `root` utiliza) están almacenados en `/sbin`, `/usr/sbin`, y `/usr/local/sbin`. `/sbin` típicamente contiene binarios esenciales para arrancar el sistema ademas de los binarios en `/bin`. Cualquier cosa que se ejecuta después de que se sabe que `/usr` se ha montado (cuando no hay problemas) debería estar en `/usr/sbin`. Los binarios de administración de sistema sólo-locales deben estar localizados en `/usr/local/sbin`.

Decidir que cosa va en los directorios de `/sbin` es sencillo: Si un usuario necesitará ejecutarlo, debe de ir en otro lado. Si sólo será ejecutado por el administrador del sistema o por `root` como scripts de administración, entonces debe ir en `/sbin` (o en `/usr/sbin` o en `/usr/local/sbin`, si el archivo no es vital para la operación del sistema).

Archivos como `chfn` que los usuarios usan sólo ocasionalmente deben aun estar en `/usr/bin`. `ping` aunque es absolutamente necesario para el `root` (recuperación de la red y diagnóstico) es tambien frecuentemente usado por los usuarios y por esa razon debe ir en `/bin`.

Los usuarios ordinarios no tendrán que poner ninguno de los directorios `sbin` en su búsqueda (`path`).

Recomendamos que los usuarios tengan permisos de lectura y ejecución en todo lo que se encuentra en /sbin excepto tal vez ciertos programas; setuid y setgid. La división entre /sbin y /bin no fue creada por motivos de seguridad o para evitar que los usuarios vieran el sistema operativo, sino para proveer una buena partición entre binarios que todos usan y los que se usan, principalmente las tareas de administración. No hay ganancia inherente en seguridad en hacer que /sbin este fuera del alcance de los usuarios.

Archivos requeridos en /sbin:

5.12.1 Comandos Generales.

`clock, getty, init, update, mkswap, swapon, swapoff, telinit.`

5.12.2 Comandos de Apagado.

`fastboot, fasthalt, halt, reboot, shutdown.`

5.12.3 Comandos de manejo de sistemas de archivos.

`fdisk, fsck, fsck.*, mkfs, mkfs.*`

donde * = uno de los siguientes.

ext, ext2 minix, msdos, xia, y tal vez otros.

5.12.4 Comandos del sistema de archivos ext2 (opcional)

`badblocks, dumpe2fs, e2fsck, mke2fs, mklost+found, tune2fs.`

Instalador del mapa del cargador de arranque.

`lilo`

5.12.5 Comandos de Red.

`arp, ifconfig, route.`

Archivos opcionales en /sbin:

5.12.6 Binarios estáticos. (compilados estáticamente)

ln estático sln y sync estático ssync son útiles cuando las cosas salen mal. El principal uso de sln (reparar enlaces simbólicos incorrectos en /lib despues de una actualización mal orquestrada) ya no es preocupación mayor ahora que existe el programa ldconfig (usualmente localizado en /usr/sbin) y puede actuar como una mano guiadora al actualizar las librerías dinámicas. sync estático es útil en algunas ocasiones de emergencia. Note que estas no necesitan ser versiones compiladas estáticamente de los ln y sync estándares, pero pueden ser.

El binario ldconfig es opcional en /sbin, dado que un site puede escoger ejecutar ldconfig al arrancar, en vez de sólo cuando se actualizan las librerías compartidas. (No está claro si es o no ventajoso ejecutar ldconfig en cada arranque). Aun así, a algunos les gusta tener ldconfig a la mano para las siguientes (muy comunes) situaciones:

Se acaba de remover /lib/<archivo>.

No se puede encontrar el nombre de la librería porque ls está enlazado dinámicamente. Se está usando una shell que no tiene ls interconstruida y no se sabe como usar "echo * " como remplazo.

Se tiene un sln, pero no se sabe como nombrar al enlace.

`ldconfig, sln, ssync.`

5.12.7 Misceláneos

Para lidiar con el hecho de que muchos teclados vienen con una tasa de repetición tan alta como para hacerlos inutilizables, se puede instalar `kbdrate` en `/sbin` en algunos sistemas. Dado que la acción por defecto del kernel ante la combinación de teclas `Ctrl-Alt-Del` es un rearranque instantáneo duro, es recomendable generalmente deshabilitar esta conducta antes de montar el sistema de archivos raíz con modo lectura-escritura. Algunas suites `init` son capaces de deshabilitar `Ctrl-Alt-Del`, pero otras pueden requerir el programa `ctrlaltdel`, el cual puede ser instalado en `/sbin` en estos sistemas.

`ctrlaltdel`, `kbdrate`

5.13 /tmp: Archivos temporales.

`tmp` se utiliza para archivos temporales, preferentemente en un dispositivo rápido (un sistema de archivos basado en memoria por ejemplo)

La "persistencia" de la información que es almacenada en `/tmp` es diferente de aquella que sea almacenada en `/var/tmp`. `/tmp` puede ser limpiada en cada arranque o a intervalos relativamente frecuentes. Por tanto, no se debe esperar que la información almacenada en `/tmp` permanezca por algún periodo largo de tiempo.

Los programas deben utilizar `/tmp` ó `/var/tmp` (que era originalmente `/usr/tmp`) de acuerdo a los requerimientos esperados de la información, pero no deben confiar en alguna persistencia temporal particular en cualquier directorio de almacenamiento temporal.

Los administradores de sistemas pueden elegir enlazar `/tmp` a algún otro directorio, tal como `/var/tmp`; esto es útil, por ejemplo, para conservar espacio en la partición raíz. Si ésto se lleva a cabo, entonces la persistencia de archivos en `/var/tmp` debe ser al menos tan larga como la de `/tmp`.

`tmp` puede estar en un disco RAM. `/var/tmp` no debe nunca localizarse en algún dispositivo RAM.

6 La jerarquía /usr

`usr` es la segunda mayor sección del sistema de archivos. `/usr` es información compartible, de sólo-lectura, esto significa que `/usr`, debe ser compartible entre varias máquinas que corren **Linux** y no se debe escribir. Cualquier información que es local a una máquina o varía con el tiempo, se almacena en otro lugar.

Ningún paquete grande (como **TeX** o **GNU Emacs**) debe utilizar un subdirectorio directo bajo `/usr`, en vez, debe haber un subdirectorio dentro de `/usr/lib` (o `/usr/local/lib` si fué instalado completamente local) para ese propósito, con el sistema X Window se hace una excepción debido a un considerable precedente y a la práctica ampliamente aceptada.

`/usr --- Segundo mayor punto de montaje (permanente)`

<code>X11R6</code>	Sistema X Window Version 11 release 6
<code>X386</code>	Sistema X Windows Version 11 release 5 en plataformas X 86
<code>bin</code>	La mayoría de los comandos de usuario
<code>dict</code>	Listas de palabras
<code>doc</code>	Documentación miscelánea
<code>etc</code>	Configuración del Sistema (todo el site)
<code>games</code>	Juegos y binarios educativos
<code>include</code>	Archivos header incluidos por programas C
<code>info</code>	Directorio primario del sistema GNU Info
<code>lib</code>	Librerías

local	Jerarquía local (vacía justo después de la instalación principal)
man	Manuales en línea
sbir	Binarios de Administración del Sistema No-Vitales
share	Información independiente de la arquitectura
src	Código fuente

Los siguientes enlaces simbólicos a directorios pueden estar presentes. Esta posibilidad se basa en la necesidad de preservar la compatibilidad con sistemas anteriores hasta que en todas las implementaciones se pueda asumir el uso de la jerarquía /var.

```

/usr/adm -----> /var/adm
/usr/preserve -----> /var/preserve
/usr/spool -----> /var/spool
/usr/tmp -----> /var/tmp
/var/spool/locks -----> /var/lock

```

Una vez que el sistema ya no requiera más alguno de los anteriores enlaces simbólicos, el enlace se puede remover, si se desea. Notablemente, sólo se necesita poco esfuerzo para remover completamente /usr/preserve, dado que sólo ex y vi lo utilizan.

6.1 /usr/X11R6: El sistema X Window, Version 11 release 6

Esta jerarquía está reservada para el sistema X Window, Version 11 release 6 y archivos relacionados.

```

/usr/X11R6 --- X Window System (Version 11, release 6)

bin
doc
include
lib
man

```

Para simplificar los problemas y hacer **XFree86** más compatible con el sistema X Window en otros sistemas, los siguientes enlaces simbólicos deben estar presentes.

```

/usr/bin/X11 -----> /usr/X11R6/bin
/usr/lib/X11 -----> /usr/X11R6/lib/X11
/usr/include/X11 -----> /usr/X11R6/include/X11

```

En general, el software no se debe instalar o manejar vía los anteriores enlaces simbólicos. Sólo están para la utilización por usuarios. La dificultad está relacionada con la versión y el release del sistema X Window; en períodos transicionales es imposible saber que release de X11 está utilizandose .

Por la misma razón no debe existir un enlace desde /usr/X11 apuntando a la jerarquía del sistema X Window actual.

6.2 /usr/X386: El sistema X Window, Version 11 release 5 en plataformas X 86

Esta jerarquía es generalmente idéntica a /usr/X11R6, excepto que los enlaces simbólicos de /usr deben estar ausentes si está instalado /usr/X11R6

6.2.1 /usr/bin: La mayoría de los comandos del usuario.

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

```
mh      comandos para el sistema de manejo de correo M H
X11    Enlace simbólico hacia /usr/X11R6/bin
```

Debido a que los interpretadores de scripts de los shell (invocados con `#! <ruta>` en la primera línea del script de shell) no pueden depender de una ruta, es ventajoso el estandarizar la localización de ellos. La shell Bourne y C están fijos en /bin, pero Perl, Python, Tcl se encuentran en muchos lugares diferentes /usr/bin/perl, /usr/bin/python y /usr/bin/tcl deben referenciar a los intérpretes de shell perl, python y tcl respectivamente. Éstos pueden ser enlaces simbólicos a la localización física de los intérpretes de shell.

6.3 /usr/dict: Listas de palabras

Archivos recomendados en /usr/dict

words

Tradicionalmente este directorio contiene sólo el archivo words de palabras inglesas, el cual es utilizado por look(1) y varios programas de ortografía, words puede utilizar ortografía americana o británica. Los sites que requieran ambos, pueden enlazar words a /usr/dict/american-english ó /usr/dict/british-english.

Las listas de palabras para otros lenguajes se pueden añadir usando el nombre en inglés para ese lenguaje, por ejemplo, /usr/dict/french, /usr/dict/danish, etc. Éstos deben, si es posible, utilizar un juego de caracteres ISO 8859 que sea apropiado para el lenguaje en cuestión, si es posible el juego de caracteres ISO 8859-1 (Latin1) debe ser utilizado (esto es a veces imposible)

Cualquier otra lista de palabras, tal como el directorio web2, debe ser incluido aquí, si está presente.

Las razones tras tener sólo las listas de palabras aquí es que ellas son los únicos archivos comunes a todos los verificadores de ortografía.

6.4 /usr/etc: Configuración del sistema (todo el site)

Almacenar la configuración en /usr/etc del software que se encuentra en /usr/bin y /usr/sbin es un problema. Hace que el montar /usr sólo-lectura de un CDROM o a través de NFS sea difícil en el mejor de los casos.

Una posible solución que se consideró fue eliminar completamente /usr/etc y especificar que todas las configuraciones se almacenen en /etc. Un problema con esta aproximación es que no anticipa propiamente la posibilidad de que muchos sites pueden querer tener algunos archivos de configuración que no sean locales de máquina.

Eventualmente se decidió que /etc deberá ser el único directorio que sea referenciado por los programas (esto es, todos deben buscar configuraciones en /etc y no en /usr/etc). Cualquier archivo de configuración que necesite ser para todo el site y que no es necesario antes de montar /usr (o en una situación de emergencia debe entonces estar localizado en /usr/etc. Entonces archivos específicos (en /etc), en máquinas específicas pueden ser o no ser enlaces simbólicos a los archivos de configuración localizados en /usr/etc. Ésto también significa que /usr/etc es técnicamente un directorio opcional en el sentido estricto, pero aún así recomendamos que todos los sistemas **Linux** lo incorporen.

No se recomienda que /usr/etc contenga enlaces simbólicos que apunten a archivos en /etc. Ésto es innecesario e interfiere con el control local en máquinas que comparten un directorio /usr.

6.5 /usr/include: Directorio para archivos include estándar.

Aquí es donde todos los archivos include de uso general del sistema para programación en lenguajes C y C++ deben ser localizados.

/usr/include	Archivos include
X11	Enlace simbólico hacia /usr/X11R6/include/X11
arpa	Definiciones del protocolo definido por ARPNET.
asm	Enlace simbólico hacia /usr/src/linux/include/asm-<arch>.
bsd	Archivos include de compatibilidad con BSD.
g++	Archivos include de GNU C++.
gnu	Archivos include GNU.
linux	Enlace simbólico a /usr/src/linux/include/linux.
net	Definiciones genéricas relacionadas con redes.
netax25	Definiciones específicas a +AX25 (ARRL AX25).
netinet	Definiciones específicas a TCP/IP.
netipx	Definiciones específicas a +IPX (Novel IPX/SPX).
protocols	Definiciones de protocolos (Mayormente basadas en INET)
readline	La librería readline GNU.
rpc	Definiciones RPC de Sun Microsystems.
rpcsvc	Definiciones de servicios RPC de Sun Microsystems.
sys	Archivos include de generación de sistemas.

El subdirectorio arpa contiene definiciones de cabecera de protocolos para los protocolos ARPANET, TCP/IP, definiciones para ftp, prototipos telnet y material similar.

El subdirectorio net contiene definiciones genéricas relacionadas con redes, define la interface sistema-kernel, detalles de la familia de protocolo, etc.

El subdirectorio netinet contiene definiciones específicas de INET (DARPA Internet, que también es conocida como TCP/IP)

ARRL AX.25 es mejor conocido como packet radio. Los protocolos novell IPX/SPX son parte de los servicios de archivos Novell Netware.

6.6 /usr/lib: Librerías para programas y paquetes.

usr/lib incluye librerías objeto, binarios del programa compilador e /información estática de varias clases, ambos, códigos ejecutable (por /ejemplo los binarios internos de gcc están localizados bajo //usr/lib/gcc-lib) y otros tipos de información.

/usr/lib/ - librerías para programación y paquetes:

X11	Enlace simbólico a /usr/X11R6/lib/X11
emacs	Archivos de soporte estáticos para el editor GNUEmacs.
games	Archivos de datos estáticos para /usr/games.
groff	Librerías / Directorios para GNU groff
gcc-lib	Archivos/ Directorios específicos del sistema para gcc.
kbd	Tablas de traducción de teclado e información relacionada.
Mh	Librerías para el sistema de manejo de correo MH:
news	Cnews/INN.
smail	Smail.
terminfo	Directorios para la base de datos terminfo.
texmf	TeX/MF (y LaTeX) librerías de información.
uucp	Comandos de UUCP.

`zoneinfo` Configuración e información de la zona horaria.

Históricamente, /usr/lib ha incluido además algunos comandos ejecutables tales como sendmail y makewhatis.

Dado que makewhatis no es referenciado por otros programas, no hay problemas al moverlo a un directorio binario. Dado que los usuarios tienen buena razón para usar makewhatis, /usr/lib es donde pertenece. El binario catman que reemplaza al script makewhatis en muchos sistemas **Linux**, debe también estar en usr/bin

El binario sendmail es referenciado por muchos programas con su nombre histórico /usr/lib/sendmail. Éste debe ser un enlace simbólico a la localización estándar para los agentes de transferencia de correo con una interfaz de línea de comando compatible con sendmail, /usr/bin/sendmail.

En sistemas que utilizan smail deben localizar smail en /usr/sbin/smail y /usr/bin/sendmail debe ser un enlace simbólico a smail.

Este arreglo también se conforma a la nueva localización estándar sendmail definida en Sendmail 8.6.x y BSD 4.4

Note que esta localización demanda que /usr/sbin y /usr/sbin/sendmail deben ser ejecutables para usuarios normales.

Cualquier paquete o programa que contenga o requiera información que no necesite ser modificada debe almacenar tal información en /usr/lib (ó /usr/local/lib, si está instalado localmente). Se recomienda la utilización de un subdirectorío en /usr/lib para este propósito.

La información de juegos almacenada en /usr/lib/games debe ser solamente información estática. Cualquier archivo modificable . tal como archivos de marcadores, registros de juego y similares, deben ser localizados en var/lib. Si es necesario para compatibilidad de juegos con el viejo estilo BSD, se puede usar un enlace simbólico desde /usr/games/lib hacia /usr/lib/games.

Nota: ninguna información específica de host para el sistema X window debe almacenarse en /usr/lib/X11 (que es realmente /usr/X11R6/lib/X11) . Los archivos de configuración específicos de host tales como Xconfig ó XF86Config deben almacenarse en /etc/X11. Éste debe incluir información de configuración como system.twmrc aún si es sólo un enlace simbólico a un archivo de configuración más global (tal vez en /usr/etc/X11 ó /usr/X11R6/lib/X11).

6.7 /usr/local: Jerarquía local

La jerarquía /usr/local está para ser utilizada por el administrador del sistema cuando se instale el software localmente. Necesita estar a salvo de ser sobrescrito cuando el software del sistema se actualiza. Puede ser usado por programas y por información que son compartibles entre un grupo de máquinas, pero no se encuentran en /usr.

<code>/usr/local</code>	Jerarquía local.
<code>bin</code>	Binarios sólo-locales
<code>doc</code>	Documentación local
<code>etc</code>	Binarios de configuración sólo-local
<code>games</code>	Juegos instalados localmente
<code>lib</code>	Librerías para /usr/local
<code>info</code>	Páginas de info local
<code>man</code>	Jerarquías de páginas de manual para /usr/local
<code>sbin</code>	Administración del sistema sólo-local
<code>scr</code>	Código fuente local.

Este directorio debe estar vacío al terminar de instalar **Linux** por primera vez. No debe haber excepciones a la regla, excepto quizá los subdirectorios vacíos listados.

El software instalado localmente debe estar localizado dentro de /usr/local, en vez de /usr a menos que este siendo instalado para reemplazar o actualizar el software en /usr.

Note que el software localizado en / o en /usr puede ser sobrescrito por actualizaciones del sistema (aunque recomendamos que las distribuciones no sobrescriban informacion en /etc bajo estas circunstancias). Por esta razón, el software local no se debe localizar fuera de /usr/local sin una buena causa.

6.8 /usr/man : Páginas del manual.

Esta sección detalla la organización de las páginas del manual a través del sistema. Incluyendo /usr/man.

Las páginas del manual están almacenadas <mandir>/<locale>/man [1-9]. Más delante se dá una explicación de <mandir> y <locale>.

```
<mandir>/<locale> --- Una jerarquía de páginas de manuales.
```

man1	Programas para usuarios.
man2	Llamadas al sistema.
man3	Subrutinas y funciones de librería
man4	Dispositivos.
man5	Formato de archivos
man6	Juegos.
man7	Misceláneas.
man8	Administración del Sistema.
man9	Funciones y variables internas del kernel.

El <mandir> primario del sistema es /usr/man contiene información del manual para comandos e información bajo los sistemas de archivos / y /usr. Obviamente no hay páginas de manual en / por que no se necesitan para arrancar ni en emergencias.

Se debe hacer provisión en la estructura de /usr/man para el soporte de páginas del manual que estén escritas en diferentes (o multiples idiomas). Estas provisiones deben tomar en cuenta el almencenamiento y referencia de estas páginas del manual. Los factores relevantes incluyen el idioma (incluyendo diferencias basadas en la geografía) y el código del conjunto de caracteres.

Esta nomenclatura de los subdirectorios de idiomas de /usr/man está basada en el apendice E del estándar POSIX 1003.1 que describe la cadena de identificación locale —El metodo más aceptado para describir un ambiente cultural. La cadena <locale>es:

```
<idioma>[<_territorio>][.<conjunto_de_caracteres>][,<version>]
```

El campo <idioma> se tomará del ISO639 (un código para la representacion de los nombres de los idiomas). Sera de dos caracteres de ancho y especificado con minúsculas solamente.

El campo <_territorio> sera el código de dos letras de ISO3116 (una especificación de la representación de los nombres de los países), si es posible. (mucha gente está familiarizada con el código de 2 letras empleado en el código de país en las direcciones de e-mail.

El campo <conjunto_de_caracteres> debe representar el estándar que describe el código de caracteres. Si el campo <conjunto_de_caracteres> es sólo una especificación numérica, el número representa el número del estándar internacional que describe a ese conjunto de caracteres. Se recomienda que este sea una representacion numérica siempre que sea posible (especialmente, estándares ISO), que no incluya símbolos de puntuación y que todas las letras sean minúsculas.

Un parámetro que especifique <version> del perfil puede ser colocada despues del campo <conjunto_de_caracteres >, delimitado con una coma. Ésto puede utilizarse para discriminar entre diferentes necesidades culturales, por ejemplo un orden de diccionario en vez de un orden de acomodo más

orientado hacia el sistema. Este estándar recomienda no usar el campo `<version>` a menos que sea necesario.

En sistemas que usen sólo un idioma y un código de conjunto de caracteres para todas las páginas del manual, pueden omitir la subcadena `<locale>` y almacenar todas las páginas del manual en `<mandir>`. Por ejemplo en sistemas que sólo tienen páginas del manual en inglés codificados en ASCII, pueden almacenar las páginas del manual (los directorios `man[1-9]`) directamente en `/usr/man` (éstas son las circunstancias y el arreglo tradicional, de hecho).

En países para los cuales existe un código de conjunto de caracteres estándar, puede omitir el campo `<conjunto_de_caracteres>`, pero recomendamos fuertemente que se incluya, especialmente para países con varios estándares en competencia.

Varios ejemplos:

Idioma	Territorio	Conjunto de caracteres	Directorio
Inglés	-----	ASCII	<code>/usr/man/en</code>
Inglés	Reino Unido	ASCII	<code>/usr/man/en_GB</code>
Inglés	Estados Unidos	ASCII	<code>/usr/man/en_US</code>
Francés	Canadá	ISO8859-1	<code>/usr/man/fr_CA</code>
Francés	Francia	ISO8859-1	<code>/usr/man/fr_FR</code>
Alemán	Alemania	ISO646-DE	<code>/usr/man/de_DE646de</code>
Alemán	Alemania	ISO6937	<code>/usr/man/de_DE6937</code>
Alemán	Alemania	ISO8859-1	<code>/usr/man/de_DE.88591</code>
Alemán	Suiza	ISO646-CH	<code>/usr/man/de_CH.646ch</code>
Japonés	Japón	JIS	<code>/usr/man/ja_JP.jis</code>
Japonés	Japón	SJCS	<code>/usr/man/ja_JP.sjis</code>
Japonés	Japón	UJ (o EUC-J)	<code>/usr/man/ja_JP.ujis</code>

Las páginas del manual para los comandos e información que se encuentra bajo `/usr/local` están almacenadas en `/usr/local/man`. las páginas del manual para el sistema X Window están almacenadas en `/usr/X11R6/man`. Luego todas las jerarquías de páginas del manual en el sistema deben tener la misma estructura que `/usr/man`. Los directorios vacíos pueden ser omitidos de la jerarquía de páginas del manual. Por ejemplo si, `/usr/local/man` no tiene páginas del manual en la sección 4 (dispositivos) entonces se puede omitir `/usr/local/man/man4`.

Las secciones de páginas `cat` (`cat[1-9]`) que contiene páginas del manual formateadas, también se encuentran dentro de los subdirectorios `/<mandir>/<locale>`, pero no son requeridas ni deben ser distribuidas en el lugar de las fuentes `nroff` de las páginas del manual.

Las páginas del Manual del sistema de manejo de correo `mh` deben tener el prefijo `mh` en todos los nombres de archivos de las páginas.

Las páginas del sistema X Window deben tener el prefijo `x` en todos los nombres de los archivos de las páginas.

La práctica de colocar las páginas del manual de diferentes idiomas, en los subdirectorios apropiados de `/usr/man` también se aplica a las otras jerarquías de páginas del manual, tales como `/usr/local/man`, y `/usr/X11R6/man`.

(Esta porción de este estándar también se aplica más adelante en la estructura opcional de `/var/catman`).

A continuación una descripción de cada sección.

6.8.1 man1: Programas de usuario.

Las páginas del manual que describen los comandos accesibles públicamente se encuentran en esta sección. La mayoría de la documentación de los programas que un usuario necesitará se encuentra aquí.

6.8.2 man2: Llamadas al sistema.

Esta sección describe todas las llamadas al sistema (requisiciones hacia el kernel de **Linux** para realizar ciertas operaciones).

6.8.3 man3: Subrutinas y funciones de librería.

La sección 3 describe programas rutinas de librería que no son llamadas directas al servicios del kernel. Esta sección y la 2 son de interés casi solamente para programadores.

6.8.4 man4: Archivos especiales.

Esta sección describe los archivos especiales, funciones relacionadas con los manejadores y el soporte para la red que estén disponibles en el sistema. Típicamente, esto incluye los archivos de dispositivo que se encuentran en /dev y la interfaz del kernel para el soporte de protocolos de red.

6.8.5 man5: Formatos de archivos.

Aquí se encuentran los formatos para muchos de los archivos cuyo formato no sea intuitivo. Ésto incluye varios archivos incluye, archivos de salida de programas, y archivos de sistema.

6.8.6 man6: Juegos.(Binarios Educativos)

Esta sección documenta los juegos,demos y programas triviales. Muchas personas tienen una opinión muy diferente de que tan esencial es esta sección.

6.8.7 man7: Misceláneos

Las páginas del manual que son difíciles de clasificar se designan como pertenecientes a la sección 7. Las de troff y otros macro paquetes de procesamiento de texto se encuentran aquí.

6.8.8 man8: Administración del Sistema

Aquí se documentan los programas utilizados por los administradores de sistemas para la operación y mantenimiento. Algunos de estos programas son ocasionalmente útiles para usuarios normales.

6.8.9 man9: Funciones y variables internas del kernel

Éste es utilizado para documentar el código fuente del kernel en los Sistemas **Linux**.

6.9 /usr/sbin: Binarios de sistema estándar no-esenciales.

Este directorio contiene cualesquier binario no-esencial utilizando exclusivamente por el administrador del sistema.

Los programas de administración del sistema que sean requeridos para la reparación del sistema, recuperación del sistema, montaje de /usr u otras funciones esenciales deben localizarse en /sbin en vez de aquí.

Típicamente /usr/sbin contiene los daemons de red, cualquier herramienta de administración no-esencial y binarios para programas servidores no-críticos. Ésto incluye los daemons de internet que son llamados por `inetd` (llamados `in.*`) tales como `in.telnetd` y `in.fingerd` y los daemons basados en `rpc` manejados por `portmap` (llamados `rpc.*`) tales como `rpc.infsd` y `rpc.mountd`.

Estos programas servidores son utilizados cuando se entra en un estado que el System V conoce como "run level 2" (estado multi-usuario) y el "run level 3" (estado en-red) o el estado que el BSD conoce como "modo multi-usuario". En este punto se hacen disponibles los servicios para los usuarios (p. ej. soporte de impresión) y hacia otras máquinas (p. ej. exportar NFS).

Los programas administrativos instalados localmente deben estar localizados en : /usr/local/sbin.

6.10 /usr/share : Información Independiente de Arquitectura.

Cualquier especificación para /usr/share se incluirá en un documento suplementario al **FSSTND**. Note que es la opinión en consenso del **FSSTND** que /usr/share no es necesario en la mayoría de los sistemas **Linux**. En este momento, si nos confinamos a proporcionar una definición extensiva de este directorio, sería una mala idea. Por favor refiérase a la sección 6 para una discusión más de /usr/share.

6.11 /usr/src : Código fuente.

```
/usr/src --- Código fuente
      linux           Código fuente para el kernel de Linux.
```

Cualquier código fuente no-local debe localizarse en este directorio. El único código fuente que siempre debe localizarse en un lugar específico es el código del kernel (cuando exista o esté enlazado como parte de una estructura en /usr/include). Se pueden usar subdirectorios si se desea.

El código fuente para el kernel debe siempre estar en su lugar o al menos los archivos include del código del kernel. Esos archivos están localizados en estos directorios.

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

`usr/include` debe contener enlaces a estos directorios, llamados `asm` y / **Linux**, dados que son necesitados por el compilador de C, al menos /estos archivos include deben siempre ser distribuidos en las instalaciones /que incluye un compilador C. Deben ser distribuidos en el directorio //usr/src/linux de forma que no existan problemas cuando los administradores /del sistema actualicen su versión del kernel por primera vez.

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

7 La Jerarquía /var

```
/var      Información variable

adm       Info administrativa del sistema (obsoleto). Enlace simbólico hacia /var/log
catman    Páginas del manual formateadas localmente
```

lib	Información del estado de aplicaciones
local	Información variable del software de /usr/local
lock	Archivos de bloqueo
log	Archivos de bitácora
named	Archivos DNS, sólo red
nis	Archivos base de datos NIS
preserve	Archivos almacenados después de una falla de ex ó vi
run	Archivos relevantes a procesos ejecutándose
spool	Directorios de trabajos en fila para realizarse después
tmp	Archivos temporales, utilizado para mantener /tmp pequeño

var contiene archivos con información variable. Ésto incluye archivos y /directorios en fila de ejecución, información de bitácora administrativa y /archivos temporales y transitorios.

Algunas porciones de /var son no-compartibles entre diferentes sistemas. Por ejemplo, /var/log, /var/lock y /var/run. Otras porciones son compartibles, notablemente /var/spool/mail y /var/spool/news.

var se especifica aquí para hacer posible el montar /usr sólo-lectura. /Todo aquello que alguna vez fué en /usr que es escrito durante la operación /normal del sistema (pero no durante la instalación y el mantenimiento del /software) debe ir en /var.

Si /var no puede ser una participación separada, es preferible mover /var fuera de la participación raíz pero dentro de la partición /usr (esto se hace algunas veces para reducir el tamaño de la partición raíz o cuando hay poco espacio en la partición raíz). Como sea, /var no debe ser enlazada a /usr, porque hace que la separación entre /usr y /var sea más difícil y seguramente creará un conflicto de nombres, En vez enlace /var a /usr/var.

7.1 /var/adm : Bitácora del sistema y archivos contables (obsoleto)

Este directorio ha sido remplazado por /var/log y otros directorios. Debe ser un enlace simbólico a /var/log hasta que todos los programas ya no se refieran más a algún archivo en /var/adm.

utmp se ha movido a /var/run. Todos los archivos bitácoras se han movido a /var/log incluyen en el archivo wtmp.

El soporte de empaquetado de distribuciones se debe almacenar en /var/lib/ <nombre> .

Nota: El enlace simbólico /var/adm no debe ser necesario en la mayoría de los sistemas Linux-i386ELF dado que el cambio fué introducido antes que ELF fuera liberado al público.

7.2 /var/catman : Páginas del Manual Formateadas localmente (opcional)

Este directorio proporcionara una localización estándar para los sites que proporcionan una partición /usr sólo-lectura, pero que desean permitir el almacenamiento temporal de páginas del manual formateados localmente. Los sites que montan /usr como escribible (p. pj. instalaciones mono-usuarios) pueden escoger no usar /var/catman y escribir las páginas del manual formateadas dentro de los directorios cat[1-9] dentro de /usr directamente. Recomendamos que la mayoría de los sites utilicen una de las siguientes opciones en su lugar.

Preformateé todas las páginas del manual dentro de /usr con el programa (catman).

: No se permita el almacenamiento temporal de las páginas formateadas del manual y requiera que se ejecute nroff cada vez que se necesite una página.

Se permita el almacenamiento temporal local de las páginas del manual en /var/catman.

La estructura de /var/catman necesita reflejar ambos, el hecho de la existencia de múltiples jerarquías de página del manual y la posibilidad del uso de múltiples idiomas.

Dada una página del manual sin formatear que normalmente aparece en /usr/<ruta1>/man/man[1-9], la versión formateada almacenada temporal debe ir en /var/catman/<ruta2>/cat[1-9], donde <ruta2> es <ruta1>. Los componentes <ruta2> y <ruta1> están ausentes en el caso de /usr/man y /var/catman.

Por ejemplo, /usr/man/man1/ls.1 se formatea en /var/catman/cat1/ls.1 y /usr/X11R6/man/<locale>/man3/XtClass.3x lo hace en /var/catman/X11R6/<locale>/cat3/XtClass.3x.

Las páginas del manual escritas en /var/catman/cat[1-9] pueden eventualmente, transferirse a /usr/<ruta>/cat[1-9] ó expirarse. De igual forma las páginas del manual formateadas dentro de /usr/<ruta>/cat[1-9] pueden expirarse si no son accesadas en un periodo de tiempo.

Si vienen páginas del manual preformateadas con un sistema **Linux** en un medio sólo-lectura. (p. ej. un COROM), deben estar instaladas en /usr/<ruta>/cat[1-9]. /var/catman está reservado como un lugar de almacenamiento temporal para páginas de manual formateados.

7.3 /var/lib : Información de Estado de Aplicaciones.

`/var/lib.- Información de Estado de Aplicaciones`

<code>emacs</code>	Directorio del estado de Emacs
<code>games</code>	Información variable de juegos(archivos de marcadores)
<code>news</code>	Archivos variables de Cnews/INN
<code>texmf</code>	Información variable asociada con TeX
<code>xdm</code>	Archivos de autenticación y bitácoras de error del manejador de despliegues X

`var/lib/<nombre>` es el lugar apropiado para el soporte de /empaquetamiento de todas las distribuciones. Diferentes distribuciones de **Linux** pueden utilizar diferentes nombres por supuesto.

7.3.1 /var/lib/emacs

El directorio del estado **GNU Emacs**, el lugar donde los archivos de información independiente de la arquitectura, que **Emacs** modifica cuando se ejecuta, debe ser /var/lib. En el presente, **Emacs** sólo localiza su directorio de archivos de bloqueo bajo el directorio de estado (en <direstado>/emacs/lock), pero puede hacer uso más extenso del mismo en el futuro. Notablemente, sólo se requiere la adición de una opción sencilla en el programa configure de **Emacs** para hacer este cambio (antes de compilar).

7.3.2 /var/lib/games

Así como los subdirectorios antes citados, cualquier información variable relacionada con los juegos que se encuentran en /usr/games, deben estar aquí. /var/lib/games debe incluir la información variable que previamente se encontraba en /usr/lib/games; La información estática, tal como textos de ayuda, descripciones del nivel y demás, debe permanecer en /usr/lib/games.

7.3.3 /var/lib/news

`var/lib/news` se debe usar para almacenar toda la información variable /asociada con los servidores de news tales como Cnews y INN, inclusive el /archivo histórico, el archivo activo y demás.

7.3.4 /var/lib/texmf

`/var/lib/texmf` se debe usar para almacenar la información variable /asociada con **TeX**. Particularmente, en `/var/lib/texmf/fonts` se /almacenarán todas las fuentes tipográficas que son generadas /automáticamente por `MakeTeXPK`.

Debe haber un enlace desde `/usr/lib/texmf/fonts/tmp` hacia `/usr/lib/texmf/fonts`. Este enlace permite a los usuarios hacer uso de una sola ruta `/usr/lib/texmf/fonts/tfm` cuando le hacen cambios a su variable de entorno `TEXFONTS` (ésta es la ruta por defecto en las herramientas **TeX** de Karl Berry distribuidas desde `ftp.cs.umb.edu/pub/tex` [La razón de mencionarlos aquí es que son el estándar de facto en las instalaciones **UNIX**, estas herramientas son ampliamente usadas entre la comunidad **Linux**]. Si se utiliza otra distribución de **TeX**, se debe hacer un enlace desde el directorio de fuentes apropiado hacia `/usr/lib/texmf/fonts`).

El `MakeTeXPK` que se distribuye con `dvipsk` colocará los archivos `.pk` en `fonts/pk/<dispositivo>/<nombre_de_la_fuente>`, (p.ej. `fonts/pk/Canon_CX/cmr10.300pk`). Los archivos `.pk` se pueden purgar periódicamente del árbol `/var/lib/texmf` o se puede mover dentro del árbol `/usr/lib/texmf`. Si se usan generadores automáticos de `.mf` ó `.tfm`, éstos deben poner su información en los subdirectorios `mf` ó `tfm` de `/var/lib/texmf/fonts`.

7.3.5 /var/lib/xdm

`/var/lib/xdm` contiene la información variable de `xdm` que consiste en los /archivos `xdm-errors` y cualquier archivo de autoridad `xdm`. Los binarios de `xdm` tales como `chooser` deben aún estar localizados en la localidad /histórica en `/usr/X11R6/lib/X11/xdm`. El archivo `xdm-pid` debe estar en `//var/lib/xdm` a pesar de existir `/var/run`. Los archivos restantes deben /estar en `/etc/X11/xdm`.

7.4 /var/local : Información variable del software que está en /usr/local

Este directorio contiene toda la información variable que esté relacionada con el software que se encuentra en `/usr/local`. Naturalmente la implementación de este subdirectorio se deja a el administrador del sistema . Como sea la información que se puede categorizar en otro lugar del directorio `/var`, no se debe colocar en `/var/local`. Por ejemplo, todos los archivos de bloqueo aún irán en `/var/lock`.

7.5 /var/lock : Archivos de Bloqueo

Los archivos de bloqueo deben almacenarse dentro de una estructura del directorio de `/var/lock`.

Para preservar la habilidad de montar `/usr` sólo-lectura, no se deberá colocar los archivos de bloqueo en la partición `/usr`.

Los archivos de bloqueo de dispositivo, tales como los archivos de bloqueo de dispositivos serie que antes se encontraban en `/usr/spool/lock` ó en `/usr/spool/uucp` deben ahora almacenarse en `/var/lock`. la convención para la nomenclatura que debe utilizarse es `LCK...` seguido del nombre base del dispositivo. Por ejemplo, para bloquear `/dev/cua0` se deberá crear el archivo `LCK... cua0`.

El formato usado para los archivos de bloqueo de dispositivo en **Linux** deberá ser el formato de archivos de bloqueo `HDB UUCP`. El formato `HDB` es almacenar el `PID` (Identificador del proceso) como un número decimal en `ASCII` de 10 bytes, con un caracter de línea nueva.

Por ejemplo, si el proceso 1230 retiene un archivo de bloqueo, contendrá los siguientes once (11) caracteres: espacio, espacio, espacio, espacio, espacio, espacio, uno, dos, tres, cero y nueva línea.

Entonces cualquier cosa que desee usar `/dev/cua0`, puede leer el archivo de bloqueo y actuar de acuerdo (Todos los archivos de bloqueo en `/var/lock` deben ser leíbles por todos).

7.6 /var/log : Archivos bitácora y directorios

Este directorio contiene archivos bitácora misceláneos. la mayoría de los archivos bitácora se deben escribir en este directorio o subdirectorios apropiados.

lastlog	Registro del último acceso de cada usuario
messages	Mensajes del sistema desde syslogd
wtmp	Registro de todos los accesos y salidas

Se puede requerir de un enlace simbólico desde /var/log/utmp hacia /var/run/utmp hasta que ningún programa se refiera a /var/adm/utmp (/var/adm es en sí mismo un enlace simbólico transicional hacia /var/log).

7.7 /var/named : Archivos DNS

Este directorio contiene todos los archivos de trabajo del servidor de nombres Internet, named.

: Recomendamos que /etc/named.boot sea un enlace simbólico hacia /var/named/named.boot, dado que /etc/named.boot es el archivo de arranque por defecto, si no se dan argumentos a named.

7.8 /var/nis : Archivos de bases de datos del servicio de información de red (NIS)

El sistema de información de red (NIS) era anteriormente conocido como las Páginas Amarillas Sun (YP). La funcionalidad y localización de directorios de ambos es el mismo pero el nombre (Yellow Pages) es una marca registrada en el Reino Unido, pertenece a British Telecommunications PLC. y no puede ser usada sin permiso.

7.9 /var/preserve:Archivos guardados después de una colisión o una terminación inesperada de ex ó vi

Este directorio contiene los archivos que son almacenados ante cualquier terminación no-esperada de ex, vi, ó de alguno de sus clones.

7.10 /var/run : Archivos variables de tiempo de ejecución

Este directorio contiene archivos con información del sistema que lo describen desde que arrancó. Generalmente los archivos en este directorio se deben limpiar (remove o truncar, según corresponda) al comenzar el proceso de arranque.

Los archivos identificados de proceso (PID), que estaban originalmente /etc, se deben colocar en /var/run. La convención de nomenclatura de archivos PID es <nombre-programa>.pid, por ejemplo el archivo PID de crond se llama /var/run/crond.pid.

El formato interno de los archivos PID permanecen sin cambio. El archivo debe de consistir del indicador del proceso en decimales codificado como ASCII, seguido por un caracter nueva línea. Por ejemplo, si crond fue el proceso número 25, /var/run/cond.pid contendrá 3 caracteres, dos cinco y nueva línea.

Los programas que léan archivos PID deben ser flexibles en lo que aceptan, p. ej. deben ignorar los espacios extras, ceros a la izquierda, ausencia del caracter nueva línea o líneas adicionales en el archivo PID. Los programas que crean archivos PID deben utilizar la sencilla especificación dada en el anterior párrafo.

El archivo utmp, que almacena información acerca de quién está actualmente utilizando el sistema, se localiza en este subdirectorio.

Los programas que mantengan sockets transitorios de dominio **UNIX**, deben colocarlos en este directorio.

7.11 /var/spool : Directorios de fila de trabajos para procesamiento posterior

var/spool es tradicionalmente utilizado para la información local de /máquina que es enviada para procesarse después, hacia o desde subsistemas /**UNIX**. Por ejemplo, trabajos de impresión que son almacenados aquí para /entrega posterior al daemon de la impresora, el correo que sale es /almacenado aquí para entrega a sistemas remotos y los archivos UUCP son /almacenados aquí para transmisión a los sistemas UUCP vecinos. El correo /que entra y las noticias son almacenados aquí para entregarse a los /usuarios y los trabajos de at y cron son almacenados aquí para ejecución /retardada por el daemon cron.

/var/spool	
at	Trabajos de at
cron	Trabajos de cron
lpd	Directorio de impresora *
mail	Archivos caja-postal(buzón) de los usuarios
mqueue	Fila del correo saliente
news	Directorio de noticias *
rwhod	Archivos rwhod
smail	Directorio de smail *
uucp	Directorio de UUCP

Nota: * Significa fila de trabajos para proceso posterior.

Los archivos de bloqueo UUCP deben localizarse en /var/lock. Véa la sección acerca de /var/lock.

7.11.1 /var/spool/lpd

```
/var/spool/lpd --- Directorio de fila de trabajos para proceso posterior o impresión
<impresora>      Directorio que tiene la fila específica de esta impresora
```

El archivo de bloqueo para lpd, lpd.lock debe estar localizado en /var/spool/lpd. El archivo de bloqueo de cada impresora debe localizarse en el directorio <impresora> de la impresora específica y se debe llamar lock.

7.12 /var/tmp : Archivos temporales, utilizando para mantener /tmp pequeño.

Los archivos que están en /var/tmp están almacenados por una duración no específica. (Recuerde que los directorios temporales del sistema no garantizan mantener la información por ningún periodo particular).

La información almacenada en /var/tmp típicamente se limpia en una "forma definida localmente" pero usualmente menos frecuentemente que /tmp. Se puede encontrar información sobre directorios temporales en la selección dedicada a /tmp (arriba).

Debe existir un enlace simbólico desde /usr/tmp hacia var/tmp por razones de compatibilidad.

8 Razonamientos adicionales y asuntos sin resolver

Esta sección discute varias áreas que pueden requerir mayor explicación.

8.1 ¿Qué es esencial?

La respuesta es: esencial para limpiar, crear, preparar, verificar, encontrar y montar otros sistemas de archivos (posiblemente en máquinas remotas). Hay otras definiciones, pero ésta es una definición general que la mayoría de las personas al menos la incorporará en la suya.

8.2 Red

La red presenta un dilema interesante, algunas personas quieren separar los binarios y de configuración de la red de los que no lo son. Como sea, estamos en desacuerdo. Sentimos que la red no es un "paquete", sino una parte integral de la mayoría de la máquinas **UNIX** (y similares). Debido a lo anterior, la red no debe colocarse en un sólo directorio sino localizarse sistemáticamente en los directorios apropiados.

8.2.1 `/bin` :Cualquier cosa que algún usuario querrá utilizar y que es también considerado vital.

hostname, netstat, ping

8.2.2 `/sbin` :Cualquier cosa que sólo root necesita y se considere vital.

arp, ifconfig, route

8.2.3 `/usr/bin` : Algunos binarios que algún usuario querrá utilizar y que no son vitales.

finger, rep, rlogin, telnet, etc.

8.2.4 `/usr/sbin` : Algunos binarios sólo para el administrador que no son vitales.

in.ftpd, inetd, lpd, portmap, etc.

Aunque puede parecer confuso al principio (y toma tiempo digerirlo), tiene sentido. Si por alguna razón usted sólo puede montar la partición raíz, y necesita acceder a la red para reparar su sistema, no se quiere que los archivos estén en `/usr/etc` (como están algunas veces). Los archivos que se necesitan para montar `/usr` en las situaciones normales (y de emergencia) están colocados dentro del sub-árbol raíz, y cualesquier otros se colocan en `/usr`, para mantener el tamaño del sistema de archivos raíz pequeño.

Los archivos de configuración para la red pertenecen a `/etc`.

8.3 Estructuras independientes de la arquitectura

El directorio `/usr/share` típicamente contiene archivos independientes de la arquitectura, tales como páginas del manual, zona horaria, información de término, etc. En el momento presente no hay diferentes arquitecturas para **Linux**, pero con el tiempo, veremos que **Linux** incluirá otras arquitecturas y otros sistemas similares a **UNIX**.

Nota: Ningún programa nunca deberá hacer referencia a alguna cosa en `/usr/share`. Por ejemplo, un programa de páginas del manual no debe nunca buscar directamente `/usr/share/man/man1/ls.1`, sino que siempre se debe referir a `/usr/man/man1/ls.1`. Cualquier cosa en `/usr/share`, será "apuntada" a través de uso de enlaces símbolos desde otras áreas del sistema de archivos, tales como `/usr/man`, `/usr/lib/ <algo>`, etc... Aún se trabaja en las especificaciones de `/usr/share`.

8.4 Enlaces Simbólicos.

Hay muchos usos para los enlaces simbólicos en cada sistemas de archivos. Aunque un estándar como éste no respalda el uso de enlaces simbólicos en la implementación por defecto (Los encontrados después de instalar **Linux**), se usan frecuentemente con buenos propósitos en diferentes sistemas. El punto es que los enlaces simbólicos deben estar allí para mantener todos los archivos y directorios donde cada quien los espera encontrar.

Está preparado para aceptar que ciertos directorios, aún aquellos contenidos en el directorio raíz, aún sean enlaces simbólicos. Por ejemplo en algunos sistemas /home no estará en el raíz, sino enlazado simbólicamente a un directorio /var o algún otro lugar. /home podría tener también su propia partición física y desde luego, ser montado como tal.

Similarmente, dado que /usr podría estar en un servidor de archivos central montado vía NFS, /usr/local se puede enlazar simbólicamente a /var/local. Este cambio se puede justificar recordando la razón principal de tener /var: separar directorios de archivos que varían con el tiempo y entre diferentes sistemas y máquinas de aquellos que se pueden compartir y sean sólo-lectura.

Algunos sistemas además enlazarán /tmp a /var/<algo> si la partición raíz se vuelve muy pequeña (ó es muy pequeña). Hay más ejemplos de "buenos" usos de enlaces simbólicos, pero todo el asunto se reduce a dos cosas: Los paquetes deben ser capaces de encontrar las cosas donde lo esperan (razonablemente) y los enlaces simbólicos se pueden utilizar para resolver los problemas de muchos casos. Como sea, se pueden generar problemas con el uso de demasiados enlaces simbólicos. este problema incluye sobre-confianza en los enlaces simbólicos para resolver problemas, confusión resultante del sobre-uso de enlaces simbólicos y las preferencias estéticas de diferentes personas.

8.5 Binarios Compilados Estáticamente.

Linux se ejecuta corre actualmente en una gama de sistemas, algunos con sólo un usuario y disco pequeño, otros como servidores en ambientes con redes muy grandes, dada esta variedad, este estándar no impone regla sobre qué binarios están compilados estáticamente o dinámicamente, con las siguientes excepciones. Ambos ln y sync, deben existir en /bin; cualquier versión estática se puede colocar en /sbin o remplazar aquellas en /bin.

Los grandes sistemas **Linux** pueden desear incluir otros binarios estáticos (sh, init, mkfs, fsch, tuneefs, mount, umount, swapon, swopff, getty, login y otros). Los desarrolladores y los administradores de sistemas, son libres de enlazar dinámica o estáticamente éstos y otros binarios según le convengan, siempre que la localización de los binarios no cambie.

En sistemas de red, (especialmente aquellos que no tienen unidades de disco flexible), pueden querer compilar estáticamente ifconfig, route, hostname y otras herramientas de red. Ésto usualmente no es necesario.

9 La lista del correo del FSSTND

La lista del correo del **FSSTND** se encuentra en <Linux-fsstnd@ucsd.edu>. Esta lista estaba originalmente localizada en <Linux-activists@niksula.hut.fi> "Mail Net" como el canal **FSSTND**. (Para subscribirse a la lista mande correo <listserv@ucsd.edu> con el mensaje "ADD Linux-fsstnd").

Gracias a Operaciones de Red en la Universidad de California en San Diego quien nos permitió utilizar su excelente servidor de listas de correo.

10 Reconocimientos:

Se debe dar crédito por este texto a las activistas **FSSTND**, desarrolladores, administradores de sistemas y usuarios cuyos comentarios fueron esenciales a este estándar. También deseo agradecer a cada contribuyente que ayudó a escribir, compilar y componer éste, un estándar de consenso.

Deseo también dar real crédito a aquellos desarrolladores que han visto que el dar a **Linux** un arreglo común del sistema de archivos es algo que expandirá el desarrollo del sistema operativo **Linux**. Deseo también la valentía y perseverancia de aquellos desarrolladores de **Linux** quienes comenzaron a seguir este estándar aún antes de que fuera completado.

Los que contribuyeron originalmente:

Drew Eckhard	(US)	drew@colorado.edu
Brondon S. Allbery	(US)	bsa@kf8nh.wariat.org
Ian Jackson	(UK)	ijackson@cus.cam.ac.uk
Rik Faith	(US)	faith@cs.unc.edu
Iar Mc Cloghrie	(US)	ian@ucsd.edu
Stephen Harris	(UK)	sweh@spuddy.mew.co.uk
Daniel Quilan	(US)	Daniel.Quinlan@linux.org
Fred N. van Kemper	(US)	waltje@infomagic.com
Mike Sangrey	(US)	mike@sojurn.lns.pa.us
Jhon A. Martir	(US)	jmartin@csc.com
David H. Silber	(US)	dhs@glowworm.firefly.com
Chris Netcalf	(US)	metcalf@lcs.mit.edu
Theodore Tsó	(US)	tytso@athena.mit.edu
Ian Murdock	(US)	imurdock@debian.org
Stephen Tweedie	(UK)	sct@dcs.ed.ac.uk
David C. Niemi	(US)	niemid@clarck.net