Linux From Scratch

Linux From Scratch

Versión 3.3

Gerard Beekmans

Copyright © 1999−2002 por Gerard Beekmans

Traducido por el proyecto LFS−ES http://www.escomposlinux.org/lfs−es

Versión de la traducción: FINAL del 4 de Julio del 2002

Este libro describe el proceso para la creación de un sistema Linux desde cero a partir de una distribución Linux actualmente instalada, usando solamente las fuentes de los programas que vayamos a usar.

Copyright (c) 1999−2002, Gerard Beekmans

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions in any form must retain the above copyright notice, this list of conditions and the following disclaimer.
- Neither the name of "Linux From Scratch" nor the names of its contributors may be used to endorse or promote products derived from this material without specific prior written permission.
- Any material derived from Linux From Scratch must contain a reference to the "Linux From Scratch" project.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Dedicatoria

Este libro está dedicado a la querida esposa de Gerard Beekmans, *Beverly Beekmans*.

Tabla de contenidos

Prólogo
- *Prefacio*
- *Quién puede querer leer este libro*
- *A quién puede que no le interese leer el libro*
- *Organización*
  - *Parte I – Introducción*
  - *Parte II – Instalación del sistema LFS*
  - *Part III – Apéndices*

I. Parte I – Introducción
- *1. Introducción*
  - *Agradecimientos*
  - *Lo que vamos a hacer*
  - *Convenciones utilizadas en este libro*
  - *Versión del libro*
  - *Servidores alternativos*
  - *Historial de modificaciones*
  - *Listas de correo y archivos*
  - *Servidores de noticias*
  - *FAQ*
  - *Información de contacto*
- *2. Información importante*
  - *Sobre $LFS*
  - *Cómo descargar los programas*
  - *Cómo instalar los programas*
  - *¿Qué plataforma?*
  - *Cómo buscar ayuda*

II. Parte II – Instalando el sistema LFS
- *3. Paquetes que hay que descargar*
  - *Introducción*
  - *Paquetes que hay que descargar*
- *4. Preparando una nueva partición*
  - *Introducción*
  - *Creando una nueva partición*
  - *Creando un sistema de ficheros en la nueva partición*
  - *Montando la nueva partición*
- *5. Preparando el sistema LFS*
  - *Introducción*
  - *¿Por qué usamos enlazado estático?*
  - *Instalando todos los programas como usuario sin privilegios*
  - *Creando directorios*
  - *Instalando Bash−2.05a*
  - *Instalando Binutils−2.12*
  - *Instalando Bzip2−1.0.2*
  - *Instalando Diffutils−2.8*
  - *Instalando Fileutils−4.1*

Instalando Gawk−3.1.0
Instalando GCC−2.95.3
Instalando Grep−2.5
Instalando Gzip−1.2.4a
Instalando Linux Kernel−2.4.18
Instalando Make−3.79.1
Instalando Patch−2.5.4
Instalando Sed−3.02
Instalando Sh−utils−2.0
Instalando Tar−1.13
Instalando Texinfo−4.1
Instalando Textutils−2.0
Creando los ficheros de contraseñas y grupos
Copiando los ficheros de la antigua librería NSS
Montando el sistema de ficheros $LFS/proc
6. Instalando los programas del sistema base
Introducción
Sobre los símbolos de depuración
Creando $LFS/root/.bash_profile
Entrando al entorno chroot
Cambiando el propietario de la partición LFS
Creando el enlace simbólico /etc/mtab
Instalando Glibc−2.2.5
Creando los dispositivos (Makedev−1.4)
Instalando Man−pages−1.48
Instalando Findutils−4.1
Instalando Gawk−3.1.0
Instalando Ncurses−5.2
Instalando Vim−6.1
Instalando GCC−2.95.3
Instalando Bison−1.34
Instalando Less−374
Instalando Groff−1.17.2
Instalando Man−1.5j
Instalando Perl−5.6.1
Instalando M4−1.4
Instalando Texinfo−4.1
Instalando Autoconf−2.53
Instalando Automake−1.6
Instalando Bash−2.05a
Instalando Flex−2.5.4a
Instalando File−3.37
Instalando Libtool−1.4.2
Instalando Bin86−0.16.2
Instalando Binutils−2.12
Instalando Bzip2−1.0.2
Instalando Ed−0.2
Instalando Gettext−0.11.1
Instalando Kbd−1.06
Instalando Diffutils−2.8
Instalando E2fsprogs−1.27

Instalando Fileutils−4.1
Instalando Grep−2.5
Instalando Gzip−1.2.4a
Instalando Lilo−22.2
Instalando Make−3.79.1
Instalando Modutils−2.4.15
Instalando Netkit−base−0.17
Instalando Patch−2.5.4
Instalando Procinfo−18
Instalando Procps−2.0.7
Instalando Psmisc−20.2
Instalando Reiserfsprogs−3.x.1b
Instalando Sed−3.02
Instalando Sh−utils−2.0
Instalando Net−tools−1.60
Instalando Shadow−4.0.3
Instalando Sysklogd−1.4.1
Instalando Sysvinit−2.84
Instalando Tar−1.13
Instalando Textutils−2.0
Instalando Util−linux−2.11o
Instalando LFS−Bootscripts−1.9
Borrando los ficheros de la antigua librería NSS
Configurando los programas esenciales
7. Preparando los guiones de arranque
Introducción
¿Cómo hacen estos guiones que funcione el proceso de arranque?
Configurando el guión setclock
¿Necesito el guión loadkeys?
Configurando el guión sysklogd
Configurando el guión localnet
Creando el fichero /etc/hosts
Configurando el guión network
8. Haciendo el sistema LFS arrancable
Introducción
Creando el fichero /etc/fstab
Instalando linux−2.4.18
Haciendo el sistema LFS arrancable
9. El final
El final
Registrarse
Arrancando el sistema
III. Parte III − Apéndices
A. Descripción de paquetes y dependencias
Introducción
Autoconf
Automake
Bash
Bin86
Binutils
Bison

Bzip2
Diffutils
E2fsprogs
Ed
File
Fileutils
Findutils
Flex
Gawk
GCC
Gettext
Glibc
Grep
Groff
Gzip
Kbd
Linux kernel
Less
LFS−Bootscripts
Libtool
Lilo
M4
Make
MAKEDEV
Man
Man−pages
Modutils
Ncurses
Netkit−base
Net−tools
Patch
Perl
Procinfo
Procps
Psmisc
Reiserfsprogs
Sed
Entorno de Contraseñas Ocultas (Shadow Password Suite)
Sh−utils
Sysklogd
Sysvinit
Tar
Texinfo
Textutils
Util Linux
Vim

B. Recursos

Introducción
Libros
COMOs y Guías
Otros

Prólogo

Prefacio

Después de haber usado diferentes distribuciones de Linux, nunca estuve satisfecho con ninguna de ellas. No me gustaba la forma en la que estaban organizados los guiones de arranque, o no me gustaba la configuración por defecto de ciertos programas, y cosas de este estilo. Llegué a darme cuenta de que si quería estar completamente satisfecho con algún sistema Linux, tenía que construir el mío propio desde cero, usando, idealmente, sólo el código fuente. Sin utilizar paquetes precompilados de ninguna clase. Sin la ayuda de un CD−ROM de arranque que instalase utilidades básicas. Utilizaría mi sistema Linux actual para construir el mío por mi cuenta.

Esta, en su momento, idea descabellada se presentó muy difícil y algunas veces casi imposible. Después de sortear toda clase de problemas de dependencias, de compilación, etc., creé un sistema Linux hecho a medida y completamente funcional. Llamé a este sistema LFS, que significa Linux From Scratch (Linux desde Cero).

¡Espero que paséis buenos momentos trabajando en vuestro LFS!

––
Gerard Beekmans
gerard@linuxfromscratch.org

Quién puede querer leer este libro

Existen muchas razones por las que alguien podría querer leer este libro para instalar un sistema LFS. La pregunta que mucha gente podría hacer es "¿Por qué pasar por todo el embrollo de instalar manualmente un sistema desde cero cuando te puedes limitar a descargar una distribución ya existente?". Esta es una pregunta muy válida que espero poder contestar.

La razón más importante de la existencia de LFS es enseñar a la gente cómo trabaja internamente un sistema Linux. Construir el sistema LFS te enseña todo lo que hace a Linux funcionar, cómo trabajan juntas las distintas partes, y cómo unas dependen de otras. Y lo más importante, cómo adaptarlo a tus propios gustos y necesidades.

Uno de los beneficios claves de LFS es que tienes el control de tu sistema sin tener que confiar en la implementación de Linux de nadie. Ahora estás en el asiento del conductor y puedes dictar cada cosa, como la estructura de directorios y la configuración de los guiones de arranque. También sabrás exactamente dónde, por qué y cómo se instalan los programas.

Otro beneficio de LFS es que puedes crear un sistema Linux verdaderamente compacto. Cuando instalas una distribución normal, acabas instalando muchos programas que, probablemente, nunca usarás. Sólo están ahí gastando (precioso) espacio de disco. No es muy difícil conseguir un sistema LFS instalado en tan sólo 100 MB. ¿Todavía te parece demasiado? Algunos de nosotros hemos estado trabajando para crear un sistema LFS embebido realmente pequeño. Hemos instalado un sistema que contiene lo suficiente para ejecutar un servidor web Apache; el espacio total de disco usado fue, aproximadamente, 8 MB. Con un repaso adicional para reducirlo, se podría llegar a 5 MB o menos. Intenta eso con una distribución normal.

Podríamos comparar una distribución de Linux con una hamburguesa que compras en un supermercado o en un restaurante de comida rápida. Acabarías comiéndola sin saber exactamente qué estás comiendo, mientras que LFS te da los ingredientes para hacerte la hamburguesa. Esto te permite inspeccionarla cuidadosamente, quitar los ingredientes no deseados y, al mismo tiempo, te permite añadir los ingredientes que mejoren el sabor de tu hamburguesa. Cuando estés satisfecho con los ingredientes, continúas con el siguiente paso, mezclarlo todo. Ahora tienes la oportunidad de cocinarla a tu gusto: asarla, cocerla, freirla, hacerla a la barbacoa, o comerla cruda.

Otra posible analogía que podemos usar es comparar a LFS con una casa. LFS te dará la estructura de la casa, pero es cosa tuya instalar cañerías, enchufes eléctricos, la cocina, la bañera, el papel pintado, etc.

Otra ventaja de un sistema Linux hecho a la medida es la seguridad añadida. Compilar el sistema entero a partir del código fuente, te permite, si lo deseas, supervisar todo y aplicar todos los parches de seguridad que quieras o necesites. No tienes que esperar a que alguien te provea de un nuevo paquete binario que tape ese agujero de seguridad. Además, no tienes ninguna garantía de que ese nuevo paquete verdaderamente solucione el problema (adecuadamente). Nunca podrás saber realmente si un agujero de seguridad está solucionado a no ser que lo hagas por ti mismo.

A quién puede que no le interese leer el libro

La gente que no quiera construir un sistema Linux entero desde cero probablemente no quiera leer este libro. Sin embargo, si quieres aprender más sobre lo que sucede entre bastidores, en particular qué pasa desde que enciendes el ordenador hasta que ves la línea de comandos, puede que quieras leer el "From−PowerUp−To−Bash−Prompt−HOWTO ("De−La−Puesta−En−Marcha−Al−Indicador−Del−Bash−CÓMO")". Este CÓMO construye un sistema escueto, de una manera similar a la de este libro, pero lo enfoca más hacia instalar un sistema que arranque, en lugar de un sistema completo.

Para decidir si leer este libro o el "De−La−Puesta−En−Marcha−Al−Indicador−Del−Bash−CÓMO", pregúntate esto: "¿Es mi objetivo principal conseguir un sistema Linux funcional que voy a construir yo mismo y aprender, durante todo el trayecto, para qué es cada componente del sistema?. ¿O es sólo aprender mi principal objetivo?" Si quieres construirlo y aprender, lee este libro. Si sólo quieres aprender lo básico, entonces el "De−La−Puesta−En−Marcha−Al−Indicador−Del−Bash−CÓMO" sea, probablemente, mejor material de lectura.

Podrás encontrar el "De−La−Puesta−En−Marcha−Al−Indicador−Del−Bash−CÓMO" en http://personal.telefonica.terra.es/web/aus/linux/p2b/power2bash.html y el original "From−PowerUp−To−Bash−Prompt−HOWTO" en http://www.netspace.net.au/~gok/power2bash/.

Organización

Este libro está dividido en las partes que explicaremos a continuación. Aunque muchos de los apéndices están repetidos en la segunda parte (lo que, de alguna manera, alarga el libro), creemos que ésta es la forma más fácil de leerlo. Simplemente te ahorra tener que ir a un Apéndice, y después volver a dónde estabas en la parte II. Eso es una faena, especialmente si estás leyendo la versión en texto de este libro.

Parte I – Introducción

La Primera Parte te aporta información general sobre este libro (versiones, dónde conseguirlo, diario de

cambios, listas de correo, y cómo entrar en contacto con nosotros). También explica algunos aspectos importantes que querrás y que necesitas leer antes de empezar a construir un sistema LFS.

Parte II – Instalación del sistema LFS

La Segunda Parte te guía a través de la instalación del sistema LFS, que será la base para el resto del sistema. Cualquier cosa que decidas hacer con tu flamante sistema LFS, será construido sobre los cimientos que se instalan en esta parte.

Part III – Apéndices

La Tercera Parte contiene varios apéndices.

I. Parte I – Introducción

Tabla de contenidos

1. Introducción
2. Información importante

Capítulo 1. Introducción

Agradecimientos

Queremos agradecer sus contribuciones al proyecto Linux From Scratch a las siguientes personas y organizaciones:

- Mark Stone <mstone@linux.com> por donar el servidor linuxfromscratch.org .
- VA Linux Systems por proveer espacio de rack y ancho de banda al servidor linuxfromscratch.org .
- Hagen Herrschaft <hrx@hrxnet.de> por mantener los mirrors (servidores alternativos) de linuxfromscratch.org.
- Mark Hymers <markh@linuxfromscratch.org> por su enorme ayuda en la edición de este libro.
- Marc Heerdink <marc_heerdink@softhome.net> también por su gran ayuda en la edición de este libro.
- DREAMWVR.COM por su continuado respaldo donando varios recursos al proyecto LFS y a subproyectos relacionados.
- Jan Niemann <jan.niemann@tu.bs.de> por mantener el mirror www.de.linuxfromscratch.org.
- Torsten Westermann <westermann@linux−provider.net> por mantener el mirror lfs.linux−provider.net.
- Ian Chilton <ian@ichilton.co.uk> por mantener los mirrors www.us.linuxfromscratch.org y www.linuxfromscratch.co.uk .
- Dag Stenstad <dag@stenstad.net> por proveer el mirror www.no.linuxfromscratch.org mirror, y Ian Chilton <ian@ichilton.co.uk> por mantenerlo.
- Antonin Sprinzl <Antonin.Sprinzl@tuwien.ac.at> por mantener el mirror www.at.linuxfromscratch.org.
- Jason Andrade <jason@dstc.edu.au> por mantener el mirror www.au.linuxfromscratch.org.
- Ian Cooper <ian@wpi.edu> por mantener el mirror www.us2.linuxfromscratch.org.
- VA Linux Systems que, en nombre de Linux.com, donó al proyecto una estación de trabajo VA Linux 420 (anteriormente StartX SP2).
- Johan Lenglet <johan@linuxfromscratch.org> por gestionar el proyecto de traducción de LFS al francés.
- Jesse Tie−Ten−Quee <highos@linuxfromscratch.org> por donar una grabadora de CD Yamaha CDRW 8824E.
- O'Reilly por donar libros sobre SQL y PHP.
- Robert Briggs por donar los nombres de dominio linuxfromscratch.org y linuxfromscratch.com.
- Frank Skettino <bkenoah@oswd.org> de OSWD por crear el diseño inicial del sitio web LFS.
- Garrett LeSage <garrett@linux.com> por crear el banner de LFS.
- Dean Benson <dean@vipersoft.co.uk> por su ayuda financiera al establecer la organización sin ánimo de lucro LFS.
- Innumerables otros en varias listas de correo LFS que están haciendo posible este libro aportando sugerencias, leyendo el libro e informando de los errores.

Lo que vamos a hacer

Vamos a construir el sistema LFS utilizando una distribución ya instalada, como Debian, SuSe, Slackware, Mandrake, RedHat, etc. No es necesario ningún tipo de disco de arranque. Vamos a utilizar un sistema Linux existente como base (ya que necesitamos un compilador, enlazador (linker), editor de texto y otras herramientas).

Cuando hayas descargado los paquetes necesarios que componen un sistema LFS, crearás un nueva partición Linux nativa sobre la que se instalará el sistema LFS.

El siguiente paso, el capítulo 5, será la instalación de una serie de paquetes en la partición LFS, enlazados estáticamente. Estos paquetes constituyen un juego básico de herramientas de desarrollo, que se usarán para instalar el sistema propiamente dicho, y también son necesarios para resolver dependencias circulares. Ejemplos de dependencias circulares son: necesitas un compilador para instalar un compilador, necesitas un intérprete de comandos para instalar un intérprete de comandos, etc.

En el capítulo 6 se instala el sistema base real. Usamos el programa chroot para arrancar un nuevo intérprete de comandos cuyo directorio raíz (root) se asignará a la partición LFS. Esto esencialmente equivale a reiniciar haciendo que el núcleo monte la partición LFS como partición raíz. La razón de que utilicemos chroot en lugar de reiniciar es que de este modo puedes seguir usando el sistema anfitrión. Mientras se está instalando el software, puedes simplemente conmutar a otra VC (consola virtual) o escritorio X y continuar usando tu ordenador como lo harías normalmente.

Cuando esté instalado todo el software, en el capítulo 7 configurará los guiones de arranque, en el capítulo 8 configurará el gestor de arranque y en el capítulo 9 hay algunas sugerencias sobre lo que puedes hacer cuando acabes el libro. En ese momento puedes por fin reiniciar tu nuevo sistema LFS, y empezar a usarlo realmente.

En pocas palabras, este es el proceso. Encontrarás información detallada sobre los pasos que has de seguir en cada capítulo, a medida que avances. Si en algún momento hay algo que no veas muy claro, no te preocupes. Lo entenderás perfectamente un poco más adelante.

Por favor, lee con atención el capítulo 2, ya que explica algunas cosas importantes que debes saber antes de comenzar a trabajar en el capítulo 5 y posteriores.

Convenciones utilizadas en este libro

Para facilitar la comprensión se utilizan ciertas convenciones a lo largo del libro. Aquí hay unos ejemplos:

`./configure --prefix=/usr`

> Un texto así debe teclearse exactamente como aparece, a menos que se indique lo contrario. También se utiliza en las secciones explicativas para identificar el comando al que se hace referencia.

```
install-info: unknown option `--dir-file=/mnt/lfs/usr/info/dir'
```

> Un texto así (de ancho fijo) representa salida por pantalla, probablemente como resultado de comandos, y también se usa para especificar nombres de archivo, como por ejemplo `/etc/lilo.conf`.

Énfasis

> Este tipo de texto se utiliza con varios fines en el libro, pero principalmente para poner de relieve puntos importantes o para dar ejemplos de qué se puede teclear.

http://www.linuxfromscratch.org/

Este tipo de texto se usa para hipervínculos, tanto al propio libro como a páginas externas, como How−To's, direcciones de descarga, sitios web, etc.

```
cat > $LFS/etc/group << "EOF"
root:x:0:
bin:x:1:
......
EOF
```

Este tipo de secciones se usa principalmente al crear archivos de configuración. El primer comando (en negrita) solicita al sistema que cree del archivo $LFS/etc/group a partir de lo que se teclee en las lineas siguientes, hasta encontrar la secuencia EOF. Por tanto, generalmente la sección entera debe teclearse tal cual.

Versión del libro

Esta es la traducción al castellano del libro LFS−BOOK versión 3.3 con fecha 7 de Abril del 2002. Si la versión tiene más de un mes de antigüedad, es probable que haya una versión más nueva disponible para su descarga. Puedes buscar nuevas versiones en los sitios de descarga alternativos (mirrors) relacionados a continuación.

Servidores alternativos

A continuación tienes una lista con los actuales servidores alternativos HTTP y FTP al 7 de Abril del 2002. Es posible que la lista no se encuentre actualizada. La información más reciente se puede encontrar en el servidor web principal: http://www.linuxfromscratch.org.

Servidores HTTP alternativos

De las versiones en castellano

- EcolNet, España [Varios servidores] – http://www.escomposlinux.org/lfs−es
- Cervera, España [126 Kbits] – http://www.macana−es.com

América del Norte

- Fremont, California, USA [100 Mbit] – http://www.linuxfromscratch.org/lfs/intro.shtml
- Columbus, Ohio, United States [1 Mbit] – http://www.us.linuxfromscratch.org/lfs/intro.shtml

Europa

- Mainz, Germany [100 Mbit] – http://lfs.linux−provider.net/lfs/intro.shtml
- Freising, Germany [4 Mbit] – http://www.de.linuxfromscratch.org/lfs/intro.shtml
- Vienna Univ. of Technology, Austria [64 Mbit] – http://www.at.linuxfromscratch.org/lfs/intro.shtml
- Oslo, Norway [100 Mbit] – http://www.no.linuxfromscratch.org/lfs/intro.shtml
- Lancaster, United Kingdom [100 Mbit] – http://linuxfromscratch.mirror.ac.uk/lfs/intro.shtml

- Teeside, United Kingdom [256 Kbit] – http://www.linuxfromscratch.co.uk/lfs/intro.shtml
- Amsterdam, The Netherlands [100 Mbit] – http://www.nl.linuxfromscratch.org/lfs/intro.shtml

Australia

- Brisbane, Australia [155 Mbit] – http://www.au.linuxfromscratch.org/lfs/intro.shtml

Servidores FTP alternativos

América del Norte

- Fremont, California, USA [FTP] [100 Mbit] – ftp://ftp.linuxfromscratch.org
- Fremont, California, USA [HTTP] [100 Mbit] – http://ftp.linuxfromscratch.org

Europa

- Vienna Univ. of Tech., Austria [FTP] [64 Mbit] – ftp://ftp.at.linuxfromscratch.org/pub/lfs
- Vienna Univ. of Tech., Austria [HTTP] [64 Mbit] – http://ftp.at.linuxfromscratch.org/pub/lfs
- Oslo, Norway [FTP] [100 Mbit] – ftp://ftp.no.linuxfromscratch.org/mirrors/lfs/
- Lancaster, United Kingdom [HTTP] [100 Mbit] – http://www.mirror.ac.uk/sites/ftp.linuxfromscratch.org
- Univ. of Twente, The Netherlands [HTTP] [100 Mbit] – http://ftp.nl.linuxfromscratch.org/linux/lfs
- Univ. of Twente, The Netherlands [FTP] [100 Mbit] – ftp:/ftp.nl.linuxfromscratch.org/pub/linux/lfs
- Freising, Germany [FTP] [4 Mbit] – ftp://ftp.de.[..]/mirrors/ftp.linuxfromscratch.org

Australia

- Brisbane, Australia [FTP] [155 Mbit] – ftp://ftp.planetmirror.com/pub/lfs/

Historial de modificaciones

3.3 – 7 de Abril del 2002

- Se actualizó a
  - autoconf–2.53
  - automake–1.6
  - bin86–0.16.2
  - binutils–2.12
  - bison–1.34
  - bzip2–1.0.2
  - diffutils–2.8
  - e2fsprogs–1.27
  - gawk–3.1.0
  - gettext–0.11.1
  - grep–2.5
  - less–374
  - lfs–bootscripts–1.9
  - lilo–22.2
  - linux–2.4.18

  - ♦ man−pages−1.48
  - ♦ modutils−2.4.15
  - ♦ reiserfsprogs−3.x.1b
  - ♦ shadow−4.0.3
  - ♦ texinfo−4.1
  - ♦ util−linux−2.11o
  - ♦ vim−6.1
- 7 Abril 2002 [gerard]: Añadido un nuevo servidor alternativo (mirror) situado en Freising, Germany
- 5 Abril 2002 [gerard]: Capítulo 07 – Loadkeys: Añadida la página que explica que puedes eliminar el enlace simbólico loadkeys de `/etc/rc.d/rcsysinit.d` si has compilado un mapa de teclado directamente en el núcleo.
- 5 Abril 2002 [gerard]: Capítulo 06 – Configurando el teclado: se explica que también puedes compilar el mapa de teclado directamente en el núcleo, lo cual ofrece ventajas adicionales.
- 5 Abril 2002 [gerard]: Actualizado a lfs−bootscripts−1.9
- 5 Abril 2002 [gerard]: Capítulo 05+06 – GCC: Añadidos comandos para eliminar el directorio `/usr/*-gnu`
- 4 Abril 2002 [gerard]: Capítulo 05 – Diffutils: Añadido −−disable−nls
- 3 Abril 2002 [gerard]: Appendix A – Gettext: Añadidas las descripciones de paquete que faltaban.
- 3 Abril 2002 [gerard]: Capítulo 05 – Montando $LFS/proc: Añadido **`chown root.root $LFS/proc`**. La operación recursiva chown en el Capítulo 6 no toca proc, de modo que aun sigue en propiedad del usuario *lfs*. No es que sea grave, pero tampoco es precisamente limpio.
- 3 Abril 2002 [gerard]: Capítulo 06 – Groff: Añadidos unos cuantos enlaces simbólicos usados por programas como **`xman`** y otros.
- 3 Abril 2002 [gerard]: Capítulo 04 – Montando particiones: Añadidas algunas notas sobre cómo manejar múltiples particiones ($LFS, $LFS/usr , etc).
- 3 Abril 2002 [gerard]: Capítulo 06 – E2fsprogs: Añadido el comando **`install−info`** para finalizar la instalación de info.
- 3 Abril 2002 [gerard]: Capítulo 06 – Bzip2: Invertido el orden de los comandos **`make`** y **`make −f Makefile−libbz2_so`**. Esto es necesario para que todos los objetos se compilen con la opción PIC (Position Independant Code).
- 3 Abril 2002 [gerard]: Capítulo 05 – Linux: Abreviadas las intrucciones de instalación, eliminando los pasos **`make config`** y **`make dep`** .
- 1 Abril 2002 [gerard]: No es broma: Capítulo 5+6 – Gawk: Añadido un aviso para no ejecutar nunca **`make uninstall`** en el paquete. Sería equivalente a **`rm −rf /usr/bin/*`** porque forzamos la definición del directorio `libexec` a `/usr/bin`
- 29 Marzo 2002 [markh]: Capítulo 05 y 06 – Actualizado a: diffutils−2.8, modutils−2.4.15 y vim−6.1. Eliminada la opción PR_PROGRAM para diffutils ya que /usr/bin/pr se detecta ahora en el guión de configuración. Eliminado sed como solución al problema con el resaltado de sintaxis en vim, ya que eso queda solucionado en la nueva versión.
- 26 Marzo 2002 [markh]: Capítulo 02 – Buscando ayuda: Añadida referencia al documento de preguntas inteligentes de ESR.
- 25 Marzo 2002 [markh]: Binutils – Añadida descripción para la librería libopcodes .
- 21 Marzo 2002 [gerard]: Capítulo 06 – Bzip2: Antes de mover `/usr/bin/bzless` y `/usr/bin/bzmore` al directorio `/bin`, eliminamos primero los archivos `/bin/bzless` y `/bin/bzmore`. En algunos sistemas no funciona sobreescribir los archivos existentes al usar enlaces fuertes.
- 21 Marzo 2002 [gerard]: Apéndice A – Sysklogd: Modificada la dirección de descarga a http://www.infodrom.org/projects/sysklogd/
- 20 Marzo 2002 [gerard]: Capítulo 06 – Configurar Cargador Dinámico: Eliminar los directorios `/lib` y `/usr/lib` del archivo `ld.so.conf`. Eran innecesarios.

- 16 Marzo 2002 [gerard]: Capítulo 06+Apéndice A: Eliminadas las dependencias de chroot. No es un paquete, así que no procede.
- 16 Marzo 2002 [gerard]: Capítulo 05+06 – Gawk: Añadidos comandos para pasar sed a `awklib/Makefile.in` para cambiar las definiciones de *datadir* y *libexecdir*
- 15 Marzo 2002 [gerard]: Capítulo 01 – Listas de correo: Añadida descripción de lfs–chat
- 15 Marzo 2002 [gerard]: Capítulo 06–Shadow: Mover también `libmisc.*a` a `/usr/lib`
- 14 Marzo 2002 [gerard]: Actualizado a bison–1.34, gettext–0.11.1, grep–2.5, lfs–bootscripts–1.8, shadow–4.0.3
- 11 Marzo 2002 [gerard]: Actualizado a binutils–2.12
- 11 Marzo 2002 [gerard]: Capítulo 07 – Setclock: Se apuntaba a la posibilidad de saltarse este paso, lo cual no es cierto a menos que se suprima el guión entero. Por ello se ha cambiado el texto para crear el archivo independientemente de cómo esté configurado el reloj de hardware.
- 11 Marzo 2002 [gerard]: Capítulo 07 – Loadkeys: Eliminada la necesidad de configurar `/etc/sysconfig/keyboard`. Esto queda obsoleto por el parche kbd (ahora se usa loadkeys –d).
- 11 Marzo 2002 [gerard]: Capítulo 05 – Gawk: Añadido el apaño –Dre_max_failures=re_max_failures2 para sistemas glibc–2.1.x
- 11 Marzo 2002 [gerard]: Capítulo 06 – Bzip2: Antes de instalar, eliminar `/usr/bin/bz*`. La instalación de bzip2 no trata adecuadamente archivos existentes cuando se usan enlaces fuertes, así que los borramos primero.
- 10 Marzo 2002 [gerard]: Capítulo 06 – Configurar teclado: Añadida sección para configurar el archivo keymap creando el enlace simbólico`/usr/share/kbd/keymaps/defkeymap.map.gz`.
- 9 Marzo 2002 [gerard]: Capítulo 08 – Configurar autoarranque: Añadido un comando a **`cp`** que copia automáticamente todas las imágenes del núcleo de `/etc/lilo.conf` a `$LFS/boot`.
- 9 Marzo 2002 [gerard]: Capítulo 06 – Man: Movido `man.conf` de `/usr/share/misc` a `/etc`.
- 9 Marzo 2002 [gerard]: Capítulo 07: Añadida página sobre el guión sysklogd y explicar que el guión por defecto incluye la opción *–m 0* para **`syslogd`**.
- 8 Marzo 2002 [gerard]: Eliminado el paquete Mawk package y sustituido por Gawk. Esto es debido a que mawk no va a seguir desarrollándose, a diferencia de gawk. Mawk tiene algunos fallos de compatibilidad con POSIX que están resueltos en Gawk.
- 8 Marzo 2002 [gerard]: Actualizado a:: autoconf–2.53, automake–1.6, bin86–0.16.2, bison–1.33, bzip2–1.0.2, e2fsprogs–1.27, gawk–3.1.0, gettext–0.11, less–374, lilo–22.2, linux–2.4.18, man–pages–1.48, modutils–2.4.14, reiserfsprogs–3.x.1b, shadow–4.0.2, texinfo–4.1, util–linux–2.11o

3.2 – 7 Marzo 2002

- Actualizado a:
    - lfs–bootscripts–1.6
- 1 Marzo 2002 [gerard]: Capítulo 05 – Creando directorios: Eliminados`/usr/var` y `/usr/local/var`. No están recomendados por el *FHS*.
- 27 Febrero 2002 [gerard]: Capítulo 06 – Make: Añadidos comandos para eliminar el bit setgid kmem bit de `/usr/bin/make`. Esto no es necesario en Linux para manejar la carga del sistema y causa algunos problemas que se solventan eliminando el bit setgid.
- 26 Febrero 2002 [gerard]: Actualizado a lfs–bootscripts–1.6
- 17 Febrero 2002 [gerard]: Capítulo 05 – Sh–utils: Añadido de nuevo el comando que mueve $LFS/usr/bin/chroot a $LFS/usr/sbin
- 17 Febrero 2002 [gerard] Actualizadas las dependencias de todos los paquetes.
- 15 Febrero 2002 [gerard] Capítulo 01: Añadido un nuevo mirror a la lista de Holanda (www.nl y ftp.nl).

- 11 Febrero 2002 [markh] Capítulo 05: Sh−utils: Eliminado && adicional del final de las instrucciones de instalación.
- 10 Febrero 2002 [gerard]: Capítulo 05 − Sh−utils: Eliminado *su* del comando *mv* ya que aún no está instalado en el Capítulo 5.

3.2−RC1 − Febrero 10, 2002

- Se actualizó a:
  - bison−1.31
  - file−3.37
  - glibc−2.2.5
  - lfs−bootscripts−1.5
  - linux−2.4.17
  - man−pages−1.47
  - psmisc−20.2
  - sysvinit−2.84
  - util−linux−2.11n

- Febrero 10, 2002 [gerard]: Capítulo 6: Se agregó un comando de sed para cambiar la ruta /usr/bin/gzip escrita dentro del código de gzexe por /bin/gzip.
- Febrero 10, 2002 [gerard]: Capítulo 5 + 6: Se movieron varios programas adicionales que son usados por los guiones de inicio al directorio ($LFS)/bin. Ningún programa usado por los guiones de inicio (excepto los demonios) debe estar en el directorio /usr en caso de que /usr no esté disponible en el proceso de inicio (cuando se monta mediante NFS, por ejemplo).
- Febrero 6, 2002 [markh]: Apéndice A − Todas las descripciones sincronizadas y actualizadas.
- Febrero 2, 2002 [gerard]: Capítulo 6 − Cambiando el dueño: Se agregó un "cd /" de manera que la barra principal pueda ser borrada de todos los directorios en los comandos chown. Es mucho más placentero escribir de esta manera.
- Febrero 2, 2002 [gerard]: Se actualizó a lfs−bootscripts−1.5
- Febrero 2, 2002 [gerard]: Capítulo 6 − Gzip: Se eliminó el enlace simbólico a compress. Gzip puede descomprimir archivos .Z, pero no puede comprimir en ese formato.
- Febrero 1, 2002 [gerard]: Se actualizó a lfs−bootscripts−1.3
- Febrero 1, 2002 [gerard]: Capítulo 6 − Glibc: En vez de usar sed en el archivo `config.make`, se crea el archivo `glibc-build/configparms` que contiene "cross−compiling = no".
- Enero 30, 2002 [marcheerdink]: Capítulo 5: Se cambió el comando para copiar los archivos de cabeceras para soportar versiones de cp menores que la 4.1.
- Enero 30, 2002 [markh]: Capítulos 5+6: Se agregó CPPFLAGS="$CPPFLAGS −D_GNU_SOURCE" al comando configure como parche. Esto arregla la compilación en las plataformas PPC y en m68k y no causa daños en x86.
- Enero 30, 2002 [gerard]: Capítulo 5 − Montando proc: Se cambió un poco el texto (dejaba implícito que solamente se podía montar el sistema de archivos proc una vez, lo cual no es cierto en la actualidad).
- Enero 30, 2002 [markh]: Capítulo 5: Se mejoró la explicación del make mrproper.
- Enero 30, 2002 [marcheerdink]: Capítulos 5+6: Se eliminó −−libexecdir de las opciones de configuración de fileutils.
- Enero 30, 2002 [marcheerdink]: Capítulo 6: Se agregó un enlace simbólico de vipw a vigr después de instalar shadow.
- Enero 30, 2002 [markh]: Capítulos 5+6: Se cambiaron las instrucciones de instalación de las binutils y de e2fsprogs para que usen directorios separados, como gcc y glibc.

- Enero 30, 2002 [gerard]: Capítulo 6 – Guiones de inicio: Se agregó un chown root.root después del cp.
- Enero 30, 2002 [gerard]: Apéndice A – Texinfo: El directorio de trabajo de los programas info es /usr/share/info no /usr/doc/info.
- Enero 30, 2002 [gerard]: Capítulo 6 – Procps: Corrección tipográfica de la ruta del directorio app–defaults (es /usr/X11R6/lib/X11/app–defaults y no usr/X11R6/lib/app–defaults).
- Enero 30, 2002 [gerard]: Capítulo 6 – Configurando los programas: Se simplificaron los comandos para crear los archivos utmp, btmp, lastlog y wtmp.
- Enero 30, 2002 [gerard]: Capítulo 1: Se movieron los Agradecimientos, de modo que aparezcan en la primera página del capítulo 1.
- Enero 30, 2002 [gerard]: Capítulo 1: Se creó una página separada para listar los servidores alternativos FTP y HTTP.
- Enero 30, 2002 [gerard]: Capítulo 4 – Creando la partición: Se aumentó el tamaño sugerido de 750MB a 1 GB.
- Enero 29, 2002 [gerard]: Capítulo 6 – Shadow: Se combinaron los comandos "mv libshadow.a /usr/lib" y "mv libshadow.la /usr/lib" en "mv libshadow.*a /usr/lib"
- Enero 26, 2002 [gerard]: Se actualizó a lfs–bootscripts–1.2
- Enero 26, 2002 [marcheerdink]: Capítulo 6: Se eliminó la opción datadir de las opciones de configuración de bison, porque las versiones recientes de bison usan por defecto el directorio correcto.
- Enero 23, 2002 [markh]: Capítulo 6: Se añadió la sección Creando el enlace simbólico /etc/mtab.
- Enero 23, 2002 [gerard]: Se eliminó el comando file –C de la instalación de file. Este paquete ejecuta este comando casi al final de la instalación, así que ya no necesitamos hacerlo.
- Enero 23, 2002 [marcheerdink]: Capítulo 4+5+6: El entorno estático ahora se construye utilizando un usuario no privilegiado, eliminando el riesgo de sobreescribir archivos de la distribución anfitriona.
- Enero 22, 2002 [markh]: Eliminamos la instalación de las páginas del manual de linuxthreads, ya que no funciona (necesitan perl, y no lo tenemos instalado en ese punto).
- Enero 21, 2002 [markh]: Se actualizó a glibc–2.2.5. A la vez, se modificó la instalación de glibc de modo que se instalen las páginas de manual de linuxthreads.
- Enero 21, 2002 [markh]: Se actualizó a bison–1.31, file–3.37, kernel–2.4.17, psmisc–20.2 y sysvinit–2.84.
- Enero 21, 2002 [markh]: Se actualizó a util–linux–2.11n y se eliminó ADD_RAW=yes ya que no se necesita más.
- Enero 21, 2002 [markh]: Se actualizó a man–pages–1.47 y se eliminó el parche de man–pages.
- Enero 15, 2002 [gerard]: Apéndice A: Se agregaron archivos de los guiones de inicio (dependencias, sitio de descarga, descripciones).
- Enero 15, 2002 [gerard]: Capítulo 6: Se añadió la instalación de los guiones de inicio.
- Enero 15, 2002 [gerard]: Capítulo 7: Se eliminaron la mayoría de los guiones, sólo quedó parte de algunos pocos donde establecemos los archivos de configuración en /etc/sysconfig.
- Enero 15, 2002 [gerard]: Capítulo 6 – Configurando Sysvinit: Se cambió el contenido de inittab para que correspondiera a los nuevos guiones.
- Enero 15, 2002 [marcheerdink]: Capítulo 6 – file: Se cambiaron la instrucciones de instalación, ya que ahora sed ya no es necesario.
- Enero 14, 2002 [marcheerdink]: Se cambió la instalación de los archivos de cabecera del núcleo en el capítulo 5 de modo que sea un poco más portable.
- Enero 6, 2002 [gerard]: Se reformateó el listado de dependencias.
- Enero 1, 2002 [gerard]: ¡Feliz año nuevo LFS!
- Enero 1, 2002 [markh]: ¡El primer historial de cambios del año! Se actualizó la nota de derechos de propiedad para cubrir el 2002 ;–) OK – Estoy triste...
- Diciembre 16, 2001 [gerard]: Capítulo 6 – Ed: Se reescribió la explicación de porqué ed es opcional, para eliminar cierta confusión.

- Diciembre 16, 2001 [gerard]: Capítulo 6 − Texinfo: Se reescribió la explicación de TEXMF para eliminar cierta confusión.
- Diciembre 15, 2001 [gerard]: Capítulo 4: Se reemplazó la referencia a "Truco para una sola partición" con una referencia a lfs_next_to_existing_systems.txt.
- Diciembre 15, 2001 [markh]: Fusión de los apéndices finalizada. Todos los viejos apéndices A, B y D ahora están en un (gran) Apéndice A.
- Diciembre 14, 2001 [markh]: Se fusionaron los apéndices A y B.
- Diciembre 13, 2001 [markh]: Apéndice B: Se cambió la etiqueta bdhtml para que la página de flex sea creada como flex.html en vez de flex.
- Diciembre 13, 2001 [markh]: Apéndice D: Se cambiaron las referencias a metalab.unc.edu y ftp.ibiblio.org al URL correcto ibiblio.org.
- Diciembre 12, 2001 [marcheerdink]: Capítulo 6: Se movió el parche de kbd a las instrucciones de instalación predeterminadas; se actualizó a kbd−1.06−2.patch para arreglar la instalación de algunos programas; se agregaron las descripciones de estos programas; se eliminó la advertencia sobre loadkeys −d que se heredó de los tiempos en que esa opción no se había fijado todavía.
- Diciembre 11, 2001 [markh]: Capítulo 6: Se agrega la explicación sobre "Por qué se accede a $LFS antes del chroot".
- Diciembre 10, 2001 [markh]: Capítulo 6: Se agregó un parche de kbd para el funcionamiento de loadkeys −d (parche de Matthias Benkmann; originalmente enviado a la lista lfs−dev).
- Diciembre 10, 2001 [markh]: Capítulo 6: Fueron recreados los enlaces simbólicos en bash, fileutils y gcc para que las instrucciones del capítulo 6 fueran independientes de las del capítulo 5.
- Diciembre 10, 2001 [marcheerdink]: Capítulo 5+6: Se eliminaron los comandos de sed para usar el archivo de respaldo que fue creado anteriormente en vez de escribir un archivo intermedio "tmp~".
- Diciembre 10, 2001 [marcheerdink]: Capítulo 5+6: El comando 'make' para la instalación de las diffutils ha cambiado a 'make PR_PROGRAM=/usr/bin/pr.' Este error fue enviado por Greg Schafer.
- Diciembre 7, 2001 [gerard]: Capítulo 6: Se cambió el comando de *./Configure −Dprefix=/usr* a *./configure.gnu −−prefix=/usr*. Este último es más consistente con las instrucciones de instalación de otros paquetes, y el resultado es idéntico que de la manera anterior.
- Diciembre 3, 2001 [markh]: Capítulo 2: Se agregó la sección ¿Qué Plataforma?

3.1 − Diciembre 3, 2001

- Se agregó:
  - reiserfsprogs−3.x.0j
- Se actualizó a:
  - MAKEDEV−1.4
  - bash−2.05a
  - e2fsprogs−1.25
  - gettext−0.10.40
  - libtool−1.4.2
  - lilo−22.1
  - linux−2.4.16
  - man−1.5j
  - man−pages−1.43
  - modutils−2.4.12
  - sysvinit−2.83
  - util−linux−2.11m
  - vim−6.0

- Noviembre 30, 2001 [markh]: Capítulo 6: Se actualizó a man−1.5j. Se eliminó el comando sed que teníamos que usar con la versión anterior ya que la nueva versión detecta el awk correctamente.
- Noviembre 30, 2001 [markh]: Capítulo 5: Se agregó la explicación sobre librerías estáticas que fue publicada en lfs−apps (cuando todavía existía) por Plasmatic.
- Noviembre 26, 2001 [markh]: Capítulo 5+6: Se actualizó a kernel−2.4.16 y modutils−2.4.12.
- Noviembre 26, 2001 [markh]: Capítulo 6: Se agregaron notas de conformidad según FHS a la instalación de findutils.
- Noviembre 19, 2001 [markh]: Capítulo 5+6: Se actualizó a bash−2.05a, lilo−22.1, MAKEDEV−1.4, man−pages−1.43 y util−linux−2.11m.
- Noviembre 5, 2001 [markh]: Capítulo 6: Se creó un nuevo guión de lex, en vez de enlazar a los comentarios sobre flex en lfs−dev. (Esto es similar a los que hicimos con bison y yacc).
- Octubre 27, 2001 [markh]: General: Se hizo un gran arreglo del XML. Esto no debería afectar el texto del libro, o su distribución. Si lo hace, ¡entonces algo ha salido mal!
- Octubre 27, 2001 [markh]: Capítulo 6: Se agregó reiserfsprogs−3.x.0j y se actualizó a lilo−22.0.2.
- Octubre 24, 2001 [markh]: General: Se arreglaron varios errores ortográficos que fueron descubiertos.
- Octubre 12, 2001 [markh]: Capítulo 5 − Núcleo: Se agregó una explicación respecto a el por qué copiamos los archivos de cabecera del núcleo en vez de enlazarlos simbólicamente.
- Octubre 12, 2001 [markh]: Apéndice A − Gzip: Se agregó uncompress a la descripción de gunzip, ya que faltaba.
- Octubre 12, 2001 [markh]: Capítulo 6 − Util−linux: Se eliminó la entrada USRGAMES_DIR=/usr/bin ya que no es necesaria con util−linux−2.11l.
- Octubre 9, 2001 [gerard]: Capítulo 6 − Kbd: Se quitó la opción −−datadir, el valor por defecto ya está establecido correctamente.
- Octubre 7, 2001 [gerard]: Capítulo 6 − Shadow: Se menciona la receta http://hints.linuxfromscratch.org/hints/shadowpasswd_plus.txt
- Octubre 7, 2001 [gerard]: Capítulo 6 − Vim: Se cambiaron las instrucciones de instalación para arreglar un error en el archivo `syntax/sh.vim` y se agregó la variable CPPFLAGS para especificar el archivo global vimrc como `/etc/vimrc`.
- Octubre 7, 2001 [gerard]: Capítulo 6: Se actualizó a libtool−1.4.2, lilo−22.0, man−pages−1.40, modutils−2.4.10, sysvinit−2.83, util−linux−2.11l y vim−6.0
- Octubre 2, 2001 [gerard]: Capítulo 9 − El Final: Se agregó una referencia al contador LFS que está en http://linuxfromscratch.org/cgi−bin/lfscounter.cgi
- Septiembre 26, 2001 [gerard]: Capítulo 1 − Servidor de Noticias (News): Se agregó una referencia al servidor de noticias.
- Septiembre 26, 2001 [markh]: Capítulo 6 − E2fsprogs: Se cambió −−with−root−prefix=/ por with−root−prefix="" en las instrucciones de instalación de e2fsprogs. La razón de este cambio es que el valor / hace que los enlaces simbólicos y las rutas de instalación usen cosas como //lib en vez de solamente /lib. No es un problema fatal, es sólo que no queda bien.
- Septiembre 26, 2001 [markh]: Capítulo 5+6: Se actualizó a e2fsprogs−1.25, gettext−0.10.40, linux−2.4.10, modutils−2.4.9 y util−linux−2.11i.
- Septiembre 22, 2001 [markh]: Apéndice A: Se reordenaron las descripciones en orden alfabético.

3.0 − Septiembre 21, 2001

- Se actualizó a:
    - e2fsprogs−1.24
- Septiembre 21, 2001 [markh]: Capítulo 1+7: Se cambió la información de las listas de correo para reflejar la nueva estructura. El cambio en el capítulo 7 es que los guiones rc y rcS ahora les dicen a las personas que informen sobre los problemas en lfs−dev en vez de hacerlo a lfs−discuss.

- Septiembre 18, 2001 [gerard]: Capítulo 5+6 – GCC: Se agregó −−enable−threads=posix al capítulo 5, y se cambió −−enable−threads por −−enable−threads=posix en el capítulo 6. A pesar de que los hilos posix son los predeterminados si no se especifica un tipo de hilo, es más claro así ver qué se habilitó.
- Septiembre 17, 2001 [gerard]: Capítulo 6 – Psmisc: Se agregaron notas respecto a cómo encargarse del enlace simbólico de pidof de psmisc (en caso de que sysvinit no esté instalado) y su página del manual. También se agregó −−exec−prefix=/ al guión de configuración para que los programas fueran instalados en /bin en vez de /usr/bin (Los guiones de inicio tal vez pueden usarlos, así que éstos deben estar en /bin).
- Septiembre 16, 2001 [markh]: Capítulo 6 – Util−linux: Se agregó USRGAMES_DIR=/usr/bin a la rutina make install para que /usr/games no fuera creado por banner y que éste se instale en /usr/bin.
- Septiembre 14, 2001 [markh]: Capítulo 6 – E2fsprogs: Se actualizó a la versión 1.24.
- Septiembre 11, 2001 [gerard]: Capítulo 6 – Man: Se agregaron unos && que le faltaban a 'done' y se cambiaron los permisos del guión configure al modo 755 en vez de 700 (un modo predeterminado para que las personas no tengan que ser el dueño del archivo).

Listas de correo y archivos

El servidor linuxfromscratch.org hospeda las siguientes listas de correo de acceso público:

- lfs−support
- lfs−dev
- lfs−announce
- lfs−security
- lfs−book
- lfs−chat
- alfs−discuss
- blfs−dev
- blfs−book
- blfs−support

lfs−support

La lista de correo lfs−support proporciona soporte a los usuarios que se están construyendo un sistema LFS como el descrito en el libro principal. Las solicitudes de ayuda para instalar software no incluido en el sistema base deben hacerse en la lista blfs−support.

lfs−dev

La lista de correo lfs−dev es para discutir cosas directamente relacionadas con el LFS−BOOK. Si hay problemas con el libro, se necesita comunicar uno o dos errores, o hacer una sugerencia para mejorar el libro, esta lista de correo es la adecuada.

Las solicitudes de ayuda deben dirigirse a lfs−support o blfs−support.

lfs−announce

La lista lfs−announce es una lista moderada. Puedes suscribirte a ella, pero no puedes enviar ningún mensaje. Esta lista se usa para anunciar las nuevas versiones estables. La lista lfs−dev también facilitará información sobre el desarrollo de las versiones. Si un usuario está en la lista lfs−dev tiene poca utilidad que se suscriba a

esta lista, porque lo que se publica en lfs−announce se publica también en lfs−dev.

lfs−security

La lista lfs−security es para discutir cuestiones relacionadas con la seguridad. Los problemas de seguridad relacionados con los paquetes que se usan en LFS deben dirigirse a esta lista.

lfs−book

La lista lfs−book es usada por los editores del LFS−BOOK para coordinar su mantenimiento, como la edición XML y similares. La discusión actual sobre lo que debería añadirse o eliminarse del libro tiene lugar en lfs−dev.

lfs−chat

La lista lfs−chat es un lugar de encuentro para los miembros de la Comunidad LFS (en la que también estás incluido) y sirve para charlar sobre todo. No hace falta que esté relacionado con ordenadores. Todo vale y nada está fuera de lugar.

alfs−discuss

La lista alfs−discuss es para discutir el desarrollo de ALFS, el Linux Desde Cero Automatizado (Automated Linux From Scratch). El objetivo de este proyecto es desarrollar una herramienta de instalación que pueda crear un sistema LFS de forma automática. Su principal objetivo es acelerar la compilación eliminando la necesidad de introducir manualmente los comandos para configurar, compilar e instalar los paquetes.

blfs−dev

La lista blfs−dev es para discutir cuestiones relacionadas con el BLFS−BOOK (Más allá de LFS). Si hay problemas con el libro, se necesita comunicar uno o dos errores, o hacer una sugerencia para mejorar el libro (como añadir instrucciones de instalación), esta lista de correo es la adecuada.

La solicitud de ayuda sobre programas que estén más allá de la puesta en marcha de un LFS base (no sólo los que están en BLFS) deben dirigirse a blfs−support.

blfs−book

La lista blfs−book es usada por los editores del BLFS−BOOK para coordinar su mantenimiento, como la edición XML y similares. La discusión actual sobre lo que debería añadirse o eliminarse del libro tiene lugar en blfs−dev.

blfs−support

La lista blfs−support es para dar soporte sobre cualquier software no instalado en en libro LFS. Esta lista no es sólo para ayudar con los paquetes explícitamente mencionados en el libro BLFS, cualquier paquete que no se instale como parte del sistema base LFS puede ser discutido aquí.

Archivos de mensajes

Todas estas listas están archivadas y pueden verse en línea en http://archive.linuxfromscratch.org/mail−archives o descargarlas de http://ftp.linuxfromscratch.org/mail−archives o ftp://ftp.linuxfromscratch.org/mail−archives.

Cómo escribir en una lista

No necesitas suscribirte a una lista para mandar un mensaje. Sin embargo, si escribes en una lista a la que no estás suscrito, debes mencionar esto para que los miembros de la lista puedan incluirte en la cabecera CC: de las respuestas para que puedas recibirlas.

La dirección de envío a una lista es del formato *nombre_de_la_lista@linuxfromscratch.org* donde *nombre_de_la_lista* puede ser una de la listas mencionadas en la anterior sección. Ejemplos de las direcciones de envío son *lfs−dev@linuxfromscratch.org, lfs−support@linuxfromscratch.org y blfs−support@linuxfromscratch.org*.

Cómo suscribirse

Puedes suscribirte a cualquiera de las listas anteriormente mencionadas enviando un mensaje a listar@linuxfromscratch.org y escribiendo *subscribe nombre_de_la_lista* como asunto del mensaje.

Puedes suscribirte a múltiples listas con un solo mensaje. Esto se hace dejando el asunto en blanco y colocando todos los comandos en el cuerpo del mensaje. El mensaje tendrá este aspecto:

```
Para: listar@linuxfromscratch.org
Asunto:

subscribe lfs−dev
subscribe blfs−support
subscribe alfs−discuss
```

Después de enviar el mensaje, el programa Listar te enviará un mensaje solicitando la confirmación a la petición de suscripción. Después de enviar la confirmación, Listar te enviará un mensaje diciendo que has sido suscrito a la lista o listas y una introducción particular para cada lista.

Cómo desuscribirse

Para desuscribirte de una lista envía un mensaje a listar@linuxfromscratch.org y pon *unsubscribe nombre_de_la_lista* como asunto del mensaje.

Puedes desuscribirte de múltiples listas con un solo mensaje. Esto se hace dejando el asunto en blanco y colocando todos los comandos en el cuerpo del mensaje. El mensaje tendrá este aspecto:

```
Para: listar@linuxfromscratch.org
Asunto:

unsubscribe lfs−dev
```

```
unsubscribe blfs-support
unsubscribe alfs-discuss
```

Después de enviar el mensaje, el programa Listar te enviará un mensaje solicitando la confirmación a la petición de desuscripción. Después de enviar la confirmación, Listar te enviará un mensaje diciendo que has sido desuscrito de la lista o listas.

Otros modos de las listas

Para que el usuario active los distintos modo de uso debe enviar un mensaje a listar@linuxfromscratch.org. Los modos de uso se establecen poniendo el comando apropiado como asunto del mensaje.

Como su nombre implica, el comando *Set* indica que se activa un modo de uso. El comando *Unset* indica que se desactiva un modo de uso.

La palabra "nombre_de_la_lista" en los asuntos de ejemplo mostrados a continuación debe reemplazarse con el nombre de la lista a la que se desea aplicar ese modo de uso. Si se necesita activar en un sólo mensaje más de un modo de uso (en la misma lista o en varias listas) puede hacerse dejando el asunto en blanco y escribiendo los comandos en el cuerpo del mensaje.

Modo de resumen (Digest)

Set: *set nombre_de_la_lista digest*
Unset: *unset nombre_de_la_lista digest*

Todas las listas tienen disponible el modo de resumen, que puede activarse después de que el usuario se suscriba a la lista. Entrar en el modo de resumen hace que pares de recibir mensajes individuales al ritmo que se van publicando y, en su lugar, recibirás un mensaje diario conteniendo todos los mensajes publicados durante ese día.

Hay otro modo de resumen llamado digest2. Cuando el usuario activa este modo de uso recibe el resumen diario, pero también continúa recibiendo los mensajes individuales. Para activar este modo de uso sustituye *digest* por *digest2* en el comando.

Vacaciones

Set: *set nombre_de_la_lista vacation*
Unset: *unset nombre_de_la_lista vacation*

Si un usuario va a ausentarse un tiempo, o desea parar de recibir mensajes sin necesidad de desuscribirse, puede cambiar al modo de vacaciones. Esto tiene el mismo efecto que desuscribirse, pero sin tener que pasar por el proceso de desuscripción y posterior suscripción.

Servidores de noticias

Todas las listas de correo hospedadas en linuxfromscratch.org también son accesibles a través del servidor NNTP. Todos los mensajes publicados en una lista de correo son copiados en el grupo de noticias

correspondiente y viceversa.

El servidor de noticias (news) a usar es *news.linuxfromscratch.org*

FAQ

Si te encuentras con problemas al construir un sistema LFS, deberías comprobar en http://www.escomposlinux.org/lfs−es/faq/ si tu problema se encuentra ya solucionado en las FAQ. La versión original la tienes en http://www.linuxfromscratch.org/faq/

Información de contacto

Por favor, envía tus mensajes a las listas de correo. En el Capítulo 1 − Listas de correo y archivos tienes información sobre las listas de correo disponibles.

Si necesitas contactar directamente con Gerard Beekmans, manda un mensaje a gerard@linuxfromscratch.org

Capítulo 2. Información importante

Sobre $LFS

Por favor, lee con atención: en este libro la variable $LFS se usará frecuentemente. $LFS en todo momento deberá sustituirse por el directorio en el que se monta la partición que contiene el sistema LFS. Cómo crear y dónde montar la partición se explicará con todo detalle en el capítulo 4. Por ejemplo, supongamos que la partición LFS está montada en /mnt/lfs.

Por ejemplo, si las instrucciones son ejecutar un comando como **`./configure --prefix=$LFS`** , en realidad debes ejecutar **`./configure --prefix=/mnt/lfs`** .

Es importante hacer esto donde quiera que aparezca, ya sea en comandos introducidos en un intérprete de comandos, o al crear o editar un archivo.

Una posible solución es establecer la variable de entorno LFS. De este modo $LFS puede introducirse literalmente, en lugar de substituirlo por /mnt/lfs. Esto se consigue ejecutando:

```
export LFS=/mnt/lfs
```

Ahora, cuando las instrucciones sean ejecutar un comando como **`./configure --prefix=$LFS`** puedes introducir eso literalmente. Tu intérprete de comandos substituirá $LFS con /mnt/lfs al procesar la linea de comando (es decir, cuando pulses Enter después de haber tecleado el comando).

Si decides usar $LFS, no olvides establecer la variable $LFS en todo momento. Si la variable no está establecida y se usa en un comando, $LFS se ignorará y se ejecutará el resto. Un comando como **`echo "root:x:0:0:root:/root:/bin/bash" > $LFS/etc/passwd`** sin la variable $LFS establecida, recreará el archivo /etc/passwd de tu sistema anfitrión. Es decir: destruirá tu archivo de contraseñas actual.

Una forma de asegurar que $LFS está establecido en todo momento es añadirlo a los archivos /root/.bash_profile y /root/.bashrc, de modo que cada vez que entres como root, o hagas "su" a root, la variable $LFS estará establecida.

Cómo descargar los programas

En este documento asumiremos que todos los paquetes descargados se situarán en alguna parte de $LFS/usr/src.

Una convención que puedes usar es tener un directorio $LFS/usr/src/sources. Debajo de sources puedes crear el directorio 0–9 y los directorios de la "a" a la "z". Un paquete como sysvinit–2.84.tar.bz2 se guarda en $LFS/usr/src/sources/s/. Un paquete como bash–2.05a.tar.bz2 se guarda en $LFS/usr/src/sources/b/, y así.

El siguiente capítulo contiene la lista de todos los paquetes a descargar, pero aún no hemos creado la partición que va a contener nuestro sistema LFS. Por ello, deberás guardar los archivos en cualquier otro lugar y moverlos a $LFS/usr/src/ cuando hayas finalizado el capítulo en el que se prepara la nueva partición.

Cómo instalar los programas

Antes de que empieces a usar el libro de LFS, debemos señalar que todos los comandos asumen que estás usando el intérprete de comandos bash. Si no es el caso, los comandos pueden funcionar pero no podemos garantizarlo. Si no quieres complicarte la vida, usa bash.

Antes de que puedas empezar a hacer algo con un paquete, debes desempaquetarlo. A menudo los archivos están empaquetados con tar, y comprimidos con gzip o bzip2. No vamos a escribir cada vez cómo desempaquetar un archivo. Vamos a explicarlo una vez, en esta sección.

Para empezar, sitúate en el directorio $LFS/usr/src ejecutando:

```
cd $LFS/usr/src
```

Si un archivo está empaquetado con tar y gzip, se desempaqueta con uno de estos comandos, dependiendo del nombre de archivo:

```
tar xvzf fichero.tar.gz
tar xvzf fichero.tgz
```

Si un archivo está empaquetado con tar y con bzip2, se descomprime ejecutando:

```
bzcat fichero.tar.bz2 | tar xv
```

Algunos programas tar (la mayoría hoy en día, pero no todos) están ligeramente modificados para poder procesar archivos bzip2 directamente, usando el parámetro –I, –y o –j, que funciona de forma similar al parámetro –z para manejar archivos gzip. El comando anterior funciona sea cual sea la forma en que tu sistema parchee bzip2.

Si el archivo sólo está empaquetado con tar, ejecuta:

```
tar xvf fichero.tar
```

Cuando se desempaqueta un archivo, se creará un nuevo directorio en el directorio actual (y este libro asume que los archivos se desempaquetan en $LFS/usr/src). Antes de continuar con las instrucciones de instalación, sitúate en ese nuevo directorio. Cada vez que este libro mencione instalar un paquete, previamente debes desempaquetar el archivo fuente y entrar (cd) en el directorio recién creado.

De vez en cuando deberás manejar archivos individuales, como parches. Estos archivos normalmente están comprimidos con gzip o bzip2. Antes de poder usarse, deben descomprimirse.

Si un archivo está comprimido con gzip, descomprímelo con

```
gunzip fichero.gz
```

Si un archivo está comprimido con bzip2 descomprímelo con:

```
bunzip2 fichero.bz2
```

Después de haber instalado un paquete, pueden hacerse dos cosas con él: se puede eliminar el directorio que contiene las fuentes, o conservarlo. Recomendamos encarecidamente eliminarlo. Si no lo haces, y tratas de reutilizar las mismas fuentes más adelante en el libro (por ejemplo, reutilizando las fuentes del capítulo 5 en el capítulo 6), puede no resultar como esperas. Los fuentes del capítulo 5 tendrán las opciones de tu distribución anfitriona, que no siempre van a ser adecuadas para el sistema LFS cuando entres en el entorno chroot. Incluso ejecutar algo como *make clean* no garantiza siempre una jerarquía de fuentes limpia.

O sea, evítate problemas y simplemente elimina el directorio fuente inmediatamente después de instalarlo.

Hay una excepción: las fuentes del núcleo. Tenlos a mano ya que vas a necesitarlos más adelante en el libro, cuando construyas un núcleo. Nada va a utilizar la jerarquía de fuentes del núcleo, de modo que no van a interferir. No obstante, si tienes limitado el espacio en disco, puedes eliminar la jerarquía de fuentes del núcleo y desempaquetarlo más tarde cuando sea necesario.

¿Qué plataforma?

LFS pretende ser independiente de la plataforma hasta donde sea posible. Dicho esto, el esfuerzo de desarrollo principal se centra en la plataforma x86. Intentamos incluir información, donde sea posible, sobre diferencias respecto a otras plataformas como la PPC. Si compilando tienes algún problema que no esté relacionado con la plataforma x86, no dudes en buscar ayuda en las listas de correo. Mejor incluso, si encuentras solución a algún problema particular relacionado con una de las otras plataformas, por favor hazlo saber en la lista de correo lfs−dev. Lo incluiremos en el libro (suponiendo que confirmemos que funciona).

Cómo buscar ayuda

Si tienes algún problema usando este libro, encontrarás que la gente en el Internet Relay Chat (IRC) y en las listas de correo estará dispuesta a ayudarte. Puedes encontrar una relación de las listas de correo de LFS en Capítulo 1 − Listas de correo y archivos. Para facilitarnos la tarea de ayudarte, no obstante, deberías asegurarte de disponer de toda la información relevante que puedas reunir. Esto nos ayudará a diagnosticar y solventar tu problema. Esta parte del libro te guiará a determinar qué tipo de información será de la mayor utilidad.

Información Básica

En primer lugar, necesitamos una explicación breve del problema. Información esencial a incluir:

- La versión del libro que estás usando, que es 3.3
- Con qué paquete o sección estás teniendo problemas
- Cuál es el mensaje de error o síntoma exacto que obtienes
- Si te has apartado de las recomendaciones del libro

Ten en cuenta que el que te hayas apartado de las recomendaciones del libro no implica, después de todo, que no vayamos a ayudarte. La razón de ser de LFS es la posibilidad de elección. Simplemente nos ayudará a detectar otras posibles causas de tu problema.

Problemas de Configuración

Cuando algo vaya mal en la fase en que se ejecuta el guión configure, consulta las últimas líneas de `config.log`. Este fichero contendrá los posibles errores encontrados durante la configuración que no se

muestran siempre en la terminal. Incluye esas líneas relevantes si decides buscar ayuda.

Problemas de Compilación

Para ayudarnos a determinar la causa del problema, nos va a ser útil tanto la salida del terminal como el contenido de varios archivos. Las salidas al terminal del guión configure y del comando make pueden ser útiles. No incluyas ciegamente todo el contenido pero, por otro lado, no incluyas demasiado poco. Por ejemplo, aquí hay una salida a terminal de make:

```
gcc -DALIASPATH=\"/mnt/lfs/usr/share/locale:.\"
-DLOCALEDIR=\"/mnt/lfs/usr/share/locale\" -DLIBDIR=\"/mnt/lfs/usr/lib\"
-DINCLUDEDIR=\"/mnt/lfs/usr/include\" -DHAVE_CONFIG_H -I. -I.
-g -O2 -c getopt1.c
gcc  -g -O2 -static -o make  ar.o arscan.o commands.o dir.o expand.o file.o
function.o getopt.o implicit.o job.o main.o misc.o read.o remake.o rule.o
signame.o variable.o vpath.o default.o remote-stub.o version.o opt1.o
-lutil job.o: In function `load_too_high':
/lfs/tmp/make-3.79.1/job.c:1565: undefined reference to `getloadavg'
collect2: ld returned 1 exit status
make[2]: *** [make] Error 1
make[2]: Leaving directory `/lfs/tmp/make-3.79.1'
make[1]: *** [all-recursive] Error 1
make[1]: Leaving directory `/lfs/tmp/make-3.79.1'
make: *** [all-recursive-am] Error 2
```

En este caso, mucha gente simplemente incluye la sección inferior desde

```
make [2]: *** [make] Error 1
```

en adelante. Esto no nos basta para diagnosticar el problema porque sólo nos dice que *algo* fue mal, no *el qué*. Lo que se debería incluir para resultar útil es la sección completa tal como se incluye en el ejemplo, ya que incluye el comando que se estaba ejecutando y sus mensajes de error.

Hay un artículo excelente sobre búsqueda de ayuda en Internet en general, de Eric S. Raymond. Está disponible en http://www.tuxedo.org/~esr/faqs/smart-questions.html. Lee y sigue los consejos de este documento y tendrás muchas más posibilidades de obtener una respuesta, y también de que obtengas la ayuda que necesitas.

II. Parte II – Instalando el sistema LFS

Tabla de contenidos

3. Paquetes que hay que descargar
4. Preparando una nueva partición
5. Preparando el sistema LFS
6. Instalando los programas del sistema base
7. Preparando los guiones de arranque
8. Haciendo el sistema LFS arrancable
9. El final

Capítulo 3. Paquetes que hay que descargar

Introducción

A continuación se dará una lista de todos los paquetes que se necesitan descargar para construir el sistema básico. Los números de versión que los acompañan corresponden a versiones del software que se sabe que funcionan y en los cuales se basa este libro. Si experimentas algún problema que no puedes resolver por ti mismo, descarga, por favor, la versión que aparece en el libro (si es que has descargado una versión posterior).

Todas las URLs que aparecen, apuntan al servidor ftp.linuxfromscratch.org. Tenemos un par de servidores FTP alternativos disponibles desde los que también puedes descargar los ficheros. Las direcciones de esos servidores las puedes encontrar en el Capítulo 1 – Servidores alternativos.

Hemos incluido una lista de los sitios oficiales de descarga de los paquetes en el Apéndice A. El servidor FTP de LFS sólo contiene las versiones de los paquetes que se recomienda usar en este libro. Puedes comprobar si en los sitios oficiales incluidos en el Apéndice A tienes disponible una versión posterior. Si descargas un paquete más nuevo, apreciaríamos que nos informases si has conseguido instalar el paquete utilizando las instrucciones del libro o no.

Por favor, ten en cuenta que todos los archivos descargados del servidor FTP de LFS están comprimidos con bzip2 en lugar de gz. Si no sabes manejar archivos bz2, lee el Capítulo 2 – Cómo instalar los programas.

Paquetes que hay que descargar

Servidor FTP Navegable:
ftp://ftp.linuxfromscratch.org/

Servidor HTTP Navegable:
http://ftp.linuxfromscratch.org/

Puedes descargar un archivo que contiene todos los paquetes usados
para compilar un sistema LFS:

Todos los paquetes de LFS – 87,260 KB:
ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/lfs–packages–3.3.tar
http://ftp.linuxfromscratch.org/lfs–packages/3.3/lfs–packages–3.3.tar

O descargar individualmente los siguientes paquetes:

Bash (2.05a) – 1,400 KB:
ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/bash–2.05a.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/3.3/bash–2.05a.tar.bz2

Binutils (2.12) – 9,312 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/binutils-2.12.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/binutils-2.12.tar.bz2

Bzip2 (1.0.2) – 610 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/bzip2-1.0.2.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/bzip2-1.0.2.tar.bz2

Diff Utils (2.8) – 640 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/diffutils-2.8.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/diffutils-2.8.tar.bz2

File Utils (4.1) – 1217 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/fileutils-4.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/fileutils-4.1.tar.bz2

GCC (2.95.3) – 9,618 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/gcc-2.95.3.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/gcc-2.95.3.tar.bz2

Parche para GCC (2.95.3-2) – 8 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/gcc-2.95.3-2.patch.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/gcc-2.95.3-2.patch.bz2

Linux Kernel (2.4.18) – 23,595 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/linux-2.4.18.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/linux-2.4.18.tar.bz2

Grep (2.5) – 545 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/grep-2.5.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/grep-2.5.tar.bz2

Gzip (1.2.4a) – 178 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/gzip-1.2.4a.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/gzip-1.2.4a.tar.bz2

Parche para Gzip (1.2.4a) – 1 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/gzip-1.2.4a.patch.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/gzip-1.2.4a.patch.bz2

Make (3.79.1) – 794 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/make-3.79.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/make-3.79.1.tar.bz2

Sed (3.02) – 221 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/sed-3.02.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/sed-3.02.tar.bz2

Sh-utils (2.0) – 824 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/sh-utils-2.0.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/sh-utils-2.0.tar.bz2

Parche para Sh-utils (2.0) – 1 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/sh-utils-2.0.patch.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/sh-utils-2.0.patch.bz2

Tar (1.13) – 730 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/tar-1.13.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/tar-1.13.tar.bz2

Parche para Tar (1.13) – 1 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/tar-1.13.patch.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/tar-1.13.patch.bz2

Text Utils (2.0) – 1,040 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/textutils-2.0.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/textutils-2.0.tar.bz2

Gawk (3.1.0) – 1,286 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/gawk-3.1.0.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/gawk-3.1.0.tar.bz2

Texinfo (4.1) – 1,161 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/texinfo-4.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/texinfo-4.1.tar.bz2

Patch (2.5.4) – 149 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/patch-2.5.4.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/patch-2.5.4.tar.bz2

MAKEDEV (1.4) – 7 KB:

ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/MAKEDEV−1.4.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/MAKEDEV−1.4.bz2

Glibc (2.2.5) − 12,114 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/glibc−2.2.5.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/glibc−2.2.5.tar.bz2

Glibc−linuxthreads (2.2.5) − 164 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/glibc−linuxthreads−2.2.5.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/glibc−linuxthreads−2.2.5.tar.bz2

Man−pages (1.48) − 537 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/man−pages−1.48.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/man−pages−1.48.tar.bz2

Ed (0.2) − 158 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/ed−0.2.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/ed−0.2.tar.bz2

Find Utils (4.1) − 226 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/findutils−4.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/findutils−4.1.tar.bz2

Parche para Find Utils (4.1) − 1 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/findutils−4.1.patch.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/findutils−4.1.patch.bz2

Ncurses (5.2) − 1,308 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/ncurses−5.2.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/ncurses−5.2.tar.bz2

Vim (6.1) − 2,890 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/vim−6.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/vim−6.1.tar.bz2

Bison (1.34) − 585 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/bison−1.34.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/bison−1.34.tar.bz2

Less (374) − 189 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/less−374.tar.bz2

http://ftp.linuxfromscratch.org/lfs−packages/3.3/less−374.tar.bz2

Groff (1.17.2) − 1,214 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/groff−1.17.2.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/groff−1.17.2.tar.bz2

Man (1.5j) − 167 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/man−1.5j.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/man−1.5j.tar.bz2

Perl (5.6.1) − 4,750 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/perl−5.6.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/perl−5.6.1.tar.bz2

M4 (1.4) − 249 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/m4−1.4.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/m4−1.4.tar.bz2

Autoconf (2.53) − 739 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/autoconf−2.53.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/autoconf−2.53.tar.bz2

Automake (1.6) − 451 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/automake−1.6.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/automake−1.6.tar.bz2

Flex (2.5.4a) − 278 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/flex−2.5.4a.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/flex−2.5.4a.tar.bz2

File (3.37) − 140 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/file−3.37.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/file−3.37.tar.bz2

Libtool (1.4.2) − 653 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/libtool−1.4.2.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/libtool−1.4.2.tar.bz2

Bin86 (0.16.2) − 112 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/bin86−0.16.2.tar.bz2

http://ftp.linuxfromscratch.org/lfs-packages/3.3/bin86-0.16.2.tar.bz2

Gettext (0.11.1) - 2,039 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/gettext-0.11.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/gettext-0.11.1.tar.bz2

Kbd (1.06) - 559 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/kbd-1.06.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/kbd-1.06.tar.bz2

Parche para Kbd (1.06-2) - 3 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/kbd-1.06-2.patch.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/kbd-1.06-2.patch.bz2

E2fsprogs (1.27) - 1,176 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/e2fsprogs-1.27.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/e2fsprogs-1.27.tar.bz2

Lilo (22.2) - 292 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/lilo-22.2.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/lilo-22.2.tar.bz2

Modutils (2.4.15) - 211 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/modutils-2.4.15.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/modutils-2.4.15.tar.bz2

Procinfo (18) - 22 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/procinfo-18.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/procinfo-18.tar.bz2

Procps (2.0.7) - 153 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/procps-2.0.7.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/procps-2.0.7.tar.bz2

Psmisc (20.2) - 123 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/psmisc-20.2.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/psmisc-20.2.tar.bz2

Reiserfsprogs (3.x.1b) - 243 KB:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/reiserfsprogs-3.x.1b.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/reiserfsprogs-3.x.1b.tar.bz2

Shadow Password Suite (4.0.3) − 760 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/shadow−4.0.3.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/shadow−4.0.3.tar.bz2

Sysklogd (1.4.1) − 67 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/sysklogd−1.4.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/sysklogd−1.4.1.tar.bz2

Sysvinit (2.84) − 76 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/sysvinit−2.84.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/sysvinit−2.84.tar.bz2

Util Linux (2.11o) − 1,020 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/util−linux−2.11o.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/util−linux−2.11o.tar.bz2

Netkit−base (0.17) − 49 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/netkit−base−0.17.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/netkit−base−0.17.tar.bz2

Net−tools (1.60) − 194 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/net−tools−1.60.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/net−tools−1.60.tar.bz2

LFS−Bootscripts (1.9) − 26 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/lfs−bootscripts−1.9.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/lfs−bootscripts−1.9.tar.bz2

Tamaño total de todos los paquetes: 87,260 KB (85.21 MB)

Capítulo 4. Preparando una nueva partición

Introducción

En este capítulo se preparará la partición que va a contener el sistema LFS. Crearemos la partición, un sistema de archivos y la estructura de directorios. Cuando hayamos hecho esto, podremos pasar al siguiente capítulo y comenzar el verdadero proceso de construcción del sistema.

Creando una nueva partición

Primero, empecemos comentando que es posible construir un sistema LFS en una sola partición, en la que esté instalada tu distribución original. No se recomienda si es la primera vez que pruebas LFS, aunque podría ser útil si tienes poco disco disponible. Si te sientes valiente, echa un vistazo al consejo *Instalando LFS en la misma partición que sistemas existentes (Install LFS next to existing systems on the same partition)* que podrás encontrar en http://hints.linuxfromscratch.org/hints/lfs_next_to_existing_systems.txt

Antes de poder construir un nuevo sistema Linux, necesitaremos tener una partición vacía Linux en la cual podamos hacerlo. Recomendamos un tamaño de partición de, aproximadamente, 1 GB. Esto nos da suficiente espacio para almacenar todos los archivos de código fuente y compilar todos los paquetes sin la preocupación de quedarnos sin el espacio temporal en disco necesario. Pero probablemente querrás más espacio si planeas utilizar tu sistema LFS como tu sistema Linux primario. Si ese es tu caso, querrás más espacio para poder instalar programas adicionales. Si ya dispones de una partición Nativa de Linux, puedes saltarte esta sección del libro.

Se debe usar el programa cfdisk (o el programa equivalente a fdisk que prefieras) utilizando como opción el disco duro apropiado (como /dev/hda si se va a crear una nueva partición en el disco maestro del bus IDE primario). Este programa se usa para crear una partición Nativa de Linux y escribir la tabla de particiones. Por favor, consulta la documentación que acompaña al programa fdisk que elijas (las páginas del manual –man– son, a menudo, un buen lugar donde empezar), y lee el procedimiento para crear una nueva partición Nativa de Linux y cómo escribir la tabla de particiones.

Se debe recordar la denominación de las nuevas particiones. Podrá ser algo como hda11. En este libro, nos referiremos a esta nueva partición creada como la partición LFS.

Creando un sistema de ficheros en la nueva partición

Una vez que se haya creado la partición, tendremos que crear un nuevo sistema de ficheros en esa partición. El sistema de ficheros estándar usado hoy en día es el sistema ext2, pero los llamados sistemas de ficheros con registro de transacciones (journaling) están aumentando su popularidad también. Desde luego, se deja a tu elección qué sistema de ficheros quieres crear pero, como tenemos que asumir alguno para trabajar con él, asumiremos que has escogido ext2.

Para crear un sistema de ficheros ext2, utiliza el comando mke2fs. El comando sólo necesita como opción la partición LFS para crear el sistema de ficheros.

```
mke2fs /dev/xxx
```

Cambia "xxx" por el nombre de tu partición (por ejemplo, hda11).

Montando la nueva partición

Nada más crear el sistema de ficheros, ya está listo para para su uso. Todo lo que tenemos que hacer para acceder a la partición (para, por ejemplo, leer datos y escribirlos), es montarla. Si se monta sobre el directorio /mnt/lfs, se puede acceder a la partición sin más que situarse en el directorio /mnt/lfs. No importa el directorio que elijas, sólo asegúrate de recordar cuál escogiste.

Crea el directorio /mnt/lfs ejecutando:

```
mkdir -p /mnt/lfs
```

Ahora, monta la partición LFS ejecutando:

```
mount /dev/xxx /mnt/lfs
```

Cambia "xxx" por el nombre de tu partición (por ejemplo, hda11).

Este directorio (/mnt/lfs) es la variable $LFS sobre la que ya has leído en el capítulo 2. Si habías pensado usar la variable de entorno $LFS, ahora debes ejecutar **`export LFS=/mnt/lfs`**

Si decidiste crear varias particiones para LFS (por ejemplo $LFS y $LFS/usr), móntalas como sigue:

```
mkdir -p /mnt/lfs &&
mount /dev/xxx /mnt/lfs &&
mkdir /mnt/lfs/usr &&
mount /dev/yyy /mnt/lfs/usr
```

Por supuesto, sustituye /dev/xxx y /dev/yyy por los nombres de tus particiones.

Capítulo 5. Preparando el sistema LFS

Introducción

En los próximos capítulos instalaremos todo el software que forma parte de un sistema Linux básico. Después de seguir las instrucciones de este capítulo y el siguiente tendrás un sistema Linux completamente operativo. Los capítulos restantes se encargan de preparar los guiones de arranque, hacer el sistema LFS arrancable y configurar una red básica.

El software utilizado en este capítulo tiene las librerías enlazadas estáticamente y será reinstalado en el siguiente capítulo con las librerías enlazadas dinámicamente. La razón por la que se instala la versión estática primero es porque puede dar la casualidad de que nuestro sistema Linux normal y nuestro sistema LFS no utilicen las mismas versiones de las librerías C. Si los programas utilizados en la primera parte estuvieran enlazados a versiones antiguas de las librerías C podrían funcionar incorrectamente en nuestro sistema LFS. Otra razón es la de resolver las dependencias circulares. Un ejemplo de estas dependencias es que necesitas un compilador para instalar un compilador, y que vas a necesitar un intérprete de comandos para instalar un intérprete de comandos y un compilador.

La clave para aprender a hacer funcionar Linux es saber qué es lo que hace cada paquete exactamente y por qué un usuario o un sistema los necesita. Tienes una descripción del contenido de cada paquete después de la subsección de instalación de cada uno de ellos y en el apéndice A.

Durante la instalación de varios paquetes verás aparecer en pantalla todo tipo de avisos (warnings). Ésto es normal y puedes ignorarlos tranquilamente. No son más que eso, avisos (la mayoría debidos a un uso inapropiado de la sintaxis de C o C++, pero no ilegal. Lo que sucede, normalmente, es que cambia el estándar C, pero los paquetes continúan utilizando el estándar antiguo).

Antes de comenzar, asegúrate de que la variable de entorno LFS tiene el valor correcto si has decidido utilizarla. Ejecuta el siguiente comando:

```
echo $LFS
```

Comprueba que la salida del comando corresponde con el directorio utilizado como punto de montaje de la partición LFS (por ejemplo /mnt/lfs).

¿Por qué usamos enlazado estático?

Doy gracias a Plasmatic por enviar el texto en el que está basado ésto a una de las listas de correo de LFS.

Cuando construyes (compilas) un programa, en vez de reescribir todas las funciones que se ocupan de tratar con el núcleo, hardware, ficheros, etc. con cada nuevo programa que escribas, puedes encontrar todas esas funciones básicas guardadas en librerías. glibc, la cual instalaremos más tarde, es una de las principales librerías, donde se encuentra el código de todas las funciones básicas que utilizan los programas, que permiten tareas como abrir ficheros, mostrar información en pantalla, y recoger datos introducidos por el usuario. Cuando el programa es compilado, estas librerías son enlazadas junto con el nuevo programa, así que puede usar cualquiera de las funciones que contengan las librerías.

Sin embargo, estas librerías pueden ser muy grandes (por ejemplo, libc.a ronda normalmente los 2.5MB). Por lo tanto, no se desea normalmente tener una copia de cada librería añadida al programa. Tan sólo imagina que

tienes un comando tan simple como ls con 2.5MB extra añadidos a él. En vez de hacer que la librería forme parte del programa, o que tenga las librerías enlazadas estáticamente, las librerías se mantienen en un fichero aparte que se carga cuando el programa lo necesita. Ésto es lo que se llama enlazar las librerías dinámicamente. Se les da este nombre porque las librerías se cargan y descargan dinámicamente según se las necesite.

Así que tenemos un fichero de 1kb y otro de 2.5MB, pero esto no nos ha ahorrado espacio (excepto, quizás, en memoria RAM mientras no se necesite la librería). La ventaja REAL del enlace dinámico es que sólo necesitamos una copia de la librería. Si `ls` y `rm` utilizan la misma librería, no necesitamos dos copias de ella, ya que ambos pueden utilizar el código del mismo fichero. Incluso cuando se cargan en memoria los dos programas comparten el código en vez de cargarlo dos veces. Así que no sólo se ahorra espacio en disco, si no que, además, se ahorra espacio en la preciada memoria RAM.

Si el enlazado dinámico nos ahorra tanto espacio, ¿por qué estamos haciendo que todo se enlace estáticamente? Bien, la razón es que cuando estableces como raíz (chroot) tu propio y nuevo (pero muy incompleto) entorno LFS, no estarán disponibles las librerías porque están situadas en alguna parte de tu antiguo árbol de directorios (por ejemplo `/usr/lib`), el cual no será accesible desde la raíz de LFS (`$LFS`).

Para que tus nuevos programas se ejecuten dentro de tu entorno, una vez establecida la nueva raíz, debes asegurarte que las librerías se enlazan estáticamente cuando los construyes. Por lo tanto, se utilizarán las opciones **`--enable-static-link`**, **`--disable-shared`**, y **`-static`** durante todo el capítulo 5. Una vez en el capítulo 6, lo primero que haremos será construir el principal conjunto de librerías del sistema, glibc. Tras hacer esto, comenzaremos a reconstruir los programas que habremos instalado en el capítulo 5, pero esta vez con las librerías enlazadas dinámicamente. De esta manera, aprovechamos las ventajas de ahorro de espacio que nos ofrecen las librerías dinámicas.

Ahora ya sabes cuáles son las razones por las que necesitas usar opciones raras como **`-static`**. Si intentas construir todo sin ellas verás rápidamente lo que sucede cuando establezcas la raíz a tu nuevo sistema LFS lisiado.

Si quieres saber más sobre librerías dinámicas consulta algún libro o página web sobre programación, en especial alguno relacionado con Linux.

Instalando todos los programas como usuario sin privilegios

Si estás conectado como administrador (root) durante los distintos pasos del capítulo 5 es posible que sobreescribas algunos ficheros del sistema con los que vayas creando en este capítulo. Hay todo tipo de causas para que esto suceda. Por ejemplo, porque no has definido la variable de entorno $LFS. Esta sobreescritura puede causar todo tipo de problemas en tu sistema, luego es una buena idea entrar en el sistema como un usuario sin privilegios durante el capítulo 5. Para evitar estas situaciones, y asegurarnos de tener el entorno lo mas limpio posible, vamos a crearnos un nuevo usuario llamado "lfs", el cual usaremos durante la instalación. Ejecutando como administrador los siguientes comandos crearemos el usuario "lfs":

```
useradd -s /bin/bash -m lfs &&
passwd lfs
```

Este es el momento de cambiar los permisos de tu partición LFS para que el usuario "lfs" tenga permiso de escritura en ella. Ejecuta como administrador el siguiente comando para que el nuevo dueño de la partición sea el usuario "lfs":

```
chown -R lfs $LFS
```

Ahora puedes conectarte como el usuario "lfs". Para conectarte dispones de dos opciones: conectarte normalmente desde el modo consola o el entorno gráfico o mediante **su - lfs**. Una vez estés conectado como el usuario "lfs", escribe los siguientes comandos para establecer el entorno de trabajo adecuado:

```
cat > ~/.bash_profile << "EOF"
umask 022

LFS=/mnt/lfs
LC_ALL=POSIX
export LFS LC_ALL
EOF
source ~/.bash_profile
```

Este perfil nos asegura que la máscara de permisos (umask) tenga el valor 022 para que los nuevos archivos y directorios que se creen posean los permisos correctos. Es recomendable que mantengas este valor para la máscara de permisos durante todo el proceso de instalación. Además, establece el valor de las variables de entorno $LFS y $LC_ALL. La variable $LFS ya se explicó en los capítulos anteriores. La variable $LC_ALL se utiliza para establecer el idioma del sistema.

En el caso de que tu distribución utilice una versión de glibc anterior a la 2.2.4, tener almacenado en la variable $LC_ALL algo que sea distinto a "C" o "POSIX", durante el capítulo 5, puede causar problemas cuando, estando en el capítulo 6, quieras salir del entorno chroot e intentes volver a entrar. Dándole el valor "POSIX" ("C" es un alias de "POSIX") nos aseguramos de que todo funcionará correctamente en el entorno cuando se establezca la nueva raíz.

Creando directorios

Ahora vamos a crear, en la partición LFS, el árbol de directorios basado en el estándar FHS, el cual se puede encontrar en http://www.pathname.com/fhs/. Siguiendo los siguientes pasos crearemos la estructura del árbol de directorios por defecto:

```
cd $LFS &&
mkdir -p bin boot dev/pts etc/opt home lib mnt proc root sbin tmp var opt &&
for dirname in $LFS/usr $LFS/usr/local
   do
   mkdir $dirname
   cd $dirname
   mkdir bin etc include lib sbin share src
   ln -s share/man
   ln -s share/doc
   ln -s share/info
   cd $dirname/share
   mkdir dict doc info locale man nls misc terminfo zoneinfo
   cd $dirname/share/man
   mkdir man{1,2,3,4,5,6,7,8}
done &&
cd $LFS/var &&
mkdir -p lock log mail run spool tmp opt cache lib/misc local &&
cd $LFS/opt &&
mkdir bin doc include info lib man &&
cd $LFS/usr &&
ln -s ../var/tmp
```

Normalmente los directorios se crean con los permisos 755, lo que no es una buena opción para todos los directorios. El primer cambio es establecer los permisos 0750 para el directorio $LFS/root. Esto nos asegura que no todos los usuarios tengan acceso al directorio /root (lo mismo que debería hacer cualquier usuario con los directorios /home/username). El segundo cambio es establecer los permisos 1777 para los directorios tmp. De esta manera cualquier usuario puede escribir en los directorios /tmp o /var/tmp, pero no pueden borrar los ficheros de otros usuarios (de esto último se encarga el llamado "bit pegajoso" (sticky bit) – bit 1 de la máscara de permisos 1777).

```
cd $LFS &&
chmod 0750 root &&
chmod 1777 tmp var/tmp
```

Ahora que se han creado los directorios debes copiar los ficheros fuente que fueron descargados en el capítulo 3 a algún directorio que cuelgue de $LFS/usr/src (tendrás que crear el directorio en el que desees meter los fuentes).

Nota: conformidad con el estándar FHS

FHS estipula que el directorio /usr/local debe contener los subdirectorios bin, games, include, lib, man, sbin y share. Puedes cambiar el directorio /usr/local si deseas seguir el estándar FHS.

Además, el estándar indica que debe existir el directorio /usr/share/games, el cual no nos gusta mucho para un sistema básico, pero puedes crearlo si deseas mantener el estándar. FHS no es muy preciso respecto a la estructura de los subdirectorios de /usr/local/share, así que nos podemos tomar la libertad de crear los directorios que nos parezcan más apropiados para nuestras necesidades.

Instalando Bash–2.05a

```
Estimación del tiempo de construcción:      3 minutos
Estimación del espacio de disco requerido:  20 MB
```

Instalación de Bash

Antes de intentar instalar Bash, tienes que comprobar y asegurarte de que tu distribución tiene los archivos `/usr/lib/libcurses.a` y `/usr/lib/libncurses.a`. Si tu distribución original es otro sistema LFS, todos estos ficheros existirán si seguiste las instrucciones del libro al pie de la letra.

Si no existe ninguno de los dos ficheros, debes instalar el paquete de desarrollo de la librería ncurses. Este paquete suele tener un nombre parecido a *ncurses–dev*. Si este paquete ya está instalado o acabas de instalarlo, comprueba los dos ficheros de nuevo. A menudo, el fichero `libcurses.a` puede faltar (todavía). Si es ese el caso, entonces crea el fichero `libcurses.a` como un enlace simbólico, ejecutando como administrador del sistema (root) las siguientes órdenes:

```
cd /usr/lib &&
ln -s libncurses.a libcurses.a
```

Ya podemos continuar. Instala Bash ejecutando las siguientes órdenes:

```
./configure --enable-static-link --prefix=$LFS/usr \
   --bindir=$LFS/bin --with-curses &&
make &&
make install &&
```

```
cd $LFS/bin &&
ln -sf bash sh
```

Si al terminar la etapa make install se imprimen unas líneas como estas:

```
install-info: unknown option `--dir-file=/mnt/lfs/usr/info/dir'
usage: install-info [--version] [--help] [--debug] [--maxwidth=nnn]
        [--section regexp title] [--infodir=xxx] [--align=nnn]
        [--calign=nnn] [--quiet] [--menuentry=xxx]
        [--info-dir=xxx]
        [--keep-old] [--description=xxx] [--test]
        [--remove] [--] filename
make[1]: *** [install] Error 1
make[1]: Leaving directory `/mnt/lfs/usr/src/bash-2.05a/doc'
make: [install] Error 2 (ignored)
```

entonces, probablemente, estás utilizando Debian y tienes una versión antigua del paquete texinfo. Este error no es grave de ninguna manera: las páginas info se instalarán cuando volvamos a compilar bash dinámicamente en el capítulo 6, así que puedes ignorarlo.

Cuando probamos la instalación con la versión más moderna de Debian, los últimos dos comandos se ejecutaron porque el proceso de instalación no devolvió un valor mayor de 0. Pero harías bien en comprobar si tienes el enlace simbólico `$LFS/bin/sh` en tu partición LFS. Si no lo tienes, ejecuta ahora manualmente los dos últimos comandos.

Explicación de los comandos

`--enable-static-link:` Esta opción de configure provoca que Bash sea enlazado estáticamente.

`--prefix=$LFS/usr:` Esta opción de configure instala todos los ficheros de Bash bajo el directorio $LFS/usr, que pasa a ser el directorio /usr cuando entramos en el entorno chroot o reiniciamos LFS.

`--bindir=$LFS/bin:` Esto instala los archivos ejecutables en $LFS/bin. Lo hacemos de esta manera porque queremos que bash esté en /bin, no en /usr/bin. Una razón para ésto es que el directorio /usr podría estar en una partición separada que tendrá que montarse en algún momento. Antes de que se monte esa partición, necesitas y querrás tener bash disponible (sería difícil ejecutar los guiones de arranque sin un intérprete de comandos, por ejemplo).

`--with-curses:` Esto provoca que Bash se enlace con la librería curses en lugar de la librería por defecto, termcap, que se está convirtiendo en obsoleta.

No es estrictamente necesario enlazar bash con libncurses (podría, por el momento, enlazarse con una librería estática termcap, y no pasaría nada porque, de todas formas, instalaremos de nuevo Bash en el capítulo 6, donde usaremos libncurses), pero es una buena forma de comprobar que el paquete ncurses se ha instalado correctamente. Si no es así, podrías tener problemas más adelante en este capítulo, cuando instales el paquete texinfo. Ese paquete necesita ncurses y no se puede utilizar termcap en ese caso.

`ln -sf bash sh:` Esta orden crea el enlace simbólico sh, que apunta a bash. Muchos guiones se ejecutan utilizando 'sh' (invocándole mediante #!/bin/sh como primera línea del guión) lo que invoca un modo especial de bash. Bash se comportará entonces (de manera tan similar como sea posible) como el intérprete de órdenes Bourne original.

Los **&&** al final de cada línea provocan que el siguiente comando a ejecutar sólo lo haga si el anterior terminó devolviendo un valor de 0, que indica éxito. Si todos estos comandos se copian en el intérprete de órdenes, es importante asegurarse de que, si ./configure falla, make no será ejecutado e, igualmente, si make falla, entonces make install no se ejecutará, y así sucesivamente.

Contenido de bash−2.05a

Programas

bash, sh (enlace a bash) y bashbug

Descripciones

bash

bash es la "Bourne−Again SHell", que es un completo intérprete de comandos usado ampliamente en sistemas Unix. Bash es un programa que lee de la entrada estándar, el teclado. Un usuario escribe algo y el programa evalúa lo que ha escrito y hace algo con ello, como lanzar un programa.

bashbug

bashbug es un guión que ayuda al usuario en la composición y envío de informes de errores relacionados con bash, en un formato estándar.

sh

sh es un enlace simbólico al programa bash. Cuando se invoca como sh, bash intenta imitar el comportamiento de las versiones antiguas de sh lo mejor posible, mientras que también cumple los estándares POSIX.

Dependencias

Para instalar Bash−2.05a se necesita lo siguiente:

bash: bash, sh
binutils: ar, as, ld, ranlib, size
diffutils: cmp
fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm
gcc: cc, cc1, collect2, cpp0, gcc
grep: egrep, grep
make: make
gawk: awk
sed: sed
sh−utils: basename, echo, expr, hostname, sleep, uname
texinfo: install−info
textutils: cat, tr, uniq

Instalando Binutils–2.12

```
Estimación del tiempo de construcción:      6 minutos
Estimación del espacio de disco requerido:  96 MB
```

Instalación de Binutils

Se sabe que este programa se comporta mal si cambias sus parámetros de optimización (incluyendo las opciones –march y –mcpu). Es mejor dejar a Binutils tal como está, así que te recomendamos borrar CFLAGS, CXXFLAGS y otras variables/parámetros como éstos que podrían cambiar la optimización con la que viene por defecto.

Instala Binutils ejecutando los siguientes comandos:

```
mkdir ../binutils-build &&
cd ../binutils-build &&
../binutils-2.12/configure --prefix=$LFS/usr --disable-nls &&
make LDFLAGS=-all-static tooldir=$LFS/usr &&
make tooldir=$LFS/usr install
```

Explicación de los comandos

`mkdir ../binutils-build`: Las instrucciones de instalación de Binutils recomiendan crear un directorio de construcción separado en lugar de compilar el paquete dentro del árbol de directorios de código fuente. Por ello, creamos un directorio binutils–build y trabajamos allí.

`--disable-nls`: Esta opción desactiva la internacionalización (también conocida como i18n). No la necesitamos para los programas estáticos, y nls, a menudo, causa problemas en el enlazado estático.

`LDFLAGS=-all-static`: Asignar a la variable LDFLAGS el valor –all–static provoca que binutils se enlace estáticamente.

`tooldir=$LFS/usr`: Normalmente, el directorio de herramientas (el directorio donde se instalarán los ejecutables de binutils) tiene el valor $(exec_prefix)/$(target_alias) lo que se convierte en, por ejemplo, /usr/i686–pc–linux–gnu. Como sólo construimos programas para nuestro propio sistema, no necesitamos en $LFS/usr este directorio específico de cada objetivo. Esa configuración se usaría si el sistema fuese usado para compilación cruzada (por ejemplo, compilando un paquete en una máquina Intel pero generando código que se ejecutará en máquinas Apple PowerPC).

Contenido de binutils–2.11.2

Programas

addr2line, ar, as, c++filt, gasp, gprof, ld, nm, objcopy, objdump, ranlib, readelf, size, strings y strip

Descripciones

addr2line

addr2line traslada direcciones de programas a nombres de ficheros y números de líneas. Dándole una dirección y un ejecutable, usa la información de depuración del ejecutable para averiguar qué fichero y

número de línea está asociado con dicha dirección.

ar

El programa ar crea, modifica y extrae desde archivos. Un archivo es un fichero que almacena una colección de otros ficheros en una estructura que hace posible obtener el original de cada fichero individual (llamados miembros del archivo).

as

as está pensado, principalmente, para ensamblar la salida del compilador GNU gcc para ser usada por el enlazador ld.

c++filt

El lenguaje C++ proporciona sobrecarga de funciones, lo que significa que es posible escribir varias funciones con el mismo nombre (suponiendo que cada una tome parámetros de diferente tipo). Todos los nombres de funciones C++ son codificadas dentro de una etiqueta de bajo nivel del ensamblador (este proceso es conocido como "mangling"). El programa c++flit hace lo contrario: decodifica (demangling) nombres de bajo nivel en nombres de nivel de usuario para que el enlazador pueda evitar conflictos en estas funciones sobrecargadas.

gasp

Gasp es el preprocesador de las macros del ensamblador.

gprof

gprof muestra el grafo de llamadas de los datos perfilados.

ld

ld combina un número de objetos y ficheros de archivo, reubica sus datos y establece las referencias a los símbolos. Frecuentemente, el último paso de la compilación de un nuevo programa es hacer una llamada a ld.

nm

nm lista los símbolos de los ficheros objeto.

objcopy

La utilidad objcopy copia el contenido de un fichero objeto en otro. objcopy usa la librería BFD de GNU para leer y escribir los ficheros objeto. Puede escribir el fichero objeto destino en un formato diferente al del fichero objeto fuente .

objdump

objdump muestra información sobre uno o más ficheros objeto. Mediante opciones se puede indicar la información a mostrar. Esta información es útil fundamentalmente para los programadores que trabajan en herramientas de compilación (al contrario de los programadores que sólo quieren que sus programas compilen y funcionen).

ranlib

ranlib genera un índice de los contenidos de un archivo, y lo coloca en el archivo. El índice lista cada símbolo definido por un miembro de un archivo que es un fichero objeto reubicable.

readelf

readelf muestra información sobre binarios de tipo elf.

size

size lista los tamaños de las secciones –y el tamaño total– para cada uno de los ficheros objeto en su lista de argumentos. Por defecto, se genera una línea de salida por cada fichero objeto o cada módulo de un archivo.

strings

Para cada fichero dado, strings muestra las cadenas de caracteres imprimibles de al menos 4 caracteres (o el número especificado en las opciones del programa) seguidas por un carácter no imprimible. Por defecto, sólo muestra las cadenas procedentes de las secciones de inicialización y carga de los ficheros objeto; para otros tipos de ficheros muestra todas las cadenas de los mismos.

strings es útil, principalmente, para determinar el contenido de ficheros que no son de texto.

strip

strip elimina todos los símbolos o sólo los especificados de los ficheros objeto. La lista de ficheros objeto puede incluir archivos. Se debe indicar, por lo menos, un fichero objeto. strip modifica los ficheros mencionados en sus argumentos, en vez de escribir copias modificadas con otro nombre.

Librerías

libbfd.a, libiberty.a y libopcodes.a

Descripciones

libbfd

libbfd es la librería de descriptores de ficheros binarios (BFD).

libiberty

libiberty es una colección de subrutinas usadas por varios programas GNU, incluidos getopt, obstack, strerror, strtol y strtoul.

libopcodes

libopcodes es una librería nativa para manejar mnemónicos y se usa durante la construcción de utilidades como objdump. Los mnemónicos son, en realidad, las versiones en texto legible de las instrucciones del procesador.

Dependencias

Para instalar Binutils−2.11.2 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: ar, as, ld, nm, ranlib, strip
diffutils: cmp
fileutils: chmod, cp, ln, ls, mkdir, mv, rm, rmdir, touch
flex: flex
gcc: cc, cc1, collect2, cpp0, gcc
glibc: ldconfig
grep: egrep, fgrep, grep
m4: m4
make: make
gawk: gawk
sed: sed
sh−utils: basename, echo, expr, hostname, sleep, true, uname
texinfo: install−info, makeinfo
textutils: cat, sort, tr, uniq

Instalando Bzip2−1.0.2

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  3 MB
```

Instalación de Bzip2

Instala Bzip2 ejecutando los siguientes comandos:

```
make CC="gcc -static" &&
make PREFIX=$LFS/usr install &&
cd $LFS/usr/bin &&
mv bzcat bunzip2 bzip2 bzip2recover bzless bzmore $LFS/bin
```

Aunque, estrictamente, no es una parte de un sistema LFS básico, merece la pena mencionar que se puede descargar un parche que permite al programa tar comprimir y descomprimir utilizando bzip2/bunzip2 fácilmente. Con el tar simple, tienes que utilizar construcciones como bzcat fichero.tar.bz2 | tar xv o tar −−use−compress−prog=bunzip2 −xvf fichero.tar.bz2 para utilizar bzip2 y bunzip2 con tar. Este parche añade la opción −j para que puedas desempaquetar un archivo Bzip2 con tar xvfj fichero.tar.bz2. Se mencionará después cómo aplicar este parche, cuando se instale el paquete Tar.

Explicación de los comandos

`make CC="gcc -static"`: Este es el método que utilizamos para decirle a gcc que queremos que bzip2 sea enlazado estáticamente.

Contenido de bzip2–1.0.1

Programas

bunzip2 (enlace a bzip2), bzcat (enlace a bzip2), bzip2 y bzip2recover

Descripciones

bunzip2

Bunzip2 descomprime ficheros que han sido comprimidos con bzip2.

bzcat

bzcat (o bzip2 –dc) descomprime todos los ficheros especificados hacia la salida estándar.

bzip2

bzip2 comprime ficheros usando el algoritmo de compresión de texto por ordenación de bloques Burrows–Wheeler y la codificación Huffman. La compresión es, en general, considerablemente superior a la obtenida por otros compresores más convencionales basados en el LZ77/LZ78 y se acerca al rendimiento de la familia de compresores estadísticos PPM.

bzip2recover

bzip2recover recupera datos de ficheros bzip2 dañados.

Librerías

libbz2.[a,so]

libbz2

libbz2 es la librería que implementa la compresión sin pérdidas por ordenación de bloques usando el algoritmo de Burrows–Wheeler.

Dependencias

Para instalar Bzip2–1.0.1 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib
fileutils: cp, ln, rm
gcc: cc1, collect2, cpp0, gcc
make: make

Instalando Diffutils–2.8

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  4 MB
```

Instalación de Diffutils

Cuando se instala Diffutils utilizando la librería glibc−2.1.x de tu sistema base, puede ser necesario usar un arreglo para evitar un conflicto en el nombre de una variable. Se pueden usar los siguientes comandos en este caso. Ten en cuenta que esas órdenes también pueden utilizarse para otras versiones de glibc, luego, si no estás seguro, entonces utiliza la primera versión.

```
export CPPFLAGS=-Dre_max_failures=re_max_failures2 &&
./configure --prefix=$LFS/usr --disable-nls &&
unset CPPFLAGS &&
make LDFLAGS=-static &&
make install
```

Si estás utilizando un versión de glibc más moderna (2.2.x), entonces puedes utilizar los siguientes comandos para instalar Diffutils:

```
./configure --prefix=$LFS/usr --disable-nls &&
make LDFLAGS=-static &&
make install
```

Explicación de los comandos

`CPPFLAGS=-Dre_max_failures=re_max_failures2:` La variable CPPFLAGS es leída por el programa cpp (PreProcesador de C). El valor de esta variable le dice al preprocesador que cambie cada aparición de re_max_failures que encuentre por re_max_failures2 antes de enviar el código fuente al compilador para su compilación. Este paquete tiene problemas en el enlazado estático en sistemas que ejecutan una versión más antigua de Glibc, y esta construcción soluciona ese problema.

Contenido de diffutils−2.7

Programas

cmp, diff, diff3 y sdiff

Descripciones

cmp y diff

Tanto cmp como diff comparan dos ficheros y muestran sus diferencias. Ambos programas tienen argumentos para comparar ficheros en diferentes situaciones.

diff3

La diferencia entre diff y diff3 es que diff compara 2 ficheros mientras diff3 compara 3.

sdiff

sdiff mezcla dos ficheros y muestra los resultados interactivamente.

Dependencias

Para instalar Diffutils−2.7 se necesita lo siguiente:

bash: sh
binutils: ld, as
diffutils: cmp
fileutils: chmod, cp, install, mv, rm
gcc: cc1, collect2, cpp0, gcc
grep: egrep, grep
make: make
sed: sed
sh−utils: date, hostname
textutils: cat, tr

Instalando Fileutils−4.1

```
Estimación del tiempo de construcción:      3 minutos
Estimación del espacio de disco requerido:  25 MB
```

Instalación de Fileutils

Los programas de un paquete fileutils enlazados estáticamente pueden causar violaciones de segmento en algunos sistemas si tu distribución tiene instalada la librería Glibc−2.2.3 o una versión posterior. Parece que también ocurre, en su mayoría, en máquinas que contienen un procesador AMD, aunque existen un caso o dos donde ha sido afectado un sistema Intel. Si tu sistema entra en esta categoría, prueba el siguiente arreglo.

Ten en cuenta que, en algunos casos, utilizar estos comandos de sed provocará problemas que impedirán compilar este paquete, incluso cuando tu sistema tenga un procesador AMD y una librería Glibc−2.2.3 (o superior) instalada. Si ese es tu caso, necesitarás borrar el directorio fileutils−4.1 y desempaquetarlo de nuevo antes de continuar. Creemos que esto puede pasar cuando tu distribución ha alterado de alguna forma la librería Glibc−2.2.3, pero los detalles no están disponibles por ahora.

Para reparar este paquete de forma que compile adecuadamente en máquinas AMD/Glibc−2.2.3, ejecuta los siguiente comandos. *No* intentes utilizar este arreglo si no tienes la librería Glibc−2.2.3 instalada. Es más que probable que provoque todo tipo de problemas a la hora de compilar.

```
cp lib/Makefile.in lib/Makefile.in.backup &&
sed -e 's/\(.*\)\(fopen-safer\.c \)\\/\1\2atexit.c \\/' \
   -e 's/\(.*\)\(idcache\$U\.\$.*\)\\/\1\2atexit$U.$(OBJEXT) \\/' \
   lib/Makefile.in.backup > lib/Makefile.in
```

Instala fileutils ejecutando los siguientes comandos:

```
./configure --disable-nls \
   --prefix=$LFS/usr --bindir=$LFS/bin &&
make LDFLAGS=-static &&
make install &&
cd $LFS/usr/bin &&
ln -sf ../../bin/install
```

Una vez hayas instalado fileutils, puedes comprobar si se ha evitado el problema de la violación de segmento ejecutando **$LFS/bin/ls**. Si esto funciona, entonces está solucionado. Si no, deberás rehacer la instalación utilizando los comandos de sed si no los usaste, o sin ellos, si en efecto los utilizaste.

Explicación de los comandos

cp lib/Makefile.in lib/Makefile.in.backup : Este comando se ejecuta para tener una copia de seguridad del fichero que vamos a cambiar.

```
cp lib/Makefile.in lib/Makefile.in.backup && sed -e
's/\(.*\)\(fopen-safer\.c \)\\/\1\2atexit.c \\/' \    -e
's/\(.*\)\(idcache\$U\.\$.*\)\\/\1\2atexit$U.$(OBJEXT) \\/' \
   lib/Makefile.in.backup > lib/Makefile.in:
```

Esto se utiliza para arreglar un problema con la construcción estática de fileutils en sistemas con la librería glibc 2.2.3. Si no se hace así, existe la posibilidad de que todos los programas de fileutils provoquen violaciones de segmento una vez entremos en el entorno chroot en el capítulo 6.

Contenido de fileutils–4.1

Programas

chgrp, chmod, chown, cp, dd, df, dir, dircolors, du, install, ln, ls, mkdir, mkfifo, mknod, mv, rm, rmdir, shred, sync, touch y vdir

Descripciones

chgrp

chgrp cambia el grupo de cada fichero al grupo especificado, que puede ser tanto el nombre de un grupo como su identificador numérico.

chmod

chmod cambia los permisos de un fichero de acuerdo con el modo, que puede ser tanto una representación simbólica de los cambios a hacer, o un número octal que representa el patrón de bits de los nuevos permisos.

chown

chown cambia el usuario y/o el grupo al que pertenece un fichero.

cp

cp copia ficheros de un lugar a otro.

dd

dd copia un fichero (por defecto, de la entrada estándar a la salida estándar) con un tamaño de bloque definido por el usuario, mientras, opcionalmente, realiza conversiones en él.

df

df muestra la cantidad de espacio disponible en los sistemas de ficheros a los que pertenece cada fichero que se le pasa como argumento. Si no se indica ningún fichero, se muestra el espacio disponible en todos los sistemas de ficheros montados actualmente

dir, ls y vdir

dir y vdir son versiones de ls con formatos de salida diferentes. Estos programas listan cada fichero o directorio especificado. El contenido de los directorios se lista alfabéticamente. Para ls, los ficheros se listan en columnas ordenados verticalmente si la salida estándar es un terminal; en otro caso se listan uno por línea. Para dir, los ficheros se listan por defecto en columnas ordenados verticalmente. Para vdir, los ficheros se listan, por defecto, en formato largo.

dircolors

dircolors imprime comandos para modificar la variable de entorno LS_COLOR. La variable LS_COLOR se usa para cambiar el esquema de color por defecto de ls y de herramientas relacionadas.

du

du muestra la cantidad de espacio en disco usado por cada fichero o directorio pasado como argumento.

install

install copia ficheros y establece sus permisos y, si es posible, su propietario y grupo.

ln

ln crea enlaces duros o blandos (simbólicos) entre ficheros.

mkdir

mkdir crea directorios con el nombre indicado.

mkfifo

mkfifo crea una tubería (FIFO) con un nombre dado.

mknod

mknod crea una tubería (FIFO), un fichero especial de caracteres o un fichero especial de bloques con el nombre indicado.

mv

mv mueve ficheros de un directorio a otro o renombra ficheros, dependiendo de los argumentos que se le pasen.

rm

rm elimina ficheros o directorios.

rmdir

rmdir elimina directorios, si están vacíos.

shred

shred borra un fichero de forma segura, sobreescribiéndolo para que su contenido no pueda ser recuperado.

sync

sync guarda los bloques modificados en disco y actualiza el superbloque.

touch

touch cambia las fechas de modificación o acceso del fichero especificado, poniéndole la fecha actual. Si el fichero no existe crea uno vacío.

Dependencias

Para instalar Fileutils−4.1 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm, rmdir
gettext: msgfmt, xgettext
gcc: cc, cc1, collect2, cpp0, gcc
grep: egrep, fgrep, grep
make: make
perl: perl
sed: sed
sh−utils: basename, echo, expr, hostname, sleep, uname
texinfo: install−info
textutils: cat, tr

Instalando Gawk−3.1.0

```
Estimación del tiempo de construcción:      2 minutos
Estimación del espacio de disco requerido:  12 MB
```

Instalación de Gawk

Atención: NO ejecutes **`make uninstall`** en este paquete si has aplicado el arreglo de *sed* para cambiar la definición del directorio `libexec`. La regla *uninstall* del fichero `Makefile` ejecuta un comando como **`rm -rf <libexecdir>/*`** Como hemos cambiado el directorio `libexec` a `/usr/bin` se ejecutará **`rm -rf /usr/bin/*`**

Al instalar Gawk utilizando la librería glibc−2.1.x en tu sistema base, puede ser necesario usar un arreglo para prevenir un conflicto en el nombre de las variables. En este caso, se pueden utilizar los comandos siguientes. Ten en cuenta que estos comandos pueden ser utilizados para otras versiones de glibc, por lo que, si no estás

seguro, usa entonces la primera versión.

```
cp awklib/Makefile.in awklib/Makefile.in.backup &&
sed -e '/^datadir/s/awk/gawk/' \
-e '/^libexecdir/s%/awk%%' awklib/Makefile.in.backup \
   > awklib/Makefile.in &&
export CPPFLAGS=-Dre_max_failures=re_max_failures2 &&
./configure --prefix=$LFS/usr --disable-nls \
   --libexecdir=$LFS/usr/bin &&
unset CPPFLAGS &&
make LDFLAGS=-static &&
make install
```

Si estás utilizando una versión más moderna de glibc (2.2.x), puedes utilizar los siguientes comandos para instalar Gawk:

```
cp awklib/Makefile.in awklib/Makefile.in.backup &&
sed -e '/^datadir/s/awk/gawk/' \
-e '/^libexecdir/s%/awk%%' awklib/Makefile.in.backup \
   > awklib/Makefile.in &&
./configure --prefix=$LFS/usr --disable-nls \
   --libexecdir=$LFS/usr/bin &&
make LDFLAGS=-static &&
make install
```

Contenido de gawk–3.1.0

Aún no se ha comprobado

Dependencias

Para instalar Gawk–3.1.0 se necesita lo siguiente:

Dependencias no comprobadas todavía

Instalando GCC–2.95.3

```
Estimación del tiempo de construcción:      22 minutos
Estimación del espacio de disco requerido:  168 MB
```

Instalación de GCC

Se sabe que este programa se comporta mal si cambias sus parámetros de optimización (incluyendo las opciones –march y –mcpu). Es mejor dejar a GCC tal como está, así que te recomendamos borrar CFLAGS, CXXFLAGS y otras variables/parámetros como éstos que podrían cambiar la optimización con la que viene por defecto.

Instala GCC ejecutando los siguientes comandos:

```
patch -Np1 -i ../gcc-2.95.3-2.patch &&
mkdir ../gcc-build &&
cd ../gcc-build &&
../gcc-2.95.3/configure --prefix=/usr --enable-languages=c,c++ \
```

```
   --disable-nls --disable-shared --enable-threads=posix &&
make BOOT_LDFLAGS=-static bootstrap &&
make prefix=$LFS/usr install &&
cd $LFS/lib &&
ln -sf ../usr/bin/cpp &&
cd $LFS/usr/lib &&
ln -sf ../bin/cpp &&
cd $LFS/usr/bin &&
ln -sf gcc cc &&
rmdir $LFS/usr/*-gnu/include &&
rmdir $LFS/usr/*-gnu
```

Explicación de los comandos

`patch -Np1 -i ../gcc-2.95.3-2.patch:` Este nuevo parche soluciona el manejo incorrecto de símbolos débiles, la sobreoptimización de las llamadas a estos símbolos débiles, y modifica atexit y el símbolo __dso_handle requerido para un funcionamiento correcto de atexit.

`make BOOT_LDFLAGS=-static:` Esto es el equivalente al "make LDFLAGS=−static" que usamos con otros paquetes para compilarlos estáticamente

`--prefix=/usr:` Esto NO es un error tipográfico. GCC establece directamente algunas rutas durante la compilación y por eso necesitamos pasar /usr como prefijo a ./configure. Pasamos el prefijo real de instalación posteriormente a "make install".

`--enable-languages=c,c++:` Con esto sólo se construyen los compiladores C y C++ y no el resto de compiladores disponibles, que no son habitualmente utilizados. Si se necesitan esos otros compiladores, el parámetro −−enable−languages debe omitirse.

`--enable-threads=posix:` Esto activa el manejo de excepciones C++ en código multi−hilo.

`ln -sf ../usr/bin/cpp:` Esto crea el enlace simbólico $LFS/lib/cpp. Algunos paquetes intentan encontrar cpp explícitamente en /lib.

`ln -sf ../bin/cpp:` Esto crea el enlace simbólico $LFS/usr/lib/cpp para los paquetes que esperan encontrar cpp en /usr/lib.

`rmdir $LFS/usr/*-gnu/include` y **`rmdir $LFS/usr/*-gnu`**: Estos directorios son creados vacíos por GCC y no nos sirven para ningún propósito concreto. Están relacionados con la compilación cruzada pero no son de aplicación para nosotros y está considerado como un fallo de GCC que no podemos corregir, sobre todo porque acaban siendo directorios vacíos. Por eso tenemos que eliminarlos manualmente.

Contenido de gcc−2.95.3

Programas

c++, c++filt, cc (enlace a gcc), cc1, cc1plus, collect2, cpp, cpp0, g++, gcc, gcov, protoize y unprotoize

Descripciones

cc, cc1, cc1plus, gcc

Estos programas forman el compilador de C. Un compilador convierte el código fuente en formato de texto a un formato que un ordenador pueda entender. Después de que un fichero de código fuente es compilado en un fichero objeto, un enlazador creará un fichero ejecutable a partir de uno o más de estos ficheros objeto generados por el compilador.

c++, cc1plus, g++

Estos programas forman el compilador de C++; El equivalente de cc, gcc, etc.

c++filt

c++filt se usa para la decodificación (demangle) de los símbolos de C++.

collect2

No hay descripción disponible.

cpp, cpp0

cpp preprocesa los ficheros fuente, como en la inclusión del contenido de los ficheros de cabecera en los ficheros fuente. Es buena idea no hacer esto manualmente para ahorrar un montón de tiempo. Basta con insertar una línea del tipo #include <fichero>. El preprocesador inserta el contenido de ese fichero en el fichero fuente. Esta es una de las cosas que el preprocesador hace.

gcov

No hay descripción disponible.

protoize

Programa adicional que convierte funciones o definiciones del antiguo estilo anterior al ANSI al nuevo estilo de prototipos ANSI C. (un fichero por defecto para saber como va esto es `/usr/lib/gcc-lib/<arch>/<versión>/SYSCALLS.c.X`)

unprotoize

Programa adicional que convierte prototipos hechos con protoize a su estilo antiguo original (solo funciona si antes fueron generados con protoize).

Librerías

libgcc.a, libiberty.a, libstdc++.[a,so]

libgcc

libgcc.a es un fichero de soporte en tiempo de ejecución para gcc. Con frecuencia, en la mayoría máquinas, libgcc.a no es realmente necesaria.

libiberty

libiberty es una colección de subrutinas usadas por muchos programas GNU, incluidos getopt, obstack, strerror, strtol y strtoul.

libstdc++

libstdc++ es la librería C++. Es utilizada por programas escritos en C++ y contiene funciones que son usadas frecuentemente por esos programas. De esta forma el programador no necesita escribir ciertas funciones (como la escritura de una cadena de texto en pantalla) desde el principio cada vez que crea un programa.

Dependencias

Para instalar GCC−2.95.3 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, nm, ranlib
diffutils: cmp
fileutils: chmod, cp, ln, ls, mkdir, mv, rm, touch
find: find
gcc: cc, cc1, collect2, cpp0, gcc
grep: egrep, grep
make: make
patch: patch
sed: sed
sh−utils: basename, dirname, echo, expr, hostname, sleep, true, uname
tar: tar
texinfo: install−info, makeinfo
textutils: cat, tail, tr

Instalando Grep−2.5

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  4 MB
```

Instalación de Grep

Al instalar Grep utilizando la librería glibc−2.1.x en tu sistema base, puede ser necesario usar un arreglo para prevenir un conflicto en el nombre de las variables. En este caso, se pueden utilizar los comandos siguientes. Ten en cuenta que estos comandos pueden ser utilizados para otras versiones de glibc, por lo que, si no estás seguro, usa entonces la primera versión.

```
export CPPFLAGS=-Dre_max_failures=re_max_failures2 &&
./configure --prefix=$LFS/usr --disable-nls --bindir=$LFS/bin &&
unset CPPFLAGS &&
make LDFLAGS=-static &&
make install
```

Si estás utilizando una versión más moderna de glibc (2.2.x), puedes utilizar los siguientes comandos para instalar Grep:

```
./configure --prefix=$LFS/usr --disable-nls \
   --bindir=$LFS/bin &&
make LDFLAGS=-static &&
make install
```

Contenido de grep–2.4.2

Programas

egrep, fgrep y grep

Descripciones

egrep

egrep muestra las líneas de un fichero que coincidan con una determinada expresión regular extendida.

fgrep

fgrep muestra las líneas de un fichero que coincidan con una lista de cadenas fijas, separadas por saltos de línea, cualquiera de las cuales puede ser coincidente.

grep

grep muestra las líneas de un fichero que coincidan con una expresión regular.

Dependencias

Para instalar Grep–2.4.2 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: as, ld
diffutils: cmp
fileutils: chmod, install, ls, mkdir, mv, rm
gettext: msgfmt, xgettext
gcc: cc, cc1, collect2, cpp0, gcc
glibc: getconf
grep: egrep, fgrep, grep
m4: m4
make: make
gawk: gawk
sed: sed
sh–utils: basename, echo, expr, hostname, sleep, uname
texinfo: install–info, makeinfo
textutils: cat, tr

Instalando Gzip–1.2.4a

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  2 MB
```

Instalación de Gzip

Antes de instalar Gzip, puede ser necesario aplicar un parche para evitar un conflicto en el nombre de las variables en sistemas con la librería Glibc–2.0, cuando compilas y enlazas estáticamente. Sólo se necesita si tu sistema base utiliza Glibc–2.0. De todas formas, es seguro aplicar el parche incluso si estás ejecutando una versión de glibc diferente, luego, si no estás seguro, lo mejor es aplicarlo.

Aplica el parche ejecutando los siguientes comandos:

```
patch -Np1 -i ../gzip-1.2.4a.patch
```

Instala Gzip ejecutando los siguientes comandos:

```
./configure --prefix=$LFS/usr &&
make LDFLAGS=-static &&
make install &&
cp $LFS/usr/bin/gunzip $LFS/usr/bin/gzip $LFS/bin &&
rm $LFS/usr/bin/gunzip $LFS/usr/bin/gzip
```

Explicación de los comandos

`cp $LFS/usr/bin/gunzip $LFS/usr/bin/gzip $LFS/bin && rm $LFS/usr/bin/gunzip $LFS/usr/bin/gzip`: La razón por la que no utilizamos simplemente "mv" para mover los ficheros a su nueva localización es porque gunzip es un enlace duro a gzip. En distribuciones más antiguas, no puedes mover un enlace duro a otra partición (y es muy posible que $LFS y $LFS/usr sean particiones separadas). Con distribuciones más recientes esto no es un problema. Si ejecutas mv para mover enlaces duros entre particiones lo único que hará será una copia ("cp") normal, y descartará el enlace duro. Pero no podemos asumir que cada distribución original tiene un núcleo suficientemente nuevo ni que fileutils se comporte de esa manera.

Contenido de gzip–1.2.4a

Programas

gunzip (enlace a gzip), gzexe, gzip, uncompress (enlace a gunzip), zcat (enlace a gzip), zcmp, zdiff, zforce, zgrep, zmore y znew

Descripción

gunzip, uncompress

gunzip y uncompress descomprimen ficheros que hayan sido comprimidos con gzip.

gzexe

gzexe permite comprimir ficheros ejecutables que automáticamente se descomprimen y ejecutan al ser lanzados (con una penalización en el rendimiento).

gzip

gzip reduce el tamaño de los ficheros usando codificación Lempel−Ziv (LZ77)

zcat

zcat descomprime una lista de ficheros en su línea de comandos o su entrada estándar, escribiendo los datos descomprimidos en la salida estándar.

zcmp

zcmp invoca al programa cmp en ficheros comprimidos.

zdiff

zdiff invoca al programa diff en ficheros comprimidos.

zforce

zforce fuerza la extensión .gz en todos los ficheros gzip para que gzip no los comprima dos veces. Esto puede ser útil para ficheros con el nombre truncado después de una transferencia de ficheros.

zgrep

zgrep invoca al programa grep en ficheros comprimidos.

zmore

zmore es un filtro que permite examinar ficheros comprimidos de texto plano pantalla a pantalla en un terminal emulado (similar al programa more).

znew

znew recomprime ficheros en formato .Z (compress) al formato .gz (gzip).

Dependencias

Para instalar Gzip−1.2.4a se necesita lo siguiente:

bash: sh
binutils: as, ld, nm
fileutils: chmod, cp, install, ln, mv, rm
gcc: cc1, collect2, cpp, cpp0, gcc
grep: egrep, grep
make: make
sed: sed

sh–utils: hostname
textutils: cat, tr

Instalando Linux Kernel–2.4.18

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  132 MB
```

Instalación del Núcleo Linux

No compilaremos todavía una nueva imagen del núcleo (kernel). Eso lo haremos después de que acabemos la instalación del software del sistema base de este capítulo. Pero como ciertos programas necesitan los ficheros de cabecera (headers) del núcleo, vamos a desempaquetar el archivo del núcleo ahora e instalar esos ficheros y así poder compilar los paquetes que necesitan el núcleo.

Los ficheros de configuración del núcleo se crean ejecutando los siguientes comandos:

```
make mrproper &&
make include/linux/version.h &&
make symlinks &&
mkdir $LFS/usr/include/asm &&
cp include/asm/* $LFS/usr/include/asm &&
cp -R include/linux $LFS/usr/include &&
touch $LFS/usr/include/linux/autoconf.h
```

Explicación de los comandos

`make mrproper`: Esto asegurará que el árbol de directorios del núcleo está absolutamente limpio. Lo hacemos porque el equipo de desarrollo del núcleo recomienda hacerlo antes de *cada* compilación del núcleo, y porque no debemos confiar en que el árbol de directorios del código fuente está automáticamente limpio tras desempaquetarlo.

`make include/linux/version.h` y **`make symlinks`**: Esto crea el fichero `include/linux/version.h`, así como el enlace simbólico `include/asm`

`mkdir $LFS/usr/include/asm` y **`cp include/asm/* $LFS/usr/include/asm`**: Esto copia los ficheros de cabecera del núcleo dependientes de la plataforma en `$LFS/usr/include/asm`

`cp -R include/linux $LFS/usr/include`: Este comando copia los ficheros de cabecera del núcleo independientes de la plataforma en `$LFS/usr/include`

`touch $LFS/usr/include/linux/autoconf.h`: Algunos ficheros de cabecera del núcleo incluyen el fichero `autconf.h` pero, como fuera del árbol de directorios de código fuente de Linux ese fichero no tiene significado, sólo creamos un fichero vacío para no tener errores de compilación cuando haya una dependencia de otro fichero de cabecera del núcleo.

El porqué de copiar las cabeceras del núcleo y no hacer enlaces simbólicos a ellas.

En el pasado, era una práctica común enlazar simbólicamente los directorios /usr/include/linux y asm con /usr/src/linux/include/linux y asm, respectivamente. Esta es una *mala* idea, como señala este extracto de un

mensaje de Linus Torvalds a la Lista de Correo del Núcleo Linux:

```
Sugeriría que la gente que compile núcleos nuevos debe:

 - no tener un sólo enlace simbólico a la vista (excepto el que crea la
   misma construcción del núcleo, el enlace simbólico llamado
   "linux/include/asm", que sólo se usa para la compilación interna del
   mismo núcleo).

Y sí, esto es lo que yo hago. Mi /usr/src/linux todavía contiene los
ficheros de cabecera del antiguo 2.2.13, aunque no he ejecutado un
núcleo 2.2.13 desde hace _muucho_ tiempo. Pero esas fueron las cabeceras
con las que fue compilada glibc, por lo que esas cabeceras son las que
coinciden con los ficheros objeto de la librería.

Y este es, de hecho, el entorno que se ha sugerido en, al menos, los
últimos cinco años. No sé por qué el asunto del enlace simbólico sigue
coleando, como un mal zombi. Casi cada distribución todavía tiene ese
enlace simbólico roto,  y la gente todavía recuerda que el código fuente
de linux debe ir en "/usr/src/linux" aunque no ha sido cierto desde hace
_muucho_ tiempo.
```

La parte relevante de este mensaje es donde él afirma que las cabeceras deberían ser con las que *fue compilada glibc*. Estas son las cabeceras que deberían permanecer accesibles y por ello, copiándolas, nos aseguramos de que seguimos esa pauta. Fíjate también en que, como no tienes esos enlaces simbólicos, es adecuado tener el código fuente del núcleo en `/usr/src/linux`.

Contenido de kernel–2.4.18

Ficheros de soporte

El núcleo Linux y los ficheros de cabecera (headers) del núcleo Linux

Descripciones

núcleo linux

El núcleo Linux es el corazón de todo sistema Linux. Es lo que da la marca Linux. Cuando se enciende un ordenador y se inicia un sistema Linux, el núcleo es lo primero que se carga. El núcleo inicializa los componentes hardware del sistema, como los puertos serie, puertos paralelo, tarjetas de sonido, tarjetas de red, controladores IDE, controladores SCSI y mucho más. En pocas palabras, el núcleo hace que el hardware esté disponible para que el software pueda ejecutarse.

ficheros de cabecera del núcleo linux

Estos son los ficheros que copiamos a /usr/include/{linux,asm} en el capítulo 5. Deben coincidir con la versión con la que glibc fue compilada. *No* deben reemplazarse cuando se actualiza el núcleo. Son esenciales para compilar muchos programas.

Dependencias

Para instalar Linux–2.4.18 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, nm, objcopy
fileutils: cp, ln, mkdir, mv, rm, touch
findutils: find, xargs
gcc: cc1, collect2, cpp0, gcc
grep: grep
gzip: gzip
make: make
gawk: awk
modutils: depmod, genksyms
net−tools: dnsdomainname, hostname
sed: sed
sh−utils: basename, date, expr, pwd, stty, uname, whoami, yes
textutils: cat, md5sum, sort, tail

Instalando Make−3.79.1

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  6 MB
```

Instalación de Make

Instala Make ejecutando los siguientes comandos:

```
./configure --prefix=$LFS/usr --disable-nls &&
make LDFLAGS=-static &&
make install
```

Durante la etapa make install verás este aviso:

```
chgrp: changing group of `/mnt/lfs/usr/bin/make': Operation not permitted
/mnt/lfs/usr/bin/make needs to be owned by group kmem and setgid;
otherwise the `-l' option will probably not work. You may need special
privileges to complete the installation of /mnt/lfs/usr/bin/make.


chgrp: cambiando grupo de `/mnt/lfs/usr/bin/make': Operación no permitida
/mnt/lfs/usr/bin/make necesita pertenecer al grupo kmem y activar el bit
SGID; de otra manera, la opción '-l' probablemente no funcionará. Puede
que necesites privilegios especiales para completar la instalación de
/mnt/lfs/usr/bin/make.
```

Puedes ignorar este aviso sin que conlleve ninguna consecuencia. make no necesita pertenecer al grupo kmem ni ejecutarse con ese ID de grupo (setgid) para que la opción *−l* funcione (que puedes usar para ordenar a make que no empiece nuevos trabajos cuando se alcance una determinada carga del sistema).

Contenido de make−3.79.1

Programas

make

Descripciones

make

make determina qué partes de un programa necesitan ser recompiladas, y lanza los comandos para hacerlo.

Dependencias

Para instalar Make–3.79.1 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: as, ld
diffutils: cmp
fileutils: chgrp, chmod, install, ls, mv, rm
gcc: cc, cc1, collect2, cpp0, gcc
glibc: getconf
grep: egrep, fgrep, grep
m4: m4
make: make
gawk: gawk
sed: sed
sh–utils: basename, echo, expr, hostname, sleep, uname
texinfo: install–info, makeinfo
textutils: cat, tr

Instalando Patch–2.5.4

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  2 MB
```

Instalación de Patch

Instala Patch ejecutando los siguientes comandos:

```
export CPPFLAGS=-D_GNU_SOURCE &&
./configure --prefix=$LFS/usr &&
unset CPPFLAGS &&
make LDFLAGS=-static &&
make install
```

Explicación de los comandos

`CPPFLAGS=-D_GNU_SOURCE:` Añadiendo **`-D_GNU_SOURCE`** a la variable CPPFLAGS antes de ejecutar ./configure se arregla la instalación de este paquete en las plataformas PPC y m68k (que nosotros sepamos).

Tampoco daña la compilación en otras plataformas (como x86) así que lo hacemos por defecto.

Contenido de patch–2.5.4

Programas

patch

Descripciones

patch

El programa patch modifica un fichero según lo indicado en un fichero de parche. Normalmente un fichero de parche es una lista creada por el programa diff que contiene instrucciones sobre cómo necesita ser modificado un fichero original. Patch se usa mucho para parchear el código fuente pues ahorra bastante tiempo y espacio. Imagina un paquete de 1MB de tamaño. La siguiente versión de ese paquete sólo cambia en dos ficheros con respecto a la primera versión. Se puede distribuir como un nuevo paquete entero de 1MB o sólo como un parche de 1KB con el que actualizar la primera versión para hacerla idéntica a la segunda. Por tanto, si la primera versión está todavía descargada, un parche evita hacer una segunda descarga larga.

Dependencias

Para instalar Patch–2.5.4 se necesita lo siguiente:

bash: sh
binutils: as, ld
diffutils: cmp
fileutils: chmod, install, mv, rm
gcc: cc, cc1, collect2, cpp0, gcc
glibc: getconf
grep: egrep, grep
make: make
sed: sed
sh–utils: echo, expr, hostname, uname
textutils: cat, tr

Instalando Sed–3.02

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  2 MB
```

Instalación de Sed

Cuando se instala Sed utilizando la librería glibc–2.1.x de tu sistema base, puede ser necesario usar un arreglo para evitar un conflicto en el nombre de una variable. Se pueden usar los siguientes comandos para ese caso. Ten en cuenta que esas órdenes también pueden utilizarse para otras versiones de glibc, luego, si no estás seguro, entonces utiliza la primera versión.

```
export CPPFLAGS=-Dre_max_failures=re_max_failures2 &&
./configure --prefix=$LFS/usr --bindir=$LFS/bin &&
```

```
unset CPPFLAGS &&
make LDFLAGS=-static &&
make install
```

Si estás utilizando un versión de glibc más moderna (2.2.x), entonces puedes utilizar los siguientes comandos para instalar Sed:

```
./configure --prefix=$LFS/usr --bindir=$LFS/bin &&
make LDFLAGS=-static &&
make install
```

Contenido de sed–3.02

Programas

sed

Descripciones

sed

sed es un editor de flujo. Un editor de flujo se usa para realizar transformaciones básicas de texto en un flujo de entrada (un fichero o una tubería).

Dependencias

Para instalar Sed–3.02 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, install, ls, mv, rm
gcc: cc1, collect2, cpp0, gcc
glibc: getconf
grep: egrep, fgrep, grep
m4: m4
make: make
gawk: gawk
sed: sed
sh–utils: echo, expr, hostname, sleep
texinfo: install–info, makeinfo
textutils: cat, tr

Instalando Sh–utils–2.0

```
Estimación del tiempo de construcción:      2 minutos
Estimación del espacio de disco requerido:  23 MB
```

Instalación de Sh–utils

Antes de instalar Sh–utils, puede ser necesario aplicarle un parche para evitar un conflicto de nombres de variables con ciertas versiones de Glibc (normalmente glibc–2.1.x), cuando se compila sh–utils estáticamente. De todas formas, es recomendable aplicar el parche incluso si tienes una versión diferente de glibc, de modo que si no estás seguro, es mejor que lo apliques.

Aplica el parche ejecutando el siguiente comando:

```
patch -Np1 -i ../sh-utils-2.0.patch
```

Instala Sh–utils ejecutando los siguientes comandos:

```
./configure --prefix=$LFS/usr --disable-nls &&
make LDFLAGS=-static &&
make install &&
cd $LFS/usr/bin &&
mv basename date echo false hostname $LFS/bin &&
mv pwd sleep stty test true uname $LFS/bin &&
mv chroot ../sbin
```

Cuando ejecutes make install verás el siguiente mensaje de aviso:

```
WARNING: insufficient access; not installing su
NOTE: to install su, run 'make install-root' as root


AVISO: privilegios insuficientes; no se instalará su
NOTA: para instalar su, ejecuta 'make install-root' como usuario root
```

Puedes ignorar este mensaje de aviso. Necesitas entrar al sistema como root para instalar su de la manera que sh–utils quiere hacerlo, es decir, activando el bit SUID para ejecutarse como administrador (root). Como no necesitamos su durante el capítulo 6, y se instalará correctamente cuando reinstalemos sh–utils en dicho capítulo, puedes hacer como si no hubieras visto el mensaje.

Contenido de sh–utils–2.0

Programas

basename, chroot, date, dirname, echo, env, expr, factor, false, groups, hostid, hostname, id, logname, nice, nohup, pathchk, pinky, printenv, printf, pwd, seq, sleep, stty, su, tee, test, true, tty, uname, uptime, users, who, whoami y yes

Descripciones

basename

basename elimina los directorios y las extensiones de los nombres de ficheros.

chroot

chroot ejecuta un comando o un intérprete de comandos (shell) interactivo dentro de un directorio raíz determinado.

date

date muestra la fecha y hora actual en un formato determinado o establece la fecha y hora del sistema.

dirname

dirname elimina los sufijos que no son directorios del nombre de un fichero.

echo

echo muestra una línea de texto.

env

env ejecuta un programa en un entorno modificado.

expr

expr evalúa expresiones.

factor

factor muestra los factores primos de los números especificados.

false

false siempre termina con un código de estado que indica un fallo.

groups

groups muestra los grupos a los que pertenece un usuario.

hostid

hostid muestra el identificador numérico (en hexadecimal) de la máquina actual.

hostname

hostname establece o muestra el nombre de la máquina actual.

id

id muestra el identificador de usuario (UID) real y efectivo y el identificador de grupo (GID) real y efectivo de un usuario.

logname

logname muestra el nombre de acceso (login name) del usuario actual.

nice

nice ejecuta un programa con una prioridad distinta.

nohup

nohup ejecuta un comando que no se interrumpe cuando se cierra la sesión, y cuya salida no se envía a un terminal.

pathchk

pathchk comprueba si los nombres de ficheros son válidos o portables.

pinky

pinky es una utilidad parecida a finger que obtiene información sobre un determinado usuario.

printenv

printenv muestra todo o parte del entorno.

printf

printf formatea y muestra datos (de la misma manera que la función printf de C).

pwd

pwd muestra el nombre del directorio de trabajo actual.

seq

seq muestra números en un cierto rango y con un cierto incremento.

sleep

sleep establece un retardo durante un determinado instante de tiempo.

stty

stty cambia y muestra las opciones de configuración del terminal.

su

su ejecuta un intérprete de comandos (shell) con un identificador de usuario y de grupo diferentes.

tee

tee lee de la entrada estándar y escribe en la salida estándar y en ficheros.

test

test comprueba el tipo de los ficheros y compara valores.

true

true siempre termina con un código de estado que indica éxito.

tty

tty muestra el nombre de fichero del terminal conectado a la entrada estándar.

uname

uname muestra información del sistema.

uptime

uptime muestra cuanto tiempo hace que el sistema está en marcha.

users

users muestra los nombres de los usuarios conectados actualmente.

who

who muestra quién está conectado.

whoami

whoami muestra el identificador de usuario efectivo.

yes

yes muestra en pantalla una cadena de texto indefinidamente.

Dependencias

Para instalar Sh−utils−2.0 se ecesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, chown, install, ls, mv, rm
gettext: msgfmt, xgettext
gcc: cc, cc1, collect2, cpp0, gcc
glibc: getconf
grep: egrep, fgrep, grep
m4: m4

make: make
gawk: gawk
perl: perl
sed: sed
sh−utils: basename, echo, expr, hostname, sleep, uname
tar: tar
texinfo: install−info, makeinfo
textutils: cat, tr

Instalando Tar−1.13

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  7 MB
```

Instalación de Tar

Para poder usar ficheros bzip2 con tar, utiliza el parche que se encuentra en el servidor FTP de LFS. Este parche añadirá la opción −j a tar, que funciona de la misma forma que la opción −z (usada para los ficheros gzip).

Aplica el parche ejecutando el siguiente comando:

```
patch -Np1 -i ../tar-1.13.patch
```

Instala Tar ejecutando los siguientes comandos:

```
./configure --prefix=$LFS/usr --disable-nls \
   --libexecdir=$LFS/usr/bin --bindir=$LFS/bin &&
make LDFLAGS=-static &&
make install
```

Contenido de tar−1.13

Programas

rmt y tar

Descripciones

rmt

rmt es un programa utilizado por los programas dump y restore para manipular una unidad de cinta magnética mediante una conexión entre procesos.

tar

tar es un programa diseñado para almacenar y extraer ficheros de un archivo conocido como fichero tar.

Dependencias

Para instalar Tar−1.13 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, install, ls, mv, rm
gettext: msgfmt, xgettext
gcc: cc, cc1, collect2, cpp0, gcc
glibc: getconf
grep: egrep, fgrep, grep
m4: m4
make: make
gawk: gawk
net−tools: hostname
patch: patch
sed: sed
sh−utils: basename, echo, expr, sleep, uname
texinfo: install−info, makeinfo
textutils: cat, tr

Instalando Texinfo−4.1

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  11 MB
```

Instalación de Texinfo

Instala Texinfo ejecutando los siguientes comandos:

```
./configure --prefix=$LFS/usr --disable-nls &&
make LDFLAGS=-static &&
make install
```

Contenido de texinfo−4.0

Programas

info, install−info, makeinfo, texi2dvi y texindex

Descripciones

info

El programa info lee documentos Info, almacenados normalmente en el directorio /usr/share/info. Los documentos Info son como las páginas de manual, pero suelen contener algo más que una simple explicación de las opciones de un programa.

install–info

El programa install–info actualiza las entradas info. Cuando se ejecuta info, muestra una lista con los temas (es decir, los documentos info) disponibles. Pues bien, el programa install–info se usa para mantener esta lista. Si los ficheros info son eliminados manualmente, también es necesario eliminar el tema en el fichero índice, para lo cual se utiliza este programa. También se utiliza para lo contrario, es decir, cuando se añaden documentos info.

makeinfo

El programa makeinfo convierte documentos fuente Texinfo a varios formatos. Los formatos disponibles son: ficheros info, texto plano y HTML.

texi2dvi

El programa texi2dvi imprime documentos Texinfo.

texindex

El programa texindex se usa para ordenar ficheros índice de Texinfo.

Dependencias

Para instalar Texinfo–4.0 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, install, ln, ls, mkdir, mv, rm
gcc: cc1, collect2, cpp0, gcc
grep: egrep, fgrep, grep
make: make
sed: sed
sh–utils: basename, echo, expr, hostname, sleep
texinfo: makeinfo
textutils: cat, tr

Instalando Textutils–2.0

```
Estimación del tiempo de construcción:      2 minutos
Estimación del espacio de disco requerido:  24 MB
```

Instalación de Textutils

Instala Textutils ejecutando los siguientes comandos:

```
./configure --prefix=$LFS/usr --disable-nls &&
make LDFLAGS=-static &&
make install &&
mv $LFS/usr/bin/cat $LFS/usr/bin/head $LFS/bin
```

Contenido de textutils–2.0

Programas

cat, cksum, comm, csplit, cut, expand, fmt, fold, head, join, md5sum, nl, od, paste, pr, ptx, sort, split, sum, tac, tail, tr, tsort, unexpand, uniq y wc

Descripciones

cat

cat concatena ficheros o la entrada estándar en la salida estándar.

cksum

cksum muestra la suma de comprobación CRC y cuenta los bytes de un fichero.

comm

comm compara dos ficheros ordenados línea por línea.

csplit

csplit escribe partes de un fichero separadas por un determinado patrón en ficheros xx01, xx02, etc, y muestra el número de bytes de cada parte en la salida estándar.

cut

cut imprime en la salida estándar las partes seleccionadas de las líneas de un fichero.

expand

expand convierte las tabulaciones de un fichero en espacios, escribiendo en la salida estándar

fmt

fmt reformatea cada párrafo de un fichero, escribiendo en la salida estándar.

fold

fold reajusta la longitud de las líneas de un fichero (por defecto, la entrada estándar), escribiendo en la salida estándar.

head

head imprime en la salida estándar las primeras xx (10 por defecto) líneas de un fichero.

join

join une líneas de dos ficheros en un campo común.

md5sum

md5sum muestra o chequea sumas de comprobación MD5.

nl

nl escribe un fichero en la salida estándar, añadiendo números de línea.

od

od escribe en la salida estándar una representación inequívoca (por defecto en octal) de un fichero.

paste

paste escribe en la entrada estándar líneas formadas por las líneas de cada uno de los ficheros especificados, separadas por tabulaciones.

pr

pr pagina o muestra en columnas el texto de un fichero, para imprimirlo posteriormente.

ptx

ptx genera un índice permutado de los contenidos de un fichero.

sort

sort escribe en la salida estándar una concatenación de ficheros ordenados.

split

split divide un fichero en partes de tamaño fijo llamadas FICHEROaa, FICHERObb,...

sum

sum muestra la suma de comprobación y el número de bloques que ocupa un fichero.

tac

tac escribe un fichero o ficheros en la salida estándar, comenzando por la última línea.

tail

tail imprime en la salida estándar las últimas xx (10 por defecto) líneas de un fichero.

tr

tr convierte, altera y/o borra caracteres de la entrada estándar, escribiendo en la salida estándar.

tsort

tsort escribe una lista totalmente ordenada de acuerdo con el orden parcial de los ficheros especificados.

unexpand

unexpand convierte los espacios de un fichero en tabulaciones, escribiendo en la salida estándar.

uniq

uniq elimina las líneas duplicadas de un fichero ordenado.

wc

wc muestra el número de líneas, palabras y bytes de un fichero, y una línea con el total si se ha especificado más de uno.

Dependencias

Para instalar Textutils−2.0 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, install, ls, mv, rm
gettext: msgfmt, xgettext
gcc: cc, cc1, collect2, cpp0, gcc
glibc: getconf
grep: egrep, fgrep, grep
m4: m4
make: make
gawk: gawk
net−tools: hostname
perl: perl
sed: sed
sh−utils: basename, echo, expr, sleep, uname
tar: tar
texinfo: install−info, makeinfo
textutils: cat, tr

Creando los ficheros de contraseñas y grupos

Para que se reconozca al usuario y al grupo root se necesita que exista una entrada en los ficheros /etc/passwd y /etc/group. Además, junto al grupo root existen otros grupos que recomiendan y necesitan los paquetes. Estos grupos y sus identificadores (GID) no están bajo ningún estándar. LSB sólo recomienda la presencia del grupo bin con el identificador 1, así que puedes crear otros grupos e identificadores. Los paquetes bien desarrollados no dependen del identificador, sino que utilizan el nombre del grupo. De esta manera, no importa qué identificador le pongas a cada grupo. Aquí, aunque como ya hemos dicho, no hay un estándar para los grupos, vamos a crear los grupos que menciona el guión MAKEDEV (el cual crea los ficheros de los dispositivos en el directorio /dev).

Crearemos el fichero `$LFS/etc/passwd` ejecutando el siguiente comando:

```
echo "root:x:0:0:root:/root:/bin/bash" > $LFS/etc/passwd
```

Y crearemos el fichero `$LFS/etc/group` ejecutando el siguiente comando:

```
cat > $LFS/etc/group << "EOF"
root:x:0:
bin:x:1:
sys:x:2:
kmem:x:3:
tty:x:4:
tape:x:5:
daemon:x:6:
floppy:x:7:
disk:x:8:
lp:x:9:
dialout:x:10:
audio:x:11:
EOF
```

Copiando los ficheros de la antigua librería NSS

Si tu sistema Linux normal utiliza Glibc−2.0 necesitas copiar los ficheros de la librería NSS a tu partición LFS. Algunos programas con las librerías enlazadas estáticamente dependen de la librería NSS, en especial los programas que necesitan mirar los nombres de los usuarios, los identificadores de los usuarios (userid) y los identificadores de los grupos (groupid). Puedes comprobar qué versión de la librería de C utilizas en tu sistema simplemente ejecutando la librería de este modo:

```
/lib/libc.so.6
```

En la primera línea puedes ver la versión. Las líneas siguientes contienen información interesante. Si tienes instalada la versión Glibc−2.0.x copia los ficheros de la librería NSS ejecutando:

```
cp -av /lib/libnss* $LFS/lib
```

Montando el sistema de ficheros $LFS/proc

Para funcionar correctamente, algunos programas necesitan que el sistema de ficheros proc esté montado y disponible tras establecer la nueva raíz de nuestro sistema. No existe ningún problema para montar el sistema de ficheros proc (o cualquier otro sistema de ficheros) dos o más veces.

Si estás conectado como el usuario "lfs" debes cerrar la sesión y conectarte como administrador (root). La razón de esto es simple: únicamente el administrador está autorizado a montar sistemas de ficheros y a establecer una nueva raíz (chroot).

El sistema de ficheros proc se montará bajo el directorio $LFS/proc ejecutando el siguiente comando. Además haremos que los archivos pertenezcan (chown) al usuario y al grupo root mientras estemos en él (el resto de los archivos se les cambiará el propietario a root.root cuando comience el capítulo 6).

```
chown root.root $LFS/proc &&
mount proc $LFS/proc -t proc
```

Capítulo 6. Instalando los programas del sistema base

Introducción

La instalación de todos los paquetes es algo bastante sencillo y puedes pensar que probablemente sea más fácil y corto dar las instrucciones genéricas para la instalación de cada paquete y sólo extenderse en cómo instalar algo si cierto paquete necesita un método de instalación alternativo. Aunque esté de acuerdo en eso, he elegido dar las instrucciones completas para cada paquete. Se trata simplemente de evitar cualquier posible error o confusión.

Ahora puede ser un buen momento para darle un vistazo a la receta sobre optimización en http://hints.linuxfromscratch.org/hints/optimization.txt si planeas usar optimización en la compilación de los paquetes instalados en el siguiente capítulo. La optimización de la compilación puede hacer que un programa funcione rápido, pero también puede causar problemas de compilación. Si te encuentras con problemas después de haber usado optimización, siempre puedes intentarlo sin optimización para ver si el problema persiste.

Sobre los símbolos de depuración

Muchos programas y librerías son compilados por defecto con símbolos de depuración (opción –g de gcc).

Un programa compilado con símbolos de depuración permite al usuario lanzar un programa o librería a través de un depurador y la salida del depurador será amigable. Estos símbolos de depuración también agrandan el programa o librería de forma significativa.

Antes de que comiences a preguntarte si estos símbolos de depuración realmente producen una gran diferencia, aquí tienes algunas estadísticas. Úsalas para sacar tu propia conclusión.

- El binario Bash dinámico con símbolos de depuración: 1.2MB
- El binario Bash dinámico sin símbolos de depuración: 478KB
- /lib y /usr/lib (los ficheros de glibc y gcc) con símbolos de depuración: 87MB
- /lib y /usr/lib (los ficheros de glibc y gcc) sin símbolos de depuración: 16MB

Los tamaños varían dependiendo de qué compilador fue usado y qué versión de la librería C se usó para enlazar dinámicamente los programas, pero los resultados son similares si comparas los programas con y sin símbolos de depuración.

Para eliminar los símbolos de depuración de un binario (debe ser un binario a.out o ELF) ejecuta **`strip --strip-debug fichero`**. Pueden usarse comodines para eliminar los símbolos de depuración de múltiples ficheros (puede usarse algo como esto: **`strip --strip-debug $LFS/usr/bin/*`**). Mucha gente probablemente nunca use un depurador, por lo que eliminando estos símbolos se puede liberar una buena cantidad de espacio del disco.

Para tu comodidad, en el capítulo 9 se incluye un comando simple para eliminar todos los símbolos de depuración de todos los programas y librerías del sistema.

Puedes encontrar información adicional en la receta de optimización que hay en http://hints.linuxfromscratch.org/hints/optimization.txt .

Creando $LFS/root/.bash_profile

Cuando entremos en el entorno chroot en la siguiente sección querremos exportar una serie de variables de entorno en ese intérprete de comandos, como PS1, PATH y otras variables que es bueno establecer. Para eso crearemos el fichero $LFS/root/.bash_profile que será leído por bash cuando entremos en el entorno chroot.

Crea un nuevo fichero `$LFS/root/.bash_profile` ejecutando lo siguiente.

```
cat > $LFS/root/.bash_profile << "EOF"
# Inicio de /root/.bash_profile

PS1='\u:\w\$ '
PATH=/bin:/usr/bin:/sbin:/usr/sbin

export PS1 PATH

# Fin de /root/.bash_profile
EOF
```

La variable PS1 es una variable de entorno que controla el aspecto de la entrada de comandos. Lee la página de manual de bash para los detalles de cómo se construye esta variable. Pueden añadirse, a tu propia discreción, otras variables de entorno adicionales, alias, etc, que puedas necesitar y/o querer.

Entrando al entorno chroot

Es la hora de entrar en nuestro entorno chroot para instalar el resto de programas que necesitamos.

Ejecuta los siguientes comandos para entrar en el entorno chroot. A partir de este punto ya no es necesario usar la variable $LFS, porque todo lo que un usuario haga estará restringido a la partición LFS (aunque / sea realmente /mnt/lfs el intérprete de comandos no lo sabe).

```
cd $LFS &&
chroot $LFS /usr/bin/env -i HOME=/root \
   TERM=$TERM /bin/bash --login
```

La opción –i borrará todas las variables de entorno actuales mientras se esté dentro del entorno chroot y solamente se exportan las variables HOME y TERM. La construcción TERM=$TERM fijará la variable TERM dentro del chroot al mismo valor que fuera del chroot porque programas como vim y less la necesitan para funcionar correctamente. Si necesitas tener presentes otras variables, como CFLAGS o CXXFLAGS, necesitas volver a establecerlas de nuevo.

La razón por la que hacemos **`cd $LFS`** antes de lanzar el comando **`chroot`** es porque el antiguo paquete sh–utils tiene un programa chroot que no puede hacer el cambio de directorio por sí mismo, por lo que debemos ejecutarlo manualmente. Aunque esto no es un problema en muchas distribuciones modernas, tampoco es perjudicial y nos aseguramos de que el comando funciona para todo el mundo.

Ahora que estamos dentro del entorno chroot, podemos continuar instalando todos los programas del sistema base. Debes asegurarte de que todos los comandos que aparecen a continuación en este y los siguientes capítulos son lanzados dentro del entorno chroot. Si por alguna razón abandonas este entorno (tras un reinicio,

por ejemplo), por favor, recuerda montar de nuevo $LFS/proc y reentrar en el chroot antes de seguir con el libro.

Ten en cuenta que en la línea de entrada de comandos de bash pondrá: "I have no name!". Esto es normal porque Glibc no ha sido instalada todavía.

Cambiando el propietario de la partición LFS

Ahora que estamos en el chroot, es un buen momento para cambiar el propietario de todos los ficheros y directorios que se instalaron en el capítulo 5 y devolvérselos al administrador del sistema (root). Ejecuta los siguientes comandos para hacerlo:

```
cd / &&
chown 0.0 . proc &&
chown -R 0.0 bin boot dev etc home lib mnt opt root sbin tmp usr var
```

Dependiendo del sistema de ficheros que se crease en la partición LFS, es posible que tengas un directorio /lost+found. Si lo tienes, ejecuta:

```
chown 0.0 lost+found
```

Estos comandos cambiarán el propietario de la partición raíz y del directorio `/proc` a root, más todo lo que haya bajo los directorios mencionados en la segunda línea. En estos comandos usamos 0.0 en lugar del habitual root.root, porque el nombre de usuario root todavía no puede resolverse debido a que glibc aún no ha sido instalada.

Creando el enlace simbólico /etc/mtab

La siguiente cosa por hacer es crear un enlace simbólico que apunte de /etc/mtab a /proc/mounts. Esto se hace con el siguiente comando:

```
ln -s /proc/mounts /etc/mtab
```

Creando este enlace simbólico se evitan los problemas que pueden aparecer si / se monta en sólo lectura y la información en /etc/mtab es errónea (p.e. anticuada). Con la creación del enlace simbólico a /proc/mounts, nos aseguramos de que /etc/mtab esté siempre actualizado.

Ten en cuenta que para usar este enlace simbólico necesitas tener el soporte para el sistema de ficheros /proc compilado dentro del núcleo. Esto se incluye por defecto y no debe ser eliminado a menos que sepas *realmente* lo que estás haciendo. Muchas otras cosas aparte del enlace simbólico /etc/mtab dependen de que /proc esté presente. En resumen, asegúrate de que tienes soporte para el sistema de ficheros /proc en tu núcleo.

Instalando Glibc–2.2.5

```
Estimación del tiempo de construcción:      46 minutos
Estimación del espacio de disco requerido:  350 MB
```

Instalación de Glibc

Antes de instalar glibc, debes entrar al directorio glibc−2.2.5 y desempaquetar glibc−linuxthreads dentro del directorio glibc−2.2.5, no en /usr/src como normalmente debe hacerse.

Se sabe que este programa se comporta mal si cambias sus parámetros de optimización (incluyendo las opciones −march y −mcpu). Es mejor dejar a Glibc tal como está, así que te recomendamos borrar CFLAGS, CXXFLAGS y otras variables/parámetros como éstos que podrían cambiar la optimización con la que viene por defecto. Igualmente, no le pases la opción −−enable−kernel al guión ./configure. Se sabe que causa violaciones de segmento cuando otros paquetes como fileutils, make y tar son enlazados de nuevo con ella.

Básicamente, compilar Glibc de forma diferente a como el libro sugiere pone tu sistema en grave riesgo.

Instala Glibc ejecutando los siguientes comandos:

```
mknod −m 0666 /dev/null c 1 3 &&
touch /etc/ld.so.conf &&
cp malloc/Makefile malloc/Makefile.backup &&
sed 's%\$(PERL)%/usr/bin/perl%' malloc/Makefile.backup > malloc/Makefile &&
cp login/Makefile login/Makefile.backup &&
sed 's/root/0/' login/Makefile.backup > login/Makefile &&
mkdir ../glibc−build &&
cd ../glibc−build &&
../glibc−2.2.5/configure −−prefix=/usr \
   −−enable−add−ons −−libexecdir=/usr/bin &&
echo "cross−compiling = no" > configparms &&
make &&
make install &&
make localedata/install−locales &&
exec /bin/bash −−login
```

Una alternativa a **`make localedata/install−locales`** es instalar solamente aquellos locales que necesites o desees. Esto puede hacerse usando el comando localedef. Se puede encontrar más información sobre esto en el fichero INSTALL del árbol de glibc−2.2.5.

Durante la fase de configuración verás los siguientes mensajes de aviso

```
configure: warning:
*** These auxiliary programs are missing or too old: msgfmt
*** some features will be disabled.
*** Check the INSTALL file for required versions.

configure: aviso:
*** Este programa auxiliar no se ha encontrado o es demasiado antiguo: msgfmt
*** algunas características serán desactivadas.
*** Compruebe en el fichero INSTALL la versión requerida.
```

Que no se encuentre msgfmt (incluido en el paquete gettext que será instalado más adelante en este capítulo) no causa ningún problema. Se usa msgfmt para generar los ficheros binarios traducidos que se usan para hacer que el sistema "hable" en un idioma diferente. Como estos ficheros de traducción ya han sido generados para ti, no es necesario msgfmt. Solamente necesitas msgfmt si cambias los ficheros fuente de traducción (los ficheros `*.po` del subdirectorio `po`) lo cual requerirá regenerar los ficheros binarios.

Explicación de los comandos

`mknod -m 0666 /dev/null c 1 3`: Glibc necesita un dispositivo nulo para compilar correctamente. Todos los demás dispositivos se crearán en la siguiente sección.

`touch /etc/ld.so.conf` Uno de los pasos finales en la instalación de Glibc es lanzar ldconfig para actualizar la caché del cargador dinámico. Si este fichero no existe, la instalación se abortará con el error de que no puede leerlo; así que, simplemente, creamos un fichero vacío (el fichero vacío hará que Glibc por defecto use /lib y /usr/lib que es suficiente).

`sed 's%\$(PERL)%/usr/bin/perl%' malloc/Makefile.backup > malloc/Makefile`: Este comando de sed busca a través de `malloc/Makefile.backup` y convierte todas las ocurrencias de `$(PERL)` en `/usr/bin/perl`. La salida es entonces escrita en el original `malloc/Makefile.in` que es usado durante la configuración. Esto se hace porque Glibc no puede autodetectar perl debido a que no ha sido instalado todavía.

`sed 's/root/0' login/Makefile.backup > login/Makefile`: Este comando de sed reemplaza todas las ocurrencias de `root` en `login/Makefile.backup` con 0. Esto es porque todavía no tenemos Glibc en el sistema, así que los nombres de usuario no pueden ser resueltos a su identificador. Por tanto, reemplazamos el nombre de usuario root con su identificador de usuario 0.

`--enable-add-ons`: Esto activa los añadidos que hemos instalado con Glibc: linuxthreads

`--libexecdir=/usr/bin`: Esto causará que el programa pt_chown sea instalado en el directorio /usr/bin.

`echo "cross-compiling = no" > configparms`: Hacemos esto porque solamente estamos compilando para nuestro sistema. La compilación cruzada se usa, por ejemplo, para construir un paquete para Apple Power PC en un sistema Intel. La razón por la que Glibc piensa que hacemos compilación cruzada es porque no puede compilar un programa de prueba para determinarlo, así que automáticamente pasa por defecto a compilación cruzada. La compilación del programa de prueba falla porque Glibc no se ha instalado aún.

`exec /bin/bash`: Este comando lanzará una nueva instancia de bash que reemplazará al intérprete de comandos actual. Esto se hace para quitar el mensaje "I have no name!" del indicador de entrada de comandos, que estaba causado por la imposibilidad de bash de resolver un identificador de usuario a su nombre de usuario (que, a su vez, se debía a que no se encontraba instalada Glibc).

Contenido de glibc–2.2.5

Programas

catchsegv, gencat, getconf, getent, glibcbug, iconv, iconvconfig, ldconfig, ldd, lddlibc4, locale, localedef, mtrace, nscd, nscd_nischeck, pcprofiledump, pt_chown, rpcgen, rpcinfo, sln, sprof, tzselect, xtrace, zdump y zic

Descripciones

catchsegv

No hay descripción disponible.

gencat

gencat genera catálogos de mensajes.

getconf

No hay descripción disponible.

getent

getent obtiene entradas de una base de datos administrativa.

glibcbug

glibcbug crea un informe de fallos sobre glibc y lo envía a la dirección de correo electrónico de errores.

iconv

iconv realiza conversiones de los juegos de caracteres.

iconvconfig

iconvconfig crea un fichero de configuración para la carga rápida del módulo iconv.

ldconfig

ldconfig configura las asociaciones en tiempo de ejecución para el enlazador dinámico.

ldd

ldd muestra las librerías compartidas requeridas por cada programa o librería especificada en la línea de comandos.

lddlibc4

No hay descripción disponible.

locale

No hay descripción disponible.

localedef

localedef compila las especificaciones para locale.

mtrace

No hay descripción disponible.

nscd

nscd es un demonio que suministra una caché para las peticiones más comunes al servidor de nombres.

nscd_nischeck

No hay descripción disponible.

pcprofiledump

pcprofiledump vuelca la información generada por un perfilador.

pt_chown

pt_chown establece el propietario, grupo y permisos de acceso del pseudo−terminal esclavo correspondiente al pseudo−terminal maestro apuntado por el descriptor de ficheros "3". Este es el programa de ayuda para la función `grantpt'. No está pensado para ejecutarse directamente desde la línea de comandos.

rpcgen

No hay descripción disponible.

rpcinfo

No hay descripción disponible.

sln

sln enlaza simbólicamente un destino a una fuente. Está enlazado estáticamente, no necesitando enlazado dinámico. Por tanto, sln es útil para crear enlaces simbólicos a librerías dinámicas si, por alguna razón, el enlazador dinámico del sistema no funciona.

sprof

sprof lee y muestra datos de perfilado de objetos compartidos.

tzselect

tzselect pregunta al usuario información sobre la localización actual y muestra la descripción de la zona horaria resultante en la salida estándar.

xtrace

xtrace traza la ejecución de un programa mostrando la función actualmente ejecutada.

zdump

zdump es el visualizador de información de huso horario.

zic

zic es el compilador de la zona horaria.

Librerías

ld.so, libBrokenLocale.[a,so], libBrokenLocale_p.a, libSegFault.so, libanl.[a,so], libanl_p.a, libbsd−compat.a, libc.[a,so], libc_nonshared.a, libc_p.a, libcrypt.[a,so], libcrypt_p.a, libdl.[a,so], libdl_p.a, libg.a, libieee.a, libm.[a,so], libm_p.a, libmcheck.a, libmemusage.so, libnsl.a, libnsl_p.a, libnss_compat.so, libnss_dns.so, libnss_files.so, libnss_hesiod.so, libnss_nis.so, libnss_nisplus.so, libpcprofile.so, libpthread.[a,so], libpthread_p.a, libresolv.[a,so], libresolv_p.a, librpcsvc.a, librpcsvc_p.a, librt.[a,so], librt_p.a, libthread_db.so, libutil.[a,so] y libutil_p.a

Descripciones

ld.so

ld.so es el programa de ayuda para las librerías compartidas ejecutables.

libBrokenLocale, libBrokenLocale_p

No hay descripción disponible.

libSegFault

No hay descripción disponible.

libanl, libanl_p

No hay descripción disponible.

libbsd−compat

No hay descripción disponible.

libc, libc_nonshared, libc_p

Estos ficheros constituyen la librería C principal. La librería C es una colección de funciones usadas frecuentemente en programas. De esta forma un programador no necesita crear sus propias funciones para cada tarea individual. Las cosas más comunes, como mostrar una cadena en pantalla, están presentes y a disposición del programador.

La librería C (en realidad, casi todas las librerías) viene en dos formas: dinámica y estática. En resumen, cuando un programa usa una librería C estática, se copia el código de la librería C dentro del ejecutable. Cuando un programa usa una librería dinámica, el ejecutable no contiene el código de la librería pero, en su lugar, tiene una rutina que carga las funciones desde esa librería en el momento en el que se ejecuta. De esta forma disminuye de forma significativa el tamaño del programa. La documentación que acompaña a la librería C describe esto con más detalle, pues es demasiado complicado explicarlo aquí en dos o tres líneas.

libcrypt, libcrypt_p

libcrypt es la librería criptográfica.

libdl, libdl_p

No hay descripción disponible.

libg

No hay descripción disponible.

libieee

No hay descripción disponible.

libm, libm_p

libm es la librería matemática.

libmcheck

No hay descripción disponible.

libmemusage

No hay descripción disponible.

libnsl, libnsl_p

No hay descripción disponible.

libnss_compat, libnss_dns, libnss_files, libnss_hesiod, libnss_nis, libnss_nisplus

No hay descripción disponible.

libpcprofile

No hay descripción disponible.

libpthread, libpthread_p

No hay descripción disponible.

libresolv, libresolv_p

No hay descripción disponible.

librpcsvc, librpcsvc_p

No hay descripción disponible.

librt, librt_p

No hay descripción disponible.

libthread_db

No hay descripción disponible.

libutil, libutil

No hay descripción disponible.

Dependencias

Para instalar Glibc−2.2.5 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib, readelf
diffutils: cmp
fileutils: chmod, cp, install, ln, mknod, mv, mkdir, rm, touch
gcc: cc, cc1, collect2, cpp, gcc
grep: egrep, grep
gzip: gzip
make: make
gawk: gawk
sed: sed
sh−utils: date, expr, hostname, pwd, uname
texinfo: install−info, makeinfo
textutils: cat, cut, sort, tr

Creando los dispositivos (Makedev−1.4)

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  57 KB
```

Creando los dispositivos

Advertencia: el fichero MAKEDEV−1.4.bz2 que debes desempaquetar no es un archivo, por lo que no creará un directorio en el que debas entrar.

Crea los ficheros de dispositivos ejecutando los siguientes comandos:

```
cp MAKEDEV-1.4 /dev/MAKEDEV &&
cd /dev &&
chmod 754 MAKEDEV
```

Ahora, dependiendo de si vas a usar o no devpts, puedes usar uno o dos comandos:

Si no piensas usar devpts, ejecuta:

```
./MAKEDEV -v generic
```

Si piensas usar devpts, entonces ejecuta:

```
./MAKEDEV -v generic-nopty
```

Ten en cuenta que, si no estás seguro, es mejor usar el comando **./MAKEDEV -v generic** con lo que te asegurarás de tener todos los dispositivos que necesites. Sin embargo, si estás seguro de que vas a usar devpts, con el otro comando te aseguras de que no se crearán un grupo de dispositivos que no vas a necesitar.

MAKEDEV creará del hda[1-20] al hdh[1-20] y semejantes, pero ten en mente que no serás capaz de usar todos esos dispositivos debido a la limitación del núcleo en cuanto al número máximo de particiones manejables.

Explicación de los comandos

./MAKEDEV -v generic: Esto crea los dispositivos genéricos. Normalmente, estos dispositivos son todos los que se necesitan. Es posible que no encuentres algún dispositivo especial necesario para configurar tu hardware. Créalo con ./MAKEDEV -v <device>. La opción **generic-nopty** hace un trabajo similar pero se salta algunos dispositivos que no son necesarios si usas devpts.

Contenido de MAKEDEV-1.4

Programas

MAKEDEV

Descripciones

MAKEDEV

MAKEDEV es un guión que puede ayudar en la creación de los ficheros de dispositivos estáticos necesarios, que usualmente residen en el directorio /dev. Más información sobre los ficheros de dispositivos puede encontrase dentro de las fuentes del núcleo en `Documentation/devices.txt`.

Dependencias

Para instalar MAKEDEV-1.4 se necesita lo siguiente:

bash: sh
fileutils: chmod, chown, cp, ln, mknod, mv, rm
grep: grep
sh-utils: expr, id

Instalando Man-pages-1.48

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  5 MB
```

Instalación de Man-pages

Instala Man-pages ejecutando los siguientes comandos:

```
make install
```

Contenido de manpages–1.47

Ficheros de soporte

Varias páginas de manual que no vienen con los paquetes.

Descripciones

páginas de manual

Un ejemplo de las páginas de manual incluidas son las que describen todas las funciones C y C++, algunos ficheros de /dev y otras cosas.

Dependencias

Para instalar Man–pages–1.47 se necesita lo siguiente:

bash: sh
fileutils: install
make: make

Instalando Findutils–4.1

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  3 MB
```

Instalando Findutils

Antes de instalar Findutils debe desempaquetarse el parche findutils.

Instala Findutils ejecutando los siguientes comandos:

```
patch -Np1 -i ../findutils-4.1.patch &&
./configure --prefix=/usr &&
make &&
make libexecdir=/usr/bin install
```

Notas sobre cumplimiento del estándar FHS

Por defecto, la base de datos de updatedb se encuentra en el directorio /usr/var. Si quieres respetar las normas del FHS deberás utilizar otra localización. Los siguientes comandos utilizan el fichero de base de datos `/var/lib/misc/locatedb` que cumple con el FHS.

```
patch -Np1 -i ../findutils-4.1.patch &&
./configure --prefix=/usr &&
make localstatedir=/var/lib/misc &&
make localstatedir=/var/lib/misc libexecdir=/usr/bin install
```

Explicación de los comandos

`patch -Np1 -i ../findutils-4.1.patch`: Este parche es para arreglar algunos errores de compilación, impidiendo un conflicto de variables y cambiando alguna sintaxis equivocada.

Contenido de findutils–4.1

Programas

bigram, code, find, frcode, locate, updatedb y xargs

Descripciones

bigram

bigram se usa junto con code para generar las bases de datos de locate en el formato antiguo. Para saber más de estos tres programas, lea la página de manual locatedb.5.

code

code es el antecesor de frcode. Se usaba en las bases de datos de formato antiguo.

find

El programa find busca los ficheros de una jerarquía de directorios que cumplan un cierto criterio. Si no se especifica un criterio de búsqueda, lista todos los ficheros del directorio actual y de los subdirectorios.

frcode

updatedb lanza el programa frcode para comprimir la lista de ficheros usando "front–compression", que reduce el tamaño de la base de datos en un factor de 4 o 5.

locate

Locate busca en una base de datos que contiene todos los ficheros y directorios de un sistema de ficheros. Este programa lista los ficheros y directorios de la base de datos que cumplan cierto criterio. Si un usuario busca un fichero, este programa buscará en la base de datos y le dirá dónde están ubicados exactamente esos ficheros. Esto sólo es válido si la base de datos de locate se encuentra actualizada. En otro caso mostrará información anticuada.

updatedb

El programa updatedb actualiza la base de datos de locate. Explora por completo el sistema de ficheros (incluidos otros sistemas de ficheros que se encuentren montados a no ser que se le indique lo contrario) e inserta todos los directorios y ficheros que encuentre en la base de datos que usa locate para recuperar dicha información. Es una buena costumbre actualizar la base de datos una vez al día para obtener información correcta cuando se necesite.

xargs

El comando xargs aplica un comando a una lista de ficheros. Si se necesita aplicar el mismo comando sobre múltiples ficheros, puede crearse un fichero que contenga la ruta a todos estos ficheros (uno por línea) y usar xargs para aplicar el comando a la lista.

Dependencias

Para instalar Findutils–4.1 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, cp, install, mv, rm
grep: egrep, grep
gcc: cc1, collect2, cpp0, gcc
make: make
patch: patch
sed: sed
sh–utils: basename, date, echo, hostname
textutils: cat, tr

Instalando Gawk–3.1.0

```
Estimación del tiempo de construcción:      2 minutos
Estimación del espacio de disco requerido:  10 MB
```

Instalación de Gawk

Atención: NO ejecutes **`make uninstall`** en este paquete si has aplicado el arreglo de *sed* para cambiar la definición del directorio `libexec`. La regla *uninstall* del fichero `Makefile` ejecuta un comando como **`rm -rf <libexecdir>/*`** Como hemos cambiado el directorio `libexec` a `/usr/bin` se ejecutará **`rm -rf /usr/bin/*`**

Instala Gawk ejecutando los siguientes comandos:

```
cp awklib/Makefile.in awklib/Makefile.in.backup &&
sed -e '/^datadir/s/awk/gawk/' \
-e '/^libexecdir/s%/awk%%' awklib/Makefile.in.backup \
   > awklib/Makefile.in &&
./configure --prefix=/usr --libexecdir=/usr/bin &&
make &&
make install
```

Contenido de gawk–3.1.0

Aún no se ha comprobado

Dependencias

Para instalar Gawk–3.1.0 se necesita lo siguiente:

Dependencias no comprobadas todavía

Instalando Ncurses–5.2

```
Estimación del tiempo de construcción:      6 minutos
Estimación del espacio de disco requerido:  29 MB
```

Instalación de Ncurses

Instala Ncurses ejecutando los siguientes comandos:

```
./configure --prefix=/usr --libdir=/lib \
   --with-shared --disable-termcap &&
make &&
make install &&
cd /lib &&
mv *.a /usr/lib &&
chmod 755 *.5.2 &&
cd /usr/lib &&
ln -sf libncurses.a libcurses.a &&
ln -sf ../../lib/libncurses.so &&
ln -sf ../../lib/libcurses.so &&
ln -sf ../../lib/libform.so &&
ln -sf ../../lib/libpanel.so &&
ln -sf ../../lib/libmenu.so
```

Explicación de los comandos

`--with-shared:` Esto activa la compilación de las librerías compartidas de ncurses.

`--disable-termcap:` Desactiva la compilación del soporte del antiguo termcap.

`cd /lib && mv *.a /usr/lib:` Esto mueve todas las librerías estáticas de ncurses de /lib a /usr/lib. El directorio /lib sólo debería contener los ficheros compartidos que son esenciales para el sistema cuando /usr no está montado.

`chmod 755 *.5.2:` Las librerías compartidas deben ser ejecutables. La rutina de instalación de ncurses no establece adecuadamente los permisos, por lo que tenemos que ponerlos manualmente.

`ln -sf libncurses.a libcurses.a:` Algunos programas intentan enlazarse usando –lcurses en lugar de –lncurses. Este enlace simbólico asegura que esos programas se enlacen sin error.

`ln -sf ../../lib/libncurses.so etc:` Estos enlaces simbólicos se crean para poner en orden la instalación. Es una buena práctica tener los ficheros *.so tanto en /lib como en /usr/lib, para asegurarse de que el enlazador será capaz de encontrar siempre los ficheros tanto si los busca en /lib o en /usr/lib

Contenido de ncurses–5.2

Programas

captoinfo (enlace a tic), clear, infocmp, infotocap (enlace a tic), reset (enlace a tset), tack, tic, toe, tput y tset.

Descripciones

captoinfo

captoinfo convierte una descripción de termcap en una descripción de terminfo.

clear

clear limpia la pantalla si es posible. Busca en el entorno el tipo de terminal y en la base de datos de terminfo para figurarse como limpiar la pantalla.

infocmp

infocmp puede usarse para comparar una entrada binaria de terminfo con otras entradas terminfo, reescribir una descripción de terminfo para aprovechar el campo "use=", o mostrar una descripción terminfo del fichero binario (term) en una variedad de formatos (lo opuesto de lo que hace tic).

infotocap

infotocap convierte una descripción terminfo en una descripción termcap.

reset

reset activa los modos "cooked" y "echo", quita los modos "cbreak" y "raw", activa la traslación de nueva línea y restablece cualquier carácter especial desactivado a sus valores por defecto, antes de hacer la inicialización del terminal de la misma manera que tset.

tack

tack es el comprobador de acciones de terminfo.

tic

tic es el compilador de entradas de descripciones de terminfo. El programa transforma un fichero terminfo en formato fuente a formato binario para su uso con las rutinas de las librerías ncurses. Los ficheros terminfo contienen información sobre las capacidades de un terminal.

toe

toe lista todos los tipos de terminal disponibles por su nombre primario, con descripciones.

tput

tput usa la base de datos de terminfo para poner a disposición del intérprete de comandos la información sobre las capacidades dependientes del terminal, para inicializar o restablecer el terminal, o para devolver el nombre largo del tipo de terminal requerido.

tset

tset inicializa los terminales para poder usarlos, pero no se usa posteriormente. Incluye compatibilidad con 4.4BSD.

Librerías

libcurses.[a,so] (enlace a libncurses.[a,so]), libform.[a,so], libform_g.a, libmenu.[a,so], libmenu_g.a, libncurses++.a, libncurses.[a,so], libncurses_g.a, libpanel.[a,so] y libpanel_g.a

libcurses, libncurses++, libncurses, libncurses_g

Las librerías que componen Ncurses se usan para mostrar texto (a menudo de forma vistosa) en la pantalla. Un ejemplo donde se usa ncurses es en el proceso "make menuconfig" del núcleo. Las librerías libncurses son la base del sistema.

libform, libform_g

libform se usa para implementar formularios en ncurses.

libmenu, libmenu_g

libmenu se usa para implementar menús en ncurses.

libpanel, libpanel_g

libpanel se usa para implementar paneles en ncurses.

Dependencias

Para instalar Ncurses−5.2 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, cp, install, ln, mkdir, mv, rm
gcc: c++, cc1, cc1plus, collect2, cpp0, gcc
glibc: ldconfig
grep: egrep, fgrep, grep
make: make
gawk: gawk
sed: sed
sh−utils: basename, date, echo, expr, hostname, uname
textutils: cat, sort, tr, wc

Instalando Vim−6.1

```
Estimación del tiempo de construcción:      2 minutos
Estimación del espacio de disco requerido:  15 MB
```

Instalación de Vim

Si no quieres instalar vim como editor en el sistema LFS, querrás descargar alguno alternativo e instalar el editor que prefieras. Hay algunas recetas disponibles sobre cómo instalar otros editores en http://hints.linuxfromscratch.org/hints/. Las recetas que hay actualmente disponibles son para Emacs, Joe y nano.

Instala Vim ejecutando los siguientes comandos:

```
./configure --prefix=/usr &&
make CPPFLAGS=-DSYS_VIMRC_FILE=\\\"/etc/vimrc\\\" &&
make install &&
cd /usr/bin &&
ln -sf vim vi
```

Si piensas instalar el sistema X Window en tu sistema LFS, piensa en recompilar Vim después de instalar X. Vim tiene una bonita versión con interfaz gráfica que necesita X y alguna otra librería instalada. Para más información lee la documentación de Vim.

Notas sobre cumplimiento del estándar FHS

El estándar FHS dice que los editores como vim deben usar /var/lib/<editor> para sus ficheros de estado temporales, como los ficheros guardados temporalmente, por ejemplo. Si deseas que vim cumpla el FHS, deberás usar este grupo de comandos en lugar del presentado anteriormente:

```
./configure --prefix=/usr --localstatedir=/var/lib/vim &&
make CPPFLAGS=-DSYS_VIMRC_FILE=\\\"/etc/vimrc\\\" &&
make install &&
cd /usr/bin &&
ln -sf vim vi
```

Explicación de los comandos

`make CPPFLAGS=-DSYS_VIMRC_FILE=\\\"/etc/vimrc\\\":` Estableciendo esto hacemos que vim busque el fichero `/etc/vimrc` que contiene los ajustes globales de vim. Normalmente el fichero buscado es `/usr/share/vim`, pero `/etc` es un sitio más lógico para este tipo de ficheros.

Contenido de Vim–6.1

Programas

ex (enlace a vim), rview (enlace a vim), rvim (enlace a vim), vi (enlace a vim), view (enlace a vim), vim, vimdiff (enlace a vim), vimtutor (enlace a vim) and xxd

Descripciones

ex

ex arranca vim en modo Ex.

rview

rview es una versión restringida de view. No pueden ejecutarse comandos del intérprete de comandos y Vim no puede ser suspendido.

rvim

rvim es una versión restringida de vim. No pueden ejecutarse comandos del intérprete de comandos y Vim no puede ser suspendido.

vi

vi arranca vim en modo compatible con vi.

view

view arranca vim en modo de solo lectura.

vim

vim arranca vim de la manera normal.

vimdiff

vimdiff edita dos o tres versiones de un fichero con Vim y muestra las diferencias.

vimtutor

vimtutor arranca el tutorial de Vim.

xxd

xxd genera un volcado hexadecimal o hace lo contrario.

Dependencias

Para instalar Vim−6.0 se necesita lo siguiente:

bash: sh
binutils: as, ld, strip
diffutils: cmp, diff
fileutils: chmod, cp, ln, mkdir, mv, rm, touch
find: find
gcc: cc1, collect2, cpp0, gcc
grep: egrep, grep
make: make
net−tools: hostname
sed: sed
sh−utils: echo, expr, uname, whoami
textutils: cat, tr, wc

Instalando GCC–2.95.3

```
Estimación del tiempo de construcción:      22 minutos
Estimación del espacio de disco requerido:  148 MB
```

Instalación de GCC

Se sabe que este programa se comporta mal si cambias sus parámetros de optimización (incluyendo las opciones –march y –mcpu). Es mejor dejar a GCC tal como está, así que te recomendamos borrar CFLAGS, CXXFLAGS y otras variables/parámetros como éstos que podrían cambiar la optimización con la que viene por defecto.

Instala GCC ejecutando los siguientes comandos. Estos comandos construirán los compiladores C y C++. Otros compiladores están disponibles dentro del paquete gcc. Si quieres compilar también todos los demás compiladores disponibles, elimina la opción ––enable–languages=c,c++ del comando ./configure. Mira la documentación de GCC para más detalles sobre qué compiladores adicionales están disponibles.

Advertencia: la construcción de otros compiladores no ha sido comprobada por la gente que trabaja activamente en LFS.

```
patch -Np1 -i ../gcc-2.95.3-2.patch &&
mkdir ../gcc-build &&
cd ../gcc-build &&
../gcc-2.95.3/configure --prefix=/usr --enable-shared \
   --enable-languages=c,c++ --enable-threads=posix &&
make bootstrap &&
make install &&
cd /lib &&
ln -sf ../usr/bin/cpp &&
cd /usr/lib &&
ln -sf ../bin/cpp &&
cd /usr/bin &&
ln -sf gcc cc &&
rmdir /usr/*-gnu/include &&
rmdir /usr/*-gnu
```

Contenido de gcc–2.95.3

Programas

c++, c++filt, cc (enlace a gcc), cc1, cc1plus, collect2, cpp, cpp0, g++, gcc, gcov, protoize y unprotoize

Descripciones

cc, cc1, cc1plus, gcc

Estos programas forman el compilador de C. Un compilador convierte el código fuente en formato de texto a un formato que un ordenador pueda entender. Después de que un fichero de código fuente es compilado en un fichero objeto, un enlazador creará un fichero ejecutable a partir de uno o más de estos ficheros objeto generados por el compilador.

c++, cc1plus, g++

Estos programas forman el compilador de C++; El equivalente de cc, gcc, etc.

c++filt

c++filt se usa para la decodificación (demangle) de los símbolos de C++.

collect2

No hay descripción disponible.

cpp, cpp0

cpp preprocesa los ficheros fuente, como en la inclusión del contenido de los ficheros de cabecera en los ficheros fuente. Es buena idea no hacer esto manualmente para ahorrar un montón de tiempo. Basta con insertar una línea del tipo #include <fichero>. El preprocesador inserta el contenido de ese fichero en el fichero fuente. Esta es una de las cosas que el preprocesador hace.

gcov

No hay descripción disponible.

protoize

Programa adicional que convierte funciones o definiciones del antiguo estilo anterior al ANSI al nuevo estilo de prototipos ANSI C. (un fichero por defecto para saber como va esto es `/usr/lib/gcc-lib/<arch>/<versión>/SYSCALLS.c.X`)

unprotoize

Programa adicional que convierte prototipos hechos con protoize a su estilo antiguo original (solo funciona si antes fueron generados con protoize).

Librerías

libgcc.a, libiberty.a, libstdc++.[a,so]

libgcc

libgcc.a es un fichero de soporte en tiempo de ejecución para gcc. Con frecuencia, en la mayoría máquinas, libgcc.a no es realmente necesaria.

libiberty

libiberty es una colección de subrutinas usadas por muchos programas GNU, incluidos getopt, obstack, strerror, strtol y strtoul.

libstdc++

libstdc++ es la librería C++. Es utilizada por programas escritos en C++ y contiene funciones que son usadas frecuentemente por esos programas. De esta forma el programador no necesita escribir ciertas funciones (como la escritura de una cadena de texto en pantalla) desde el principio cada vez que crea un programa.

Dependencias

Para instalar GCC−2.95.3 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, nm, ranlib
diffutils: cmp
fileutils: chmod, cp, ln, ls, mkdir, mv, rm, touch
find: find
gcc: cc, cc1, collect2, cpp0, gcc
grep: egrep, grep
make: make
patch: patch
sed: sed
sh−utils: basename, dirname, echo, expr, hostname, sleep, true, uname
tar: tar
texinfo: install−info, makeinfo
textutils: cat, tail, tr

Instalando Bison−1.34

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  3 MB
```

Instalación de Bison

Instala Bison ejecutando los siguientes comandos:

```
./configure --prefix=/usr &&
make &&
make install
```

Algunos programas no conocen bison e intentan encontrar el programa yacc (bison es una alternativa [mejor] para yacc). Para complacer a estos programas creamos un guión yacc que llama a bison haciendo que emule las convenciones de los nombres de ficheros de salida de yacc.

Crea un nuevo fichero `/usr/bin/yacc` ejecutando lo siguiente

```
cat > /usr/bin/yacc << "EOF"
#!/bin/sh
# Inicio de /usr/bin/yacc

exec /usr/bin/bison -y "$@"

# Fin de /usr/bin/yacc
EOF
chmod 755 /usr/bin/yacc
```

Contenido de bison−1.31

Programas

bison y yacc

Descripciones

bison

Bison es un generador de analizadores sintácticos, un sustituto de YACC ("Yet Another Compiler Compiler", otro compilador de compiladores). Entonces, ¿qué es Bison? Es un programa que genera un programa que analiza la estructura de un fichero de texto. En lugar de escribir el programa, el usuario especifica qué cosas deben ser conectadas y con esas reglas se genera un programa que analiza el fichero de texto. Hay muchos ejemplos en los que se necesita esta estructura y uno de ellos es la calculadora.

Tomando la cadena :

```
1 + 2 * 3
```

Una persona puede fácilmente saber que el resultado es 7. ¿Por qué? Porque al ver la estructura nuestro cerebro sabe como interpretar la cadena. La computadora no sabe eso y Bison es una herramienta que le ayuda a interpretarla presentando la cadena de la siguiente forma al compilador:

```
  +
 / \
 *  1
/ \
2  3
```

Comenzando por la base del árbol y subiendo por los números 2 y 3, que están unidos por el símbolo de la multiplicación, la computadora multiplica 2 y 3. Almacena el resultado de la multiplicación y, lo siguiente que ve, es el resultado de 2*3 y el número 1 unido con el símbolo de la suma. Añadiendo 1 al resultado previo se obtiene 7. El cálculo de formulas más complejas puede hacerse pasándolas a este formato de árbol. El ordenador comienza justo por la base y sigue trabajando hacia arriba hasta alcanzar el resultado correcto. Por supuesto, Bison no se usa sólo en calculadoras.

yacc

Nosotros creamos un guión llamado "yacc" que invoca a bison usando la opción −y. Esto es para la compatibilidad con programas que usan yacc en lugar de bison.

Dependencias

Para instalar Bison−1.31 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm, rmdir

gcc: cc, cc1, collect2, cpp0, gcc
grep: egrep, fgrep, grep
make: make
sed: sed
sh−utils: basename, dirname, echo, expr, hostname, sleep, uname
texinfo: install−info
textutils: cat, head, tr, uniq

Instalando Less−374

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  2 MB
```

Instalación de Less

Instala Less ejecutando los siguientes comandos:

```
./configure --prefix=/usr --bindir=/bin &&
make &&
make install
```

Contenido de less−358

Programas

less, lessecho y lesskey

Descripción

less

El programa less es un paginador de ficheros (o visor de texto). Muestra el contenido de un fichero con la posibilidad de recorrerlo. Less es una evolución del paginador habitual llamado "more". Less tiene la habilidad de poder moverse a través de los ficheros y no necesita leer por completo el fichero al principio, lo que le hace rápido cuando se leen ficheros largos.

lessecho

lessecho es necesario para expandir meta−caracteres, como * y ?, en los nombres de ficheros en sistemas Unix.

lesskey

lesskey se usa para especificar los códigos de teclas usados por less.

Dependencias

Para instalar Less−358 se necesita lo siguiente:

bash: sh

binutils: as, ld
diffutils: cmp
fileutils: chmod, install, mv, rm, touch
grep: egrep, grep
gcc: cc1, collect2, cpp0, gcc
make: make
sed: sed
sh–utils: expr, hostname, uname
textutils: cat, tr

Instalando Groff–1.17.2

```
Estimación del tiempo de construcción:      2 minutos
Estimación del espacio de disco requerido:  16 MB
```

Instalación de Groff

Instala Groff ejecutando los siguientes comandos:

```
./configure --prefix=/usr &&
make &&
make install &&
cd /usr/bin &&
ln -s soelim zsoelim &&
ln -s eqn geqn &&
ln -s tbl gtbl
```

Explicación de los comandos

`ln -s ...`: Estos enlaces simbólicos son necesarios para **`xman`** y otros programas de documentación groff/man para trabajar correctamente.

Contenido de groff–1.17.2

Programas

addftinfo, afmtodit, eqn, grn, grodvi, groff, grog, grolbp, grolj4, grops, grotty, hpftodit, indxbib, lkbib, lookbib, mmroff, neqn, nroff, pfbtops, pic, post–grohtml, pre–grohtml, refer, soelim, tbl, tfmtodit y troff

Descripciones

addftinfo

addftinfo lee un fichero de fuentes troff y añade alguna información adicional sobre la métrica de la fuente, que es usada por el sistema groff.

afmtodit

afmtodit crea un fichero de fuentes para usarlo con groff y grops.

eqn

eqn compila las descripciones de las formulas embebidas en los ficheros de entrada de troff a comandos que pueda entender troff.

grn

grn es un preprocesador groff para ficheros gremlin.

grodvi

grodvi es un controlador para groff que genera formatos dvi de TeX.

groff

groff es una interfaz para el sistema de formateado de documentos groff. Normalmente lanza el programa troff y un post−procesador apropiado para el dispositivo seleccionado.

grog

grog lee ficheros y supone cual de las opciones −e, −man, −me, −mm, −ms, −p, −s, y −t de groff se necesitan para mostrar los ficheros, y muestra el comando de groff incluyendo esas opciones en la salida estándar.

grolbp

grolbp es un controlador de groff para las impresoras Canon CAPSL (series LBP−4 y LBP−8 de impresoras láser)

grolj4

grolj4 es un controlador para groff que produce salidas en el formato PCL5 adecuado para impresoras HP Laserjet 4.

grops

grops transforma la salida de GNU troff en Postscript.

grotty

grotty transforma la salida de GNU troff en un formato adecuado para dispositivos tipo máquina de escribir.

hpftodit

hpftodit crea un fichero de fuentes para usar con groff −Tlj4 a partir de ficheros de marcas de fuentes métricas de HP.

indxbib

indxbib hace un índice inverso para la base de datos bibliográfica, un fichero específico para usarlo con refer, lookbib, y lkbib.

lkbib

lkbib busca, en las bases de datos bibliográficas, referencias que contengan las claves especificadas y muestra cualquier referencia encontrada en la salida estándar.

lookbib

lookbib muestra un aviso en la salida de error estándar (excepto si la entrada estándar no es un terminal), lee de la entrada estándar una línea conteniendo un grupo de palabras clave, busca en las bases de datos bibliográficos en un fichero especificado las referencias que contengan dichas claves, muestra cualquier referencia encontrada en la salida estándar y repite el proceso hasta el final de la entrada.

mmroff

mmroff es un preprocesador simple para groff.

neqn

El guión neqn formatea ecuaciones para salida ASCII.

nroff

El guión nroff emula al comando UNIX nroff usando groff.

pfbtops

pfbtops transforma una fuente en formato .pfb de Postscript a ASCII.

pic

pic compila descripciones de gráficos embebidos dentro de ficheros de entrada de troff o TeX a comandos que puedan ser entendidos por TeX o troff.

pre−grohtml y post−grohtml

pre− y post−grohtml transforman la salida de GNU troff a html.

refer

refer copia el contenido de un fichero en la salida estándar, excepto que las líneas entre .[y .] son interpretadas como citas, y las líneas entre .R1 y .R2 son interpretadas como comandos sobre cómo deben ser procesadas las citas.

soelim

soelim lee ficheros y reemplaza líneas de la forma *.so fichero* por el contenido de *fichero*.

tbl

tbl compila descripciones de tablas embebidas dentro de ficheros de entrada troff a comandos que puedan ser entendidos por troff.

tfmtodit

tfmtodit crea un fichero de fuentes para su uso con **`groff -Tdvi`**

troff

troff es altamente compatible con Unix troff. Normalmente debe ser invocado usando el comando groff, que también lanzará los preprocesadores y post procesadores en el orden correcto y con las opciones necesarias.

Dependencias

Para instalar Groff−1.17.2 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib
bison: bison
diffutils: cmp
fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm, touch
gcc: cc1, cc1plus, collect2, cpp0, g++, gcc
grep: egrep, grep
make: make
gawk: awk
sed: sed
sh−utils: basename, date, echo, expr, hostname, uname
textutils: cat, tr

Instalando Man−1.5j

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  1 MB
```

Instalación de Man

Instala Man ejecutando los siguientes comandos:

```
./configure -default &&
make &&
make install &&
mv /usr/share/misc/man.conf /etc
```

Puede que quieras mirar la receta sobre man en http://hints.linuxfromscratch.org/hints/man.txt que se ocupa de cuestiones de formateado y compresión de las páginas del manual.

Contenido de man−1.5j

Programas

apropos, makewhatis, man, man2dvi, man2html y whatis

Descripciones

apropos

apropos busca las palabras claves indicadas en un grupo de ficheros de bases de datos que contienen descripciones cortas de los comandos del sistema, y muestra el resultado en la salida estándar.

makewhatis

makewhatis lee todas las páginas de manual contenidas en las secciones incluidas en las rutas "manpath" o las páginas preformateadas contenidas en las secciones de las rutas "catpath". Por cada página incluye una línea en la base datos de whatis; cada línea consiste en el nombre de la página y una descripción corta, separados por un guión. La descripción se extrae del contenido de la sección NAME de la página de manual.

man

man formatea y muestra en línea las páginas de manual.

man2dvi

man2dvi convierte una página de manual al formato dvi.

man2html

man2html convierte una página de manual en html.

whatis

whatis busca las palabras claves indicadas en un grupo de ficheros de bases de datos que contienen descripciones cortas de los comandos del sistema, y muestra el resultado en la salida estándar. Sólo las coincidencias de palabras completas son mostradas.

Dependencias

Para instalar Man−1.5i2 se necesita lo siguiente:

bash: sh
binutils: as, ld
fileutils: chmod, cp, install, mkdir, rm
gcc: c11, collect2, cpp0, gcc
grep: grep
make: make
gawk: awk
sed: sed
sh−utils: echo
textutils: cat

Instalando Perl−5.6.1

```
Estimación del tiempo de construcción:      6 minutos
```

```
Estimación del espacio de disco requerido:  35 MB
```

Instalación de Perl

Instala Perl ejecutando los siguientes comandos:

```
./configure.gnu --prefix=/usr &&
make &&
make install
```

Si quieres más control sobre la forma en que perl se auto−configura para construirse, puedes ejecutar el guión interactivo **Configure** y modificar el modo en que perl se construye. Si piensas que puedes vivir con los (razonables) valores por defecto que perl auto−detecta, entonces usa los comandos listados anteriormente.

Contenido de perl−5.6.1

Programas

a2p, c2ph, dprofpp, find2perl, h2ph, h2xs, perl, perl5.6.1, perlbug, perlcc, perldoc, pl2pm, pod2html, pod2latex, pod2man, pod2text, pod2usage, podchecker, podselect, pstruct, s2p y splain

Descripciones

a2p

a2p es un traductor de awk a perl.

c2ph

c2ph vuelca estructuras C similares a las generadas por "cc −g −S".

dprofpp

dprofpp muestra datos de perfiles perl.

find2perl

find2perl traduce líneas del comando find a código Perl.

h2ph

h2ph convierte ficheros de cabecera .h de C en ficheros de cabecera .ph de Perl.

h2xs

h2xs convierte ficheros de cabecera .h de C en extensiones de Perl.

perl, perl5.6.1

perl es el Lenguaje Práctico de Extracción e Informe. Combina algunas de las mejores características de C, sed, awk y sh en un poderoso lenguaje.

perlbug

perlbug ayuda en la generación de informes de errores sobre perl o de los módulos incorporados, y los envía por correo.

perlcc

perlcc genera ejecutables a partir de programas Perl.

perldoc

perldoc examina la parte de la documentación en formato .pod que se incluye en el árbol de instalación de perl o en un guión de perl, y lo muestra mediante "pod2man | nroff −man | $PAGER".

pl2pm

pl2pm es una herramienta que ayuda en la conversión de librerías .pl de estilo Perl4 en módulos de librería de estilo Perl5.

pod2html

pod2html convierte ficheros de formato pod a formato HTML.

pod2latex

pod2latex convierte ficheros de formato pod a formato LaTeX.

pod2man

pod2man convierte datos pod en entradas formateadas *roff.

pod2text

pod2text convierte datos pod en texto formateado ASCII.

pod2usage

pod2usage muestra mensajes de uso a partir de documentos pod incluidos en ficheros.

podchecker

podchecker chequea la sintaxis de los ficheros de documentación en formato pod.

podselect

podselect muestra las secciones seleccionadas de la documentación pod en la salida estándar.

pstruct

pstruct vuelca estructuras C similares a las generadas por "cc −g −S".

s2p

s2p es un traductor de sed a perl.

splain

splain es un programa que fuerza diagnósticos de avisos exhaustivos en perl.

Dependencias

Para instalar Perl−5.6.1 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, nm
diffutils: cmp
fileutils: chmod, cp, ln, ls, mkdir, mv, rm, touch
gcc: cc, cc1, collect2, cpp0, gcc
grep: egrep, grep
make: make
gawk: awk
sed: sed
sh−utils: basename, date, echo, expr, hostname, pwd, uname, whoami
textutils: cat, comm, sort, split, tr, uniq, wc

Instalando M4−1.4

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  3 MB
```

Instalación de M4

Instala M4 ejecutando los siguientes comandos:

```
./configure --prefix=/usr &&
make &&
make install
```

Contenido de m4−1.4

Programas

m4

Descripciones

m4

M4 es un procesador de macros. Copia la entrada a la salida expandiendo las macros en el proceso. Las macros pueden ser internas o definidas por el usuario y pueden tomar cualquier número de argumentos. Aparte de hacer la expansión de macros, m4 tiene funciones internas para la inclusión de los ficheros

indicados, lanzar comandos UNIX, hacer aritmética entera, manipular texto de diversas formas, recursión, etc. M4 puede ser usado como interfaz para un compilador o como procesador de macros por sí mismo.

Dependencias

Para instalar M4−1.4 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, cp, install, mv, rm
make: make
gcc: cc1, collect2, cpp0, gcc
grep: egrep, grep
sed: sed
sh−utils: date, echo, hostname
textutils: cat, tr

Instalando Texinfo−4.1

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  10 MB
```

Instalación de Texinfo

Instala Texinfo ejecutando los siguientes comandos:

```
./configure --prefix=/usr &&
make &&
make install &&
make TEXMF=/usr/share/texmf install-tex
```

Explicación de los comandos

`make TEXMF=/usr/share/texmf install-tex`: Esto instala los componentes de texinfo que pertenecen a la instalación de TeX. Aunque TeX no se instala en LFS, se instala aquí para completar la instalación de texinfo.

Contenido de texinfo−4.0

Programas

info, install−info, makeinfo, texi2dvi y texindex

Descripciones

info

El programa info lee documentos Info, almacenados normalmente en el directorio /usr/share/info. Los documentos Info son como las páginas de manual, pero suelen contener algo más que una simple explicación

de las opciones de un programa.

install–info

El programa install–info actualiza las entradas info. Cuando se ejecuta info, muestra una lista con los temas (es decir, los documentos info) disponibles. Pues bien, el programa install–info se usa para mantener esta lista. Si los ficheros info son eliminados manualmente, también es necesario eliminar el tema en el fichero índice, para lo cual se utiliza este programa. También se utiliza para lo contrario, es decir, cuando se añaden documentos info.

makeinfo

El programa makeinfo convierte documentos fuente Texinfo a varios formatos. Los formatos disponibles son: ficheros info, texto plano y HTML.

texi2dvi

El programa texi2dvi imprime documentos Texinfo.

texindex

El programa texindex se usa para ordenar ficheros índice de Texinfo.

Dependencias

Para instalar Texinfo–4.0 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, install, ln, ls, mkdir, mv, rm
gcc: cc1, collect2, cpp0, gcc
grep: egrep, fgrep, grep
make: make
sed: sed
sh–utils: basename, echo, expr, hostname, sleep
texinfo: makeinfo
textutils: cat, tr

Instalando Autoconf–2.53

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  4 MB
```

Instalación de Autoconf

Instala Autoconf ejecutando los siguientes comandos:

```
./configure --prefix=/usr &&
make &&
make install
```

Contenido de autoconf–2.52

Programas

autoconf, autoheader, autoreconf, autoscan, autoupdate y ifnames

Descripciones

autoconf

Autoconf es una herramienta para generar guiones del intérprete de comandos que automáticamente configuran paquetes de código fuente, adaptándolos a muchas clases de sistemas tipo UNIX. Los guiones de configuración creados por Autoconf son independientes de Autoconf cuando se ejecutan, por tanto sus usuarios no necesitan tenerlo instalado.

autoheader

El programa autoheader puede crear un plantilla de declaraciones #define de C, usada posteriormente por el guión configure.

autoreconf

Si hay que generar varios guiones de configuración con autoconf, el programa autoreconf puede ahorrar algo de trabajo. Ejecuta autoconf (y autoheader, cuando es necesario) repetidamente para recrear los guiones de configuración de autoconf y las plantillas de las cabeceras de configuración en el árbol de directorios actual.

autoscan

El programa autoscan ayuda en la creación de ficheros configure.in para los paquetes. Este programa analiza los ficheros fuente en el árbol de directorios que se le pasa como argumento, o a partir del actual si no se especifica ninguno. Busca en los ficheros fuente problemas comunes de portabilidad y crea un fichero configure.scan que es una versión preliminar del fichero configure.in para ese paquete.

autoupdate

El programa autoupdate actualiza las llamadas a los macros de autoconf en los ficheros configure.in cambiando los nombres antiguos por los actuales.

ifnames

ifnames ayuda en la creación de los ficheros configure.in. Escribe los identificadores que el paquete usa en las construcciones condicionales del preprocesador de C. Si un paquete está preparado para tener cierta portabilidad, este programa le ayuda a averiguar qué debe verificar el guión configure. Puede ayudar a fijar ciertas carencias en los configure.in generados por autoscan.

Dependencias

Para instalar Autoconf–2.52 se necesita lo siguiente:

bash: sh

diffutils: cmp
fileutils: chmod, install, ln, ls, mkdir, mv, rm
grep: fgrep, grep
m4: m4
make: make
gawk: gawk
sed: sed
sh–utils: echo, expr, hostname, sleep, uname
texinfo: install–info
textutils: cat, tr

Instalando Automake–1.6

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  3 MB
```

Instalación de Automake

Instala Automake ejecutando los siguientes comandos:

```
./configure --prefix=/usr &&
make install
```

Contenido de automake–1.5

Programas

aclocal y automake

Descripciones

aclocal

Automake incluye una serie de macros de Autoconf que pueden ser usadas en los paquetes; algunas de ellas son requeridas por Automake en ciertas situaciones. Estas macros deben estar definidas en el fichero aclocal.m4; de otra forma no serán vistas por autoconf.

El programa aclocal genera automáticamente los ficheros aclocal.m4 basados en el contenido de configure.in. Esto proporciona una forma conveniente de obtener las macros facilitadas por Automake sin tener que buscarlas. Por otra parte, el mecanismo de aclocal es extensible para poder ser usado por otros paquetes

automake

Para crear todos los Makefile.in de un paquete, ejecuta el programa automake en el directorio base, sin argumentos. Automake automáticamente encontrará cada fichero Makefile.am apropiado (tras escanear configure.in) y generará el correspondiente Makefile.in.

Dependencias

Para instalar Automake–1.5 se necesita lo siguiente:

bash: sh
diffutils: cmp
fileutils: chmod, install, ls, mkdir, mv, rm, rmdir
grep: fgrep, grep
make: make
perl: perl
sed: sed
sh−utils: echo, expr, hostname, sleep
texinfo: install−info
textutils: cat, tr

Instalando Bash−2.05a

```
Estimación del tiempo de construcción:      3 minutos
Estimación del espacio de disco requerido:  19 MB
```

Instalación de Bash

Instala Bash ejecutando los siguientes comandos:

```
./configure --prefix=/usr --with-curses \
   --bindir=/bin &&
make &&
make install &&
cd /bin &&
ln -sf bash sh &&
exec /bin/bash --login
```

Contenido de bash−2.05a

Programas

bash, sh (enlace a bash) y bashbug

Descripciones

bash

bash es la "Bourne−Again SHell", que es un completo intérprete de comandos usado ampliamente en sistemas Unix. Bash es un programa que lee de la entrada estándar, el teclado. Un usuario escribe algo y el programa evalúa lo que ha escrito y hace algo con ello, como lanzar un programa.

bashbug

bashbug es un guión que ayuda al usuario en la composición y envío de informes de errores relacionados con bash, en un formato estándar.

sh

sh es un enlace simbólico al programa bash. Cuando se invoca como sh, bash intenta imitar el comportamiento de las versiones antiguas de sh lo mejor posible, mientras que también cumple los estándares

POSIX.

Dependencias

Para instalar Bash−2.05a se necesita lo siguiente:

bash: bash, sh
binutils: ar, as, ld, ranlib, size
diffutils: cmp
fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm
gcc: cc, cc1, collect2, cpp0, gcc
grep: egrep, grep
make: make
gawk: awk
sed: sed
sh−utils: basename, echo, expr, hostname, sleep, uname
texinfo: install−info
textutils: cat, tr, uniq

Instalando Flex−2.5.4a

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  3MB
```

Instalación de Flex

Instala Flex ejecutando los siguientes comandos:

```
./configure --prefix=/usr &&
make &&
make install
```

Algunos programas no conocen flex e intentan encontrar el programa lex (flex es una alternativa [mejor] para lex). Para complacer a estos programas creamos un guión lex que llama a flex haciéndole que emule a lex.

Crea un nuevo fichero `/usr/bin/lex` ejecutando lo siguiente:

```
cat > /usr/bin/lex << "EOF"
#!/bin/sh
# Inicio de /usr/bin/lex

exec /usr/bin/flex -l "$@"

# Fin de /usr/bin/lex
EOF
chmod 755 /usr/bin/lex
```

Contenido de flex−2.5.4a

Programas

flex, flex++ (enlace a flex) y lex

Descripciones

flex

flex es una herramienta para generar programas capaces de reconocer patrones de texto. El reconocimiento de patrones es muy útil en muchas aplicaciones. El usuario establece las reglas a usar y flex generará el programa que buscará esos patrones. La razón por la que la gente usa flex es porque es mucho más fácil establecer las reglas a usar que escribir directamente un programa que busque el texto.

flex++

flex++ invoca una versión de flex usada exclusivamente para generar analizadores en C++.

lex

Creamos un guión "lex" que invoca a flex con la opción –l. Esto es por compatibilidad con programas que usan lex en lugar de flex

Librerías

libfl.a

Descripciones

libfl

No hay descripción disponible.

Dependencias

Para instalar Flex−2.5.4a se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib
bison: bison
diffutils: cmp
fileutils: chmod, cp, install, ln, mv, rm, touch
gcc: cc1, collect2, cpp0, gcc
grep: egrep, grep
make: make
sed: sed
sh−utils: echo, hostname
textutils: cat, tr

Instalando File–3.37

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  2 MB
```

Instalación de File

Instala File ejecutando los siguientes comandos:

```
touch aclocal.m4 configure Makefile.in stamp-h.in &&
./configure --prefix=/usr --datadir=/usr/share/misc &&
make &&
make install
```

Explicación de los comandos

`touch aclocal.m4 configure Makefile.in stamp-h.in`: Este comando evita un error que aparece cuando se compila file con automake–1.5 instalado, cambiando las fechas de modificación de algunos ficheros a la fecha actual. Con el cambio de fecha haremos que make piense que esos ficheros son actuales y no los regenerará.

Contenido de file–3.37

Programas

file

Descripciones

file

File comprueba cada fichero especificado para clasificarlo. Se hacen tres tipos de pruebas, en este orden: pruebas de sistemas de ficheros, pruebas de números mágicos y pruebas de lenguajes. La primera prueba que tenga éxito hace que se muestre el tipo de fichero.

Dependencias

Para instalar File–3.37 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: as, ld
diffutils: cmp
fileutils: chmod, install, ln, ls, mv, rm, touch
gcc: cc1, collect2, cpp0, gcc
grep: egrep, grep
m4: m4
make: make
gawk: gawk

sed: sed
sh–utils: echo, expr, hostname, sleep
texinfo: makeinfo
textutils: cat, tr

Instalando Libtool–1.4.2

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  5 MB
```

Instalación de Libtool

Instala Libtool ejecutando los siguientes comandos:

```
./configure --prefix=/usr &&
make &&
make install
```

Contenido de libtool–1.4.2

Programas

libtool y libtoolize

Descripciones

libtool

Libtool proporciona servicios de soporte generalizados para la compilación de librerías.

libtoolize

libtoolize proporciona una forma estándar de añadir soporte para libtool a un paquete.

Librerías

libltdl.[a,so]

Descripciones

libltdl

Libtool proporciona una pequeña librería, llamada `libltdl', cuyo objetivo es ocultar las dificultades en la carga dinámica de librerías a los programadores.

Dependencias

Para instalar Libtool–1.4.2 se necesita lo siguiente:

bash: sh

binutils: ar, as, ld, nm, ranlib, strip
diffutils: cmp
fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm, rmdir
gcc: cc, cc1, collect2, cpp0
glibc: ldconfig
grep: egrep, fgrep, grep
make: make
sed: sed
sh−utils: echo, expr, hostname, sleep, uname
texinfo: install−info
textutils: cat, sort, tr, uniq

Instalando Bin86−0.16.2

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  1 MB
```

Instalación de Bin86

Este paquete sólo es necesario si decides utilizar Lilo en tu sistema LFS. Si vas a usar alguna otra utilidad, como Grub, no necesitas bin86. Revisa la documentación de tu gestor de arranque favorito para ver si necesita el paquete bin86 (normalmente, sólo son necesarios ld86 y/o as86).

Recuerda que no sólo los gestores de arranque requieren el paquete bin86. Siempre existe la posibilidad de que algún otro paquete necesite los programas de éste, así que tenlo en cuenta si decides saltarte este paso.

Instala Bin86 ejecutando los siguientes comandos:

```
make &&
make PREFIX=/usr install
```

Contenido de bin86−0.16.0

Programas

as86, as86_encap, ld86, nm86 (enlace a objdump86), objdump86 y size86 (enlace a objdump86)

Descripciones

as86

as86 es un ensamblador para los procesadores 8086...80386.

as86_encap

as86_encap es un guión del intérprete de comandos que llama a as86 y que convierte el binario creado en un fichero prog.v de C para ser incluido o enlazado con programas como instaladores de bloques de arranque.

ld86

ld86 entiende solamente los ficheros objeto creados por el ensamblador as86. Puede enlazarlos tanto en un ejecutable I&D impuro como en uno separado.

nm86

No hay descripción disponible.

objdump86

No hay descripción disponible.

size86

No hay descripción disponible.

Dependencias

Para instalar Bin86–0.16.0 se necesita lo siguiente:

bash: sh
binutils: as, ld, strip
fileutils: chmod, install, ln, mv
gcc: cc, cc1, collect2, cpp0
make: make
sed: sed

Instalando Binutils–2.12

```
Estimación del tiempo de construcción:      6 minutos
Estimación del espacio de disco requerido:  85 MB
```

Instalación de Binutils

Se sabe que este programa se comporta mal si cambias sus parámetros de optimización (incluyendo las opciones –march y –mcpu). Es mejor dejar a Binutils tal como está, así que te recomendamos borrar CFLAGS, CXXFLAGS y otras variables/parámetros como éstos que podrían cambiar la optimización con la que viene por defecto.

Instala Binutils ejecutando los siguientes comandos:

```
mkdir ../binutils-build &&
cd ../binutils-build &&
../binutils-2.12/configure --prefix=/usr --enable-shared &&
make tooldir=/usr &&
make tooldir=/usr install &&
make tooldir=/usr install-info
```

Explicación de los comandos

`make tooldir=/usr install-info:` Esto instalará las páginas info de binutils.

Contenido de binutils–2.11.2

Programas

addr2line, ar, as, c++filt, gasp, gprof, ld, nm, objcopy, objdump, ranlib, readelf, size, strings y strip

Descripciones

addr2line

addr2line traslada direcciones de programas a nombres de ficheros y números de líneas. Dándole una dirección y un ejecutable, usa la información de depuración del ejecutable para averiguar qué fichero y número de línea está asociado con dicha dirección.

ar

El programa ar crea, modifica y extrae desde archivos. Un archivo es un fichero que almacena una colección de otros ficheros en una estructura que hace posible obtener el original de cada fichero individual (llamados miembros del archivo).

as

as está pensado, principalmente, para ensamblar la salida del compilador GNU gcc para ser usada por el enlazador ld.

c++filt

El lenguaje C++ proporciona sobrecarga de funciones, lo que significa que es posible escribir varias funciones con el mismo nombre (suponiendo que cada una tome parámetros de diferente tipo). Todos los nombres de funciones C++ son codificadas dentro de una etiqueta de bajo nivel del ensamblador (este proceso es conocido como "mangling"). El programa c++flit hace lo contrario: decodifica (demangling) nombres de bajo nivel en nombres de nivel de usuario para que el enlazador pueda evitar conflictos en estas funciones sobrecargadas.

gasp

Gasp es el preprocesador de las macros del ensamblador.

gprof

gprof muestra el grafo de llamadas de los datos perfilados.

ld

ld combina un número de objetos y ficheros de archivo, reubica sus datos y establece las referencias a los símbolos. Frecuentemente, el último paso de la compilación de un nuevo programa es hacer una llamada a ld.

nm

nm lista los símbolos de los ficheros objeto.

objcopy

La utilidad objcopy copia el contenido de un fichero objeto en otro. objcopy usa la librería BFD de GNU para leer y escribir los ficheros objeto. Puede escribir el fichero objeto destino en un formato diferente al del fichero objeto fuente .

objdump

objdump muestra información sobre uno o más ficheros objeto. Mediante opciones se puede indicar la información a mostrar. Esta información es útil fundamentalmente para los programadores que trabajan en herramientas de compilación (al contrario de los programadores que sólo quieren que sus programas compilen y funcionen).

ranlib

ranlib genera un índice de los contenidos de un archivo, y lo coloca en el archivo. El índice lista cada símbolo definido por un miembro de un archivo que es un fichero objeto reubicable.

readelf

readelf muestra información sobre binarios de tipo elf.

size

size lista los tamaños de las secciones –y el tamaño total– para cada uno de los ficheros objeto en su lista de argumentos. Por defecto, se genera una línea de salida por cada fichero objeto o cada módulo de un archivo.

strings

Para cada fichero dado, strings muestra las cadenas de caracteres imprimibles de al menos 4 caracteres (o el número especificado en las opciones del programa) seguidas por un carácter no imprimible. Por defecto, sólo muestra las cadenas procedentes de las secciones de inicialización y carga de los ficheros objeto; para otros tipos de ficheros muestra todas las cadenas de los mismos.

strings es útil, principalmente, para determinar el contenido de ficheros que no son de texto.

strip

strip elimina todos los símbolos o sólo los especificados de los ficheros objeto. La lista de ficheros objeto puede incluir archivos. Se debe indicar, por lo menos, un fichero objeto. strip modifica los ficheros mencionados en sus argumentos, en vez de escribir copias modificadas con otro nombre.

Librerías

libbfd.a, libiberty.a y libopcodes.a

Descripciones

libbfd

libbfd es la librería de descriptores de ficheros binarios (BFD).

libiberty

libiberty es una colección de subrutinas usadas por varios programas GNU, incluidos getopt, obstack, strerror, strtol y strtoul.

libopcodes

libopcodes es una librería nativa para manejar mnemónicos y se usa durante la construcción de utilidades como objdump. Los mnemónicos son, en realidad, las versiones en texto legible de las instrucciones del procesador.

Dependencias

Para instalar Binutils–2.11.2 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: ar, as, ld, nm, ranlib, strip
diffutils: cmp
fileutils: chmod, cp, ln, ls, mkdir, mv, rm, rmdir, touch
flex: flex
gcc: cc, cc1, collect2, cpp0, gcc
glibc: ldconfig
grep: egrep, fgrep, grep
m4: m4
make: make
gawk: gawk
sed: sed
sh–utils: basename, echo, expr, hostname, sleep, true, uname
texinfo: install–info, makeinfo
textutils: cat, sort, tr, uniq

Instalando Bzip2–1.0.2

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  2 MB
```

Instalación de Bzip2

Instala Bzip2 ejecutando los siguientes comandos:

```
make -f Makefile-libbz2_so &&
make &&
rm /usr/bin/bz* &&
```

```
make PREFIX=/usr install &&
cp bzip2-shared /bin/bzip2 &&
ln -s libbz2.so.1.0 libbz2.so &&
cp -a libbz2.so* /lib &&
cd /usr/lib &&
ln -sf ../../lib/libbz2.so &&
cd /usr/bin &&
rm bunzip2 bzcat bzip2 &&
rm /bin/bzless /bin/bzmore &&
mv bzip2recover bzless bzmore /bin &&
cd /bin &&
ln -sf bzip2 bunzip2 &&
ln -sf bzip2 bzcat
```

Aunque, estrictamente, no es parte de un sistema LFS básico, merece la pena mencionar que se puede descargar un parche para Tar que permite comprimir y descomprimir fácilmente utilizando bzip2/bunzip2. Con el tar simple, tienes que utilizar construcciones como bzcat fichero.tar.bz2 | tar xv o tar --use-compress-prog=bunzip2 -xvf fichero.tar.bz2 para utilizar bzip2 y bunzip2 con tar. Este parche añade la opción -j para que puedas desempaquetar un archivo Bzip2 con tar xvfj fichero.tar.bz2. Después, cuando se instale el paquete Tar, se mencionará cómo aplicar este parche.

Explicación de los comandos

`make -f Makefile-libbz2_so:` Esto provocará que bzip2 sea construido usando un fichero Makefile diferente (en este caso, el fichero Makefile-libbz2_so), el cual crea una librería dinámica libbz2.so y enlaza las utilidades bzip2 con ella.

Contenido de bzip2-1.0.1

Programas

bunzip2 (enlace a bzip2), bzcat (enlace a bzip2), bzip2 y bzip2recover

Descripciones

bunzip2

Bunzip2 descomprime ficheros que han sido comprimidos con bzip2.

bzcat

bzcat (o bzip2 -dc) descomprime todos los ficheros especificados hacia la salida estándar.

bzip2

bzip2 comprime ficheros usando el algoritmo de compresión de texto por ordenación de bloques Burrows-Wheeler y la codificación Huffman. La compresión es, en general, considerablemente superior a la obtenida por otros compresores más convencionales basados en el LZ77/LZ78 y se acerca al rendimiento de la familia de compresores estadísticos PPM.

bzip2recover

bzip2recover recupera datos de ficheros bzip2 dañados.

Librerías

libbz2.[a,so]

libbz2

libbz2 es la librería que implementa la compresión sin pérdidas por ordenación de bloques usando el algoritmo de Burrows–Wheeler.

Dependencias

Para instalar Bzip2–1.0.1 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib
fileutils: cp, ln, rm
gcc: cc1, collect2, cpp0, gcc
make: make

Instalando Ed–0.2

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  2 MB
```

Instalación de Ed

Ed no es algo que querrás utilizar personalmente. Se instala aquí porque puede que lo use el programa patch si te encuentras con algún parche basado en ed. Esto no suele ocurrir porque ahora se prefieren los parches basados en diff.

Instala Ed ejecutando los siguientes comandos:

```
cp buf.c buf.c.backup &&
sed 's/int u/int u, sfd/' buf.c.backup | \
   sed '/.*\*mktemp.*/d' | \
   sed 's/.*if (mktemp.*/  sfd = mkstemp(sfn);\
   if ((sfd == -1) || (sfp = fopen (sfn, "w+")) == NULL)/' > buf.c &&
./configure --prefix=/usr &&
make &&
make install &&
mv /usr/bin/ed /usr/bin/red /bin
```

Explicación de los comandos

Los comandos de sed arreglan una vulnerabilidad de enlaces simbólicos en ed. El ejecutable ed crea ficheros en /tmp con nombres predecibles. Usando varios tipos de ataques de enlaces simbólicos, es posible hacer a ed escribir en ficheros que no debería, cambiar los permisos de varios ficheros, etc.

Contenido de ed–0.2

Programas

ed y red (enlace a ed)

Descripciones

ed

Ed es un editor de líneas de texto. Se usa para crear, mostrar, modificar o cualquier otra manipulación de ficheros de texto.

red

red es un ed restringido: sólo puede editar ficheros del directorio actual y no puede ejecutar comandos del intérprete de comandos.

Dependencias

Para instalar Ed–0.2 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, cp, install, ln, mv, rm, touch
gcc: cc1, collect2, cpp0, gcc
grep: egrep, grep
make: make
sed: sed
sh–utils: hostname
textutils: cat, tr

Instalando Gettext–0.11.1

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  11MB
```

Instalación de Gettext

Instala Gettext ejecutando los siguientes comandos:

```
./configure --prefix=/usr &&
make &&
make install
```

Contenido de gettext–0.10.40

Programas

gettext, gettextize, msgcmp, msgcomm, msgfmt, msgmerge, msgunfmt, ngettext y xgettext

Descripciones

gettext

El paquete gettext se usa en la internacionalización (también conocida como i18n) y en la localización (conocida como l10n). Los programas pueden ser compilados con soporte para lenguaje nativo (NLS) que activa el que se muestren los mensajes de salida en el idioma del usuario en vez de en el idioma por defecto, el inglés.

gettextize

El programa gettextize copia todos los ficheros estándar gettext en un directorio. Se usa para hacer paquetes con traducción gettext.

msgcmp

El programa msgcmp compara dos ficheros de traducción directa.

msgcomm

El programa msgcomm busca los mensajes que aparecen en varios ficheros .po. Se usa para comparar cómo deben traducirse las cosas.

msgfmt

El programa msgfmt compila traducciones directas en código máquina. Se usa para crear el fichero de traducción final de un programa/paquete.

msgmerge

El programa msgmerge combina dos traducciones directas en un fichero. Se usa para actualizar la traducción directa con el extracto de las fuentes.

msgunfmt

El programa msgunfmt descompila ficheros de traducciones en traducciones directas de texto. Sólo puede ser usado si la versión compilada está disponible.

ngettext

El programa ngettext muestra traducciones en lenguaje nativo de un mensaje de texto cuya forma gramatical depende de un número.

xgettext

El programa xgettext extrae las líneas de mensajes de los ficheros C del programador. Se usa para hacer la primera plantilla de traducción.

Dependencias

Para instalar Gettext–0.10.40 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: ar, as, ld, nm, ranlib, strip
bison: bison
diffutils: cmp
fileutils: chmod, install, ln, ls, mkdir, mv, rm, rmdir
gcc: cc, cc1, collect2, cpp0, gcc
grep: egrep, fgrep, grep
m4: m4
make: make
gawk: gawk
sed: sed
sh–utils: basename, echo, expr, hostname, sleep, uname
texinfo: install–info, makeinfo
textutils: cat, sort, tr, uniq

Instalando Kbd–1.06

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  8 MB
```

Instalación de Kbd

Instala Kbd ejecutando los siguientes comandos:

```
patch -Np1 -i ../kbd-1.06-2.patch &&
./configure &&
make &&
make install
```

Explicación de los comandos

`patch -Np1 -i ../kbd-1.06-2.patch`: Este parche soluciona dos problemas. El primero es el comportamiento de **`loadkeys -d`**, que es incorrecto en las versiones actuales de kbd. Es necesario arreglar esto, porque los guiones de arranque confían en un comportamiento correcto de **`loadkeys -d`**. La segunda parte del parche cambia un fichero Makefile para que algunas utilidades (setlogcons, setvesablank y getunimap) que no se instalan por defecto, se instalen también.

Contenido de kbd–1.06

Programas

chvt, deallocvt, dumpkeys, fgconsole, getkeycodes, getunimap, kbd_mode, kbdrate, loadkeys, loadunimap, mapscrn, openvt, psfaddtable (enlace a psfxtable), psfgettable (enlace a psfxtable), psfstriptable (enlace a psfxtable), psfxtable, resizecons, setfont, setkeycodes, setleds, setlogcons, setmetamode, setvesablank,

showfont, showkey, unicode_start, y unicode_stop

Descripciones

chvt

chvt cambia la terminal virtual que aparece en primer plano.

deallocvt

deallocvt desasigna las terminales virtuales no usadas.

dumpkeys

dumpkeys vuelca las tablas de traducción del teclado.

fgconsole

fgconsole muestra el número del terminal virtual activo.

getkeycodes

getkeycodes muestra la tabla de correspondencias de código de exploración (scan code) a código de teclas del núcleo.

getunimap

getunimap muestra el mapa unicode actualmente usado.

kbd_mode

kbd_mode muestra o establece el modo del teclado.

kbdrate

kbdrate establece la repetición y retardo del teclado.

loadkeys

loadkeys carga las tablas de traducción del teclado.

loadunimap

loadunimap carga la tabla de correspondencia de unicode a fuente del núcleo.

mapscrn

mapscrn carga una tabla de correspondencia de caracteres de salida definida por el usuario en el controlador de la consola. Este comando está obsoleto y sus funciones se encuentran incluidas en setfont.

openvt

openvt comienza un programa en un nuevo terminal virtual (VT)

psfaddtable, psfgettable, psfstriptable, psfxtable

Este es un grupo de herramientas para obtener tablas de caracteres Unicode a partir de fuentes de consola.

resizecons

resizecons cambia la idea del núcleo sobre el tamaño de la consola.

setfont

Esto permite cambiar las fuentes EGA/VGA de la consola.

setkeycodes

setkeycodes carga las entradas de la tabla de correspondencia de código de exploración (scan code) a código de tecla del núcleo.

setleds

setleds establece los LEDs del teclado. Mucha gente encuentra útil tener el bloqueo numérico (numlock) activado por defecto, y usando este programa puede conseguirse.

setlogcons

setlogcons envía los mensajes del núcleo a la consola.

setmetamode

setmetamode define como se manejan las teclas meta del teclado.

setvesablank

Esto permite afinar el salvapantallas incorporado en el hardware (no animados, sólo una pantalla en blanco).

showfont

showfont muestra los datos de una fuente. La información mostrada incluye información de la fuente, sus propiedades, la métrica de los caracteres y el mapa de bits de cada carácter.

showkey

showkey examina los códigos de exploración (scan codes) y los códigos de tecla enviados por el teclado.

unicode_start

unicode_start pone la consola en modo Unicode.

unicode_stop

unicode_stop revierte el teclado y la consola del modo Unicode.

Dependencias

Para instalar Kbd−1.06 se necesita lo siguiente:

bash: sh
binutils: as, ld, strip
bison: bison
diffutils: cmp
fileutils: cp, install, ln, mv, rm
flex: flex
gettext: msgfmt, xgettext
gcc: cc1, collect2, cpp0, gcc
grep: grep
gzip: gunzip, gzip
make: make
patch: patch
sed: sed
sh−utils: uname

Instalando Diffutils−2.8

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  2 MB
```

Instalación de Diffutils

Instala Diffutils ejecutando los siguientes comandos:

```
./configure --prefix=/usr &&
make &&
make install
```

Contenido de diffutils−2.7

Programas

cmp, diff, diff3 y sdiff

Descripciones

cmp y diff

Tanto cmp como diff comparan dos ficheros y muestran sus diferencias. Ambos programas tienen argumentos para comparar ficheros en diferentes situaciones.

diff3

La diferencia entre diff y diff3 es que diff compara 2 ficheros mientras diff3 compara 3.

sdiff

sdiff mezcla dos ficheros y muestra los resultados interactivamente.

Dependencias

Para instalar Diffutils−2.7 se necesita lo siguiente:

bash: sh
binutils: ld, as
diffutils: cmp
fileutils: chmod, cp, install, mv, rm
gcc: cc1, collect2, cpp0, gcc
grep: egrep, grep
make: make
sed: sed
sh−utils: date, hostname
textutils: cat, tr

Instalando E2fsprogs−1.27

```
Estimación del tiempo de construcción:      2 minutos
Estimación del espacio de disco requerido:  21 MB
```

Instalación de E2fsprogs

Instala E2fsprogs ejecutando los siguientes comandos:

```
mkdir ../e2fsprogs-build &&
cd ../e2fsprogs-build &&
../e2fsprogs-1.27/configure --prefix=/usr --with-root-prefix="" \
   --enable-elf-shlibs &&
make &&
make install &&
make install-libs &&
install-info /usr/share/info/libext2fs.info /usr/share/info/dir
```

Explicación de los comandos

`--with-root-prefix="":` La razón por la que proporcionamos esta opción es por la configuración del fichero Makefile de e2fsprogs. Algunos programas son esenciales para el uso del sistema cuando, por ejemplo, /usr todavía no ha sido montada (como el programa e2fsck). Por lo tanto, estos programas y librerías corresponden a los directorios /lib y /sbin. Si no se pasase esta opción al comando ./configure de e2fsprogs, colocaría estos programas en /usr, que no es lo que queremos.

`--enable-elf-shlibs:` Esto crea librerías compartidas, que puede que usen algunos programas de este paquete.

`make install-libs`: Esto instala las librerías compartidas que han sido construidas.

Contenido de e2fsprogs–1.25

Programas

badblocks, chattr, compile_et, debugfs, dumpe2fs, e2fsck, e2image, e2label, fsck, fsck.ext2, fsck.ext3, lsattr, mk_cmds, mke2fs, mkfs.ext2, mklost+found, resize2fs, tune2fs y uuidgen

Descripciones

badblocks

badblocks se usa para buscar bloques dañados en un dispositivo (normalmente una partición de disco).

chattr

chattr cambia los atributos de un fichero en un sistema de ficheros ext2 de Linux.

compile_et

compile_et es usado para convertir una tabla con códigos de error y sus mensajes asociados en un fichero fuente C apropiado para usar con la librería com_err.

debugfs

El programa debugfs es un depurador de sistemas de ficheros. Puede usarse para examinar y cambiar el estado de un sistema de ficheros ext2.

dumpe2fs

dumpe2fs muestra la información del superbloque y de los grupos de bloques del sistema de ficheros presente en un dispositivo específico.

e2fsck and fsck.ext2

e2fsck se usa para chequear y reparar sistemas de ficheros ext2. fsck.ext2 hace lo mismo que e2fsck.

e2image

e2image se usa para salvar información crítica de un sistema de ficheros ext2 en un fichero.

e2label

e2label muestra o cambia la etiqueta de un sistema de ficheros ext2 situado en el dispositivo especificado.

fsck

fsck se usa para chequear y, opcionalmente, reparar un sistema de ficheros.

fsck.ext3

fsck.ext3 se usa para chequear y, opcionalmente, reparar un sistema de ficheros ext3.

lsattr

lsattr muestra los atributos de un fichero en un sistema de ficheros ext2.

mk_cmds

No hay descripción disponible.

mke2fs and mkfs.ext2

mke2fs se usa para crear sistemas de ficheros ext2 en un dispositivo (normalmente una partición de disco). mkfs.ext2 hace lo mismo que mke2fs.

mklost+found

mklost+found se usa para crear un directorio lost+found en el directorio de trabajo actual de un sistema de ficheros ext2. mklost+found reserva una serie de bloques de disco en el directorio para que sean usados por e2fsck.

resize2fs

resize2fs se usa para redimensionar sistemas de ficheros ext2.

tune2fs

tune2fs ajusta los parámetros de un sistema de ficheros ext2.

uuidgen

El programa uuidgen crea un nuevo identificador universal único (UUID) usando la librería libuuid. El nuevo UUID puede considerarse razonablemente único por muchos UUID que se hayan creado en el sistema local o en otros sistemas en el pasado o en el futuro.

Librerías

libcom_err.[a,so], libe2p.[a,so], libext2fs.[a,so], libss.[a,so] y libuuid.[a,so]

Descripciones

libcom_err

No hay descripción disponible.

libe2p

No hay descripción disponible.

libext2fs

No hay descripción disponible.

libss

No hay descripción disponible.

libuuid

No hay descripción disponible.

Dependencias

Para instalar E2fsprogs−1.25 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib, strip
diffutils: cmp
fileutils: chmod, cp, install, ln, mkdir, mv, rm, sync
gcc: cc, cc1, collect2, cpp0
glibc: ldconfig
grep: egrep, grep
gzip: gzip
make: make
gawk: awk
sed: sed
sh−utils: basename, echo, expr, hostname, uname
texinfo: makeinfo
textutils: cat, tr

Instalando Fileutils−4.1

```
Estimación del tiempo de construcción:      3 minutos
Estimación del espacio de disco requerido:  16 MB
```

Instalación de Fileutils

Instala Fileutils ejecutando los siguientes comandos:

```
./configure --prefix=/usr --bindir=/bin &&
make &&
make install &&
cd /usr/bin &&
ln -sf ../../bin/install
```

Contenido de fileutils−4.1

Programas

chgrp, chmod, chown, cp, dd, df, dir, dircolors, du, install, ln, ls, mkdir, mkfifo, mknod, mv, rm, rmdir, shred, sync, touch y vdir

Descripciones

chgrp

chgrp cambia el grupo de cada fichero al grupo especificado, que puede ser tanto el nombre de un grupo como su identificador numérico.

chmod

chmod cambia los permisos de un fichero de acuerdo con el modo, que puede ser tanto una representación simbólica de los cambios a hacer, o un número octal que representa el patrón de bits de los nuevos permisos.

chown

chown cambia el usuario y/o el grupo al que pertenece un fichero.

cp

cp copia ficheros de un lugar a otro.

dd

dd copia un fichero (por defecto, de la entrada estándar a la salida estándar) con un tamaño de bloque definido por el usuario, mientras, opcionalmente, realiza conversiones en él.

df

df muestra la cantidad de espacio disponible en los sistemas de ficheros a los que pertenece cada fichero que se le pasa como argumento. Si no se indica ningún fichero, se muestra el espacio disponible en todos los sistemas de ficheros montados actualmente

dir, ls y vdir

dir y vdir son versiones de ls con formatos de salida diferentes. Estos programas listan cada fichero o directorio especificado. El contenido de los directorios se lista alfabéticamente. Para ls, los ficheros se listan en columnas ordenados verticalmente si la salida estándar es un terminal; en otro caso se listan uno por línea. Para dir, los ficheros se listan por defecto en columnas ordenados verticalmente. Para vdir, los ficheros se listan, por defecto, en formato largo.

dircolors

dircolors imprime comandos para modificar la variable de entorno LS_COLOR. La variable LS_COLOR se usa para cambiar el esquema de color por defecto de ls y de herramientas relacionadas.

du

du muestra la cantidad de espacio en disco usado por cada fichero o directorio pasado como argumento.

install

install copia ficheros y establece sus permisos y, si es posible, su propietario y grupo.

ln

ln crea enlaces duros o blandos (simbólicos) entre ficheros.

mkdir

mkdir crea directorios con el nombre indicado.

mkfifo

mkfifo crea una tubería (FIFO) con un nombre dado.

mknod

mknod crea una tubería (FIFO), un fichero especial de caracteres o un fichero especial de bloques con el nombre indicado.

mv

mv mueve ficheros de un directorio a otro o renombra ficheros, dependiendo de los argumentos que se le pasen.

rm

rm elimina ficheros o directorios.

rmdir

rmdir elimina directorios, si están vacíos.

shred

shred borra un fichero de forma segura, sobreescribiéndolo para que su contenido no pueda ser recuperado.

sync

sync guarda los bloques modificados en disco y actualiza el superbloque.

touch

touch cambia las fechas de modificación o acceso del fichero especificado, poniéndole la fecha actual. Si el fichero no existe crea uno vacío.

Dependencias

Para instalar Fileutils−4.1 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm, rmdir
gettext: msgfmt, xgettext
gcc: cc, cc1, collect2, cpp0, gcc
grep: egrep, fgrep, grep
make: make
perl: perl
sed: sed
sh−utils: basename, echo, expr, hostname, sleep, uname
texinfo: install−info
textutils: cat, tr

Instalando Grep−2.5

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  3 MB
```

Instalación de Grep

Instala Grep ejecutando los siguientes comandos:

```
./configure --prefix=/usr --bindir=/bin &&
make &&
rm /bin/egrep /bin/fgrep &&
make install
```

Contenido de grep−2.4.2

Programas

egrep, fgrep y grep

Descripciones

egrep

egrep muestra las líneas de un fichero que coincidan con una determinada expresión regular extendida.

fgrep

fgrep muestra las líneas de un fichero que coincidan con una lista de cadenas fijas, separadas por saltos de línea, cualquiera de las cuales puede ser coincidente.

grep

grep muestra las líneas de un fichero que coincidan con una expresión regular.

Dependencias

Para instalar Grep−2.4.2 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: as, ld
diffutils: cmp
fileutils: chmod, install, ls, mkdir, mv, rm
gettext: msgfmt, xgettext
gcc: cc, cc1, collect2, cpp0, gcc
glibc: getconf
grep: egrep, fgrep, grep
m4: m4
make: make
gawk: gawk
sed: sed
sh−utils: basename, echo, expr, hostname, sleep, uname
texinfo: install−info, makeinfo
textutils: cat, tr

Instalando Gzip−1.2.4a

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  1 MB
```

Instalación de Gzip

Instala Gzip ejecutando los siguientes comandos:

```
./configure --prefix=/usr &&
cp gzexe.in gzexe.in.backup &&
sed 's%"BINDIR"%/bin%' gzexe.in.backup > gzexe.in &&
make &&
make install &&
cd /usr/bin &&
mv gzip /bin &&
rm gunzip zcat &&
cd /bin &&
ln -sf gzip gunzip &&
ln -sf gzip zcat &&
ln -sf gunzip uncompress
```

Contenido de gzip−1.2.4a

Programas

gunzip (enlace a gzip), gzexe, gzip, uncompress (enlace a gunzip), zcat (enlace a gzip), zcmp, zdiff, zforce, zgrep, zmore y znew

Descripción

gunzip, uncompress

gunzip y uncompress descomprimen ficheros que hayan sido comprimidos con gzip.

gzexe

gzexe permite comprimir ficheros ejecutables que automáticamente se descomprimen y ejecutan al ser lanzados (con una penalización en el rendimiento).

gzip

gzip reduce el tamaño de los ficheros usando codificación Lempel–Ziv (LZ77)

zcat

zcat descomprime una lista de ficheros en su línea de comandos o su entrada estándar, escribiendo los datos descomprimidos en la salida estándar.

zcmp

zcmp invoca al programa cmp en ficheros comprimidos.

zdiff

zdiff invoca al programa diff en ficheros comprimidos.

zforce

zforce fuerza la extensión .gz en todos los ficheros gzip para que gzip no los comprima dos veces. Esto puede ser útil para ficheros con el nombre truncado después de una transferencia de ficheros.

zgrep

zgrep invoca al programa grep en ficheros comprimidos.

zmore

zmore es un filtro que permite examinar ficheros comprimidos de texto plano pantalla a pantalla en un terminal emulado (similar al programa more).

znew

znew recomprime ficheros en formato .Z (compress) al formato .gz (gzip).

Dependencias

Para instalar Gzip−1.2.4a se necesita lo siguiente:

bash: sh
binutils: as, ld, nm
fileutils: chmod, cp, install, ln, mv, rm
gcc: cc1, collect2, cpp, cpp0, gcc
grep: egrep, grep
make: make
sed: sed
sh−utils: hostname
textutils: cat, tr

Instalando Lilo−22.2

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  3 MB
```

Instalación de Lilo

Hemos elegido Lilo porque nos sentimos cómodos con él, pero puede que desees echar un vistazo a otra cosa. Alguien ha escrito una receta sobre GRUB en http://hints.linuxfromscratch.org/hints/grub−howto.txt, ya que es un gestor de arranque alternativo.

Instala Lilo ejecutando los siguientes comandos:

```
make &&
make install
```

Parece que la compilación de este paquete falla en ciertas máquinas cuando se usa la opción del compilador −g. Si no consigues compilar Lilo, deberías tratar de borrar el −g de la variable CFLAGS en el fichero `Makefile`.

Al final de la instalación, el proceso make install imprimirá un mensaje afirmando que se debe ejecutar /sbin/lilo para completar la actualización. No lo hagas, de momento no es útil. El fichero /etc/lilo.conf no está presente todavía. Completaremos la instalación de lilo en el capítulo 8.

Quizá te interese saber que alguien escribió una receta sobre cómo conseguir un logotipo en lugar de la línea de órdenes estándar de LILO o el menú. Échalo un vistazo en http://hints.linuxfromscratch.org/hints/bootlogo.txt .

Contenido de lilo−22.1

Programas

lilo y mkrescue

Descripciones

lilo

lilo instala el gestor de arranque que se usa para iniciar un sistema Linux.

mkrescue

mkrescue crea un disquete de rescate arrancable usando el núcleo existente y cualquier disco RAM de inicio.

Dependencias

Para instalar Lilo−22.1 se necesita lo siguiente:

bash: sh
bin86: as86, ld86
binutils: as, ld, strip
fileutils: cp, dd, ln
gcc: cc, cc1, collect2, cpp0
make: make
sed: sed
textutils: cat

Instalando Make−3.79.1

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  6 MB
```

Instalación de Make

Instala Make ejecutando los siguientes comandos:

```
./configure --prefix=/usr &&
make &&
make install &&
chgrp root /usr/bin/make &&
chmod 755 /usr/bin/make
```

Explicación de los comandos

Por defecto, `/usr/bin/make` se instala con kmem como grupo efectivo de ejecución (setgid). Esto es necesario en algunos sistemas para que pueda comprobar la carga media del sistema utilizando `/dev/kmem`. Sin embargo, en los sistemas Linux, cambiar el grupo de ejecución a kmem no es necesario, luego podemos quitar el bit SGID del programa `make`. Esto también soluciona algunos problemas que hacen que `make` ignore ciertas variables, como *LD_LIBRARY_PATH*.

Contenido de make−3.79.1

Programas

make

Descripciones

make

make determina qué partes de un programa necesitan ser recompiladas, y lanza los comandos para hacerlo.

Dependencias

Para instalar Make−3.79.1 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: as, ld
diffutils: cmp
fileutils: chgrp, chmod, install, ls, mv, rm
gcc: cc, cc1, collect2, cpp0, gcc
glibc: getconf
grep: egrep, fgrep, grep
m4: m4
make: make
gawk: gawk
sed: sed
sh−utils: basename, echo, expr, hostname, sleep, uname
texinfo: install−info, makeinfo
textutils: cat, tr

Instalando Modutils−2.4.15

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  2 MB
```

Instalación de Modutils

Instala Modutils ejecutando los siguientes comandos:

```
./configure &&
make &&
make install
```

Contenido de modutils−2.4.12

Programas

depmod, genksyms, insmod, insmod_ksymoops_clean, kallsyms (enlace a insmod), kernelversion, ksyms, lsmod (enlace a insmod), modinfo, modprobe (enlace a insmod) y rmmod

Descripciones

depmod

depmod maneja las descripciones de las dependencias para los módulos del núcleo

genksyms

genksyms lee (de la entrada estándar) la salida de gcc −E source.c y genera un fichero que contiene información sobre la versión.

insmod

insmod instala un módulo dentro del núcleo en ejecución.

insmod_ksymoops_clean

insmod_ksymoops_clean borra los símbolos del núcleo (ksyms) salvados y los módulos a los que no se ha accedido en los últimos 2 días.

kallsyms

kallsyms extrae todos los símbolos del núcleo para la depuración.

kernelversion

kernelversion informa sobre la versión mayor del núcleo en ejecución.

ksyms

ksyms muestra los símbolos exportados del núcleo

lsmod

lsmod muestra información sobre todos los módulos cargados.

modinfo

modinfo examina un fichero objeto asociado con un módulo del núcleo y muestra la información que pueda encontrar.

modprobe

modprobe usa un fichero de dependencias similar a un Makefile creado por depmod, para cargar automáticamente el módulo o módulos necesarios del conjunto de módulos disponibles en el árbol de directorios predefinido.

rmmod

rmmod descarga los módulos del núcleo en ejecución.

Dependencias

Para instalar Modutils−2.4.12 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib, strip
bison: bison
diffutils: cmp
fileutils: chmod, install, ln, mkdir, mv, rm
flex: flex
gcc: cc, cc1, collect2, cpp0, gcc
grep: egrep, grep
make: make
sed: sed
sh−utils: basename, expr, hostname, uname
textutils: cat, tr

Instalando Netkit−base−0.17

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  1 MB
```

Instalación de Netkit−base

Instala Netkit−base ejecutando los siguientes comandos:

```
./configure &&
make &&
make install &&
cd etc.sample &&
cp services protocols /etc
```

Hay otros ficheros en el directorio `etc.sample` que podrían interesarte.

Contenido de netkit−base−0.17

Programas

inetd y ping

Descripciones

inetd

inetd es la madre de todos los demonios (daemons). Escucha las peticiones de conexión y transfiere la llamada al demonio apropiado.

ping

ping envía paquetes ICMP ECHO_REQUEST a otra máquina y determina su tiempo de respuesta.

Dependencias

Para instalar Netkit−base−0.17 se necesita lo siguiente:

bash: sh
binutils: as, ld, strip
fileutils: cp, install, rm
make: make
gcc: cc1, collect2, cpp0, gcc
sed: sed
sh−utils: date
textutils: cat

Instalando Patch−2.5.4

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  2 MB
```

Instalación de Patch

Instala Patch ejecutando los siguientes comandos:

```
export CPPFLAGS=-D_GNU_SOURCE &&
./configure --prefix=/usr &&
unset CPPFLAGS &&
make &&
make install
```

Contenido de patch−2.5.4

Programas

patch

Descripciones

patch

El programa patch modifica un fichero según lo indicado en un fichero de parche. Normalmente un fichero de parche es una lista creada por el programa diff que contiene instrucciones sobre cómo necesita ser modificado un fichero original. Patch se usa mucho para parchear el código fuente pues ahorra bastante tiempo y espacio. Imagina un paquete de 1MB de tamaño. La siguiente versión de ese paquete sólo cambia en dos ficheros con respecto a la primera versión. Se puede distribuir como un nuevo paquete entero de 1MB o sólo como un parche de 1KB con el que actualizar la primera versión para hacerla idéntica a la segunda. Por tanto, si la primera versión está todavía descargada, un parche evita hacer una segunda descarga larga.

Dependencias

Para instalar Patch−2.5.4 se necesita lo siguiente:

bash: sh
binutils: as, ld
diffutils: cmp
fileutils: chmod, install, mv, rm
gcc: cc, cc1, collect2, cpp0, gcc
glibc: getconf
grep: egrep, grep
make: make
sed: sed
sh−utils: echo, expr, hostname, uname
textutils: cat, tr

Instalando Procinfo−18

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  170 KB
```

Instalación de Procinfo

Instala Procinfo ejecutando los siguientes comandos:

```
make LDLIBS=-lncurses &&
make install
```

Explicación de los comandos

`make LDLIBS=-lncurses :` Esto utiliza −lncurses en lugar de −ltermcap cuando se construye procinfo. Se hace porque libtermcap ha sido declarada obsoleta en favor de libncurses.

Contenido de procinfo−18

Programas

lsdev, procinfo y socklist

Descripciones

lsdev

lsdev recopila la información sobre los dispositivos físicos instalados en su ordenador a partir de los ficheros con las interrupciones, puertos de entrada/salida y acceso directo a memoria (DMA) del directorio /proc, facilitando una descripción rápida de qué direcciones de entrada/salida (I/O) y qué canales IRQ y DMA usa cada dispositivo.

procinfo

procinfo recopila algunos datos del sistema a partir del directorio /proc y los muestra en un bonito formato en el dispositivo de salida estándar.

socklist

Es un guión de Perl que facilita una lista de todos los puertos abiertos, enumerando el tipo, puerto, inodo, identificador de usuario (uid), identificador de proceso (pid), descriptor de ficheros (fd) y el programa al que pertenece.

Dependencias

Para instalar Procinfo−18 se necesita lo siguiente:

binutils: as, ld
fileutils: install, mkdir
gcc: cc1, collect2, cpp0, gcc
make: make

Instalando Procps−2.0.7

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  2 MB
```

Instalación de Procps

Instala Procps ejecutando los siguientes comandos:

```
make &&
make XSCPT='' install &&
mv /usr/bin/kill /bin
```

Explicación de los comandos

`make XSCPT='' install:` Esto asigna un valor vacío a la variable de Makefile XSCPT, lo cual deshabilita la instalación de XConsole. De otro modo "Make install" intenta copiar el fichero XConsole en /usr/X11R6/lib/X11/app−defaults, y ese directorio no existe, ya que el entorno X no está instalado

Contenido de procps−2.0.7

Programas

free, kill, oldps, pgrep, pkill, ps, skill, snice, sysctl, tload, top, uptime, vmstat, w y watch

Descripciones

free

free muestra la cantidad total de memoria física y de intercambio (swap) libre y usada en el sistema, al igual que la memoria compartida (shared) y de almacenamiento intermedio (buffers) usada por el núcleo.

kill

kill envía señales a los procesos.

oldps y ps

ps facilita una instantánea de los procesos actuales.

pgrep

pgrep visualiza procesos basándose en el nombre u otros atributos

pkill

pkill envía señales a procesos basándose en el nombre u otros atributos

skill

skill envía señales a procesos que coincidan con un criterio.

snice

snice cambia la prioridad de planificación de procesos que coincidan con un criterio.

sysctl

sysctl modifica los parámetros del núcleo en tiempo de ejecución.

tload

tload imprime un gráfico de la carga actual del sistema en la consola (tty) especificada (o la consola del proceso tload si no se especifica ninguna).

top

top proporciona una vista dinámica de la actividad del procesador en tiempo real.

uptime

uptime facilita en una línea de pantalla la siguiente información: la hora, cuanto tiempo lleva el sistema funcionando, cuantos usuarios hay actualmente dentro del sistema, y la carga del sistema en los últimos 1, 5 y 15 minutos.

vmstat

vmstat muestra información sobre los procesos, memoria, paginación, bloques de entrada/salida y actividad del procesador.

w

w muestra información sobre los usuarios que hay actualmente en el sistema y sus procesos.

watch

watch lanza comandos repetidamente, mostrando sus salidas (a pantalla completa la primera).

Librerías

libproc.so

Descripciones

libproc

libproc es la librería contra la que muchos de los programas de este grupo están enlazados para ahorrar espacio en disco implementando las funciones comunes sólo una vez.

Dependencias

Para instalar Procps−2.0.7 se necesita lo siguiente:

bash: sh
binutils: as, ld, strip
fileutils: install, ln, mv, rm
gcc: cc1, collect2, cpp0, gcc
grep: grep
make: make
gawk: awk
sed: sed
sh−utils: basename, pwd
textutils: sort, tr

Instalando Psmisc−20.2

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  500 KB
```

Instalación de Psmisc

Instala Psmisc ejecutando los siguientes comandos:

```
./configure --prefix=/usr --exec-prefix=/ &&
make &&
make install
```

psmisc instala la página del manual de `/usr/share/man/man1/pidof.1`, sin embargo, el programa pidof no se instala por defecto. Normalmente esto no es ningún problema, ya que más tarde instalaremos el paquete sysvinit, el cual nos facilita una versión mejor del programa pidof.

Es hora de decidir si vas a utilizar el paquete sysvinit que contiene un programa pidof o no. Si vas a hacerlo, debes borrar la página del manual de pidof ejecutando:

```
rm /usr/share/man/man1/pidof.1
```

Si no vas a usar sysvinit, debes completar la instalación de este paquete creando el enlace simbólico `/bin/pidof`. Para ello, ejecuta:

```
cd /bin
ln -s killall pidof
```

Explicación de los comandos

`--exec-prefix=/`: Esto provocará que los programas se instalen en `/bin` en vez de en `/usr/bin`. Los programas de este paquete se usan a menudo en los guiones de inicio, así que deben estar en el directorio `/bin` para que puedan utilizarse cuando la partición `/usr` no se haya montado aún

Contenido de psmisc–20.2

Programas

fuser, killall, pidof (enlace a killall) y pstree

Nota: en LFS no instalamos el enlace pidof por defecto porque usamos en su lugar el programa pidof de sysvinit.

Descripciones

fuser

fuser muestra los números de identificación de los procesos (PID) usando los ficheros o sistemas de ficheros especificados.

killall

killall envía una señal a todos los procesos en ejecución de los comandos especificados .

pidof

Pidof encuentra los identificadores de procesos (PIDs) de los programas mencionados e imprime esos identificadores en la salida estándar.

pstree

pstree muestra los procesos en ejecución como un árbol.

Dependencias

Para instalar Psmisc–20.2 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
bison: bison
binutils: as, ld
diffutils: cmp
fileutils: chmod, install, ls, mkdir, mv, rm
gettext: msgfmt, xgettext
gcc: cc, cc1, collect2, cpp0, gcc
grep: egrep, grep
m4: m4
make: make
gawk: gawk
sed: sed
sh–utils: basename, echo, expr, hostname, sleep, uname
texinfo: makeinfo
textutils: cat, tr

Instalando Reiserfsprogs–3.x.1b

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  7 MB
```

Instalación de Reiserfsprogs

Reiserfsprogs sólo necesita instalarse si tienes la intención de usar el sistema de ficheros reiserfs. Instala Reiserfsprogs ejecutando los siguientes comandos:

```
./configure --mandir=/usr/share/man &&
make &&
make install
```

Explicación de los comandos

`--mandir=/usr/share/man`: Esto asegura que las páginas del manual se instalan en el lugar correcto, mientras que los programas se siguen instalando en `/sbin` como tiene que ser.

Contenido de reiserfsprogs–3.x.0j

Programas

debugreiserfs, mkreiserfs, reiserfsck, resize_reiserfs y unpack

Descripciones

debugreiserfs

debugreiserfs puede ayudar en ocasiones en la resolución de problemas con los sistemas de ficheros reiserfs. Si se le llama sin opciones muestra el superbloque de cualquier sistema de ficheros reiserfs encontrado en el dispositivo.

mkreiserfs

mkreiserfs crea un sistema de ficheros reiserfs.

reiserfsck

reiserfsck chequea un sistema de ficheros reiserfs.

resize_reiserfs

resize_reiserfs se usa para redimensionar un sistema de ficheros reiserfs no montado.

unpack

No hay descripción disponible.

Dependencias

Para instalar Reiserfs−3.x.0j se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, install, ls, rm
gcc: cc1, collect2, cpp0, gcc
grep: egrep, grep
m4: m4
make: make
gawk: gawk
sed: sed
sh−utils: echo, expr, hostname, sleep
texinfo: makeinfo
textutils: cat, tr

Instalando Sed−3.02

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  2 MB
```

Instalación de Sed

Instala Sed ejecutando los siguientes comandos:

```
./configure --prefix=/usr --bindir=/bin &&
make &&
make install
```

Contenido de sed–3.02

Programas

sed

Descripciones

sed

sed es un editor de flujo. Un editor de flujo se usa para realizar transformaciones básicas de texto en un flujo de entrada (un fichero o una tubería).

Dependencias

Para instalar Sed–3.02 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, install, ls, mv, rm
gcc: cc1, collect2, cpp0, gcc
glibc: getconf
grep: egrep, fgrep, grep
m4: m4
make: make
gawk: gawk
sed: sed
sh–utils: echo, expr, hostname, sleep
texinfo: install–info, makeinfo
textutils: cat, tr

Instalando Sh–utils–2.0

```
Estimación del tiempo de construcción:      2 minutos
Estimación del espacio de disco requerido:  11 MB
```

Instalación de Sh–utils

Instala Shellutils ejecutando los siguientes comandos:

```
./configure --prefix=/usr &&
make &&
make install &&
cd /usr/bin &&
mv basename date echo false hostname /bin &&
mv pwd sleep stty su test true uname /bin &&
mv chroot ../sbin
```

Notas sobre la conformidad con el estándar FHS

Hay un comando que se instala con este paquete que se llama test. Se usa a menudo en guiones del intérprete de comandos (shell scripts) para evaluar condiciones, pero habitualmente se encuentra como **[condición]**. Estos corchetes son comandos internos del intérprete bash. Sin embargo, el estándar FHS determina que debe haber un programa llamado `[`. Vamos a crearlo de la forma siguiente, mientras estamos en el directorio `/bin`:

```
cd /bin &&
ln -sf test [
```

Contenido de sh-utils-2.0

Programas

basename, chroot, date, dirname, echo, env, expr, factor, false, groups, hostid, hostname, id, logname, nice, nohup, pathchk, pinky, printenv, printf, pwd, seq, sleep, stty, su, tee, test, true, tty, uname, uptime, users, who, whoami y yes

Descripciones

basename

basename elimina los directorios y las extensiones de los nombres de ficheros.

chroot

chroot ejecuta un comando o un intérprete de comandos (shell) interactivo dentro de un directorio raíz determinado.

date

date muestra la fecha y hora actual en un formato determinado o establece la fecha y hora del sistema.

dirname

dirname elimina los sufijos que no son directorios del nombre de un fichero.

echo

echo muestra una línea de texto.

env

env ejecuta un programa en un entorno modificado.

expr

expr evalúa expresiones.

factor

factor muestra los factores primos de los números especificados.

false

false siempre termina con un código de estado que indica un fallo.

groups

groups muestra los grupos a los que pertenece un usuario.

hostid

hostid muestra el identificador numérico (en hexadecimal) de la máquina actual.

hostname

hostname establece o muestra el nombre de la máquina actual.

id

id muestra el identificador de usuario (UID) real y efectivo y el identificador de grupo (GID) real y efectivo de un usuario.

logname

logname muestra el nombre de acceso (login name) del usuario actual.

nice

nice ejecuta un programa con una prioridad distinta.

nohup

nohup ejecuta un comando que no se interrumpe cuando se cierra la sesión, y cuya salida no se envía a un terminal.

pathchk

pathchk comprueba si los nombres de ficheros son válidos o portables.

pinky

pinky es una utilidad parecida a finger que obtiene información sobre un determinado usuario.

printenv

printenv muestra todo o parte del entorno.

printf

printf formatea y muestra datos (de la misma manera que la función printf de C).

pwd

pwd muestra el nombre del directorio de trabajo actual.

seq

seq muestra números en un cierto rango y con un cierto incremento.

sleep

sleep establece un retardo durante un determinado instante de tiempo.

stty

stty cambia y muestra las opciones de configuración del terminal.

su

su ejecuta un intérprete de comandos (shell) con un identificador de usuario y de grupo diferentes.

tee

tee lee de la entrada estándar y escribe en la salida estándar y en ficheros.

test

test comprueba el tipo de los ficheros y compara valores.

true

true siempre termina con un código de estado que indica éxito.

tty

tty muestra el nombre de fichero del terminal conectado a la entrada estándar.

uname

uname muestra información del sistema.

uptime

uptime muestra cuanto tiempo hace que el sistema está en marcha.

users

users muestra los nombres de los usuarios conectados actualmente.

who

who muestra quién está conectado.

whoami

whoami muestra el identificador de usuario efectivo.

yes

yes muestra en pantalla una cadena de texto indefinidamente.

Dependencias

Para instalar Sh−utils−2.0 se ecesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, chown, install, ls, mv, rm
gettext: msgfmt, xgettext
gcc: cc, cc1, collect2, cpp0, gcc
glibc: getconf
grep: egrep, fgrep, grep
m4: m4
make: make
gawk: gawk
perl: perl
sed: sed
sh−utils: basename, echo, expr, hostname, sleep, uname
tar: tar
texinfo: install−info, makeinfo
textutils: cat, tr

Instalando Net−tools−1.60

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  5 MB
```

Instalación de Net−tools

Instala Net−tools ejecutando los siguientes comandos:

```
make &&
make update
```

Si no te importa aceptar todas las respuestas por defecto, puedes ejecutar en su lugar, estos comandos:

```
yes "" | make &&
make update
```

Si no sabes qué contestar a todas las preguntas que se hacen durante la etapa **make**, entonces sólo acepta los valores por defecto, ya que será lo correcto en la mayoría de los casos. Lo que se pregunta aquí es una serie de cuestiones relativas al tipo de protocolos de red que tienes activados en tu núcleo.

Las respuestas por defecto activarán las herramientas de este paquete para trabajar con la mayoría de los protocolos más comunes como TCP, PPP y algunos otros. En realidad, todavía necesitarás activar esos protocolos en el núcleo. Lo que estás haciendo aquí es, simplemente, ordenar a los programas que sean capaces de usar esos protocolos pero corre de cuenta del núcleo dejarlo disponible para el sistema.

Explicación de los comandos

make update: Esto hace lo mismo que **make install** con la excepción de que make update no hace copias de seguridad de los ficheros que está reemplazando. Una de las cosas que reemplaza net−tools es la versión de sh−utils de `/bin/hostname` (la versión de net−tools es bastante mejor que la de sh−utils).

Además, si decides volver a instalar este paquete en algún momento, ejecutar **make update** no hará copias de seguridad de todos los ficheros de la anterior instalación de net−tools.

Contenido de net−tools−1.60

Programas

arp, dnsdomainname (enlace a hostname), domainname (enlace a hostname), hostname, ifconfig, nameif, netstat, nisdomainname (enlace a hostname), plipconfig, rarp, route, slattach y ypdomainname (enlace a hostname)

Descripciones

arp

arp se usa para manipular la caché ARP del núcleo, usualmente para añadir o borrar una entrada o volcar dicha caché.

dnsdomainname

dnsdomainname muestra el nombre del dominio DNS del sistema.

domainname

domainname muestra o establece el nombre del dominio NIS/YP del sistema.

hostname

hostname se usa para establecer o mostrar el nombre del sistema.

ifconfig

El comando ifconfig es el comando general usado para configurar las interfaces de red.

nameif

nameif nombra interfaces de red basándose en direcciones MAC.

netstat

netstat es una herramienta multipropósito usada para mostrar las conexiones de red, tablas de encaminamiento, estadísticas de las interfaces, conexiones enmascaradas y los miembros de conexiones multidestino (multicast).

nisdomainname

nisdomainname muestra o establece el nombre de dominio NIS/YP del sistema.

plipconfig

plipconfig se usa para afinar los parámetros del dispositivo PLIP, para hacerlo más rápido.

rarp

Relacionado con el programa arp, el programa rarp manipula la tabla RARP del sistema.

route

route es la utilidad general que se usa para manipular la tabla de encaminamiento IP.

slattach

slattach conecta una interfaz de red a una línea serie, esto es, pone una línea de terminal normal en uno o varios modos "de red".

ypdomainname

ypdomainname muestra o establece el nombre de dominio NIS/YP del sistema.

Dependencias

Para instalar Net−tools−1.60 se necesita lo siguiente:

bash: bash, sh
binutils: ar, as, ld
fileutils: install, ln, ls, mv, rm
gcc: cc, cc1, collect2, cpp0
make: make
sh−utils: echo

Instalando Shadow–4.0.3

```
Estimación del tiempo de construcción:      3 minutos
Estimación del espacio de disco requerido:  6 MB
```

Instalación del Entorno de Contraseñas Ocultas (Shadow Password Suite)

Antes de instalar este paquete, puede que quieras echar un vistazo a la receta de lfs http://hints.linuxfromscratch.org/hints/shadowpasswd_plus.txt . En ella se discute cómo puedes hacer tu sistema más seguro respecto a las contraseñas y cómo conseguir exprimir hasta el máximo este paquete.

Instala el Entorno de Contraseñas Ocultas (Shadow Password Suite) ejecutando los siguientes comandos:

```
./configure --prefix=/usr --enable-shared &&
make &&
make install &&
cd etc &&
cp limits login.access /etc &&
sed 's%/var/spool/mail%/var/mail%' login.defs.linux > /etc/login.defs &&
cd /usr/sbin &&
ln -sf vipw vigr &&
rm /bin/vipw &&
mv /bin/sg /usr/bin &&
cd /lib &&
mv libmisc.*a libshadow.*a /usr/lib &&
cd /usr/lib &&
ln -s ../../lib/libshadow.so
```

Explicación de los comandos

`cp limits login.access /etc:` Estos ficheros no se han instalado durante la instalación del paquete, por eso los copiamos manualmente. Esos ficheros se usan para configurar los detalles de autenticación del sistema.

`sed "s%/var/spool/mail%/var/mail%" login.defs.linux > /etc/login.defs`: /var/spool/mail es la antigua localización de los buzones de los usuarios. El lugar que se usa hoy en día es /var/mail.

`ln -sf vipw vigr`: De acuerdo a la página del manual (man) de vipw, vigr debería ser un enlace simbólico a aquél. Como el procedimiento de instalación de shadow no crea este enlace simbólico, lo hacemos manualmente.

Contenido de shadow–20001016

Programas

chage, chfn, chpasswd, chsh, dpasswd, expiry, faillog, gpasswd, groupadd, groupdel, groupmod, grpck, grpconv, grpunconv, lastlog, login, logoutd, mkpasswd, newgrp, newusers, passwd, pwck, pwconv, pwunconv, sg (enlace a newgrp), su, useradd, userdel, usermod, vigr (enlace a vipw) y vipw

Descripciones

chage

chage cambia el número de días entre cambios de la contraseña y la fecha del último cambio de contraseña.

chfn

chfn cambia la información en las cuentas de usuario relativa al nombre completo del usuario, el número y la extensión de su oficina, y el número de teléfono personal.

chpasswd

chpasswd lee un fichero con pares de usuarios y contraseñas en la entrada estándar y usa esta información para actualizar un grupo de usuarios existentes.

chsh

chsh cambia el intérprete de comandos que se ejecuta cuando un usuario entra al sistema.

dpasswd

dpasswd añade, borra y actualiza las contraseñas de acceso telefónico del usuario.

expiry

Comprueba y fuerza la política de expiración de contraseñas.

faillog

faillog formatea el contenido del registro de fallos /var/log/faillog, y mantiene el contador y los límites de fallos.

gpasswd

gpasswd se usa para administrar el fichero /etc/group.

groupadd

El comando groupadd crea un nuevo grupo usando los valores especificados en la línea de comandos y los valores por defecto del sistema.

groupdel

El comando groupdel modifica los ficheros con las cuentas del sistema, borrando todas las entradas referidas a un determinado grupo.

groupmod

El comando groupmod modifica los ficheros de las cuentas del sistema para reflejar los cambios especificados en la línea de comandos.

grpck

grpck verifica la integridad de la información de autenticación del sistema.

grpconv

grpunconv convierte a ficheros de grupos ocultos los ficheros de grupos normales.

grpunconv

grpunconv convierte los ficheros de grupos ocultos en ficheros de grupos normales.

lastlog

lastlog formatea y muestra el contenido del registro de último acceso /var/log/lastlog. Muestra el nombre con el que se ha accedido, puerto y hora del último acceso.

login

login se usa para establecer una nueva sesión con el sistema.

logoutd

logoutd fuerza las restricciones de hora de acceso y puerto especificadas en /etc/porttime.

mkpasswd

mkpasswd lee un fichero en el formato facilitado por las opciones y lo convierte al formato de fichero de base de datos correspondiente.

newgrp

newgrp se usa para cambiar el identificador de grupo actual durante una sesión de acceso.

newusers

newusers lee un fichero con pares de nombres de usuario y contraseñas en texto plano y usa esa información para actualizar un grupo de usuarios existentes o crear nuevos usuarios.

passwd

passwd cambia las contraseñas de las cuentas de usuarios y grupos.

pwck

pwck verifica la integridad de la información de autenticación del sistema.

pwconv

pwconv convierte a ficheros de contraseñas ocultas los ficheros de contraseñas normales.

pwunconv

pwunconv convierte los ficheros de contraseñas ocultas en ficheros normales.

sg

sg ejecuta comandos con un identificador de grupo diferente.

su

Cambia los identificadores efectivos de usuario y grupo de un usuario. Reemplaza al programa su instalado por el paquete Shellutils.

useradd

useradd crea un nuevo usuario o actualiza la información por defecto de un nuevo usuario.

userdel

userdel modifica los ficheros con las cuentas del sistema, borrando todas las entradas referidas al nombre de acceso especificado.

usermod

usermod modifica los ficheros con las cuentas del sistema para reflejar los cambios especificados en la línea de comandos.

vipw y vigr

vipw y vipgr pueden editar los ficheros /etc/passwd y /etc/group, respectivamente. Con la opción −s, pueden editar la versión oculta de dichos ficheros, /etc/shadow y /etc/gshadow, respectivamente.

Librerías

libshadow.[a,so]

Descripciones

libshadow

libshadow proporciona funcionalidades comunes para los programas de ocultación de contraseñas.

Dependencias

Para instalar Shadow−20001016 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: ar, as, ld, nm, ranlib
diffutils: cmp

fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm, rmdir
gettext: msgfmt, xgettext
gcc: cc1, collect2, cpp0, gcc
glibc: ldconfig
grep: egrep, grep
m4: m4
make: make
gawk: gawk
net−tools: hostname
sed: sed
sh−utils: basename, echo, expr, sleep, uname
texinfo: makeinfo
textutils: cat, sort, tr, uniq

Instalando Sysklogd−1.4.1

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  710 KB
```

Instalación de Sysklogd

Instala Sysklogd ejecutando los siguientes comandos:

```
make &&
make install
```

Contenido de sysklogd−1.4.1

Programas

klogd y syslogd

Descripciones

klogd

klogd es un demonio del sistema que intercepta y registra los mensajes del núcleo Linux.

syslogd

syslogd proporciona una forma de registrar sucesos que muchos programas modernos utilizan. Cada mensaje registrado contiene como mínimo un campo con el nombre de la máquina y la fecha, y normalmente un campo con el nombre del programa también, pero eso depende de lo fiable que sea el programa a registrar.

Dependencias

Para instalar Sysklogd−1.4.1 se necesita lo siguiente:

binutils: as, ld, strip
fileutils: install

gcc: cc1, collect2, cpp0, gcc
make: make

Instalando Sysvinit–2.84

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  630 KB
```

Instalación de Sysvinit

Cuando se cambia de nivel de ejecución (por ejemplo cuando vamos a apagar el sistema) el programa init envía las señales TERM y KILL a todos los procesos que él mismo inició. Además, init muestra un mensaje en el monitor que dice "sending all processes the TERM signal" (enviando la señal TERM a todos los procesos) y lo mismo para la señal KILL. Esto implica que init envía esta señal a todos los procesos en ejecución, lo cual no es cierto. Para evitar esta confusión, cambiamos el archivo init.c de manera que el mensaje diga "sending all processes started by init the TERM signal" (enviando la señal TERM a todos los procesos iniciados por init) ejecutando los siguientes comandos. Si no quieres cambiar esto, sáltate este paso.

```
cp src/init.c src/init.c.backup &&
sed 's/\(.*\)\(Sending processes\)\(.*\)/\1\2 started by init\3/' \
   src/init.c.backup > src/init.c
```

Instala Sysvinit ejecutando los siguientes comandos:

```
make -C src &&
make -C src install
```

Contenido de sysvinit–2.84

Programas

halt, init, killall5, last, lastb (enlace a last), mesg, pidof (enlace a killall5), poweroff (enlace a halt), reboot (enlace a halt), runlevel, shutdown, sulogin, telinit (enlace a init), utmpdump y wall

Descripciones

halt

halt anota en el fichero /var/log/wtmp que el sistema se va a venir abajo, y entonces le indica al núcleo que cierre, reinicie o apague el sistema. Si halt o reboot son llamados cuando el sistema no está en los niveles de ejecución 0 o 6, shutdown es invocado en su lugar (con las opciones –h o –r).

init

init es el padre de todos los procesos. Su función principal es crear procesos a partir de un guión almacenado en el fichero /etc/inittab. Este fichero normalmente tiene unas entradas que hacen que init active la creación de los terminales en cada línea en la que los usuarios pueden conectarse. También controla los procesos autónomos requeridos por un sistema particular.

killall5

killall5 es el comando killall de SystemV. Envía una señal a todos los procesos excepto a los procesos de su propia sesión, por tanto no puede matar el intérprete de comandos en el que se esté ejecutando el guión desde el que fue llamado.

last

last busca hacia atrás en el fichero /var/log/wtmp (o el fichero indicado con la opción −f) y muestra una lista con todos los usuarios conectados (y desconectados) desde que el fichero fue creado.

lastb

lastb es lo mismo que last, excepto que por defecto muestra los registros del fichero /var/log/btmp, que contiene todos los intentos fallidos de conexión.

mesg

mesg controla el acceso al terminal de un usuario por otros. Se usa para permitir o denegar a otros usuarios escribir en su terminal.

pidof

pidof encuentra los identificadores de procesos (PIDs) de los programas mencionados y muestra esos identificadores en la salida estándar.

poweroff

poweroff es equivalente a "shutdown −h −p now". Para el ordenador y lo apaga (cuando se usa una BIOS compatible APM y APM está activado en el núcleo).

reboot

reboot es equivalente a "shutdown −r now". Reinicia el ordenador.

runlevel

runlevel lee el fichero utmp del sistema (típicamente /var/run/utmp) para localizar el registro del nivel de ejecución, y entonces muestra el anterior y el nivel de ejecución actual del sistema en la salida estándar, separados por un espacio.

shutdown

shutdown provoca la caída del sistema de una forma segura. Todos los usuarios conectados son notificados de que el sistema se va a venir abajo, y se bloquean los intentos de conexión al sistema.

sulogin

sulogin es invocado por init cuando el sistema entra en el modo monousuario (esto se hace mediante una entrada en /etc/inittab). Init también intenta ejecutar sulogin cuando se le pasa la opción −b desde el gestor de arranque (p.e., LILO).

telinit

telinit envía las señales apropiadas a init, diciendo a qué nivel de ejecución debe cambiar.

utmpdump

utmpdump muestra el contenido de un fichero (usualmente /var/run/utmp) en la salida estándar en un formato comprensible por el usuario.

wall

wall envía un mensaje a todos los usuarios conectados que tengan los permisos de mesg puestos a "yes".

Dependencias

Para instalar Sysvinit−2.84 se necesita lo siguiente:

bash: sh
binutils: as, ld
fileutils: chown, cp, install, ln, mknod, rm
gcc: cc, cc1, collect2, cpp0
make: make
sed: sed

Instalando Tar−1.13

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  7 MB
```

Instalación de Tar

Si quieres usar directamente ficheros bzip2 con tar, utiliza el parche que se encuentra en el servidor FTP de LFS. Este parche añadirá a tar la opción −j, que funciona de la misma forma que la opción −z (usada para los ficheros gzip).

Aplica el parche ejecutando los siguientes comandos:

```
patch -Np1 -i ../tar-1.13.patch
```

Instala Tar ejecutando los siguientes comandos desde el directorio de nivel superior:

```
./configure --prefix=/usr --libexecdir=/usr/bin \
   --bindir=/bin &&
make &&
make install
```

Contenido de tar−1.13

Programas

rmt y tar

Descripciones

rmt

rmt es un programa utilizado por los programas dump y restore para manipular una unidad de cinta magnética mediante una conexión entre procesos.

tar

tar es un programa diseñado para almacenar y extraer ficheros de un archivo conocido como fichero tar.

Dependencias

Para instalar Tar−1.13 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, install, ls, mv, rm
gettext: msgfmt, xgettext
gcc: cc, cc1, collect2, cpp0, gcc
glibc: getconf
grep: egrep, fgrep, grep
m4: m4
make: make
gawk: gawk
net−tools: hostname
patch: patch
sed: sed
sh−utils: basename, echo, expr, sleep, uname
texinfo: install−info, makeinfo
textutils: cat, tr

Instalando Textutils−2.0

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  15 MB
```

Instalación de Textutils

Instala Textutils ejecutando los siguientes comandos:

```
./configure --prefix=/usr &&
make &&
make install &&
```

```
mv /usr/bin/cat /usr/bin/head /bin
```

Contenido de textutils–2.0

Programas

cat, cksum, comm, csplit, cut, expand, fmt, fold, head, join, md5sum, nl, od, paste, pr, ptx, sort, split, sum, tac, tail, tr, tsort, unexpand, uniq y wc

Descripciones

cat

cat concatena ficheros o la entrada estándar en la salida estándar.

cksum

cksum muestra la suma de comprobación CRC y cuenta los bytes de un fichero.

comm

comm compara dos ficheros ordenados línea por línea.

csplit

csplit escribe partes de un fichero separadas por un determinado patrón en ficheros xx01, xx02, etc, y muestra el número de bytes de cada parte en la salida estándar.

cut

cut imprime en la salida estándar las partes seleccionadas de las líneas de un fichero.

expand

expand convierte las tabulaciones de un fichero en espacios, escribiendo en la salida estándar

fmt

fmt reformatea cada párrafo de un fichero, escribiendo en la salida estándar.

fold

fold reajusta la longitud de las líneas de un fichero (por defecto, la entrada estándar), escribiendo en la salida estándar.

head

head imprime en la salida estándar las primeras xx (10 por defecto) líneas de un fichero.

join

join une líneas de dos ficheros en un campo común.

md5sum

md5sum muestra o chequea sumas de comprobación MD5.

nl

nl escribe un fichero en la salida estándar, añadiendo números de línea.

od

od escribe en la salida estándar una representación inequívoca (por defecto en octal) de un fichero.

paste

paste escribe en la entrada estándar líneas formadas por las líneas de cada uno de los ficheros especificados, separadas por tabulaciones.

pr

pr pagina o muestra en columnas el texto de un fichero, para imprimirlo posteriormente.

ptx

ptx genera un índice permutado de los contenidos de un fichero.

sort

sort escribe en la salida estándar una concatenación de ficheros ordenados.

split

split divide un fichero en partes de tamaño fijo llamadas FICHEROaa, FICHERObb,...

sum

sum muestra la suma de comprobación y el número de bloques que ocupa un fichero.

tac

tac escribe un fichero o ficheros en la salida estándar, comenzando por la última línea.

tail

tail imprime en la salida estándar las últimas xx (10 por defecto) líneas de un fichero.

tr

tr convierte, altera y/o borra caracteres de la entrada estándar, escribiendo en la salida estándar.

tsort

tsort escribe una lista totalmente ordenada de acuerdo con el orden parcial de los ficheros especificados.

unexpand

unexpand convierte los espacios de un fichero en tabulaciones, escribiendo en la salida estándar.

uniq

uniq elimina las líneas duplicadas de un fichero ordenado.

wc

wc muestra el número de líneas, palabras y bytes de un fichero, y una línea con el total si se ha especificado más de uno.

Dependencias

Para instalar Textutils−2.0 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, install, ls, mv, rm
gettext: msgfmt, xgettext
gcc: cc, cc1, collect2, cpp0, gcc
glibc: getconf
grep: egrep, fgrep, grep
m4: m4
make: make
gawk: gawk
net−tools: hostname
perl: perl
sed: sed
sh−utils: basename, echo, expr, sleep, uname
tar: tar
texinfo: install−info, makeinfo
textutils: cat, tr

Instalando Util−linux−2.11o

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  9 MB
```

Notas sobre la conformidad con el estándar FHS

El estándar FHS recomienda que usemos /var/lib/hwclock para la ubicación del archivo adjtime, en lugar del habitual /etc. Para hacer que hwclock, el cual es parte del paquete util–linux, sea conforme a FHS, ejecuta lo siguiente.

```
cp hwclock/hwclock.c hwclock/hwclock.c.backup &&
sed 's%etc/adjtime%var/lib/hwclock/adjtime%' \
   hwclock/hwclock.c.backup > hwclock/hwclock.c &&
mkdir -p /var/lib/hwclock
```

Instalación de Util–Linux

Instala Util–Linux ejecutando los siguientes comandos:

```
./configure &&
make HAVE_SLN=yes &&
make HAVE_SLN=yes install
```

Explicación de los comandos

`HAVE_SLN=yes`: No construimos este programa porque ya había sido instalado por Glibc.

Contenido de util–linux–2.11n

Programas

agetty, arch, blockdev, cal, cfdisk, chkdupexe, col, colcrt, colrm, column, ctrlaltdel, cytune, ddate, dmesg, elvtune, fdformat, fdisk, fsck.minix, getopt, hexdump, hwclock, ipcrm, ipcs, isosize kill, line, logger, look, losetup, mcookie, mkfs, mkfs.bfs, mkfs.minix, mkswap, more, mount, namei, pivot_root, ramsize (enlace a rdev), raw, rdev, readprofile, rename, renice, rev, rootflags (enlace a rdev), script, setfdprm, setsid, setterm, sfdisk, swapoff (enlace a swapon), swapon, tunelp, ul, umount, vidmode, whereis y write

Descripciones

agetty

agetty abre un puerto de terminal, espera la introducción de un nombre de usuario, e invoca al comando /bin/login.

arch

arch muestra la arquitectura de la máquina.

blockdev

blockdev permite llamar a los controles de entrada/salida (ioctls) de los dispositivos de bloque desde la línea de comandos.

cal

cal muestra un calendario simple.

cfdisk

cfdisk es un manipulador de la tabla de particiones del disco basado en libncurses.

chkdupexe

chkdupexe encuentra ejecutables duplicados.

col

col filtra avances inversos de línea de la entrada.

colcrt

colcrt filtra la salida de nroff para su visualización en un CRT.

colrm

colrm elimina columnas de un fichero.

column

column muestra listas en columnas.

ctrlaltdel

ctrlaltdel establece la función de la combinación de teclas CTRL+ALT+DEL (reinicio duro o blando).

cytune

cytune afina los parámetros del controlador de Cyclades.

ddate

ddate convierte las fechas Gregorianas en fechas Discordantes.

dmesg

dmesg se usa para examinar o controlar el anillo de almacenamiento intermedio del núcleo (los mensajes de arranque del núcleo).

elvtune

elvtune permite afinar el nivel de entrada/salida por cola de un dispositivo de bloques.

fdformat

fdformat formatea un disquete a bajo nivel.

fdisk

fdisk es un manejador de la tabla de particiones.

fsck.minix

fsck.minix realiza una comprobación de consistencia en sistemas de ficheros MINIX de Linux.

getopt

getops analiza opciones de comandos de la misma forma que el comando getopt de C.

hexdump

hexdump muestra un fichero, o la entrada estándar, en un formato especificado por el usuario (ASCII, decimal, hexadecimal, octal).

hwclock

hwclock interroga y pone en hora el reloj del ordenador (también llamado RTC o reloj BIOS).

ipcrm

ipcrm elimina el recurso especificado.

ipcs

ipcs facilita información sobre las capacidades IPC.

isosize

isosize muestra el tamaño de un sistema de ficheros iso9660.

kill

kill envía la señal especificada al proceso indicado.

line

line copia una línea (hasta una nueva línea) de la entrada estándar y la escribe en la salida estándar.

logger

logger crea entradas en los registros del sistema.

look

look muestra líneas que comiencen con una cadena dada.

losetup

losetup activa y controla los dispositivos de bucle (loop).

mcookie

mcookie genera galletas mágicas (magic cookies) para xauth.

mkfs

mkfs construye un sistema de ficheros Linux en un dispositivo, normalmente una partición del disco duro.

mkfs.bfs

mkfs.bfs crea un sistema de ficheros bfs de SCO en un dispositivo, normalmente una partición del disco duro.

mkfs.minix

mkfs.minix crea un sistema de ficheros Minix en un dispositivo, normalmente una partición del disco duro

mkswap

mkswap configura un área de intercambio (swap) de Linux en un dispositivo o en un fichero.

more

more es un filtro para paginar texto pantalla a pantalla.

mount

mount monta un sistema de ficheros de un dispositivo en un directorio (punto de montaje).

namei

namei sigue el nombre de una ruta hasta que el punto terminal es encontrado.

pivot_root

pivot_root mueve el sistema de ficheros raíz del proceso actual.

ramsize

ramsize muestra y establece el tamaño del disco RAM.

raw

raw se usa para unir un dispositivo Linux de carácter a un dispositivo de bloque.

rdev

rdev muestra y establece el dispositivo raíz de la imagen, el dispositivo de intercambio, el tamaño del disco RAM, o el modo de vídeo.

readprofile

readprofile lee la información de los perfiles del núcleo.

rename

rename renombra ficheros.

renice

renice altera la prioridad de los procesos lanzados.

rev

rev invierte el orden de las líneas de un fichero.

rootflags

rootflags muestra y establece la información extra usada cuando se monta el sistema de ficheros raíz.

script

script registra todo lo que se ha tecleado en la sesión de terminal.

setfdprm

setfdprm establece los parámetros para los disquetes facilitados por el usuario

setsid

setsid lanza programas en una nueva sesión.

setterm

setterm establece los parámetros del terminal.

sfdisk

sfdisk es un manipulador de la tabla de particiones del disco.

swapoff

swapoff desactiva los dispositivos y ficheros de paginación e intercambio.

swapon

swapon activa los dispositivos y ficheros de paginación e intercambio.

tunelp

tunelp establece varios parámetros para el dispositivo de puerto paralelo (LP).

ul

ul lee un fichero y traduce las ocurrencias de marcas de texto a la secuencia que indica subrayado, para el terminal en uso.

umount

umount desmonta un sistema de ficheros montado.

vidmode

vidmode muestra y establece el modo de vídeo.

whereis

whereis localiza el binario, la fuente y la página de manual de un comando.

write

write envía un mensaje a otro usuario

Dependencias

Para instalar Util−linux−2.11n se necesita lo siguiente:

bash: sh
binutils: as, ld
diffutils: cmp
fileutils: chgrp, chmod, cp, install, ln, mv, rm
gettext: msgfmt, xgettext
gcc: cc, cc1, collect2, cpp, cpp0
glibc: rpcgen
grep: grep
make: make
sed: sed
sh−utils: uname, whoami
textutils: cat

Instalando LFS−Bootscripts−1.9

```
Estimación del tiempo de construcción:      1 minuto
Estimación del espacio de disco requerido:  23 KB
```

Instalación de LFS−Bootscripts

Nosotros usamos guiones de inicio al estilo SysV. Lo hemos elegido porque es ampliamente usado y nos sentimos cómodos con él. Si quieres probar alguna otra cosa, alguien ha escrito una receta para LFS con guiones de arranque al estilo BSD. Está en: http://hints.linuxfromscratch.org/hints/bsd−init.txt.

Si decides usar el estilo BSD o cualquier otro estilo de guiones, puedes saltarte el capítulo 7 e ir directamente al capítulo 8.

Instala LFS−Bootscripts ejecutando los siguientes comandos:

```
cp −a rc.d sysconfig /etc &&
chown −R root.root /etc/rc.d /etc/sysconfig
```

Contenido de LFS–bootscripts–1.9

Ficheros

checkfs, cleanfs, functions, halt, loadkeys, localnet, mountfs, network, rc, reboot, sendsignals, setclock, swap, sysklogd y template

Descripciones

checkfs

El guión checkfs comprueba los sistemas de ficheros justo antes de ser montados (con la excepción de los que usan registros de transacciones [journal] o los que se montan desde la red)

cleanfs

El guión cleanfs elimina los ficheros que no deben guardarse cuando se arranca de nuevo el sistema, como /var/run/* y /var/lock/*, regenera /var/run/utmp y elimina los ficheros /etc/nologin, /fastboot y /forcefsck que pueda haber.

functions

El guión functions contiene funciones usadas por diferentes guiones: chequeo de errores, chequeo de estado, etc.

halt

El guión halt se encarga de cerrar el sistema.

loadkeys

El guión loadkeys carga el mapa de teclado apropiado para tu modelo de teclado.

localnet

El guión localnet establece el nombre de máquina usado por el sistema (hostname) y activa el dispositivo de red "loopback".

mountfs

El guión mountfs monta todos los sistemas de ficheros que no estén marcados como "noauto" o que no se monten a través de la red.

network

El guión network activa las interfaces de red (como las tarjetas de red) y establece la puerta de enlace por defecto (gateway) cuando es necesario.

rc

El guión rc es el controlador maestro de los niveles de arranque y es responsable de lanzar todos los demás guiones uno a uno en una secuencia específica.

reboot

El guión reboot se encarga de reiniciar el sistema.

sendsignals

El guión sendsignals se asegura de que todos los procesos terminen antes de parar o reiniciar el sistema.

setclock

El guión setclock fija el reloj del núcleo a la hora local en caso de que el reloj del ordenador no esté fijado a la hora GMT.

swap

El guión swap activa y desactiva las particiones y ficheros de intercambio (swap).

sysklogd

El guión sysklogd lanza y detiene los demonios de registro de eventos del sistema y del núcleo (syslogd y klogd).

template

El guión template es una plantilla para crear guiones de arranque personales para otros demonios y procesos.

Dependencias

Para instalar bootscripts−1.9 se necesita lo siguiente:

fileutils: chown, cp

Borrando los ficheros de la antigua librería NSS

Si has copiado los archivos de la librería NSS desde tu sistema Linux normal al sistema LFS (porque el sistema normal corre glibc−2.0) es hora de borrarlos ejecutando:

```
rm /lib/libnss*.so.1 /lib/libnss*2.0*
```

Configurando los programas esenciales

Ahora que están todos los paquetes instalados, todo lo que tenemos que hacer para conseguir que algunos programas funcionen correctamente es crear sus ficheros de configuración.

Configurando Vim

Por defecto, Vim se ejecuta en modo compatible con vi. Hay gente a la que le puede gustar esto, pero nosotros aconsejamos ejecutar vim en modo vim (de otra forma, no habríamos incluido Vim en este libro, sino el Vi original). Crea el fichero `/root/.vimrc` ejecutando lo siguiente:

```
cat > /root/.vimrc << "EOF"
# Inicio de /root/.vimrc

set nocompatible
set bs=2

# Fin de /root/.vimrc
EOF
```

Configurando Glibc

Necesitamos crear el fichero /etc/nsswitch.conf. Aunque glibc debería darnos valores por defecto cuando este fichero no se encuentra o está corrupto, estos valores por defecto no funcionan bien con la conexión de red, lo cual se tratará en un capítulo posterior. También tendremos que configurar nuestra zona horaria.

Crea un fichero nuevo `/etc/nsswitch.conf` ejecutando lo siguiente:

```
cat > /etc/nsswitch.conf << "EOF"
# Inicio de /etc/nsswitch.conf

passwd: files
group: files
shadow: files

publickey: files

hosts: files dns
networks: files

protocols: db files
services: db files
ethers: db files
rpc: db files

netgroup: db files

# Fin de /etc/nsswitch.conf
EOF
```

Tienes que ejecutar el guión **`tzselect`** y contestar a las preguntas referentes a tu zona horaria. Cuando lo hayas hecho, el guión te dará la localización del fichero con la zona horaria requerida.

Crea el enlace simbólico `/etc/localtime` ejecutando:

```
cd /etc &&
ln -sf ../usr/share/zoneinfo/<salida de tzselect> localtime
```

La salida de tzselect puede ser algo como *EST5EDT* o *Canada/Eastern*.

El enlace simbólico que tienes que crear con esa información será:

```
ln -sf ../usr/share/zoneinfo/EST5EDT localtime
```

O:

```
ln -sf ../usr/share/zoneinfo/Canada/Eastern localtime
```

Configurando el cargador dinámico

Por defecto, el cargador dinámico (`/lib/ld-linux.so.2`) busca en `/lib` y `/usr/lib` las librerías dinámicas que necesitan los programas cuando los ejecutas. No obstante, si hay librerías en otros directorios que no sean `/lib` y `/usr/lib`, necesitas añadirlos en el fichero de configuración `/etc/ld.so.conf` para que el cargador dinámico pueda encontrarlas. Dos directorios típicos que contienen librerías adicionales son `/usr/local/lib` y `/opt/lib`, así que añadimos estos directorios a la ruta de búsqueda del cargador dinámico.

Crea un nuevo fichero `/etc/ld.so.conf` ejecutando lo siguiente:

```
cat > /etc/ld.so.conf << "EOF"
# Inicio de /etc/ld.so.conf

/usr/local/lib
/opt/lib

# Fin de /etc/ld.so.conf
EOF
```

Configurando Sysklogd

Crea un nuevo fichero `/etc/syslog.conf` ejecutando lo siguiente:

```
cat > /etc/syslog.conf << "EOF"
# Inicio de /etc/syslog.conf

auth,authpriv.* -/var/log/auth.log
*.*;auth,authpriv.none -/var/log/sys.log
daemon.* -/var/log/daemon.log
kern.* -/var/log/kern.log
mail.* -/var/log/mail.log
user.* -/var/log/user.log
*.emerg *

# Fin de /etc/syslog.conf
EOF
```

Configurando el entorno de contraseñas ocultas (Shadow Password Suite)

Este paquete contiene las utilidades para modificar las contraseñas de los usuarios, añadir nuevos usuarios/grupos, borrar usuarios/grupos, etc. No vamos a explicar lo que significa 'password shadowing'. Todas estas cosas pueden leerse en el fichero doc/HOWTO que está en el árbol de fuentes de la Shadow Password Suite al desempaquetarla. Hay una cosa que debes recordar si decides usar soporte para contraseñas ocultas: Aquellos programas que necesiten verificar contraseñas (por ejemplo xdm, demonios de ftp, demonios de pop3, etc) necesitarán ser 'compatibles con shadow', es decir, necesitan ser capaces de trabajar con 'shadow passwords'.

Para habilitar las contraseñas ocultas, ejecuta el siguiente comando:

```
/usr/sbin/pwconv
```

Configurando Sysvinit

Crea un nuevo fichero `/etc/inittab` ejecutando lo siguiente:

```
cat > /etc/inittab << "EOF"
# Inicio de /etc/inittab

id:3:initdefault:

si::sysinit:/etc/rc.d/init.d/rc sysinit

l0:0:wait:/etc/rc.d/init.d/rc 0
l1:S1:wait:/etc/rc.d/init.d/rc 1
l2:2:wait:/etc/rc.d/init.d/rc 2
l3:3:wait:/etc/rc.d/init.d/rc 3
l4:4:wait:/etc/rc.d/init.d/rc 4
l5:5:wait:/etc/rc.d/init.d/rc 5
l6:6:wait:/etc/rc.d/init.d/rc 6

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

su:S016:respawn:/sbin/sulogin

1:2345:respawn:/sbin/agetty tty1 9600
2:2345:respawn:/sbin/agetty tty2 9600
3:2345:respawn:/sbin/agetty tty3 9600
4:2345:respawn:/sbin/agetty tty4 9600
5:2345:respawn:/sbin/agetty tty5 9600
6:2345:respawn:/sbin/agetty tty6 9600

# Fin de /etc/inittab
EOF
```

Configurando tu teclado

Nada es más molesto que usar Linux con el mapa de teclas incorrecto. Si tienes un teclado estándar de US (EEUU), te puedes saltar esta sección. El mapa de teclado US es el mapa por defecto si no lo cambias.

Para asignar un mapa de teclado por defecto, crea el enlace simbólico `/usr/share/kbd/keymaps/defkeymap.map.gz` ejecutando los siguientes comandos:

```
cd /usr/share/kbd/keymaps &&
ln -s <ruta/al/mapa/de/teclado> defkeymap.map.gz
```

Reemplaza <ruta/al/mapa/del/teclado> por tu fichero de mapa de teclado. Por ejemplo, si tienes un teclado alemán, deberías ejecutar:

```
ln -s i386/qwerty/nl.map.gz defkeymap.map.gz
```

La segunda opción para configurar la composición de tu teclado es compilar el mapa de teclado directamente en el núcleo. Esto asegurará que tu teclado siempre funcione como se espera, incluso cuando has arrancado en modo de rescate (pasando `init=/bin/sh' al núcleo) y los guiones de arranque que normalmente se encargan de cargar el mapa de teclado adecuado no se hayan ejecutado.

Si no creaste el fichero defkeymap.map.gz y manejas el mapa de teclado por defecto (US), entonces, de nuevo, no tienes que hacer nada. El núcleo lleva un mapa de teclado por defecto apropiado que funcionará bien para ti, así que pasa al siguiente comando.

Ejecuta el siguiente comando para llevar a cabo eso:

```
loadkeys -m /usr/share/kbd/keymaps/defkeymap.map.gz > \
   /usr/src/linux/drivers/char/defkeymap.c
```

Creando los ficheros /var/run/utmp, /var/log/wtmp y /var/log/btmp

Programas como login, shutdown, uptime y otros necesitan leer y escribir en /var/run/utmp, /var/log/btmp y /var/log/wtmp. Estos ficheros contienen información acerca de quién está conectado en ese momento. También contienen información acerca de cuándo fue arrancado y parado por última vez el ordenador y un registro de los intentos de conexión fallidos.

Crea estos ficheros con los permisos apropiados ejecutando los siguientes comandos:

```
touch /var/run/utmp /var/log/{btmp,lastlog,wtmp} &&
chmod 644 /var/run/utmp /var/log/{btmp,lastlog,wtmp}
```

Creando la contraseña de root

Elige una contraseña para el usuario administrador (root) y créala ejecutando el siguiente comando:

```
passwd root
```

Capítulo 7. Preparando los guiones de arranque

Introducción

En este capítulo se configurarán los guiones de arranque que has instalado en el capítulo 6. Muchos de estos guiones funcionarán sin necesidad de modificaciones, pero algunos requieren ficheros de configuración adicionales para que puedan manejar la información dependiente del hardware específico de nuestro sistema.

¿Cómo hacen estos guiones que funcione el proceso de arranque?

Linux utiliza como sistema de inicio SysVinit, que se basa en el concepto de *niveles de ejecución*. Este sistema de inicio puede variar ampliamente de un sistema a otro, por lo tanto, no se debe asumir que porque las cosas funcionen en <inserte el nombre de una distribución> tengan que funcionar en LFS también. LFS tiene su propia manera de hacer las cosas, la cual suele respetar los estándares aceptados.

SysVinit (al que llamaremos *init* a partir de este momento) se basa en un esquema de niveles de ejecución. Hay 7 (desde el 0 al 6) niveles de ejecución (en realidad, existen más pero son para casos especiales y es raro utilizarlos. Puedes leer la página man de init para obtener más información), y cada uno de ellos indica lo que debe hacer el sistema durante el arranque. El nivel de ejecución por omisión es el 3. He aquí una breve descripción de los distintos niveles de ejecución como suelen implementarse:

0: parada del sistema
1: modo monousuario
2: modo multiusuario sin red
3: modo multiusuario con red
4: reservado para personalizar, si no, hace lo mismo que el 3
5: Igual que el 4. Normalmente se utiliza para iniciar el entorno gráfico (como xdm de X o kdm de KDE)
6: reinicio del sistema

Para cambiar el nivel de ejecución se utiliza el comando **`init <nivel de ejecución>`** donde <nivel de ejecución> representa el nivel de ejecución que deseemos arrancar. Por ejemplo, para reiniciar el sistema se utilizaría el comando init 6. El comando reboot no es más que un alias de dicho comando, al igual que el comando halt lo es de init 0.

Debajo de /etc/rc.d existen una serie de directorios rc?.d, donde ? representa el número del nivel de ejecución, y el directorio rcsysinit.d, que contienen un conjunto de enlaces simbólicos. Los nombres de estos enlaces simbólicos empiezan con K o con S seguidos de 2 cifras. Los enlaces que comienzan por una K indican la parada (kill) de un servicio, mientras que la S indica su inicio (start). Las dos cifras determinan el orden de ejecución, desde 00 hasta 99; cuanto menor sea el número antes se ejecutará. En el momento que se desee cambiar de nivel se pararán los servicios del nivel actual, para iniciar los del nuevo nivel.

Los enlaces simbólicos apuntan a los guiones situados en el directorio /etc/rc.d/init.d, que son los que realmente se ejecutan. Tanto los enlaces de parada como los de inicio apuntan al mismo guión. Esto se debe a que se pueden ejecutar usando parámetros como start, stop, restart, reload, status. Cuando se encuentra un enlace que comienza por K se ejecuta el guión con el parámetro stop. Y cuando comienza por S con el parámetro start.

Hay una excepción. Los enlaces que comienzan por S en los directorios rc0.d y rc6.d no inician nada. Todos estos guiones se ejecutan con el parámetro *stop* para parar algo. Es evidente que cuando quieres apagar o reiniciar el sistema, no quieres ejecutar nada, sólo quieres pararlo.

He aquí una descripción de lo que hace cada parámetro:

- *start*: Inicia el servicio.
- *stop*: Para el servicio.
- *restart*: El servicio se para y se vuelve a iniciar.
- *reload*: Se actualiza la configuración del servicio. Este parámetro se utiliza tras la modificación del fichero de configuración, cuando no se necesita reiniciar el servicio para que actualice su configuración.
- *status*: Dice si el servicio se está ejecutando y con qué PID.

Por supuesto, puedes modificar el proceso de inicio para adecuarlo a tus necesidades (después de todo es tu sistema LFS, no el nuestro). Lo aquí expuesto es un ejemplo de cómo nos gusta a nosotros (claro que lo que a nosotros nos gusta, puede que tú lo odies).

Configurando el guión setclock

El guión setclock lee la hora del reloj interno del ordenador (también conocido como el reloj CMOS o BIOS) y la convierte a la hora local mediante el fichero `/etc/localtime` (si el reloj interno del ordenador computadora utiliza GMT) o no (si el reloj interno de la computadora ya está puesto a la hora local). No hay manera de detectar automáticamente si el reloj utiliza GMT o no, así que necesitamos configurarlo nosotros mismos.

Si el reloj interno del ordenador no utiliza GMT hay que cambiar el valor de la variable *UTC* a *0* (cero).

Para ello vamos a crear el fichero `/etc/sysconfig/clock` mediante la ejecución del siguiente comando:

```
cat > /etc/sysconfig/clock << "EOF"
# Inicio de /etc/sysconfig/clock

UTC=1

# Fin de /etc/sysconfig/clock
EOF
```

Para más información sobre la hora en LFS tienes una buena explicación en http://hints.linuxfromscratch.org/hints/time.txt. En él se explican conceptos como las zonas horarias, UTC, y la variable de entorno TZ.

¿Necesito el guión loadkeys?

Si decidiste compilar tu fichero de mapa de caracteres dentro del núcleo al final del capítulo 6 no es estrictamente necesario ejecutar el guión loadkeys, ya que será el núcleo el que activará dicho mapa. Aunque puedes ejecutarlo de todas maneras sin que te cause ningún problema. De todas maneras, es beneficioso que mantengas el guión en el caso de tener varios núcleos y no te has acordado o no has querido introducir el fichero de mapa de caracteres en todos ellos.

Si has decidido que no necesitas o que no quieres el guión loadkeys, elimina el enlace simbólico `/etc/rc.d/rcsysinit.d/S70loadkeys`

Configurando el guión sysklogd

El guión `sysklogd` invoca al programa **`syslogd`** con la opción *−m 0*. Esta opción deshabilita la marca de tiempo periódica que se escribe por defecto en el fichero de registro cada 20 minutos. Si quieres habilitar esta marca de tiempo periódica debes editar el guión `sysklogd` y realizar los cambios necesarios. Para más información utiliza el comando **`man syslogd`**.

Configurando el guión localnet

Una de las cosas que hace el guión localnet es establecer el nombre de la máquina. Es necesario configurar dicho nombre en /etc/sysconfig/network.

Puedes crear el fichero /etc/sysconfig/network y configurar el nombre de tu máquina ejecutando:

```
echo "HOSTNAME=lfs" > /etc/sysconfig/network
```

Debes substituir "lfs" por el nombre de tu máquina. No debes escribir el FQDN (nombre completo de la máquina, incluido su dominio). Esta información la escribiremos más tarde en el fichero `/etc/hosts`

Creando el fichero /etc/hosts

Si se va a configurar una tarjeta de red, debes decidir la dirección IP, el FQDN y los posibles alias para escribirlos en el fichero /etc/hosts. La sintaxis es:

```
<dirección IP> minombre.midominio.org alias
```

Debes asegurarte de utilizar una dirección IP que pertenezca al rango de direcciones IP privadas. Los rangos válidos son:

```
        Clases de redes
        A     10.0.0.0
        B     Entre 172.16.0.0 y 172.31.0.0
        C     Entre 192.168.0.0 y 192.168.255.0
```

Una dirección IP válida puede ser 192.168.1.1. Un FQDN válido para esa dirección IP puede ser www.linuxfromscratch.org.

Aunque no vayas a configurar la tarjeta de red necesitas un FQDN, ya que algunos programas lo necesitan para funcionar correctamente.

Si no vas a configurar la tarjeta de red crea el fichero `/etc/hosts` ejecutando:

```
cat > /etc/hosts << "EOF"
# Inicio de /etc/hosts (versión sin tarjeta de red)

127.0.0.1 www.midominio.com <valor de HOSTNAME> localhost

# Fin de /etc/hosts (versión sin tarjeta de red)
```

```
EOF
```

Si vas a configurar la tarjeta de red crea el fichero `/etc/hosts` ejecutando:

```
cat > /etc/hosts << "EOF"
# Inicio de /etc/hosts (versión con tarjeta de red)

127.0.0.1 localhost.localdomain localhost
192.168.1.1 www.midominio.org <valor de HOSTNAME>

# Fin de /etc/hosts (versión con tarjeta de red)
EOF
```

Por supuesto, puedes cambiar 192.168.1.1 y www.midominio.org a tu gusto (o a lo que te indique el administrador de la red/sistema si está planeado que se conecte esta máquina a una red que ya existe).

Configurando el guión network

Esta sección solamente es aplicable en el caso de que vayas a configurar una tarjeta de red.

Si no tienes tarjeta de red es muy probable que no vayas a crear ninguna configuración relacionada con ellas. En ese caso, debes eliminar los enlaces simbólicos a `network` de todos los directorios de los niveles de ejecución (`/etc/rc.d/rc*.d`)

Configuración de la puerta de enlace por defecto

Si estás conectado a una red puede que necesites establecer cual es la puerta de enlace por defecto para esa máquina. Para ello, se deben añadir los valores apropiados al fichero /etc/sysconfig/network ejecutando lo siguiente:

```
cat >> /etc/sysconfig/network << "EOF"
GATEWAY=192.168.1.2
GATEWAY_IF=eth0
EOF
```

Debes cambiar los valores de GATEWAY y GATEWAY_IF por los que correspondan en tu red. GATEWAY contiene la dirección IP de la puerta de enlace por omisión, y GATEWAY_IF la interfaz de red por la que es accesible dicha dirección IP.

Creando los ficheros de configuración de la interfaz de red

Qué interfaces de red activa o desactiva el guión network depende de los ficheros situados en el directorio /etc/sysconfig/network–devices. Este directorio debe contener ficheros con el nombre ifconfig.xyz, donde xyz corresponde con el nombre de la interfaz de red (como eth0 o eth0:1).

Si decides renombrar o mover el directorio /etc/sysconfig/network–devices, asegúrate de que actualizas el fichero /etc/sysconfig/rc, asignando a la variable network_devices la nueva localización.

Ahora, los nuevos ficheros que creemos en este directorio contienen lo siguiente. Como ejemplo vamos a crear el fichero ifconfig.eth0 ejecutando:

```
cat > /etc/sysconfig/network-devices/ifconfig.eth0 << "EOF"
ONBOOT=yes
IP=192.168.1.1
NETMASK=255.255.255.0
BROADCAST=192.168.1.255
EOF
```

Por supuesto, los valores de estas variables se deben cambiar en todos los ficheros que creemos por los valores apropiados para nuestra máquina. Si la variable ONBOOT tiene el valor yes, el guión network activará la interfaz durante el arranque del sistema. Si contiene cualquier otro valor, el guión network ignorará el contenido del archivo y, por lo tanto, no la activará.

Capítulo 8. Haciendo el sistema LFS arrancable

Introducción

Este capítulo hará arrancable el sistema LFS. Trataremos la creación de un nuevo fichero fstab, la construcción de un nuevo núcleo para el nuevo sistema LFS y la adición de las entradas apropiadas a LILO para que el sistema LFS se pueda seleccionar en la línea de comandos de LILO.

Creando el fichero /etc/fstab

Para que ciertos programas sean capaces de determinar dónde se supone que están montadas por defecto las particiones, se usa el fichero /etc/fstab. Crea un nuevo fichero `/etc/fstab` conteniendo lo siguiente:

```
cat > /etc/fstab << "EOF"
# Inicio de /etc/fstab

# sistema de   punto de      tipo del       opciones  volcado  orden de
# archivos       montaje     sistema de                                 chequeo
#                                       archivos

/dev/*LFS*       /                  *tipo*           defaults       1            1
/dev/*swap*     swap            swap          pri=1           0            0
proc                 /proc            proc           defaults        0            0

# Fin de /etc/fstab
EOF
```

LFS, ***swap*** y ***tipo*** deben ser reemplazados por los valores apropiados (/dev/hda2, /dev/hda5 y reiserfs, por ejemplo).

Cuando se añada una partición reiserfs, los valores **1 1** que aparecen al final de la línea deberían cambiarse a **0 0**.

Para más información sobre los campos que aparecen en el fichero fstab, ver **man 5 fstab**.

Existen otras líneas que puedes considerar añadir al fichero fstab. Un ejemplo es la línea que debería contener si estás utilizando devpts:

```
devpts         /dev/pts    devpts     gid=4,mode=620    0    0
```

Otro ejemplo es la línea a usar si pretendes utilizar dispositivos USB:

```
usbdevfs       /proc/bus/usb usbdevfs defaults    0       0
```

Tanto una como otra opción sólo funcionarán si se tiene el soporte pertinente compilado dentro del núcleo.

Instalando linux–2.4.18

```
Estimación del tiempo de construcción:           Depende de las opciones seleccionadas
Estimación del espacio de disco requerido:  Depende de las opciones seleccionadas
```

Construir el núcleo comprende dos pasos: configurarlo y compilarlo. Hay varias maneras de configurar el núcleo. Si no te gusta la que utiliza este libro, lee el fichero `README` que acompaña al árbol de código fuente del núcleo, y busca qué otras opciones existen.

Lo que puedes hacer es tomar el fichero `.config` del código fuente del núcleo de tu distribución original y copiarlo al directorio `$LFS/usr/src/linux`. De esta forma, no tienes que configurar el núcleo completo desde cero y puedes usar tus valores actuales. Si eliges hacerlo de esta manera, primero ejecuta el comando **`make mrproper`** que aparece debajo, después copia el fichero `.config` y, a continuación, ejecuta **`make menuconfig`** seguido del resto de comandos (puede que **`make oldconfig`** sea mejor elección en algunas situaciones. Lee el fichero `README` para más detalles sobre cuándo se usa **`make oldconfig`**).

Los siguientes comandos se ejecutan para construir el núcleo

```
cd /usr/src/linux &&
make mrproper &&
make menuconfig &&
make dep &&
make bzImage &&
make modules &&
make modules_install &&
cp arch/i386/boot/bzImage /boot/lfskernel &&
cp System.map /boot
```

Nota: la ruta a arch/i386/boot/bzImage puede variar en las distintas plataformas.

Dependencias

Para instalar Linux−2.4.18 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, nm, objcopy
fileutils: cp, ln, mkdir, mv, rm, touch
findutils: find, xargs
gcc: cc1, collect2, cpp0, gcc
grep: grep
gzip: gzip
make: make
gawk: awk
modutils: depmod, genksyms
net−tools: dnsdomainname, hostname
sed: sed
sh−utils: basename, date, expr, pwd, stty, uname, whoami, yes
textutils: cat, md5sum, sort, tail

Haciendo el sistema LFS arrancable

Para poder arrancar el sistema LFS, necesitamos actualizar nuestro gestor de arranque. Asumiremos que tu sistema original usa Lilo (ya que es el gestor de arranque más comúnmente usado en este momento).

No ejecutaremos el programa lilo dentro del entorno chroot. Ejecutar lilo dentro de un entorno chroot puede

acarrear fatales efectos secundarios que lleguen a inutilizar tu Registro Maestro de Arranque (MBR – Master Boot Record), y necesitarías un disco de arranque para poder comenzar cualquier sistema Linux (tanto el original como el sistema LFS).

Primero, saldremos del entorno chroot y copiaremos el fichero lfskernel al sistema original:

```
logout
cp $LFS/boot/lfskernel /boot
```

El siguiente paso es añadir una entrada en /etc/lilo.conf, para que podamos escoger a LFS cuando reiniciemos el ordenador:

```
cat >> /etc/lilo.conf << "EOF"
image=/boot/lfskernel
        label=lfs
        root=<partición>
        read-only
EOF
```

<partición> debe ser reemplazada con el nombre de la partición LFS.

Advierte que si estás utilizando el sistema de ficheros reiserfs para tu partición raíz, se debe cambiar la línea **`read-only`** por **`read-write`**.

Ahora, actualiza el gestor de arranque ejecutando:

```
/sbin/lilo -v
```

El último paso es sincronizar los ficheros de configuración de lilo del sistema original con los del sistema LFS:

```
cp /etc/lilo.conf $LFS/etc &&
cp $(grep "image.*=" /etc/lilo.conf | cut -f 2 -d "=") $LFS/boot
```

Capítulo 9. El final

El final

¡Bien hecho! Has terminado de instalar tu sistema LFS. Puede que haya sido un proceso largo pero ha merecido la pena. Te deseamos mucha diversión con tu flamante sistema Linux hecho a la medida.

Ahora podría ser un buen momento para quitar todos los símbolos de depurado de los archivos binarios de tu sistema LFS. Si no eres un programador y no planeas depurar tus programas, entonces te alegrará saber que puedes recuperar algunas decenas de megabytes borrando estos símbolos. Este proceso no produce ningún otro inconveniente que no sea no poder depurar los programas nunca más, lo que no es problema si no sabes cómo depurarlos.

Advertencia: El 98% de la gente que usa el comando mencionado más adelante no experimenta ningún problema. Pero haz una copia de seguridad de tu sistema LFS antes de ejecutar este comando. Hay una pequeña posibilidad de que te salga el tiro por la culata, y convierta tu sistema en inutilizable (principalmente destruyendo los módulos del núcleo y las librerías dinámicas y compartidas). Sin embargo, suele ocurrir más a menudo por un error tipográfico que por un problema con el comando utilizado.

Después de haber dicho esto, la opción −−strip−debug que usaremos para quitar los símbolos de depuración es, bajo circunstancias normales, bastante inocua. No borrará nada vital de los ficheros. También es bastante seguro usar −−strip−all con programas normales (no se debe usar en librerías −se destruirían), pero no es tan seguro como el anterior y el espacio que ganas no es tan grande. Pero si andas justo de espacio de disco, cada granito de arena ayuda, así que decide por ti mismo. Por favor, lee la página del manual (man) de strip para ver las opciones que puedes usar. La idea general es no ejecutar strip sobre librerías (usando otra opción que no sea −−strip−debug) para asegurarnos de hacer la apuesta segura.

```
find $LFS/{,usr/,usr/local/}{bin,sbin,lib} -type f \
   -exec /usr/bin/strip --strip-debug '{}' ';'
```

Puede ser una buena idea crear el fichero $LFS/etc/lfs−3.3. Teniendo este fichero, te será muy fácil (y a nosotros, si es que vas a pedir ayuda en algún momento) saber qué versión de LFS tienes instalada en tu sistema. Este archivo puede ser un archivo vacío, ejecutando:

```
touch $LFS/etc/lfs-3.3
```

Registrarse

¿Quieres registrarte como usuario de LFS ahora que has terminado el libro? Visita http://linuxfromscratch.org/cgi−bin/lfscounter.cgi y regístrate como usuario de LFS introduciendo tu nombre y la primera versión de LFS que has usado.

Arranquemos el sistema LFS ahora...

Arrancando el sistema

Ahora que se han instalado todos los programas y se han creado los guiones de arranque, ya es hora de reiniciar el ordenador. Antes de reiniciar, desmontemos $LFS/proc y la partición LFS, ejecutando

```
umount $LFS/proc &&
umount $LFS
```

Y ahora puedes reiniciar el sistema ejecutando algo como:

```
/sbin/shutdown -r now
```

Asegúrate de indicar, en la línea de órdenes de LILO, que arranque *lfs* y no la entrada por defecto, que haría arrancar de nuevo tu sistema original.

Una vez hayas reiniciado, tu sistema LFS está listo para su uso, y puedes empezar a añadir los programas que desees.

Una última cosa que puede que quieras hacer es ejecutar lilo, ahora que ya estás dentro del sistema LFS. De esta forma, pondrás en el MBR la versión de LILO del sistema LFS, en lugar de la que hay allí en este momento, que es la de tu sistema original. Dependiendo de como sea de antigua tu distribución original, la versión del sistema LFS puede tener más características avanzadas que necesites (y puedes) usar.

De cualquier forma, ejecuta lo siguiente para activar la versión de lilo instalada en el sistema LFS:

```
/sbin/lilo
```

SI te estás preguntando: "Bien, ¿dónde acudo ahora?"; te alegrará escuchar que alguien ha escrito una receta en LFS sobre el tema en http://hints.linuxfromscratch.org/hints/afterlfs.txt. De igual manera, si no sólo eres novato en LFS, sino en Linux en general, quizá encuentres el consejo para novatos en http://hints.linuxfromscratch.org/hints/newbie.txt muy interesante.

No olvides que existen varias listas de correo en LFS a las que puedes suscribirte si necesitas ayuda, consejo, etc. Visita el Capítulo 1 – Listas de correo y archivos para más información.

De nuevo, te agradecemos el haber utilizado el libro LFS, esperamos que lo hayas encontrado útil y que haya merecido la pena el tiempo invertido.

III. Parte III – Apéndices

Tabla de contenidos

A. Descripción de paquetes y dependencias
B. Recursos

Apéndice A. Descripción de paquetes y dependencias

Introducción

En este apéndice se describen los siguiente aspectos de cada paquete instalado en este libro.

- La localización oficial para la descarga del paquete.
- El contenido de cada paquete.
- Lo que cada programa de dicho paquete hace.
- Lo que cada paquete necesita para poder compilarlo.

Mucha de la información sobre estos paquetes (especialmente, su descripción) se ha extraído de las páginas del manual de esos paquetes. No vamos a mostrar las páginas del manual completas, sólo los elementos centrales que hagan posible entender lo que cada programa hace. Para conocer y entender todos los detalles sobre un programa aconsejamos comenzar leyendo la página del manual completa además de este apéndice.

Ciertos paquetes están documentados con mayor profundidad que otros, sencillamente porque sabemos más sobre unos que sobre otros. Si algo debería ser añadido a las siguientes descripciones, por favor no dudes en comunicarlo en las listas de correo. Intentamos que la lista contenga una descripción detallada de cada paquete, pero no podemos hacerlo sin ayuda.

Ten en cuenta que actualmente sólo está descrito lo que hace un paquete , y no lo que necesita que esté instalado. Esto se añadirá más adelante.

También están listadas todas las dependencias para la instalación de todos los paquetes instalados en el libro. La lista incluye qué programas de qué paquetes son necesarios para compilar correctamente el paquete a instalar.

Estas no son las dependencias necesarias para su ejecución, por lo tanto no te ayudarán para saber qué programas son necesarios para usar los programas del paquete. Son solamente las dependencias necesarias para compilarlo.

La lista de dependencias puede estar en ocasiones anticuada con respecto a la versión del paquete usada actualmente. Comprobar las dependencias es un trabajo pesado por lo que puede haber un desfase en la actualización de los paquetes. Pero, normalmente, en la actualización de versiones menores del paquete, las dependencias de instalación no cambian, por lo que son actuales en muchos casos. Si actualizamos a una versión mayor, con seguridad haremos un nuevo chequeo de las dependencias al mismo tiempo.

Autoconf

Localización oficial para descarga

Autoconf (2.53):
ftp://ftp.gnu.org/gnu/autoconf/

Contenido de autoconf–2.52

Programas

autoconf, autoheader, autoreconf, autoscan, autoupdate y ifnames

Descripciones

autoconf

Autoconf es una herramienta para generar guiones del intérprete de comandos que automáticamente configuran paquetes de código fuente, adaptándolos a muchas clases de sistemas tipo UNIX. Los guiones de configuración creados por Autoconf son independientes de Autoconf cuando se ejecutan, por tanto sus usuarios no necesitan tenerlo instalado.

autoheader

El programa autoheader puede crear un plantilla de declaraciones #define de C, usada posteriormente por el guión configure.

autoreconf

Si hay que generar varios guiones de configuración con autoconf, el programa autoreconf puede ahorrar algo de trabajo. Ejecuta autoconf (y autoheader, cuando es necesario) repetidamente para recrear los guiones de configuración de autoconf y las plantillas de las cabeceras de configuración en el árbol de directorios actual.

autoscan

El programa autoscan ayuda en la creación de ficheros configure.in para los paquetes. Este programa analiza los ficheros fuente en el árbol de directorios que se le pasa como argumento, o a partir del actual si no se especifica ninguno. Busca en los ficheros fuente problemas comunes de portabilidad y crea un fichero configure.scan que es una versión preliminar del fichero configure.in para ese paquete.

autoupdate

El programa autoupdate actualiza las llamadas a los macros de autoconf en los ficheros configure.in cambiando los nombres antiguos por los actuales.

ifnames

ifnames ayuda en la creación de los ficheros configure.in. Escribe los identificadores que el paquete usa en las construcciones condicionales del preprocesador de C. Si un paquete está preparado para tener cierta portabilidad, este programa le ayuda a averiguar qué debe verificar el guión configure. Puede ayudar a fijar ciertas carencias en los configure.in generados por autoscan.

Dependencias

Para instalar Autoconf–2.52 se necesita lo siguiente:

bash: sh

diffutils: cmp
fileutils: chmod, install, ln, ls, mkdir, mv, rm
grep: fgrep, grep
m4: m4
make: make
gawk: gawk
sed: sed
sh−utils: echo, expr, hostname, sleep, uname
texinfo: install−info
textutils: cat, tr

Automake

Localización oficial para descarga

Automake (1.6):
ftp://ftp.gnu.org/gnu/automake/

Contenido de automake−1.5

Programas

aclocal y automake

Descripciones

aclocal

Automake incluye una serie de macros de Autoconf que pueden ser usadas en los paquetes; algunas de ellas son requeridas por Automake en ciertas situaciones. Estas macros deben estar definidas en el fichero aclocal.m4; de otra forma no serán vistas por autoconf.

El programa aclocal genera automáticamente los ficheros aclocal.m4 basados en el contenido de configure.in. Esto proporciona una forma conveniente de obtener las macros facilitadas por Automake sin tener que buscarlas. Por otra parte, el mecanismo de aclocal es extensible para poder ser usado por otros paquetes

automake

Para crear todos los Makefile.in de un paquete, ejecuta el programa automake en el directorio base, sin argumentos. Automake automáticamente encontrará cada fichero Makefile.am apropiado (tras escanear configure.in) y generará el correspondiente Makefile.in.

Dependencias

Para instalar Automake−1.5 se necesita lo siguiente:

bash: sh
diffutils: cmp
fileutils: chmod, install, ls, mkdir, mv, rm, rmdir

grep: fgrep, grep
make: make
perl: perl
sed: sed
sh−utils: echo, expr, hostname, sleep
texinfo: install−info
textutils: cat, tr

Bash

Localización oficial para descarga

Bash (2.05a):
ftp://ftp.gnu.org/gnu/bash/

Contenido de bash−2.05a

Programas

bash, sh (enlace a bash) y bashbug

Descripciones

bash

bash es la "Bourne−Again SHell", que es un completo intérprete de comandos usado ampliamente en sistemas Unix. Bash es un programa que lee de la entrada estándar, el teclado. Un usuario escribe algo y el programa evalúa lo que ha escrito y hace algo con ello, como lanzar un programa.

bashbug

bashbug es un guión que ayuda al usuario en la composición y envío de informes de errores relacionados con bash, en un formato estándar.

sh

sh es un enlace simbólico al programa bash. Cuando se invoca como sh, bash intenta imitar el comportamiento de las versiones antiguas de sh lo mejor posible, mientras que también cumple los estándares POSIX.

Dependencias

Para instalar Bash−2.05a se necesita lo siguiente:

bash: bash, sh
binutils: ar, as, ld, ranlib, size
diffutils: cmp
fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm
gcc: cc, cc1, collect2, cpp0, gcc

grep: egrep, grep
make: make
gawk: awk
sed: sed
sh−utils: basename, echo, expr, hostname, sleep, uname
texinfo: install−info
textutils: cat, tr, uniq

Bin86

Localización oficial para descarga

Bin86 (0.16.2):
http://www.cix.co.uk/~mayday/

Contenido de bin86−0.16.0

Programas

as86, as86_encap, ld86, nm86 (enlace a objdump86), objdump86 y size86 (enlace a objdump86)

Descripciones

as86

as86 es un ensamblador para los procesadores 8086...80386.

as86_encap

as86_encap es un guión del intérprete de comandos que llama a as86 y que convierte el binario creado en un fichero prog.v de C para ser incluido o enlazado con programas como instaladores de bloques de arranque.

ld86

ld86 entiende solamente los ficheros objeto creados por el ensamblador as86. Puede enlazarlos tanto en un ejecutable I&D impuro como en uno separado.

nm86

No hay descripción disponible.

objdump86

No hay descripción disponible.

size86

No hay descripción disponible.

Dependencias

Para instalar Bin86–0.16.0 se necesita lo siguiente:

bash: sh
binutils: as, ld, strip
fileutils: chmod, install, ln, mv
gcc: cc, cc1, collect2, cpp0
make: make
sed: sed

Binutils

Localización oficial para descarga

Binutils (2.12):
ftp://ftp.gnu.org/gnu/binutils/

Contenido de binutils–2.11.2

Programas

addr2line, ar, as, c++filt, gasp, gprof, ld, nm, objcopy, objdump, ranlib, readelf, size, strings y strip

Descripciones

addr2line

addr2line traslada direcciones de programas a nombres de ficheros y números de líneas. Dándole una dirección y un ejecutable, usa la información de depuración del ejecutable para averiguar qué fichero y número de línea está asociado con dicha dirección.

ar

El programa ar crea, modifica y extrae desde archivos. Un archivo es un fichero que almacena una colección de otros ficheros en una estructura que hace posible obtener el original de cada fichero individual (llamados miembros del archivo).

as

as está pensado, principalmente, para ensamblar la salida del compilador GNU gcc para ser usada por el enlazador ld.

c++filt

El lenguaje C++ proporciona sobrecarga de funciones, lo que significa que es posible escribir varias funciones con el mismo nombre (suponiendo que cada una tome parámetros de diferente tipo). Todos los nombres de funciones C++ son codificadas dentro de una etiqueta de bajo nivel del ensamblador (este proceso es conocido como "mangling"). El programa c++flit hace lo contrario: decodifica (demangling) nombres de bajo nivel en

nombres de nivel de usuario para que el enlazador pueda evitar conflictos en estas funciones sobrecargadas.

gasp

Gasp es el preprocesador de las macros del ensamblador.

gprof

gprof muestra el grafo de llamadas de los datos perfilados.

ld

ld combina un número de objetos y ficheros de archivo, reubica sus datos y establece las referencias a los símbolos. Frecuentemente, el último paso de la compilación de un nuevo programa es hacer una llamada a ld.

nm

nm lista los símbolos de los ficheros objeto.

objcopy

La utilidad objcopy copia el contenido de un fichero objeto en otro. objcopy usa la librería BFD de GNU para leer y escribir los ficheros objeto. Puede escribir el fichero objeto destino en un formato diferente al del fichero objeto fuente .

objdump

objdump muestra información sobre uno o más ficheros objeto. Mediante opciones se puede indicar la información a mostrar. Esta información es útil fundamentalmente para los programadores que trabajan en herramientas de compilación (al contrario de los programadores que sólo quieren que sus programas compilen y funcionen).

ranlib

ranlib genera un índice de los contenidos de un archivo, y lo coloca en el archivo. El índice lista cada símbolo definido por un miembro de un archivo que es un fichero objeto reubicable.

readelf

readelf muestra información sobre binarios de tipo elf.

size

size lista los tamaños de las secciones –y el tamaño total– para cada uno de los ficheros objeto en su lista de argumentos. Por defecto, se genera una línea de salida por cada fichero objeto o cada módulo de un archivo.

strings

Para cada fichero dado, strings muestra las cadenas de caracteres imprimibles de al menos 4 caracteres (o el número especificado en las opciones del programa) seguidas por un carácter no imprimible. Por defecto, sólo muestra las cadenas procedentes de las secciones de inicialización y carga de los ficheros objeto; para otros tipos de ficheros muestra todas las cadenas de los mismos.

strings es útil, principalmente, para determinar el contenido de ficheros que no son de texto.

strip

strip elimina todos los símbolos o sólo los especificados de los ficheros objeto. La lista de ficheros objeto puede incluir archivos. Se debe indicar, por lo menos, un fichero objeto. strip modifica los ficheros mencionados en sus argumentos, en vez de escribir copias modificadas con otro nombre.

Librerías

libbfd.a, libiberty.a y libopcodes.a

Descripciones

libbfd

libbfd es la librería de descriptores de ficheros binarios (BFD).

libiberty

libiberty es una colección de subrutinas usadas por varios programas GNU, incluidos getopt, obstack, strerror, strtol y strtoul.

libopcodes

libopcodes es una librería nativa para manejar mnemónicos y se usa durante la construcción de utilidades como objdump. Los mnemónicos son, en realidad, las versiones en texto legible de las instrucciones del procesador.

Dependencias

Para instalar Binutils−2.11.2 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: ar, as, ld, nm, ranlib, strip
diffutils: cmp
fileutils: chmod, cp, ln, ls, mkdir, mv, rm, rmdir, touch
flex: flex
gcc: cc, cc1, collect2, cpp0, gcc
glibc: ldconfig
grep: egrep, fgrep, grep
m4: m4
make: make
gawk: gawk
sed: sed
sh−utils: basename, echo, expr, hostname, sleep, true, uname
texinfo: install−info, makeinfo
textutils: cat, sort, tr, uniq

Bison

Localización oficial para descarga

Bison (1.34):
ftp://ftp.gnu.org/gnu/bison/

Contenido de bison−1.31

Programas

bison y yacc

Descripciones

bison

Bison es un generador de analizadores sintácticos, un sustituto de YACC ("Yet Another Compiler Compiler", otro compilador de compiladores). Entonces, ¿qué es Bison? Es un programa que genera un programa que analiza la estructura de un fichero de texto. En lugar de escribir el programa, el usuario especifica qué cosas deben ser conectadas y con esas reglas se genera un programa que analiza el fichero de texto. Hay muchos ejemplos en los que se necesita esta estructura y uno de ellos es la calculadora.

Tomando la cadena :

```
1 + 2 * 3
```

Una persona puede fácilmente saber que el resultado es 7. ¿Por qué? Porque al ver la estructura nuestro cerebro sabe como interpretar la cadena. La computadora no sabe eso y Bison es una herramienta que le ayuda a interpretarla presentando la cadena de la siguiente forma al compilador:

```
  +
 / \
 *  1
/ \
2  3
```

Comenzando por la base del árbol y subiendo por los números 2 y 3, que están unidos por el símbolo de la multiplicación, la computadora multiplica 2 y 3. Almacena el resultado de la multiplicación y, lo siguiente que ve, es el resultado de 2*3 y el número 1 unido con el símbolo de la suma. Añadiendo 1 al resultado previo se obtiene 7. El cálculo de formulas más complejas puede hacerse pasándolas a este formato de árbol. El ordenador comienza justo por la base y sigue trabajando hacia arriba hasta alcanzar el resultado correcto. Por supuesto, Bison no se usa sólo en calculadoras.

yacc

Nosotros creamos un guión llamado "yacc" que invoca a bison usando la opción −y. Esto es para la compatibilidad con programas que usan yacc en lugar de bison.

Dependencias

Para instalar Bison−1.31 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm, rmdir
gcc: cc, cc1, collect2, cpp0, gcc
grep: egrep, fgrep, grep
make: make
sed: sed
sh−utils: basename, dirname, echo, expr, hostname, sleep, uname
texinfo: install−info
textutils: cat, head, tr, uniq

Bzip2

Localización oficial para descarga

Bzip2 (1.0.2):
ftp://sourceware.cygnus.com/pub/bzip2/

Contenido de bzip2−1.0.1

Programas

bunzip2 (enlace a bzip2), bzcat (enlace a bzip2), bzip2 y bzip2recover

Descripciones

bunzip2

Bunzip2 descomprime ficheros que han sido comprimidos con bzip2.

bzcat

bzcat (o bzip2 −dc) descomprime todos los ficheros especificados hacia la salida estándar.

bzip2

bzip2 comprime ficheros usando el algoritmo de compresión de texto por ordenación de bloques Burrows−Wheeler y la codificación Huffman. La compresión es, en general, considerablemente superior a la obtenida por otros compresores más convencionales basados en el LZ77/LZ78 y se acerca al rendimiento de la familia de compresores estadísticos PPM.

bzip2recover

bzip2recover recupera datos de ficheros bzip2 dañados.

Librerías

libbz2.[a,so]

libbz2

libbz2 es la librería que implementa la compresión sin pérdidas por ordenación de bloques usando el algoritmo de Burrows−Wheeler.

Dependencias

Para instalar Bzip2−1.0.1 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib
fileutils: cp, ln, rm
gcc: cc1, collect2, cpp0, gcc
make: make

Diffutils

Localización oficial para descarga

Diff Utils (2.8):
ftp://ftp.gnu.org/gnu/diffutils/

Contenido de diffutils−2.7

Programas

cmp, diff, diff3 y sdiff

Descripciones

cmp y diff

Tanto cmp como diff comparan dos ficheros y muestran sus diferencias. Ambos programas tienen argumentos para comparar ficheros en diferentes situaciones.

diff3

La diferencia entre diff y diff3 es que diff compara 2 ficheros mientras diff3 compara 3.

sdiff

sdiff mezcla dos ficheros y muestra los resultados interactivamente.

Dependencias

Para instalar Diffutils−2.7 se necesita lo siguiente:

bash: sh
binutils: ld, as
diffutils: cmp
fileutils: chmod, cp, install, mv, rm
gcc: cc1, collect2, cpp0, gcc
grep: egrep, grep
make: make
sed: sed
sh−utils: date, hostname
textutils: cat, tr

E2fsprogs

Localización oficial para descarga

E2fsprogs (1.27):
ftp://download.sourceforge.net/pub/sourceforge/e2fsprogs/
http://download.sourceforge.net/e2fsprogs/

Contenido de e2fsprogs−1.25

Programas

badblocks, chattr, compile_et, debugfs, dumpe2fs, e2fsck, e2image, e2label, fsck, fsck.ext2, fsck.ext3, lsattr, mk_cmds, mke2fs, mkfs.ext2, mklost+found, resize2fs, tune2fs y uuidgen

Descripciones

badblocks

badblocks se usa para buscar bloques dañados en un dispositivo (normalmente una partición de disco).

chattr

chattr cambia los atributos de un fichero en un sistema de ficheros ext2 de Linux.

compile_et

compile_et es usado para convertir una tabla con códigos de error y sus mensajes asociados en un fichero fuente C apropiado para usar con la librería com_err.

debugfs

El programa debugfs es un depurador de sistemas de ficheros. Puede usarse para examinar y cambiar el estado de un sistema de ficheros ext2.

dumpe2fs

dumpe2fs muestra la información del superbloque y de los grupos de bloques del sistema de ficheros presente en un dispositivo específico.

e2fsck and fsck.ext2

e2fsck se usa para chequear y reparar sistemas de ficheros ext2. fsck.ext2 hace lo mismo que e2fsck.

e2image

e2image se usa para salvar información crítica de un sistema de ficheros ext2 en un fichero.

e2label

e2label muestra o cambia la etiqueta de un sistema de ficheros ext2 situado en el dispositivo especificado.

fsck

fsck se usa para chequear y, opcionalmente, reparar un sistema de ficheros.

fsck.ext3

fsck.ext3 se usa para chequear y, opcionalmente, reparar un sistema de ficheros ext3.

lsattr

lsattr muestra los atributos de un fichero en un sistema de ficheros ext2.

mk_cmds

No hay descripción disponible.

mke2fs and mkfs.ext2

mke2fs se usa para crear sistemas de ficheros ext2 en un dispositivo (normalmente una partición de disco). mkfs.ext2 hace lo mismo que mke2fs.

mklost+found

mklost+found se usa para crear un directorio lost+found en el directorio de trabajo actual de un sistema de ficheros ext2. mklost+found reserva una serie de bloques de disco en el directorio para que sean usados por e2fsck.

resize2fs

resize2fs se usa para redimensionar sistemas de ficheros ext2.

tune2fs

tune2fs ajusta los parámetros de un sistema de ficheros ext2.

uuidgen

El programa uuidgen crea un nuevo identificador universal único (UUID) usando la librería libuuid. El nuevo UUID puede considerarse razonablemente único por muchos UUID que se hayan creado en el sistema local o en otros sistemas en el pasado o en el futuro.

Librerías

libcom_err.[a,so], libe2p.[a,so], libext2fs.[a,so], libss.[a,so] y libuuid.[a,so]

Descripciones

libcom_err

No hay descripción disponible.

libe2p

No hay descripción disponible.

libext2fs

No hay descripción disponible.

libss

No hay descripción disponible.

libuuid

No hay descripción disponible.

Dependencias

Para instalar E2fsprogs−1.25 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib, strip
diffutils: cmp
fileutils: chmod, cp, install, ln, mkdir, mv, rm, sync
gcc: cc, cc1, collect2, cpp0
glibc: ldconfig
grep: egrep, grep
gzip: gzip
make: make
gawk: awk
sed: sed

sh−utils: basename, echo, expr, hostname, uname
texinfo: makeinfo
textutils: cat, tr

Ed

Localización oficial para descarga

Ed (0.2):
ftp://ftp.gnu.org/gnu/ed/

Contenido de ed−0.2

Programas

ed y red (enlace a ed)

Descripciones

ed

Ed es un editor de líneas de texto. Se usa para crear, mostrar, modificar o cualquier otra manipulación de ficheros de texto.

red

red es un ed restringido: sólo puede editar ficheros del directorio actual y no puede ejecutar comandos del intérprete de comandos.

Dependencias

Para instalar Ed−0.2 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, cp, install, ln, mv, rm, touch
gcc: cc1, collect2, cpp0, gcc
grep: egrep, grep
make: make
sed: sed
sh−utils: hostname
textutils: cat, tr

File

Localización oficial para descarga

File (3.37):
ftp://ftp.gw.com/mirrors/pub/unix/file/

Contenido de file–3.37

Programas

file

Descripciones

file

File comprueba cada fichero especificado para clasificarlo. Se hacen tres tipos de pruebas, en este orden: pruebas de sistemas de ficheros, pruebas de números mágicos y pruebas de lenguajes. La primera prueba que tenga éxito hace que se muestre el tipo de fichero.

Dependencias

Para instalar File–3.37 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: as, ld
diffutils: cmp
fileutils: chmod, install, ln, ls, mv, rm, touch
gcc: cc1, collect2, cpp0, gcc
grep: egrep, grep
m4: m4
make: make
gawk: gawk
sed: sed
sh–utils: echo, expr, hostname, sleep
texinfo: makeinfo
textutils: cat, tr

Fileutils

Localización oficial para descarga

File Utils (4.1):
ftp://ftp.gnu.org/gnu/fileutils/

Contenido de fileutils–4.1

Programas

chgrp, chmod, chown, cp, dd, df, dir, dircolors, du, install, ln, ls, mkdir, mkfifo, mknod, mv, rm, rmdir, shred, sync, touch y vdir

Descripciones

chgrp

chgrp cambia el grupo de cada fichero al grupo especificado, que puede ser tanto el nombre de un grupo como su identificador numérico.

chmod

chmod cambia los permisos de un fichero de acuerdo con el modo, que puede ser tanto una representación simbólica de los cambios a hacer, o un número octal que representa el patrón de bits de los nuevos permisos.

chown

chown cambia el usuario y/o el grupo al que pertenece un fichero.

cp

cp copia ficheros de un lugar a otro.

dd

dd copia un fichero (por defecto, de la entrada estándar a la salida estándar) con un tamaño de bloque definido por el usuario, mientras, opcionalmente, realiza conversiones en él.

df

df muestra la cantidad de espacio disponible en los sistemas de ficheros a los que pertenece cada fichero que se le pasa como argumento. Si no se indica ningún fichero, se muestra el espacio disponible en todos los sistemas de ficheros montados actualmente

dir, ls y vdir

dir y vdir son versiones de ls con formatos de salida diferentes. Estos programas listan cada fichero o directorio especificado. El contenido de los directorios se lista alfabéticamente. Para ls, los ficheros se listan en columnas ordenados verticalmente si la salida estándar es un terminal; en otro caso se listan uno por línea. Para dir, los ficheros se listan por defecto en columnas ordenados verticalmente. Para vdir, los ficheros se listan, por defecto, en formato largo.

dircolors

dircolors imprime comandos para modificar la variable de entorno LS_COLOR. La variable LS_COLOR se usa para cambiar el esquema de color por defecto de ls y de herramientas relacionadas.

du

du muestra la cantidad de espacio en disco usado por cada fichero o directorio pasado como argumento.

install

install copia ficheros y establece sus permisos y, si es posible, su propietario y grupo.

ln

ln crea enlaces duros o blandos (simbólicos) entre ficheros.

mkdir

mkdir crea directorios con el nombre indicado.

mkfifo

mkfifo crea una tubería (FIFO) con un nombre dado.

mknod

mknod crea una tubería (FIFO), un fichero especial de caracteres o un fichero especial de bloques con el nombre indicado.

mv

mv mueve ficheros de un directorio a otro o renombra ficheros, dependiendo de los argumentos que se le pasen.

rm

rm elimina ficheros o directorios.

rmdir

rmdir elimina directorios, si están vacíos.

shred

shred borra un fichero de forma segura, sobreescribiéndolo para que su contenido no pueda ser recuperado.

sync

sync guarda los bloques modificados en disco y actualiza el superbloque.

touch

touch cambia las fechas de modificación o acceso del fichero especificado, poniéndole la fecha actual. Si el fichero no existe crea uno vacío.

Dependencias

Para instalar Fileutils−4.1 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm, rmdir
gettext: msgfmt, xgettext
gcc: cc, cc1, collect2, cpp0, gcc
grep: egrep, fgrep, grep
make: make
perl: perl
sed: sed
sh−utils: basename, echo, expr, hostname, sleep, uname
texinfo: install−info
textutils: cat, tr

Findutils

Localización oficial para descarga

Find Utils (4.1):
ftp://ftp.gnu.org/gnu/findutils/

Find Utils Patch (4.1):
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/
http://ftp.linuxfromscratch.org/lfs−packages/3.3/

Contenido de findutils−4.1

Programas

bigram, code, find, frcode, locate, updatedb y xargs

Descripciones

bigram

bigram se usa junto con code para generar las bases de datos de locate en el formato antiguo. Para saber más de estos tres programas, lea la página de manual locatedb.5.

code

code es el antecesor de frcode. Se usaba en las bases de datos de formato antiguo.

find

El programa find busca los ficheros de una jerarquía de directorios que cumplan un cierto criterio. Si no se especifica un criterio de búsqueda, lista todos los ficheros del directorio actual y de los subdirectorios.

frcode

updatedb lanza el programa frcode para comprimir la lista de ficheros usando "front−compression", que reduce el tamaño de la base de datos en un factor de 4 o 5.

locate

Locate busca en una base de datos que contiene todos los ficheros y directorios de un sistema de ficheros. Este programa lista los ficheros y directorios de la base de datos que cumplan cierto criterio. Si un usuario busca un fichero, este programa buscará en la base de datos y le dirá dónde están ubicados exactamente esos ficheros. Esto sólo es válido si la base de datos de locate se encuentra actualizada. En otro caso mostrará información anticuada.

updatedb

El programa updatedb actualiza la base de datos de locate. Explora por completo el sistema de ficheros (incluidos otros sistemas de ficheros que se encuentren montados a no ser que se le indique lo contrario) e inserta todos los directorios y ficheros que encuentre en la base de datos que usa locate para recuperar dicha información. Es una buena costumbre actualizar la base de datos una vez al día para obtener información correcta cuando se necesite.

xargs

El comando xargs aplica un comando a una lista de ficheros. Si se necesita aplicar el mismo comando sobre múltiples ficheros, puede crearse un fichero que contenga la ruta a todos estos ficheros (uno por línea) y usar xargs para aplicar el comando a la lista.

Dependencias

Para instalar Findutils−4.1 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, cp, install, mv, rm
grep: egrep, grep
gcc: cc1, collect2, cpp0, gcc
make: make
patch: patch
sed: sed
sh−utils: basename, date, echo, hostname
textutils: cat, tr

Flex

Localización oficial para descarga

Flex (2.5.4a):
ftp://ftp.gnu.org/non−gnu/flex/

Contenido de flex−2.5.4a

Programas

flex, flex++ (enlace a flex) y lex

Descripciones

flex

flex es una herramienta para generar programas capaces de reconocer patrones de texto. El reconocimiento de patrones es muy útil en muchas aplicaciones. El usuario establece las reglas a usar y flex generará el programa que buscará esos patrones. La razón por la que la gente usa flex es porque es mucho más fácil establecer las reglas a usar que escribir directamente un programa que busque el texto.

flex++

flex++ invoca una versión de flex usada exclusivamente para generar analizadores en C++.

lex

Creamos un guión "lex" que invoca a flex con la opción −l. Esto es por compatibilidad con programas que usan lex en lugar de flex

Librerías

libfl.a

Descripciones

libfl

No hay descripción disponible.

Dependencias

Para instalar Flex−2.5.4a se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib
bison: bison
diffutils: cmp

fileutils: chmod, cp, install, ln, mv, rm, touch
gcc: cc1, collect2, cpp0, gcc
grep: egrep, grep
make: make
sed: sed
sh−utils: echo, hostname
textutils: cat, tr

Gawk

Localización oficial para descarga

Gawk (3.1.0):
ftp://ftp.gnu.org/pub/gnu/gawk/

Contenido de gawk−3.1.0

Aún no se ha comprobado

Dependencias

Para instalar Gawk−3.1.0 se necesita lo siguiente:

Dependencias no comprobadas todavía

GCC

Localización oficial para descarga

GCC (2.95.3):
ftp://ftp.gnu.org/pub/gnu/gcc/

GCC Patch (2.95.3−2):
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/
http://ftp.linuxfromscratch.org/lfs−packages/3.3/

Contenido de gcc−2.95.3

Programas

c++, c++filt, cc (enlace a gcc), cc1, cc1plus, collect2, cpp, cpp0, g++, gcc, gcov, protoize y unprotoize

Descripciones

cc, cc1, cc1plus, gcc

Estos programas forman el compilador de C. Un compilador convierte el código fuente en formato de texto a un formato que un ordenador pueda entender. Después de que un fichero de código fuente es compilado en un fichero objeto, un enlazador creará un fichero ejecutable a partir de uno o más de estos ficheros objeto generados por el compilador.

c++, cc1plus, g++

Estos programas forman el compilador de C++; El equivalente de cc, gcc, etc.

c++filt

c++filt se usa para la decodificación (demangle) de los símbolos de C++.

collect2

No hay descripción disponible.

cpp, cpp0

cpp preprocesa los ficheros fuente, como en la inclusión del contenido de los ficheros de cabecera en los ficheros fuente. Es buena idea no hacer esto manualmente para ahorrar un montón de tiempo. Basta con insertar una línea del tipo #include <fichero>. El preprocesador inserta el contenido de ese fichero en el fichero fuente. Esta es una de las cosas que el preprocesador hace.

gcov

No hay descripción disponible.

protoize

Programa adicional que convierte funciones o definiciones del antiguo estilo anterior al ANSI al nuevo estilo de prototipos ANSI C. (un fichero por defecto para saber como va esto es `/usr/lib/gcc-lib/<arch>/<versión>/SYSCALLS.c.X`)

unprotoize

Programa adicional que convierte prototipos hechos con protoize a su estilo antiguo original (solo funciona si antes fueron generados con protoize).

Librerías

libgcc.a, libiberty.a, libstdc++.[a,so]

libgcc

libgcc.a es un fichero de soporte en tiempo de ejecución para gcc. Con frecuencia, en la mayoría máquinas, libgcc.a no es realmente necesaria.

libiberty

libiberty es una colección de subrutinas usadas por muchos programas GNU, incluidos getopt, obstack, strerror, strtol y strtoul.

libstdc++

libstdc++ es la librería C++. Es utilizada por programas escritos en C++ y contiene funciones que son usadas frecuentemente por esos programas. De esta forma el programador no necesita escribir ciertas funciones (como la escritura de una cadena de texto en pantalla) desde el principio cada vez que crea un programa.

Dependencias

Para instalar GCC−2.95.3 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, nm, ranlib
diffutils: cmp
fileutils: chmod, cp, ln, ls, mkdir, mv, rm, touch
find: find
gcc: cc, cc1, collect2, cpp0, gcc
grep: egrep, grep
make: make
patch: patch
sed: sed
sh−utils: basename, dirname, echo, expr, hostname, sleep, true, uname
tar: tar
texinfo: install−info, makeinfo
textutils: cat, tail, tr

Gettext

Localización oficial para descarga

Gettext (0.11.1):
ftp://ftp.gnu.org/gnu/gettext/

Contenido de gettext−0.10.40

Programas

gettext, gettextize, msgcmp, msgcomm, msgfmt, msgmerge, msgunfmt, ngettext y xgettext

Descripciones

gettext

El paquete gettext se usa en la internacionalización (también conocida como i18n) y en la localización (conocida como l10n). Los programas pueden ser compilados con soporte para lenguaje nativo (NLS) que

activa el que se muestren los mensajes de salida en el idioma del usuario en vez de en el idioma por defecto, el inglés.

gettextize

El programa gettextize copia todos los ficheros estándar gettext en un directorio. Se usa para hacer paquetes con traducción gettext.

msgcmp

El programa msgcmp compara dos ficheros de traducción directa.

msgcomm

El programa msgcomm busca los mensajes que aparecen en varios ficheros .po. Se usa para comparar cómo deben traducirse las cosas.

msgfmt

El programa msgfmt compila traducciones directas en código máquina. Se usa para crear el fichero de traducción final de un programa/paquete.

msgmerge

El programa msgmerge combina dos traducciones directas en un fichero. Se usa para actualizar la traducción directa con el extracto de las fuentes.

msgunfmt

El programa msgunfmt descompila ficheros de traducciones en traducciones directas de texto. Sólo puede ser usado si la versión compilada está disponible.

ngettext

El programa ngettext muestra traducciones en lenguaje nativo de un mensaje de texto cuya forma gramatical depende de un número.

xgettext

El programa xgettext extrae las líneas de mensajes de los ficheros C del programador. Se usa para hacer la primera plantilla de traducción.

Dependencias

Para instalar Gettext−0.10.40 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: ar, as, ld, nm, ranlib, strip
bison: bison

diffutils: cmp
fileutils: chmod, install, ln, ls, mkdir, mv, rm, rmdir
gcc: cc, cc1, collect2, cpp0, gcc
grep: egrep, fgrep, grep
m4: m4
make: make
gawk: gawk
sed: sed
sh−utils: basename, echo, expr, hostname, sleep, uname
texinfo: install−info, makeinfo
textutils: cat, sort, tr, uniq

Glibc

Localización oficial para descarga

Glibc (2.2.5):
ftp://ftp.gnu.org/gnu/glibc/

Glibc−linuxthreads (2.2.5):
ftp://ftp.gnu.org/gnu/glibc/

Contenido de glibc−2.2.5

Programas

catchsegv, gencat, getconf, getent, glibcbug, iconv, iconvconfig, ldconfig, ldd, lddlibc4, locale, localedef, mtrace, nscd, nscd_nischeck, pcprofiledump, pt_chown, rpcgen, rpcinfo, sln, sprof, tzselect, xtrace, zdump y zic

Descripciones

catchsegv

No hay descripción disponible.

gencat

gencat genera catálogos de mensajes.

getconf

No hay descripción disponible.

getent

getent obtiene entradas de una base de datos administrativa.

glibcbug

glibcbug crea un informe de fallos sobre glibc y lo envía a la dirección de correo electrónico de errores.

iconv

iconv realiza conversiones de los juegos de caracteres.

iconvconfig

iconvconfig crea un fichero de configuración para la carga rápida del módulo iconv.

ldconfig

ldconfig configura las asociaciones en tiempo de ejecución para el enlazador dinámico.

ldd

ldd muestra las librerías compartidas requeridas por cada programa o librería especificada en la línea de comandos.

lddlibc4

No hay descripción disponible.

locale

No hay descripción disponible.

localedef

localedef compila las especificaciones para locale.

mtrace

No hay descripción disponible.

nscd

nscd es un demonio que suministra una caché para las peticiones más comunes al servidor de nombres.

nscd_nischeck

No hay descripción disponible.

pcprofiledump

pcprofiledump vuelca la información generada por un perfilador.

pt_chown

pt_chown establece el propietario, grupo y permisos de acceso del pseudo−terminal esclavo correspondiente al pseudo−terminal maestro apuntado por el descriptor de ficheros "3". Este es el programa de ayuda para la

función `grantpt'. No está pensado para ejecutarse directamente desde la línea de comandos.

rpcgen

No hay descripción disponible.

rpcinfo

No hay descripción disponible.

sln

sln enlaza simbólicamente un destino a una fuente. Está enlazado estáticamente, no necesitando enlazado dinámico. Por tanto, sln es útil para crear enlaces simbólicos a librerías dinámicas si, por alguna razón, el enlazador dinámico del sistema no funciona.

sprof

sprof lee y muestra datos de perfilado de objetos compartidos.

tzselect

tzselect pregunta al usuario información sobre la localización actual y muestra la descripción de la zona horaria resultante en la salida estándar.

xtrace

xtrace traza la ejecución de un programa mostrando la función actualmente ejecutada.

zdump

zdump es el visualizador de información de huso horario.

zic

zic es el compilador de la zona horaria.

Librerías

ld.so, libBrokenLocale.[a,so], libBrokenLocale_p.a, libSegFault.so, libanl.[a,so], libanl_p.a, libbsd−compat.a, libc.[a,so], libc_nonshared.a, libc_p.a, libcrypt.[a,so], libcrypt_p.a, libdl.[a,so], libdl_p.a, libg.a, libieee.a, libm.[a,so], libm_p.a, libmcheck.a, libmemusage.so, libnsl.a, libnsl_p.a, libnss_compat.so, libnss_dns.so, libnss_files.so, libnss_hesiod.so, libnss_nis.so, libnss_nisplus.so, libpcprofile.so, libpthread.[a,so], libpthread_p.a, libresolv.[a,so], libresolv_p.a, librpcsvc.a, librpcsvc_p.a, librt.[a,so], librt_p.a, libthread_db.so, libutil.[a,so] y libutil_p.a

Descripciones

ld.so

ld.so es el programa de ayuda para las librerías compartidas ejecutables.

libBrokenLocale, libBrokenLocale_p

No hay descripción disponible.

libSegFault

No hay descripción disponible.

libanl, libanl_p

No hay descripción disponible.

libbsd–compat

No hay descripción disponible.

libc, libc_nonshared, libc_p

Estos ficheros constituyen la librería C principal. La librería C es una colección de funciones usadas frecuentemente en programas. De esta forma un programador no necesita crear sus propias funciones para cada tarea individual. Las cosas más comunes, como mostrar una cadena en pantalla, están presentes y a disposición del programador.

La librería C (en realidad, casi todas las librerías) viene en dos formas: dinámica y estática. En resumen, cuando un programa usa una librería C estática, se copia el código de la librería C dentro del ejecutable. Cuando un programa usa una librería dinámica, el ejecutable no contiene el código de la librería pero, en su lugar, tiene una rutina que carga las funciones desde esa librería en el momento en el que se ejecuta. De esta forma disminuye de forma significativa el tamaño del programa. La documentación que acompaña a la librería C describe esto con más detalle, pues es demasiado complicado explicarlo aquí en dos o tres líneas.

libcrypt, libcrypt_p

libcrypt es la librería criptográfica.

libdl, libdl_p

No hay descripción disponible.

libg

No hay descripción disponible.

libieee

No hay descripción disponible.

libm, libm_p

libm es la librería matemática.

libmcheck

No hay descripción disponible.

libmemusage

No hay descripción disponible.

libnsl, libnsl_p

No hay descripción disponible.

libnss_compat, libnss_dns, libnss_files, libnss_hesiod, libnss_nis, libnss_nisplus

No hay descripción disponible.

libpcprofile

No hay descripción disponible.

libpthread, libpthread_p

No hay descripción disponible.

libresolv, libresolv_p

No hay descripción disponible.

librpcsvc, librpcsvc_p

No hay descripción disponible.

librt, librt_p

No hay descripción disponible.

libthread_db

No hay descripción disponible.

libutil, libutil

No hay descripción disponible.

Dependencias

Para instalar Glibc−2.2.5 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib, readelf
diffutils: cmp

fileutils: chmod, cp, install, ln, mknod, mv, mkdir, rm, touch
gcc: cc, cc1, collect2, cpp, gcc
grep: egrep, grep
gzip: gzip
make: make
gawk: gawk
sed: sed
sh−utils: date, expr, hostname, pwd, uname
texinfo: install−info, makeinfo
textutils: cat, cut, sort, tr

Grep

Localización oficial para descarga

Grep (2.5):
ftp://ftp.gnu.org/gnu/grep/

Contenido de grep−2.4.2

Programas

egrep, fgrep y grep

Descripciones

egrep

egrep muestra las líneas de un fichero que coincidan con una determinada expresión regular extendida.

fgrep

fgrep muestra las líneas de un fichero que coincidan con una lista de cadenas fijas, separadas por saltos de línea, cualquiera de las cuales puede ser coincidente.

grep

grep muestra las líneas de un fichero que coincidan con una expresión regular.

Dependencias

Para instalar Grep−2.4.2 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: as, ld
diffutils: cmp
fileutils: chmod, install, ls, mkdir, mv, rm

gettext: msgfmt, xgettext
gcc: cc, cc1, collect2, cpp0, gcc
glibc: getconf
grep: egrep, fgrep, grep
m4: m4
make: make
gawk: gawk
sed: sed
sh−utils: basename, echo, expr, hostname, sleep, uname
texinfo: install−info, makeinfo
textutils: cat, tr

Groff

Localización oficial para descarga

Groff (1.17.2):
ftp://ftp.gnu.org/gnu/groff/

Contenido de groff−1.17.2

Programas

addftinfo, afmtodit, eqn, grn, grodvi, groff, grog, grolbp, grolj4, grops, grotty, hpftodit, indxbib, lkbib, lookbib, mmroff, neqn, nroff, pfbtops, pic, post−grohtml, pre−grohtml, refer, soelim, tbl, tfmtodit y troff

Descripciones

addftinfo

addftinfo lee un fichero de fuentes troff y añade alguna información adicional sobre la métrica de la fuente, que es usada por el sistema groff.

afmtodit

afmtodit crea un fichero de fuentes para usarlo con groff y grops.

eqn

eqn compila las descripciones de las formulas embebidas en los ficheros de entrada de troff a comandos que pueda entender troff.

grn

grn es un preprocesador groff para ficheros gremlin.

grodvi

grodvi es un controlador para groff que genera formatos dvi de TeX.

groff

groff es una interfaz para el sistema de formateado de documentos groff. Normalmente lanza el programa troff y un post−procesador apropiado para el dispositivo seleccionado.

grog

grog lee ficheros y supone cual de las opciones −e, −man, −me, −mm, −ms, −p, −s, y −t de groff se necesitan para mostrar los ficheros, y muestra el comando de groff incluyendo esas opciones en la salida estándar.

grolbp

grolbp es un controlador de groff para las impresoras Canon CAPSL (series LBP−4 y LBP−8 de impresoras láser)

grolj4

grolj4 es un controlador para groff que produce salidas en el formato PCL5 adecuado para impresoras HP Laserjet 4.

grops

grops transforma la salida de GNU troff en Postscript.

grotty

grotty transforma la salida de GNU troff en un formato adecuado para dispositivos tipo máquina de escribir.

hpftodit

hpftodit crea un fichero de fuentes para usar con groff −Tlj4 a partir de ficheros de marcas de fuentes métricas de HP.

indxbib

indxbib hace un índice inverso para la base de datos bibliográfica, un fichero específico para usarlo con refer, lookbib, y lkbib.

lkbib

lkbib busca, en las bases de datos bibliográficas, referencias que contengan las claves especificadas y muestra cualquier referencia encontrada en la salida estándar.

lookbib

lookbib muestra un aviso en la salida de error estándar (excepto si la entrada estándar no es un terminal), lee de la entrada estándar una línea conteniendo un grupo de palabras clave, busca en las bases de datos bibliográficos en un fichero especificado las referencias que contengan dichas claves, muestra cualquier referencia encontrada en la salida estándar y repite el proceso hasta el final de la entrada.

mmroff

mmroff es un preprocesador simple para groff.

neqn

El guión neqn formatea ecuaciones para salida ASCII.

nroff

El guión nroff emula al comando UNIX nroff usando groff.

pfbtops

pfbtops transforma una fuente en formato .pfb de Postscript a ASCII.

pic

pic compila descripciones de gráficos embebidos dentro de ficheros de entrada de troff o TeX a comandos que puedan ser entendidos por TeX o troff.

pre−grohtml y post−grohtml

pre− y post−grohtml transforman la salida de GNU troff a html.

refer

refer copia el contenido de un fichero en la salida estándar, excepto que las líneas entre .[y .] son interpretadas como citas, y las líneas entre .R1 y .R2 son interpretadas como comandos sobre cómo deben ser procesadas las citas.

soelim

soelim lee ficheros y reemplaza líneas de la forma *.so fichero* por el contenido de *fichero*.

tbl

tbl compila descripciones de tablas embebidas dentro de ficheros de entrada troff a comandos que puedan ser entendidos por troff.

tfmtodit

tfmtodit crea un fichero de fuentes para su uso con **`groff -Tdvi`**

troff

troff es altamente compatible con Unix troff. Normalmente debe ser invocado usando el comando groff, que también lanzará los preprocesadores y post procesadores en el orden correcto y con las opciones necesarias.

Dependencias

Para instalar Groff−1.17.2 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib
bison: bison
diffutils: cmp
fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm, touch
gcc: cc1, cc1plus, collect2, cpp0, g++, gcc
grep: egrep, grep
make: make
gawk: awk
sed: sed
sh−utils: basename, date, echo, expr, hostname, uname
textutils: cat, tr

Gzip

Localización oficial para descarga

Gzip (1.2.4a):
ftp://ftp.gnu.org/gnu/gzip/

Gzip Patch (1.2.4a):
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/
http://ftp.linuxfromscratch.org/lfs−packages/3.3/

Contenido de gzip−1.2.4a

Programas

gunzip (enlace a gzip), gzexe, gzip, uncompress (enlace a gunzip), zcat (enlace a gzip), zcmp, zdiff, zforce, zgrep, zmore y znew

Descripción

gunzip, uncompress

gunzip y uncompress descomprimen ficheros que hayan sido comprimidos con gzip.

gzexe

gzexe permite comprimir ficheros ejecutables que automáticamente se descomprimen y ejecutan al ser lanzados (con una penalización en el rendimiento).

gzip

gzip reduce el tamaño de los ficheros usando codificación Lempel−Ziv (LZ77)

zcat

zcat descomprime una lista de ficheros en su línea de comandos o su entrada estándar, escribiendo los datos descomprimidos en la salida estándar.

zcmp

zcmp invoca al programa cmp en ficheros comprimidos.

zdiff

zdiff invoca al programa diff en ficheros comprimidos.

zforce

zforce fuerza la extensión .gz en todos los ficheros gzip para que gzip no los comprima dos veces. Esto puede ser útil para ficheros con el nombre truncado después de una transferencia de ficheros.

zgrep

zgrep invoca al programa grep en ficheros comprimidos.

zmore

zmore es un filtro que permite examinar ficheros comprimidos de texto plano pantalla a pantalla en un terminal emulado (similar al programa more).

znew

znew recomprime ficheros en formato .Z (compress) al formato .gz (gzip).

Dependencias

Para instalar Gzip−1.2.4a se necesita lo siguiente:

bash: sh
binutils: as, ld, nm
fileutils: chmod, cp, install, ln, mv, rm
gcc: cc1, collect2, cpp, cpp0, gcc
grep: egrep, grep
make: make
sed: sed
sh−utils: hostname
textutils: cat, tr

Kbd

Localización oficial para descarga

Kbd (1.06):
ftp://ftp.win.tue.nl/pub/linux−local/utils/kbd/

Kbd Patch (1.06−2):
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/
http://ftp.linuxfromscratch.org/lfs−packages/3.3/

Contenido de kbd−1.06

Programas

chvt, deallocvt, dumpkeys, fgconsole, getkeycodes, getunimap, kbd_mode, kbdrate, loadkeys, loadunimap, mapscrn, openvt, psfaddtable (enlace a psfxtable), psfgettable (enlace a psfxtable), psfstriptable (enlace a psfxtable), psfxtable, resizecons, setfont, setkeycodes, setleds, setlogcons, setmetamode, setvesablank, showfont, showkey, unicode_start, y unicode_stop

Descripciones

chvt

chvt cambia la terminal virtual que aparece en primer plano.

deallocvt

deallocvt desasigna las terminales virtuales no usadas.

dumpkeys

dumpkeys vuelca las tablas de traducción del teclado.

fgconsole

fgconsole muestra el número del terminal virtual activo.

getkeycodes

getkeycodes muestra la tabla de correspondencias de código de exploración (scan code) a código de teclas del núcleo.

getunimap

getunimap muestra el mapa unicode actualmente usado.

kbd_mode

kbd_mode muestra o establece el modo del teclado.

kbdrate

kbdrate establece la repetición y retardo del teclado.

loadkeys

loadkeys carga las tablas de traducción del teclado.

loadunimap

loadunimap carga la tabla de correspondencia de unicode a fuente del núcleo.

mapscrn

mapscrn carga una tabla de correspondencia de caracteres de salida definida por el usuario en el controlador de la consola. Este comando está obsoleto y sus funciones se encuentran incluidas en setfont.

openvt

openvt comienza un programa en un nuevo terminal virtual (VT)

psfaddtable, psfgettable, psfstriptable, psfxtable

Este es un grupo de herramientas para obtener tablas de caracteres Unicode a partir de fuentes de consola.

resizecons

resizecons cambia la idea del núcleo sobre el tamaño de la consola.

setfont

Esto permite cambiar las fuentes EGA/VGA de la consola.

setkeycodes

setkeycodes carga las entradas de la tabla de correspondencia de código de exploración (scan code) a código de tecla del núcleo.

setleds

setleds establece los LEDs del teclado. Mucha gente encuentra útil tener el bloqueo numérico (numlock) activado por defecto, y usando este programa puede conseguirse.

setlogcons

setlogcons envía los mensajes del núcleo a la consola.

setmetamode

setmetamode define como se manejan las teclas meta del teclado.

setvesablank

Esto permite afinar el salvapantallas incorporado en el hardware (no animados, sólo una pantalla en blanco).

showfont

showfont muestra los datos de una fuente. La información mostrada incluye información de la fuente, sus propiedades, la métrica de los caracteres y el mapa de bits de cada carácter.

showkey

showkey examina los códigos de exploración (scan codes) y los códigos de tecla enviados por el teclado.

unicode_start

unicode_start pone la consola en modo Unicode.

unicode_stop

unicode_stop revierte el teclado y la consola del modo Unicode.

Dependencias

Para instalar Kbd−1.06 se necesita lo siguiente:

bash: sh
binutils: as, ld, strip
bison: bison
diffutils: cmp
fileutils: cp, install, ln, mv, rm
flex: flex
gettext: msgfmt, xgettext
gcc: cc1, collect2, cpp0, gcc
grep: grep
gzip: gunzip, gzip
make: make
patch: patch
sed: sed
sh−utils: uname

Linux kernel

Localización oficial para descarga

Linux Kernel (2.4.18):
ftp://ftp.kernel.org/pub/linux/kernel/

Contenido de kernel–2.4.18

Ficheros de soporte

El núcleo Linux y los ficheros de cabecera (headers) del núcleo Linux

Descripciones

núcleo linux

El núcleo Linux es el corazón de todo sistema Linux. Es lo que da la marca Linux. Cuando se enciende un ordenador y se inicia un sistema Linux, el núcleo es lo primero que se carga. El núcleo inicializa los componentes hardware del sistema, como los puertos serie, puertos paralelo, tarjetas de sonido, tarjetas de red, controladores IDE, controladores SCSI y mucho más. En pocas palabras, el núcleo hace que el hardware esté disponible para que el software pueda ejecutarse.

ficheros de cabecera del núcleo linux

Estos son los ficheros que copiamos a /usr/include/{linux,asm} en el capítulo 5. Deben coincidir con la versión con la que glibc fue compilada. *No* deben reemplazarse cuando se actualiza el núcleo. Son esenciales para compilar muchos programas.

Dependencias

Para instalar Linux–2.4.18 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, nm, objcopy
fileutils: cp, ln, mkdir, mv, rm, touch
findutils: find, xargs
gcc: cc1, collect2, cpp0, gcc
grep: grep
gzip: gzip
make: make
gawk: awk
modutils: depmod, genksyms
net–tools: dnsdomainname, hostname
sed: sed
sh–utils: basename, date, expr, pwd, stty, uname, whoami, yes
textutils: cat, md5sum, sort, tail

Less

Localización oficial para descarga

Less (374):
ftp://ftp.gnu.org/gnu/less/

Contenido de less−358

Programas

less, lessecho y lesskey

Descripción

less

El programa less es un paginador de ficheros (o visor de texto). Muestra el contenido de un fichero con la posibilidad de recorrerlo. Less es una evolución del paginador habitual llamado "more". Less tiene la habilidad de poder moverse a través de los ficheros y no necesita leer por completo el fichero al principio, lo que le hace rápido cuando se leen ficheros largos.

lessecho

lessecho es necesario para expandir meta−caracteres, como * y ?, en los nombres de ficheros en sistemas Unix.

lesskey

lesskey se usa para especificar los códigos de teclas usados por less.

Dependencias

Para instalar Less−358 se necesita lo siguiente:

bash: sh
binutils: as, ld
diffutils: cmp
fileutils: chmod, install, mv, rm, touch
grep: egrep, grep
gcc: cc1, collect2, cpp0, gcc
make: make
sed: sed
sh−utils: expr, hostname, uname
textutils: cat, tr

LFS−Bootscripts

Localización oficial para descarga

LFS−Bootscripts (1.9):
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/
http://ftp.linuxfromscratch.org/lfs−packages/3.3/

Contenido de LFS–bootscripts–1.9

Ficheros

checkfs, cleanfs, functions, halt, loadkeys, localnet, mountfs, network, rc, reboot, sendsignals, setclock, swap, sysklogd y template

Descripciones

checkfs

El guión checkfs comprueba los sistemas de ficheros justo antes de ser montados (con la excepción de los que usan registros de transacciones [journal] o los que se montan desde la red)

cleanfs

El guión cleanfs elimina los ficheros que no deben guardarse cuando se arranca de nuevo el sistema, como /var/run/* y /var/lock/*, regenera /var/run/utmp y elimina los ficheros /etc/nologin, /fastboot y /forcefsck que pueda haber.

functions

El guión functions contiene funciones usadas por diferentes guiones: chequeo de errores, chequeo de estado, etc.

halt

El guión halt se encarga de cerrar el sistema.

loadkeys

El guión loadkeys carga el mapa de teclado apropiado para tu modelo de teclado.

localnet

El guión localnet establece el nombre de máquina usado por el sistema (hostname) y activa el dispositivo de red "loopback".

mountfs

El guión mountfs monta todos los sistemas de ficheros que no estén marcados como "noauto" o que no se monten a través de la red.

network

El guión network activa las interfaces de red (como las tarjetas de red) y establece la puerta de enlace por defecto (gateway) cuando es necesario.

rc

El guión rc es el controlador maestro de los niveles de arranque y es responsable de lanzar todos los demás guiones uno a uno en una secuencia específica.

reboot

El guión reboot se encarga de reiniciar el sistema.

sendsignals

El guión sendsignals se asegura de que todos los procesos terminen antes de parar o reiniciar el sistema.

setclock

El guión setclock fija el reloj del núcleo a la hora local en caso de que el reloj del ordenador no esté fijado a la hora GMT.

swap

El guión swap activa y desactiva las particiones y ficheros de intercambio (swap).

sysklogd

El guión sysklogd lanza y detiene los demonios de registro de eventos del sistema y del núcleo (syslogd y klogd).

template

El guión template es una plantilla para crear guiones de arranque personales para otros demonios y procesos.

Dependencias

Para instalar bootscripts−1.9 se necesita lo siguiente:

fileutils: chown, cp

Libtool

Localización oficial para descarga

Libtool (1.4.2):
ftp://ftp.gnu.org/gnu/libtool/

Contenido de libtool−1.4.2

Programas

libtool y libtoolize

Descripciones

libtool

Libtool proporciona servicios de soporte generalizados para la compilación de librerías.

libtoolize

libtoolize proporciona una forma estándar de añadir soporte para libtool a un paquete.

Librerías

libltdl.[a,so]

Descripciones

libltdl

Libtool proporciona una pequeña librería, llamada `libltdl', cuyo objetivo es ocultar las dificultades en la carga dinámica de librerías a los programadores.

Dependencias

Para instalar Libtool−1.4.2 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, nm, ranlib, strip
diffutils: cmp
fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm, rmdir
gcc: cc, cc1, collect2, cpp0
glibc: ldconfig
grep: egrep, fgrep, grep
make: make
sed: sed
sh−utils: echo, expr, hostname, sleep, uname
texinfo: install−info
textutils: cat, sort, tr, uniq

Lilo

Localización oficial para descarga

Lilo (22.2):
ftp://ibiblio.org/pub/Linux/system/boot/lilo/
http://ibiblio.org/pub/Linux/system/boot/lilo/

Contenido de lilo−22.1

Programas

lilo y mkrescue

Descripciones

lilo

lilo instala el gestor de arranque que se usa para iniciar un sistema Linux.

mkrescue

mkrescue crea un disquete de rescate arrancable usando el núcleo existente y cualquier disco RAM de inicio.

Dependencias

Para instalar Lilo−22.1 se necesita lo siguiente:

bash: sh
bin86: as86, ld86
binutils: as, ld, strip
fileutils: cp, dd, ln
gcc: cc, cc1, collect2, cpp0
make: make
sed: sed
textutils: cat

M4

Localización oficial para descarga

M4 (1.4):
ftp://ftp.gnu.org/gnu/m4/

Contenido de m4−1.4

Programas

m4

Descripciones

m4

M4 es un procesador de macros. Copia la entrada a la salida expandiendo las macros en el proceso. Las macros pueden ser internas o definidas por el usuario y pueden tomar cualquier número de argumentos. Aparte de hacer la expansión de macros, m4 tiene funciones internas para la inclusión de los ficheros indicados, lanzar comandos UNIX, hacer aritmética entera, manipular texto de diversas formas, recursión, etc. M4 puede ser usado como interfaz para un compilador o como procesador de macros por sí mismo.

Dependencias

Para instalar M4−1.4 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, cp, install, mv, rm
make: make
gcc: cc1, collect2, cpp0, gcc
grep: egrep, grep
sed: sed
sh−utils: date, echo, hostname
textutils: cat, tr

Make

Localización oficial para descarga

Make (3.79.1):
ftp://ftp.gnu.org/gnu/make/

Contenido de make−3.79.1

Programas

make

Descripciones

make

make determina qué partes de un programa necesitan ser recompiladas, y lanza los comandos para hacerlo.

Dependencias

Para instalar Make−3.79.1 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: as, ld
diffutils: cmp
fileutils: chgrp, chmod, install, ls, mv, rm
gcc: cc, cc1, collect2, cpp0, gcc
glibc: getconf
grep: egrep, fgrep, grep

m4: m4
make: make
gawk: gawk
sed: sed
sh−utils: basename, echo, expr, hostname, sleep, uname
texinfo: install−info, makeinfo
textutils: cat, tr

MAKEDEV

Localización oficial para descarga

MAKEDEV (1.4):
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/
http://ftp.linuxfromscratch.org/lfs−packages/3.3/

Contenido de MAKEDEV−1.4

Programas

MAKEDEV

Descripciones

MAKEDEV

MAKEDEV es un guión que puede ayudar en la creación de los ficheros de dispositivos estáticos necesarios, que usualmente residen en el directorio /dev. Más información sobre los ficheros de dispositivos puede encontrase dentro de las fuentes del núcleo en `Documentation/devices.txt`.

Dependencias

Para instalar MAKEDEV−1.4 se necesita lo siguiente:

bash: sh
fileutils: chmod, chown, cp, ln, mknod, mv, rm
grep: grep
sh−utils: expr, id

Man

Localización oficial para descarga

Man (1.5j):
ftp://ftp.win.tue.nl/pub/linux−local/utils/man/

Contenido de man−1.5j

Programas

apropos, makewhatis, man, man2dvi, man2html y whatis

Descripciones

apropos

apropos busca las palabras claves indicadas en un grupo de ficheros de bases de datos que contienen descripciones cortas de los comandos del sistema, y muestra el resultado en la salida estándar.

makewhatis

makewhatis lee todas las páginas de manual contenidas en las secciones incluidas en las rutas "manpath" o las páginas preformateadas contenidas en las secciones de las rutas "catpath". Por cada página incluye una línea en la base datos de whatis; cada línea consiste en el nombre de la página y una descripción corta, separados por un guión. La descripción se extrae del contenido de la sección NAME de la página de manual.

man

man formatea y muestra en línea las páginas de manual.

man2dvi

man2dvi convierte una página de manual al formato dvi.

man2html

man2html convierte una página de manual en html.

whatis

whatis busca las palabras claves indicadas en un grupo de ficheros de bases de datos que contienen descripciones cortas de los comandos del sistema, y muestra el resultado en la salida estándar. Sólo las coincidencias de palabras completas son mostradas.

Dependencias

Para instalar Man−1.5i2 se necesita lo siguiente:

bash: sh
binutils: as, ld
fileutils: chmod, cp, install, mkdir, rm
gcc: c11, collect2, cpp0, gcc
grep: grep
make: make
gawk: awk
sed: sed

sh−utils: echo
textutils: cat

Man−pages

Localización oficial para descarga

Man−pages (1.48):
ftp://ftp.kernel.org/pub/linux/docs/manpages/

Contenido de manpages−1.47

Ficheros de soporte

Varias páginas de manual que no vienen con los paquetes.

Descripciones

páginas de manual

Un ejemplo de las páginas de manual incluidas son las que describen todas las funciones C y C++, algunos ficheros de /dev y otras cosas.

Dependencias

Para instalar Man−pages−1.47 se necesita lo siguiente:

bash: sh
fileutils: install
make: make

Modutils

Localización oficial para descarga

Modutils (2.4.15):
ftp://ftp.kernel.org/pub/linux/utils/kernel/modutils/

Contenido de modutils−2.4.12

Programas

depmod, genksyms, insmod, insmod_ksymoops_clean, kallsyms (enlace a insmod), kernelversion, ksyms, lsmod (enlace a insmod), modinfo, modprobe (enlace a insmod) y rmmod

Descripciones

depmod

depmod maneja las descripciones de las dependencias para los módulos del núcleo

genksyms

genksyms lee (de la entrada estándar) la salida de gcc −E source.c y genera un fichero que contiene información sobre la versión.

insmod

insmod instala un módulo dentro del núcleo en ejecución.

insmod_ksymoops_clean

insmod_ksymoops_clean borra los símbolos del núcleo (ksyms) salvados y los módulos a los que no se ha accedido en los últimos 2 días.

kallsyms

kallsyms extrae todos los símbolos del núcleo para la depuración.

kernelversion

kernelversion informa sobre la versión mayor del núcleo en ejecución.

ksyms

ksyms muestra los símbolos exportados del núcleo

lsmod

lsmod muestra información sobre todos los módulos cargados.

modinfo

modinfo examina un fichero objeto asociado con un módulo del núcleo y muestra la información que pueda encontrar.

modprobe

modprobe usa un fichero de dependencias similar a un Makefile creado por depmod, para cargar automáticamente el módulo o módulos necesarios del conjunto de módulos disponibles en el árbol de directorios predefinido.

rmmod

rmmod descarga los módulos del núcleo en ejecución.

Dependencias

Para instalar Modutils−2.4.12 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib, strip
bison: bison
diffutils: cmp
fileutils: chmod, install, ln, mkdir, mv, rm
flex: flex
gcc: cc, cc1, collect2, cpp0, gcc
grep: egrep, grep
make: make
sed: sed
sh−utils: basename, expr, hostname, uname
textutils: cat, tr

Ncurses

Localización oficial para descarga

Ncurses (5.2):
ftp://ftp.gnu.org/gnu/ncurses/

Contenido de ncurses−5.2

Programas

captoinfo (enlace a tic), clear, infocmp, infotocap (enlace a tic), reset (enlace a tset), tack, tic, toe, tput y tset.

Descripciones

captoinfo

captoinfo convierte una descripción de termcap en una descripción de terminfo.

clear

clear limpia la pantalla si es posible. Busca en el entorno el tipo de terminal y en la base de datos de terminfo para figurarse como limpiar la pantalla.

infocmp

infocmp puede usarse para comparar una entrada binaria de terminfo con otras entradas terminfo, reescribir una descripción de terminfo para aprovechar el campo "use=", o mostrar una descripción terminfo del fichero binario (term) en una variedad de formatos (lo opuesto de lo que hace tic).

infotocap

infotocap convierte una descripción terminfo en una descripción termcap.

reset

reset activa los modos "cooked" y "echo", quita los modos "cbreak" y "raw", activa la traslación de nueva línea y restablece cualquier carácter especial desactivado a sus valores por defecto, antes de hacer la inicialización del terminal de la misma manera que tset.

tack

tack es el comprobador de acciones de terminfo.

tic

tic es el compilador de entradas de descripciones de terminfo. El programa transforma un fichero terminfo en formato fuente a formato binario para su uso con las rutinas de las librerías ncurses. Los ficheros terminfo contienen información sobre las capacidades de un terminal.

toe

toe lista todos los tipos de terminal disponibles por su nombre primario, con descripciones.

tput

tput usa la base de datos de terminfo para poner a disposición del intérprete de comandos la información sobre las capacidades dependientes del terminal, para inicializar o restablecer el terminal, o para devolver el nombre largo del tipo de terminal requerido.

tset

tset inicializa los terminales para poder usarlos, pero no se usa posteriormente. Incluye compatibilidad con 4.4BSD.

Librerías

libcurses.[a,so] (enlace a libncurses.[a,so]), libform.[a,so], libform_g.a, libmenu.[a,so], libmenu_g.a, libncurses++.a, libncurses.[a,so], libncurses_g.a, libpanel.[a,so] y libpanel_g.a

libcurses, libncurses++, libncurses, libncurses_g

Las librerías que componen Ncurses se usan para mostrar texto (a menudo de forma vistosa) en la pantalla. Un ejemplo donde se usa ncurses es en el proceso "make menuconfig" del núcleo. Las librerías libncurses son la base del sistema.

libform, libform_g

libform se usa para implementar formularios en ncurses.

libmenu, libmenu_g

libmenu se usa para implementar menús en ncurses.

libpanel, libpanel_g

libpanel se usa para implementar paneles en ncurses.

Dependencias

Para instalar Ncurses−5.2 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, cp, install, ln, mkdir, mv, rm
gcc: c++, cc1, cc1plus, collect2, cpp0, gcc
glibc: ldconfig
grep: egrep, fgrep, grep
make: make
gawk: gawk
sed: sed
sh−utils: basename, date, echo, expr, hostname, uname
textutils: cat, sort, tr, wc

Netkit−base

Localización oficial para descarga

Netkit−base (0.17):
ftp://ftp.uk.linux.org/pub/linux/Networking/netkit/

Contenido de netkit−base−0.17

Programas

inetd y ping

Descripciones

inetd

inetd es la madre de todos los demonios (daemons). Escucha las peticiones de conexión y transfiere la llamada al demonio apropiado.

ping

ping envía paquetes ICMP ECHO_REQUEST a otra máquina y determina su tiempo de respuesta.

Dependencias

Para instalar Netkit−base−0.17 se necesita lo siguiente:

bash: sh
binutils: as, ld, strip
fileutils: cp, install, rm
make: make
gcc: cc1, collect2, cpp0, gcc
sed: sed
sh−utils: date
textutils: cat

Net−tools

Localización oficial para descarga

Net−tools (1.60):
http://www.tazenda.demon.co.uk/phil/net−tools/

Contenido de net−tools−1.60

Programas

arp, dnsdomainname (enlace a hostname), domainname (enlace a hostname), hostname, ifconfig, nameif, netstat, nisdomainname (enlace a hostname), plipconfig, rarp, route, slattach y ypdomainname (enlace a hostname)

Descripciones

arp

arp se usa para manipular la caché ARP del núcleo, usualmente para añadir o borrar una entrada o volcar dicha caché.

dnsdomainname

dnsdomainname muestra el nombre del dominio DNS del sistema.

domainname

domainname muestra o establece el nombre del dominio NIS/YP del sistema.

hostname

hostname se usa para establecer o mostrar el nombre del sistema.

ifconfig

El comando ifconfig es el comando general usado para configurar las interfaces de red.

nameif

nameif nombra interfaces de red basándose en direcciones MAC.

netstat

netstat es una herramienta multipropósito usada para mostrar las conexiones de red, tablas de encaminamiento, estadísticas de las interfaces, conexiones enmascaradas y los miembros de conexiones multidestino (multicast).

nisdomainname

nisdomainname muestra o establece el nombre de dominio NIS/YP del sistema.

plipconfig

plipconfig se usa para afinar los parámetros del dispositivo PLIP, para hacerlo más rápido.

rarp

Relacionado con el programa arp, el programa rarp manipula la tabla RARP del sistema.

route

route es la utilidad general que se usa para manipular la tabla de encaminamiento IP.

slattach

slattach conecta una interfaz de red a una línea serie, esto es, pone una línea de terminal normal en uno o varios modos "de red".

ypdomainname

ypdomainname muestra o establece el nombre de dominio NIS/YP del sistema.

Dependencias

Para instalar Net−tools−1.60 se necesita lo siguiente:

bash: bash, sh
binutils: ar, as, ld
fileutils: install, ln, ls, mv, rm
gcc: cc, cc1, collect2, cpp0
make: make
sh−utils: echo

Patch

Localización oficial para descarga

Patch (2.5.4):
ftp://ftp.gnu.org/gnu/patch/

Contenido de patch−2.5.4

Programas

patch

Descripciones

patch

El programa patch modifica un fichero según lo indicado en un fichero de parche. Normalmente un fichero de parche es una lista creada por el programa diff que contiene instrucciones sobre cómo necesita ser modificado un fichero original. Patch se usa mucho para parchear el código fuente pues ahorra bastante tiempo y espacio. Imagina un paquete de 1MB de tamaño. La siguiente versión de ese paquete sólo cambia en dos ficheros con respecto a la primera versión. Se puede distribuir como un nuevo paquete entero de 1MB o sólo como un parche de 1KB con el que actualizar la primera versión para hacerla idéntica a la segunda. Por tanto, si la primera versión está todavía descargada, un parche evita hacer una segunda descarga larga.

Dependencias

Para instalar Patch−2.5.4 se necesita lo siguiente:

bash: sh
binutils: as, ld
diffutils: cmp
fileutils: chmod, install, mv, rm
gcc: cc, cc1, collect2, cpp0, gcc
glibc: getconf
grep: egrep, grep
make: make
sed: sed
sh−utils: echo, expr, hostname, uname
textutils: cat, tr

Perl

Localización oficial para descarga

Perl (5.6.1):
http://www.perl.com/

Contenido de perl–5.6.1

Programas

a2p, c2ph, dprofpp, find2perl, h2ph, h2xs, perl, perl5.6.1, perlbug, perlcc, perldoc, pl2pm, pod2html, pod2latex, pod2man, pod2text, pod2usage, podchecker, podselect, pstruct, s2p y splain

Descripciones

a2p

a2p es un traductor de awk a perl.

c2ph

c2ph vuelca estructuras C similares a las generadas por "cc −g −S".

dprofpp

dprofpp muestra datos de perfiles perl.

find2perl

find2perl traduce líneas del comando find a código Perl.

h2ph

h2ph convierte ficheros de cabecera .h de C en ficheros de cabecera .ph de Perl.

h2xs

h2xs convierte ficheros de cabecera .h de C en extensiones de Perl.

perl, perl5.6.1

perl es el Lenguaje Práctico de Extracción e Informe. Combina algunas de las mejores características de C, sed, awk y sh en un poderoso lenguaje.

perlbug

perlbug ayuda en la generación de informes de errores sobre perl o de los módulos incorporados, y los envía por correo.

perlcc

perlcc genera ejecutables a partir de programas Perl.

perldoc

perldoc examina la parte de la documentación en formato .pod que se incluye en el árbol de instalación de perl o en un guión de perl, y lo muestra mediante "pod2man | nroff −man | $PAGER".

pl2pm

pl2pm es una herramienta que ayuda en la conversión de librerías .pl de estilo Perl4 en módulos de librería de estilo Perl5.

pod2html

pod2html convierte ficheros de formato pod a formato HTML.

pod2latex

pod2latex convierte ficheros de formato pod a formato LaTeX.

pod2man

pod2man convierte datos pod en entradas formateadas *roff.

pod2text

pod2text convierte datos pod en texto formateado ASCII.

pod2usage

pod2usage muestra mensajes de uso a partir de documentos pod incluidos en ficheros.

podchecker

podchecker chequea la sintaxis de los ficheros de documentación en formato pod.

podselect

podselect muestra las secciones seleccionadas de la documentación pod en la salida estándar.

pstruct

pstruct vuelca estructuras C similares a las generadas por "cc −g −S".

s2p

s2p es un traductor de sed a perl.

splain

splain es un programa que fuerza diagnósticos de avisos exhaustivos en perl.

Dependencias

Para instalar Perl−5.6.1 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, nm

diffutils: cmp
fileutils: chmod, cp, ln, ls, mkdir, mv, rm, touch
gcc: cc, cc1, collect2, cpp0, gcc
grep: egrep, grep
make: make
gawk: awk
sed: sed
sh−utils: basename, date, echo, expr, hostname, pwd, uname, whoami
textutils: cat, comm, sort, split, tr, uniq, wc

Procinfo

Localización oficial para descarga

Procinfo (18):
ftp://ftp.cistron.nl/pub/people/svm/

Contenido de procinfo−18

Programas

lsdev, procinfo y socklist

Descripciones

lsdev

lsdev recopila la información sobre los dispositivos físicos instalados en su ordenador a partir de los ficheros con las interrupciones, puertos de entrada/salida y acceso directo a memoria (DMA) del directorio /proc, facilitando una descripción rápida de qué direcciones de entrada/salida (I/O) y qué canales IRQ y DMA usa cada dispositivo.

procinfo

procinfo recopila algunos datos del sistema a partir del directorio /proc y los muestra en un bonito formato en el dispositivo de salida estándar.

socklist

Es un guión de Perl que facilita una lista de todos los puertos abiertos, enumerando el tipo, puerto, inodo, identificador de usuario (uid), identificador de proceso (pid), descriptor de ficheros (fd) y el programa al que pertenece.

Dependencias

Para instalar Procinfo−18 se necesita lo siguiente:

binutils: as, ld
fileutils: install, mkdir

gcc: cc1, collect2, cpp0, gcc
make: make

Procps

Localización oficial para descarga

Procps (2.0.7):
ftp://people.redhat.com/johnsonm/procps/

Contenido de procps–2.0.7

Programas

free, kill, oldps, pgrep, pkill, ps, skill, snice, sysctl, tload, top, uptime, vmstat, w y watch

Descripciones

free

free muestra la cantidad total de memoria física y de intercambio (swap) libre y usada en el sistema, al igual que la memoria compartida (shared) y de almacenamiento intermedio (buffers) usada por el núcleo.

kill

kill envía señales a los procesos.

oldps y ps

ps facilita una instantánea de los procesos actuales.

pgrep

pgrep visualiza procesos basándose en el nombre u otros atributos

pkill

pkill envía señales a procesos basándose en el nombre u otros atributos

skill

skill envía señales a procesos que coincidan con un criterio.

snice

snice cambia la prioridad de planificación de procesos que coincidan con un criterio.

sysctl

sysctl modifica los parámetros del núcleo en tiempo de ejecución.

tload

tload imprime un gráfico de la carga actual del sistema en la consola (tty) especificada (o la consola del proceso tload si no se especifica ninguna).

top

top proporciona una vista dinámica de la actividad del procesador en tiempo real.

uptime

uptime facilita en una línea de pantalla la siguiente información: la hora, cuanto tiempo lleva el sistema funcionando, cuantos usuarios hay actualmente dentro del sistema, y la carga del sistema en los últimos 1, 5 y 15 minutos.

vmstat

vmstat muestra información sobre los procesos, memoria, paginación, bloques de entrada/salida y actividad del procesador.

w

w muestra información sobre los usuarios que hay actualmente en el sistema y sus procesos.

watch

watch lanza comandos repetidamente, mostrando sus salidas (a pantalla completa la primera).

Librerías

libproc.so

Descripciones

libproc

libproc es la librería contra la que muchos de los programas de este grupo están enlazados para ahorrar espacio en disco implementando las funciones comunes sólo una vez.

Dependencias

Para instalar Procps−2.0.7 se necesita lo siguiente:

bash: sh
binutils: as, ld, strip
fileutils: install, ln, mv, rm
gcc: cc1, collect2, cpp0, gcc
grep: grep
make: make
gawk: awk
sed: sed

sh−utils: basename, pwd
textutils: sort, tr

Psmisc

Localización oficial para descarga

Psmisc (20.2):
http://download.sourceforge.net/psmisc/
ftp://download.sourceforge.net/pub/sourceforge/psmisc/

Contenido de psmisc−20.2

Programas

fuser, killall, pidof (enlace a killall) y pstree

Nota: en LFS no instalamos el enlace pidof por defecto porque usamos en su lugar el programa pidof de sysvinit.

Descripciones

fuser

fuser muestra los números de identificación de los procesos (PID) usando los ficheros o sistemas de ficheros especificados.

killall

killall envía una señal a todos los procesos en ejecución de los comandos especificados .

pidof

Pidof encuentra los identificadores de procesos (PIDs) de los programas mencionados e imprime esos identificadores en la salida estándar.

pstree

pstree muestra los procesos en ejecución como un árbol.

Dependencias

Para instalar Psmisc−20.2 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
bison: bison
binutils: as, ld

diffutils: cmp
fileutils: chmod, install, ls, mkdir, mv, rm
gettext: msgfmt, xgettext
gcc: cc, cc1, collect2, cpp0, gcc
grep: egrep, grep
m4: m4
make: make
gawk: gawk
sed: sed
sh−utils: basename, echo, expr, hostname, sleep, uname
texinfo: makeinfo
textutils: cat, tr

Reiserfsprogs

Localización oficial para descarga

Reiserfs (3.x.1b):
ftp://ftp.namesys.com/pub/reiserfsprogs/

Contenido de reiserfsprogs−3.x.0j

Programas

debugreiserfs, mkreiserfs, reiserfsck, resize_reiserfs y unpack

Descripciones

debugreiserfs

debugreiserfs puede ayudar en ocasiones en la resolución de problemas con los sistemas de ficheros reiserfs. Si se le llama sin opciones muestra el superbloque de cualquier sistema de ficheros reiserfs encontrado en el dispositivo.

mkreiserfs

mkreiserfs crea un sistema de ficheros reiserfs.

reiserfsck

reiserfsck chequea un sistema de ficheros reiserfs.

resize_reiserfs

resize_reiserfs se usa para redimensionar un sistema de ficheros reiserfs no montado.

unpack

No hay descripción disponible.

Dependencias

Para instalar Reiserfs−3.x.0j se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, install, ls, rm
gcc: cc1, collect2, cpp0, gcc
grep: egrep, grep
m4: m4
make: make
gawk: gawk
sed: sed
sh−utils: echo, expr, hostname, sleep
texinfo: makeinfo
textutils: cat, tr

Sed

Localización oficial para descarga

Sed (3.02):
ftp://ftp.gnu.org/gnu/sed/

Contenido de sed−3.02

Programas

sed

Descripciones

sed

sed es un editor de flujo. Un editor de flujo se usa para realizar transformaciones básicas de texto en un flujo de entrada (un fichero o una tubería).

Dependencias

Para instalar Sed−3.02 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh

binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, install, ls, mv, rm
gcc: cc1, collect2, cpp0, gcc
glibc: getconf
grep: egrep, fgrep, grep
m4: m4
make: make
gawk: gawk
sed: sed
sh−utils: echo, expr, hostname, sleep
texinfo: install−info, makeinfo
textutils: cat, tr

Entorno de Contraseñas Ocultas (Shadow Password Suite)

Localización oficial para descarga

Shadow Password Suite (4.0.3):
ftp://ftp.pld.org.pl/software/shadow/

Contenido de shadow−20001016

Programas

chage, chfn, chpasswd, chsh, dpasswd, expiry, faillog, gpasswd, groupadd, groupdel, groupmod, grpck, grpconv, grpunconv, lastlog, login, logoutd, mkpasswd, newgrp, newusers, passwd, pwck, pwconv, pwunconv, sg (enlace a newgrp), su, useradd, userdel, usermod, vigr (enlace a vipw) y vipw

Descripciones

chage

chage cambia el número de días entre cambios de la contraseña y la fecha del último cambio de contraseña.

chfn

chfn cambia la información en las cuentas de usuario relativa al nombre completo del usuario, el número y la extensión de su oficina, y el número de teléfono personal.

chpasswd

chpasswd lee un fichero con pares de usuarios y contraseñas en la entrada estándar y usa esta información para actualizar un grupo de usuarios existentes.

chsh

chsh cambia el intérprete de comandos que se ejecuta cuando un usuario entra al sistema.

dpasswd

dpasswd añade, borra y actualiza las contraseñas de acceso telefónico del usuario.

expiry

Comprueba y fuerza la política de expiración de contraseñas.

faillog

faillog formatea el contenido del registro de fallos /var/log/faillog, y mantiene el contador y los límites de fallos.

gpasswd

gpasswd se usa para administrar el fichero /etc/group.

groupadd

El comando groupadd crea un nuevo grupo usando los valores especificados en la línea de comandos y los valores por defecto del sistema.

groupdel

El comando groupdel modifica los ficheros con las cuentas del sistema, borrando todas las entradas referidas a un determinado grupo.

groupmod

El comando groupmod modifica los ficheros de las cuentas del sistema para reflejar los cambios especificados en la línea de comandos.

grpck

grpck verifica la integridad de la información de autenticación del sistema.

grpconv

grpunconv convierte a ficheros de grupos ocultos los ficheros de grupos normales.

grpunconv

grpunconv convierte los ficheros de grupos ocultos en ficheros de grupos normales.

lastlog

lastlog formatea y muestra el contenido del registro de último acceso /var/log/lastlog. Muestra el nombre con el que se ha accedido, puerto y hora del último acceso.

login

login se usa para establecer una nueva sesión con el sistema.

logoutd

logoutd fuerza las restricciones de hora de acceso y puerto especificadas en /etc/porttime.

mkpasswd

mkpasswd lee un fichero en el formato facilitado por las opciones y lo convierte al formato de fichero de base de datos correspondiente.

newgrp

newgrp se usa para cambiar el identificador de grupo actual durante una sesión de acceso.

newusers

newusers lee un fichero con pares de nombres de usuario y contraseñas en texto plano y usa esa información para actualizar un grupo de usuarios existentes o crear nuevos usuarios.

passwd

passwd cambia las contraseñas de las cuentas de usuarios y grupos.

pwck

pwck verifica la integridad de la información de autenticación del sistema.

pwconv

pwconv convierte a ficheros de contraseñas ocultas los ficheros de contraseñas normales.

pwunconv

pwunconv convierte los ficheros de contraseñas ocultas en ficheros normales.

sg

sg ejecuta comandos con un identificador de grupo diferente.

su

Cambia los identificadores efectivos de usuario y grupo de un usuario. Reemplaza al programa su instalado por el paquete Shellutils.

useradd

useradd crea un nuevo usuario o actualiza la información por defecto de un nuevo usuario.

userdel

userdel modifica los ficheros con las cuentas del sistema, borrando todas las entradas referidas al nombre de acceso especificado.

usermod

usermod modifica los ficheros con las cuentas del sistema para reflejar los cambios especificados en la línea de comandos.

vipw y vigr

vipw y vipgr pueden editar los ficheros /etc/passwd y /etc/group, respectivamente. Con la opción −s, pueden editar la versión oculta de dichos ficheros, /etc/shadow y /etc/gshadow, respectivamente.

Librerías

libshadow.[a,so]

Descripciones

libshadow

libshadow proporciona funcionalidades comunes para los programas de ocultación de contraseñas.

Dependencias

Para instalar Shadow−20001016 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: ar, as, ld, nm, ranlib
diffutils: cmp
fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm, rmdir
gettext: msgfmt, xgettext
gcc: cc1, collect2, cpp0, gcc
glibc: ldconfig
grep: egrep, grep
m4: m4
make: make
gawk: gawk
net−tools: hostname
sed: sed
sh−utils: basename, echo, expr, sleep, uname
texinfo: makeinfo
textutils: cat, sort, tr, uniq

Sh−utils

Localización oficial para descarga

Sh−utils (2.0):
ftp://ftp.gnu.org/gnu/sh−utils/

Sh−utils Patch (2.0):
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/
http://ftp.linuxfromscratch.org/lfs−packages/3.3/

Contenido de sh−utils−2.0

Programas

basename, chroot, date, dirname, echo, env, expr, factor, false, groups, hostid, hostname, id, logname, nice, nohup, pathchk, pinky, printenv, printf, pwd, seq, sleep, stty, su, tee, test, true, tty, uname, uptime, users, who, whoami y yes

Descripciones

basename

basename elimina los directorios y las extensiones de los nombres de ficheros.

chroot

chroot ejecuta un comando o un intérprete de comandos (shell) interactivo dentro de un directorio raíz determinado.

date

date muestra la fecha y hora actual en un formato determinado o establece la fecha y hora del sistema.

dirname

dirname elimina los sufijos que no son directorios del nombre de un fichero.

echo

echo muestra una línea de texto.

env

env ejecuta un programa en un entorno modificado.

expr

expr evalúa expresiones.

factor

factor muestra los factores primos de los números especificados.

false

false siempre termina con un código de estado que indica un fallo.

groups

groups muestra los grupos a los que pertenece un usuario.

hostid

hostid muestra el identificador numérico (en hexadecimal) de la máquina actual.

hostname

hostname establece o muestra el nombre de la máquina actual.

id

id muestra el identificador de usuario (UID) real y efectivo y el identificador de grupo (GID) real y efectivo de un usuario.

logname

logname muestra el nombre de acceso (login name) del usuario actual.

nice

nice ejecuta un programa con una prioridad distinta.

nohup

nohup ejecuta un comando que no se interrumpe cuando se cierra la sesión, y cuya salida no se envía a un terminal.

pathchk

pathchk comprueba si los nombres de ficheros son válidos o portables.

pinky

pinky es una utilidad parecida a finger que obtiene información sobre un determinado usuario.

printenv

printenv muestra todo o parte del entorno.

printf

printf formatea y muestra datos (de la misma manera que la función printf de C).

pwd

pwd muestra el nombre del directorio de trabajo actual.

seq

seq muestra números en un cierto rango y con un cierto incremento.

sleep

sleep establece un retardo durante un determinado instante de tiempo.

stty

stty cambia y muestra las opciones de configuración del terminal.

su

su ejecuta un intérprete de comandos (shell) con un identificador de usuario y de grupo diferentes.

tee

tee lee de la entrada estándar y escribe en la salida estándar y en ficheros.

test

test comprueba el tipo de los ficheros y compara valores.

true

true siempre termina con un código de estado que indica éxito.

tty

tty muestra el nombre de fichero del terminal conectado a la entrada estándar.

uname

uname muestra información del sistema.

uptime

uptime muestra cuanto tiempo hace que el sistema está en marcha.

users

users muestra los nombres de los usuarios conectados actualmente.

who

who muestra quién está conectado.

whoami

whoami muestra el identificador de usuario efectivo.

yes

yes muestra en pantalla una cadena de texto indefinidamente.

Dependencias

Para instalar Sh−utils−2.0 se ecesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, chown, install, ls, mv, rm
gettext: msgfmt, xgettext
gcc: cc, cc1, collect2, cpp0, gcc
glibc: getconf
grep: egrep, fgrep, grep
m4: m4
make: make
gawk: gawk
perl: perl
sed: sed
sh−utils: basename, echo, expr, hostname, sleep, uname
tar: tar
texinfo: install−info, makeinfo
textutils: cat, tr

Sysklogd

Localización oficial para descarga

Sysklogd (1.4.1):
http://www.infodrom.org/projects/sysklogd/

Contenido de sysklogd−1.4.1

Programas

klogd y syslogd

Descripciones

klogd

klogd es un demonio del sistema que intercepta y registra los mensajes del núcleo Linux.

syslogd

syslogd proporciona una forma de registrar sucesos que muchos programas modernos utilizan. Cada mensaje registrado contiene como mínimo un campo con el nombre de la máquina y la fecha, y normalmente un campo con el nombre del programa también, pero eso depende de lo fiable que sea el programa a registrar.

Dependencias

Para instalar Sysklogd−1.4.1 se necesita lo siguiente:

binutils: as, ld, strip
fileutils: install
gcc: cc1, collect2, cpp0, gcc
make: make

Sysvinit

Localización oficial para descarga

Sysvinit (2.84):
ftp://ftp.cistron.nl/pub/people/miquels/sysvinit/

Contenido de sysvinit−2.84

Programas

halt, init, killall5, last, lastb (enlace a last), mesg, pidof (enlace a killall5), poweroff (enlace a halt), reboot (enlace a halt), runlevel, shutdown, sulogin, telinit (enlace a init), utmpdump y wall

Descripciones

halt

halt anota en el fichero /var/log/wtmp que el sistema se va a venir abajo, y entonces le indica al núcleo que cierre, reinicie o apague el sistema. Si halt o reboot son llamados cuando el sistema no está en los niveles de ejecución 0 o 6, shutdown es invocado en su lugar (con las opciones −h o −r).

init

init es el padre de todos los procesos. Su función principal es crear procesos a partir de un guión almacenado en el fichero /etc/inittab. Este fichero normalmente tiene unas entradas que hacen que init active la creación de los terminales en cada línea en la que los usuarios pueden conectarse. También controla los procesos autónomos requeridos por un sistema particular.

killall5

killall5 es el comando killall de SystemV. Envía una señal a todos los procesos excepto a los procesos de su propia sesión, por tanto no puede matar el intérprete de comandos en el que se esté ejecutando el guión desde el que fue llamado.

last

last busca hacia atrás en el fichero /var/log/wtmp (o el fichero indicado con la opción −f) y muestra una lista con todos los usuarios conectados (y desconectados) desde que el fichero fue creado.

lastb

lastb es lo mismo que last, excepto que por defecto muestra los registros del fichero /var/log/btmp, que contiene todos los intentos fallidos de conexión.

mesg

mesg controla el acceso al terminal de un usuario por otros. Se usa para permitir o denegar a otros usuarios escribir en su terminal.

pidof

pidof encuentra los identificadores de procesos (PIDs) de los programas mencionados y muestra esos identificadores en la salida estándar.

poweroff

poweroff es equivalente a "shutdown −h −p now". Para el ordenador y lo apaga (cuando se usa una BIOS compatible APM y APM está activado en el núcleo).

reboot

reboot es equivalente a "shutdown −r now". Reinicia el ordenador.

runlevel

runlevel lee el fichero utmp del sistema (típicamente /var/run/utmp) para localizar el registro del nivel de ejecución, y entonces muestra el anterior y el nivel de ejecución actual del sistema en la salida estándar, separados por un espacio.

shutdown

shutdown provoca la caída del sistema de una forma segura. Todos los usuarios conectados son notificados de que el sistema se va a venir abajo, y se bloquean los intentos de conexión al sistema.

sulogin

sulogin es invocado por init cuando el sistema entra en el modo monousuario (esto se hace mediante una entrada en /etc/inittab). Init también intenta ejecutar sulogin cuando se le pasa la opción −b desde el gestor de arranque (p.e., LILO).

telinit

telinit envía las señales apropiadas a init, diciendo a qué nivel de ejecución debe cambiar.

utmpdump

utmpdump muestra el contenido de un fichero (usualmente /var/run/utmp) en la salida estándar en un formato comprensible por el usuario.

wall

wall envía un mensaje a todos los usuarios conectados que tengan los permisos de mesg puestos a "yes".

Dependencias

Para instalar Sysvinit−2.84 se necesita lo siguiente:

bash: sh
binutils: as, ld
fileutils: chown, cp, install, ln, mknod, rm
gcc: cc, cc1, collect2, cpp0
make: make
sed: sed

Tar

Localización oficial para descarga

Tar (1.13):
ftp://ftp.gnu.org/gnu/tar/

Tar Patch (1.13):
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/
http://ftp.linuxfromscratch.org/lfs−packages/3.3/

Contenido de tar−1.13

Programas

rmt y tar

Descripciones

rmt

rmt es un programa utilizado por los programas dump y restore para manipular una unidad de cinta magnética mediante una conexión entre procesos.

tar

tar es un programa diseñado para almacenar y extraer ficheros de un archivo conocido como fichero tar.

Dependencias

Para instalar Tar−1.13 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, install, ls, mv, rm
gettext: msgfmt, xgettext
gcc: cc, cc1, collect2, cpp0, gcc
glibc: getconf
grep: egrep, fgrep, grep
m4: m4
make: make
gawk: gawk
net−tools: hostname
patch: patch
sed: sed
sh−utils: basename, echo, expr, sleep, uname
texinfo: install−info, makeinfo
textutils: cat, tr

Texinfo

Localización oficial para descarga

Texinfo (4.1):
ftp://ftp.gnu.org/gnu/texinfo/

Contenido de texinfo−4.0

Programas

info, install−info, makeinfo, texi2dvi y texindex

Descripciones

info

El programa info lee documentos Info, almacenados normalmente en el directorio /usr/share/info. Los documentos Info son como las páginas de manual, pero suelen contener algo más que una simple explicación de las opciones de un programa.

install−info

El programa install−info actualiza las entradas info. Cuando se ejecuta info, muestra una lista con los temas (es decir, los documentos info) disponibles. Pues bien, el programa install−info se usa para mantener esta

lista. Si los ficheros info son eliminados manualmente, también es necesario eliminar el tema en el fichero índice, para lo cual se utiliza este programa. También se utiliza para lo contrario, es decir, cuando se añaden documentos info.

makeinfo

El programa makeinfo convierte documentos fuente Texinfo a varios formatos. Los formatos disponibles son: ficheros info, texto plano y HTML.

texi2dvi

El programa texi2dvi imprime documentos Texinfo.

texindex

El programa texindex se usa para ordenar ficheros índice de Texinfo.

Dependencias

Para instalar Texinfo–4.0 se necesita lo siguiente:

bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, install, ln, ls, mkdir, mv, rm
gcc: cc1, collect2, cpp0, gcc
grep: egrep, fgrep, grep
make: make
sed: sed
sh–utils: basename, echo, expr, hostname, sleep
texinfo: makeinfo
textutils: cat, tr

Textutils

Localización oficial para descarga

Text Utils (2.0):
ftp://ftp.gnu.org/gnu/textutils/

Contenido de textutils–2.0

Programas

cat, cksum, comm, csplit, cut, expand, fmt, fold, head, join, md5sum, nl, od, paste, pr, ptx, sort, split, sum, tac, tail, tr, tsort, unexpand, uniq y wc

Descripciones

cat

cat concatena ficheros o la entrada estándar en la salida estándar.

cksum

cksum muestra la suma de comprobación CRC y cuenta los bytes de un fichero.

comm

comm compara dos ficheros ordenados línea por línea.

csplit

csplit escribe partes de un fichero separadas por un determinado patrón en ficheros xx01, xx02, etc, y muestra el número de bytes de cada parte en la salida estándar.

cut

cut imprime en la salida estándar las partes seleccionadas de las líneas de un fichero.

expand

expand convierte las tabulaciones de un fichero en espacios, escribiendo en la salida estándar

fmt

fmt reformatea cada párrafo de un fichero, escribiendo en la salida estándar.

fold

fold reajusta la longitud de las líneas de un fichero (por defecto, la entrada estándar), escribiendo en la salida estándar.

head

head imprime en la salida estándar las primeras xx (10 por defecto) líneas de un fichero.

join

join une líneas de dos ficheros en un campo común.

md5sum

md5sum muestra o chequea sumas de comprobación MD5.

nl

nl escribe un fichero en la salida estándar, añadiendo números de línea.

od

od escribe en la salida estándar una representación inequívoca (por defecto en octal) de un fichero.

paste

paste escribe en la entrada estándar líneas formadas por las líneas de cada uno de los ficheros especificados, separadas por tabulaciones.

pr

pr pagina o muestra en columnas el texto de un fichero, para imprimirlo posteriormente.

ptx

ptx genera un índice permutado de los contenidos de un fichero.

sort

sort escribe en la salida estándar una concatenación de ficheros ordenados.

split

split divide un fichero en partes de tamaño fijo llamadas FICHEROaa, FICHERObb,...

sum

sum muestra la suma de comprobación y el número de bloques que ocupa un fichero.

tac

tac escribe un fichero o ficheros en la salida estándar, comenzando por la última línea.

tail

tail imprime en la salida estándar las últimas xx (10 por defecto) líneas de un fichero.

tr

tr convierte, altera y/o borra caracteres de la entrada estándar, escribiendo en la salida estándar.

tsort

tsort escribe una lista totalmente ordenada de acuerdo con el orden parcial de los ficheros especificados.

unexpand

unexpand convierte los espacios de un fichero en tabulaciones, escribiendo en la salida estándar.

uniq

uniq elimina las líneas duplicadas de un fichero ordenado.

wc

wc muestra el número de líneas, palabras y bytes de un fichero, y una línea con el total si se ha especificado más de uno.

Dependencias

Para instalar Textutils−2.0 se necesita lo siguiente:

autoconf: autoconf, autoheader
automake: aclocal, automake
bash: sh
binutils: ar, as, ld, ranlib
diffutils: cmp
fileutils: chmod, install, ls, mv, rm
gettext: msgfmt, xgettext
gcc: cc, cc1, collect2, cpp0, gcc
glibc: getconf
grep: egrep, fgrep, grep
m4: m4
make: make
gawk: gawk
net−tools: hostname
perl: perl
sed: sed
sh−utils: basename, echo, expr, sleep, uname
tar: tar
texinfo: install−info, makeinfo
textutils: cat, tr

Util Linux

Localización oficial para descarga

Util Linux (2.11o):
ftp://ftp.win.tue.nl/pub/linux−local/utils/util−linux/

Contenido de util−linux−2.11n

Programas

agetty, arch, blockdev, cal, cfdisk, chkdupexe, col, colcrt, colrm, column, ctrlaltdel, cytune, ddate, dmesg, elvtune, fdformat, fdisk, fsck.minix, getopt, hexdump, hwclock, ipcrm, ipcs, isosize kill, line, logger, look, losetup, mcookie, mkfs, mkfs.bfs, mkfs.minix, mkswap, more, mount, namei, pivot_root, ramsize (enlace a rdev), raw, rdev, readprofile, rename, renice, rev, rootflags (enlace a rdev), script, setfdprm, setsid, setterm, sfdisk, swapoff (enlace a swapon), swapon, tunelp, ul, umount, vidmode, whereis y write

Descripciones

agetty

agetty abre un puerto de terminal, espera la introducción de un nombre de usuario, e invoca al comando /bin/login.

arch

arch muestra la arquitectura de la máquina.

blockdev

blockdev permite llamar a los controles de entrada/salida (ioctls) de los dispositivos de bloque desde la línea de comandos.

cal

cal muestra un calendario simple.

cfdisk

cfdisk es un manipulador de la tabla de particiones del disco basado en libncurses.

chkdupexe

chkdupexe encuentra ejecutables duplicados.

col

col filtra avances inversos de línea de la entrada.

colcrt

colcrt filtra la salida de nroff para su visualización en un CRT.

colrm

colrm elimina columnas de un fichero.

column

column muestra listas en columnas.

ctrlaltdel

ctrlaltdel establece la función de la combinación de teclas CTRL+ALT+DEL (reinicio duro o blando).

cytune

cytune afina los parámetros del controlador de Cyclades.

ddate

ddate convierte las fechas Gregorianas en fechas Discordantes.

dmesg

dmesg se usa para examinar o controlar el anillo de almacenamiento intermedio del núcleo (los mensajes de arranque del núcleo).

elvtune

elvtune permite afinar el nivel de entrada/salida por cola de un dispositivo de bloques.

fdformat

fdformat formatea un disquete a bajo nivel.

fdisk

fdisk es un manejador de la tabla de particiones.

fsck.minix

fsck.minix realiza una comprobación de consistencia en sistemas de ficheros MINIX de Linux.

getopt

getops analiza opciones de comandos de la misma forma que el comando getopt de C.

hexdump

hexdump muestra un fichero, o la entrada estándar, en un formato especificado por el usuario (ASCII, decimal, hexadecimal, octal).

hwclock

hwclock interroga y pone en hora el reloj del ordenador (también llamado RTC o reloj BIOS).

ipcrm

ipcrm elimina el recurso especificado.

ipcs

ipcs facilita información sobre las capacidades IPC.

isosize

isosize muestra el tamaño de un sistema de ficheros iso9660.

kill

kill envía la señal especificada al proceso indicado.

line

line copia una línea (hasta una nueva línea) de la entrada estándar y la escribe en la salida estándar.

logger

logger crea entradas en los registros del sistema.

look

look muestra líneas que comiencen con una cadena dada.

losetup

losetup activa y controla los dispositivos de bucle (loop).

mcookie

mcookie genera galletas mágicas (magic cookies) para xauth.

mkfs

mkfs construye un sistema de ficheros Linux en un dispositivo, normalmente una partición del disco duro.

mkfs.bfs

mkfs.bfs crea un sistema de ficheros bfs de SCO en un dispositivo, normalmente una partición del disco duro.

mkfs.minix

mkfs.minix crea un sistema de ficheros Minix en un dispositivo, normalmente una partición del disco duro

mkswap

mkswap configura un área de intercambio (swap) de Linux en un dispositivo o en un fichero.

more

more es un filtro para paginar texto pantalla a pantalla.

mount

mount monta un sistema de ficheros de un dispositivo en un directorio (punto de montaje).

namei

namei sigue el nombre de una ruta hasta que el punto terminal es encontrado.

pivot_root

pivot_root mueve el sistema de ficheros raíz del proceso actual.

ramsize

ramsize muestra y establece el tamaño del disco RAM.

raw

raw se usa para unir un dispositivo Linux de carácter a un dispositivo de bloque.

rdev

rdev muestra y establece el dispositivo raíz de la imagen, el dispositivo de intercambio, el tamaño del disco RAM, o el modo de vídeo.

readprofile

readprofile lee la información de los perfiles del núcleo.

rename

rename renombra ficheros.

renice

renice altera la prioridad de los procesos lanzados.

rev

rev invierte el orden de las líneas de un fichero.

rootflags

rootflags muestra y establece la información extra usada cuando se monta el sistema de ficheros raíz.

script

script registra todo lo que se ha tecleado en la sesión de terminal.

setfdprm

setfdprm establece los parámetros para los disquetes facilitados por el usuario

setsid

setsid lanza programas en una nueva sesión.

setterm

setterm establece los parámetros del terminal.

sfdisk

sfdisk es un manipulador de la tabla de particiones del disco.

swapoff

swapoff desactiva los dispositivos y ficheros de paginación e intercambio.

swapon

swapon activa los dispositivos y ficheros de paginación e intercambio.

tunelp

tunelp establece varios parámetros para el dispositivo de puerto paralelo (LP).

ul

ul lee un fichero y traduce las ocurrencias de marcas de texto a la secuencia que indica subrayado, para el terminal en uso.

umount

umount desmonta un sistema de ficheros montado.

vidmode

vidmode muestra y establece el modo de vídeo.

whereis

whereis localiza el binario, la fuente y la página de manual de un comando.

write

write envía un mensaje a otro usuario

Dependencias

Para instalar Util−linux−2.11n se necesita lo siguiente:

bash: sh
binutils: as, ld
diffutils: cmp
fileutils: chgrp, chmod, cp, install, ln, mv, rm
gettext: msgfmt, xgettext
gcc: cc, cc1, collect2, cpp, cpp0
glibc: rpcgen
grep: grep
make: make
sed: sed

sh−utils: uname, whoami
textutils: cat

Vim

Localización oficial para descarga

Vim (6.1):
ftp://ftp.vim.org/pub/editors/vim/unix/

Contenido de Vim−6.1

Programas

ex (enlace a vim), rview (enlace a vim), rvim (enlace a vim), vi (enlace a vim), view (enlace a vim), vim, vimdiff (enlace a vim), vimtutor (enlace a vim) and xxd

Descripciones

ex

ex arranca vim en modo Ex.

rview

rview es una versión restringida de view. No pueden ejecutarse comandos del intérprete de comandos y Vim no puede ser suspendido.

rvim

rvim es una versión restringida de vim. No pueden ejecutarse comandos del intérprete de comandos y Vim no puede ser suspendido.

vi

vi arranca vim en modo compatible con vi.

view

view arranca vim en modo de solo lectura.

vim

vim arranca vim de la manera normal.

vimdiff

vimdiff edita dos o tres versiones de un fichero con Vim y muestra las diferencias.

vimtutor

vimtutor arranca el tutorial de Vim.

xxd

xxd genera un volcado hexadecimal o hace lo contrario.

Dependencias

Para instalar Vim−6.0 se necesita lo siguiente:

bash: sh
binutils: as, ld, strip
diffutils: cmp, diff
fileutils: chmod, cp, ln, mkdir, mv, rm, touch
find: find
gcc: cc1, collect2, cpp0, gcc
grep: egrep, grep
make: make
net−tools: hostname
sed: sed
sh−utils: echo, expr, uname, whoami
textutils: cat, tr, wc

Apéndice B. Recursos

Introducción

A continuación, se da una lista de libros, COMOs y otros documentos que podría resultar útil descargar o comprar. Esta lista no es más que un pequeño ejemplo con el que empezar. Esperamos ser capaces de aumentar esta lista con el tiempo, tan pronto como descubramos documentos o libros más útiles.

Libros

- Linux Network Administrator's Guide (Guía del Administrador de Red Linux) publicado por O'Reilly. ISBN: 1−56502−087−2
- Running Linux (Ejecutando Linux) publicado por O'Reilly. ISBN: 1−56592−151−8

COMOs y Guías

Todos los siguientes COMOs pueden descargarse del proyecto LUCAS en http://lucas.hispalinux.es (la documentación en inglés se puede encontrar en http://www.linuxdoc.org)

- Guía de Administración de Redes (Linux Network Administrator's Guide)
- De−La−Puesta−En−Marcha−Al−Indicador−Del−Bash−CÓMO (ver Prólogo − A quién puede que no le interese leer el libro)

Otros

- Las distintas páginas del manual (man e info) que acompañan a los paquetes.