Предисловие

Мой путь к изучению и лучшему пониманию Linux начался еще в 1998 году. Я только что установил свой первый дистрибутив Linux и быстро увлекся его концепцией и философией.

У задачи может быть несколько вариантов решения. То же самое можно сказать и о дистрибутивах Linux. Многие из них существовали годами. Некоторые всё еще существуют, некоторые превратились во что-то иное, а некоторые остались только в наших воспоминаниях. Все они выполняют задачи по-разному, чтобы удовлетворить потребности своей целевой аудитории. И я понял - раз существует так много всевозможных способов добиться поставленной цели, мне больше не нужно ограничивать себя какой-то одной реализацией. До появления Linux мы просто мирились с проблемами в других операционных системах, поскольку у нас не было выбора. Что есть, то есть, нравилось нам это или нет. С Linux появился выбор. Если вам что-то не понравилось, вы можете изменить это, к тому же, это всецело поощряется.

Я попробовал разные дистрибутивы, но так и не смог ни на одном остановиться. Они были отличными системами сами по себе. Это больше не было вопросом правильно или неправильно. Это стало делом личного вкуса. При всём разнообразии выбора не было ни одного дистрибутива, который был для меня идеален. Поэтому я решил создать свою собственную Linux-систему, которая бы полностью соответствовала моим личным предпочтениям.

Чтобы создать свою собственную систему, я решил скомпилировать всё из исходного кода вместо использования предварительно скомпилированных пакетов. Эта «идеальная» Linux-система должна была иметь сильные стороны других систем без их недостатков. Сначала эта мысль казалась пугающей. Но я придерживался идеи, что такая система должна быть создана.

Разобравшись с такими проблемами, как циклические зависимости и ошибки во время компиляции, я, наконец, создал собственную систему Linux. Она была полностью работоспособна и вполне пригодна для использования, как и любая другая Linux-система того времени. Но это было мое собственное творение. Было очень приятно собрать такую систему самому. Единственное, что было бы лучше, это создавать каждую часть программного обеспечения самостоятельно. Это было следующее, к чему я стремился

Когда я поделился своими идеями и опытом с другими членами сообщества Linux, стал очевиден явный интерес к ним. Вскоре стало понятно, что такие специально созданные Linux-системы служат не только для удовлетворения специфических потребностей пользователей, но и являются идеальной возможностью для обучения программистов и системных администраторов, чтобы улучшить их (существующие) навыки работы с Linux. Так родился проект Linux From Scratch.

Книга Linux From Scratch является ядром этого проекта. В ней содержится информация и инструкции, необходимые для разработки и создания собственной системы. Хотя эта книга представляет шаблон, который позволит создать правильно работающую систему, вы можете изменить инструкции по своему усмотрению, что отчасти является важной частью этого проекта. Вы всё контролируете; мы просто протягиваем руку помощи, чтобы вы начали свой собственный путь.

Я искренне надеюсь, что вы прекрасно проведете время, работая над своей собственной

системой Linux From Scratch, и оцените ее многочисленные преимущества.

– Gerard Beekmans gerard@linuxfromscratch.org

Аудитория

Есть много причин, по которым вы хотели бы прочитать эту книгу. Один из вопросов, который задают многие люди, звучит так: «Зачем тратить время на сборку Linux-системы вручную с нуля, если можно просто загрузить и установить существующую?»

Одной из важных целей существования этого проекта является помощь в изучении того, как работает система Linux изнутри. Создание системы LFS помогает продемонстрировать, что заставляет работать Linux, как все работает вместе и зависит друг от друга. Одна из лучших вещей, которую может дать этот учебный опыт, — это возможность настроить систему Linux в соответствии с вашими уникальными потребностями.

Другое ключевое преимущество - LFS предоставляет более глубокий контроль, не полагаясь на чью-либо реализацию Linux. С LFS вы находитесь в кресле водителя, и Вы управляете каждым аспектом системы.

LFS позволяет создавать очень компактные системы Linux. При установке обычных дистрибутивов вам часто приходится устанавливать очень много программ, которые, вероятно, никогда не используются. Эти программы тратят ресурсы впустую. Вы можете возразить, что с сегодняшними жесткими дисками и процессорами такие ресурсы не имеют значения. Иногда, однако, вы все еще ограничены размером. Подумайте о загрузочных компакт-дисках, USB-накопителях и встраиваемых системах. Это области, в которых LFS может быть полезным.

Ещё одним преимуществом собственной сборки Linux является безопасность. При компиляции каждого компонента системы из исходных кодов вы можете всё проверить и применить необходимые патчи. Больше не нужно ждать, когда кто-то другой скомпилирует пакет с требуемыми исправлениями. Если вы не изучите патч и не примените его самостоятельно, нет гарантий, что новый пакет будет собран корректно и устранит проблему.

Цель Linux From Scratch — создать законченную и пригодную для использования систему базового уровня. Если вы не хотите создавать свою собственную систему Linux с нуля, вы, тем не менее, можете извлечь пользу из информации, содержащейся в этой книге.

Есть много других веских причин для создания собственной системы LFS. В конце концов, образование, безусловно, является самой важной из них. Продолжая работать с LFS, вы откроете для себя силу, которую действительно приносят информация и знания.

Целевые архитектуры LFS

Основными целевыми архитектурами LFS являются процессоры AMD/Intel x86 (32-разрядные) и x86_64 (64-разрядные). Однако, известно, что инструкции, приведенные в этой книге, с некоторыми изменениями работают с процессорами Power PC и ARM. Для создания системы, использующей один из этих процессоров, основным предварительным условием, в дополнение к описанным на следующей странице, является существующая система Linux, например,

собранная ранее LFS, Ubuntu, Red Hat/Fedora, SuSE или другой дистрибутив, ориентированный на имеющуюся у вас архитектуру. Также обратите внимание, что 32-разрядный дистрибутив можно установить и использовать в качестве хост-системы на 64-разрядном компьютере AMD/Intel.

При сборке LFS выигрыш от сборки на 64-битной системе по сравнению с 32-битной системой минимален. Например, в тестовой сборке LFS-9.1 в системе на базе процессора Core i7-4790 с использованием 4 ядер были получены следующие статистические данные:

Архитектура	Время сборки	Размер сборки
x86	239.9 минут	3.6 ГБ
x64	233.2 минут	4.4 ГБ

Как видите, на том же оборудовании 64-битная сборка всего на 3% быстрее и на 22% больше, чем 32-битная. Если вы планируете использовать LFS в качестве LAMP-сервера или брандмауэра, 32-разрядной сборки может быть достаточно. С другой стороны, для сборки и/или запуска некоторых пакетов в BLFS теперь требуется более 4 ГБ ОЗУ, поэтому, если вы планируете использовать LFS в качестве настольной ОС, авторы LFS рекомендуют собирать 64-битную систему.

По умолчанию 64-разрядная сборка LFS, считается «чистой» 64-разрядной системой. То есть она поддерживает только 64-разрядные исполняемые файлы. Сборка «multilib» системы требует компиляции многих программ дважды - один раз для 32-битной и один раз для 64-битной. Напрямую в книге данная опция не поддерживается, потому что это будет только мешать образовательной цели этой книги, предлагающей инструкции, необходимые для сборки базовой системы. Некоторые редакторы LFS/BLFS поддерживают ответвление LFS для multilib, которое доступно по адресу https://book.linuxfromscratch.ru/12.1-ml/systemd. Но это более продвинутая тема.

Предпосылки

Создание системы LFS — непростая задача. Это требует определенного уровня знаний системного администрирования Unix для решения проблем и правильного выполнения перечисленных команд. В частности, как абсолютный минимум, Вы должны уметь пользоваться командной оболочкой для копирования или перемещения файлов и каталогов, просмотра содержимого каталогов и файлов и изменения текущего каталога. Также ожидается, что у вас есть достаточные знания об использовании и установке программного обеспечения в Linux.

Поскольку книга LFS предполагает наличие хотя бы этого базового уровня навыков, различные форумы поддержки LFS вряд ли смогут предоставить вам большую помощь в этих вопросах. Вы обнаружите, что ваши вопросы, касающиеся таких базовых знаний, скорее всего, останутся без ответа (или вас просто направят к списку основных материалов для предварительного ознакомления).

Перед созданием системы LFS мы настоятельно рекомендуем прочитать следующие статьи:

- HOWTO по сборке программного обеспечения https://tldp.org/HOWTO/Software-Building-HOWTO.html

Это исчерпывающее руководство по сборке и установке «универсальных» программ Unix под Linux. Несмотря на то что руководство написано достаточно давно, оно по-прежнему дает хороший обзор основных методов, применяемых для сборки и установки программного обеспечения.

- Руководство для начинающих по установке из исходников https://moi.vonos.net/linux/beginners-installing-from-source/

В этом руководстве содержится хороший обзор основных навыков и методов, необходимых для сборки программ из исходного кода

LFS и стандарты

Структура LFS максимально соответствует стандартам Linux. Первичными стандартами являются:

- POSIX.1-2008.
- Filesystem Hierarchy Standard (FHS) Version 3.0
- Linux Standard Base (LSB) Version 5.0 (2015)

LSB имеет четыре отдельных стандарта: Core, Desktop, Runtime Languages и Imaging. Некоторые части спецификаций Core и Desktop зависят от архитектуры. Есть также две области не являющиеся обязательными: Gtk3 и Graphics. LFS старается соответствовать стандартам LSB, для архитектур IA32 (32-bit x86) или AMD64 (x86_64), рассмотренных в предыдущем разделе.

> Примечание Многие не согласны с требованиями LSB. Основные цели стандартов - быть уверенным в том, что проприетарное ПО будет правильно установлено и сможет корректно работать на совместимой системе. Поскольку в LFS установка программ идёт из исходных кодов, у пользователя имеется полный контроль над тем, какие пакеты ему необходимы, вы можете не устанавливать некоторые пакеты, определяемые в LSB.

Создать законченную систему, которая пройдет сертификационные тесты LSB «с нуля», возможно, но этого нельзя сделать без установки множества дополнительных пакетов, которые выходят за рамки этой книги. Однако, инструкции по их установке можно найти в книге BLFS.

Пакеты LFS, которые необходимы для удовлетворения требований LSB

LSB Core:	Bash, Bc, Binutils, Coreutils, Diffutils, File, Findutils, Gawk, Grep, Gzip, M4, Man-DB, Ncurses, Procps, Psmisc, Sed, Shadow, Tar, Util-linux, Zlib
LSB Desktop:	Нет
LSB Runtime Languages:	Perl, Python
LSB Imaging:	Нет
LSB Gtk3 и LSB Graphics (Необязательные):	Нет

Пакеты, поставляемые BLFS, необходимые для удовлетворения требований LSB

LSB Core:	At, Batch (часть At), Cpio, Ed, Fcrontab, LSB-Tools, NSPR, NSS, PAM, Pax, Sendmail (или Postfix, или Exim), time
LSB Desktop:	Alsa, ATK, Cairo, Desktop-file-utils, Freetype, Fontconfig, Gdk-pixbuf, Glib2, GTK+2, Icon-naming-utils, Libjpeg-turbo, Libpng, Libtiff, Libxml2, MesaLib, Pango, Xdg-utils, Xorg
LSB Runtime Languages:	Libxml2, Libxslt
LSB Imaging:	CUPS, Cups-filters, Ghostscript, SANE
LSB Gtk3 и LSB Graphics (Необязательные):	GTK3+

Пакеты, не поставляемые LFS или BLFS, необходимые для удовлетворения требований LSB

LSB Core:	Нет
LSB Desktop:	Qt4 (но предоставляется Qt5)
LSB Runtime Languages:	Нет
LSB Imaging:	Нет
LSB Gtk3 и LSB Graphics (Необязательные):	Нет

Информация о пакетах

Целью LFS является создание законченной и пригодной для использования базовой системы, которая содержит все пакеты, необходимые для её функционирования, состоящую при этом из относительно небольшого набора программ, и возможности которой можно расширять в зависимости от потребностей пользователя. Это не означает, что LFS является самой маленькой из возможных систем. В систему включено несколько важных пакетов, которые не являются обязательными. Приведенный ниже список объясняет почему в книгу включен тот или иной пакет.

Acl	Access Control List или ACL — список управления доступом, который определяет, кто или что может получать доступ к объекту (программе, процессу или файлу), и какие именно операции разрешено или запрещено выполнять субъекту (пользователю, группе пользователей).Данный пакет содержит утилиты для администрирования списков управления доступом, которые используются для определения дискреционных прав доступа к файлам и каталогам.
Attr	Этот пакет содержит программы для управления расширенными атрибутами объектов файловой системы.
Autoconf	Этот пакет содержит программы для создания сценариев оболочки, которые могут выполнять автоматическую настройку исходного кода из шаблона разработчика. Он часто необходим для повторной компиляции пакета после обновления процедур сборки.
Automake	Этот пакет содержит программы для создания Make-файлов из шаблона. Он также необходим для повторной компиляции пакета после обновления процедур сборки.
Bash	Этот пакет удовлетворяет требования LSB по предоставлению интерфейса Bourne Shell для системы. Он был выбран среди других пакетов оболочки из-за его повсеместного использования и широких возможностей.
Bc	Этот пакет предоставляет язык числовой обработки произвольной точности. Он необходимым для сборки ядра Linux

Binutils	Этот пакет содержит компоновщик, ассемблер и другие инструменты для работы с объектными файлами. Программы в этом пакете необходимы для компиляции большинства пакетов в системе LFS.
Bison	Этот пакет содержит GNU-версию yacc (Yet Another Compiler Compiler), необходимого для сборки некоторых пакетов в LFS.
Bzip2	Этот пакет содержит программы для сжатия и распаковки файлов. Используется для распаковки множества пакетов LFS.
Check	Этот пакет содержит тестовую обвязку для других программ.
Coreutils	Этот пакет содержит ряд программ для просмотра файлов и каталогов, и управления ими. Эти программы необходимы для управления файлами через командную строку и для сборки каждого пакета в LFS.
D-Bus	Этот пакет содержит программы для реализации системы межпроцессного взаимодействия - простого способа взаимодействия приложений друг с другом.
DejaGNU	Этот пакет предоставляет фреймворк для тестирования других программ.
Diffutils	Этот пакет содержит программы, которые показывают различия между файлами или каталогами. Их можно использовать для создания патчей, а также они применяются во многих процедурах сборки
E2fsprogs	Этот пакет содержит утилиты для работы с файловыми системами ext2, ext3 и ext4. Это наиболее распространенные и тщательно протестированные файловые системы, поддерживаемые Linux
Expat	Этот пакет содержит небольшую библиотеку разбора XML.Она необходима для модуля Perl XML::Parser.
Expect	Этот пакет содержит инструменты для автоматизации и тестирования, и является расширением к скриптовому языку Tcl, для многих интерактивных приложений. Он обычно используется для тестирования других пакетов.
File	Этот пакет содержит утилиту для определения типа файла или файлов. Некоторым пакетам она нужна в сценариях сборки.
Findutils	Этот пакет предоставляет программы для поиска файлов. Он используется во многих сценариях сборки пакетов.
Flex	Этот пакет содержит утилиту для генерации программ, распознающих шаблоны в тексте. Это версия GNU программы lex (лексический анализатор). Пакет необходим для сборки некоторых пакетов LFS.
Gawk	Этот пакет содержит программы для работы с текстовыми файлами. Это GNU версия awk (Aho-Weinberg-Kernighan). Он используется во многих сценариях сборки пакетов.
GCC	Это коллекция компиляторов Gnu. Он содержит компиляторы C и C++, а также несколько других компиляторов, поддержка которых не предусмотрена в LFS.
GDBM	Этот пакет содержит библиотеку GNU Database Manager. Он используется пакетом Man-DB
Gettext	Этот пакет содержит утилиты и библиотеки для интернационализации и локализации многочисленных пакетов.
Glibc	Этот пакет содержит основную библиотеку C. Программы Linux не будут работать без неё.
GMP	Этот пакет содержит математические библиотеки, предоставляющие полезные функции для вычислений с плавающей точкой. Требуется для сборки GCC.
Gperf	Этот пакет содержит программу, которая генерирует идеальную хеш-функцию из набора ключей. Необходим для пакета Systemd.
Grep	Этот пакет содержит программы для поиска по файлам. Пакет используется в скриптах сборки большинства пакетов.
Groff	Этот пакет содержит программы для обработки и форматирования текста. Одной из важнейших функций этих программ является форматирование man страниц.

GRUB	Это загрузчик операционной системы (GRand Unified Bootloader). Самый гибкий из нескольких доступных загрузчиков.
Gzip	Этот пакет содержит программы для сжатия и распаковки файлов. Он необходим для распаковки множества пакетов в LFS.
Iana-etc	Этот пакет предоставляет данные для сетевых служб и протоколов. Он необходим для обеспечения правильных сетевых возможностей.
Inetutils	Этот пакет содержит программы для базового сетевого администрирования.
Intltool	Этот пакет содержит инструменты для извлечения переводимых строк из исходных файлов.
IProute2	Этот пакет содержит программы для базовой и расширенной работы в сетях IPv4 и IPv6. Он был выбран среди других распространенных пакетов сетевых инструментов (net-tools) из-за его поддержки IPv6.
Jinja2	Этот пакет представляет собой модуль Python для создания текстовых шаблонов. Требуется для сборки Systemd.
Kbd	Этот пакет содержит таблицы раскладок, утилиты управления клавиатурой для неамериканских клавиатур, кроме этого, с ним поставляется большой набор консольных шрифтов.
Kmod	Этот пакет содержит программы, необходимые для администрирования модулей ядра Linux.
Less	Этот пакет содержит очень хороший просмотрщик текстовых файлов, который позволяет использовать прокрутку верх/вниз при просмотре. Многие пакеты используют его для постраничного вывода.
Libcap	Этот пакет реализует интерфейсы пользовательского пространства для возможностей POSIX 1003.1e, доступных в ядре Linux.
Libelf	Проект elfutils предоставляет библиотеки и инструменты для файлов ELF и данных DWARF. Большинство утилит в этом пакете доступны в других пакетах, но эта библиотека необходима для сборки ядра Linux с использованием стандартной (и наиболее эффективной) конфигурации.
Libffi	Этот пакет реализует переносимый программный интерфейс высокого уровня для различных соглашений о вызовах. Некоторые программы могут не знать во время компиляции, какие аргументы должны быть переданы в функцию. Например, интерпретатору во время выполнения может быть сообщено о количестве и типах аргументов, используемых для вызова данной функции. Libffi можно использовать как мост от интерпретатора к скомпилированному коду.
Libpipeline	Пакет Libpipeline содержит библиотеку для гибкого и удобного управления конвейерами подпроцессов. Она необходима для Man-DB.
Libtool	Этот пакет содержит сценарий поддержки универсальной библиотеки GNU. Он объединяет сложность использования общих библиотек в согласованный переносимый интерфейс. Библиотека необходима наборам тестов в других пакетах LFS.
Libxcrypt	Этот пакет предоставляет библиотеку libcrypt, необходимую различным пакетам (в частности, Shadow) для хеширования паролей. Он заменяет устаревшую реализацию libcrypt в Glibc.
Linux Kernel	Этот пакет является ядром операционной системой.
M4	Этот пакет содержит текстовый макропроцессор, полезный в качестве инструмента сборки для других программ.
Make	Этот пакет содержит программу для управления сборкой пакетов. При сборке она необходима почти для каждого пакета в LFS.
MarkupSafe	Этот пакет представляет собой модуль Python для безопасной обработки строк в HTML/XHTML/XML. Необходим для Jinja2

Man-DB	Этот пакет содержит программы для поиска и просмотра справочных страниц. Он был выбран вместо пакета man из-за превосходных возможностей интернационализации. Содержит man.
Man-pages	Этот пакет представляет собой содержимое основных справочных страниц Linux.
Meson	Этот пакет предоставляет программный инструмент для автоматизации создания программного обеспечения. Основная цель Meson — свести к минимуму количество времени, которое разработчики программного обеспечения должны тратить на настройку своей системы сборки. Требуется для сборки Systemd, а также многих пакетов BLFS.
MPC	Этот пакет содержит функции для арифметики комплексных чисел. Необходим GCC.
MPFR	Этот пакет содержит функции для арифметики с произвольной точностью. Необходим GCC.
Ninja	Этот пакет предоставляет небольшую систему сборки, ориентированную на скорость. Он предназначен для того, чтобы его входные файлы генерировались системой сборки более высокого уровня, и для максимально быстрого запуска сборок. Необходим для Meson.
Ncurses	Этот пакет содержит библиотеки для независимой от терминала обработки символьных экранов. Он часто используется для управления курсором в меню. Необходим ряду пакетов в LFS.
Openssl	Этот пакет содержит инструменты управления и библиотеки, относящиеся к криптографии. Они предоставляют криптографические функций другим пакетам, включая ядро Linux.
Patch	Этот пакет содержит программу для изменения или создания файлов путем применения файла patch, обычно создаваемого программой diff. Он необходим процедуре сборки для некоторых пакетов LFS.
Perl	Этот пакет является интерпретатором языка PERL. Он необходим для установки и тестирования некоторых пакетов LFS.
Pkgconf	Этот пакет содержит программу, которая помогает настраивать флаги компилятора и компоновщика для библиотек разработки. Программа может быть использована в качестве замены pkg-config, который необходим системе сборки многих пакетов. Он поддерживается более активно и развивается немного быстрее, чем оригинальный пакет Pkg-config.
Procps-NG	Этот пакет содержит программы для мониторинга процессов. Набор полезен для системного администрирования, а также используются загрузочными сценариями LFS.
Psmisc	Этот пакет содержит программы для отображения информации о запущенных процессах. Этот набор программ полезен для системного администрирования.
Python 3	Этот пакет предоставляет интерпретируемый язык программирования, философия которого делает упор на удобочитаемость кода.
Readline	Этот пакет представляет собой набор библиотек, предлагающих возможности редактирования командной строки и средства для работы с историей команд. Используется командным интерпретатором Bash.
Sed	Этот пакет позволяет редактировать текст, не открывая его в текстовом редакторе. Он необходим сценариям настройки многих пакетов LFS.
Shadow	Этот пакет содержит программы для безопасной обработки паролей.
Systemd	Этот пакет предоставляет систему инициализации init и ряд дополнительных возможностей загрузки и управления системой в качестве альтернативы Sysvinit. Он используется многими дистрибутивами.
Tar	Этот пакет предоставляет возможность архивирования и извлечения практически всех пакетов, используемых в LFS.

Tcl	Этот пакет содержит командный язык инструментов, используется во многих наборах тестов.
Texinfo	Этот пакет предоставляет программы для чтения, записи и преобразования информационных страниц. Используется в процедурах установки многих пакетов LFS.
Util-linux	Этот пакет содержит различные служебные программы. Среди них утилиты для работы с файловыми системами, консолями, разделами и сообщениями.
Vim	Этот пакет содержит редактор. Его выбрали из-за совместимости с классическим редактором vi и огромного количества возможностей. Редактор является очень личным выбором для каждого пользователя. По желанию можно заменить любым другим редактором.
Wheel	Этот пакет содержит модуль Python, который представляет собой эталонную реализацию механизма упаковки Python.
XML::Parser	Этот пакет представляет собой модуль Perl, который взаимодействует с Expat.
XZ Utils	Этот пакет содержит программы для сжатия и распаковки файлов. Он обеспечивает высокое сжатие и используется для распаковки пакетов в формате XZ или LZMA.
Zlib	Этот пакет содержит процедуры сжатия и распаковки, используемые некоторыми программами.
Zstd	Этот пакет содержит процедуры сжатия и распаковки, используемые некоторыми программами. Он обеспечивает высокие коэффициенты сжатия и очень широкий диапазон компромиссов между сжатием и скоростью.

Оформление

Чтобы облегчить понимание, в этой книге используются условные обозначения. Этот раздел содержит примеры оформления, используемые в Linux From Scratch.

```
./configure --prefix=/usr
```

Такое оформление предназначено для ввода именно так, как показано, если иное не сказано в тексте рядом. Это оформление также используется в разделах пояснений, чтобы указать, на какую команду ссылается.

В некоторых случаях логическая строка расширяется до двух или более физических строк с обратной косой чертой в конце строки.

```
CC="gcc -B/usr/bin/" ../binutils-2.18/configure \
  --prefix=/tools --disable-nls --disable-werror
```

Обратите внимание, что за обратной косой чертой должен следовать перевод строки. Другие символы, такие как пробелы или символы табуляции, приведут к неправильным результатам.

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

Такое оформление (текст фиксированной ширины) показывает вывод на экран, как правило, в результате выполнения команд. Этот формат также используется для отображения имен файлов, таких как /etc/ld.so.conf.

Примечание Пожалуйста, настройте свой браузер для отображения текста фиксированной ширины с хорошим моноширинным шрифтом, с помощью которого вы сможете четко различать символы Il1 или O0.

Акцент

Эта форма текста используется в книге для нескольких целей. Его основная цель — подчеркнуть важные моменты.

https://mirror.linuxfromscratch.ru/

Этот формат используется для гиперссылок как на сайт сообщества LFS, так и на внешние ресурсы. Может включать справочную информацию, места загрузки и веб-сайты.

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

Этот формат используется при создании файлов конфигурации. Первая команда указывает системе создать файл $LFS/etc/group из всего, что введено далее, пока не встретится последовательность End Of File (EOF). Поэтому весь этот раздел обычно печатается как есть.

<ЗАМЕНЯЕМЫЙ ТЕКСТ>

Этот формат используется для текста, который не должен быть напечатан так, как отображается, или для операций копирования и вставки.

[НЕОБЯЗАТЕЛЬНЫЙ ТЕКСТ]

Этот формат используется для текста, который является необязательным.

passwd(5)

Этот формат используется для ссылки на определенную страницу руководства (man). Число в скобках указывает на конкретный раздел внутри руководств. Например, у passwd есть две справочные страницы. В соответствии с инструкциями по установке LFS эти две справочные страницы будут расположены в */usr/share/man/man1/passwd.1* и */usr/share/man/man5/passwd.5*. Когда в книге используется passwd(5), имеется в виду конкретно */usr/share/man/man5/passwd.5. man passwd* напечатает первую найденную справочную страницу, совпадающую с «passwd» - это будет */usr/share/man/man1/passwd.1*. В этом примере вам нужно будет запустить man 5 passwd, чтобы прочитать указанную страницу. Обратите внимание, что большинство справочных страниц не имеют повторяющихся страниц в разных разделах. Поэтому обычно достаточно man <имя программы>. В книге LFS ссылки на справочные страницы также являются гиперссылками, поэтому нажатие на такую ссылку откроет справочную страницу, в формате HTML, со страниц руководства Arch Linux.

Структура

Эта книга разделена на несколько частей.

Часть I - Введение
Эта часть содержит важные замечания о том, как выполнить установку LFS. Также здесь представлена метаинформация о книге

Часть II - Подготовка к сборке
Часть II описывает, как подготовиться к процессу сборки — создать разделы, загрузить пакеты и выполнить компиляцию временных инструментов.

Часть III - Создание кросс-тулчейна LFS и временных инструментов
Часть III содержит инструкции по созданию инструментов, необходимых для создания конечной системы LFS.

Часть IV - Сборка системы LFS
Часть IV проводит читателя через сборку системы LFS—компиляцию и установку всех пакетов один за другим, настройку сценариев загрузки и установку ядра. Полученная в результате система Linux является основой, на которой можно собрать другое программное обеспечение для расширения системы по желанию. В конце этой книги есть простой в использовании справочник со списком всех программ, библиотек и важных файлов, которые были установлены.

Часть V - Приложения
Часть V содержит информацию о самой книге, включая акронимы и термины, благодарности, зависимости пакетов, список загрузочных сценариев LFS, лицензии на распространение книги и исчерпывающий указатель пакетов, программ, библиотек и сценариев.

Ошибки и рекомендации по безопасности

Программное обеспечение, используемое для создания системы LFS, постоянно обновляется и совершенствуется. Предупреждения безопасности и исправления ошибок могут появиться после выхода книги LFS. Чтобы проверить, нуждаются ли пакеты или инструкции в этом выпуске LFS в каких-либо изменениях для устранения уязвимостей в системе безопасности или исправления других ошибок, посетите https://mirror.linuxfromscratch.ru/lfs/errata/12.1-systemd/, прежде чем приступить к сборке. Вы должны внести требуемые изменения и применить их к соответствующему разделу книги по мере сборки системы LFS.

Кроме того, редакторы Linux From Scratch ведут список уязвимостей безопасности, обнаруженных после выхода книги. Чтобы проверить наличие каких-либо известных уязвимостей безопасности, посетите https://mirror.linuxfromscratch.ru/lfs/advisories/, прежде чем продолжить сборку. И, если вы будете использовать систему LFS в качестве реальной настольной или серверной системы, вам следует обращаться к рекомендациям и устранять любые уязвимости в системе безопасности, даже когда система LFS полностью собрана.

chapter01

From:
https://wwoss.ru/ - **worldwide open-source software**

Permanent link:
https://wwoss.ru/doku.php?id=software:linux_server:lfs-example:foreword

Last update: **2024/07/18 17:17**