



ОГЛАВЛЕНИЕ

Предисловие	11
Аудитория	12
Целевая архитектура проекта LFS	14
Проект LFS и стандарты	15
Обоснование использования пакетов, которые приведены в книге	17
Предварительные требования.....	24
Требование к хост-системе	25
Типографские соглашения	27
Структура книги	29
Возможные проблемы.....	30
ЧАСТЬ I.	
Введение	31
Глава 1. Введение.....	32
1.1. Как собрать систему LFS.....	32
1.2. Что изменилось по сравнению с предыдущей версией?	34
1.3. Список изменений.....	35
1.4. Ресурсы.....	43
1.5. Помощь	44
ЧАСТЬ II.	
Подготовка к сборке.....	47
Глава 2. Подготовка нового раздела	48
2.1. Введение.....	48
2.2. Создание нового раздела	48
2.3. Создание в разделе файловой системы	51
2.4. Монтирование нового раздела	53
Глава 3. Пакеты и патчи	55
3.1. Введение.....	55
3.2. Все пакеты.....	56
3.3. Необходимые патчи.....	64
Глава 4. Последние приготовления	66
4.1. О переменной \$LFS	66
4.2. Создание каталога \$LFS/tools.....	66

4.3. Добавление пользователя LFS.....	67
4.4. Настройка среды окружения.....	69
4.5. О единице измерения времени SBU	71
4.6. О наборах тестов	72
Глава 5. Создание временной версии системы	74
5.1. Введение.....	74
5.2. Технические замечания об инструментальном наборе	74
5.3. Общие инструкции, касающиеся компиляции.....	78
5.4. Пакет Binutils-2.23.1 – первый проход.....	80
5.5. Пакет GCC-4.7.2 – первый проход.....	82
5.6. Заголовки Linux-2.6.37 Linux-3.8.1 API.....	86
5.7. Пакет Glibc-2.17.....	87
5.8. Пакет Binutils-2.23.1 – второй проход.....	90
5.9. Пакет GCC-4.7.2 – второй проход	92
5.10. Пакет Tcl-8.6.0	96
5.11. Пакет Expect-5.45	97
5.12. Пакет DejaGNU-1.5.....	99
5.13. Пакет Check-0.9.9.....	100
5.14. Пакет Ncurses-5.9.....	101
5.15. Пакет Bash-4.2.....	102
5.16. Пакет Bzip2-1.0.6.....	103
5.17. Пакет Coreutils-8.21	103
5.18. Пакет Diffutils-3.2	104
5.19. Пакет File-5.13	105
5.20. Пакет Findutils-4.4.2	105
5.21. Пакет Gawk-4.0.2.....	106
5.22. Пакет Gettext-0.18.2.....	107
5.23. Пакет Grep-2.14	108
5.24. Пакет Gzip-1.5.....	108
5.25. Пакет M4-1.4.16.....	109
5.26. Пакет Make-3.82	110
5.27. Пакет Patch-2.7.1	110
5.28. Пакет Perl-5.16.2.....	111
5.29. Пакет Sed-4.2.2.....	112
5.30. Пакет Tar-1.26	112
5.31. Пакет Texinfo-5.0.....	113
5.32. Пакет Xz-5.0.4	114
5.33. Удаление ненужных файлов	114
5.34. Смена владельца	115
ЧАСТЬ III.	
Сборка системы LFS.....	117
Глава 6. Установка программ базовой системы.....	118

6.1. Введение	118
6.2. Подготовка виртуальных файловых систем ядра.....	119
6.3. Управление пакетами	121
6.4. Переход в среду chroot	126
6.5. Создание каталогов.....	127
6.6. Создание важных файлов и символических ссылок	129
6.7. Заголовки Linux-3.8.1 API.....	131
6.8. Пакет Man-pages-3.47	132
6.9. Пакет Glibc-2.17	133
6.10. Перенастройка инструментального набора	143
6.11. Пакет Zlib-1.2.7	145
6.12. Пакет File-5.13	146
6.13. Пакет Binutils-2.23.1.....	147
6.14. Пакет GMP-5.1.1	150
6.15. Пакет MPFR-3.1.1.....	152
6.16. Пакет MPC-1.0.1	153
6.17. Пакет GCC-4.7.2.....	154
6.18. Пакет Sed-4.2.2.....	161
6.19. Пакет Bzip2-1.0.6	162
6.20. Пакет Pkg-config-0.28	164
6.21. Пакет Ncurses-5.9.....	165
6.22. Пакет Util-linux-2.22.2.....	168
6.23. Пакет Psmisc-22.20.....	174
6.24. Пакет Procps-ng-3.3.6.....	175
6.25. Пакет E2fsprogs-1.42.7.....	178
6.26. Пакет Shadow-4.1.5.1	182
6.27. Пакет Coreutils-8.21	187
6.28. Пакет Iana-Etc-2.30.....	195
6.29. Пакет M4-1.4.16.....	195
6.30. Пакет Bison-2.7	196
6.31. Пакет Grep-2.14	197
6.32. Пакет Readline-6.2	198
6.33. Пакет Bash-4.2.....	200
6.34. Пакет Libtool-2.4.2.....	201
6.35. Пакет GDBM-1.10.....	202
6.36. Пакет Inetutils-1.9.1	203
6.37. Пакет Perl-5.16.2.....	206
6.38. Пакет Autoconf-2.69	210
6.39. Пакет Automake-1.13.1	211
6.40. Пакет Diffutils-3.2	213
6.41. Пакет Gawk-4.0.2	214
6.42. Пакет Findutils-4.4.2	215
6.43. Пакет Flex-2.5.37.....	217
6.44. Пакет Gettext-0.18.2.....	218

6.45. Пакет Groff-1.22.2	221
6.46. Пакет Xz-5.0.4	225
6.47. Пакет GRUB-2.00	227
6.48. Пакет Less-451	229
6.49. Пакет Gzip-1.5	230
6.50. Пакет IPRoute2-3.8.0	231
6.51. Пакет Kbd-1.15.5	234
6.52. Пакет Kmod-12	237
6.53. Пакет Libpipeline-1.2.2	239
6.54. Пакет Make-3.82	240
6.55. Пакет Man-DB-2.6.3	241
6.56. Пакет Patch-2.7.1	244
6.57. Пакет Sysklogd-1.5	244
6.58. Пакет Sysvinit-2.88dsf	246
6.59. Пакет Tar-1.26	248
6.60. Пакет Texinfo-5.0	249
6.61. Пакет Udev-197 (Берем из пакета systemd-197)	251
6.62. Пакет Vim-7.3	253
6.63. Об отладочных символах	257
6.64. Снова стираем все ненужное	257
6.65. Окончательное приведение системы в порядок	258
Глава 7. Настройка скриптов загрузки системы	260
7.1. Введение	260
7.2. Общие вопросы конфигурирования сети	261
7.3. Настройка файла /etc/hosts	266
7.4. Управление устройствами и модулями в системе LFS	267
7.5. Создание пользовательских символических ссылок на устройства	274
7.6. Пакет LFS-Bootscripts-20130123	277
7.7. Как работают скрипты загрузки системы	279
7.8. Конфигурирование имени хоста системы	283
7.9. Конфигурирование скрипта setclock	283
7.10. Конфигурирование консоли Linux	284
7.11. Конфигурирование скрипта sysklogd	288
7.12. Файл rc.site	289
7.13. Файлы запуска командной оболочки Bash	292
7.14. Создание файла /etc/inputrc	296
Глава 8. Делаем систему LFS загружаемой	298
8.1. Введение	298
8.2. Создание файла /etc/fstab	298
8.3. Пакет Linux-3.8.1	300
8.4. Использование загрузчика GRUB для настройки процесса загрузки	305

Глава 9. Завершение	308
9.1. Результат.....	308
9.2. Зарегистрируем систему.....	308
9.3. Перезагрузка системы.....	309
9.4. А что дальше?	311
ЧАСТЬ IV.	
Приложения	313
Приложение А. Сокращения и терминология	314
Приложение В. Благодарности	319
Приложение С. Зависимости	323
Приложение D. Загрузочные и конфигурационные скрипты версии 20130123.....	335
D.1. Скрипт /etc/rc.d/init.d/rc	335
D.2. Скрипт /lib/lsb/init-functions.....	340
D.3. Скрипт /etc/rc.d/init.d/functions	357
D.4. Скрипт /etc/rc.d/init.d/mountvirtfs	374
D.5. Скрипт /etc/rc.d/init.d/modules	375
D.6. Скрипт /etc/rc.d/init.d/udev.....	377
D.7. Скрипт /etc/rc.d/init.d/swap	379
D.8. Скрипт /etc/rc.d/init.d/setclock	380
D.9. Скрипт /etc/rc.d/init.d/checkfs	382
D.10. Скрипт /etc/rc.d/init.d/mountfs.....	385
D.11. Скрипт /etc/rc.d/init.d/udev_retry	387
D.12. Скрипт /etc/rc.d/init.d/cleanfs	388
D.13. Скрипт /etc/rc.d/init.d/console	391
D.14. Скрипт /etc/rc.d/init.d/localnet	394
D.15. Скрипт /etc/rc.d/init.d/sysctl	395
D.16. Скрипт /etc/rc.d/init.d/syslogd.....	396
D.17. Скрипт /etc/rc.d/init.d/network	398
D.18. Скрипт /etc/rc.d/init.d/sendsignals	400
D.19. Скрипт /etc/rc.d/init.d/reboot	401
D.20. Скрипт /etc/rc.d/init.d/halt	402
D.21. Скрипт /etc/rc.d/init.d/template.....	403
D.22. Скрипт /etc/sysconfig/modules.....	405
D.23. Скрипт /etc/sysconfig/createfiles	405
D.24. Скрипт /etc/sysconfig/udev-retry	406
D.25. Скрипт /sbin/ifup.....	406
D.26. Скрипт /sbin/ifdown	409
D.27. Скрипт /lib/services/ipv4-static	412
D.28. Скрипт /lib/services/ipv4-static-route.....	413

Приложение Е. Конфигурационные правила udev	416
Е.1. Правила 55-lfs.rules	416
Приложение F. Лицензии LFS	417
F.1. Лицензия Creative Commons	417
F.2. Лицензия MIT	426



ПРЕДИСЛОВИЕ

Мое турне по изучению и лучшему пониманию системы Linux началось более десяти лет назад, еще в 1998 году. Тогда я только что установил мой первый дистрибутив Linux и сразу был заинтригован общей концепцией и философией, стоящими за Linux.

Одну и ту же задачу всегда можно выполнить многими способами. То же самое можно сказать о дистрибутивах Linux. Очень многие из них существовали на протяжении ряда лет. Некоторые из них все еще существуют, некоторые превратились в нечто другое, другие же остались только в нашей памяти. Все они создавались по-разному с тем, чтобы удовлетворить потребности своей целевой аудитории. Поскольку существует очень много различных способов достижений одной и той же конечной цели, я понял, что больше не должен ограничиваться какой-либо одной реализацией. До тех пор, пока не появился Linux, мы просто мирились с проблемами других операционных систем, т. к. у нас не было выбора. Было то, что было, вне зависимости от того, нравится ли это или нет. С системой Linux появилась возможность выбора. Если вам что-то не нравится, вам разрешается и даже предлагается это изменить.

Я попробовал ряд дистрибутивов и не смог решить, какой из них выбрать. Это были замечательные системы со своими собственными ценными особенностями. Вопрос был не в том, что правильно или что нет. Это дело личного вкуса. Из-за всех этих возможностей выбора стало очевидным, что не удастся найти систему, которая была бы идеальной для меня. Поэтому я решил создать свою собственную систему Linux, которая бы в полной мере соответствовала моим личным предпочтениям.

Чтобы действительно сделать свою собственную систему, я решил собирать все из исходных кодов и не использовать предварительно скомпилированные двоичные пакеты. Эта «идеальная» система Linux могла бы обладать преимуществами различных систем и в ней бы отсутствовали их очевидные недостатки. Поначалу такая идея казалась довольно пугающей. Я продолжал считать, что такую систему собрать можно.

После того, как удалось справиться с такими вещами, как циклические зависимости или ошибки времени компиляции, я, наконец, собрал свою собственную версию системы Linux. Это была полностью функциональная система, и ей свободно можно было пользоваться, как и любой другой системой Linux, имеющейся на тот момент. Но это было мое собственное творение. Было очень приятно собрать такую систему самостоятельно. Единственное, что могло быть лучшим, это создать каждую часть программного обеспечения самостоятельно. Это было бы еще лучшим.

Когда я рассказал о своих целях и поделился опытом с другими членами сообщества Linux, стало очевидно, что к этим идеям есть устойчивый интерес. Очень быстро стало ясно, что такие специально собранные системы Linux не только годятся для удовлетворения конкретных требований пользователей, но также служат идеальным средством обучения, позволяющим программистам и системным администраторам повысить свои (уже существующие) навыки работы с Linux. Благодаря тому, что интерес к системе оказался более широким, появился проект «Linux From Scratch» («Linux с нуля»).

Данная книга «Linux с нуля» является центральным ядром, вокруг которого строится этот проект. В ней предлагаются объяснения и инструкции, необходимые для того, чтобы вы сконструировали и собрали свою собственную систему. Хотя в этой книге предлагается шаблон, который в результате выльется в корректно работающую систему, вы вправе изменить инструкции по своему усмотрению, что, в частности, является важной частью данного проекта. Управление остается за вами; мы просто протягиваем вам руку помощи с тем, чтобы вы начали на свое собственное турне.

Я искренне надеюсь, что вы с удовольствием проведете время за работой над вашей собственной системой Linux From Scratch и насладитесь многочисленными преимуществами обладания системой, которая по-настоящему является вашей собственной.

Жерар Бикманс
gerard@linuxfromscratch.org

Аудитория

Есть много причин, по которым вам бы хотелось прочитать эту книгу. Один из вопросов, который ставят перед собой многие — зачем проходить через все трудности ручной сборки системы Linux с нуля,

когда вы можете просто скачать и установить одну из существующих систем?

Одним из важных доводов в пользу существования этого проекта является то, что он поможет вам изнутри узнать, как работает система Linux. Процесс сборки системы LFS поможет продемонстрировать, что происходит в каждом такте работы Linux, как все это работает вместе и как отдельные части зависят друг от друга. И самое лучшее, что этот опыт изучения предоставит возможность настроить систему Linux в соответствии с вашими собственными индивидуальными потребностями.

Другое ключевое преимущество проекта LFS в том, что можно получить большую степень контроля над системой и не полагаться на чужую реализацию Linux. В системе LFS вы будете сидеть на месте водителя, и определять все аспекты работы системы.

Проект LFS позволит вам создавать очень компактные системы Linux. Когда вы устанавливаете обычные дистрибутивы, вы часто вынуждены устанавливать множество программ, которыми вы, вероятно, никогда не будете пользоваться и с которыми вы не будете разбираться. На эти программы бесполезно тратятся ресурсы. Вы можете утверждать, что с жесткими дисками и процессорами, которые есть в настоящее время, этими затратами можно пренебречь. Но даже если не найдется ничего другого, у вас иногда все еще могут возникать ограничения, связанные с размерами. Вспомните о загрузочных компакт дисках, флешках USB и встроенных системах. Это как раз те области, в которых LFS может оказаться полезным.

Еще одним преимуществом собственной сборки системы Linux является ее безопасность. Благодаря компиляции всей системы из исходного кода, у вас будет возможность все проверить и применить все нужные патчи безопасности. Больше не нужно ждать, чтобы кто-нибудь скомпилировал двоичные пакеты, в которых исправлена дыра безопасности. Если вы сами не проверите патч и его не примените, у вас не будет гарантии, что новый двоичный пакет собран правильно и адекватно решает проблему.

Цель проекта Linux From Scratch состоит в сборке полной и полезной системы базового уровня. Если у вас нет желания создавать свою собственную систему Linux с нуля, вы не сможете получить всех преимуществ от сведений, изложенных в этой книге.

Имеется слишком много других веских причин сборки вашей собственной системы LFS и их все невозможно здесь перечислить. В конечном счете, самой важной причиной на сегодняшний день является

образование. По мере того, как вы будете продолжать экспериментировать с системой LFS, вам будет открываться вся мощь, которую могут дать информация и знания.

Целевая архитектура проекта LFS

Первичной целевой архитектурой проекта LFS являются процессоры x86 (32-разрядные) и x86_64 (64-разрядные) AMD/Intel. С другой стороны, также известно, что инструкции из этой книги можно применять с некоторыми изменениями к процессорам Power PC. Для того, чтобы собрать систему, в которой используется один из таких процессоров, основным необходимым условием, кроме тех, что изложены на нескольких следующих страницах, является наличие системы Linux, например, ранее установленной системы LFS, Ubuntu, Red Hat/Fedora, SuSE, или другого дистрибутива, предназначенного для имеющейся у вас архитектуры. Также отметим, что на компьютерах, имеющих 64-разрядный процессор AMD/Intel, можно в качестве хост-системы устанавливать и использовать 32-разрядные дистрибутивы.

Здесь нужно сообщить о некоторых других дополнительных фактах, относящихся к 64-разрядным системам. По сравнению с 32-разрядными системами, размеры исполняемых программ чуть больше, а скорость их выполнения лишь немного быстрее. Например, при тестовой сборке системы LFS-6.5 на системе с процессором Core2Duo, были получены следующие статистические данные:

Архитектура	Время сборки	Размер сборки
32-разрядная	198.5 минут	648 МВ
64-разрядная	190.6 минут	709 МВ

Видно, что 64-разрядная сборка только на 4% быстрее и на 9% больше по размеру, чем 32-разрядная сборка. Преимущество от перехода на 64-разрядную систему сравнительно небольшая. Конечно, если у вас более 4 Гб оперативной памяти или вы хотите манипулировать данными, размер которых превышает 4 Гб, то преимущества 64-разрядной системы весьма существенны.

По умолчанию 64-разрядная сборка, которая получается в результате создания системы LFS, считается «чистой» 64-разрядной системой. То есть, она поддерживает работу только 64-разрядных исполняемых файлов. Для сборки «мультибиблиотечной» системы требуется двухразовая компиляции многих приложений, один раз –

для 32-разрядной системы и еще раз – для 64-разрядной системы. В проекте LFS это явно не поддерживается, поскольку противоречит образовательным целям приводимых инструкций, необходимых для достаточно простой базисной системы Linux. С этой более сложной темой вы можете ознакомиться в проекте Cross Linux From Scratch.

Есть еще одно последнее замечание, касающееся 64-разрядных систем. Есть некоторые старые пакеты, которые в настоящее время нельзя собрать в «чистой» 64-разрядной системе, либо для их сборки нужны специальные инструкции. Как правило, в этих пакетах используются специальные 32-разрядные инструкции языка ассемблера, которые не будут работать в 64-разрядной системе. К таким пакетам относятся некоторые драйвера Xorg для некоторых устаревших видеокарт, смотрите <http://xorg.freedesktop.org/releases/individual/driver/>. Большинство из этих проблем можно обойти, но могут потребоваться некоторые специальные процедуры или патчи.

Проект LFS и стандарты

Структура LFS соответствует стандартам Linux настолько строго, насколько это возможно. Первичными стандартами являются следующие:

- POSIX.1-2008.
- Стандарт иерархической файловой системы – Filesystem Hierarchy Standard (FHS).
- Спецификации стандартной базовой системы Linux – Linux Standard Base (LSB) Specifications.

Спецификации LSB состоят из пяти отдельных стандартов: Core (Ядро), C++, Desktop (Рабочий стол), Runtime Languages (Языки времени выполнения) и Printing (Печать). В дополнение к общим требованиям есть также конкретные требования к архитектуре. В LFS сделана попытка обеспечить соответствие требованиям архитектуры, которые обсуждались в предыдущем разделе.



Замечание

Многие не согласны с требованиями спецификаций LSB. Основное назначение этих требований – обеспечить, чтобы на совместимой системе можно было устанавливать и должным образом запускать проприетарное программное обеспечение. Поскольку система LFS создается из исходных кодов, пользователь осуществляет полный контроль над тем, какие нужны пакеты, и может не устанавливать некоторые пакеты, которые необходимы согласно спецификациям LSB.

Можно создать завершённую систему LFS, которая способна пройти сертификационные тесты LSB, но только не без дополнительных пакетов, которые выходят за рамки LFS. Инструкции по установке этих дополнительных пакетов имеются в проекте BLFS.

Пакеты, имеющиеся в системе LFS, которые необходимы для соответствия требованиям спецификации LSB

<i>Стандарт LSB Core (Ядро):</i>	Bash, Binutils, Coreutils, Diffutils, File, Findutils, Gawk, Grep, Gzip, M4, Man-DB, Ncurses, Procps, Psmisc, Sed, Shadow, Tar, Util-linux, Zlib
<i>Стандарт LSB C++:</i>	Gcc
<i>Стандарт LSB Desktop (Рабочий стол):</i>	Нет
<i>Стандарт LSB Runtime Languages (Языки времени выполнения):</i>	Perl
<i>Стандарт LSB Printing (Печать):</i>	Нет
<i>Стандарт LSB Multimedia (Мультимедиа):</i>	Нет

Пакеты, имеющиеся в системе BLFS, которые необходимы для соответствия требованиям спецификации LSB

<i>Стандарт LSB Core (Ядро):</i>	At, Batch (a part of At), Bc, Cpio, Ed, Fcfrontab, Initd-tools, Lsb_release, PAM, Sendmail (либо Postfix, либо Exim)
<i>Стандарт LSB C++:</i>	Нет
<i>Стандарт LSB Desktop (Рабочий стол):</i>	ATK, Cairo, Desktop-file-utils, Freetype, Fontconfig, Glib2, GTK+2, Icon-naming-utils, Libjpeg, Libpng, Libxml2, MesaLib, Pango, Qt3, Qt4, Xorg
<i>Стандарт LSB Runtime Languages (Языки времени выполнения):</i>	Python
<i>Стандарт LSB Printing (Печать):</i>	CUPS
<i>Стандарт LSB Multimedia (Мультимедиа):</i>	Библиотеки alsa, NSPR, NSS, OpenSSL, Java, Xdg-utils

Пакеты, отсутствующие в системах LFS и BLFS, которые необходимы для соответствия требованиям спецификации LSB

<i>Стандарт LSB Core (Ядро):</i>	Нет
<i>Стандарт LSB C++:</i>	Нет
<i>Стандарт LSB Desktop (Рабочий стол):</i>	Нет
<i>Стандарт LSB Runtime Languages (Языки времени выполнения):</i>	Нет
<i>Стандарт LSB Printing (Печать):</i>	Нет
<i>Стандарт LSB Multimedea (Мультимедиа):</i>	Нет

Обоснование использования пакетов, которые приведены в книге

Как отмечалось ранее, целью проекта LFS является создание полной и полезной системы базового уровня. В ее состав входят все пакеты, которые нужны для тиражирования системы, причем в их сравнительно минимальном базовом количестве, на базе которых пользователь может по собственному выбору создать более полную систему. Это не значит, что LFS является самой маленькой из возможных систем. В ее состав входят важные пакеты, которые не являются строго необходимыми. В приведенном ниже списке дается обоснование использования каждого пакета, имеющегося в данной книге.

- **Autoconf**

В этом пакете находятся программы создания скриптов командных оболочек, с помощью которых можно по шаблону, предоставляемому разработчиками, автоматически конфигурировать исходный код. Часто необходим в процедурах сборки для того, чтобы пересобрать пакет после его обновления.

- **Automake**

В этом пакете находятся программы для генерации файлов Make в соответствии с шаблоном. Часто необходим в процедурах сборки для пересборки пакета после его обновления.

- **Bash**

С помощью этого пакета в системе реализуется интерфейс командной оболочки Bourne Shell в соответствии со стандартом LSB Core (Ядро). Эта командная оболочка выбрана среди других из-за того, что обычно используется именно она и

что в ней есть дополнительные возможности, выходящие за рамки базовых функциональных возможностей командных оболочек.

- **Binutils**

В этом пакете находится компоновщик, ассемблер и другие инструментальные средства, предназначенные для работы с объектными файлами. Программы, имеющиеся в данном пакете, необходимы для компиляции большинства пакетов, как входящих в состав системы LFS, так и выходящих за ее рамки.

- **Bison**

В этом пакете находится версия GNU компилятора yacc (Yet Another Compiler Compiler – Еще Один Компилятор Компиляторов), необходимого для сборки некоторых других программ LFS.

- **Bzip2**

В этом пакете находятся программы, используемые для сжатия и распаковки файлов. Необходим для распаковки многих пакетов LFS.

- **Check**

В этом пакете находится инструментарий для тестирования других программ. Устанавливается только во временном наборе инструментальных средств.

- **Coreutils**

В этом пакете находится ряд важных программ, необходимых для просмотра файлов и каталогов и для работы с ними. Эти программы необходимы для управления файлами из командной строки; также необходимы в процедурах установки каждого пакета LFS.

- **DejaGNU**

В этом пакете находится фреймворк тестирования других программ. Устанавливается только во временном наборе инструментальных средств.

- **Diffutils**

В этом пакете находятся программы, с помощью которых выявляются различия в файлах или каталогах. Эти программы можно использовать для создания патчей; они также используются в процедурах сборки многих пакетов.

- **E2fsprogs**

В этом пакете находятся утилиты для работы с файловыми системами ext2, ext3 и ext4. Это чаще всего используемые и

тщательно протестированные файловые системы, которые поддерживаются в Linux.

- **Expect**
В этом пакете находится программа, реализующая скриптовые диалоги с другими интерактивными программами. Обычно используется для тестирования других пакетов. Устанавливается только во временном наборе инструментальных средств.
- **File**
В этом пакете находится утилита определения типа указанного файла или файлов. Утилита нужна для сборки нескольких пакетов.
- **Findutils**
В этом пакете находятся программы поиска файлов в файловой системе. Используется в скриптах сборки многих пакетов.
- **Flex**
В этом пакете находится утилита генерации программ, которые могут обрабатывать шаблоны, находящиеся в тексте. Это версия GNU программы lex (лексического анализатора). Требуется при сборке нескольких пакетов LFS.
- **Gawk**
В этом пакете находятся программы для работы с текстовыми файлами. Это версия GNU программы AWK (Aho-Weinberg-Kernighan – Ахо-Вайнберг-Керниган). Используется в скриптах сборки многих других пакетов.
- **Gcc**
Этот пакет является коллекцией компиляторов GNU. В нем находятся компиляторы языков C и C++, а также ряд других, которые не собираются в LFS.
- **GDBM**
В этом пакете находится библиотека менеджера баз данных GNU (GNU Database Manager). Он используется одним из пакетов LFS, а именно – Man-DB.
- **Gettext**
В этом пакете находятся утилиты и библиотеки интернационализации и локализации, используемые в огромном количестве других пакетов.
- **Glibc**
В этом пакете находится основная библиотека языка C. Без нее программы Linux работать не будут.

- **GMP**
В этом пакете находятся математические библиотеки, предоставляющие полезные арифметические функции для расчетов произвольной точности. Требуется для сборки Gcc.
- **Grep**
В этом пакете находятся программы поиска в файлах. Эти программы используются скриптами сборки большинства пакетов.
- **Groff**
В этом пакете находятся программы обработки и форматирования текста. Одной из важных функций этих программ является форматирование страниц man.
- **GRUB**
Этот пакет является начальным загрузчиком Grand Unified Boot Loader. Это один из нескольких имеющихся загрузчиков, но этот загрузчик наиболее гибкий.
- **Gzip**
В этом пакете находятся программы сжатия и распаковки файлов. Он необходим для распаковки многих пакетов, как входящих в состав системы LFS, так и выходящих за ее рамки.
- **Iana-etc**
В этом пакете находятся данные, используемые сетевыми сервисами и протоколами. Он необходим для правильной работы сетевых средств.
- **Inetutils**
В этом пакете находятся программы базового администрирования сети.
- **IProute2**
В этом пакете находятся программы базового и расширенного сетевых протоколов IPv4 и IPv6. Он был выбран среди других пакетов, обычно используемых в качестве сетевых инструментальных средств (net-tools), из-за того, что позволяет использовать протокол IPv6.
- **Kbd**
В этом пакете находятся основные файлы таблиц обработки клавиатуры, клавиатурные утилиты для неамериканских раскладок, а также ряд консольных шрифтов.
- **Kmod**
В этом пакете находятся программы, необходимые для управления модулями ядра Linux.

- **Less**
В этом пакете находится очень хороший просмотрщик текстовых файлов, который при просмотре файла позволяет выполнять прокрутку текста вверх или вниз. Он также используется в пакете Man-DB для просмотра страниц man.
- **Libpipeline**
В пакете Libpipeline находится библиотека для работы с конвейерами подпроцессов гибким и удобным образом. Этот пакет необходим для пакета Man-DB.
- **Libtool**
В этом пакете находится скрипт поддержки основных библиотек GNU. Он является оберткой, экранирующей сложность обращения к совместно используемым библиотекам и предоставляющей удобный системно-независимый интерфейс. Используется в тестовых наборах в других пакетах LFS.
- **Linux Kernel**
Этот пакет является операционной системой. Это – Linux в среде GNU/Linux.
- **M4**
В этом пакете находится универсальный текстовый макропроцессор, используемый как инструмент сборки других программ.
- **Make**
В этом пакете находится программа, управляющая сборкой пакетов. Он необходим почти для каждого пакета в LFS.
- **Man-DB**
В этом пакете находятся программы поиска и просмотра страниц man. Он был выбран вместо пакета man из-за его превосходных возможностей локализации. В этом пакете присутствует программа man.
- **Man-pages**
В этом пакете находится фактическое содержание основных страниц man системы Linux.
- **MPC**
В этом пакете находятся функции арифметики комплексных чисел. Он необходим для пакета Gcc.
- **MPFR**
В этом пакете находятся функции арифметики операций различной степени точности. Он необходим для пакета Gcc.
- **Ncurses**
В этом пакете находятся библиотеки обработки экранных символов, не зависящей от используемого терминала. Пакет часто

используется для управления курсором в системе меню. Он необходим в ряде пакетов системы LFS.

- **Patch**

В этом пакете находится программа изменения или создания файлов, использующая файл *patch*, обычно создаваемый с помощью программы *diff*. Пакет необходим для процедур сборки нескольких пакетов LFS.

- **Perl**

Этот пакет является интерпретатором времени выполнения языка PERL. Он необходим для инсталляции и тестирования нескольких пакетов LFS.

- **Pkg-config**

В этом пакете находится программа, возвращающая метаданные, описывающие установленную библиотеку или пакет.

- **Procps-NG**

В этом пакете находятся программы мониторинга процессов. Эти программы необходимы для системного администрирования, а также используются загрузочными скриптами LFS Bootscript.

- **Psmisc**

В этом пакете находятся программы отображения информации о запущенных процессах. Эти программы используются при администрировании системы.

- **Readline**

Этот пакет представляет собой набор библиотек, которые предоставляют собой средства редактирования командной строки и средства работы с историей команд. Используется в пакете Bash.

- **Sed**

Этот пакет позволяет редактировать текст, не открывая его в текстовом редакторе. Пакет также нужен настройным скриптам в большинстве пакетов LFS.

- **Shadow**

В этом пакете расположены программы обработки паролей в защищенном режиме.

- **Sysklogd**

В этом пакете находятся программы для ведения журнала системных сообщений, похожих на те, которые выдаются процессами ядра или процессами-демонами при возникновении непредусмотренных ситуаций.

- **Sysvinit**
В этом пакете расположена программа `init`, являющаяся родительским процессом всех остальных процессов системы Linux.
- **Tar**
В этом пакете предоставляются средства архивирования и распаковки архивов, используемые практически во всех пакетах LFS.
- **Tcl**
В этом пакете находится язык обработки команд Tool Command Language, используемый во многих наборах тестов в пакетах LFS. Он устанавливается только во временном наборе инструментальных средств.
- **Texinfo**
В этом пакете находятся программы чтения, записи и преобразования страниц `info`. Пакет используется при инсталляции многих пакетов LFS
- **Udev**
В этом пакете находятся программы динамического создания нодов устройств. Это альтернативный способ создания тысяч статических устройств в каталоге `/dev`.
- **Util-linux**
В этом пакете находятся многочисленные утилиты. Среди них есть утилиты обработки файловых систем, консолей, разделов и сообщений.
- **Vim**
В этом пакете находится редактор. Он был выбран из-за его совместимости с классическим редактором `vi` и наличием у него огромного количества мощных возможностей. Для многих пользователей выбор редактора является сугубо личным делом и, при желании, его можно заменить любым другим редактором.
- **XZ Utils**
В этом пакете находятся программы сжатия и распаковки файлов. Пакет обеспечивает высокую степень сжатия и обычно используется пакетами распаковки пакетов в форматах XZ или LZMA.
- **Zlib**
В этом пакете находятся процедуры упаковки и распаковки, используемые некоторыми программами.

Предварительные требования

Сборка системы LFS – непростая задача. Для нее требуется определенный уровень знаний системного администрирования Unix, позволяющий решать возникающие проблемы и правильно выполнять указываемые команды. В частности, как абсолютный минимум, вы уже должны уметь пользоваться командной строкой (командной оболочкой), уметь копировать или перемещать файлы и каталоги, просматривать содержимое каталогов и файлов и переходить из одного каталога в другой. Также предполагается, что вы достаточно знакомы с использованием и установкой программ под Linux.

Поскольку книга LFS предполагает, *по крайней мере*, базовый уровень знаний, различные форумы поддержки LFS вряд ли будут в состоянии в этом обеспечить вам достаточную помощь. Вы обнаружите, что ваши вопросы, связанные с базовым уровнем знания, скорее всего, останутся без ответа, или вас просто перенаправят к списку документов, с которыми надо ознакомиться перед началом сборки LFS.

Перед тем, как браться за сборку системы LFS, мы рекомендуем прочитать следующие HOWTO:

- Software-Building-HOWTO (HOWTO по сборке программного обеспечения) <http://www.tldp.org/HOWTO/Software-Building-HOWTO.html>.

Это подробное руководство по сборке и установке в Linux «типичных» пакетов Unix программ. Хотя руководство было написано некоторое время назад, оно все еще является хорошим обзором основных методов, необходимых для сборки и установки программ.

- The Linux Users' Guide (Руководство пользователя Linux) <http://tldp.org/pub/Linux/docs/ldp-archived/users-guide/>

В этом руководстве описываются специально подобранные программы Linux. Эта ссылка также сравнительно старая, но она все еще доступна.

- The Essential Pre-Reading Hint (Важные предварительные советы) http://www.linuxfromscratch.org/hints/downloads/files/essential_prereading.txt

Это советы по проекту LFS, специально предназначенные для новых пользователей Linux. В них приводится список ссылок на замечательные источники информации по широкому кругу вопросов. Любой, кто пытается установить LFS, должен разбираться во многих темах, изложенных в этих советах.

Требование к хост-системе

На вашей хост-системе должно быть следующее программное обеспечение с номерами версий, не ниже указанных. Для большинства современных дистрибутивов Linux это не должно быть проблемой. Также отметим, что во многих дистрибутивах заголовочные файлы программ помещаются в отдельные пакеты, имеющие часто вид <имя-пакета>-devel или <имя-пакета>-dev. Не забудьте установить их, если они предлагаются в вашем дистрибутиве.

Более ранние версии перечисленных программ, возможно, будут работать, но они не проверялись.

- Bash-3.2 (должна быть символическая или обычная ссылка /bin/sh, указывающая на bash.)
- Binutils-2.17 (версии, выше чем 2.23.1, не рекомендуются, поскольку они не проверялись.)
- Bison-2.3 (должна быть символическая или обычная ссылка /usr/bin/уасс, указывающая на bison, либо нужен небольшой скрипт, который выполняет программу bison.)
- Bzip2-1.0.4
- Coreutils-6.9
- Diffutils-2.8.1
- Findutils-4.2.31
- Gawk-3.1.5 (должна быть ссылка /usr/bin/awk, указывающая на gawk.)
- Gcc-4.1.2 (версии, выше 4.7.2, не рекомендуются, поскольку они не проверялись.)
- Glibc-2.5.1 (версии, выше 2.17, не рекомендуются, поскольку они не проверялись.)
- Grep-2.5.1a
- Gzip-1.3.12
- Linux Kernel-2.6.25 (необходимо компилировать компилятором версии GCC-4.1.2 или выше.)

Причина такого требования к версии ядра обусловлена тем, что мы используем именно эту версию в главе 6 при сборке glibc в соответствии с рекомендациями разработчиков. Она также необходима для udev.

Если версия ядра хост-системы либо ниже 2.6.25, либо она не компилировалась с помощью компилятора GCC-4.1.2 (или более поздней), вам потребуется заменить ядро другим, которое будет соответствовать спецификациям. Есть два способа

сделать это. Во-первых, посмотреть, предоставляет ли поставщик вашей системы Linux пакет ядра версии 2.6.25 или более поздней. Если да, то вы можете установить его. Если ваш поставщик не предлагает приемлемого пакета с ядром, или вы предпочитаете его не устанавливать, вы можете скомпилировать ядро самостоятельно. В главе 8 есть инструкции по компиляции ядра и конфигурированию начального загрузчика (при условии, что в хостовой системе используется GRUB).

- M4-1.4.10
- Make-3.81
- Patch-2.5.4
- Perl-5.8.8
- Sed-4.1.5
- Tar-1.18
- Texinfo-4.9
- Xz-5.0.0

Заметим, что символические ссылки, упомянутые выше, необходимы для построения системы LFS с использованием инструкций, содержащихся в этой книге. Символические ссылки, которые указывают на другие программы (например, на `dash`, `mawk` и т. д.), могут также работать, но они не проверялись и не поддерживаются командой разработчиков LFS. Поэтому может потребоваться либо отходить от инструкций, либо для некоторых пакетов могут потребоваться дополнительные патчи.

Чтобы убедиться, что на вашей хост-системе есть все соответствующие версии, а также можно компилировать программы, выполните следующее:

```
cat > version-check.sh << "EOF"
#!/bin/bash
# Простой скрипт, выдающий список номеров версий инструментальных
# программ, важных для разработки

export LC_ALL=C
bash --version | head -n1 | cut -d" " -f2-4
echo "/bin/sh -> `readlink -f /bin/sh`"
echo -n "Binutils: "; ld --version | head -n1 | cut -d" " -f3-
bison --version | head -n1
if [ -e /usr/bin/yacc ];
  then echo "/usr/bin/yacc -> `readlink -f /usr/bin/yacc`;";
  else echo "yacc not found"; fi

bzip2 --version 2>&1 < /dev/null | head -n1 | cut -d" " -f1,6-
echo -n "Coreutils: "; chown --version | head -n1 | cut -d")" -f2
```

```
diff --version | head -n1
find --version | head -n1
gawk --version | head -n1
if [ -e /usr/bin/awk ];
    then echo "/usr/bin/awk -> `readlink -f /usr/bin/awk`";
    else echo "awk not found"; fi

gcc --version | head -n1
ldd --version | head -n1 | cut -d" " -f2- # glibc version
grep --version | head -n1
gzip --version | head -n1
cat /proc/version
m4 --version | head -n1
make --version | head -n1
patch --version | head -n1
echo Perl `perl -V:version`
sed --version | head -n1
tar --version | head -n1
echo `Texinfo: `makeinfo --version | head -n1`
xz --version | head -n1

echo `main(){}' > dummy.c && gcc -o dummy dummy.c
if [ -x dummy ];
    then echo "gcc compilation OK";
    else echo "gcc compilation failed"; fi
rm -f dummy.c dummy
EOF
```

```
bash version-check.sh
```

Типографские соглашения

Чтобы было проще понять излагаемый материал, везде в настоящей книге используются следующие типографские соглашения. В данном разделе проведено несколько примеров типографского оформления текста, используемых повсюду в книге «Linux с нуля».

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

С помощью этого варианта оформления текста указывается, как нужно точно вводить команду, если, конечно, иное не указано в сопровождающем тексте. Этот вариант оформления также используется в объяснениях, когда указываются команды.

В некоторых случаях логическую строку можно продлить на две или большее количество физических строк, если в конце строки указать символ обратного слеша.


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

Обратите внимание, что за символом обратного слеша должен сразу следовать символ возврата каретки. Если будут присутствовать другие символы, например, пробелы или символы табуляции, результат будет другой.

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

С помощью этого варианта оформления текста (моноширинный текст) изображаются данные, выводимые на экран, обычно, это результат ввода команды. Такой вариант оформления используется также для указания имен файлов, например, `/etc/ld.so.conf`.

Выделение

Этот вариант оформления текста используется в книге для нескольких целей. В основном так выделяются важные моменты или компоненты.

<http://www.linuxfromscratch.org/>

Этот вариант оформления используется (в оригинале книги и на сайте проекта – *прим.пер.*) для гиперссылок на страницы, расположенные внутри проекта LFS, и на страницы внешних источников. К ним относятся различные HOWTO, ссылки для скачивания данных и веб-сайты.

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

Этот вариант оформления используется, когда показываются конфигурационные файлы. В первой команде системе сообщается, что нужно создать файл `$LFS/etc/group` из того, что будет введено в следующих строках, и продолжать добавлять данные в файл до тех, пока не встретится символ конца файла (EOF). Следовательно, все надо набирать именно так, как изображено.

<ЗАМЕНЯЕМЫЙ ТЕКСТ>

Этот вариант оформления используется для указания текста, который не следует вводить так, как он показан, и не следует копировать и вставлять без каких-либо изменений.

[НЕОБЯЗАТЕЛЬНЫЙ ТЕКСТ]

Этот вариант оформления используется для указания текста, который не является обязательным.

`passwd(5)`

Этот вариант оформления используется для указания ссылок на страницу конкретного руководства (страницу `man`). Число в скобках указывает на конкретный раздел внутри руководства. Например, команда **passwd** имеет две страницы `man`. После установки инструкций LFS эти две страницы `man` будут находиться в `/usr/share/man/man1/passwd.1` и в `/usr/share/man/man5/passwd.5`. Когда в книге указывается `passwd(5)`, то ссылка будет относиться непосредственно к `/usr/share/man/man5/passwd.5`. **man passwd** будет выдавать первую страницу `man`, которая будет соответствовать запросу «`passwd`» и это будет `/usr/share/man/man1/passwd.1`. В данном примере вам нужно выполнить команду **man 5 passwd** с тем, чтобы прочитать страницу, о которой идет речь. Следует отметить, что у большинства страниц `man` нет повторяющихся названий в различных разделах. Следовательно, как правило, достаточно указать **man <имя программы>**.

Структура книги

Книга состоит из следующих частей.

Часть I – Введение

В части I приводятся несколько важных замечаний, касающихся того, как приступить к установке системы LFS. Здесь же также приведена мета-информация о книге.

Часть II – Подготовка к сборке

В части II описывается, как подготовиться к процессу сборки — создать раздел, загрузить пакеты и откомпилировать временный набор инструментальных средств.

Часть III – Сборка системы LFS

В части III руководства читатель пройдет через этапы сборки системы LFS — один за другим откомпилирует и установит все пакеты, настроит скрипты загрузки и установит ядро. Результирующая система Linux будет основой, на которой можно будет собирать другие программы для расширения системы нужным вам образом. В конце книги для удобства использования приводятся списки всех программ, библиотек и важных файлов, которые были установлены.

Возможные проблемы

Программы, используемые для создания системы LFS, постоянно обновляются и улучшаются. После того, как книга выпущена, могут появиться предупреждения, касающиеся безопасности системы, и патчи, исправляющие ошибки. Перед тем, как приступить к сборке, пожалуйста, посетите страницу <http://www.linuxfromscratch.org/lfs/errata/7.3/> с тем, чтобы проверить, нужны ли изменения версий пакетов или инструкций LFS, связанные с безопасностью или исправлением ошибок. Вам нужно обратить внимание на все изменения и применить их в соответствующих разделах книги в процессе сборки системы LFS.