

За пределами

Beyond LINUX from Scratch

LINUX

с нуля

Том 1

Версия 7.4



УДК 004.451Linux
ББК 32.973.26-018.2
Б60

Б60 Команда разработчиков BLFS
За пределами проекта «Linux® с нуля». Версия 7.4. Том 1 / пер. с англ.
Ромоданов Н. А. – М.: ДМК Пресс, 2014. – 792 с.: ил.

ISBN 978-5-97060-060-3

Проект «За пределами «Linux с нуля» является развитием проекта «Linux с нуля». Перевод сделан с англоязычной версии 7.4 – первой стабильной версии, выпущенной в течение последних пяти лет. Эта версия синхронизирована с версией 7.4 проекта «Linux с нуля» и представляет собой расширение последнего приблизительно 750 пакетами.

В первой части проекта, которая представлена в настоящей книге, приводится подробное описание установки и настройки пакетов, необходимых для безопасного использования системы и подключения ее к локальным и глобальным сетям. Из этой части вы также узнаете как правильно установить общесистемные библиотеки, сетевые и системные утилиты, основные командные оболочки и некоторые текстовые редакторы. Вы также сможете установить среду виртуализации и ряд инструментальных пакетов, необходимых при программировании на таких языках, как C, C++, Perl, Python, PHP, Ruby и Java.

УДК 004.451Linux
ББК 32.973.26-018.2

Linux® является зарегистрированной торговой маркой Linus Torvalds.

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

Материал, изложенный в данной книге, многократно проверен. Но, поскольку вероятность технических ошибок все равно существует, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 978-5-97060-060-3 (рус.)

© 2001-2013, The BLFS Development Team
© Оформление, перевод на русский язык
ДМК Пресс, 2014



ОГЛАВЛЕНИЕ

Пролог	14
Предисловие	15
Для кого предназначена данная книга	17
Структура книги	18
Возможные проблемы	20
ЧАСТЬ I.	
Введение	21
Глава 1. Добро пожаловать в проект BLFS	22
Какие разделы книги вам необходимы?	22
Типографские соглашения, используемые в данной книге	24
Версия книги	25
Сайты зеркал	26
Как получить пакеты с исходным кодом	26
Журнал изменений	27
Списки рассылок	27
Страницы Wiki проекта BLFS	27
Как обратиться за помощью и часто задаваемые вопросы FAQ	28
Вклад в проект	30
Контактная информация	32
Глава 2. Важная информация	33
Замечания, касающиеся сборки программ	33
В чем разница между /usr и /usr/local	41
Необязательные патчи	43
Загрузочные скрипты BLFS	43
Библиотеки: статические или динамически разделяемые?	44
Вопросы, касающиеся локалей	47
За пределами BLFS	52
ЧАСТЬ II.	
Постпроектная настройка LFS и дополнительные программы	57
Глава 3. Вопросы конфигурирования, не вошедшие в LFS	58

Создание специального загрузочного носителя	59
Конфигурирование процесса добавления пользователей	60
О системных пользователях и группах	62
Об устройствах	64
Файлы запуска командной оболочки Bash	68
Файлы /etc/vimrc и ~/.vimrc	76
Настраиваем ваш вход в систему с помощью /etc/issue	78
Файл /etc/shells	79
Генерация случайных чисел	80
Сжатие страниц Man и Info	80
Пакет lsb_release-1.4	93
Глава 4. Безопасность	94
Уязвимости	94
Пакет acl-2.2.52	96
Пакет attr-2.4.47	98
Сертификаты Certificate Authority	100
Пакет ConsoleKit-0.4.6	107
Пакет CrackLib-2.9.0	110
Пакет Cyrus SASL-2.1.26	113
Пакет GnuPG-1.4.14	117
Пакет GnuPG-2.0.21	118
Пакет GnuTLS-3.2.4	122
Пакет GPGME-1.4.3	125
Пакет iptables-1.4.20	126
Настройка сетевого брандмауэра	130
Пакет libcap2-2.22	141
Пакет Linux-PAM-1.1.7	143
Пакет MIT Kerberos V5-1.11.3	146
Пакет Nettle-2.7.1	154
Пакет NSS-3.15.1	156
Пакет OpenSSH-6.3p1	158
Пакет OpenSSL-1.0.1e	163
Пакет p11-kit-0.20.1	166
Пакет Polkit-0.111	168
Пакет Shadow-4.1.5.1	171
Пакет stunnel-4.56	180
Пакет Sudo-1.8.7	184
Пакет Tripwire-2.4.2.2	187
Глава 5. Файловые системы и управление дисками	192
О файловой системе initramfs	192
Пакет Fuse-2.9.3	200
Пакет jfsutils-1.1.15	203
Пакет LVM2-2.02.100	205

О массиве RAID	207
Пакет mdadm-3.3	211
Пакет ntfs-3g-2013.1.13.....	213
Пакет gptfdisk-0.8.7	216
Пакет parted-3.1	218
Пакет reiserfsprogs-3.6.24.....	220
Пакет sshfs-fuse-2.4	222
Пакет xfsprogs-3.1.11	223
Глава 6. Текстовые редакторы	227
Пакет Bluefish-2.2.4	227
Пакет Ed-1.9	228
Пакет Emacs-24.3.....	230
Пакет JOE-3.7.....	231
Пакет Nano-2.3.2	233
Пакет Vim-7.4	235
Другие редакторы	238
Глава 7. Командные оболочки	239
Пакет Dash-0.5.7	239
Пакет Tcsh-6.18.01	241
Пакет zsh-5.0.2	243
Глава 8. Виртуализация	247
Пакет qemu-1.6.0	247
ЧАСТЬ III.	
Библиотеки и утилиты общего назначения	253
Глава 9. Библиотеки общего назначения.....	254
Пакет Apr-1.4.8	255
Пакет Apr-Util-1.5.2	256
Пакет Aspell-0.60.6.1	258
Пакет Boost-1.54.0	260
Пакет dbus-glib-0.100.2.....	262
Пакет enchant-1.6.0	264
Пакет Exempi-2.2.1	265
Пакет Expat-2.1.0	266
Пакет Gamin-0.1.10	268
Пакет GLib-2.36.4	269
Пакет GLibmm-2.36.2	272
Пакет GMime-2.6.17	274
Пакет gobject-introspection-1.36.0	275
Пакет Gsl-1.16	277
Пакет ICU-51.2	278
Пакет ISO Codes-3.46	281

Пакет JS-17.0.0	282
Пакет JSON-C-0.11	283
Пакет JSON-GLib-0.16.0	285
Пакет keyutils-1.5.6	286
Пакет libarchive-3.1.2	287
Пакет libassuan-2.1.1	288
Пакет libatasmart-0.19	290
Пакет libatomic_ops-7.2d	291
Пакет libcroco-0.6.8	293
Пакет libdaemon-0.14	294
Пакет libdbusmenu-qt-0.9.2	296
Пакет libdrm-2.4.46	297
Пакет libESMTP-1.0.6	299
Пакет libffi-3.0.13	300
Пакет libgcrypt-1.5.3	301
Пакет libgee-0.6.8	303
Пакет libglade-2.6.4	304
Пакет libgpg-error-1.12	306
Пакет libgsf-1.14.28	307
Пакет libgusb-0.1.6	309
Пакет libical-1.0	310
Пакет LibIDL-0.8.14	312
Пакет libidn-1.28	313
Пакет libiodbc-3.52.8	315
Пакет Libksba-1.3.0	317
Пакет libsigc++-2.2.11	318
Пакет libtasn1-3.3	319
Пакет libunistring-0.9.3	321
Пакет libusb-1.0.9	322
Пакет libusb-compat-0.1.5	323
Пакет libxml2-2.9.1	325
Пакет libxslt-1.1.28	327
Пакет libzeitgeist-0.3.18	328
Пакет LZO-2.06	330
Пакет mtdev-1.1.4	331
Пакет NSPR-4.10	332
Пакет OpenOBEX-1.7.1	334
Пакет PCRE-8.33	336
Пакет Popt-1.16	338
Пакет Pth-2.0.7	339
Пакет Ptlib-2.10.10	341
Пакет Qca-2.0.3	342
Пакет QJson-0.8.1	344
Пакет SBC-1.1	345

Пакет SLIB-3b4	346
Пакет Tallos-2.1.0	348
Пакет wv-1.2.9	349

Глава 10. Графические библиотеки и библиотеки шрифтов 351

Пакет AAlib-1.4rc5	351
Пакет babl-0.1.10	353
Пакет Exiv2-0.23	354
Пакет FreeType-2.5.0.1	355
Пакет Fontconfig-2.10.93	357
Пакет FriBidi-0.19.5	360
Пакет gegl-0.2.0	362
Пакет giflib-5.0.5	363
Пакет Harfbuzz-0.9.20	365
Пакет IJS-0.35	366
Пакет lmlib2-1.4.5	368
Пакет JasPer-1.900.1	369
Пакет Little CMS-1.19	371
Пакет Little CMS-2.5	373
Пакет libexif-0.6.21	374
Пакет libjpeg-turbo-1.3.0	376
Пакет libmng-2.0.2	378
Пакет libpng-1.6.4	379
Пакет librsvg-2.37.0	381
Пакет LibTIFF-4.0.3	383
Пакет libwebp-0.3.1	385
Пакет OpenJPEG-1.5.1	387
Пакет Pixman-0.30.2	388
Пакет Poppler-0.24.51	389
Пакет Qpdf-5.0.0	393

Глава 11. Утилиты общего назначения 395

Пакет bc-1.06.95	395
Пакет Comfaced-1.5.2	396
Пакет desktop-file-utils-0.22	398
Пакет Gperf-3.0.4	400
Пакет Graphviz-2.34.0	401
Пакет GTK-Doc-1.19	408
Пакет Hd2u-1.0.3	410
Пакет icon-naming-utils-0.8.90	411
Пакет ImageMagick-6.8.6-9	412
Пакет Intltool-0.50.2	416
Пакет PIN-Entry-0.8.3	418
Пакет qtchooser-31	419

Пакет Rarian-0.8.1	422
Пакет Rep-gtk-0.90.8.1	424
Пакет rxvt-unicode-9.18	425
Пакет Screen-4.0.3	428
Пакет Sharutils-4.13.5	430
Пакет HTML Tidy-cvs_20101110	431
Пакет unixODBC-2.3.1	433
Пакет XScreenSaver-5.22	436
Глава 12. Системные утилиты	439
Пакет ascpid-2.0.19	439
Пакет at-3.1.14	441
Пакет autofs-5.0.7	443
Пакет BlueZ-4.101	447
Пакет Colord-1.0.3	451
Пакет cpio-2.11	453
Пакет D-Bus-1.6.14	455
Пакет eject-2.1.5	460
Пакет Fcron-3.1.2	461
Пакет GPM-1.20.7	464
Пакет Hdparm-9.43	468
Пакет IBus-1.5.3	469
Пакет LSB Tools для управления загрузочными скриптами	471
Пакет lm_sensors-3.3.4	472
Пакет MC-4.8.10	475
Пакет obex-data-server-0.4.6	478
Пакет Obexd-0.48	479
Пакет pciutils-3.2.0	480
Пакет pm-utils-1.4.1	482
Пакет Raptor-2.0.10	484
Пакет Rasqal-0.9.30	486
Пакет Redland-1.0.16	487
Пакет sg3_utils-1.36	488
Пакет Strigi-0.7.8	493
Пакет Sysstat-10.1.7	495
Пакет Udev – версия, установленная в LFS	498
Дополнения пакета Udev (из пакета systemd)	500
Пакет UDisks-1.0.4	503
Пакет UDisks-2.1.1	504
Пакет UnRar-5.0.11	506
Пакет UnZip-6.0	507
Пакет UPower-0.9.21	510
Пакет usbutils-007	512
Пакет Which-2.20 и его альтернативные варианты	514
Пакет Zip-3.0	516

Глава 13. Программирование	518
Пакет Bazaar-2.5.1	518
Пакет Check-0.9.10	519
Пакет CMake-2.8.11.2	520
Пакет CVS-1.11.23	522
Запуск сервера CVS	525
Пакет DejaGnu-1.5.1	528
Пакет Doxygen-1.8.5	529
Пакет elfutils-0.156	531
Пакет Expect-5.45	532
Пакет GCC-4.8.1	536
Пакет GC-7.2d	543
Пакет GDB-7.6.1	545
Пакет Git-1.8.4	547
Пакет Guile-2.0.9	551
Пакет Librep-0.92.3	553
Пакет LLVM-3.3	554
Пакет Lua-5.1.5	559
Пакет Mercurial-2.7.1	561
Пакет NASM-2.10.09	562
Пакет Perl Modules	564
Пакет PHP-5.5.3	569
Пакет pkg-config-0.28	574
Пакет Python-2.7.5	576
Пакет Python-3.3.2	579
Пакет Python Modules	581
Модуль D-Bus Python	582
Модуль Notify Python-0.1.1	584
Модуль Py2cairo-1.10.0	585
Модуль PyCairo-1.10.0	586
Модуль PyGObject-2.28.6	587
Модуль PyGObject-3.8.3	588
Модуль PyGTK-2.24.0	590
Модуль PyAtSpi2-2.8.0	592
Модуль Pyrex-0.9.9	593
Модуль PyXDG-0.25	594
Пакет Ruby-2.0.0	595
Пакет SCons-2.3.0	596
Пакет S-Lang-2.2.4	597
Пакет Subversion-1.8.3	599
Запуск сервера Subversion	603
Пакет Tcl-8.6.0	607
Пакет Tk-8.6.0	609
Пакет Vala-0.20.1	611
Пакет yasm-1.2.0	613

Другие инструментальные средства программирования	614
О языке Java	648
Пакет JUnit-4.11	650
Пакет apache-ant-1.9.2	652
Пакет OpenJDK-1.7.0.40/IcedTea-2.4.1	655

ЧАСТЬ IV.

Сеть	671
-------------------	------------

Глава 14. Подключение к сети	672
-------------------------------------------	------------

Пакет dhcpd-6.0.5	672
Пакет DHCP-4.2.5-P1	675

Глава 15. Программы для работы с сетью	681
-----------------------------------------------------	------------

Пакет bridge-utils-1.5	681
Пакет cifs-utils-6.1	684
Пакет NcFTP-3.2.5	686
Пакет Net-tools-CVS_20101030	688
Пакет NFS-Utills-1.2.8	691
Конфигурирование сетевых файловых систем	696
Пакет ntp-4.2.6p5	697
Пакет rpcbind-0.2.1	701
Пакет rsync-3.0.9	702
Пакет Samba-4.0.9	705
Пакет Wget-1.14	717
Пакет Wireless Tools-29	719
Пакет wpa_supplicant-2.0	721
Другие программы для работы с сетью	727

Глава 16. Сетевые утилиты	729
----------------------------------------	------------

Пакет Avahi-0.6.31	729
Пакет BIND Utilities-9.9.3-P2	733
Пакет mod_dnssd-0.6	734
Пакет NetworkManager-0.9.8.2	736
Пакет Nmap-6.40	739
Пакет Traceroute-2.0.19	740
Пакет Whois-5.0.26	742
Пакет Wicd-1.7.2.4	743
Пакет Wireshark-1.10.2	746

Глава 17. Сетевые библиотеки	752
-------------------------------------------	------------

Пакет cURL-7.32.0	752
Пакет GeoClue-0.12.0	754
Пакет glib-networking-2.36.2	756
Пакет Iidns-1.6.16	757
Пакет libevent-2.0.21	759

Пакет libnice-0.1.4	761
Пакет libnl-3.2.22	762
Пакет libpcap-1.4.0	764
Пакет libsoup-2.42.2	765
Пакет libtirpc-0.2.3	767
Пакет neon-0.30.0	768
Пакет Serf-1.3.1	770
Глава 18. Текстовые веб-браузеры	772
Пакет Links-2.7	772
Пакет Lynx-2.8.8dev.16	774
Пакет W3m-0.5.3	777
Глава 19. Клиентские программы для работы с почтой и новостными группами.....	780
Пакет Heirloom mailx-12.4	780
Пакет Procmail-3.22	782
Пакет Fetchmail-6.3.26	784
Пакет Mutt-1.5.21	786
Пакет Re-alpine-2.03	789
Другие программы для работы с почтой и новостными группами	791



ГЛАВА 5.

Файловые системы и управление дисками

Журналируемые файловые системы сокращают время, необходимое для восстановления файловой системы в случае, если она не была размонтирована должным образом. Хотя такие файловые системы крайне важны для сокращения времени простоя серверов, они также становятся популярными и в настольных системах. В настоящем разделе рассматриваются журналируемые файловые системы, которые вы можете использовать вместо тех систем (ext2/3/4), которые были установлены по умолчанию в проекте LFS. Также предлагается вводный материал по управлению массивами дисков.

О файловой системе `initramfs`

Единственное назначение файловой системы `initramfs` – смонтировать файловую систему `root`. Файловая система `initramfs` является полным набором каталогов, которые вы можете найти в обычной файловой системе `root`. Она собрана в один архив `сrio` и сжата с помощью одного из нескольких алгоритмов сжатия.

Во время загрузки системы загрузчик загружает ядро и образ файловой системы `initramfs` в память и запускает ядро. Ядро проверяет наличие файловой системы `initramfs` и, если она найдена, монтирует ее в точке монтирования и запускает `/init`. Программа `init` обычно является скриптом командной оболочки. Обратите внимание, что, если используется файловая система `initramfs`, процесс загрузки длится дольше, возможно, существенно дольше.

Во многих дистрибутивах основной причиной наличия файловой системы `initramfs` является необходимость использования модулей ядра. В типичном дистрибутиве может быть много неопределенностей разного рода, например, типы файловых систем и разбиение дисков на разделы. В некотором смысле это является противоположностью

проекта LFS, в котором известны все особенности системы и разбиение диска на разделы, а также в котором обычно собирается достаточно типичное ядро. В такой ситуации файловая система `initramfs` требуется редко.

Есть только четыре главные причины использовать файловую систему в среде LFS: загрузка файловой системы `rootfs` из сети, загрузка ее с логического тома LVM, наличие зашифрованной файловой системы `rootfs`, для которой требуется пароль, или для того, чтобы было удобно определить файловую систему `rootfs` с использованием метки LABEL или идентификатора UUID. Все другое обычно означает, что ядро не было сконфигурировано должным образом.

Сборка файловой системы `initramfs`

Если вы решили собрать файловую систему `initramfs`, то приводимые далее скрипты будут хорошей для этого основой. Скрипты позволяют определить файловую систему `rootfs` через идентификатор UUID или метку LABEL, определяющую раздел, либо задать файловую систему `rootfs` на логическом томе LVM. В них не поддерживается шифрование файловой системы `root` или ее монтирование с использованием сетевой карты. В более сложных ситуациях обращайтесь к советам проекта LFS (LFS Hints).

Для того, чтобы установить эти скрипты, запустите в роли пользователя `root` следующие команды:

```
cat >/sbin/mkinitramfs <<"EOF"
#!/bin/bash
# Этот файл частично основан на скрипте mkinitramfs для LFS LiveCD,
# написанном Александром Е. Патраковым и Jeremy Huntwork.

copy()
{
    local file

    if [ "$2"=="lib"]; then
        file=$(PATH=/lib:/usr/lib type -p $1)
    else
        file=$(type -p $1)
    fi

    if [ -n $file ] ; then
        cp $file $WDIR/$2
    else
        echo "Missing required file: $1 for directory $2"
        rm -rf $WDIR
    fi
}
```

```

        exit 1
    fi
}

if [ -z $1 ] ; then
    INITRAMFS_FILE=initrd.img-no-kmods
else
    KERNEL_VERSION=$1
    INITRAMFS_FILE=initrd.img-$KERNEL_VERSION
fi

if [ -n "$KERNEL_VERSION" ] &&[ ! -d "/lib/modules/$1" ] ; then
    echo "No modules directory named $1"
    exit 1
fi

printf "Creating $INITRAMFS_FILE... "

binfiles="sh cat cp dd killall ls mkdir mknod mount "
binfiles="$binfiles umount sed sleep ln rm uname"

sbinfiles="udevadm modprobe blkid switch_root"

#Optional files and locations
for f in mdadm udevd; do
    if [ -x /sbin/$f ] ; then sbinfiles="$sbinfiles $f"; fi
done

unsorted=$(mktemp /tmp/unsorted.XXXXXXXXXX)

DATADIR=/usr/share/mkinitramfs
INITIN=init.in

# Создаем временный рабочий каталог
WDIR=$(mktemp -d /tmp/initrd-work.XXXXXXXXXX)

# Создаем структуру базового каталога
mkdir -p $WDIR/{bin,dev,lib/firmware,run,sbin,sys,proc}
mkdir -p $WDIR/etc/{modprobe.d,udev/rules.d}
touch $WDIR/etc/modprobe.d/modprobe.conf
ln -s lib $WDIR/lib64

# Создаем ноды необходимых устройств
mknod -m 640 $WDIR/dev/console c 5 1
mknod -m 664 $WDIR/dev/null c 1 3

# Устанавливаем конфигурационные файлы udev
cp /etc/udev/udev.conf $WDIR/etc/udev/udev.conf

for file in $(find /etc/udev/rules.d/ -type f) ; do

```

```
cp $file $WDIR/etc/udev/rules.d
done

# Устанавливаем любое имеющееся firmware
cp -a /lib/firmware $WDIR/lib

# Копируем конфигурационный файл RAID, если он имеется
if [ -f /etc/mdadm.conf ] ; then
  cp /etc/mdadm.conf $WDIR/etc
fi

# Устанавливаем файл init
install -m0755 $DATADIR/$INITIN $WDIR/init

if [ -n "$KERNEL_VERSION" ] ; then
  if [ -x /bin/kmod ] ; then
    binfiles="$binfiles kmod"
  else
    binfiles="$binfiles lsmod"
    sbinfiles="$sbinfiles insmod"
  fi
fi

# Устанавливаем основные двоичные файлы
for f in $binfiles ; do
  ldd /bin/$f | sed "s/\t//" | cut -d "-" -f1 >>$unsorted
  copy $f bin
done

# Добавляем lvm, если он есть
if [ -x /sbin/lvm ] ; then sbinfiles="$sbinfiles lvm"; fi

for f in $sbinfiles ; do
  ldd /sbin/$f | sed "s/\t//" | cut -d "-" -f1 >>$unsorted
  copy $f sbin
done

# Добавляем библиотеки udevd, если их нет в /sbin
if [ -x /lib/udev/udevd ] ; then
  ldd /lib/udev/udevd | sed "s/\t//" | cut -d "-" -f1 >>$unsorted
fi

# Добавляем символические ссылки, если они уместны
if [ -n "$KERNEL_VERSION" ] && [ -x /bin/kmod ] ; then
  ln -s kmod $WDIR/bin/lsmod
  ln -s kmod $WDIR/bin/insmod
fi

# Добавляем символические ссылки lvm, если они уместны
if [ -x /sbin/lvm ] ; then
```

```

ln -s lvm $WDIR/sbin/lvchange
ln -s lvm $WDIR/sbin/lvrename
ln -s lvm $WDIR/sbin/lvextend
ln -s lvm $WDIR/sbin/lvcreate
ln -s lvm $WDIR/sbin/lvdisplay
ln -s lvm $WDIR/sbin/lvscan

ln -s lvm $WDIR/sbin/pvchange
ln -s lvm $WDIR/sbin/pvck
ln -s lvm $WDIR/sbin/pvcreate
ln -s lvm $WDIR/sbin/pvdisplay
ln -s lvm $WDIR/sbin/pvscan

ln -s lvm $WDIR/sbin/vgchange
ln -s lvm $WDIR/sbin/vgcreate
ln -s lvm $WDIR/sbin/vgscan
ln -s lvm $WDIR/sbin/vgrename
ln -s lvm $WDIR/sbin/vgck
fi

# Устанавливаем библиотеки
sort $unsorted | uniq | while read library ; do
    if [ "$library"=="linux-vdso.so.1" ] ||
        [ "$library"=="linux-gate.so.1" ]; then
        continue
    fi

    copy $library lib
done

cp -a /lib/udev $WDIR/lib

# Устанавливаем модули ядра, если это требуется
if [ -n "$KERNEL_VERSION" ]; then
    find \
        /lib/modules/$KERNEL_VERSION/kernel/{crypto,fs,lib} \
        /lib/modules/$KERNEL_VERSION/kernel/drivers/{block,ata,md,firewire} \
        /lib/modules/$KERNEL_VERSION/kernel/drivers/{scsi,message,pcmcia,virtio} \
        /lib/modules/$KERNEL_VERSION/kernel/drivers/usb/{host,storage} \
        -type f 2>/dev/null | cpio --make-directories -p --quiet $WDIR

    cp /lib/modules/$KERNEL_VERSION/modules.{builtin,order} \
        $WDIR/lib/modules/$KERNEL_VERSION

    depmod -b $WDIR $KERNEL_VERSION
fi

( cd $WDIR ; find . | cpio -o -H newc --quiet | gzip -9 ) >$INITRAMFS_FILE

# Удаляем временный каталог и файл

```



```
rm -rf $WDIR $unsorted
printf "done.\n"
```

```
EOF
```

```
chmod 0755 /sbin/mkinitramfs
mkdir -p /usr/share/mkinitramfs &&
cat >/usr/share/mkinitramfs/init.in <<"EOF"
#!/bin/sh
```

```
PATH=/bin:/usr/bin:/sbin:/usr/sbin
export PATH
```

```
problem()
{
    printf "Encountered a problem!\n\nDropping you to a shell.\n\n"
    sh
}
```

```
no_device()
{
    printf "The device %s, which is supposed to contain the\n"$1
    printf "root file system, does not exist.\n"
    printf "Please fix this problem and exit this shell.\n\n"
}
```

```
no_mount()
{
    printf "Could not mount device %s\n"$1
    printf "Sleeping forever. Please reboot and fix the kernel command
line.\n\n"
    printf "Maybe the device is formatted with an unsupported file
system?\n\n"
    printf "Or maybe filesystem type autodetection went wrong, in
which case\n"
    printf "you should add the rootfstype=... parameter to the kernel
command line.\n\n"
    printf "Available partitions:\n"
}
```

```
do_mount_root()
{
    mkdir /.root
    [ -n "$rootflags" ] &&rootflags="$rootflags,"
    rootflags="$rootflags$ro"

    case "$root" in
        /dev/* ) device=$root ;;
        UUID=* ) eval $root; device="/dev/disk/by-uuid/$UUID" ;;
        LABEL=* ) eval $root; device="/dev/disk/by-label/$LABEL";;
```

```

    "" ) echo "No root device specified."; problem ;;
esac

while [ ! -b "$device" ] ; do
    no_device $device
    problem
done

if ! mount -n -t "$rootfstype"-o "$rootflags"$device"/.root ; then
    no_mount $device
    cat /proc/partitions
    while true ; do sleep 10000 ; done
else
    echo "Successfully mounted device $root"
fi
}

init=/sbin/init
root=
rootdelay=
rootfstype=auto
ro="ro"
rootflags=
device=

mount -n -t devtmpfs devtmpfs /dev
mount -n -t proc proc /proc
mount -n -t sysfs sysfs /sys
mount -n -t tmpfs tmpfs /run

read -r cmdline </proc/cmdline

for param in $cmdline ; do
    case $param in
        init=* ) init=${param#init=} ;;
        root=* ) root=${param#root=} ;;
        rootdelay=* ) rootdelay=${param#rootdelay=} ;;
        rootfstype=* ) rootfstype=${param#rootfstype=} ;;
        rootflags=* ) rootflags=${param#rootflags=} ;;
        ro ) ro="ro" ;;
        rw ) ro="rw" ;;
    esac
done

# месторасположение udevd зависит от используемой версии
if [ -x /sbin/udev ] ; then
    UDEV_PATH=/sbin
else
    UDEV_PATH=/lib/udev
fi

${UDEV_PATH}/udev --daemon --resolve-names=never

```

```

udevadm trigger
udevadm settle

if [ -f /etc/mdadm.conf ] ; then mdadm -As                ; fi
if [ -x /sbin/vgchange ] ; then /sbin/vgchange --noudevsync -a y >
/dev/null ; fi
if [ -n "$rootdelay" ] ; then sleep "$rootdelay"        ; fi

do_mount_root

killall -w ${UDEV_PATH}/udev

exec switch_root /.root "$init" "$@"

EOF

```

Использование файловой системы `initramfs`

Требуемые зависимости времени выполнения

срjо-2.11

Замечания для пользователей:

<http://wiki.linuxfromscratch.org/blfs/wiki/initramfs>

Чтобы собрать файловую систему `initramfs`, запустите в роли пользователя `root` следующую команду:

```
mkinitramfs [KERNEL VERSION]
```

Необязательным аргументом является каталог, в котором размещаются соответствующие модули ядра. Это должен быть подкаталог каталога `/lib/modules`. Если модули не указаны, то файловая система `initramfs` будет иметь название `initrd.img-no-kmods`. Если указана версия ядра, то `initrd` будет иметь название `initrd.img-$ВЕРСИЯ_ЯДРА` и будет использоваться только с конкретным ядром. Результирующий файл будет размещаться в текущем каталоге.

После генерации `initrd` скопируйте его в каталог `/boot`.

Теперь отредактируйте файл `/boot/grub/grub.cf` и добавьте в него новые пункты меню. Ниже приведены несколько примеров.

```

# Общая initramfs и файловая система root,
# указываемая с помощью идентификатора UUID
menuentry "LFS Dev (LFS-7.0-Feb14) initrd, Linux 3.0.4"
{
    linux /vmlinuz-3.0.4-lfs-20120214 root=UUID=54b934a9-302d-415e-
ac11-4988408eb0a8 ro

```

```
    initrd /initrd.img-no-kmods
}
# Общая initramfs и файловая система root в разделе тома LVM
menuentry "LFS Dev (LFS-7.0-Feb18) initrd lvm, Linux 3.0.4"
{
    linux /vmlinuz-3.0.4-lfs-20120218 root=/dev/mapper/myroot ro
    initrd /initrd.img-no-kmods
}
# Конкретная initramfs файловая система root, идентифицируемая
# меткой LABEL
menuentry "LFS Dev (LFS-7.1-Feb20) initrd label, Linux 3.2.6"
{
    linux /vmlinuz-3.2.6-lfs71-120220 root=LABEL=lfs71 ro
    initrd /initrd.img-3.2.6-lfs71-120220
}
}
```

И, в заключении, перезагрузите машину и выберите нужную систему.

Перевод с оригинала от 2013-02-11 10:51:17 -0800

Пакет Fuse-2.9.3

Знакомимся с пакетом Fuse

FUSE (Filesystem in Userspace – файловая система в пользовательском пространстве) является простым интерфейсом для программ пользовательского пространства, который экспортирует виртуальную файловую систему в ядро Linux. Назначение пакета Fuse также состоит в предоставлении непривилегированным пользователям безопасного метода для создания и монтирования своих собственных реализаций файловых систем.

Известно, что пакет правильно собирается и работает на платформе LFS-7.4.

Информация о пакете

- Загрузка (HTTP): <http://downloads.sourceforge.net/fuse/fuse-2.9.3.tar.gz>
- Контрольная сумма MD5: 33cae22ca50311446400daf8a6255c6a
- Размер загружаемого пакета: 564 KB
- Оценочный размер требуемого дискового пространства: 9,5 MB
- Оценочное время сборки: 0,1 SBU

Замечания для пользователей:

<http://wiki.linuxfromscratch.org/blfs/wiki/fuse>

Конфигурирование ядра

Включите следующие параметры в конфигурации ядра и перекомпилируйте ядро, если это необходимо:

```
File systems --->
  [*] FUSE (Filesystem in Userspace) support
```

Установка пакета Fuse

Замечание

После того, как скрипт **configure** закончит свою работу, вы увидите предупреждающее сообщение, приведенное ниже и рекомендуемое установить util-linux версии 2.18 или выше, в которых в операциях mount и umount поддерживается использование параметров **--fake** и **--no-canonicalize**. Вы можете спокойно игнорировать это предупреждение,

```
configure: WARNING:
*****
* Please install util-linux version 2.18 or later which *
* supports --fake and --no-canonicalize options in mount *
* and umount                                           *
*****
```

Установите пакет Fuse с помощью запуска следующих команд:

```
./configure --prefix=/usr --disable-static INIT_D_PATH=/tmp/init.d &&
make
```

Если у вас установлен пакет [Doxygen-1.8.5](#) и вы хотите собрать документацию API, выполните команду **doxygen doc/Doxyfile**.

Этот пакет поставляется без тестового набора.

Теперь в роли пользователя **root** выполните:

```
make install &&

mv -v /usr/lib/libfuse.so.* /lib &&
ln -sfv ../../lib/libfuse.so.2.9.3 /usr/lib/libfuse.so &&
rm -rf /tmp/init.d &&

install -v -m755 -d /usr/share/doc/fuse-2.9.3 &&
install -v -m644 doc/{how-fuse-works, kernel.txt} \
  /usr/share/doc/fuse-2.9.3
```

Если вы собрали документацию API, то в роли пользователя **root** установите ее с помощью следующих команд:

```
install -v -m755 -d /usr/share/doc/fuse-2.9.3/api &&
```

```
install -v -m644 doc/html/* \
/usr/share/doc/fuse-2.9.3/api
```

Пояснение команд

`--disable-static`: Этот переключатель предотвращает установку статических версий библиотек.

`INIT_D_PATH=/tmp/init.d`: Этот параметр устанавливает загрузочный скрипт в каталог `/tmp/init.d`, т. к. загрузочный скрипт не требуется.

`mv -v /usr/lib/libfuse.so.* /lib`: С помощью этой команды происходит перемещение библиотеки FUSE в файловую систему `root`, поскольку она в случае, когда `/usr` монтируется в отдельном разделе, а пакет [ntfs-3g-2013.1.13](#) собирается с версий пакета FUSE, установленного в системе, должна быть в наличии на ранних этапах процесса.

`rm -rf /tmp/init.d`: Эта команда удаляет ненужный загрузочный скрипт.

Конфигурирование пакета *fuse*

Конфигурационные файлы

Некоторые необязательные параметры монтирования, относящиеся к политике монтирования, могут быть установлены в файле `/etc/fuse.conf`. Чтобы установить этот файл, в роли пользователя `root` выполните следующие команды:

```
cat > /etc/fuse.conf << "EOF"
# Установка максимального числа точек монтирования, доступных для
# пользователей, не являющихся пользователями root. По умолчанию
# равно 1000.
#
#mount_max = 1000

# Разрешает пользователям, не являющимся пользователями root,
# указывать параметры монтирования 'allow_other' или 'allow_root'.
#
#user_allow_other
EOF
```

Дополнительную информацию о смысле конфигурационных параметров ищите на странице `man`.

Описание пакета

Установленные программы: `fusermount`, `mount.fuse` и `ulockmgr_server`

Установленные библиотеки: `libfuse.so` и `libulockmgr.so`

Установленные каталоги: `/usr/include/fuse`

Краткое описание

fusermount Устанавливает программу `root` с пользовательским идентификатором `ID`, которая монтирует и демонтирует файловые системы `Fuse`.

mount.fuse Является командой `mount`, которая должна вызываться для монтирования файловой системы `Fuse`.

ulockmgr_server Является сервером менеджера блокировок пользовательского пространства `Userspace Lock Manager Server` для файловых систем `Fuse`.

`libfuse.so` Содержит функции API пакета `Fuse`.

`libulockmgr.so` Содержит функции API менеджера блокировок пользовательского пространства `Userspace Lock Manager`.

Перевод с оригинала от 2013-08-22 04:10:10 -0700

Пакет `jfsutils-1.1.15`

Знакомимся с пакетом `jfsutils`

В пакете `jfsutils` находятся административные и отладочные средства для файловой системы `jfs`.

Известно, что пакет правильно собирается и работает на платформе `LFS-7.4`.

Информация о пакете

- Загрузка (HTTP): <http://jfs.sourceforge.net/project/pub/jfsutils-1.1.15.tar.gz>
- Контрольная сумма MD5: `8809465cd48a202895bc2a12e1923b5d`
- Размер загружаемого пакета: 532 KB
- Оценочный размер требуемого дискового пространства: 8,9 MB
- Оценочное время сборки: 0,1 SBU

Замечания для пользователей:

<http://wiki.linuxfromscratch.org/blfs/wiki/jfs>

Конфигурирование ядра

Включите следующие параметры в конфигурации ядра и перекомпилируйте ядро:

```
File Systems:
  JFS filesystem support: M or Y
```

Установка пакета *jfsutils*

Установите пакет *jfsutils* с помощью следующих команд:

```
sed "s@<unistd.h>@&\n#include <sys/types.h>@g" -i fscklog/extract.c &&
./configure &&
make
```

Данный пакет не поставляется с тестовым набором.

Теперь в роли пользователя `root` выполните:

```
make install
```

Пояснение команд

```
sed "s@<unistd.h>@&\n#include <sys/types.h>@g" -i fscklog/
extract.c: Фиксирует ошибки сборки, связанные с Glibc 2.17.
```

Описание пакета

Установленные программы: `fsck.jfs`, `jfs_debugfs`, `jfs_fsck`, `jfs_fscklog`,
`jfs_logdump`, `jfs_mkfs`, `jfs_tune`, `mkfs.jfs`

Установленные библиотеки: Нет

Установленные каталоги: Нет

Краткое описание

- `fsck.jfs`** Используется для прокрутки журнала транзакций JFS, проверки ошибок устройства, форматированного в JFS, а также фиксации всех найденных ошибок.
- `jfs_fsck`** Жесткая ссылка на `fsck.jfs`.
- `mkfs.jfs`** Создает файловую систему JFS.
- `jfs_mkfs`** Жесткая ссылка на `mkfs.jfs`.
- `jfs_debugfs`** Программа, которую можно использовать для выполнения различных низкоуровневых действий на устройстве, форматированном в JFS.
- `jfs_fscklog`** Преобразует журнал сервиса JFS `fsck` в файла и/или форматирует и выдает содержимое этого файла.

jfs_logdump Преобразует дамп содержимого журнала для указанного устройства, форматированного в JFS, в результирующий файл `./jfslog.dmp`.

jfs_tune Регулирует настраиваемые параметры файловой системы в файловых системах JFS.

Перевод с оригинала от 2013-08-23 03:32:24 -0700

Пакет LVM2-2.02.100

Знакомимся с пакетом LVM2

Пакет LVM2 является пакетом, который управляет логическими разделами. Он позволяет распределять файловые системы по многим физическим устройствам и разделам диска и предоставляет динамические средства, позволяющие увеличивать размеры логических разделов и их динамически продлевать на другие устройства.

Известно, что пакет правильно собирается и работает на платформе LFS-7.4.

Информация о пакете

- Загрузка (FTP): <ftp://sources.redhat.com/pub/lvm2/LVM2.2.02.100.tgz>
- Контрольная сумма MD5: 9629cf5728544d7e637cafde1f73d777
- Размер загружаемого пакета: 1,3 МВ
- Оценочный размер требуемого дискового пространства: 22 МВ
- Оценочное время сборки: 0,3 SBU

Зависимости пакета LVM2

Необязательные

mdadm-3.3 (для проверки) и xfsprogs-3.1.11 (для проверки)

Замечания для пользователей:

<http://wiki.linuxfromscratch.org/blfs/wiki/lvm2>

Конфигурирование ядра

Замечание

В ядре помимо тех параметров Device Mapper, которые указаны ниже, есть несколько других параметров. Для того, чтобы при работе регрессивных тестов получить приемлемые результаты, все пара-