Часть III. Глава 6. Кросс-Компиляция временных инструментов

Содержание

- 6.1. Введение
- 6.2. M4-1.4.19
- 6.3. Ncurses-6.4-20230520
- 6.4. Bash-5.2.21
- 6.5. Coreutils-9.4
- 6.6. Diffutils-3.10
- 6.7. File-5.45
- 6.8. Findutils-4.9.0
- 6.9. Gawk-5.3.0
- 6.10. Grep-3.11
- 6.11. Gzip-1.13
- 6.12. Make-4.4.1
- 6.13. Patch-2.7.6
- 6.14. Sed-4.9
- 6.15. Tar-1.35
- 6.16. Xz-5.4.6
- 6.17. Binutils-2.42 - Проход 2
- 6.18. GCC-13.2.0 - Проход 2

6.1. Введение

В этой главе рассказывается, как выполнить кросс-компиляцию базовых утилит с использованием только что собранного кросс-тулчейна. Эти утилиты установлены в свое конечное местоположение, но пока не могут быть использованы. Выполняемые инструкции по-прежнему зависят от инструментария хоста. Тем не менее, установленные библиотеки используются при компоновке.

Использование утилит станет возможным в следующей главе после входа в среду «chroot». Все пакеты из этой главы, должны быть собраны до того, как мы это сделаем. Поэтому пока наша система зависима от хост-системы.

Еще раз напомним, что неправильная настройка LFS вместе со сборкой от root может сделать ваш компьютер непригодным для использования. Всю эту главу нужно выполнить от имени пользователя lfs, в его рабочем окружении, как описано в Разделе 4.4. «Настройка окружения». выполним проверки

```
id
```

```
lfs:/mnt/lfs/sources/glibc-2.39/build$ id
uid=1001(lfs) gid=1001(lfs) groups=1001(lfs)
lfs:/mnt/lfs/sources/glibc-2.39/build$
```

```
echo $LFS
```

```
lfs:/mnt/lfs/sources/glibc-2.39/build$ echo $LFS
/mnt/lfs
lfs:/mnt/lfs/sources/glibc-2.39/build$
```

6.2. M4-1.4.19

Пакет M4 содержит макропроцессор.	
Приблизительное время сборки:	0.1 SBU
Требуемое дисковое пространство:	31 MB

Распаковываем архив и переходим в каталог с его содержимым

```
tar -xvf m4-1.4.19.tar.xz
cd m4-1.4.19
```

```
m4-1.4.19/tests/vma-iter.c
m4-1.4.19/configure.ac
lfs:/mnt/lfs/sources/m4-1.4.19$
```

6.2.1. Установка пакета M4

Подготовьте пакет M4 к компиляции:

```
time { ./configure --prefix=/usr   \
            --host=$LFS_TGT \
            --build=$(build-aux/config.guess) && time make && make
DESTDIR=$LFS install; }
```

```
lfs:/mnt/lfs/sources/m4-1.4.19$ time { ./configure --prefix=/usr   \
            --host=$LFS_TGT \
            --build=$(build-aux/config.guess) && time make && make DESTDIR=$LFS install; }
```

Вывод окончания компиляции и установки

```
make[2]: Leaving directory '/mnt/lfs/sources/m4-1.4.19/po'
Making install in tests
make[2]: Entering directory '/mnt/lfs/sources/m4-1.4.19/tests'
make  install-recursive
make[3]: Entering directory '/mnt/lfs/sources/m4-1.4.19/tests'
Making install in .
make[4]: Entering directory '/mnt/lfs/sources/m4-1.4.19/tests'
make[5]: Entering directory '/mnt/lfs/sources/m4-1.4.19/tests'
make[5]: Nothing to be done for 'install-exec-am'.
make[5]: Nothing to be done for 'install-data-am'.
make[5]: Leaving directory '/mnt/lfs/sources/m4-1.4.19/tests'
make[4]: Leaving directory '/mnt/lfs/sources/m4-1.4.19/tests'
make[3]: Leaving directory '/mnt/lfs/sources/m4-1.4.19/tests'
make[2]: Leaving directory '/mnt/lfs/sources/m4-1.4.19/tests'
make[1]: Leaving directory '/mnt/lfs/sources/m4-1.4.19'

real    0m22.244s
user    0m26.019s
sys     0m10.200s
lfs:/mnt/lfs/sources/m4-1.4.19$
```

Перейдем в каталог sources и удалим более не нужный разорхивированный каталог

```
cd ..
rm -Rf m4-1.4.19
```

```
lfs:/mnt/lfs/sources/m4-1.4.19$ cd ..
rm -Rf m4-1.4.19
lfs:/mnt/lfs/sources$
```

Подробная информация об этом пакете находится в Разделе 8.12.2. «Содержимое пакета M4.»

6.3. Ncurses-6.4-20230520

Пакет Ncurses содержит библиотеки для независимой от терминала обработки ввода/вывода	
Приблизительное время сборки:	0.3 SBU
Требуемое дисковое пространство:	51 MB

Распаковываем архив и переходим в каталог с его содержимым

```
tar -xvf ncurses-6.4-20230520.tar.xz
cd ncurses-6.4-20230520
```

```
lfs:/mnt/lfs/sources$ tar -xvf ncurses-6.4-20230520.tar.xz
cd ncurses-6.4-20230520
```

6.3.1. Установка пакета Ncurses

Во-первых, убедитесь, что gawk найден первым во время настройки:

```
sed -i s/mawk// configure
```

```
lfs:/mnt/lfs/sources/ncurses-6.4-20230520$ sed -i s/mawk// configure
lfs:/mnt/lfs/sources/ncurses-6.4-20230520$
```

Затем выполните следующие команды, чтобы собрать программу «tic» на хосте сборки:

```
mkdir build
pushd build
  ../configure
  make -C include
  make -C progs tic
popd
```

```
lfs:/mnt/lfs/sources/ncurses-6.4-20230520$ mkdir build
pushd build
  ../configure
  make -C include
  make -C progs tic
popd
```

Результат сборки

```
2 --param max-inline-insns-single=1200  -DNCURSES_STATIC -L../lib  -lncurses -lncurses     -o tic
make: Leaving directory '/mnt/lfs/sources/ncurses-6.4-20230520/build/progs'
/mnt/lfs/sources/ncurses-6.4-20230520
lfs:/mnt/lfs/sources/ncurses-6.4-20230520$
```

Проверим tic

```
cd build
find . -name tic
cd ..
```

```
lfs:/mnt/lfs/sources/ncurses-6.4-20230520$ cd build
find . -name tic
cd ..
./progs/tic
lfs:/mnt/lfs/sources/ncurses-6.4-20230520$
```

Подготовьте Ncurses к компиляции:

```
./configure --prefix=/usr                \
            --host=$LFS_TGT              \
            --build=$(./config.guess)    \
            --mandir=/usr/share/man      \
            --with-manpage-format=normal \
            --with-shared                \
            --without-normal             \
            --with-cxx-shared            \
            --without-debug              \
            --without-ada                \
            --disable-stripping          \
            --enable-widec
```

```
lfs:/mnt/lfs/sources/ncurses-6.4-20230520$ ./configure --prefix=/usr                \
            --host=$LFS_TGT              \
            --build=$(./config.guess)    \
            --mandir=/usr/share/man      \
            --with-manpage-format=normal \
            --with-shared                \
            --without-normal             \
            --with-cxx-shared            \
            --without-debug              \
            --without-ada                \
            --disable-stripping          \
            --enable-widec
```

Результат подготовки к компиляции

```
Appending rules for shared model (test: ticlib+termlib+ext_tinfo+base+widechar+ext_funcs)
creating headers.sh

** Configuration summary for NCURSES 6.4 20230520:

       extended funcs: yes
       xterm terminfo: xterm-new

        bin directory: /usr/bin
        lib directory: /usr/lib
    include directory: /usr/include
        man directory: /usr/share/man
   terminfo directory: /usr/share/terminfo

lfs:/mnt/lfs/sources/ncurses-6.4-20230520$
```

«**Значение параметров настройки:**»

- **–with-manpage-format=normal**

Этот аргумент предотвращает установку Ncurses сжатых страниц руководства, это может произойти, если сам дистрибутив хоста содержит сжатые страницы руководства.

- **–with-shared**

Этот аргумент позволяет Ncurses собирать и устанавливать разделяемые библиотеки C.

- **–without-normal**

Этот аргумент предотвращает сборку и установку статических библиотек C.

- **–without-debug**

Этот аргумент предотвращает сборку и установку отладочных библиотек.

- **–with-cxx-shared**

Это аргумент позволяет Ncurses собирать и устанавливать общие привязки C++. А также предотвращает сборку и установку статических привязок C++.

- **–without-ada**

Этот аргумент гарантирует, что Ncurses будет собран без поддержки компилятора Ada, который может присутствовать на хосте, но будет недоступен, как только мы войдем в среду chroot.

- **–disable-stripping**

Этот аргумент не позволяет системе сборки использовать программу strip с хоста. Использование инструментов хоста в кросс-компилируемой программе может привести к сбою.

- **–enable-widec**

Этот аргумент указывает, что необходимо скомпилировать библиотеки расширенных символов (такие как, libncursesw.so.6.4-20230520) вместо обычных (таких как, libncurses.so.6.4-20230520). Эти библиотеки расширенных символов можно использовать как в многобайтовой, так и традиционной 8-битной локали, в то время как обычные библиотеки корректно работают только в 8-битных локалях. Библиотеки расширенных символов и обычные совместимы на уровне исходного кода, но не совместимы в двоичном.

Скомпилируйте пакет:

```
time make
```

```
lfs:/mnt/lfs/sources/ncurses-6.4-20230520$ time make
```

Результат компиляции

```
make[1]: Leaving directory '/mnt/lfs/sources/ncurses-6.4-20230520/misc'

real    0m8.745s
user    0m38.365s
sys     0m8.789s
lfs:/mnt/lfs/sources/ncurses-6.4-20230520$
```

Установите пакет:

```
make DESTDIR=$LFS TIC_PATH=$(pwd)/build/progs/tic install
```

```
lfs:/mnt/lfs/sources/ncurses-6.4-20230520$ make DESTDIR=$LFS TIC_PATH=$(pwd)/build/progs/tic install
```

Результат установки

```
/usr/bin/install -c ncurses-config /mnt/lfs/usr/bin/ncursesw6-config
make[1]: Leaving directory '/mnt/lfs/sources/ncurses-6.4-20230520/misc'
lfs:/mnt/lfs/sources/ncurses-6.4-20230520$
```

Создадим ссылку

```
ln -sv libncursesw.so $LFS/usr/lib/libncurses.so
```

```
lfs:/mnt/lfs/sources/ncurses-6.4-20230520$ ln -sv libncursesw.so $LFS/usr/lib/libncurses.so
'/mnt/lfs/usr/lib/libncurses.so' -> 'libncursesw.so'
lfs:/mnt/lfs/sources/ncurses-6.4-20230520$
```

Отредактируем заголовочный файл

```
sed -e 's/^#if.*XOPEN.*$/#if 1/' \
    -i $LFS/usr/include/curses.h
```

```
'/mnt/lfs/usr/lib/libncurses.so' -> 'libncursesw.so'
lfs:/mnt/lfs/sources/ncurses-6.4-20230520$ sed -e 's/^#if.*XOPEN.*$/#if 1/' \
    -i $LFS/usr/include/curses.h
lfs:/mnt/lfs/sources/ncurses-6.4-20230520$
```

«**Значение параметров настройки:**»

- **TIC_PATH=$(pwd)/build/progs/tic**

Нам нужно передать путь до только что собранной программы tic, которая работает на сборочной машине, чтобы база данных терминала была создана без ошибок.

- **ln -sv libncursesw.so $LFS/usr/lib/libncurses.so**

Библиотека libncurses.so необходима для нескольких пакетов, которые мы скоро соберем. Мы создаем эту символическую ссылку, чтобы использовать libncursesw.so в качестве замены.

- **sed -e 's/^#if.*XOPEN.*$/#if 1/' ...**

Заголовочный файл curses.h содержит определения различных структур данных Ncurses. С разными определениями макросов препроцессора могут использоваться два разных набора определений структуры данных: 8-битное определение совместимо с libncurses.so, а определение расширенного набора символов совместимо с libncursesw.so . Поскольку мы используем libncursesw.so вместо libncurses.so, отредактируйте заголовочный файл, чтобы он всегда использовал определение структуры данных расширенного набора символов, совместимое с . libncursesw.so.

Перейдем в каталог sources и удалим более не нужный разорхивированный каталог

```
cd ..
rm -Rf ncurses-6.4-20230520
```

```
lfs:/mnt/lfs/sources/ncurses-6.4-20230520$ cd ..
rm -Rf ncurses-6.4-20230520
lfs:/mnt/lfs/sources$
```

Подробная информация об этом пакете находится в Разделе 8.29.2. «Содержимое пакета Ncurses.»

6.4. Bash-5.2.21

Пакет Bash содержит Bourne-Again Shell.	
Приблизительное время сборки:	0.2 SBU
Требуемое дисковое пространство:	67 MB

В Bash — и многих других языках программирования — **&&** означает **«И»**. И в контексте выполнения команды, это означает, что элементы слева и справа от **&&** должны быть запущены последовательно в этом случае. Каждая следующая команда выполняется только при успешном выполнении предыдущей.

Соберем распаковку архива, подготовку к компиляции, компиляцию и установку, создание ссылки, выход и удаление временного каталога в воедино

```
tar -xvf bash-5.2.21.tar.gz && cd bash-5.2.21 &&
time { ./configure --prefix=/usr                \
            --build=$(sh support/config.guess) \
            --host=$LFS_TGT                    \
            --without-bash-malloc && time make && make DESTDIR=$LFS install;
} &&
ln -sv bash $LFS/bin/sh &&
cd .. &&
rm -Rf bash-5.2.21
```

```
lfs:/mnt/lfs/sources$ tar -xvf bash-5.2.21.tar.gz && cd bash-5.2.21 &&
time { ./configure --prefix=/usr                \
           --build=$(sh support/config.guess) \
           --host=$LFS_TGT                    \
           --without-bash-malloc && time make && make DESTDIR=$LFS install; } &&
ln -sv bash $LFS/bin/sh &&
cd .. &&
rm -Rf bash-5.2.21
```

Вывод выполнения нашей команды с созданием ссылки, выходом и удалением временного каталога.

```
stat
getconf
make[1]: Leaving directory '/mnt/lfs/sources/bash-5.2.21/examples/loadables'

real    0m25.515s
user    0m49.712s
sys     0m12.912s
'/mnt/lfs/bin/sh' -> 'bash'
lfs:/mnt/lfs/sources$
```

Если вы успешно приминили этот метод то пропустите следующие команды установки bash и переходите к установке 6.5. Coreutils-9.4

6.4.1. Установка пакета Bash

Подготовьте Bash к компиляции:

```
./configure --prefix=/usr                       \
            --build=$(sh support/config.guess) \
            --host=$LFS_TGT                    \
            --without-bash-malloc
```

Значение параметров настройки:

- **–without-bash-malloc**

Этот параметр отключает использование функции распределения памяти (malloc) Bash, которая, как известно, вызывает ошибки сегментации. Если опция отключена, Bash будет

использовать функции malloc из Glibc, которые более стабильны.

Скомпилируйте пакет:

```
make
```

Установите пакет:

```
make DESTDIR=$LFS install
```

Создайте символическую ссылку для программ, которые используют sh как оболочку:

```
ln -sv bash $LFS/bin/sh
```

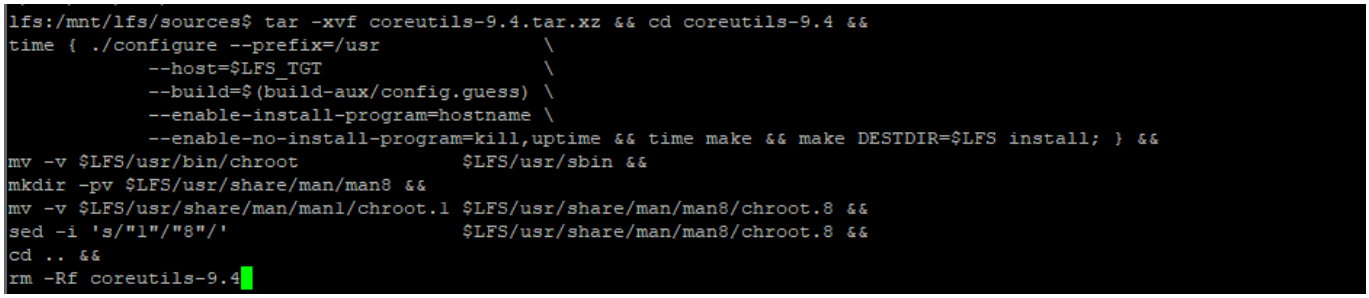
Подробная информация об этом пакете находится в Разделе 8.35.2. «Содержимое пакета Bash.»

6.5. Coreutils-9.4

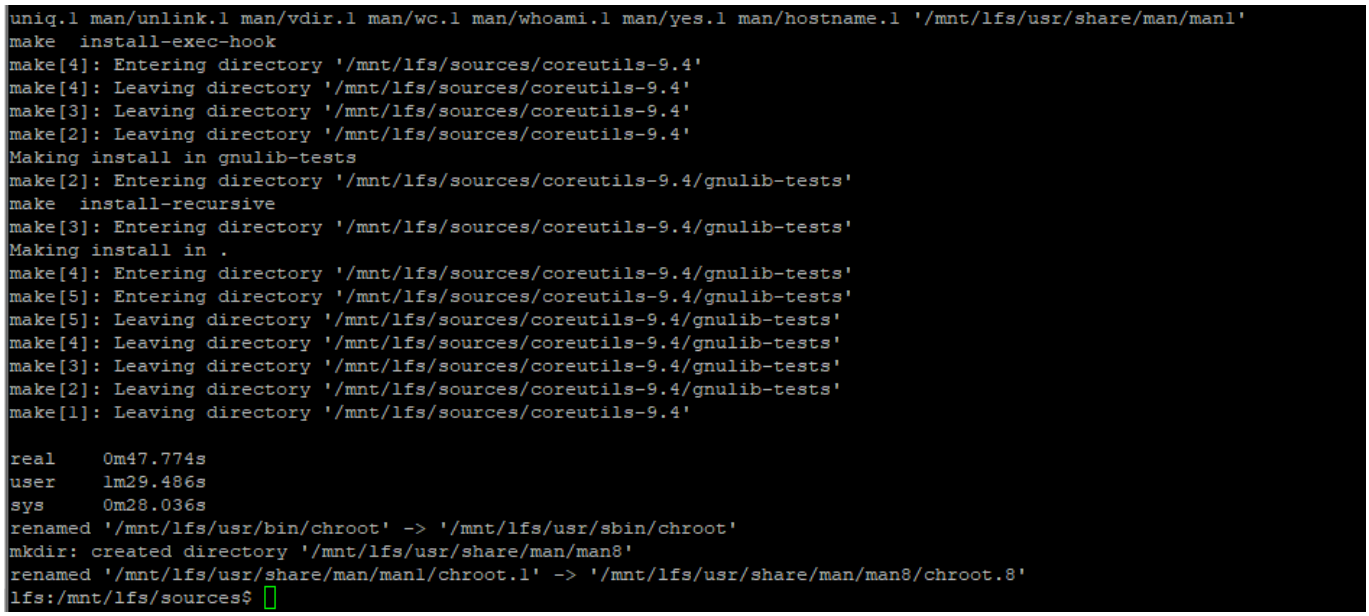
Пакет Coreutils содержит основные утилиты, необходимые каждой операционной системе.	
Приблизительное время сборки:	0.3 SBU
Требуемое дисковое пространство:	173 MB

Соберем распаковку архива, подготовку к компиляции, компиляцию и установку, создание ссылки, выход и удаление временного каталога в воедино

```
tar -xvf coreutils-9.4.tar.xz && cd coreutils-9.4 &&
time { ./configure --prefix=/usr              \
            --host=$LFS_TGT                   \
            --build=$(build-aux/config.guess) \
            --enable-install-program=hostname \
            --enable-no-install-program=kill,uptime && time make && make
DESTDIR=$LFS install; } &&
mv -v $LFS/usr/bin/chroot              $LFS/usr/sbin &&
mkdir -pv $LFS/usr/share/man/man8 &&
mv -v $LFS/usr/share/man/man1/chroot.1 $LFS/usr/share/man/man8/chroot.8 &&
sed -i 's/"1"/"8"/'                    $LFS/usr/share/man/man8/chroot.8 &&
cd .. &&
rm -Rf coreutils-9.4
```



Вывод выполнения нашей команды с созданием ссылки, выходом и удалением временного каталога.



Если вы успешно приминили этот метод то пропустите следующие команды установки bash и переходите к установке 6.6. Diffutils-3.10

6.5.1. Установка пакета Coreutils

Подготовьте Coreutils к компиляции:

```
./configure --prefix=/usr                     \
            --host=$LFS_TGT                   \
            --build=$(build-aux/config.guess) \
            --enable-install-program=hostname \
            --enable-no-install-program=kill,uptime
```

Значение параметров настройки:

- **–enable-install-program=hostname**

Этот параметр позволяет создать и установить двоичный файл hostname – по умолчанию он отключен, но требуется для набора тестов Perl.

Скомпилируйте пакет:

```
make
```

Установите пакет:

```
make DESTDIR=$LFS install
```

Переместите программы в их конечное местоположение. Хотя во временной среде в этом нет необходимости, мы должны это сделать, потому что некоторые программы жестко прописывают местоположение исполняемых файлов:

```
mv -v $LFS/usr/bin/chroot              $LFS/usr/sbin
mkdir -pv $LFS/usr/share/man/man8
mv -v $LFS/usr/share/man/man1/chroot.1 $LFS/usr/share/man/man8/chroot.8
sed -i 's/"1"/"8"/'                    $LFS/usr/share/man/man8/chroot.8
```

Подробная информация об этом пакете находится в Разделе 8.57.2. «Содержимое пакета Coreutils.»

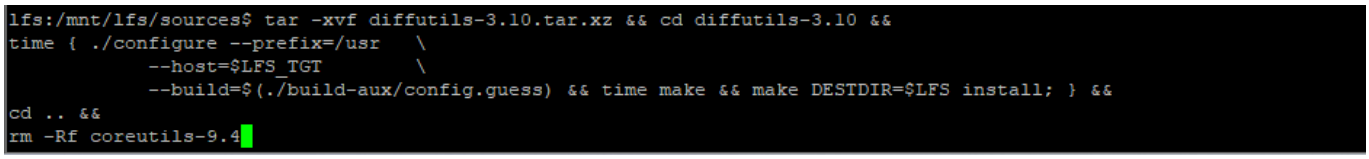
6.6. Diffutils-3.10

Пакет Diffutils содержит программы, которые показывают различия между файлами или каталогами.	
Приблизительное время сборки:	0.1 SBU
Требуемое дисковое пространство:	29 MB

6.6.1. Установка пакета Diffutils

Разорхивируем,подготовим Diffutils для компиляции, скомпилируем, установите пакет и удалим временный каталог:

```
tar -xvf diffutils-3.10.tar.xz && cd diffutils-3.10 &&
time { ./configure --prefix=/usr   \
            --host=$LFS_TGT        \
            --build=$(./build-aux/config.guess) && time make && make
DESTDIR=$LFS install; } &&
cd .. &&
rm -Rf coreutils-9.4
```



Результат выполнения команд

```
else \
  : ; \
fi
make[1]: Leaving directory '/mnt/lfs/sources/diffutils-3.10/po'
Making install in gnulib-tests
make[1]: Entering directory '/mnt/lfs/sources/diffutils-3.10/gnulib-tests'
make  install-recursive
make[2]: Entering directory '/mnt/lfs/sources/diffutils-3.10/gnulib-tests'
Making install in .
make[3]: Entering directory '/mnt/lfs/sources/diffutils-3.10/gnulib-tests'
make[4]: Entering directory '/mnt/lfs/sources/diffutils-3.10/gnulib-tests'
make[4]: Nothing to be done for 'install-exec-am'.
make[4]: Nothing to be done for 'install-data-am'.
make[4]: Leaving directory '/mnt/lfs/sources/diffutils-3.10/gnulib-tests'
make[3]: Leaving directory '/mnt/lfs/sources/diffutils-3.10/gnulib-tests'
make[2]: Leaving directory '/mnt/lfs/sources/diffutils-3.10/gnulib-tests'
make[1]: Leaving directory '/mnt/lfs/sources/diffutils-3.10/gnulib-tests'
make[1]: Entering directory '/mnt/lfs/sources/diffutils-3.10'
make[2]: Entering directory '/mnt/lfs/sources/diffutils-3.10'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/mnt/lfs/sources/diffutils-3.10'
make[1]: Leaving directory '/mnt/lfs/sources/diffutils-3.10'

real    0m18.979s
user    0m21.948s
sys     0m8.655s
lfs:/mnt/lfs/sources$
```

Подробная информация об этом пакете находится в Разделе 8.59.2. «Содержимое пакета Diffutils.»

6.7. File-5.45

Пакет File содержит утилиту для определения типа указанного файла или файлов.	
Приблизительное время сборки:	0.1 SBU
Требуемое дисковое пространство:	37 MB

Распаковываем архив и переходим в каталог с его содержимым

```
tar -xvf file-5.45.tar.gz
cd file-5.45
```

```
file-5.45/m4/ltversion.m4
file-5.45/m4/lt~obsolete.m4
file-5.45/m4/visibility.m4
lfs:/mnt/lfs/sources/file-5.45$
```

6.7.1. Установка пакета File

Команда file на хосте сборки должна быть той же версии, что и собираемая, чтобы создать файл подписи. Выполните следующие команды, чтобы создать временную копию команды file.

```
mkdir build
pushd build
  ../configure --disable-bzlib      \
               --disable-libseccomp \
               --disable-xzlib      \
               --disable-zlib
  make
popd
```

```
lfs:/mnt/lfs/sources/file-5.45$ mkdir build
pushd build
  ../configure --disable-bzlib      \
               --disable-libseccomp \
               --disable-xzlib      \
               --disable-zlib
  make
popd
```

Результат создания временной копии

```
make[2]: Leaving directory '/mnt/lfs/sources/file-5.45/build/doc'
Making all in python
make[2]: Entering directory '/mnt/lfs/sources/file-5.45/build/python'
make[2]: Nothing to be done for 'all'.
make[2]: Leaving directory '/mnt/lfs/sources/file-5.45/build/python'
make[2]: Entering directory '/mnt/lfs/sources/file-5.45/build'
make[2]: Leaving directory '/mnt/lfs/sources/file-5.45/build'
make[1]: Leaving directory '/mnt/lfs/sources/file-5.45/build'
/mnt/lfs/sources/file-5.45
lfs:/mnt/lfs/sources/file-5.45$
```

Значение параметров настройки:

- **-disable-**

Сценарий конфигурации пытается использовать некоторые пакеты из основного дистрибутива, если существуют соответствующие файлы библиотек. Это может привести к сбою компиляции, если файлы библиотек существует, но отсутствуют соответствующие заголовочные файлы. Эти параметры предотвращают использование ненужных возможностей хоста.

```
find . -name file
```

```
lfs:/mnt/lfs/sources/file-5.45$ find . -name file
./build/src/file
./build/src/.libs/file
lfs:/mnt/lfs/sources/file-5.45$
```

Подготовьте файл для компиляции:

```
./configure --prefix=/usr --host=$LFS_TGT --build=$(./config.guess)
```

```
lfs:/mnt/lfs/sources/file-5.45$ ./configure --prefix=/usr --host=$LFS_TGT --build=$(./config.guess)
```

Результат подготовки

```
config.status: creating config.h
config.status: executing depfiles commands
config.status: executing libtool commands
lfs:/mnt/lfs/sources/file-5.45$
```

Скомпилируйте пакет:

```
make FILE_COMPILE=$(pwd)/build/src/file
```

```
lfs:/mnt/lfs/sources/file-5.45$ make FILE_COMPILE=$(pwd)/build/src/file
```

Результат компиляции

```
make[2]: Leaving directory '/mnt/lfs/sources/file-5.45/doc'
Making all in python
make[2]: Entering directory '/mnt/lfs/sources/file-5.45/python'
make[2]: Nothing to be done for 'all'.
make[2]: Leaving directory '/mnt/lfs/sources/file-5.45/python'
make[2]: Entering directory '/mnt/lfs/sources/file-5.45'
make[2]: Leaving directory '/mnt/lfs/sources/file-5.45'
make[1]: Leaving directory '/mnt/lfs/sources/file-5.45'
lfs:/mnt/lfs/sources/file-5.45$
```

Установите пакет:

```
make DESTDIR=$LFS install
```

```
lfs:/mnt/lfs/sources/file-5.45$ make DESTDIR=$LFS install
```

Результат установки

```
make[2]: Leaving directory '/mnt/lfs/sources/file-5.45/tests'
make[1]: Leaving directory '/mnt/lfs/sources/file-5.45/tests'
Making install in doc
make[1]: Entering directory '/mnt/lfs/sources/file-5.45/doc'
make[2]: Entering directory '/mnt/lfs/sources/file-5.45/doc'
make[2]: Nothing to be done for 'install-exec-am'.
 /usr/bin/mkdir -p '/mnt/lfs/usr/share/man/man1'
 /usr/bin/mkdir -p '/mnt/lfs/usr/share/man/man3'
 /usr/bin/mkdir -p '/mnt/lfs/usr/share/man/man4'
 /usr/bin/mkdir -p '/mnt/lfs/usr/share/man/man5'
 /usr/bin/install -c -m 644 file.1 '/mnt/lfs/usr/share/man/man1'
 /usr/bin/install -c -m 644 libmagic.3 '/mnt/lfs/usr/share/man/man3'
 /usr/bin/install -c -m 644 magic.4 '/mnt/lfs/usr/share/man/man4'
make[2]: Leaving directory '/mnt/lfs/sources/file-5.45/doc'
make[1]: Leaving directory '/mnt/lfs/sources/file-5.45/doc'
Making install in python
make[1]: Entering directory '/mnt/lfs/sources/file-5.45/python'
make[2]: Entering directory '/mnt/lfs/sources/file-5.45/python'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/mnt/lfs/sources/file-5.45/python'
make[1]: Leaving directory '/mnt/lfs/sources/file-5.45/python'
make[1]: Entering directory '/mnt/lfs/sources/file-5.45'
make[2]: Entering directory '/mnt/lfs/sources/file-5.45'
make[2]: Nothing to be done for 'install-data-am'.
 /usr/bin/mkdir -p '/mnt/lfs/usr/lib/pkgconfig'
 /usr/bin/install -c -m 644 libmagic.pc '/mnt/lfs/usr/lib/pkgconfig'
make[2]: Leaving directory '/mnt/lfs/sources/file-5.45'
make[1]: Leaving directory '/mnt/lfs/sources/file-5.45'
lfs:/mnt/lfs/sources/file-5.45$
```

Удалите архивный файл libtool, поскольку он потенциально опасен при кросс-компиляции:

```
rm -v $LFS/usr/lib/libmagic.la
```

```
lfs:/mnt/lfs/sources/file-5.45$ rm -v $LFS/usr/lib/libmagic.la
removed '/mnt/lfs/usr/lib/libmagic.la'
lfs:/mnt/lfs/sources/file-5.45$
```

Перейдем в каталог sources и удалим более не нужный разорхивированный каталог

```
cd ..
rm -Rf file-5.45
```

```
lfs:/mnt/lfs/sources/file-5.45$ cd ..
rm -Rf file-5.45
lfs:/mnt/lfs/sources$
```

Подробная информация об этом пакете находится в Разделе 8.10.2. «Содержимое пакета File.»

6.8. Findutils-4.9.0

Пакет Findutils содержит программы для поиска файлов. Эти программы предназначены для поиска по всем файлам в дереве каталогов, а также для создания, обслуживания и поиска в базе данных (часто быстрее, чем рекурсивный поиск, но ненадежно, если база данных давно не обновлялась).Findutils также предоставляет программу xargs, которую можно использовать для запуска указанной команды для каждого файла, выбранного при поиске.	
Приблизительное время сборки:	0.1 SBU
Требуемое дисковое пространство:	42 MB

6.8.1. Установка пакета Findutils

Разорхивируем,подготовим файл для компиляции, скомпилируем, установите пакет и удалим временный каталог:

```
tar -xvf findutils-4.9.0.tar.xz && cd findutils-4.9.0 &&
time { ./configure --prefix=/usr             \
            --localstatedir=/var/lib/locate \
            --host=$LFS_TGT                 \
            --build=$(build-aux/config.guess) && time make && make
DESTDIR=$LFS install; } &&
cd .. &&
rm -Rf findutils-4.9.0
```

```
lfs:/mnt/lfs/sources/findutils-4.9.0$ tar -xvf findutils-4.9.0.tar.xz && cd findutils-4.9.0 &&
time { ./configure --prefix=/usr             \
           --localstatedir=/var/lib/locate \
           --host=$LFS_TGT                 \
           --build=$(build-aux/config.guess) && time make && make DESTDIR=$LFS install; } &&
cd .. &&
rm -Rf findutils-4.9.0
```

Результат выполнения команд

```
make  install-recursive
make[2]: Entering directory '/mnt/lfs/sources/findutils-4.9.0/gnulib-tests'
Making install in .
make[3]: Entering directory '/mnt/lfs/sources/findutils-4.9.0/gnulib-tests'
make[4]: Entering directory '/mnt/lfs/sources/findutils-4.9.0/gnulib-tests'
make[4]: Nothing to be done for 'install-exec-am'.
make[4]: Nothing to be done for 'install-data-am'.
make[4]: Leaving directory '/mnt/lfs/sources/findutils-4.9.0/gnulib-tests'
make[3]: Leaving directory '/mnt/lfs/sources/findutils-4.9.0/gnulib-tests'
make[2]: Leaving directory '/mnt/lfs/sources/findutils-4.9.0/gnulib-tests'
make[1]: Leaving directory '/mnt/lfs/sources/findutils-4.9.0/gnulib-tests'
make[1]: Entering directory '/mnt/lfs/sources/findutils-4.9.0'
make[2]: Entering directory '/mnt/lfs/sources/findutils-4.9.0'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/mnt/lfs/sources/findutils-4.9.0'
make[1]: Leaving directory '/mnt/lfs/sources/findutils-4.9.0'

real    0m28.315s
user    0m30.517s
sys     0m12.794s
lfs:/mnt/lfs/sources$
```

Подробная информация об этом пакете находится в Разделе 8.61.2. «Содержимое пакета Findutils.»

6.9. Gawk-5.3.0

Пакет Gawk содержит программы для работы с текстовыми файлами.	
Приблизительное время сборки:	0.1 SBU
Требуемое дисковое пространство:	47 MB

6.9.1. Установка пакета Gawk

Разорхивируем и убедимся, что некоторые ненужные файлы не будут установлены:

```
tar -xvf gawk-5.3.0.tar.xz && cd gawk-5.3.0 &&
sed -i 's/extras//' Makefile.in
```

```
lfs:/mnt/lfs/sources$ tar -xvf gawk-5.3.0.tar.xz && cd gawk-5.3.0 && sed -i 's/extras//' Makefile.in
```

Результат выполнения команды

```
gawk-5.3.0/custom.h
gawk-5.3.0/main.c
gawk-5.3.0/floatcomp.c
gawk-5.3.0/replace.c
lfs:/mnt/lfs/sources/gawk-5.3.0$
```

подготовим файл для компиляции, скомпилируем, установите пакет и удалим временный каталог:

```
time { ./configure --prefix=/usr   \
            --host=$LFS_TGT        \
            --build=$(build-aux/config.guess) && time make && make
DESTDIR=$LFS install; } &&
cd .. &&
rm -Rf gawk-5.3.0
```

```
lfs:/mnt/lfs/sources/gawk-5.3.0$ time { ./configure --prefix=/usr   \
            --host=$LFS_TGT        \
            --build=$(build-aux/config.guess) && time make && make DESTDIR=$LFS install; } &&
cd .. &&
rm -Rf gawk-5.3.0
```

Результат выполнения команд

```
make[2]: Leaving directory '/mnt/lfs/sources/gawk-5.3.0/po'
Making install in test
make[2]: Entering directory '/mnt/lfs/sources/gawk-5.3.0/test'
make[3]: Entering directory '/mnt/lfs/sources/gawk-5.3.0/test'
make[3]: Nothing to be done for 'install-exec-am'.
make[3]: Nothing to be done for 'install-data-am'.
make[3]: Leaving directory '/mnt/lfs/sources/gawk-5.3.0/test'
make[2]: Leaving directory '/mnt/lfs/sources/gawk-5.3.0/test'
make[1]: Leaving directory '/mnt/lfs/sources/gawk-5.3.0'

real    0m18.420s
user    0m30.747s
sys     0m7.601s
lfs:/mnt/lfs/sources$
```

Подробная информация об этом пакете находится в Разделе 8.60.2. «Содержимое пакета Gawk.»

6.10. Grep-3.11

Пакет Grep содержит программы для поиска по содержимому файлов.	
Приблизительное время сборки:	0.1 SBU
Требуемое дисковое пространство:	27 MB

6.10.1. Установка пакета Grep

Разорхивируем,подготовим файл для компиляции, скомпилируем, установите пакет и удалим временный каталог:

```
tar -xvf grep-3.11.tar.xz && cd grep-3.11 &&
time { ./configure --prefix=/usr   \
            --host=$LFS_TGT \
            --build=$(./build-aux/config.guess) && time make && make
DESTDIR=$LFS install; } &&
cd .. &&
rm -Rf grep-3.11
```

```
lfs:/mnt/lfs/sources$ tar -xvf grep-3.11.tar.xz && cd grep-3.11 &&
time { ./configure --prefix=/usr   \
            --host=$LFS_TGT \
            --build=$(./build-aux/config.guess) && time make && make DESTDIR=$LFS install; } &&
cd .. &&
rm -Rf grep-3.11
```

Результат выполнения команд

```
make[3]: Entering directory '/mnt/lfs/sources/grep-3.11/gnulib-tests'
make[4]: Entering directory '/mnt/lfs/sources/grep-3.11/gnulib-tests'
make[4]: Nothing to be done for 'install-exec-am'.
make[4]: Nothing to be done for 'install-data-am'.
make[4]: Leaving directory '/mnt/lfs/sources/grep-3.11/gnulib-tests'
make[3]: Leaving directory '/mnt/lfs/sources/grep-3.11/gnulib-tests'
make[2]: Leaving directory '/mnt/lfs/sources/grep-3.11/gnulib-tests'
make[1]: Leaving directory '/mnt/lfs/sources/grep-3.11/gnulib-tests'
make[1]: Entering directory '/mnt/lfs/sources/grep-3.11'
make[2]: Entering directory '/mnt/lfs/sources/grep-3.11'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/mnt/lfs/sources/grep-3.11'
make[1]: Leaving directory '/mnt/lfs/sources/grep-3.11'

real    0m20.227s
user    0m23.303s
sys     0m8.861s
lfs:/mnt/lfs/sources$
```

Подробная информация об этом пакете находится в Разделе 8.34.2. «Содержимое пакета Grep.»

6.11. Gzip-1.13

Пакет Gzip содержит программы для сжатия и распаковки файлов.	
Приблизительное время сборки:	0.1 SBU
Требуемое дисковое пространство:	11 MB

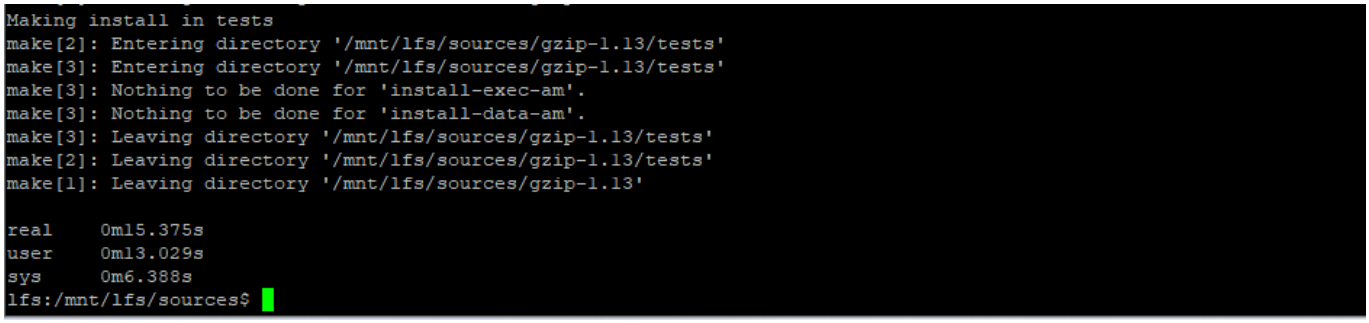
6.11.1. Установка пакета Gzip

Разорхивируем,подготовим файл для компиляции, скомпилируем, установите пакет и удалим временный каталог:

```
tar -xvf gzip-1.13.tar.xz && cd gzip-1.13 &&
time { ./configure --prefix=/usr --host=$LFS_TGT && time make && make
DESTDIR=$LFS install; } &&
cd .. &&
rm -Rf gzip-1.13
```

```
lfs:/mnt/lfs/sources$ tar -xvf gzip-1.13.tar.xz && cd gzip-1.13 &&
time { ./configure --prefix=/usr --host=$LFS_TGT && time make && make DESTDIR=$LFS install; } &&
cd .. &&
rm -Rf gzip-1.13
```

Результат выполнения команд



Подробная информация об этом пакете находится в Разделе 8.64.2. «Содержимое пакета Gzip.»

6.12. Make-4.4.1

Пакет Make содержит программу, управляющую генерацией исполняемых и других файлов, из исходного кода.	
Приблизительное время сборки:	0.1 SBU
Требуемое дисковое пространство:	15 MB

6.12.1. Установка пакета Make

Подготовьте Make к компиляции:

```
./configure --prefix=/usr   \
            --without-guile \
            --host=$LFS_TGT \
            --build=$(build-aux/config.guess)
```

Значение новой опции настройки:

- **–without-guile**

Несмотря на то, что мы выполняем кросс-компиляцию, configure пытается использовать guile с узла сборки, если он его находит. Это приводит к сбою компиляции, этот аргумент предотвращает его использование.

Скомпилируйте пакет:

```
make
```

Установите пакет:

```
make DESTDIR=$LFS install
```

Подробная информация об этом пакете находится в Разделе 8.68.2. «Содержимое пакета Make.»

6.13. Patch-2.7.6

Пакет Patch содержит программу для изменения или создания файлов путём наложение «патча», обычно, создаваемого программой diff.	
Приблизительное время сборки:	0.1 SBU
Требуемое дисковое пространство:	12 MB

6.13.1. Установка пакета Patch

Подготовьте Patch к компиляции:

```
./configure --prefix=/usr   \
            --host=$LFS_TGT \
            --build=$(build-aux/config.guess)
```

Скомпилируйте пакет:

```
make
```

Установите пакет:

```
make DESTDIR=$LFS install
```

Подробная информация об этом пакете находится в Разделе 8.69.2. «Содержимое пакета Patch.»

6.14. Sed-4.9

Пакет Sed содержит потоковый редактор текста	
Приблизительное время сборки:	0.1 SBU
Требуемое дисковое пространство:	21 MB

6.14.1. Установка пакета Sed

Подготовьте Sed к компиляции:

```
./configure --prefix=/usr   \
            --host=$LFS_TGT \
            --build=$(./build-aux/config.guess)
```

Скомпилируйте пакет:

```
make
```

Установите пакет:

```
make DESTDIR=$LFS install
```

Подробная информация об этом пакете находится в Разделе 8.30.2. «Содержимое пакета Sed.»

6.15. Tar-1.35

Пакет Tar предоставляет возможность создавать tar архивы, а также производить с ними различные манипуляции. Tar может распаковать предварительно созданный архив, добавить или обновить файлы в нём, вернуть список файлов в архиве.	
Приблизительное время сборки:	0.1 SBU
Требуемое дисковое пространство:	42 MB

6.15.1. Установка пакета Tar

Подготовьте Tar к компиляции:

```
./configure --prefix=/usr                     \
            --host=$LFS_TGT                   \
            --build=$(build-aux/config.guess)
```

Скомпилируйте пакет:

```
make
```

Установите пакет:

```
make DESTDIR=$LFS install
```

Подробная информация об этом пакете находится в Разделе 8.70.2. «Содержимое пакета Tar.»

6.16. Xz-5.4.6

Пакет Xz содержит программы для сжатия и распаковки файлов. Он предоставляет возможности для lzma и более новых форматов сжатия xz. Сжатие текстовых файлов с помощью xz дает лучший процент сжатия, чем с традиционные gzip или bzip2.	
Приблизительное время сборки:	0.1 SBU
Требуемое дисковое пространство:	22 MB

6.16.1. Установка пакета Xz

Подготовьте Xz к компиляции:

```
./configure --prefix=/usr                     \
            --host=$LFS_TGT                   \
            --build=$(build-aux/config.guess) \
            --disable-static                  \
```

```
            --docdir=/usr/share/doc/xz-5.4.6
```

Скомпилируйте пакет:

```
make
```

Установите пакет:

```
make DESTDIR=$LFS install
```

Удалите архивный файл libtool, поскольку он потенциально опасен при кросс-компиляции:

```
rm -v $LFS/usr/lib/liblzma.la
```

Подробная информация об этом пакете находится в Разделе 8.8.2. «Содержимое пакета Xz.»

6.17. Binutils-2.42 - Проход 2

Пакет Binutils содержит компоновщик, ассемблер и другие инструменты для работы с объектными файлами.	
Приблизительное время сборки:	0.5 SBU
Требуемое дисковое пространство:	537 MB

6.17.1. Установка пакета Binutils

Система сборки Binutils использует содержащуюся в пакете копию libtool для линковки с внутренними статическими библиотеками, но копии libiberty и zlib, поставляемые с пакетом, не используют libtool. Это несоответствие может привести к тому, что созданные двоичные файлы будут ошибочно связаны с библиотеками из основного дистрибутива. Решение этой проблемы:

```
sed '6009s/$add_dir//' -i ltmain.sh
```

Создайте отдельный каталог для сборки:

```
mkdir -v build
cd       build
```

Подготовьте Binutils к компиляции:

```
../configure                   \
    --prefix=/usr              \
    --build=$(../config.guess) \
    --host=$LFS_TGT            \
    --disable-nls              \
    --enable-shared            \
    --enable-gprofng=no        \
    --disable-werror           \
```

```
    --enable-64-bit-bfd        \
    --enable-default-hash-style=gnu
```

Значение новых параметров настройки:

- **–enable-shared**

Собирает libbfd как разделяемую библиотеку

- **–enable-64-bit-bfd**

Включает 64-разрядную поддержку (на хостах с меньшим размером слова). В 64-разрядных системах это может и не понадобиться, но вреда от этого не будет

Скомпилируйте пакет:

```
make
```

Установите пакет:

```
make DESTDIR=$LFS install
```

Удалите архивные файлы libtool, поскольку они потенциально опасны при кросс-компиляции, также удалите ненужные статические библиотеки

```
rm -v $LFS/usr/lib/lib{bfd,ctf,ctf-nobfd,opcodes,sframe}.{a,la}
```

Подробная информация об этом пакете находится в Разделе 8.19.2. «Содержимое пакета Binutils.»

6.18. GCC-13.2.0 - Проход 2

Пакет GCC содержит коллекцию компиляторов GNU, которая включает компиляторы C и C++.	
Приблизительное время сборки:	4.4 SBU
Требуемое дисковое пространство:	4.8 GB

6.18.1. Установка пакета GCC

Как и при первой сборке GCC, требуются пакеты GMP, MPFR и MPC. Распакуйте архивы и переименуйте каталоги:

```
tar -xf ../mpfr-4.2.1.tar.xz
mv -v mpfr-4.2.1 mpfr
tar -xf ../gmp-6.3.0.tar.xz
mv -v gmp-6.3.0 gmp
tar -xf ../mpc-1.3.1.tar.gz
mv -v mpc-1.3.1 mpc
```

При сборке на x86_64 измените имя каталога по умолчанию для 64-разрядных библиотек на «lib»:

```
case $(uname -m) in
  x86_64)
    sed -e '/m64=/s/lib64/lib/' \
        -i.orig gcc/config/i386/t-linux64
  ;;
esac
```

Переопределите правила сборки заголовочных файлов libgcc и libstdc++, чтобы разрешить создание этих библиотек с поддержкой потоков POSIX:

```
sed '/thread_header =/s/@.*@/gthr-posix.h/' \
    -i libgcc/Makefile.in libstdc++-v3/include/Makefile.in
```

Снова создайте отдельный каталог сборки:

```
mkdir -v build
cd       build
```

Перед началом сборки GCC не забудьте отключить все переменные среды, которые переопределяют флаги оптимизации по умолчанию.

Теперь подготовьте GCC к компиляции:

```
../configure                                       \
    --build=$(../config.guess)                     \
    --host=$LFS_TGT                                \
    --target=$LFS_TGT                              \
    LDFLAGS_FOR_TARGET=-L$PWD/$LFS_TGT/libgcc      \
    --prefix=/usr                                  \
    --with-build-sysroot=$LFS                      \
    --enable-default-pie                           \
    --enable-default-ssp                           \
    --disable-nls                                  \
    --disable-multilib                             \
    --disable-libatomic                            \
    --disable-libgomp                              \
    --disable-libquadmath                          \
    --disable-libsanitizer                         \
    --disable-libssp                               \
    --disable-libvtv                               \
    --enable-languages=c,c++
```

Значение новых параметров настройки:

- **–with-build-sysroot=$LFS**

Обычно, использование –host гарантирует, что для сборки GCC используется кросс-компилятор, и этот компилятор знает, что он должен искать заголовочные файлы и библиотеки в $LFS. Но сборочная система GCC использует другие инструменты, которые не знают об этом местоположении. Этот параметр необходим для того, чтобы они могли найти нужные файлы в $LFS, а не на хосте.

- **–target=$LFS_TGT**

Поскольку мы выполняем кросс-компиляцию GCC, невозможно создать целевые библиотеки (libgcc и libstdc++) с ранее скомпилированными двоичными файлами GCC, потому что эти двоичные файлы не будут работать на хост-дистрибутиве. Система сборки GCC по умолчанию попытается использовать компиляторы C и C++ хоста в качестве обходного пути. Сейчас не поддерживается создание целевых библиотек GCC с помощью другой версии GCC, поэтому использование компиляторов хоста может привести к сбою сборки. Этот параметр гарантирует сборку библиотек с помощью GCC собранного на первом проходе.

- **LDFLAGS_FOR_TARGET=…**

Разрешить libstdc++ использовать общую библиотеку libgcc, собранную на этом этапе, вместо статической версии, собранной в GCC Проход 1. Это необходимо для поддержки обработки исключений C++

- **–disable-libsanitizer**

Отключает библиотеки среды выполнения GCC sanitizer. Они не нужны для временного набора инструментов. Этот параметр необходим для сборки GCC без установки libcrypt для целевого объекта. В GCC-Проход 1 это решалось с помощью параметра –disable-libstdcxx, но теперь мы должны передать его явно.

Скомпилируйте пакет:

```
make
```

Установите пакет:

```
make DESTDIR=$LFS install
```

В качестве завершающего штриха создайте символическую ссылку на утилиту. Многие программы и скрипты используют cc вместо gcc, чтобы сделать программы более универсальными и, следовательно, для совместимости со всеми типами UNIX-систем, где компилятор GNU C не всегда установлен. Наличие cc оставляет системному администратору право самостоятельно решать, какой компилятор C устанавливать:

```
ln -sv gcc $LFS/usr/bin/cc
```

Подробная информация об этом пакете находится в Разделе 8.28.2. «Содержимое пакета GCC.»

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

Permanent link:
https://wwoss.ru/doku.php?id=software:linux_server:lfs-example:chapter06&rev=1721030375

Last update: **2024/07/15 10:59**