



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«МИРЭА – Российский технологический университет»
РТУ МИРЭА**

Институт информационных технологий
Кафедра Математического обеспечения и стандартизации
информационных технологий

Отчет по практической работе №3

по дисциплине
«Системное программное обеспечение»

Тема практической работы:
«Docker»

Выполнил студент группы ИВБО-02-19

Проверил ассистент

К. Ю. Денисов

Ю. А. Вороноцов

Москва 2021

Содержание

1	Ход работы	3
1.1	Установка нового пакета	3
1.2	Удаление пакета	3
1.3	Поиск пакета и вывод информации о нем	3
2	Добавление репозиториев	4
3	Сборка пакетов из исходных кодов	6
4	Создание собственного deb-пакета	6
4.1	Подготовка окружения	6
4.2	Описание пакетирования	6
4.3	Сборка пакета	7
4.4	Установка пакета	8
5	Создание и настройка локального deb-репозитория	9
	ПРИЛОЖЕНИЕ А	11

1 Ход работы

В ходе данной практической работы будет рассмотрена работа с пакетным менеджером *apt* в операционной системе Ubuntu Linux.

1.1 Установка нового пакета

После выполнения команд в терминале

```
1 $ sudo apt-get install mc
   $ sudo apt-get install htop
3 $ sudo apt-get install neofetch
```

на машину будут установлены соответствующие пакеты.

1.2 Удаление пакета

Для удаления пакета *neofetch* без удаления конфигурационных файлов следует использовать команду

```
1 $ sudo apt-get remove neofetch
```

При использовании такой команды в системе остаются конфигурационные файлы программы, а также дополнительные пакеты. Чтобы удалить конфигурационные файлы можно использовать опцию `-purge` или команду `purge`:

```
1 $ sudo apt-get --purge remove neofetch
```

1.3 Поиск пакета и вывод информации о нем

Для поиска пакета *sqlite* третьей версии в дереве репозитория и вывода информации о нем, следует использовать нижеуказанные команды:

```
1 $ sudo apt-cache pkgnames
$ sudo apt-cache show sqlite3
```

В результате выполнения этих команд мы получим подробную информацию о пакете. См рисунок 1

```
denilai@lirik-pc:~$ apt-cache show sqlite3
Package: sqlite3
Architecture: amd64
Version: 3.31.1-4ubuntu0.2
Multi-Arch: foreign
Priority: optional
Section: misc
Origin: Ubuntu
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Laszlo Boszormenyi (GCS) <gcs@debian.org>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Installed-Size: 2737
Depends: libc6 (>= 2.29), libreadline8 (>= 6.0), zlib1g (>= 1:1.2.0), libsqlite3-0 (= 3.31.1-4ubuntu0.2)
Suggests: sqlite3-doc
Filename: pool/main/s/sqlite3/sqlite3_3.31.1-4ubuntu0.2_amd64.deb
Size: 860460
MD5sum: 3b62590920156f1804b37e5a6f06babd
SHA1: b48abced434a4efd959214f63a671b90f5f20ea4
SHA256: da53cca48ddcd0b668996f6ffcf7957021b17ec303bb8e0911f7714843e45d4c
SHA512: cbb191107f1656ce25380d09d9f8f9c0137c5f30fe4148214c842e04beb55d1379f782d5f870fd8fe3f65c11656373f6f13d4479342bde0dd1b8a383886b3141
Homepage: https://www.sqlite.org/
Description-ru: Интерфейс командной строки для SQLite 3
  SQLite -- это C-библиотека, реализующая движок базы данных SQL. Программы,
  использующие библиотеку SQLite, могут обращаться к базе данных с помощью
  языка SQL без работающего выделенного процесса СУБД.
Description-md5: 60cd5a660be500525f955c43bd7981cf

Package: sqlite3
Architecture: amd64
Version: 3.31.1-4
Multi-Arch: foreign
Priority: optional
Section: misc
Origin: Ubuntu
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Laszlo Boszormenyi (GCS) <gcs@debian.org>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Installed-Size: 2736
Depends: libc6 (>= 2.29), libreadline8 (>= 6.0), zlib1g (>= 1:1.2.0), libsqlite3-0 (= 3.31.1-4)
Suggests: sqlite3-doc
Filename: pool/main/s/sqlite3/sqlite3_3.31.1-4_amd64.deb
Size: 859772
MD5sum: b6866810e606cfbec8ecfe0a6a2b5782
SHA1: 001218f4c34f2dd1e47a3a68675781ab98fd3274
SHA256: c2cfa2cae33a644eebd7de1e93e21d0adc84d9ce5ce39b07f9cb0fe89f42e2b5
Homepage: https://www.sqlite.org/
Description-ru: Интерфейс командной строки для SQLite 3
  SQLite -- это C-библиотека, реализующая движок базы данных SQL. Программы,
  использующие библиотеку SQLite, могут обращаться к базе данных с помощью
  языка SQL без работающего выделенного процесса СУБД.
Description-md5: 60cd5a660be500525f955c43bd7981cf
```

Рис. 1: Поиск пакета в дереве репозитория

2 Добавление репозитория

Ознакомимся с командами, необходимыми для добавления репозитория PostgreSQL:

```
2 $ sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/
  'lsb_release -cs'-pgdg main" >> /etc/apt/sources.list.d/pgdg.list'
$ wget -q https://www.postgresql.org/media/keys/ACCC4CF8.asc -O - |
  sudo apt-key add -
$ sudo apt-get update
4 $ sudo apt install postgresql postgresql-contrib
```

Что содержится в командах:

1. Добавляется репозиторий в список репозитория apt (добавление строки в `sources.list`)
2. Загружается ключ репозитория и добавляется в пакетный менеджер
3. Обновляется список пакетов
4. Установка пакета

В общем случае может быть более простой способ:

```
$ sudo add-apt repository ppa:repo/ppa
```

После ознакомления с примерами, установим `docker`, следуя рекомендациям, приведенным в официальной [документации](#).

3 Сборка пакетов из исходных кодов

В общем случае сборка пакетов выглядит следующим образом:

1. Загрузка архива с исходным кодом (чаще всего `.tar.gz`)
2. Разархивирование (либо в `ui`, либо см. `tar` и его использование)
3. Переход в директорию
4. `./configure`
5. `make`
6. `sudo make install`

Загрузим архив с исходными кодами программы `sqlite` и установим ее в систему в соответствии с вышеуказанными действиями. См рисунки в [Приложении А](#).

4 Создание собственного deb-пакета

4.1 Подготовка окружения

1. В качестве скрипта, который будет помещен в пакет, выберем скрипт `parseawk.sh`, который был написан в ходе первой практической работы по дисциплине "Системное программное обеспечение".
2. Установим необходимые пакеты для сборки пакетов

```
$ sudo apt-get install dpkg debconf debhelper lintian
```

3. Создадим такую структуру папок, содержащих файлы, необходимые для создания deb-пакета. См. рисунок [2](#). Файлы в структуре выделены цветом.

4.2 Описание пакетирования

Создадим файл `control` — цельный файл пакета, описывающий все основные свойства в виде пары «Атрибут: Значение». Заполним необходимые атрибуты.

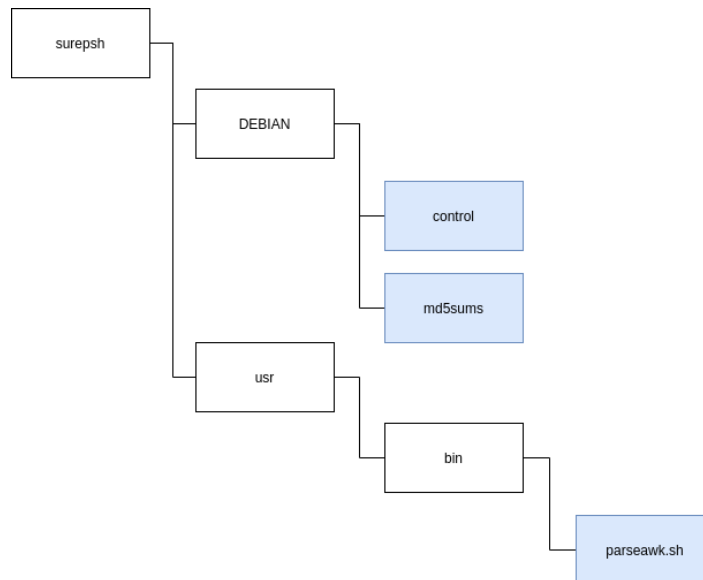


Рис. 2: Структура директории supersh

/home/denilai/supersh/DEBIAN/control

```
1 Package: parseawk
  Version: 0.1-1
3 Architecture: all
  Maintainer: denilai <admin@admin.com>
5 Depends: bash, awk, sed
  Description: Program for testing.
```

Этого файла достаточно для создания работоспособного пакета. После заполнения файла *control* следует приступить к сборке пакета.

4.3 Сборка пакета

Первое, что нужно сделать — это рекурсивно выставить всем файлам в корне пакета пользователя и группу root:root (или другие, если потребуется). Это нужно затем, что файлы пакета упаковываются в tar.gz архив который сохраняет и права доступа к файлам, и владельца. Выполним команду

```
$ sudo chmod -R root:root
```

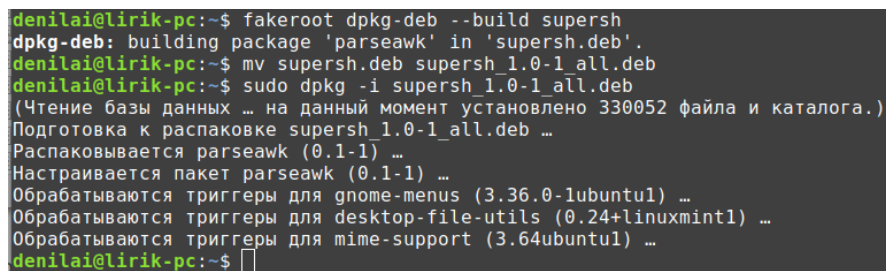
Как альтернативу данной команде следует упомянуть *fakeroot*. Данная утилита позволяет запускать программы в Linux с привилегиями суперпользователя для выполнения любых файловых операций. Изменения видны только для запущенной под *fakeroot* программы, реально в системе ничего не меняется, т.е. для программы создается некая виртуальная оболочка, в которой отражаются все действия. Выполним команду.

```
1 $ fakeroot dpkg-deb --build supersh
```

Созданный пакет необходимо переименовать, чтобы он соответствовал порядку именования *.deb пакетов:

<имя пакета><версия><архитектура>.deb. См. рисунок 3.

```
1 $ mv supersh.deb supersh_1.0-1_all.deb
```



```
denilai@lirik-pc:~$ fakeroot dpkg-deb --build supersh
dpkg-deb: building package 'parseawk' in 'supersh.deb'.
denilai@lirik-pc:~$ mv supersh.deb supersh_1.0-1_all.deb
denilai@lirik-pc:~$ sudo dpkg -i supersh_1.0-1_all.deb
(Чтение базы данных ... на данный момент установлено 330052 файла и каталога.)
Подготовка к распаковке supersh_1.0-1_all.deb ...
Распаковывается parseawk (0.1-1) ...
Настраивается пакет parseawk (0.1-1) ...
Обрабатываются триггеры для gnome-menus (3.36.0-1ubuntu1) ...
Обрабатываются триггеры для desktop-file-utils (0.24+linuxmint1) ...
Обрабатываются триггеры для mime-support (3.64ubuntu1) ...
denilai@lirik-pc:~$
```

Рис. 3: Сборка пакета

4.4 Установка пакета

Установим в систему собранный пакет с помощью команды. См. рисунок 3.

```
1 $ sudo dpkg -i supersh_1.0-1_all.deb
```

Пользовательский deb-пакет успешно установлен в систему. Скрипт расположен по пути `/usr/bin`, в соответствии со структурой директорий, указанной при создании пакета.

5 Создание и настройка локального deb-репозитория

Теперь у нас есть собственный пакет. Когда их будет несколько, и тем более — с зависимостями, окажется, что намного удобнее быстренько поднять собственный локальный микро-репозиторий, и включить его в список источников менеджера пакетов. Сперва установим помощника командой

```
1 $ sudo apt-get install reprepro
```

Затем создадим директорию нашего локального репозитория. Она будет корневой. Создаем файл `conf/distributions`, который заполняем следующим образом.

`/home/denilai/repomy/conf/distributions`

```
1 Description: my local repository
   Origin: Ubuntu
3 Suite: testing
   AlsoAcceptFor: unstable experimental
5 Codename: karmic
   Version: 5.0
7 Architectures: i386 amd64 source
   Components: soft games
9 UDebComponents: soft games
```

Репозиторий описан. Необходимо создать шаблон репозитория по заданному описанию. Для этого воспользуемся следующими командами:

```
1 $ reprepro export
   $ reprepro createsymlinks
```

Следующим шагом необходимо добавить репозиторий в `/etc/apt/sources.list` указав такую строку

```
deb file :///home/denilai/repomy/ karmic soft games
```

Для управления пакетами в репозитории поместим *.deb файл в корневую папку репозитория, и добавляем их в компоненту soft дистрибутива karmic с помощью команды

```
1 reprepro -C soft includedeb karmic *.deb
```

Удаление пакетов доступно с помощью команды

```
1 reprepro -C soft remove karmic supersh
```

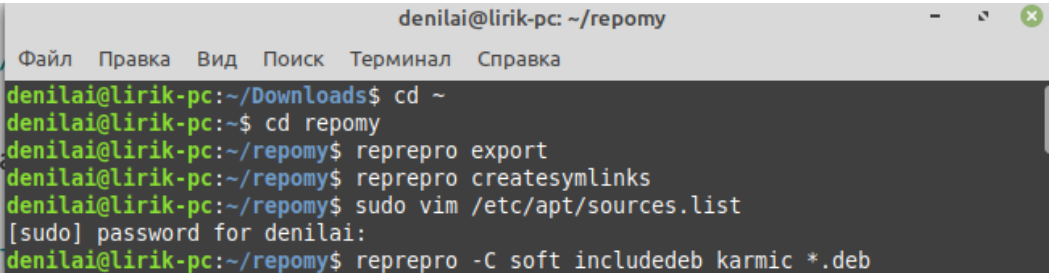
A screenshot of a terminal window titled 'denilai@lirik-pc: ~/репому'. The window has a menu bar with 'Файл', 'Правка', 'Вид', 'Поиск', 'Терминал', and 'Справка'. The terminal shows the following commands and output:
denilai@lirik-pc:~/Downloads\$ cd ~
denilai@lirik-pc:~\$ cd репому
denilai@lirik-pc:~/репому\$ reprepro export
denilai@lirik-pc:~/репому\$ reprepro createsymlinks
denilai@lirik-pc:~/репому\$ sudo vim /etc/apt/sources.list
[sudo] password for denilai:
denilai@lirik-pc:~/репому\$ reprepro -C soft includedeb karmic *.deb

Рис. 4: Работа с репозиторием

ПРИЛОЖЕНИЕ А

```
denilai@lirik-pc:~/Documents$ ls
LaTeX      logisim.jar  repos      tasks
logisim     ProgPractice sqlite-autoconf-3350200.tar.gz  Москва
denilai@lirik-pc:~/Documents$ tar -xzf sqlite-autoconf-3350200.tar.gz
sqlite-autoconf-3350200/
sqlite-autoconf-3350200/sqlite3.rc
sqlite-autoconf-3350200/README.txt
sqlite-autoconf-3350200/aclocal.m4
sqlite-autoconf-3350200/sqlite3.c
sqlite-autoconf-3350200/Makefile.am
sqlite-autoconf-3350200/configure
sqlite-autoconf-3350200/configure.ac
sqlite-autoconf-3350200/config.guess
sqlite-autoconf-3350200/Makefile.in
sqlite-autoconf-3350200/config.sub
sqlite-autoconf-3350200/install-sh
sqlite-autoconf-3350200/sqlite3rc.h
sqlite-autoconf-3350200/sqlite3.1
sqlite-autoconf-3350200/shell.c
sqlite-autoconf-3350200/sqlite3.pc.in
sqlite-autoconf-3350200/Replace.cs
sqlite-autoconf-3350200/depcomp
sqlite-autoconf-3350200/missing
sqlite-autoconf-3350200/ltmain.sh
sqlite-autoconf-3350200/sqlite3ext.h
sqlite-autoconf-3350200/Makefile.msc
sqlite-autoconf-3350200/sqlite3.h
sqlite-autoconf-3350200/Makefile.fallback
sqlite-autoconf-3350200/tea/
sqlite-autoconf-3350200/tea/aclocal.m4
sqlite-autoconf-3350200/tea/configure
sqlite-autoconf-3350200/tea/configure.ac
sqlite-autoconf-3350200/tea/Makefile.in
sqlite-autoconf-3350200/tea/README
sqlite-autoconf-3350200/tea/generic/
sqlite-autoconf-3350200/tea/generic/tclsqlite3.c
sqlite-autoconf-3350200/tea/tclconfig/
sqlite-autoconf-3350200/tea/tclconfig/tcl.m4
sqlite-autoconf-3350200/tea/tclconfig/install-sh
sqlite-autoconf-3350200/tea/doc/
sqlite-autoconf-3350200/tea/doc/sqlite3.n
sqlite-autoconf-3350200/tea/pkgIndex.tcl.in
sqlite-autoconf-3350200/tea/win/
sqlite-autoconf-3350200/tea/win/rules.vc
sqlite-autoconf-3350200/tea/win/nmakehlp.c
sqlite-autoconf-3350200/tea/win/makefile.vc
sqlite-autoconf-3350200/tea/license.terms
sqlite-autoconf-3350200/INSTALL
sqlite-autoconf-3350200/compile
denilai@lirik-pc:~/Documents$ ls
LaTeX      ProgPractice      sqlite-autoconf-3350200.tar.gz
logisim     repos            tasks
logisim.jar  sqlite-autoconf-3350200  Москва
denilai@lirik-pc:~/Documents$ cd sqlite-autoconf-3350200
denilai@lirik-pc:~/Documents/sqlite-autoconf-3350200$ ls
aclocal.m4  configure.ac  Makefile.am  README.txt  sqlite3ext.h  tea
compile     depcomp      Makefile.fallback  Replace.cs  sqlite3.h
config.guess  INSTALL      Makefile.in  shell.c    sqlite3.pc.in
```

Рис. 5: Разархивирование файла

```

aclocal.m4  config.sub  configure.ac  INSTALL  ltmain.sh  Makefile.in  README  sqlite3.l  sqlite3ext.h  sqlite3.pc  tea
config.guess  configure  depcomp  install-sh  Makefile.am  missing  shell.c  sqlite3.c  sqlite3.h  sqlite3.pc.in
denila@lirik-pc:~/Documents/sqlite-autoconf-3070603$ ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking for style of include used by make... GNU
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking dependency style of gcc... gcc3
checking for special C compiler options needed for large files... no
checking for _FILE_OFFSET_BITS value needed for large files... no
checking for gcc... (cached) gcc
checking whether we are using the GNU C compiler... (cached) yes
checking whether gcc accepts -g... (cached) yes
checking for gcc option to accept ISO C89... (cached) none needed
checking dependency style of gcc... (cached) gcc3
checking for ranlib... ranlib
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for a sed that does not truncate output... /usr/bin/sed
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for ld used by gcc... /usr/bin/ld
checking if the linker (/usr/bin/ld) is GNU ld... yes
checking for /usr/bin/ld option to reload object files... -r
checking for BSD-compatible nm... /usr/bin/nm -B
checking whether ln -s works... yes
checking how to recognise dependent libraries... pass_all
checking how to run the C preprocessor... gcc -E
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking forunistd.h... yes
checking for dlfcn.h usability... yes
checking for dlfcn.h presence... yes
checking for dlfcn.h... yes
checking for g++... g++
checking whether we are using the GNU C++ compiler... yes
checking whether g++ accepts -g... yes
checking dependency style of g++... gcc3
checking how to run the C++ preprocessor... g++ -E
checking for g77... no
checking for xlf... no
checking for f77... no
checking for frt... no
checking for pgf77... no
checking for cf77... no
checking for fort77... no
checking for fl32... no
checking for af77... no
checking for xlf90... no
checking for f90... no
checking for pgf90... no

```

Рис. 6: Запуск файла config

```

X_FALLOCATE=1 -I. -I. -D_REENTRANT=1 -DSQLITE_THREADSAFE=1 -DSQLITE_ENABLE_FTS3 -DSQLITE_ENABLE_RTREE -g -O2 -MT sqlite3.lo -MD -MP -MF ".deps/sqlite3.Tpo" -c -o sqlite3.lo sqlite3.c; \
then mv -f ".deps/sqlite3.Tpo" ".deps/sqlite3.Plo"; else rm -f ".deps/sqlite3.Tpo"; exit 1; fi
mkdir .libs
gcc -DPACKAGE_NAME="sqlite" -DPACKAGE_TARNAME="sqlite" -DPACKAGE_VERSION="3.7.6.3" -DPACKAGE_STRING="sqlite 3.7.6.3" -DPACKAGE_BUGREPORT="http://www.sqlite.org" -DPACKAGE="sqlite" -DVERSION="3.7.6.3" -DSTDC_HEADERS=1 -DHAVE_SYS_TYPES_H=1 -DHAVE_SYS_STAT_H=1 -DHAVE_STDLIB_H=1 -DHAVE_STRING_H=1 -DHAVE_MEMORY_H=1 -DHAVE_STRINGS_H=1 -DHAVE_INTTYPES_H=1 -DHAVE_STDINT_H=1 -DHAVE_UNISTD_H=1 -DHAVE_DLFCN_H=1 -DHAVE_FDATASYNC=1 -DHAVE_USLEEP=1 -DHAVE_LOCALTIME_R=1 -DHAVE_GMTIME_R=1 -DHAVE_DECL_STRError=1 -DHAVE_STRError=1 -DHAVE_POSIX_FALLOCATE=1 -I. -I. -D_REENTRANT=1 -DSQLITE_THREADSAFE=1 -DSQLITE_ENABLE_FTS3 -DSQLITE_ENABLE_RTREE -g -O2 -MT sqlite3.lo -MD -MP -MF ".deps/sqlite3.Tpo" -c sqlite3.c -fPIC -DPIC -o .libs/sqlite3.o
gcc -DPACKAGE_NAME="sqlite" -DPACKAGE_TARNAME="sqlite" -DPACKAGE_VERSION="3.7.6.3" -DPACKAGE_STRING="sqlite 3.7.6.3" -DPACKAGE_BUGREPORT="http://www.sqlite.org" -DPACKAGE="sqlite" -DVERSION="3.7.6.3" -DSTDC_HEADERS=1 -DHAVE_SYS_TYPES_H=1 -DHAVE_SYS_STAT_H=1 -DHAVE_STDLIB_H=1 -DHAVE_STRING_H=1 -DHAVE_MEMORY_H=1 -DHAVE_STRINGS_H=1 -DHAVE_INTTYPES_H=1 -DHAVE_STDINT_H=1 -DHAVE_UNISTD_H=1 -DHAVE_DLFCN_H=1 -DHAVE_FDATASYNC=1 -DHAVE_USLEEP=1 -DHAVE_LOCALTIME_R=1 -DHAVE_GMTIME_R=1 -DHAVE_DECL_STRError=1 -DHAVE_STRError=1 -DHAVE_POSIX_FALLOCATE=1 -I. -I. -D_REENTRANT=1 -DSQLITE_THREADSAFE=1 -DSQLITE_ENABLE_FTS3 -DSQLITE_ENABLE_RTREE -g -O2 -MT sqlite3.lo -MD -MP -MF ".deps/sqlite3.Tpo" -c sqlite3.c -o sqlite3.o >/dev/null 2>& 1
/bin/bash ./libtool --tag=CC --mode=link gcc -D_REENTRANT=1 -DSQLITE_THREADSAFE=1 -DSQLITE_ENABLE_FTS3 -DSQLITE_ENABLE_RTREE -g -O2 -o libsqlite3.la -rpath /usr/local/lib -no-undefined -version-info 8:6:8 sqlite3.lo -ldl -lpthread
gcc -shared .libs/sqlite3.o -ldl -lpthread -Wl,-soname -Wl,libsqlite3.so.0 -o .libs/libsqlite3.so.0.8.6
(cd .libs && rm -f libsqlite3.so.0 && ln -s libsqlite3.so.0.8.6 libsqlite3.so.0)
(cd .libs && rm -f libsqlite3.so && ln -s libsqlite3.so.0.8.6 libsqlite3.so)
ar cru .libs/libsqlite3.a sqlite3.o
ar: 'u' modifier ignored since 'D' is the default (see 'U')
ranlib .libs/libsqlite3.a
creating libsqlite3.la
(cd .libs && rm -f libsqlite3.la && ln -s ../libsqlite3.la libsqlite3.la)
if gcc -DPACKAGE_NAME="sqlite" -DPACKAGE_TARNAME="sqlite" -DPACKAGE_VERSION="3.7.6.3" -DPACKAGE_STRING="sqlite 3.7.6.3" -DPACKAGE_BUGREPORT="http://www.sqlite.org" -DPACKAGE="sqlite" -DVERSION="3.7.6.3" -DSTDC_HEADERS=1 -DHAVE_SYS_TYPES_H=1 -DHAVE_SYS_STAT_H=1 -DHAVE_STDLIB_H=1 -DHAVE_STRING_H=1 -DHAVE_MEMORY_H=1 -DHAVE_STRINGS_H=1 -DHAVE_INTTYPES_H=1 -DHAVE_STDINT_H=1 -DHAVE_UNISTD_H=1 -DHAVE_DLFCN_H=1 -DHAVE_FDATASYNC=1 -DHAVE_USLEEP=1 -DHAVE_LOCALTIME_R=1 -DHAVE_GMTIME_R=1 -DHAVE_DECL_STRError=1 -DHAVE_STRError=1 -DHAVE_POSIX_FALLOCATE=1 -I. -I. -D_REENTRANT=1 -DSQLITE_THREADSAFE=1 -DSQLITE_ENABLE_FTS3 -DSQLITE_ENABLE_RTREE -g -O2 -MT shell.o -MD -MP -MF ".deps/shell.Tpo" -c -o shell.o shell.c; \
then mv -f ".deps/shell.Tpo" ".deps/shell.Po"; else rm -f ".deps/shell.Tpo"; exit 1; fi
/bin/bash ./libtool --tag=CC --mode=link gcc -D_REENTRANT=1 -DSQLITE_THREADSAFE=1 -DSQLITE_ENABLE_FTS3 -DSQLITE_ENABLE_RTREE -g -O2 -o sqlite3.shell.o .libs/libsqlite3.la -ldl -lpthread
gcc -D_REENTRANT=1 -DSQLITE_THREADSAFE=1 -DSQLITE_ENABLE_FTS3 -DSQLITE_ENABLE_RTREE -g -O2 -o .libs/sqlite3.shell.o .libs/libsqlite3.so -ldl -lpthread
creating sqlite3
denilai@lrk-pc:~/Documents/sqlite-autoconf-3070603$ make install
make[1]: Entering directory '/home/denilai/Documents/sqlite-autoconf-3070603'
test -z "/usr/local/lib" || mkdir -p -- "/usr/local/lib"
/bin/bash ./libtool --mode=install /usr/bin/install -c 'libsqlite3.la' '/usr/local/lib/libsqlite3.la'
/usr/bin/install -c .libs/libsqlite3.so.0.8.6 /usr/local/lib/libsqlite3.so.0.8.6
/usr/bin/install: cannot create regular file '/usr/local/lib/libsqlite3.so.0.8.6': Permission denied
make[1]: *** [Makefile:267: install-libLTLIBRARIES] Error 1
make[1]: Leaving directory '/home/denilai/Documents/sqlite-autoconf-3070603'
make: *** [Makefile:634: install-am] Error 2
denilai@lrk-pc:~/Documents/sqlite-autoconf-3070603$ sudo make install
make[1]: Entering directory '/home/denilai/Documents/sqlite-autoconf-3070603'
test -z "/usr/local/lib" || mkdir -p -- "/usr/local/lib"
/bin/bash ./libtool --mode=install /usr/bin/install -c 'libsqlite3.la' '/usr/local/lib/libsqlite3.la'
/usr/bin/install -c .libs/libsqlite3.so.0.8.6 /usr/local/lib/libsqlite3.so.0.8.6
(cd /usr/local/lib && { ln -s -f libsqlite3.so.0.8.6 libsqlite3.so.0 || { rm -f libsqlite3.so.0 && ln -s libsqlite3.so.0.8.6 libsqlite3.so.0; }; })
(cd /usr/local/lib && { ln -s -f libsqlite3.so.0.8.6 libsqlite3.so || { rm -f libsqlite3.so && ln -s libsqlite3.so.0.8.6 libsqlite3.so; }; })
/usr/bin/install -c .libs/libsqlite3.la /usr/local/lib/libsqlite3.la
/usr/bin/install -c .libs/libsqlite3.a /usr/local/lib/libsqlite3.a
chmod 644 /usr/local/lib/libsqlite3.a
ranlib /usr/local/lib/libsqlite3.a
PATH="$PATH:/sbin" ldconfig -n /usr/local/lib
-----
Libraries have been installed in:
  /usr/local/lib

If you ever happen to want to link against installed libraries
in a given directory, LIBDIR, you must either use libtool, and
specify the full pathname of the library, or use the '-LLIBDIR'
flag during linking and do at least one of the following:
  - add LIBDIR to the 'LD_LIBRARY_PATH' environment variable
    during execution

```

Рис. 7: Процесс сборки командой make