

# Сборка deb-пакета

Завернём в пакет наш шелл-скрипт, который реализует команду mv для директорий.

Установим необходимые программы.

```
sudo apt-get install dh-make devscripts
```

Назовём пакет, в нём — папка DEBIAN, в которую мы положим все файлы с метаданными и наш шелл-скрипт.

Создадим директорию, назовём её по имени нашего скрипта и добавим версию через дефис.

```
mkdir mvdir-0.1  
cd mvdir-0.1
```

Теперь скопируем наш шелл-скрипт в директорию.

```
cp ../mvdir.sh .
```

dh\_make понадобится информация об имени и имейле. Упростим ему задачу, сделав следующее:

```
vim .bashrc  
DEBEMAIL="person@company.tld"  
DEBFULLNAME="Person McTester"  
source .bashrc
```

Теперь давайте попросим dh\_make создать нам шаблон. Обратите внимание, что запускать его нужно изнутри нашей папки. Так как мы опакечиваем шелл-скрипт, нам потребуется пара флагов.

```
dh_make --indep --createorig
```

Давайте заглянем в один из скриптов, например preinst. Это оператор выбора, в нём можно задать какие-то свои действия, которые будут выполняться перед установкой пакета либо после неё.

dh\_make создал нам все необходимые файлы, кроме install. Файл install необходим нам для того, чтобы передать информацию утилите debuild, собирающей пакет, какие файлы должны быть в него включены.

Поэтому создадим debian/install:

```
touch debian/install  
vi debian/install  
mvdir.sh usr/bin
```

Если бы у нас было несколько скриптов, мы могли бы перечислить их отдельно или заменить на такую запись:

```
*.sh usr/bin
```

Ещё один важный файл — changelog. Это документ, который рассказывает о том, что изменилось в новой версии.

Редактировать файл можно вручную:

```
vim changelog
```

Давайте добавим release message специальной командой *dch -i*.

Выйдем из редактора без сохранения.

**Соберём пакет!** Для этого используется команда:

```
debuild -us -uc
```

Эти флаги говорят программе, что мы не будем подписывать файл .changes или .dsc.

**Установим пакет:**

```
sudo dpkg -i mmdir_1.0-1_all.deb
ls -la /usr/bin/mmdir
mkdir test1
mmdir.sh test1 test3
ls -la
```

Готово.

Теперь, если мы **удалим пакет**:

```
dpkg -r mmdir
ls /usr/bin/
```

Мы увидим, что скрипт исчез.

Подпишем наш пакет, заодно посмотрим, как пересобирается пакет при внесении изменений.

Чтобы подписать изменения, потребовалось сгенерировать пару ключей при помощи утилиты gpg:

```
gpg --gen-key
```

Теперь нам необходимо поднять версию пакета и написать, что изменилось.

Ещё один важный момент. Realname и имейл, который вы вводили при генерации ключевой пары, необходимо написать и в чейнджлоге.

Если вы вдруг забыли, всегда можно посмотреть, используя gpg --list-keys.

Теперь, если мы соберём пакет:

```
debuild -b
```

Процесс спросит нас пасс-фразу от ключа. Введём её и получим сообщение о том, что всё подписано успешно.

Чтобы передать кому-то публичный ключ, его понадобится экспортировать из хранилища операционной системы. Для этого вам понадобится команда:

```
gpg --export -a "email@address.com" > public.key
```