

Сборка образа OpenWRT

Если тебя по какой-то причине не устраивает стандартный образ OpenWRT, то можно собрать свой, для чего необходимо получить тулчейн и OpenWRT Buildroot. Прежде всего установим соответствующие пакеты:

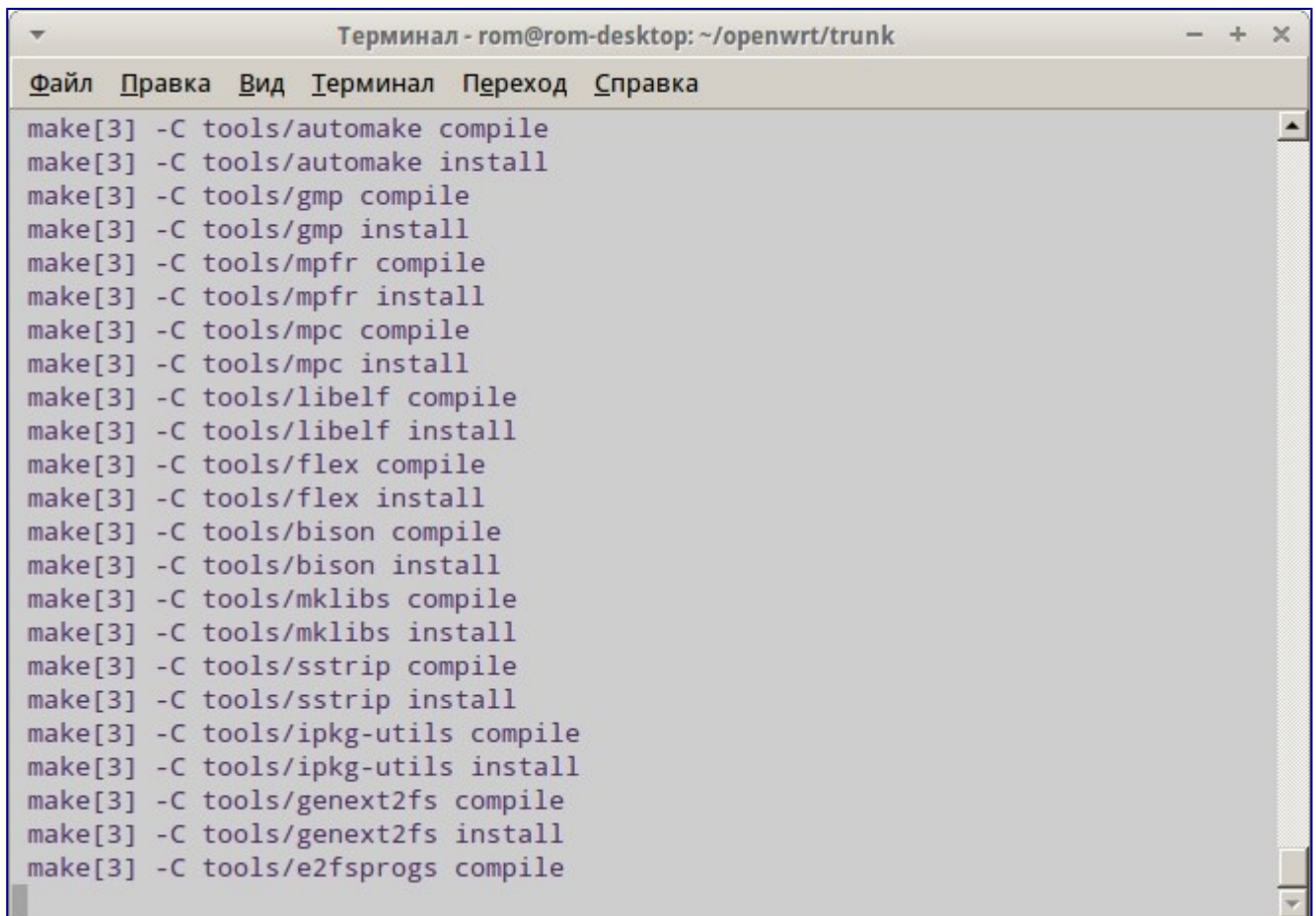
```
$ sudo apt-get install subversion build-essential git-core libncurses5-dev  
zlib1g-dev gawk
```

Собирать мы будем текущую нестабильную версию OpenWRT и материалы (feeds). Скачаем их.

```
$ mkdir openwrt && cd $_  
$ svn co svn://svn.openwrt.org/openwrt/trunk/  
$ cd trunk  
$ ./scripts/feeds update -a && ./scripts/feeds install -a
```

После этого проверим зависимости — мало ли, вдруг какой-нибудь необходимый для сборки пакет не установлен.

```
$ make prereq
```

A screenshot of a terminal window titled "Терминал - rom@rom-desktop: ~/openwrt/trunk". The terminal shows the output of the 'make prereq' command, which lists various tools and their compilation and installation status. The output is as follows:

```
make[3] -C tools/automake compile  
make[3] -C tools/automake install  
make[3] -C tools/gmp compile  
make[3] -C tools/gmp install  
make[3] -C tools/mpfr compile  
make[3] -C tools/mpfr install  
make[3] -C tools/mpc compile  
make[3] -C tools/mpc install  
make[3] -C tools/libelf compile  
make[3] -C tools/libelf install  
make[3] -C tools/flex compile  
make[3] -C tools/flex install  
make[3] -C tools/bison compile  
make[3] -C tools/bison install  
make[3] -C tools/mklibs compile  
make[3] -C tools/mklibs install  
make[3] -C tools/sstrip compile  
make[3] -C tools/sstrip install  
make[3] -C tools/ipkg-utils compile  
make[3] -C tools/ipkg-utils install  
make[3] -C tools/genext2fs compile  
make[3] -C tools/genext2fs install  
make[3] -C tools/e2fsprogs compile
```

Сборка необходимых инструментов

Если все нормально, можем конфигурировать образ. Для чего вводим

```
$ make menuconfig
```

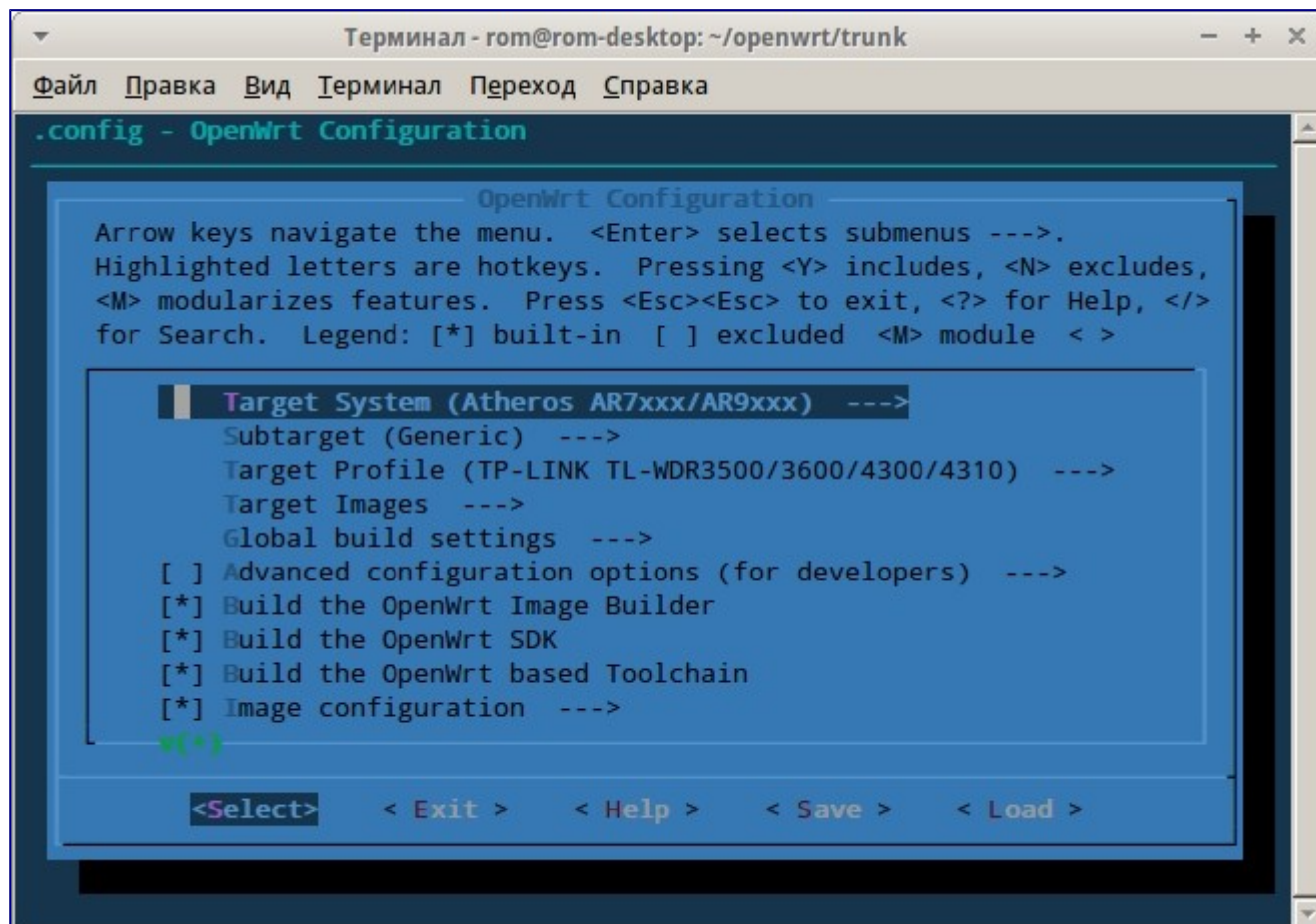
И выбираем, точно так же, как и при конфигурации ядра, нужные тебе вещи. Единственное отличие от «ядерного» `menuconfig` — звездочка означает, что объект будет встроен в образ, а М — что будет доступен в виде пакета `ipk`, который позже можно будет установить отдельно. Сильно увлекаться, однако, не советую — помни о том, что места на внутренней флеш-памяти не просто мало, а очень мало.

В основном процесс конфигурирования включает в себя следующие шаги:

- Выбор конкретной целевой системы и профиля. Их необходимо указывать как можно точнее — если укажешь неправильно, ты рискуешь превратить роутер в кирпич.
- Выбор пакетов. Здесь действует правило — чем меньше тыстроишь их в образ, тем лучше. Поэтому выбирай только самые необходимые. Я бы посоветовал включить `LuCI`.
- Настройки сборки. Тут ты можешь разве что в разделе `Global build settings` включить/выключить IPv6. В `Advanced configuration options` стоит лезть, только когда ты четко знаешь, для чего, к примеру, служит та или иная опция `GCC`, достаточна ли мощность процессора для включения защиты стека и так далее.
- Выбор модулей ядра. Тут те же самые рекомендации, что и при выборе пакетов. Загляни в секцию `USB Support` и включи опцию `kmod-usb-storage`. Остальные включай по желанию.

Если же тебе зачем-то понадобится подправить ядерный конфиг, используй

```
$ make kernel_menuconfig
```



Выбор опций при сборке кастомной прошивки

Учти, что если ты потом сделаешь очистку, то конфиг ядра не очистится. Для его очистки набери

```
$ svn revert -R target/linux/
```

После всего этого можешь смело набирать команду

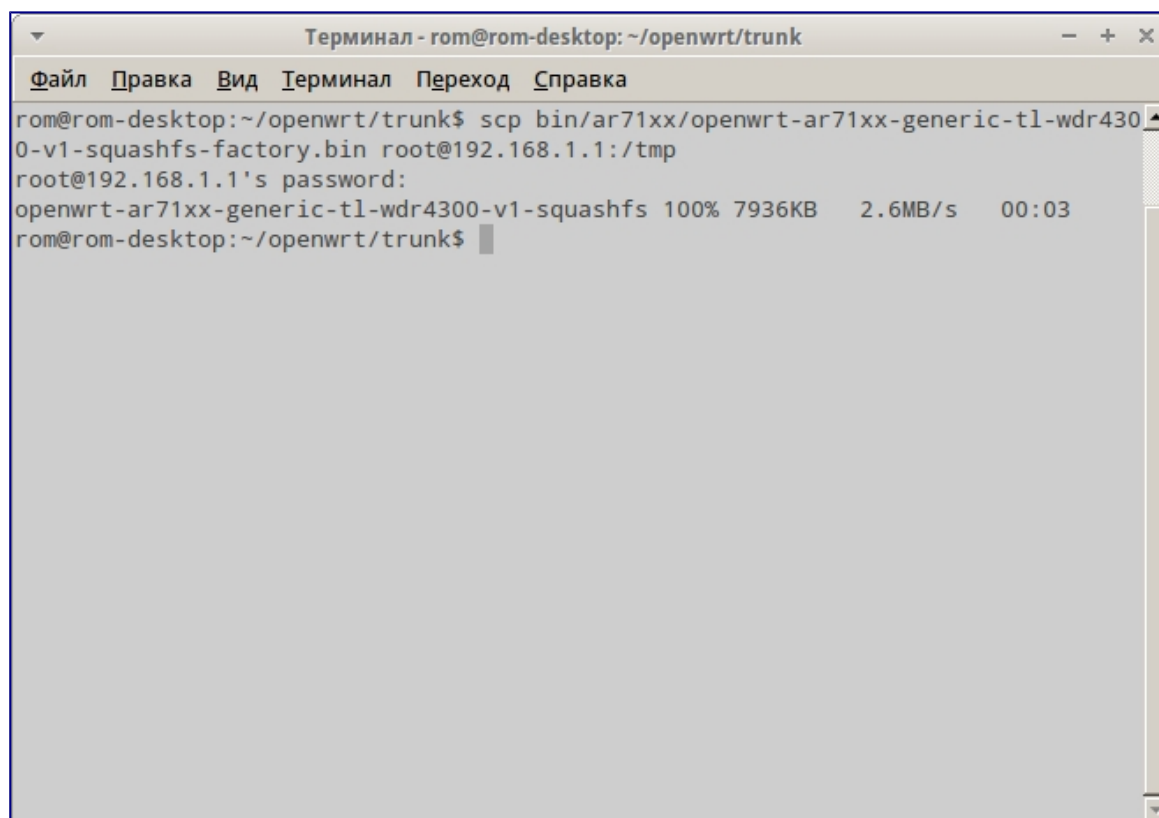
```
$ make
```

для сборки образа. Это займет длительное время, по истечении которого в каталоге `bin/` появятся файлы образа.

Существует несколько путей для того, чтобы залить прошивку. Самый легкий из них — через веб-интерфейс, и описывать его я смысла не вижу. Второй способ — использовать утилиту `scp` на хостовом компьютере вкупе с `mtd` на роутере для заливки образа и его прошивки. Поскольку сейчас стоит уже OpenWRT, то можно прошивать как `factory-`, так и `sysupgrade-` образ. В моем случае команды были такими:

```
$ scp bin/ar71xx/openwrt-ar71xx-generic-tl-wdr4300-v1-squashfs-factory.bin root@192.168.1.1:/tmp
openwrt# mtd -r write /tmp/openwrt-ar71xx-generic-tl-wdr4300-v1-squashfs-factory.bin firmware
```

Вторая команда шьет (`write`) свежескачанную прошивку в раздел, именуемый `firmware`, и вслед за этим роутер перезагружается (`-r`). В случае обновления OpenWRT со стабильной версии до текущей я бы советовал не восстанавливать сохраненную конфигурацию, а настроить все заново — у меня по каким-то причинам старые конфиги не подошли к свежескомпилированной версии.



```
Терминал - rom@rom-desktop: ~/openwrt/trunk
Файл  Правка  Вид  Терминал  Переход  Справка
rom@rom-desktop:~/openwrt/trunk$ scp bin/ar71xx/openwrt-ar71xx-generic-tl-wdr4300-v1-squashfs-factory.bin root@192.168.1.1:/tmp
root@192.168.1.1's password:
openwrt-ar71xx-generic-tl-wdr4300-v1-squashfs 100% 7936KB  2.6MB/s  00:03
rom@rom-desktop:~/openwrt/trunk$
```

