

Редактирование образа Raspberry Pi с помощью qemu-user-static (Ubuntu 14.04).

Иногда необходимо редактирование стандартных загрузочных образов, а также конфигурирование систем с последующим тиражированием на большое количество плат Raspberry Pi. Для решения подобных задач удобно использовать пакеты qemu-user-static и binfmt-support.

Запускаем терминал и устанавливаем пакеты qemu-user-static и binfmt-support:

```
sudo apt-get install qemu qemu-user-static binfmt-support
```

QEMU в режиме qemu-user-static позволяет запускать бинарные файлы, скомпилированные для одного процессора, используя процессор другой архитектуры. Пакет binfmt-support позволяет запускать бинарные файлы напрямую.

После установки смотрим список поддерживаемых бинарных файлов:

```
simm@simm-UX32VD:~$ update-binfmts --display
```

```
qemu-aarch64 (enabled):
.....
qemu-microblaze (enabled):
.....
qemu-arm (enabled):
.....
qemu-m68k (enabled):
.....
qemu-ppc64abi32 (enabled):
.....
qemu-sparc64 (enabled):
.....
qemu-sparc (enabled):
.....
qemu-sh4 (enabled):
.....
qemu-sh4eb (enabled):
.....
qemu-sparc32plus (enabled):
.....
qemu-ppc64 (enabled):
.....
qemu-ppc (enabled):
.....
qemu-mipsel (enabled):
.....
qemu-alpha (enabled):
```

```
.....
qemu-mips (enabled):
.....
qemu-cris (enabled):
.....
qemu-s390x (enabled):
.....
qemu-armeb (enabled):
.....
```

Как видим, поддержка ARM файлов включена - qemu-arm (enabled).

Переходим на страницу <https://www.raspberrypi.org/downloads/> и выбираем нужную систему. В качестве примера возьмем Raspbian Wheezy, на данный момент актуальная версия 2015-05-05-raspbian-wheezy.img

Скачиваем и распаковываем архив:

```
sudo mkdir ~/rpi_image
cd ~/rpi_image
sudo wget http://downloads.raspberrypi.org/raspbian/images/raspbian-2015-05-07/2015-05-05-raspbian-wheezy.zip
sudo unzip 2015-05-05-raspbian-wheezy.zip
sudo rm 2015-05-05-raspbian-wheezy.zip
```

Сначала получим информацию об образе:

```
simm@simm-UX32VD:~/rpi_image$ fdisk -lu 2015-05-05-raspbian-wheezy.img
```

```
Диск 2015-05-05-raspbian-wheezy.img: 3276 МБ, 3276800000 байт
255 головок, 63 секторов/треков, 398 цилиндров, всего 6400000 секторов
Units = секторы of 1 * 512 = 512 bytes
Размер сектора (логического/физического): 512 байт / 512 байт
I/O size (minimum/optimal): 512 bytes / 512 bytes
Идентификатор диска: 0xa6202af7
```

	Устр-во	Загр	Начало	Конец	Блоки	Id	Система
2015-05-05-raspbian-wheezy.img1		FAT32 (LBA)	8192	122879	57344	c	W95
2015-05-05-raspbian-wheezy.img2			122880	6399999	3138560	83	Linux

Добавляем 1Gb к образу:

```
sudo chmod 775 2015-05-05-raspbian-wheezy.img
sudo dd if=/dev/zero bs=1M count=1024 >> 2015-05-05-raspbian-wheezy.img
```

Подцепляем весь образ к устройству loop0, а второй раздел (начинается с сектора 122880, каждый сектор по 512 байт) к loop1.

```
sudo losetup -f --show 2015-05-05-raspbian-wheezy.img
sudo losetup -f --show -o $((122880*512)) 2015-05-05-raspbian-wheezy.img
```

Это привяжет устройство /dev/loop0 ко всему образу и /dev/loop1 к разделу, который мы хотим расширить.

Запускаем parted, удаляем второй раздел в устройстве /dev/loop0 и создаем его с новым размером.

```
simm@simm-UX32VD:~/rpi_image$ sudo parted /dev/loop0
```

```
GNU Parted 2.3
```

```
Используется /dev/loop0
```

```
Добро пожаловать в GNU Parted! Наберите 'help' для просмотра списка команд.
```

```
(parted) print
```

```
Модель: Loopback device (loop)
```

```
Диск /dev/loop0: 4351MB
```

```
Размер сектора (логич./физич.): 512B/512B
```

```
Таблица разделов: msdos
```

Номер	Начало	Конец	Размер	Тип	Файловая система	Флаги
1	4194kB	62,9MB	58,7MB	primary	fat16	lba
2	62,9MB	3277MB	3214MB	primary	ext4	

```
(parted) rm 2
```

```
(parted) mkpart primary 62.9 4351
```

```
(parted) print
```

```
Модель: Loopback device (loop)
```

```
Диск /dev/loop0: 4351MB
```

```
Размер сектора (логич./физич.): 512B/512B
```

```
Таблица разделов: msdos
```

Номер	Начало	Конец	Размер	Тип	Файловая система	Флаги
1	4194kB	62,9MB	58,7MB	primary	fat16	lba
2	62,9MB	4351MB	4288MB	primary	ext4	

```
(parted) quit
```

Затем проверим и изменяем размер нового раздела:

```
simm@simm-UX32VD:~/rpi_image$ sudo e2fsck -f /dev/loop1
```

```
e2fsck 1.42.9 (4-Feb-2014)
```

```
Проход 1: Проверка inodes, блоков, а также размеров
```

```
Pass 2: Checking каталог structure
```

```
Pass 3: Checking каталог connectivity
```

```
Pass 4: Checking reference counts
```

```
Pass 5: Checking группа summary information
```

```
/dev/loop1: 86233/196224 files (0.1% non-contiguous), 630146/784640 blocks
```

```
simm@simm-UX32VD:~/rpi_image$ sudo resize2fs /dev/loop1
```

```
resize2fs 1.42.9 (4-Feb-2014)
```

```
Resizing the filesystem on /dev/loop1 to 1046784 (4k) blocks.
```

```
The filesystem on /dev/loop1 is now 1046784 blocks long.
```

Теперь убедимся, что размер нового раздела увеличился на 1 Gb:

```
simm@simm-UX32VD:~/rpi_image$ sudo parted /dev/loop0
```

```
GNU Parted 2.3
```

```
Используется /dev/loop0
```

```
Добро пожаловать в GNU Parted! Наберите 'help' для просмотра списка команд.
```

```
(parted) print
```

```
Модель: Loopback device (loop)
```

```
Диск /dev/loop0: 4351MB
```

```
Размер сектора (логич./физич.): 512B/512B
```

```
Таблица разделов: msdos
```

Номер	Начало	Конец	Размер	Тип	Файловая система	Флаги
1	4194kB	62,9MB	58,7MB	primary	fat16	lba
2	62,9MB	4351MB	4288MB	primary	ext4	

```
(parted) quit
```

Очистим loop-устройства:

```
losetup -d /dev/loop0 /dev/loop1
```

Монтируем образ:

```
sudo mkdir ~/rpi_mnt
```

```
sudo mount ~/rpi_image/2015-05-05-raspbian-wheezy.img -o loop,offset=$((122880*512)),rw ~/rpi_mnt
```

(необязательно) Монтируем /boot :

```
sudo mount ~/rpi_image/2015-05-05-raspbian-wheezy.img -o loop,offset=$((8192*512)),rw ~/rpi_mnt/boot
```

(необязательно):

```
cd ~/rpi_mnt
```

```
sudo mount --bind /dev dev/
```

```
sudo mount --bind /sys sys/
```

```
sudo mount --bind /proc proc/
```

```
sudo mount --bind /dev/pts dev/pts
```

Чтобы все работало правильно (например, сеть) перед изменением корневого каталога нужно закомментировать все строки в файле ~/rpi_mnt/etc/ld.so.preload

```
sudo vi ~/rpi_mnt/etc/ld.so.preload
```

для редактирования нажмите клавишу i, впишите # перед каждой строкой, затем нажмите <ESC> :wq <ENTER>

Изменение корневого каталога (CHROOT).

Прежде всего, нужно убедиться что binfmt-support запустит наш код, как только мы изменим корневую файловую систему. Для этого скопируем файл в корневой каталог образа

```
sudo cp /usr/bin/qemu-arm-static ~/rpi_mnt/usr/bin
```

Меняем корень

```
cd ~/rpi_mnt  
sudo chroot. bin/bash
```

Проверим изменение корневого каталога

```
root@simm-UX32VD:/# uname -a
```

```
Linux simm-UX32VD 3.19.0-33-generic #38~14.04.1-Ubuntu SMP Fri Nov 6 18:17:28 UTC  
2015 armv7l GNU/Linux
```

Теперь можно добавлять и удалять программы, конфигурировать систему, после чего скопировать полученный образ на множество устройств Raspberry Pi, без необходимости подключения монитора и клавиатуры к каждой отдельной плате.

Вход в меню конфигурации

```
sudo raspi-config
```

Удаление окружение рабочего стола .

```
apt-get remove --dry-run --auto-remove --purge libx11-.*
```

Убедитесь, что в списке нет лишних пакетов и запустите еще раз без "--dry-run".

Обновление системы.

Для обновления до новой версии (например, jessie, stretch и др.), должен быть смонтирован /boot (также проверьте все файлы в /etc/apt/sources.list.d для обновления). Перед обновлением сохраните нужные конфигурационные файлы.

```
sed -i 's/wheezy/jessie/g' /etc/apt/sources.list
```

```
apt-get update
apt-get dist-upgrade -o Dpkg::Options::="--force-confold"
```

Для записи образа на SD карту необходимо:

1. Выйти из CHROOT (наберите exit)
2. Раскомментировать строки в файле /etc/ld.so.preload.

```
sudo vi ~/rpi_mnt/etc/ld.so.preload
```

удалите добавленные ранее символы # нажатием x, затем нажмите <ESC> :wq <ENTER>

3. Размонтировать все разделы

```
sudo umount ~/rpi_mnt/sys
sudo umount ~/rpi_mnt/proc
sudo umount ~/rpi_mnt/dev/pts
sudo umount ~/rpi_mnt/boot
sudo umount ~/rpi_mnt/dev
cd ..
sudo umount ~/rpi_mnt
```

Вставляем SD карту, смотрим путь и записываем образ:

```
sudo fdisk -l
sudo dd if=~/rpi_image/2015-05-05-raspbian-wheezy.img of=/dev/mmcblk0
```