

## [Multi-booting Raspberry Pi Online](#)

This document is a reference for multi-booting Raspberry Pi with RDKB images. This method can be used to flash new RDKB images and switch between different images. The major advantage here is that, we can do previously mentioned tasks remotely without the help from someone at the remote location and without even removing the existing memory card from the Raspberry Pi.

*Note:* This method assumes that there is already an RDKB image running in the Raspberry Pi, and it can be accessed remotely.

### Requirements:

- A remote Raspberry Pi running an RDKB image.
- `resize2fs` binary.
- Memory card should have reasonable size for multi-booting (at least 2Gb memory card is recommended)

### Steps

◆ **Step 1 :** Copy `resize2fs` binary to `/usr/sbin/` of the Raspberry Pi.

◆ **Step 2 :** Edit partition table

We need to expand the existing partition(`/dev/mmcblk0p2`) and create a new partition(`/dev/mmcblk0p3`). In order to reflect the partition changes we need to reboot the device.

For demonstrating this step I am going to use a 16 Gb memory card. I will be expanding partition 2 to about 5 Gb and will be creating a new partition 3 with similar size. After the following partition, there will be around 5 Gb unallocated space in the memory card.

```
root@RaspberryPi-Gateway:~# fdisk -u /dev/mmcblk0
```

The number of cylinders for this disk is set to 243096.

There is nothing wrong with that, but this is larger than 1024, and could in certain setups cause problems with:

- 1) software that runs at boot time (e.g., old versions of LILO)
- 2) booting and partitioning software from other OSs (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): p

Disk /dev/mmcblk0: 15.9 GB, 15931539456 bytes

4 heads, 32 sectors/track, 243096 cylinders, total 31116288 sectors

Units = sectors of 1 \* 512 = 512 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/mmcblk0p1	*	8192	90111	40960	c	Win95 FAT32 (LBA)
/dev/mmcblk0p2		90112	770047	339968	83	Linux

Command (m for help): d

Partition number (1-4): 2

Command (m for help): n

```

Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 2
First sector (32-31116287, default 32): 90112
Last sector or +size or +sizeM or +sizeK (90112-31116287, default 31116287):
10372096

Command (m for help): p

Disk /dev/mmcblk0: 15.9 GB, 15931539456 bytes
4 heads, 32 sectors/track, 243096 cylinders, total 31116288 sectors
Units = sectors of 1 * 512 = 512 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/mmcblk0p1    *           8192        90111       40960    c   Win95 FAT32 (LBA)
/dev/mmcblk0p2             90112       10372096     5140992+  83   Linux

Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 3
First sector (32-31116287, default 32): 10372097
Last sector or +size or +sizeM or +sizeK (10372097-31116287, default 31116287):
20744192

Command (m for help): p

Disk /dev/mmcblk0: 15.9 GB, 15931539456 bytes
4 heads, 32 sectors/track, 243096 cylinders, total 31116288 sectors
Units = sectors of 1 * 512 = 512 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/mmcblk0p1    *           8192        90111       40960    c   Win95 FAT32 (LBA)
/dev/mmcblk0p2             90112       10372096     5140992+  83   Linux
/dev/mmcblk0p3      10372097       20744192     5186048    83   Linux

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table
fdisk: WARNING: rereading partition table failed, kernel still uses old table:
Device or resource busy
root@RaspberryPi-Gateway:~# reboot

```

After executing `fdisk -u /dev/mmcblk0` enter `p`, you would get the details of your memory card. Using that details you can calculate the First sector and Last sector for your partitions. To know more on how to calculate partition size, check the end of this document.

### ◆ Step 3: Expand /dev/mmcblk0p2

```

root@telekom:~# df -h
Filesystem      Size      Used Available Use% Mounted on
/dev/root       313.5M    313.5M         0 100% /
devtmpfs        458.6M         0     458.6M   0% /dev
tmpfs           463.1M      8.0K     463.1M   0% /dev/shm

```

```

tmpfs                463.1M      21.1M      442.0M      5% /run
tmpfs                463.1M          0      463.1M      0% /sys/fs/cgroup
tmpfs                463.1M       6.7M      456.4M      1% /tmp
tmpfs                463.1M       2.1M      461.0M      0% /var/volatile
root@RaspberryPi-Gateway:~# resize2fs /dev/mmcblk0p2
resize2fs 1.43 (17-May-2016)
Filesystem at /dev/mmcblk0p2 is mounted on /; on-line resizing required
old_desc_blocks = 2, new_desc_blocks = 20
The filesystem on /dev/mmcblk0p2 is now 5140992 (1k) blocks long.
root@RaspberryPi-Gateway:~# df -h                // verify partition has expanded
Filesystem            Size      Used Available Use% Mounted on
/dev/root              4.7G      273.4M      4.2G      6% /
devtmpfs              458.6M          0      458.6M      0% /dev
tmpfs                 463.1M       4.0K      463.1M      0% /dev/shm
tmpfs                 463.1M      21.1M      442.0M      5% /run
tmpfs                 463.1M          0      463.1M      0% /sys/fs/cgroup
tmpfs                 463.1M       2.5M      460.6M      1% /tmp
tmpfs                 463.1M       1.1M      462.0M      0% /var/volatile

```

#### ◆ Step 4: Format /dev/mmcblk0p3

```

root@RaspberryPi-Gateway:~# mkfs.ext4 /dev/mmcblk0p3
mke2fs 1.43 (17-May-2016)
Discarding device blocks: done
Creating filesystem with 1296512 4k blocks and 324480 inodes
Filesystem UUID: f0a1f763-a16b-4546-9cd8-b890e808d150
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information:
done

```

#### ◆ Step 5: Copy the image that needs to be flashed to /tmp/

Note that we are only interested in rootfs files. So need an rdkb image with .rootfs.tar.gz format. (eg: rdk-generic-broadband-image-raspberrypi-rdk-broadband.tar.gz)

#### ◆ Step 6: Extract the image

```

root@RaspberryPi-Gateway:~# mount /dev/mmcblk0p3 /mnt
root@RaspberryPi-Gateway:~# rm -rf /mnt/lost+found/
root@RaspberryPi-Gateway:~# tar -xzf /tmp/rdk-generic-broadband-image-raspberrypi-
rdk-broadband.tar.gz -C /mnt
root@RaspberryPi-Gateway:~# umount /mnt

```

#### ◆ Step 7: Change boot configuration

Next we need to edit root option in the cmdline.txt under the boot partition (/dev/mmcblk0p1). Change root=/dev/mmcblk0p2 to root=/dev/mmcblk0p3 in cmdline.txt.

```

root@RaspberryPi-Gateway:~# mount /dev/mmcblk0p1 /mnt
root@RaspberryPi-Gateway:~# vi /mnt/cmdline.txt
root@RaspberryPi-Gateway:~# umount /mnt
root@RaspberryPi-Gateway:~# reboot

```

## Flashing new images

Steps 1-4 are one time configurations. Once we have done those steps, no need of doing it again for flashing new images using this method. We can flash images in the partition that we are not currently using. For example let's assume that you are using partition 3 (/dev/mmcblk0p3) then flash the new image in the partition 2 (/dev/mmcblk0p2) by formatting partition 2 and following steps 5-7, and vice versa.

## Switching between images

Since we are using multi-boot, we can switch between images. Say you have one version of image in partition 2 (/dev/mmcblk0p2) and another version in partition 3 (/dev/mmcblk0p3), then you want to switch the versions just change the root option in the cmdline.txt in the boot partition(/dev/mmcblk0p1). Ref: Step-7.

## Changing the kernel

All the images in the multi-booted Raspberry Pi share the same kernel. We can replace the existing kernel while the Raspberry Pi is running. In order to reflect the kernel change we need to reboot the Raspberry Pi after replacing the kernel (ref below steps).

```
root@RaspberryPi-Gateway:~# mount /dev/mmcblk0p1 /mnt
root@RaspberryPi-Gateway:~# rm /mnt/kernel7.img // remove existing kernel
root@RaspberryPi-Gateway:~# mv /tmp/lernel7.img /mnt/ // copy your new kernel
root@RaspberryPi-Gateway:~# umount /mnt
root@RaspberryPi-Gateway:~# reboot
```

## Partition size calculation

If we know the Starting sector, Ending sector and Bytes per sector, then we can calculate size of the partition.

```
Number of sectors = Ending sector - Starting sector + 1
Total bytes in the partition = Number of sectors x Bytes per sector
Size of partition in GB = Total bytes in the partition / (1024 x 1024 x 1024)
```

Following table is an example for such partition size calculation.

Partition	/dev/mmcblk0p2
Starting sector	90112
Ending sector	10372096
Number of sectors	10281985
Bytes per sector	512
Total bytes in the partition	5264376320
Size of partition in GB	4.9028325081

To verify our partition has the same size as we calculated, we can use `fdisk`. In the following output I got the same number of bytes as my calculated total bytes in the partition (Note that `fdisk` has calculated the size in MB by rounding off 1024 to 1000, and that field will be inaccurate).

```
root@RaspberryPi-Gateway:~# fdisk -l /dev/mmcblk0p2

Disk /dev/mmcblk0p2: 5264 MB, 5264376320 bytes
4 heads, 16 sectors/track, 160656 cylinders
Units = cylinders of 64 * 512 = 32768 bytes

Disk /dev/mmcblk0p2 doesn't contain a valid partition table
```

You would have noticed that there is a discrepancy with the size shown by `df -h` (4.7 GB) and `fdisk -l /dev/mmcblk0p2` (4.9 GB). It could be because some space is reserved for root, in the event the partitions become full. If there is no space reserved for root, and the partitions become full, the system cannot function. However, this difference is usually of the order of 4-6% of the total size as per my calculation. It is mentioned in the [man page of `df`](#) that, `df` only shows the space available on that file system.