How to Resize a Partition using fdisk

Updated August 9 2017 at 4:00 PM - English ▾ ()

fdisk is a dialog-driven program for the creation and manipulation of partition tables and understands GPT, MBR, Sun, SGI, and BSD partition tables. This article describes how to enlarge a partition with **fdisk**. Note that **fdisk**, while fdisk is easier to use, parted is more robust for ext4 file systems. However, resizing with parted is not available in Red Hat Enterprise Linux 7.

Prerequisites

- A created partition that you know the name of. To check the name, run **cat /etc/fstab**. The first field is the name of the partition. The only way to change a partition size using **fdisk** is by deleting and recreating it so ensure that the information on the file system is backed up.
- Make sure the partition you are resizing is the last partition on a particular disk.

Disclaimer: The following information has been provided by Red Hat, but is outside the scope of our posted Service Level Agreements (https://www.redhat.com/support/service/sla/) and support procedures. The information has been tested however is provided as-is and any configuration settings or installed applications made from the information in this article could make your Operating System unsupported by Red Hat Support Services and non-recoverable. The intent of this article is to provide you with information to accomplish your system needs. Use the information in this article at your own risk.

On disks with a GUID Partition Table (GPT), using the parted utility is recommended, as fdisk GPT support is in an experimental phase.

Procedure

1. Unmount the partition:

```
~]# umount /dev/vdb1
```

2. Run **fdisk** ***disk_name***.

For example:

```
~]# fdisk /dev/vdb Welcome to fdisk (util-linux 2.23.2). Changes will remain in memory only, until you decide to write them.
Be careful before using the write command. Command (m for help):
```

3. Check the partition number you wish to delete with the **p**. The partitions are listed under the heading “Device”.

For example:

```
Command (m for help): p Disk /dev/vda: 407.6 GiB, 437629485056 bytes, 854745088 sectors Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes I/O size (minimum/optimal): 4096 bytes / 4096 bytes Disklabel type:
dos Disk identifier: 0x5c873cba Partition 2 does not start on physical sector boundary. Device Boot Start End Blocks Id
System /dev/vda1 * 2048 1026047 512000 83 Linux /dev/vda2 1026048 1640447 307200 8e Linux LVM
```

4. Use the option **d** to delete a partition. If there is more than one, **fdisk** prompts for which one to delete.

For example:

```
Command (m for help): d Partition number (1,2, default 2): 2 Partition 2 has been deleted.
```

5. Use the option **n** to create a new partition. Follow the prompts and ensure you allow enough space for any future resizing that is needed. It is possible to specify a set, human-readable size instead of using sectors if this is preferred.

> *Note:* It is recommended to follow **fdisk**’s defaults as the default values (for example, the first partition sectors) and partition sizes specified are always aligned according to the device properties.

Warning: If you are recreating a partition in order to allow for more room on a mounted file system, ensure you create it with the same starting disk sector as before. Otherwise the resize operation will not work and the entire file system may be lost.

For example:

```
Command (m for help): n Partition type: p primary (1 primary, 0 extended, 3 free) e extended Select (default p): *Enter*
Using default response p. Partition number (2-4, default 2): *Enter* First sector (1026048-854745087, default 1026048):
*Enter* Last sector, +sectors or +size{K,M,G,T,P} (1026048-854745087, default 854745087): +500M Created a new partition 2 of
type 'Linux' and of size 500 MiB.
```

6. Check the partition table to ensure that the partitions are created as required using the **p** option.

For example:

```
Command (m for help): p Disk /dev/vda: 407.6 GiB, 437629485056 bytes, 854745088 sectors Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes I/O size (minimum/optimal): 4096 bytes / 4096 bytes Disklabel type:
dos Disk identifier: 0xf6e2b6cb Device Boot Start End Blocks Id System /dev/vda1 * 2048 1026047 512000 83 Linux /dev/vda2
1026048 2050047 512000 8e Linux LVM
```

7. Write the changes with the **w** option when you are sure they are correct.

Important: Errors in this process that are written could cause instability with the selected file system.

8. Run fsck on the partition.

```
~]# e2fsck /dev/vdb1 e2fsck 1.41.12 (17-May-2010) Pass 1:Checking inodes, blocks, and sizes Pass 2:Checking directory
structure Pass 3:Checking directory connectivity Pass 4:Checking reference counts Pass 5:Checking group summary information
ext4-1:11/131072 files (0.0% non-contiguous),27050/524128 blocks
```

9. Finally, if you need to increase or decrease the file system, refer to the How to Shrink an ext2/3/4 File System with resize2fs (https://access.redhat.com/articles/1196333), or the How to Grow an ext2/3/4 File System with resize2fs (https://access.redhat.com/articles/1196353).

If you don't need to increase or decrease the file system, mount the partition.

```
~]# mount /dev/vdb1
```

More Information

- **man fdisk**

  – man page for **fdisk**, has basic information about what fdisk is and what it supports.

- **man fstab**

  – man page for the /etc/fstab file including how to read the table.

- The **m** option within **fdisk**

  – this option lists all the available commands within **fdisk**.

Product(s) **Red Hat Enterprise Linux (/taxonomy/products/red-hat-enterprise-linux)**

Category **Customize or extend (/category/customize-or-extend)**

Tags **ext2 (/tags/ext2)** **ext3 (/tags/ext3)** **ext4 (/tags/ext4)** **file_systems (/tags/file_systems)** **storage (/tags/storage)**

Article Type **General (/article-type/general)**