Linux Disk Management Tutorial

This tutorial explains Linux File System (RAID, LVM, SWAP, VFAT, XFS, EXT4, EXT3, EXT2 and EXT), Linux partition scheme (GPT and MBR), Disk measurement units (GiB, MiB, KiB, GB, MB, KB, etc.) and System booting utility (BIOS and UEFI) in detail including how to add a new virtual hard in VMware step by step.

Linux disk management includes several important tasks such as adding or removing storage devices, creating and deleting partitions, mounting partitions on appropriate directories and making file system in partitions.

BIOS / UEFI

When we power on a computer, the first program which starts is either a BIOS or a UEFI. This is the starting point of computer. It performs a series of diagnostic test to detect and connect CPU, Memory, Keyboard, Hard disks and other peripherals. This process is known as Power on Self-Test (POST). If all peripherals are connected without any issue, BIOS/UEFI will find and execute the boot loader program.

Difference between BIOS and UEFI

| BIOS | UEFI |
| --- | --- |
| BIOS stands for Basic Input Output System. | UEFI stands for Unified Extensible Firmware Interface. |
| BIOS is the classical approach to detect and connect peripherals in system. | UEFI is the modern way to start the system. It is developed as the replacement of BIOS. |
| BIOS was invented for the personal computers in early 1982. | UEFI (based on Intel’s EFI) became available for all manufactures in 2007. |
| BIOS provides very basic functionality. It does not provide any troubleshooting utility. | UEFI provides several advanced features including backward compatibility. UEFI provides a pre-boot environment which has its own shell, drivers and applications. This environment includes basic features for troubleshooting such as remote diagnosis, emergency mode, Internet connectivity and storage backup. |
| All settings are stored in Nonvolatile CMOS Memory. | All settings are stored in Flash memory. |
| BIOS does not include driver for modern network card, video and storage devices. | Usually UEFI can detect all modern devices. |
| BIOS cannot boot from a partition which is larger than 2TiB. | The UEFI firmware can boot from a drive which is up to 9ZiB in size. |
| The BIOS runs in 16-bit processor mode, and has only 1 MB of space to execute in. | UEFI can run in 32-bit or 64-bit mode. |
| BIOS supports only MBR partition scheme. | UEFI supports both MBR and GPT partition scheme. |

This tutorial is the first of our article "**Linux Disk Management Explained in Easy Language with Examples**". You can read other parts of this tutorial here.

[Manage Linux Disk Partition with fdisk Command](https://www.computernetworkingnotes.com/rhce-study-guide/manage-linux-disk-partition-with-fdisk-command.html)

*This is the second part of this article. This part explains how to create primary, extended and logical partitions from fdisk command in Linux step by step with examples.*

[Manage Linux Disk Partition with gdisk Command](https://www.computernetworkingnotes.com/rhce-study-guide/manage-linux-disk-partition-with-gdisk-command.html)

*This is the third part of this article. This part explains how to create GPT (GUID partition table) partitions from gdisk command in Linux step by step with examples.*

[Linux Disk Management with parted command](https://www.computernetworkingnotes.com/rhce-study-guide/linux-disk-management-with-parted-command.html)

*This is the fourth part of this article. This part explains how to create primary, extended, logical and GPT partitions from parted command in Linux step by step with examples.*

[How to create SWAP partition in Linux](https://www.computernetworkingnotes.com/rhce-study-guide/how-to-create-swap-partition-in-linux.html)

*This is the fifth part of this article. This part explains how to create swap partition in Linux with examples including basic swap management tasks such as how to increase, mount or clear swap memory.*

[Learn how to configure LVM in Linux step by step](https://www.computernetworkingnotes.com/rhce-study-guide/learn-how-to-configure-lvm-in-linux-step-by-step.html)

*This is the sixth part of this article. This part explains basic concepts of LVM in detail with examples including how to configure and manage LVM in Linux step by step.*

[How to configure RAID in Linux Step by Step Guide](https://www.computernetworkingnotes.com/rhce-study-guide/how-to-configure-raid-in-linux-step-by-step-guide.html)

*This is the last part of this article. This part explains basic concepts of RAID in detail with examples including how to configure and manage software RAID in Linux step by step.*

Following figure illustrates a screen shot from BIOS

Following figure illustrates a screen shot from UEFI

From disk management points of view the important difference between BIOS and UEFI is that BIOS only supports MBR partition scheme while UEFI supports both MBR and GPT partition scheme. There’s no way to switch from BIOS to UEFI on an existing PC. We need to buy new hardware which supports and includes UEFI. Luckily majority of new computers use UEFI instead of traditional BIOS.

A hard disk needs partition table before it can be used. There are two types of partition table MBR and GPT. If system is equipped with BIOS, we can use only MBR partition table while if system is equipped with UEFI, we can use any partition table.

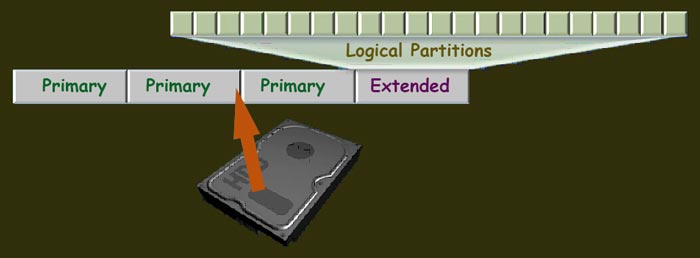
MBR Partition Scheme

This scheme was invented with the invention of personal computer in 1982. This scheme uses first sector of hard disk to store all necessary information which is required to boot the system including boot loader and partition table.

*Boot loader is a small script that is used to load the operating system. Partition table is the layout of hard disk.*

Default size of a sector is 512 bytes. From these 512 bytes only 64 bytes are used to store the partition table information. The partition table information is stored in 32 bit values. Due to this limited space, partition table can store information only about four partitions with a limitation of 2 TiB in each partition.

Gradually four partitions limit became an issue. To solve this issue a little change was made in MBR partition scheme. In this change one partition was allowed to use as extended partition. Within extended partition, a number of logical partitions could be created. How many logical partitions can be created is depend on which operating system is used. In Linux we can create up to 15 partitions. Regardless how many logical partitions we create in extended partition, the size of extended partition including all its logical partitions cannot be exceed 2 TiB.



*An extended partition is only used to create the logical partitions. It cannot be used to store the data directly.*

MBR partition scheme can be used in a hard disk which is 8TiB or less in size. To use a hard disk which is greater than 8TiB or to create a partition which is larger than 2TiB we need GPT partition scheme.

GPT Partition scheme

This scheme was invented with UEFI standards to meet with the requirement of modern storage devices. In GPT maximum size of a partition is 8 ZiB. GPT allows maximum 128 partitions. GPT uses a 128 bit global unique ID (GUID) to identify the partition. Since GPT provides a lot of partitions (128) and much bigger space in each partition there is no need to divide the partitions in primary, extended and logical partitions.

Difference between MBR and GPT

| MBR | GPT |
| --- | --- |
| MBR supports maximum 4 primary partitions. One primary partition can be used an extended partition. Extended partition can be used to create logical partitions. | GPT supports maximum 128 partitions. Primary, extended and logical partition method is not used in GPT. All partitions are equal. |
| Maximum partition size is 2TiB. | Maximum partition size is 8 ZiB. |
| Use 32 bits standard to store the partition table. | Use 64 bits standard to store the partition information. |
| All information is stored in first sector of hard disk. If this sector is corrupted, booting will be failed. | Information is stored in two locations. If one location is corrupted, system will boot from another location. |

In this tutorial I used measurement unit MiB and GiB instead of MB or GB because Linux use these units. There is a different of 24bytes in both units. Megabyte is the multiple of 1000 bytes while Mebibyte is the multiple of 1024 bytes. Following table lists the difference between these units.

Measurement units

| Value | Unit | Symbol | Value | Unit | Symbol |
| --- | --- | --- | --- | --- | --- |
| 10241 | Kibibyte | KiB | 10001 | Kilobyte | KB |
| 10242 | Mebibyte | MiB | 10002 | Megabyte | MB |
| 10243 | Gibibyte | GiB | 10003 | Gigabyte | GB |
| 10244 | Tebibyte | TiB | 10004 | Terabyte | TB |
| 10245 | Pebibyte | PiB | 10005 | Petabyte | PB |
| 10246 | Exbibyte | EiB | 10006 | Exabyte | EB |
| 10247 | Zebibyte | ZiB | 10007 | Zettabyte | ZB |
| 10248 | Yobibyte | YiB | 10008 | Yottabyte | YB |

Linux File System

We cannot create files and directories directly in the partition. Before we use a partition for data storage we need to create a file system on it. File system is a logical container that is used to store the files and directories. We can create a separate file system type in each partition or can use same file system type in all partitions. Each partition must be formatted with file system before it can be used for data storage. RHEL supports several file system types.

EXT (Extended File system)

This is the first Linux file system. It was used in early versions of Linux. It has been removed from RHEL.

EXT2 File System

This is the second generation of EXT file system. It provides very basic features of file system. It was developed in 1980. It was the default file system before RHEL5. It has been deprecated in RHEL7 and would be removed in next version.

EXT3 File System

This is third generation of EXT file system. This was the default file system in RHEL5. It includes several enhanced features.

* It supports file systems up to 16TiB in size.
* It supports file up to 2TiB in size.
* It supports up to 32000 subdirectories.
* It is the first file system in EXT series which supports journaling mechanism.

Unless you have any specific reason to use, there is no need to use this file system in RHEL7.

EXT4 File System

This is fourth generation of EXT file system. This was the default file system in RHEL6.

* It supports file systems up to 1EiB.
* It supports file up 16TiB in size.
* It supports unlimited directories.
* It uses a series of contiguous physical blocks on hard disk known as extents. The extents are used to improve the performance of very large files.

XFS (X-File System) File System

This file system was developed by Silicon Graphics for their version of UNIX. Later it was adopted by most Linux distributions including RHEL. This is the default file system in RHEL 7. This file system is based on 64-bit extent. It uses journaling for metadata operations. It supports file systems and files of sizes up to 8EiB. The only drawback of this system is that it does not support shrink like EXT3 and EXT4.

VFAT (Virtual File Allocation Table) File System

This is the enhanced version of legacy FAT file system. It was developed in 1995 for Windows95. Linux does not use this file system for any operation but if require it can read and write files written in this format. This support was added in Linux so a user can exchange data between Linux and Windows using an external device such as USB drive or compact disk.

NTFS File System

This is the enhance version of VFAT. This file system is the default file system in Windows system. Just like VFAT, Linux can read and write files written in this format but it does not use this format for its own purpose.

Besides above file systems Linux supports several other file systems but they are out of the scope of this tutorial. I will discuss them in separate tutorial.

SWAP Space

Swap space is the special space in hard disk that is used as a temporary memory. This space can be allocated as a separate swap partition, LVM partition or as a file (*file is used only to extend the available swap space*). Swap space is used only if a shortage of physical memory occurs. In shortage situation system moves recently unused data from memory to swap space. When requires, system moves back this data from swap to memory. This is the convenient way to improve kernel memory usage.

LVM (Logical Volume Manager)

Classical partition scheme is fixed in nature. It means, once created partition size cannot be changed later. We are not allowed to add additional space in a partition which is filled up with data. Same way we cannot shrink a partition which has a lot of unused free space. LVM not only solves this issue but also provides several other advantages over the classical approach. LVM is flexible in nature. We can shrink or grow a partition as per requirement.

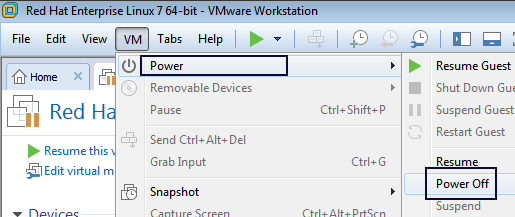
RAID

RAID is a mechanism to combine multiple disk drives into an array to achieve performance and redundancy. The array of disks will appear as a single logical storage unit or drive in computer. RAID enhances the data access speed and increases data storage capacity using a single virtual disk. There are two types of RAID; software RAID and hardware RAID. Hardware RAID is built from independent hard disks and raid controller device. It does not use any system resource. Software RAID is built from attached hard disks. It uses all resources from system.

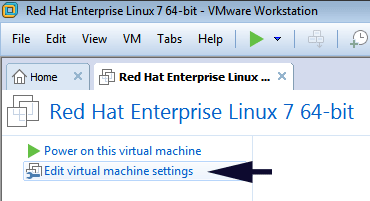
I will explain SWAP, LVM and RAID in detail with examples in next parts of this tutorial. For demonstration purpose I have installed a RHEL server in system in VMware workstation. In practice we will add/format/mount/remove partitions. We should not perform these actions on primary disk which contains the operating system and boot partition. For practice we will use two separate disks.

Adding new virtual hard disk in VMware

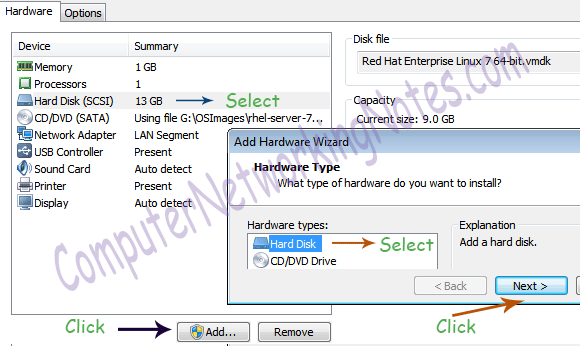
Since we cannot add or remove new hardware devices in running machine, power off the virtual machine.



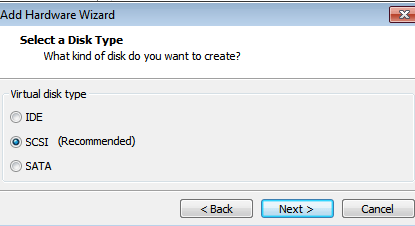
Click **Edit** virtual machine settings



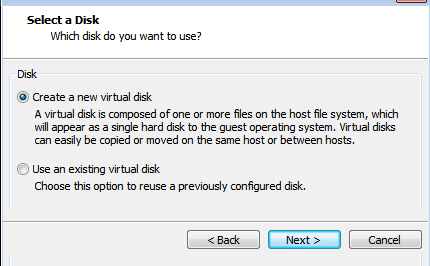
Select **Hard Disk** and click **Add** and select **Hard Disk** from opened window and click **Next**



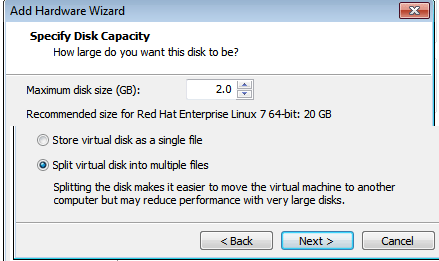
Select **SCSI** and click Next



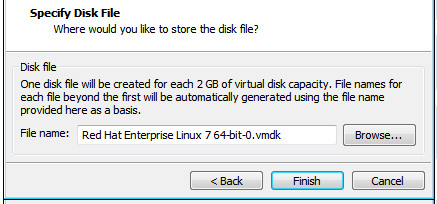
Select **Create a new virtual disk** and click Next



Select **Split virtual disk into multiple files** and click Next

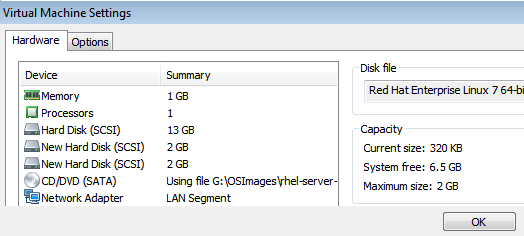


Click **Finish**

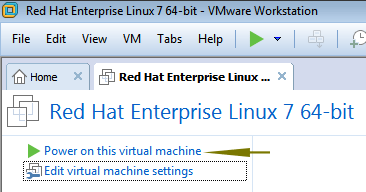


Add another hard disk of 2GB by following same steps.

When you are done, you should see the new hard disks in virtual machine hardware pane.



Now power on the virtual machine



In next part of this tutorial we will use this machine to understand the disk management practically. Before closing this introductory part let’s have a quick look on how storage devices are named in Linux.

Common disk device names

| /dev/had | IDE hard disk |
| --- | --- |
| /dev/sda | SCSI or SATA hard disk |
| /dev/vda | Virtual machine hard disk which uses virtio disk driver |
| /dev/xvda | Virtual machine hard disk which uses Xen disk driver |

Name of first disk starts with letter **a** for example all disks name listed in above table represent first disk in their type. If system has multiple disks, the next available letter will be used for example sdb represents second SCSI disk, sdc represents third SCSI disk and sdf represents sixth SCSI disk.

That’s all for this part. In next part I will explain how to manage MBR partitions step by step with examples.