Managing Disks and Swap Space

Block Devices

Block devices under Linux are referenced by **special files** found in the **/dev** directory:

- hd[a-d]
  - IDE disks and ATAPI devices
- sd[a-z]
  - SCSI/SATA disks
- scd[0-7]
  - SCSI CDRoms
- xd[a-d]
  - The original IBM XT disks
- fd[0-7]
  - Floppy drives
- st[0-7]
  - SCSI streamers that can rewind
- nst[0-7]
  - SCSI streamers that cannot rewind
- rmt8, rmt16, tape-d, tape-reset
  - QIC-80 devices
- ram[0-15]
  - Virtual memory disks
- loop[0-15]
  - loop devices used to access a filesystem in a file (such as an ISO file)
- md[x]
  - Software **RAID** device
- vg[x]
  - Volume Group
- lv[x]

    - Logical Volume

Partitions

Master Boot Record

This partition type, also know as **MBR**, **MS-DOS Partitions** or **BIOS partitions**, is the most common one on disks under 2 **tebibytes** in size. That limitation comes from the fact that it uses 32 bit pointers to refer to disk sectors. Most disk manufacturers use a sector size of 512 bytes. Taking these two points into account indicates that the maximum disk size permitted is $2^{32} * 512$ bytes = $2.2 * 10^{12}$ bytes = 2TiB.

<note important> Some disk manufacturers are now moving to 4096-byte sectors which effectively increases the maximum size permitted to 16 TiB. </note>

Generally, a PC is equiped with two disk controllers each being able to manage two disks refered to respectively as the **Master** and **Slave**. Each disk is refered to differently :

- Controller 0
    - Master
        - **hda** - IDE disk
        - **sda** - SATA/SCSI disk
    - Slave
        - **hdb** - IDE disk
        - **sdb** - SATA/SCSI disk
- Controller 1
    - Master
        - **hdc** - IDE disk
        - **sdc** - SATA/SCSI disk
    - Slave
        - **hdd** - IDE disk
        - **sdd** - SATA/SCSI disk

Each disk can have three types of partitions:

- **Primary Partitions**,
  - Maximum of **4**. The **FAT** (*File Allocation Table*) is 64 bytes in length and 16 bytes are needed to code each partition,
- **Extended Partitions**,
  - Normally only one extended partition per disk. An Extended Partition contains Logical Drives.
- **Logical Drives**.

The 4 primary partions are numbered 1 through 4:

- **hda1**, **hda2**, **hda3** and **hda4** for the IDE Master on Controller 0,
- **sda1**, **sda2**, **sda3** and **sda4** for the **SCSI/SATA** Master on Controller 0.

;#;;#;

An Extended Partition contains Logical Drives that start at **hda5** or **sda5** and takes the place of a Primary Partition, rendering it unusable:

;#;;#;

<note important> Linux does not suffer from the same problem as Windows™ when using MBR. Linux does **not** have to be booted from a primary partition whereas Windows™ does. </note>

The total number of partitions on a disk is limited as follows:

- **IDE**,
  - Upto **63**,
- **SCSI**,
  - Upto **15**,
- Disks using the **libata** API,
  - Upto **15**.

<note important> These limits can be exceeded by using the **LVM** technology. </note>

Apple Partition Map

Also know as **APM**, this partition table type was used by Apple on its PowerPC based Macintoshes. Apple switched to GPT when it started to use Intel

CPUs.

<note important> APM shares the same limitations as MBR concerning the maximum size of the disk. </note>

GUID Partition Table

Also known as **GPT**, this partition table type is defined in the **UEFI** (**U**nified **E**xtensible **F**irmware **I**nterface) definition.

Under GPT, there is no difference between primary, extended or logical partitions, instead GPT supports a fixed number of partitions which is **128** by default

GPT uses 64-bit sector pointers and can therefore handle disks upto a size of $9.4 * 10^{21}$ bytes = 8ZiB (**zebibytes**)

Partitioning

Partitioning can be performed by several programs, some of which are partition-type dependant:

- **The libparted Tools**,
  - contain the **parted** text-mode program,
  - are at the heart of the **GParted** graphical program,
  - are compatible with MBR, APM and GPT partiton table types,
- **The fdisk Family**,
  - the **fdisk** program for MBR partition table types included in the **util-linux** or **util-linux-ng** packages,
  - the **cfdisk** program for MBR partition table types included in the **util-linux** or **util-linux-ng** packages,
- **GPT fdisk**,
  - a package that supplies the **gdisk** and **sgdisk** programs which are fdisk-like and cfdisk-like programs for GPT partiton types.

LAB #1 - Using fdisk

Partioning under Linux can be accomplished using the **fdisk** utility:

```
[root@centos ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').

Command (m for help):
```

Hit the **m** key and ↵ Enter to see a menu of the commands available:

```
Command (m for help): m
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)

Command (m for help):
```

<note important> To create a new partition you need to use the **n** command. </note>

Create the following partitions on your disk:

Partition	Type	Size
/dev/sda4	Extended	From the first available cylinder to the last available cylinder.
/dev/sda5	Logical	500 MB
/dev/sda6	Logical	200 MB
/dev/sda7	Logical	300 MB
/dev/sda8	Logical	500 MB
/dev/sda9	Logical	400 MB
/dev/sda10	Logical	500 MB
/dev/sda11	Logical	500 MB
/dev/sda12	Logical	200 MB

When you have finished, use the **p** command to see the resulting partition table:

```
Command (m for help): p

Disk /dev/sda: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00098187

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          13      102400   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2              13         651     5120000   83  Linux
Partition 2 does not end on cylinder boundary.
/dev/sda3             651         912     2096128   82  Linux swap / Solaris
Partition 3 does not end on cylinder boundary.
/dev/sda4             912        2610    13645273    5  Extended
/dev/sda5             912         976      520136+  83  Linux
/dev/sda6             977        1002      208813+  83  Linux
/dev/sda7            1003        1041      313236   83  Linux
```

```
/dev/sda8            1042        1106      522081   83  Linux
/dev/sda9            1107        1158      417658+  83  Linux
/dev/sda10           1159        1223      522081   83  Linux
/dev/sda11           1224        1288      522081   83  Linux
/dev/sda12           1289        1314      208813+  83  Linux

Command (m for help):
```

<note important> Each block contains 1,024 bytes. Each sector contains 512 bytes. When a partition contains an uneven number of sectors it is marked with a **+**. </note>

<note important> Note that the first three partitions show an error - **Partition X does not end on cylinder boundary.** </note>

This error seems to imply that the partitions overlap. In order to check if they really do, change the units used in the output from cylinders to sectors by using the **u** command :

```
Command (m for help): u
Changing display/entry units to sectors

Command (m for help): p

Disk /dev/sda: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00098187

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *        2048      206847      102400   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2          206848    10446847     5120000   83  Linux
Partition 2 does not end on cylinder boundary.
/dev/sda3        10446848    14639103     2096128   82  Linux swap / Solaris
```

```
Partition 3 does not end on cylinder boundary.
/dev/sda4        14639104    41929649    13645273    5  Extended
/dev/sda5        14639167    15679439      520136+  83  Linux
/dev/sda6        15679503    16097129      208813+  83  Linux
/dev/sda7        16097193    16723664      313236   83  Linux
/dev/sda8        16723728    17767889      522081   83  Linux
/dev/sda9        17767953    18603269      417658+  83  Linux
/dev/sda10       18603333    19647494      522081   83  Linux
/dev/sda11       19647558    20691719      522081   83  Linux
/dev/sda12       20691783    21109409      208813+  83  Linux

Command (m for help):
```

<note important> Note that the first three partitions still show an error - **Partition X does not end on cylinder boundary.** However if you study the Start and End sectors of each partition you can see that they do not overlap. </note>

Write the partition table to disk using the **w** command and restart your virtual machine:

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
[root@centos ~]# reboot
```

Login and launch fdisk to check if your changes have been taken into account:

```
[trainee@centos ~]$ su -
Password:
[root@centos ~]# fdisk /dev/sda
```

```
WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').

Command (m for help): p

Disk /dev/sda: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00098187

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          13      102400   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2              13         651     5120000   83  Linux
Partition 2 does not end on cylinder boundary.
/dev/sda3             651         912     2096128   82  Linux swap / Solaris
Partition 3 does not end on cylinder boundary.
/dev/sda4             912        2610    13645273    5  Extended
/dev/sda5             912         976      520136+  83  Linux
/dev/sda6             977        1002      208813+  83  Linux
/dev/sda7            1003        1041      313236   83  Linux
/dev/sda8            1042        1106      522081   83  Linux
/dev/sda9            1107        1158      417658+  83  Linux
/dev/sda10           1159        1223      522081   83  Linux
/dev/sda11           1224        1288      522081   83  Linux
/dev/sda12           1289        1314      208813+  83  Linux

Command (m for help):
```

Exit fdisk using the **q** command.

Journaled Filesystems

Presentation

A **journal** is part of a **journaled** or **journaling** filesystem. It's role is to keep track of any write operations in order to guarantee data integrity in the case of a system crash.

Red Hat Linux can use one of the following two filesystems:

- Ext3
- Ext4
- ReiserFS
- XFS
- JFS

<note important> Red Hat only supports EXT2/3/4 filesystems. You cannot create a journaling filesystem on a floppy disk. </note>

<note> You can compare filesystems by consulting **this page** </note>

Ext3

Ext3 is a journaling filesystem 100% compatible with the traditional Ext2 filesystem. The principal difference between the two is the addition of the journal.

The commands used to manage an Ext3 filesystem are :

Command	**Description**
mke2fs -j	Create a filesystem
mke2fs -t ext3	Create a filesystem
mkfs.ext3	Create a filesystem
fsck	Check/Repair a filesystem

Command	Description
e2fsck	Check/Repair a filesystem
tune2fs	Tune a filesystem
debugfs	Debug a filesystem
dump2fs	Obtain information about the filesystem

<note important> The fsck program is normally called automatically at boot every 6 months or 20 reboots whichever comes first. </note>

For more information concerning Ext3, please see **this page**

Ext4

The **Ext4** filesystem was first introduced with the **2.6.19** kernel. It became stable in the **2.6.28** kernel.

Ext4, although not an evolution of the Ext3 filesystem, is backward compatible with the latter.

The major characteristics of an Ext4 filesystem are:

- volume sizes of upto **1 024 pebioctets** (1 pebioctet (Pio) = 250 octets = 1024 Tio = 1125899906842624 octets),
- space allocation by using **extents**. An extent is a contiguous area of storage reserved for a file.

Extents were introduced with the **2.6.23** kernel.

The backward compatibility with Ext3 means that:

- you can mount an Ext3 filesystem as an Ext4 filesystem,
- you can mount an Ext4 filesystem as an Ext3 filesystem except when extents have been turned on.

The commands used to manage an Ext4 filesystem are :

Command	Description
mke2fs -t ext4	Create a filesystem
mkfs.ext4	Create a filesystem

Command	Description
fsck.ext4	Check/Repair a filesystem
e2fsck	Check/Repair a filesystem
tune4fs	Tune a filesystem
debugfs	Debug a filesystem
dump2fs	Obtain information about the filesystem

For more information concerning Ext4, please see **this page**.

<note important> When an ext2/ext3/ext4 filesystem is formated by default 5% is reserved for root. Reserved space is supposed to reduce fragementation and allow root to login in case the filesystem becomes 100% used. You can use tune2fs to reduce the amount of reserved space as follows **tune2fs -m n /dev/sdXY** where n is the new percentage to reserve. </note>

ReiserFS

The principal adavantage of ReiserFS is that is is much more efficient than Ext3 at storing files of a size of a couple of KB. This can lead to a 10% disk space gain when compared to Ext3.

The commands used to manage an ReiserFS filesystem are :

Command	Description
mkreiserfs	Create a filesystem
mkfs.reiserfs	Create a filesystem
reiserfsck	Check/Repair a filesystem
reiserfstune	Tune a filesystem
debugreiserfs	Debug a filesystem
debugreiserfs	Obtain information about the filesystem

For more information concerning ReiserFS , please see **this page**.

XFS

XFS is a 64-bit journaling filesystem created by SGI for its IRIX operating system. XFS was introduced into Linux in the 2.6.xx kernels.

The commands used to manage an XFS filesystem are :

Command	Description
mkfs.xfs	Create a filesystem
xfs_check / xfs_repair	Check/Repair a filesystem
xfs_admin	Tune a filesystem
xfs_db	Debug a filesystem
xfs_info	Obtain information about the filesystem. Requires the filesystem to be mounted.

For more information concerning XFS, please see **this page**.

Swap Space

Swap Size

The following table gives the recommended swap size as a function of the quantity of RAM in the system:

RAM	Swap Size
4 GB or less	2 GB
4 GB to 16 GB	4 GB
16 GB to 64 GB	8 GB
64 GB to 256 GB	16 GB

Swap Partitions

A swap partition can be created on:

- a Disk Partition,
- a Software RAID device,
- a Logical Volume.

The swapon Command

To see a list of the current swap devices, use the **swapon** command with the **-s** switch.

```
[root@centos ~]# swapon -s
Filename                Type        Size    Used    Priority
/dev/sda3                               partition   2096120    0    -1
```

<note important> Note that in the above example the swap is not being used. There is also a notion of swap priority which we will detail later. </note>

Command Line Switches

The switches associated with this command are:

```
[root@centos ~]# swapon --help

Usage:
 swapon -a [-e] [-v] [-f]             enable all swaps from /etc/fstab
 swapon [-p priority] [-v] [-f] <special>  enable given swap
 swapon -s                            display swap usage summary
 swapon -h                            display help
 swapon -V                            display version

The <special> parameter:
 {-L label | LABEL=label}             LABEL of device to be used
 {-U uuid  | UUID=uuid}               UUID of device to be used
 <device>                             name of device to be used
```

```
 <file>                              name of file to be used
```

<note important> Note that the **-p** switch is used to set the swap priority. </note>

La Commande swapoff

In our case the swap partition is **/dev/sda3**. To turn off the swap, use the following command:

```
[root@centos ~]# swapoff /dev/sda3
[root@centos ~]# swapon -s
Filename                Type        Size    Used    Priority
```

To turn the swap back on, use the **swapon** command:

```
[root@centos ~]# swapon /dev/sda3
[root@centos ~]# swapon -s
Filename                Type        Size    Used    Priority
/dev/sda3                               partition   2096120    0    -1
```

Command Line Switches

The switches associated with the swapoff command are:

```
[root@centos ~]# swapoff --help

Usage:
 swapoff -a [-v]                     disable all swaps
 swapoff [-v] <special>              disable given swap
 swapoff -h                          display help
 swapoff -V                          display version
```

```
The <special> parameter:
 {-L label | LABEL=label}            LABEL of device to be used
 {-U uuid  | UUID=uuid}              UUID of device to be used
 <device>                            name of device to be used
 <file>                              name of file to be used
```

The /etc/fstab file

For each swap partition, there must be an entry in the **/etc/fstab** file:

```
[root@centos ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Fri Oct 25 09:32:46 2013
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=c7b1d3e8-6471-4cba-947b-430db974e774 /                       ext4    defaults        1 1
UUID=d8988475-7dc7-4a61-8081-6153b7c9551b /boot                   ext4    defaults        1 2
UUID=a1d6a043-6f10-4f60-bb9c-aaaac9632c57 swap                    swap    defaults        0 0
tmpfs                   /dev/shm                tmpfs   defaults        0 0
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
sysfs                   /sys                    sysfs   defaults        0 0
proc                    /proc                   proc    defaults        0 0
```

Each line in this file has 6 fields :

Field 1	**Field 2**	**Field 3**	**Field 4**	**Field 5**	**Field 6**
Special file or UUID or Virtual File System	Mount Point	Filesystem Type	Comma separated list of options	Used by the dump command (1 = dump, 0 or empty = do not dump)	The order in which the *fsck* command checks the disks/partitions at boot time

The **UUID** (*Universally Unique Identifier*) is a randomly generated 128 bit string that is automatically generated by the system when a filesystem is created on the partition.

<note> Please see the manual for the **mount** command to document yourself on the different options available in field 4. </note>

Swap Files

You can also use a file as swap space. Create a swap file of 256 MB using the dd command:

```
[root@centos ~]# dd if=/dev/zero of=/swap bs=1024k count=256
256+0 records in
256+0 records out
268435456 bytes (268 MB) copied, 5.62261 s, 47.7 MB/s
```

To set up this file as swap space you need to use the following command:

```
[root@centos ~]# mkswap /swap
mkswap: /swap: warning: don't erase bootbits sectors
        on whole disk. Use -f to force.
Setting up swapspace version 1, size = 262140 KiB
no label, UUID=77dc521a-e239-4be2-8764-ab5b31fde1ed
```

Now activate the swap file with a priority of 3:

```
[root@centos ~]# swapon -p3 /swap
```

Pour visualiser les éspaces swap, saisissez la commande suivante :

```
[root@centos ~]# swapon -s
Filename                Type        Size    Used    Priority
/dev/sda3                               partition   2096120    0    -1
/swap                                   file        262136    0    3
```

<note important> The swap file has a priority of 3. This means it will be used in preference to the swap partition that has a lower priority. </note>

In order to activate the swap file at boot time, you need to edit the **/etc/fstab** file:

fstab

```
#
# /etc/fstab
# Created by anaconda on Fri Oct 25 09:32:46 2013
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=c7b1d3e8-6471-4cba-947b-430db974e774 /                       ext4    defaults        1 1
UUID=d8988475-7dc7-4a61-8081-6153b7c9551b /boot                   ext4    defaults        1 2
UUID=a1d6a043-6f10-4f60-bb9c-aaaac9632c57 swap                    swap    defaults        0 0
/swap                                   swap                    swap      defaults      0 0
tmpfs                   /dev/shm                tmpfs   defaults        0 0
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
sysfs                   /sys                    sysfs   defaults        0 0
proc                    /proc                   proc    defaults        0 0
```

<note important> Do not modify your /etc/fstab file since you are going to delete the swap file. </note>

Now turn off the swap file:

```
[root@centos ~]# swapoff /swap
[root@centos ~]# swapon -s
Filename                Type        Size    Used    Priority
/dev/sda3                               partition   2096120    0    -1
```

Now delete the swap file:

```
[root@centos ~]# rm /swap
rm: remove regular file `/swap'? y
```

~~DISCUSSION:off~~
