INDEX

SN	Experiment Title	Date	Page Number	Remark
1	Basics concept- Introduction to Linux Operating system, Basic command in Linux and writing shell script in Vi editor.			
2	Process management 1.Use ps to search for the init process by name. 2 what is the process id of init process 3 use the who am I command to determine terminal name 5 what is the process id of your shell 6 what is the parent process id of your shell 7 start two instances of sleep 4432 in the background 8 locate the process id of all sleep commands 9 display those two sleep process on top. 10 use the standard kill to kill one of the process			
3	Process priorities: 1Create a new directory and create six pipes in that directory. 2.Bounce another character between two other pipes but this same time start the command nice, verify that all cat processes are battling to the CPU. 3.Use ps to verify that the two new cat process have a nice value. Use the o and c. 4.use renice to increase the nice value from 10 to 15. Notice the difference between the usual commands.			

4	Disk partioions 1 use fdisk -l to display existing partitions and sizes 2compare the output of fdisk and df 3 compare the output of fdisk Create a 200MB primary partition on a small disk 4Create a 400 MB primary partion and two 300 MB logical drives on a big disk 4 Use df-h and f disk -l to verify your work 5 Compare the output again fdisk and df.			
5	Logical volume management : 1) Create a volume group that contains a complete disk and partition onanother disk 2) Create two logical volumes (as small one and a bigger one) in this volumegroup. Format them with ext 3, mount them and copy some files tothem.			

6	File systems 1.List the file systems that are known by your system. 2.Create ab ext2 file system on the 200MB partition. 3.Create an ext3 file system on one of the 300MB logical drives. 4.Create an ext4 on the 400MB partition. 5.Set the reserved space for the root On the ext3 file system to 0 percent. 6.Verify your work with fdisk and df. 7.Perform a file system check on all The new file systems.			
7	Scheduling 1.Schedule two jobs with at, display The at queue and remove a job. 2.As normal user, use crontab -e To schedule a script to run every four minutes. 3.As a root, display the crontab file of your normal user. 4.As the normal user again, remove your crontab file 5.Take a look at the cron files and directories in/etc and understand Them. What is the runparts command doing?			

8	Memory Management 1.Use dmevg to find the total amount of memory in your computer. 2.Use free to display memory usage In kilobytes.(then in megabytes) 3.On a virtual machine , create a swap partitions. 4.Add a 20 megabytes swap file to the system. 5.Put all swap spaces in/etc/fstab and activate them. Test with a reboot that they are mounted. 6.Use free to verify usage of current swap			

Date : / /

PRACTICAL 5

AIM: Logical volume management :

3) Create a volume group that contains a complete disk and partition onanother disk

4) Create two logical volumes (as small one and a bigger one) in this volumegroup. Format them with ext 3, mount them and copy some files tothem.

THEORY:

Logical Volume Management :

In computer storage, logical volume management or LVM provides a method of allocating space on mass-storage devices that is more flexible than conventional partitioning schemes to store volumes. In particular, a volume manager can concatenate, stripe together or otherwise combine partitions (or block devices in general) into larger virtual partitions that administrators can re-size or move, potentially without interrupting system use.

Volume management represents just one of many forms of storage virtualization; its implementation takes place in a layer in the device-driver stack of an operating system (OS) (as opposed to within storage devices or in anetwork).

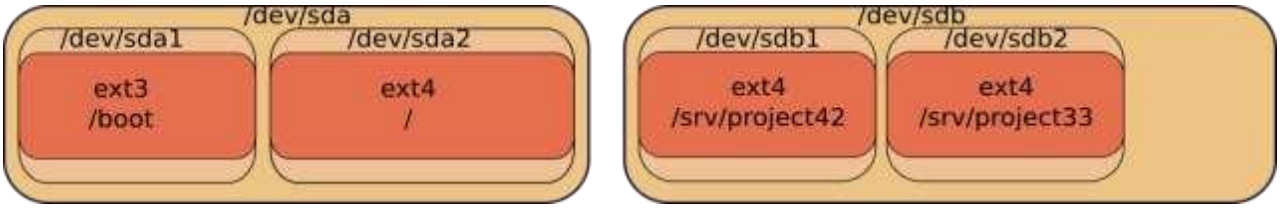
In Linux, Logical Volume Manager (LVM) is a device mapper framework that provides logical volume management for the Linux kernel. Most modern Linux distributions are LVM-aware to the point of being able to have their root file systems on a logical volume.

INTRODUCTION TO LVM

Problems with standard partitions

There are some problems when working with hard disks and standard partitions.

Consider a system with a small and a large hard disk device, partitioned like this. The first disk (/dev/sda) is partitioned in two, the second disk (/dev/sdb) has two partitions and some empty space.

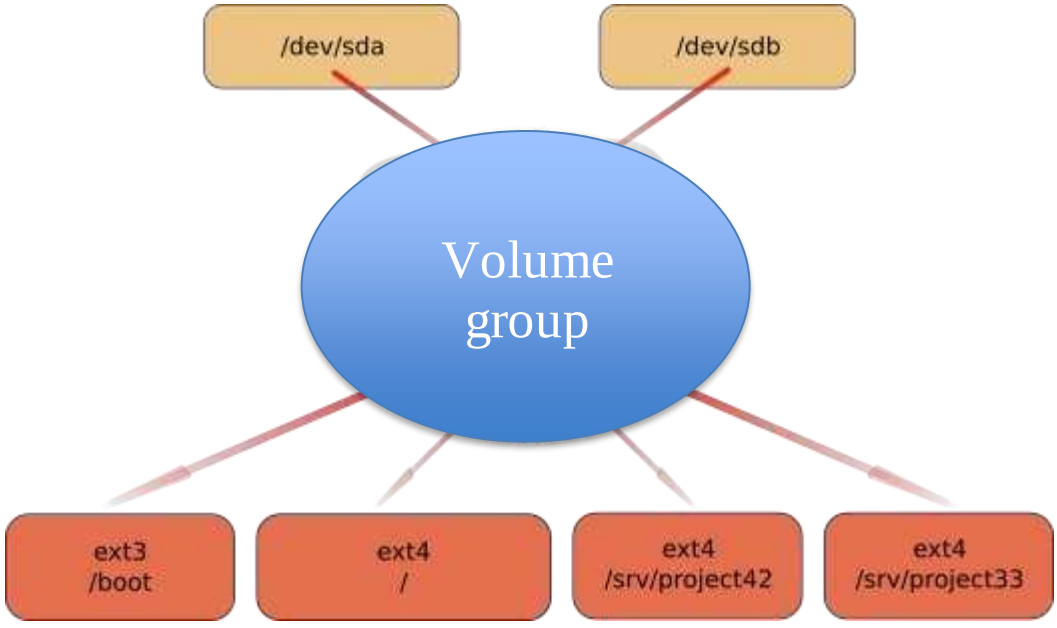


Partition in disk

In the example above, consider the options when you want to enlarge the space available for **/srv/project42**. What can you do ? The solution will always force you to unmount the file system, take a backup of the data, remove and recreate partitions, and then restore the data and remount the file system.

solution with lvm

Using **lvm** will create a virtual layer between the mounted file systems and the hardware devices. This virtual layer will allow for an administrator to enlarge a mounted file system in use. When **lvm** is properly used, then there is no need to unmount the file system to enlarge it.



lvm terminology

physical volume (pv)

A **physical volume** is any block device (a disk, a partition, a RAID device or even an iSCSI device). All these devices can become a member of a **volume group**.

The commands used to manage a **physical volume** start with pv.

```
[root@centos65 ~]# pv
pvchange   pvck       pvcreate   pvdisplay
           pvmove pvremove
pvresize   pvs        pvscan
```

volume group (vg)

A **volume group** is an abstraction layer between **block devices** and **logical volumes**.

The commands used to manage a **volume group** start with vg.

```
[root@centos65 ~]# vg
vgcfgbackup    vgconvert      vgextend       vgmknodes
vgs
vgcfgrestore   vgcreate       vgimport       vgreduce
vgscan
vgchange       vgdisplay      vgimportclone  vgremove
vgsplit
vgck           vgexport       vgmerge        vgrename
```

logical volume (lv)

A **logical volume** is created in a **volume group**. Logical volumes that contain a file system can be mounted. The use of **logical volumes** is similar to the use of **partitions** and is accomplished with the same standard commands (mkfs, mount, fsck, df, ...).

The commands used to manage a **logical volume** start with lv.

```
[root@centos65 ~]# lv
```

```
lvchange     lvextend     lvmdiskscan  lvmsar
lvresize
lvconvert    lvm          lvmdump      lvreduce     lvs


lvcreate     lvmchange    lvmetad      lvremove
lvscan
lvdisplay    lvmconf      lvmsadc      lvrename
```

pvs

The easiest way to verify whether devices are known to lvm is with the **pvs** command. The screenshot below shows that only /dev/sda2 is currently known for use with LVM. It shows that /dev/sda2 is part of Volgroup00 and is almost 16GB in size. It also shows /dev/sdc and /dev/sdd as part of vg33. The device /dev/sdb is knwon to lvm, but not linked to any Volume Group.

```
 [root@RHEL5 ~]# pvs
  PV          VG          Fmt  Attr PSize   PFree
   /dev/sda2  VolGroup00  lvm2 a-    15.88G      0
  /dev/sdb                lvm2 --   409.60M 409.60M
  /dev/sdc    vg33        lvm2 a-   408.00M 408.00M
  /dev/sdd    vg33        lvm2 a-   408.00M 408.00M
[root@RHEL5  ~]#
```

Pvscan

```
[root@RHEL5 ~]# vgs
  VG         #PV #LV#SNAttr     VSize
             VFree VolGroup00          1 2
             0wz--n-15.88G      0
[root@RHEL5 ~]#
```

vgscan

The **vgscan** command will scan all disks for existing Volume Groups. It will also update the **/etc/lvm/.cache** file. This file contains a list of all current lvm devices.

```
[root@RHEL5 ~]# vgscan
  Reading allphysicalvolumes.    This may take a
while...
  Found volume group "VolGroup00" using metadata type
lvm2
[root@RHEL5 ~]#
```

LVM will run the vgscan automatically at boot-up, so if you add hot swap devices, then you will need to run vgscan to update /etc/lvm/.cache with the new devices.

vgdisplay

The **vgdisplay** command will give you more detailed information about a volume group (or about all volume groups if you omit the argument).

```
[root@RHEL5 ~]# vgdisplay VolGroup00
  --- Volume group ---
  VG Name               VolGroup00
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  3
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                2
  Open LV               2
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               15.88 GB
  PE Size               32.00 MB
  Total PE              508
```

```
  Alloc PE/Size         508 / 15.88 GB
  Free  PE/Size         0 /0
  VGUUID                qsXvJb-71qV-9l7U-ishX-
FobM- qptE-VXmKIg

[root@RHEL5 ~]#
```

Verifying existing logical volumes

lvs

Use **lvs** for a quick look at all existing logical volumes. Below you can see two logical volumes named LogVol00 and LogVol01.

```
[root@RHEL5 ~]# lvs
  LV       VG         Attr   LSize  OriginSnap%
           Move LogCopy%
  LogVol00 VolGroup00 -wi-ao 14.88G
  LogVol01VolGroup00-wi-ao    1.00G
[root@RHEL5 ~]#
```

lvscan

The **lvscan** command will scan all disks for existing Logical Volumes.

```
[root@RHEL5 ~]# lvscan
  ACTIVE            '/dev/VolGroup00/LogVol00'[14.88
GB]inherit
  ACTIVE            '/dev/VolGroup00/LogVol01'[1.00
GB]inherit
[root@RHEL5 ~]#
```

lvdisplay

More detailed information about logical volumes is available through the **lvdisplay(1)** command.

```
[root@RHEL5 ~]# lvdisplay VolGroup00/LogVol01
  --- Logical volume ---
```

```
  LVName                 /dev/VolGroup00/LogVol01
  VGName                 VolGroup00
  LVUUID                 RnTGK6-xWsi-t530-ksJx-
7cax- co5c-A1KlDp
  LVWriteAccess          read/write
  LVStatus               available
  #open                  1
  LVSize                 1.00GB
  CurrentLE              32
  Segments               1
  Allocation             inherit
  Readaheadsectors       0
  Blockdevice            253:1

[root@RHEL5 ~]#
```

Manage volume groups

vgcreate

Use the **vgcreate** command to create a volume group. You can immediately name all the physical volumes that span the volume group.

```
[root@RHEL5 ~]# vgcreate vg42 /dev/sde /dev/sdf
  Volume group "vg42" successfully created
[root@RHEL5 ~]#
```

vgextend

Use the **vgextend** command to extend an existing volume group with a physical volume.

```
[root@RHEL5 ~]# vgextend vg42 /dev/sdg
  Volume group "vg42" successfully extended
[root@RHEL5 ~]#
```

vgremove

Use the **vgremove** command to remove volume groups from lvm. The volume

groups may not be in use.

```
[root@RHEL5 ~]# vgremove vg42
  Volume group "vg42" successfully removed
[root@RHEL5 ~]#
```

vgreduce

Use the **vgreduce** command to remove a Physical Volume from the Volume Group.

The following example adds Physical Volume /dev/sdg to the vg1 Volume Group using vgextend. And then removes it again using vgreduce.

```
[root@RHEL5 ~]# pvs | grep sdg
  /dev/sdg              lvm2--    819.20M
819.20M [root@RHEL5 ~]# vgextend vg1/dev/sdg
  Volume group "vg1" successfully extended
[root@RHEL5 ~]# pvs | grep sdg
  /dev/sdg   vg1        lvm2a-    816.00M
816.00M [root@RHEL5 ~]# vgreduce vg1/dev/sdg
  Removed "/dev/sdg" from volume group "vg1"
[root@RHEL5 ~]# pvs | grep sdg
  /dev/sdg              lvm2--    819.20M819.20M
```

vgchange

Use the **vgchange** command to change parameters of a Volume Group.

This example shows how to prevent Physical Volumes from being added or removed to the Volume Group vg1.

```
[root@RHEL5 ~]# vgchange -xn vg1
  Volume group "vg1" successfully changed
[root@RHEL5 ~]# vgextend vg1 /dev/sdg
  Volume group vg1 is not resizable.
```

You can also use vgchange to change most other properties of a Volume Group. This example changes the maximum number of Logical Volumes andmaximum

number of Physical Volumes that vg1 can serve.

```
[root@RHEL5 ~]# vgdisplay vg1 | grep -i max
  MAXLV                0
  MaxPV                0
[root@RHEL5 ~]# vgchange -l16 vg1
  Volume group "vg1" successfully changed
[root@RHEL5 ~]# vgchange -p8 vg1
  Volume group "vg1" successfully changed
[root@RHEL5 ~]# vgdisplay vg1 | grep -i max
  MAXLV                16
  MaxPV                8
```

vgmerge

Merging two Volume Groups into one is done with **vgmerge**. The following example merges vg2 into vg1, keeping all the properties of vg1.

```
[root@RHEL5 ~]# vgmerge vg1 vg2
  Volume group "vg2" successfully merged into "vg1"
[root@RHEL5 ~]#
```

Manage logical volumes

lvcreate

Use the **lvcreate** command to create Logical Volumes in a Volume Group. This example creates an 8GB Logical Volume in Volume Group vg42.

```
[root@RHEL5 ~]# lvcreate -L5G vg42
  Logical volume "lvol0" created
[root@RHEL5 ~]#
```

As you can see, lvm automatically names the Logical Volume **lvol0**. The next example creates a 200MB Logical Volume named MyLV in Volume Group vg42.

```
[root@RHEL5 ~]# lvcreate -L200M -nMyLV vg42
  Logical volume "MyLV" created
```

```
[root@RHEL5 ~]#
```

The next example does the same thing, but with different syntax.

```
[root@RHEL5 ~]# lvcreate --size 200M -n MyLV vg42
  Logical volume "MyLV" created
[root@RHEL5 ~]#
```

This example creates a Logical Volume that occupies 10 percent of the Volume Group.

```
[root@RHEL5 ~]# lvcreate -l 10%VG -n MyLV2 vg42
  Logical volume "MyLV2" created
[root@RHEL5 ~]#
```

This example creates a Logical Volume that occupies 30 percent of the remaining free space in the Volume Group.

```
[root@RHEL5 ~]# lvcreate -l 30%FREE -n MyLV3 vg42

  Logical volume "MyLV3" created
[root@RHEL5 ~]#
```

lvremove

Use the **lvremove** command to remove Logical Volumes from a Volume Group. Removing a Logical Volume requires the name of the Volume Group.

```
[root@RHEL5 ~]# lvremove vg42/MyLV
Do you really want to remove active logical volume
"MyLV"? [y/n]: y
  Logical volume "MyLV" successfully removed
[root@RHEL5 ~]#
```

Removing multiple Logical Volumes will request confirmation for each individual volume.

```
[root@RHEL5 ~]# lvremove vg42/MyLV vg42/MyLV2
vg42/MyLV3
Do you really want to remove active logical volume
"MyLV"? [y/n]: y
  Logical volume "MyLV" successfully removed
```

```
Do you really want to remove active logical volume
"MyLV2"? [y/n]: y
  Logical volume "MyLV2" successfully removed
Do you really want to remove active logical volume
"MyLV3"? [y/n]: y
  Logical volume "MyLV3" successfully removed
[root@RHEL5 ~]#
```

lvextend

Extending the volume is easy with **lvextend**. This example extends a 200MB Logical Volume with 100 MB.

```
[root@RHEL5 ~]# lvdisplay /dev/vg2/lvol0 | grepSize
  LVSize                 200.00MB
[root@RHEL5 ~]# lvextend -L +100 /dev/vg2/lvol0
  Extending logical volume lvol0 to 300.00 MB
  Logical volume lvol0 successfully resized
[root@RHEL5 ~]# lvdisplay /dev/vg2/lvol0 | grepSize
  LVSize                 300.00MB
```

The next example creates a 100MB Logical Volume, and then extends it to 500MB.

```
[root@RHEL5 ~]# lvcreate --size 100M -n extLV vg42
  Logical volume "extLV" created
[root@RHEL5 ~]# lvextend -L 500M vg42/extLV
  Extending logical volume extLV to 500.00 MB
  Logical volume extLV successfully resized
[root@RHEL5 ~]#
```

This example doubles the size of a Logical Volume.

```
[root@RHEL5 ~]# lvextend -l+100%LV vg42/extLV
  Extending logical volume extLV to 1000.00 MB
  Logical volume extLV successfully resized
[root@RHEL5 ~]#
```

lvrename

Renaming a Logical Volume is done with **lvrename**. This example renames extLV

to bigLV in the vg42 Volume Group.

```
[root@RHEL5 ~]# lvrename vg42/extLV vg42/bigLV
  Renamed "extLV" to "bigLV" in volume group "vg42"
[root@RHEL5 ~]#
```

OUTPUTS:

1. Create a volume group that contains a complete disk and a partition on anotherdisk.

Step 1: select disks:

```
root@rhel65:~# fdisk -l | grep Disk
Disk /dev/sda: 8589 MB, 8589934592 bytes Disk
identifier:0x00055ca0
Disk /dev/sdb: 1073 MB, 1073741824 bytes Disk
identifier:0x00000000
Disk /dev/sdc: 1073 MB, 1073741824 bytes Disk
identifier:0x00000000
```

I **choose /dev/sdb and /dev/sdc for now.**

Step 2: partition /dev/sdc

```
root@rhel65:~# fdisk /dev/sdc
Device contains neither a valid DOS partition table, nor Sun,
SGI or OSF disk\
label
Building a new DOS disklabel with disk identifier 0x94c0e5d5.
Changes will remain in memory only, until you decide to write
them.
After that, of course, the previous content won't be
recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be
corrected by w(rite)

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): n
Command action
   e   extended
   p   primary partition(1-4)
p
Partition number (1-4): 1
First cylinder (1-130, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-130, default 130):
Using default value130

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

Calling ioctl() to re-read partition table.
Syncing disks.
```

Step 3: pvcreate and vgcreate

```
root@rhel65:~# pvcreate /dev/sdb /dev/sdc1 Physical
  volume "/dev/sdb" successfully created Physical
  volume "/dev/sdc1" successfullycreated
root@rhel65:~# vgcreate VG42 /dev/sdb /dev/sdc1
  Volume group "VG42" succ
```
essfully created

1. Create two logical volumes (a small one and a bigger one) in this volumegroup. Format them wih ext3, mount them and copy some filesto them.

```
  root@rhel65:~# lvcreate --size 200m --name LVsmall VG42
Logical volume "LVsmall" created
root@rhel65:~# lvcreate --size 600m --name LVbig VG42
Logical volume "LVbig" created root@rhel65:~# ls -
l /dev/mapper/VG42-LVsmall
lrwxrwxrwx. 1 root root 7 Apr 20 20:41 /dev/mapper/VG42-LVsmall
  -> ../dm-2
  root@rhel65:~# ls -l /dev/VG42/LVsmall
lrwxrwxrwx. 1 root root 7 Apr 20 20:41 /dev/VG42/LVsmall ->
../dm-2
  root@rhel65:~# ls -l /dev/dm-2
brw-rw----. 1 root disk 253, 2 Apr 20 20:41 /dev/dm-2
```

```
root@rhel65:~# mkfs.ext3 /dev/mapper/VG42-LVsmall
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
51200 inodes, 204800 blocks
10240 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=67371008
25 block groups
8192 blocks per group, 8192 fragments per group
2048 inodes per group
Superblock backups stored on blocks:
   8193, 24577, 40961, 57345, 73729
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 39 mounts or
180 days, whichevercomesfirst.    Use tune2fs -c or -i to
override.
```

```
root@rhel65:~# mkfs.ext3 /dev/VG42/LVbig

mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
38400 inodes, 153600 blocks
7680 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=159383552
5 block groups
32768 blocks per group, 32768 fragments per group
7680 inodes per group
Superblock backups stored on blocks:
   32768, 98304
```

```
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 25 mounts or
180 days, whichevercomesfirst.    Use tune2fs -c or -i to
override.
```

The mounting and copying of files.

```
root@rhel65:~# mkdir /srv/LVsmall
root@rhel65:~# mkdir /srv/LVbig
root@rhel65:~# mount /dev/mapper/VG42-LVsmall /srv/LVsmall
root@rhel65:~# mount /dev/VG42/LVbig /srv/LVbig root@rhel65:~#
cp -r /etc /srv/LVsmall/
root@rhel65:~# cp -r /var/log /srv/LVbig/
```

Result: 1. I have studied about logical volume management.

2. Creating volume groups and logical volume groups successfully.