PROJECT REPORT ON

The Logical Volume Manager (LVM)

Submitted By:

Abhishek Kumar Thakur, UID-24MCA20326

Under The Guidance of:

Mrs. Geetanjali Sharma

April, 2024

University Institute of Computing

Chandigarh University,

Mohali, Punjab

CERTIFICATE

This is to certify that Abhishek Kumar Thakur (UID- 24MCA20326) have successfully completed the project title **“The Logical Volume Management (LVM)”** at University Institute of Computing under my supervision and guidance in the fulfilment of requirements of first semester, **Master of Computer Application.** Of Chandigarh University, Mohali, Punjab.

Dr. Abdullah	Mrs. Geetanjali Sharma
Head of the Department	Project Guide Supervisor
University Institute of Computing	University Institute of Computing

ACKNOWLEDGEMENT

We deem it a pleasure to acknowledge our sense of gratitude to our project guide Mrs. Geetanjali Sharma under whom we have carried out the project work. His incisive and objective guidance and timely advice encouraged us with constant flow of energy to continue the work.

We wish to reciprocate in full measure the kindness shown by Dr. Abdullah (H.O.D, University Institute of Computing) who inspired us with his valuable suggestions in successfully completing the project work.

We shall remain grateful to Dr. Manisha Malhotra, Additional Director, University Institute of Technology, for providing us a strong academic atmosphere by enforcing strict discipline to do the project work with utmost concentration and dedication.

Finally, we must say that no height is ever achieved without some sacrifices made at some end and it is here where we owe our special debt to our parents and our friends for showing their generous love and care throughout the entire period of time.

Date: 28.10.2024

Place: Chandigarh University, Mohali, Punjab

Abhishek Kumar Thakur, UID- 24MCA20326

ABSTRACT

Logical Volume Management (LVM) in Linux offers a flexible and efficient way to manage disk space by abstracting physical storage into logical volumes. Unlike traditional partitioning, LVM enables dynamic resizing of volumes, snapshot creation, and spanning of filesystems across multiple physical disks, providing a solution that adapts well to changing storage requirements.

The LVM system in Linux operates with three core concepts: Physical Volumes (PVs), Volume Groups (VGs), and Logical Volumes (LVs). Physical Volumes refer to the actual physical disks or partitions, which are grouped into Volume Groups. Volume Groups act as a pool of storage from which Logical Volumes are allocated. This approach not only facilitates efficient storage utilization but also simplifies disk management by enabling users to resize volumes without affecting data integrity.

This project report explores the structure, setup, and operational benefits of LVM. It provides a detailed analysis of the implementation steps involved in creating and managing LVM, including creating physical volumes, combining them into volume groups, and creating logical volumes. The report also covers advanced LVM functionalities like snapshots for data backup and restoration, resizing of volumes, and integrating LVM with RAID for added redundancy and performance.

Through various test scenarios and performance metrics, this study demonstrates the advantages of LVM in optimizing storage flexibility and performance in enterprise Linux environments. The findings suggest that LVM, due to its versatility and robustness, is a valuable tool for Linux administrators in handling complex storage requirements. This project underlines the importance of LVM as a dynamic storage management tool and its practical applications in various Linux environments.

TABLE OF CONTENTS

LOGICAL VOLUME MANAGEMENT − INTRODUCTION

Volume Management creates a layer of abstraction over the storage. Applications use a virtual storage, which is managed using a volume management software, a *Logical Volume Manager (LVM)*. This LVM hides the details about where data is stored, on which actual hardware and where on that hardware, from the entire system. Volume management lets you edit the storage configuration without actually changing anything on the hardware side, and vice versa. By hiding the hardware details it completely separates hardware− and software storage management, so that it is possible to change the hardware side without the software ever noticing, all during runtime.

With LVM system uptime can be enhanced significantly, because for changes in the storage configuration no application has to be stopped any more.

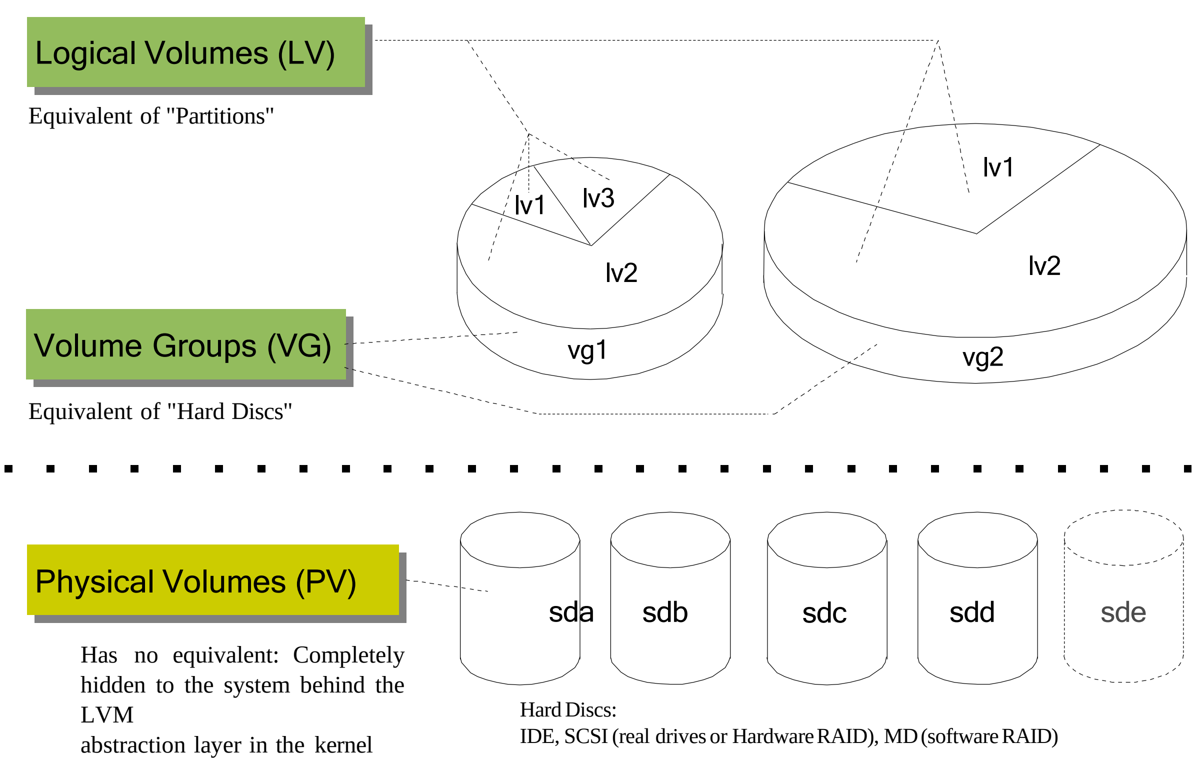
How it works

No LVM:
Without an LVM there are physical disks with partitions. On these partitions are filesystems or they are used *raw* (i.e. not managed by the kernel but with direct low−level access from the application), e.g. by databases like Oracle.

With LVM:
A *Volume Manager* puts the physical entities (*Physical Volumes*) into storage pools called *Volume Groups*. The LVM in SuSE Linux can manage whole SCSI− or IDE−disks and partitions in a Volume Group and also hardware− and even software RAID devices.

A Volume Group is the equivalent of a physical disk from the systems point of view. The equivalent of partitions into which this storage space is divided for creating different filesystems and raw partitions on it is called a *Logical Volume*.

What a *Logical Volume Manager* does:



HOW TO SETUP THE LVM

Command line tools, ASCII UI (YaST1), Graphical UI (YaST2)

Command line: As everything in Unix/Linux, the logical volume management is controlled using command line tools. For a list see the appendix B or the manual page for lvm(8) on your system.

YaST1: This interface has been there since SuSE Linux 7.0. To make it easier for those who do not want to remember the name of all the various commands and the sequence and arguments how to call them e.g. to create a configuration from scratch SuSE Linux provides a GUI interface to these tools. There is one in YaST (the SuSE setup− and configuration tool), and starting with SuSE Linux 7.2 there is also a module for YaST2 (successor of YaST).
To get to the LVM setup function of yast1. start YaST via the command *yast* or from the menu in KDE or Gnome, and go to *Installation settings −> Configure the Logical Volume Manager.*

YaST2: A module for LVM setup was introduced with SuSE Linux 7.2. To start the YaST2 module for the LVM configuration, if available on your system, call the YaST2 control center either by starting *yast2* from the command line (as *root*) or from the KDE− or Gnome menu. Note that YaST2 can also be started in an ASCII mode, without X. In the control center, look for and start the LVM module, in the graphical mode by clicking on the icon, in text mode by selecting it using *TAB* and then pressing *ENTER*.

How to start YaST1 or YaST2 from the KDE menu, here: SuSE Linux 7.1 default KDE screen

The three LVM setup steps

To start any new LVM configuration, we

- have to **define and initialize the Physical Volumes** that we plan to use, then
- **define Volume Groups** where we group the PVs together that we want to manage in one group, next
- we **set up Logical Volumes** on each Volume Group.

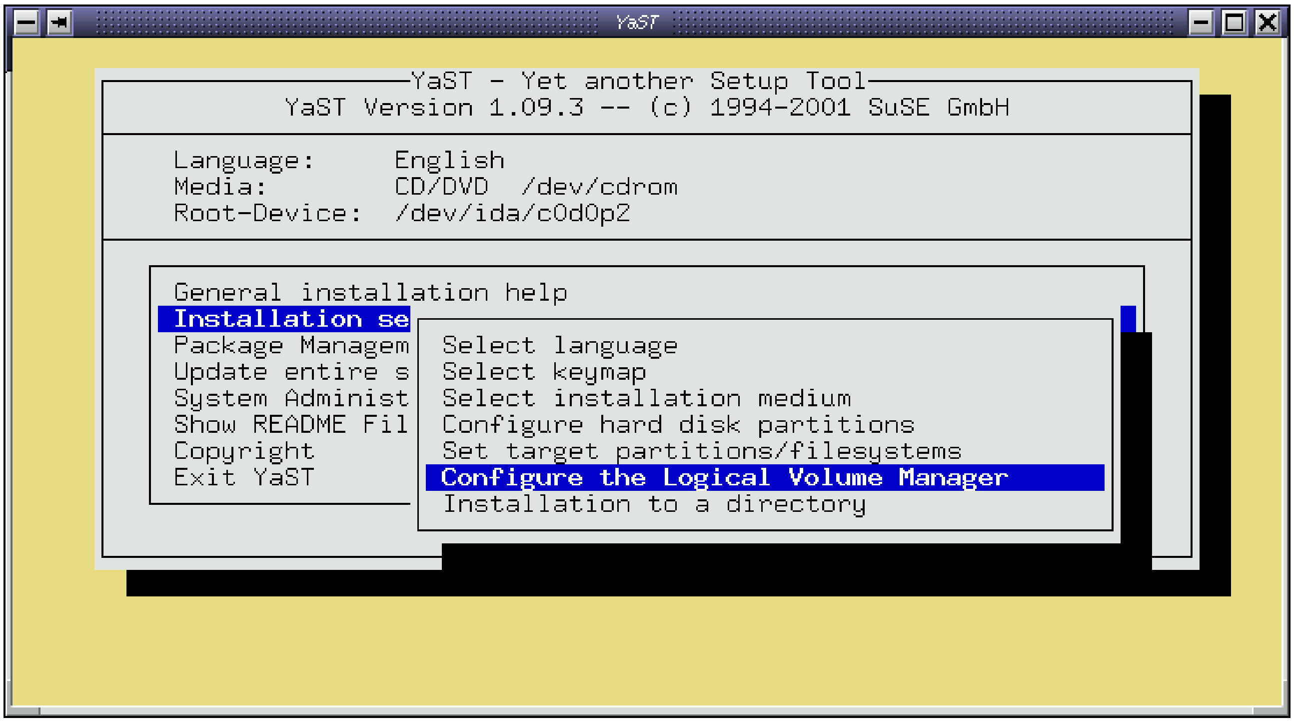
Using any interface, setting up any LVM, not just this one, always consists of these three steps. Step 1 (PVs) may be part of Step 2 (VGs) because selecting any disk or partition for inclusion in a VG automatically means they can be prepared as a PV for LVM and the more intelligent setup tools will do it for you. If you use the command line tools you will have to do it manually.

Setting up a PV: The device is initialized with an LVM ID and space is allocated and reserved for LVM data structures later storing information about the LVM setup on that PV (which VG is is part of, what LVs does it contain in what PEs, etc.).

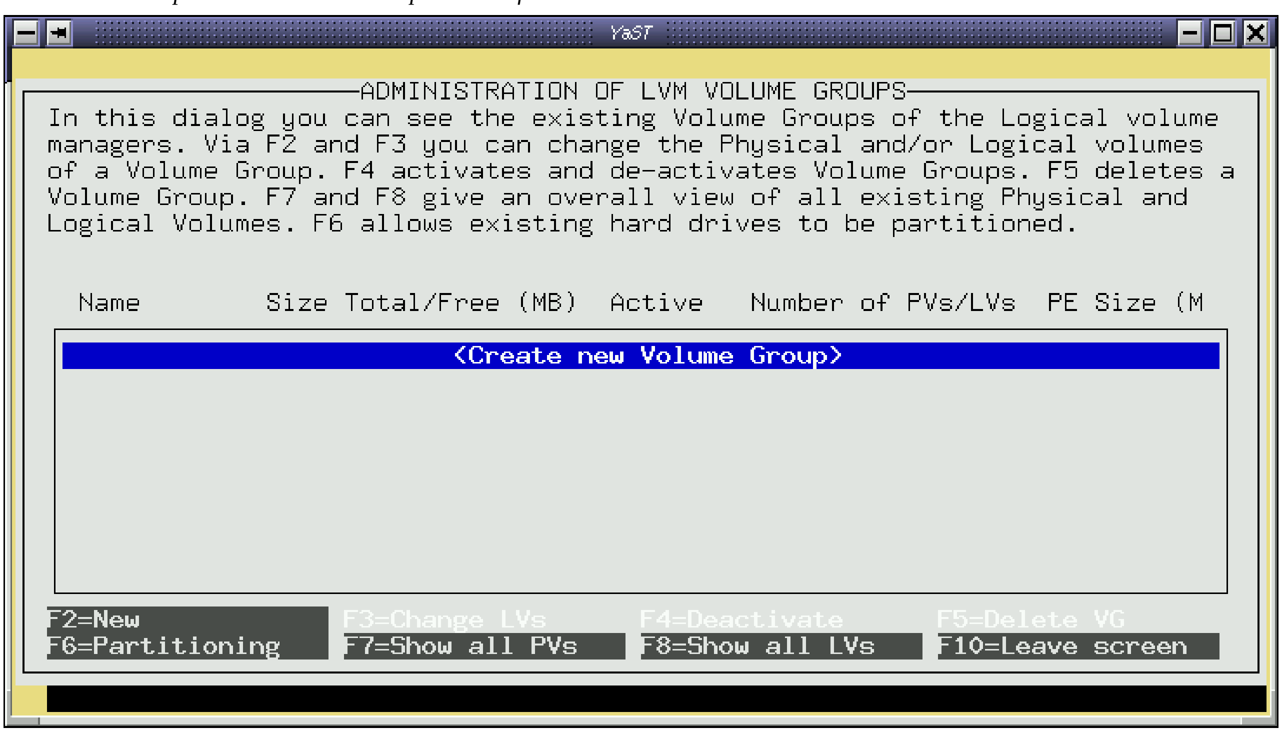
SETTING UP LVM USING YAST1

After starting YaST1 we go to the LVM configuration by selecting *Installation settings –> Configure the Logical Volume Manager*.

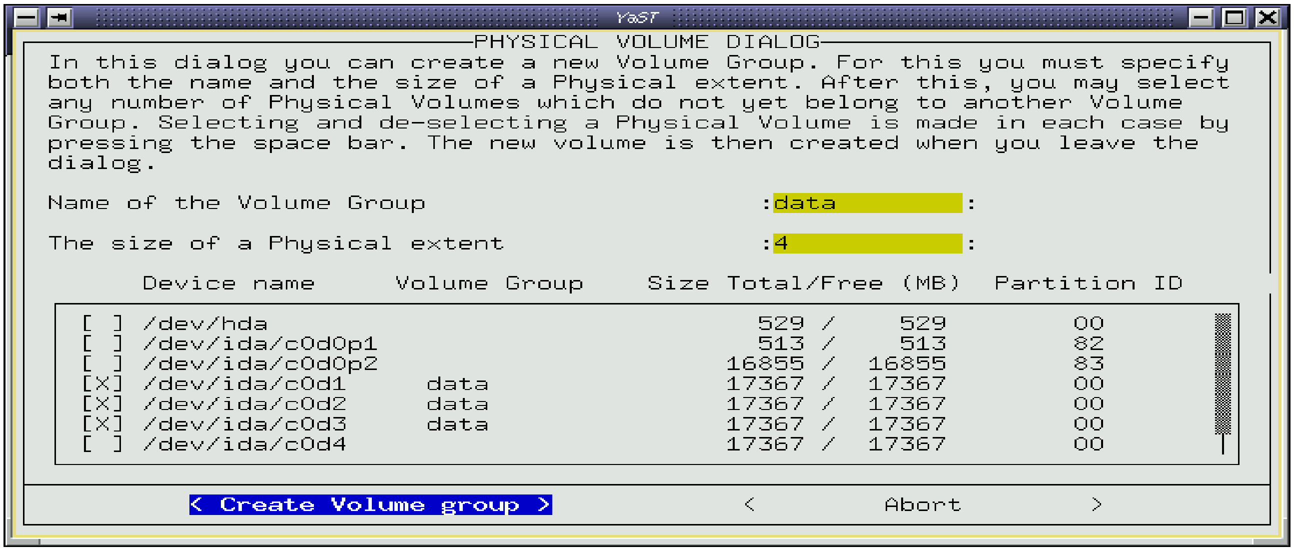
YaST 1 screen (showing the main menu with the Installation menu folded out)



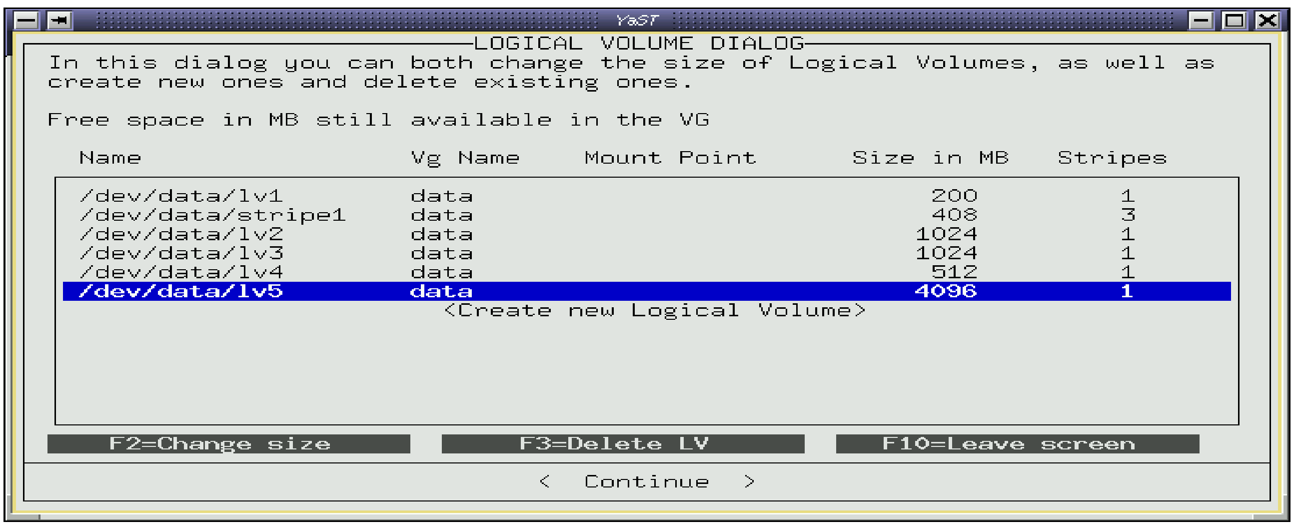
YaST 1 LVM setup screen – no LVM setup exists so far



Steps 1 and 2: Initializing PVs and creating VGs



Step 3: Setting up LVs



SETTING UP LVM USING THE COMMAND LINE TOOLS

This is very useful to automate the setup, e.g. when you have lots of LVs to be created. We again follow the three basic steps:

Step 1 (PVs):

```
pvcreate /dev/device [/dev/device] ...
```

The *pvcreate* command initializes (a) physical device(s) as (a) physical volume(s) (PV). Partitions must have the partition ID 8E (use fdisk, cfdisk or yast for partitioning), (whole) disks must have no partition table (none, not just an empty one). To see all physical devices that are available, use the command `lvmdiskscan`. The command `pvscan` shows all existing PVs.

If a disk (the whole disk, e.g. /dev/sdc) cannot be used and pvcreate complains about it having a partition table, after making sure you really want to use this disk for LVM and do not have anything else on it you can erase the partition table with `dd if=/dev/zero of=/dev/device-name bs=1k count=1` (followed by `blockdev --rereadpt <devicename>` to tell the kernel to re−read the p.table).

Step 2 (VGs):

```
vgcreate vg-name /dev/device [/dev/device] ...
```

The *vgcreate* command creates a volume group with all the physical volumes specified. The default extent size is 4 MB. The limit for the number of physical extents/logical extents on a PV/LV is 65534 due to kernel limits. Therefore, the default size of 4MB limits the max. size of any PV or LV to 4M*65534~255GB. Should you have larger PVs or require bigger LVs you can increase the extent size. A PE is also the smallest unit by which you can increase, decrease or address (e.g. to move PEs to other PVs) storage in LVM.

Step 3 (LVs):

```
lvcreate -L size[K|M|G|T] -n lv-name vg-name
```

The *lvcreate* command creates a logical volume on the volume group *vg−name*. If no name is given for the LV the default name lvol# is used, where # is the internal number of the LV. The LV can be created as striped (RAID0) over several or all available PVs using the *−i #* parameter, where # specifies the number of stripes. Normally LVs use up any space available on the PVs on a next−free basis, i.e. if you do the creation of LVs one by one you essentially get concatenation, until you start shrinking, growing, deleting and so on, when space gets freed and reallocated anywhere on the PVs.

This is all. From now on you can use the LVs by referring to them via */dev/vg−name/lv−name*. Use them like you would use any partition.

USING LVM

With a logical volume you can do anything you can do with a physical partition. You can create filesystems and mount them, you can create a raw device mapping for raw (unbuffered character device−) access to the LV, you can even format and use them as swap space.

Creating a Filesystem on an LV

It works just like with any physical device, just use the device name of an LV instead:

ext2: `mkfs.ext2 /dev/vg-name/lv-name`

reiserfs: `mkreiserfs /dev/vg-name/lv-name`

You can of course also create and use any other filesystems supported by Linux.

Using an LV as Swap Space

First format the LV you want to use as swap space and then enter the device name of the LV into /etc/fstab:

```
mkswap /dev/vg-name/lv-name
swapon -a
```

and/or an entry in /etc/fstab:

```
/dev/vg-name/lv-name     swap    swap    defaults  0  2
```

You may or may not be able to do so during the installation already, depending on if the installer supports it. If not, just do it manually afterwards. Note that we do not discuss how useful this option is. It *may* be useful when you use striping (RAID0) to increase throughput. Otherwise it probably does not have any advantage over using regular swap partitions. We merely mention it in case the question comes up.

Resizing of Volume Groups (VGs)

Using either of the YaST1 LVM setup dialog, the YaST2 LVM module or of course the LVM command line tools (*vgreduce, vgextend)*.

YaST1: See the YaST1 screenshot in "Step 1 & 2" of *Setting up LVM using YaST1*, it is the same dialog. To remove a PV from a VG highlight it and press *SPACE*. Do the same to add a physical device (it will be initialized as a new PV automatically).

Command line tools: There is one command for extending a VG by more PVs (*vgextend*) and another one to remove PVs (*vgreduce*).

Add two more disks to a VG:

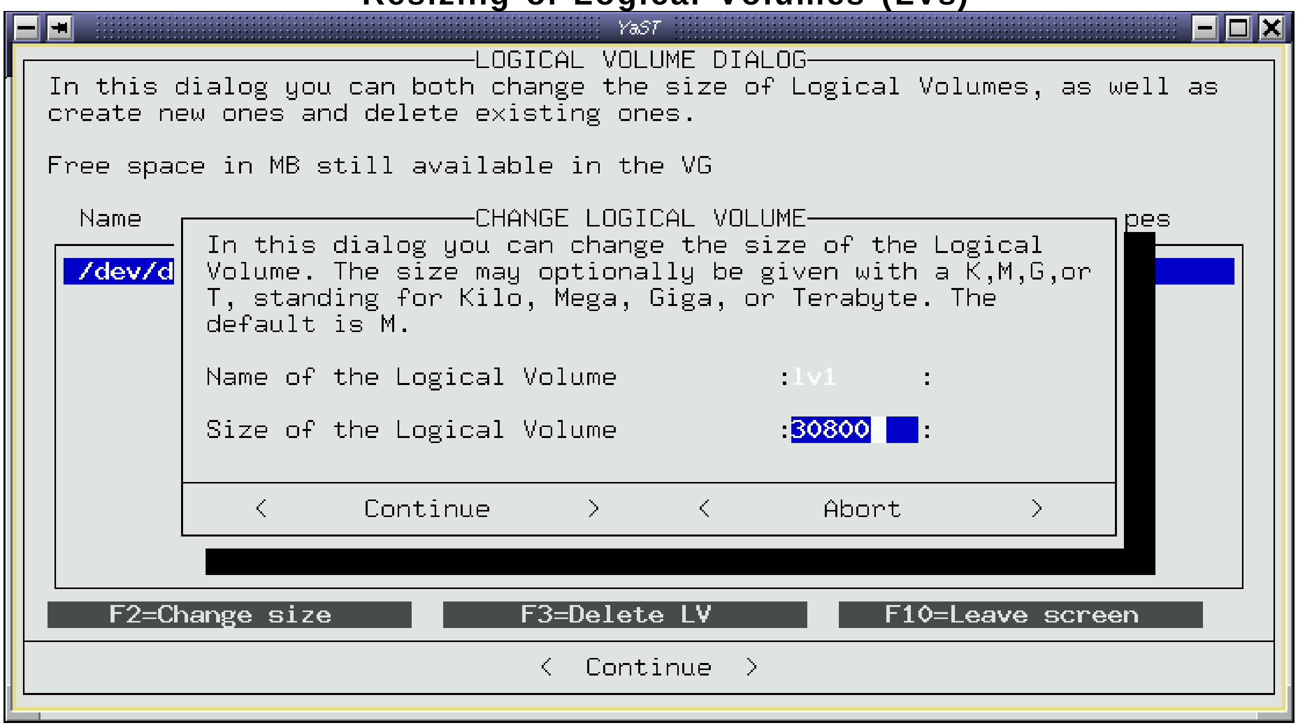
```
pvcreate /dev/device1 /dev/device2
vgextend vg-name /dev/device1 /dev/device2
```

Remove one PV:

```
vgreduce vg-name /dev/device
```

If you get an error message that the PV cannot be removed because it is still being used by an LV you will have to make sure there is no LV using it before you can proceed. If you have a striped LV one stripe may be on the PV you are trying to remove. See below *Moving Physical Extents (PEs)* for how to move LV storage to other PVs, or reduce the size of the LV(s) until the PV is not used any more.

Resizing of Logical Volumes (LVs)



Command line examples for resizing LVs:

Extend an LV with an ext2 filesystem on it by 1 GB:

```
umount /mountpoint-of-LV
e2fsadm -L+1G /dev/vg-name/lv-name
.... (output from the command) ....
mount /dev/vg-name/lv-name /mountpoint-of-LV
```

Extend an LV with a (**mounted and active**) ReiserFS on it by 512 MB:

```
lvextend -L+512M /dev/vg-name/lv-name
resize_reiserfs -s+512M -f /dev/vg-name/lv-name
```

Shrink an LV with a ReiserFS on it by 768 MB:

```
umount /mountpoint-of-LV
resize_reiserfs -s-768M /dev/vg-name/lv-name
lvreduce /dev/vg-name/lv-name
mount /dev/vg-name/lv-name /mountpoint-of-LV
```

Snapshot Logical Volumes

Snapshot logical volumes are possible since SuSE Linux 7.1 (LVM version 0.9). Snapshot devices provide **read−only** access to a logical volume image **frozen at the time of creation** of the snapshot, while the real LV continuous to be use and changed. This is useful for making consistent backups. Snapshot LVs have a relatively short lifetime, they are removed as soon as their purpose − e.g. a consistent backup of the LV − is fulfilled.

The size of the snapshot LV must be large enough to hold all changes made to the original LV during the life−time of the snapshot. The max. size is 1.1x the size of the original volume. If a snapshot LV becomes full it will become unusable!

Step 1: Create a snapshot LV:

```
lvcreate -L 512M -s -n snap-lv-name /dev/vg-name/lv-name
```

Step 2: Do the backup using the snapshot LV

Step 3: Delete a snapshot LV:

```
lvremove /dev/vg-name/snap-lv-name
```

You can create as many snapshot devices of an LV as you like.