Chapter 8

Creating and Administering *lv* Logical Volumes

This chapter describes the procedures for creating and administering *lv* logical volumes. Support for *lv* logical volumes will be removed from IRIX following IRIX Release 6.2. The procedure for converting from *lv* logical volumes to XLV logical volumes is described in the section "Converting lv Logical Volumes to XLV Logical Volumes" in Chapter 7.

The major sections in this chapter are:

- "Creating Entries in the /etc/lvtab File" on page 168
- "Creating New Logical Volume With mklv" on page 169
- "Checking Logical Volumes With lvck" on page 170
- "Creating a Logical Volume and a Filesystem on Newly Added Disks" on page 171
- "Increasing the Size of a Logical Volume" on page 173
- "Shrinking a Logical Volume" on page 174

Creating Entries in the */etc/lvtab* File

The file */etc/lvtab* contains a table of logical volumes. It is read by the commands that create, install, and check the consistency of logical volumes. You can modify it with a text editor to add new logical volumes or to change existing ones.

The entries in */etc/lvtab* have this form (it is shown wrapped here, but it is a single line with no blank spaces):

volume_device_name`:[`*volume_name*`][:stripes=`*stripe_number*`[:step=`*stripe_unit*`]]:devs=`*device_pathnames*

The *volume_device_name* is of the form `lv`*n*, where *n* is a one or two digit integer. The logical volume is accessed through the device special files */dev/dsk/lv<n>* and */dev/rdsk/lv<n>*.

The *volume_name* is an arbitrary identifying name for the logical volume. This name is included in the logical volume labels on each of the partitions making up the logical volume. It is then used by commands to verify that the logical volume on the disks is actually the volume expected by */etc/lvtab*. Any name of up to 80 characters can be used; you should probably choose something that other users can identify. You can leave this field blank, but this is not recommended.

The **stripes** option creates a logical volume that is striped across its constituent devices (see Figure 6-1 for an illustration of how the data is written to the devices). The number of *device_pathnames* must be a multiple of *stripe_number*. *stripe_number* specifies the number of disks the volume is striped across. For example, suppose you have a logical volume with 6 constituent devices and a *stripe_number* of 3. The logical volume is set up to stripe across the first three devices until they are filled, then to stripe across the second three.

The **step** option further specifies the stripe unit (the granularity with which the storage is distributed across the components of a striped volume). *stripe_unit* is measured in disk blocks. The default stripe unit is the device track size, which is generally a good value to use.

The *device_pathnames* are listed following any options. They are the block device filenames of the devices constituting the logical volume. *device_pathnames* must be separated by commas. The partitions named must be legal for use as normal data storage, and not dedicated partitions, such as *swap*.

Creating New Logical Volume With *mklv*

The *mklv* command creates logical volumes by writing logical volume labels for the devices that will make up the volume. The basic *mklv* command is:

```
# mklv volume_device_name
```

mklv reads the entry in */etc/lvtab* identified by *volume_device_name* (*volume_device_name* is the first field in the */etc/lvtab* file) and creates the logical volume described. It labels the devices appropriately by writing logical volume labels in the volume headers of the disks used in the logical volume, then initializes the logical volume device for use. It is not necessary to run the *lvinit* command after running *mklv.*

Normally, *mklv* checks all the named devices to see if they are already part of a logical volume or contain a filesystem. The option **-f** forces *mklv* to skip those checks.

Various errors can arise when trying to create a logical volume. For example, one of the specified disks might be missing, the new *lvtab* entry might have a typographical error, or the partitions of a striped volume might not be exactly the same size.

If the partitions aren't exactly the same size, for example because the default partitioning for drives of similar sizes but from different manufacturers is slightly different, you will see an error message similar to this from *mklv*:

```
lv0:proj:stripes=2:devs= \
        /dev/dsk/dks0d2s7,  \
        /dev/dsk/dks1d0s7 <INCORRECT PARTITION SIZE>
```

In this case, you need to adjust the partition sizes; see the section "Repartitioning a Disk With fx" in Chapter 2 for instructions.

The mklv(1M) reference page describes the possible error messages and their meanings.

Checking Logical Volumes With *lvck*

As described in the section "lv Logical Volumes" in Chapter 6, the *lvck* command is used to check the consistency of logical volumes.

To check every logical volume for which there is an entry in */etc/lvtab*, give this command:

```
# lvck
```

To check only one entry in */etc/lvtab*, give the name of a logical volume device (the first field in the */etc/lvtab* entry), for example:

```
# lvck lv0
```

The **-d** option of *lvck* facilitates re-creation of an *lvtab* for the system, if necessary. You might use this option if, for example, */etc/lvtab* became corrupted or if you somehow lost track of which disks were connected during a system reconfiguration. With the **-d** flag, *lvck* ignores */etc/lvtab* and searches through all disks connected to the system in order to locate all logical volumes present. It prints a description of each logical volume found in a form resembling an *lvtab* entry. For example:

```
# lvck -d
# The following logical volumes are present and correct:

# Volume id:   IRIX: Mon Jun  5 16:58:15 1995

lv?:Zebra Project:devs=/dev/dsk/dks0d2s0,       \
/dev/dsk/dks0d3s0
...
```

lvck can print out any logical volume label that exists for a block device file. The output resembles an *lvtab* entry. This mode of *lvck* is purely informational; no checks are made of any other devices mentioned in the logical volume label. To print the label, invoke *lvck* with the block device file, for example:

```
# lvck /dev/dsk/dks0d2s7

# Volume id:   IRIX: Thu Oct  5 16:29:33 1995

lv?:Project Data:devs=/dev/dsk/dks0d2s7,        \
        /dev/dsk/dks0d3s7
```

Possible errors and the messages from *lvck* are described in the lvck(1M) reference page.

Creating a Logical Volume and a Filesystem on Newly Added Disks

Suppose that new disks are added to your system in order to provide additional storage. Instead of simply creating a filesystem on each disk, you could create a logical volume consisting of these new disks and make a filesystem on the logical volume. (To extend an existing filesystem onto a logical volume created out of an existing disk and the new disks, see the sections "Growing an EFS Filesystem Onto Another Disk" and "Growing an XFS Filesystem Onto Another Disk" in Chapter 4.)

Caution: All files on the new disks are destroyed by this procedure. If the new disks contain files that you want to save, back up all files to tape or another disk before beginning this procedure.

Follow this procedure to create a logical volume and a filesystem on new disks that have been initialized and partitioned:

1. Decide which partitions of these new disks you want to use for the new filesystem. (Normally, when adding a new filesystem like this, you want to use the entire disks, that is, partition 7 of each disk.)
2. Decide if you want to make a striped volume. (See the section "To Stripe or Not to Stripe?" in Chapter 6 for information about the benefits and restrictions of striped volumes.)
3. Add an entry to */etc/lvtab* containing the device special pathnames of the new disks that are to be part of the new volume. (See the section "Creating Entries in the /etc/lvtab File" in this chapter for details of the syntax of *lvtab* entries.) For example:

```
lv0:Zebra Project:stripes=2:devs=/dev/dsk/dks0d2s7,/dev/dsk/dks1d1s7
```

In this example, the logical volume named Zebra Project consists of two partitions from two separate disks on different controllers. Storage is striped across the two disks. (Note that it is not normally necessary to specify the **step** parameter. The default stripe unit value is used automatically.)

4. Give the command *lvck* to check other logical volumes, if any, and the syntax of the new entry in */etc/lvtab*:

   ```
   # lvck
   ```

   The <NO LABEL PRESENT> message you see is normal for the devices in the new logical volume.

5. Give the command *mklv* to place the logical volume labels on the new disks to identify them as parts of a logical volume:

   ```
   # mklv lv0
   ```

   The device names */dev/dsk/lv0* and */dev/rdsk/lv0* are created. They can now be accessed exactly like any regular disk partition.

6. To create the new filesystem, run *mkfs* on */dev/rdsk/lv0* as you would run it on a regular disk, for example:

   ```
   # mkfs /dev/rdsk/lv0
   ```

7. You can mount */dev/dsk/lv0* exactly as you would mount a filesystem on a regular disk:

   ```
   # mkdir /zebra
   # mount /dev/dsk/lv0 /zebra
   ```

8. You may want to add an entry to */etc/fstab* to mount this filesystem automatically, for example:

   ```
   /dev/dsk/lv0 /zebra efs rw,raw=/dev/rdsk/lv0 0 0
   ```

9. Give the command *lvinfo* to verify that the new logical volume is active:

   ```
   # lvinfo lv0
   ```

Increasing the Size of a Logical Volume

Caution: The procedure in this section can result in the loss of data if it is not performed properly. It is recommended only for experienced IRIX system administrators.

An existing logical volume can be extended to include one or more additional disk partitions. Remember that if the original logical volume is striped, you must add a number of disk partitions that is a multiple of the **stripes** parameter.

Caution: All files on the disk partition added to the logical volume are destroyed. If the disk partition contains files that you want to save, back up all files on the partition to tape or another disk before beginning this procedure.

Follow this procedure to increase the size of a logical volume, for example lv0:

1. Unmount the logical volume you plan to extend:

   ```
   # umount /dev/dsk/lv0
   ```

2. Add the block device files for the new disk partition(s) to the end of the */etc/lvtab* entry for the logical volume. For example, say the original entry is:

```
lv0:Zebra Project:stripes=2:devs=/dev/dsk/dks0d2s7,/dev/dsk/dks1d1s7
```

   Add more *device_pathnames* for the new disk partitions to the end of the entry (although the line shown wrapped here, it is one line in the file):

```
lv0:Zebra Project:stripes=2:devs=/dev/dsk/dks0d2s7,/dev/dsk/dks1d1s7,
/dev/dsk/dks0d3s7, /dev/dsk/dks1d2s7
```

3. Give the command *lvck* to check other logical volumes, if any, and the syntax of the changed entry in */etc/lvtab*:

   ```
   # lvck
   ```

   The `<NO LABEL PRESENT>` message you see is normal for the devices in the new logical volume.

4. Give the command *mklv* with the **-f** option to update the logical volume labels:

   ```
   # mklv -f lv0
   ```

5. If there is a filesystem on the logical volume, extend it with the *growfs* command:

   ```
   # growfs /dev/dsk/lv0
   ```

6. Remount the logical volume:

   ```
   # mount /dev/dsk/lv0
   ```

Shrinking a Logical Volume

Caution: The procedure in this section can result in the loss of data if it is not performed properly. It is recommended only for experienced IRIX system administrators.

Follow this procedure to remove one or more partitions from a logical volume:

1. Back up all files on the logical volume to tape or another filesystem. The entire logical volume is erased during this procedure.
2. Make a list that contains the controller, drive address, and partition number of each partition in the logical volume by looking at the */etc/lvtab* entry for the volume.
3. For each controller and drive address pair in your list, follow the procedure in the section "Removing Files in the Volume Header With dvhtool" in Chapter 2 to remove one or more logical volume labels. These logical volume labels (files) are called *lvlab*<*n*>, where <*n*> is a partition number of a partition on this disk that is included in the logical volume you are removing.
4. Modify the */etc/lvtab* entry for this logical volume so that it includes only the disk partitions that you want to include in the "shrunk" logical volume.
5. Give the *mklv* command to re-create the logical volume labels for the shrunk logical volume:

   ```
   # mklv volume_device_name
   ```

   volume_device_name is the first entry in the */etc/lvtab* entry for this logical volume.
6. Make a filesystem on the logical volume by using the instructions in either the section "Making an XFS Filesystem" or the section "Making an EFS Filesystem" in Chapter 4.
7. Restore the files from the original logical volume to the shrunk logical volume. Be aware that the files from the original logical volume may no longer fit on the shrunken logical volume.