Tape Devices/Drives

Controlling Tape Drives

- The `mt` command is the general-purpose command that manipulates tapes. It is used to assist the backup process.

- Some of the options for `mt`
  - `rewind`: Rewinds a tape
  - `offline` : Prepares the currently loaded tape for ejection and, if possible, ejects it
  - `fsf`: Moves the currently loaded tape to the specified position
  - `erase`: Erases the currently loaded tape

Controlling Tape Drives cont'd

- The `mt` command syntax is:

  `mt -f device command`

- To specify the device, use the `-f` option followed by the desired target
  - The standard SCSI tape devices are named `st0`, `st1`, etc., and `nst0`, `nst1`, etc
  - The standard IDE tape devices are named `ht0`, `ht1`, etc., and `nrht0`, `nrht1`, etc

Controlling Tape Drives cont'd

- Tape drive normally rewinds the media after the tape operation has completed. If you don't want the tape to rewind, you can access the device by its no rewind name.

- Examples
  - # **`mt -f /dev/st0 rewind`** // Rewinds the tape.
  - # **`mt -f /dev/nst0 fsf 50`** // Positions the tape (don't forget to use "n").
  - # **`mt -f /dev/st0 offline`** // Ejects the tape (but doesn't rewind it first).
  - # **`mt -f /dev/st0 erase`** // Erases the tape.
  - # **`mt -f /dev/st0 rewoff`** // Rewinds and ejects the tape.

Using `tar`/`star` Commands

- Archive to tapes or other media or files
- `star` command backups `SELinux` contexts and `ACL` attributes.
- Options
  - `c:` To create new archive
  - `t:` To list the content of existing archive
  - `x:` To extract existing archive
  - `v:` verbose
  - `z:` gzip compress
  - `j:` bzip2 compress

Examples

```
# tar cf /dev/st0 fname
# tar zcf /dev/st0 fname
# tar zxf /dev/st0
```

* Where does `tar` extract the files ???

Incremental and Full Back Ups

- A full backup is a complete file system backup.
- An incremental backup copies only files in the file system that have been added or modified since a previous lower-level backup.
- Backup increment = dump level

Level	Definition
0	Full backup
1-9	The backup copies new or modified files since the last lower-level backup

Using dump Command

- It can backs up filesystems. By providing the mount point of the filesystem to back up.

- Example

```
# dump -0u -f /dev/nst1 /home
# dump -4u -f /dev/nst1 /home
```

Recovering dump Data

- To recover an entire filesystem

```
# restore -rf /dev/st0
```

- To restore individual files and directories

```
# restore -xf /dev/st0 file1 file2
# restore -if /dev/st0
```

Advanced File System Management

Software RAID Configuration

- Types of RAID
  - RAID 0
  - RAID 1
  - RAID 5

- Steps to set up RAID
  - Use the `fdisk` command to create RAID partitions of type RAID (`fd`)
  - Use `mdadm` command to create RAID device, command options
    - `v`: for verbose mode
    - `C`: create device specified
    - `--level`: the device RAID level
    - `--raid-devices`: the number of devices in the RAID
  - Create new file system using `mke2fs` command which has special options for the RAID
    - `-R stride=n`
  - Mount the RIAD device and add it to `fstab` file

Example

```
# fdisk /dev/hda
# mdadm –v –C /dev/md0 --level 5 --raid-
  devices=3 /dev/hda9 /dev/hda10 /dev/hda11
# mke2fs -j -b 4096 -R stride=16 /dev/md0
# mkdir /data
# mount /dev/md0 /data
# mount | grep /dev/md0
/dev/md0  on /data  type ext3 (rw)
```

- To view information about RAID status

```
# cat /proc/mdstat
# mdadm --detail /dev/md0
```

Flexible Filesystems with LVM

- LVM is used to create virtual partitions called `logical volumes` from the space that is available on one or more hard drives, disk partitions, or RAID devices.
- For LVM to use these devices, they need to be initialized as `physical volumes` and assigned to a "container" called a `volume group`.
- You can assign multiple `physical volumes` to the same `volume group` and use a `volume group` to create multiple `logical volumes`

Flexible Filesystems with LVM

- Each `volume group` divides its disk space pool into `extents` of identical size.
- An extent's size is set for a particular volume group when that volume group is created.
- An extent is typically between 1 MB and 64 MB in size.
- A single logical volume may contain at most 65,534 extents, so larger extent sizes allow larger logical volumes.

Creating Volume Groups and Logical Volumes

- Create LVM configuration files
- Create physical volumes
- Create volume groups which function equivalent to a disk
- Create logical volumes which is equivalent to a partition
- Create file system

Example

```
# vgscan                    // store information in /etc/lvm
# pvcreate /dev/hda9
  Physical volume “/dev/hda9 successfully created
# pvcreate /dev/hda10
  Physical volume “/dev/hda10 successfully created
# pvcreate /dev/hda11
  Physical volume “/dev/hda11 successfully created
# pvcreate /dev/hda12
  Physical volume “/dev/hda12 successfully created
```

Example cont'd

```
# vgcreate vg0 /dev/hda9 /dev/hda10 //A directory under /dev directory is created called vg0
  Volume group "vg0" successfully created
# vgcreate vg1 /dev/hda11 /dev/hda12
  Volume group "vg1" successfully created

# lvcreate –L 100M –n lv1 vg0 // A device file called /dev/vg0/lv1 is created
  Logical volume "lv1" created
# mkfs –t ext3 /dev/vg0/lv1
# mount /dev/vg0/lv1    /data
```

Example cont'd

```
# df –h
Filesystem     Size     Used  Avail   Use%    Mounted on
/dev/mapper/vg0-lv1    97M 5.6M    87M 7%  /data
# lvextend –L +40M /dev/vg0/lv1
Extending logical volume lv1 to 140.00 MB
Logical volume lv1 successfully resized
# ext2online /data
# df –h /data
Filesystem       Size Used Avail    Use% Mounted on
/dev/mapper/vg0-lv1   135M 5.6M 127.4M   7%  /data
```

Other LVM Commands

- To add `physical volumes` to a `volume group` : `vgextend`
- To Move data in a `physical volume` to other `physical volumes` in the `volume group` : `pvmove`
  - needed before removing a `physical volume` from the `volume group`
- To remove a `physical volume` from a `volume group` : `vgreduce`
- To display information about a `physical volume` : `pvdisplay`
- To display information about a `volume group` : `vgdisplay`
- To display information about a `logical volume` : `lvdisplay`

The Linux Quota System

- Features of the Linux quota system
  - Implemented within the kernel.
  - Enabled on a per-filesystem basis.
  - Individual policies for groups and users
  - Enables limit setting by number of blocks or inodes.
  - Both soft and hard limits can be implemented.
    - Soft limit can be exceeded for a limited period of time. When the grace period is over, the soft limit is converted into a hard limit
    - Hard limit can’t be exceeded

The Linux Quota System cont'd

- Steps to set up quotas
  - Edit `/etc/fstab` and add the `usrquota` and/or `grpquota` option to the filesystem.
  - Remount the filesystem with the command `mount -o remount <filesystem>`.
  - Use the `quotacheck` command to create the quota-tracking files.
  - Use the `quotaon` command to enable quota tracking by the kernel.
  - Use the `edquota` or `setquota` commands to specify the quotas for each user and/or group.
  - Use the `repquota` command to verify the settings and the current usage.

Example

```
# vi /etc/fstab
LABEL=/home  /home  ext3  defaults,usrquota,grpquota  1  2
# mount –o remount LABEL=/home
# mount | grep /home
/dev/hda7  on  /home  type  ext3  (rw,usrquota,grpquota)
# quotacheck –vmac
# ls /home
aquota.group  aquota.user ...
# quotaon /home
# setquota –u guest  50000  60000  1500  1700  /home
```

- Or use

```
# edquota –u guest
Filesystem blocks  soft   hard   inodes  soft   hard
/dev/hda7    16    50000  60000    6     1500   1700
```

Example cont'd

- Another user account quota setting can be used

```
# edquota –a guest guest1
[fred@host1 ~] $ quota
Disk quotas for user guest (uid 501)
Filesystem  blocks quota limit grace files quota   limit grace
/dev/hda7    16    50000 60000        6    1500 1700
# repqouta /home
***Report for user quotas on device /dev/hda7
Block grace time:7days Inode grace time:7 days
        Block Limit    File Limit
USER    used    soft    hard    grace    used soft hard       grace
root    14491    0   0         11     0     0
guest1   16   50000 60000        8   1500    1700
guest    20      50000 60000        8   1500    1700
```

Thanks ☺