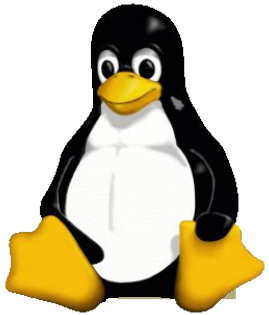


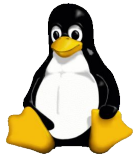
Linux Storage Administration

Maruthi Inukonda

16th Feb 2019

Agenda

- Storage devices
- Disk partitioning
- RAID
- Logical volume management
- File systems



Storage Devices

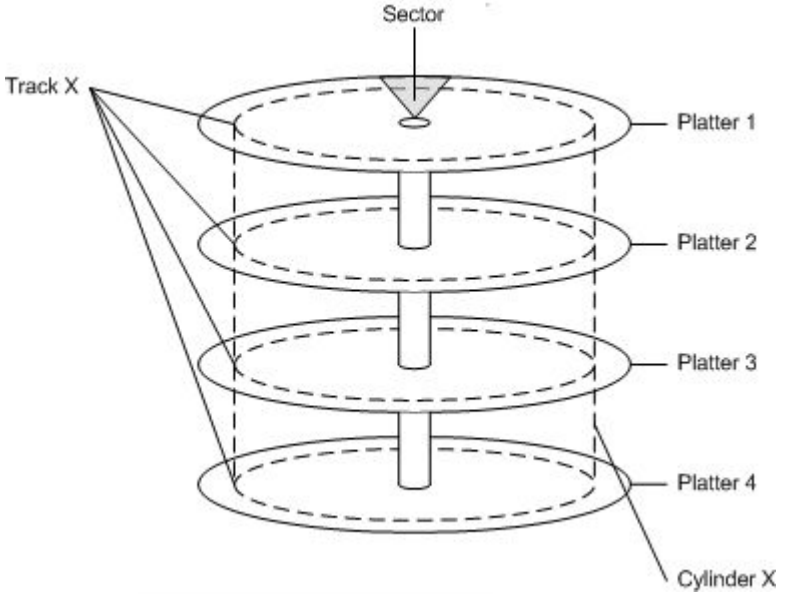
Classification of Storage Devices

- By Media
  - Magnetic, Optical, Flash
- By Access
  - Rotational, Sequential, Direct
- By purpose
  - Secondary, Tertiary

```
$ lsscsi
[0:0:1:0]   cd/dvd   TSSTcorp CDW/DVD SH-M522C TS04  /dev/sr0
[2:0:0:0]   disk      ATA       ST3500418AS       CC38  /dev/sda
[3:0:0:0]   disk      ATA       SEAGATE ST330006 NS00  /dev/sdb
[4:0:0:0]   disk      ATA       HITACHI HUA72202 N100  /dev/sdc
[5:0:0:0]   mediumx   EMC       DDVTL             0306  /dev/sch0
[5:0:0:1]   tape      IBM       ULTRIUM-TD4       8711  /dev/st0
[5:0:0:2]   tape      IBM       ULTRIUM-TD4       8711  /dev/st1
[5:0:0:3]   tape      IBM       ULTRIUM-TD4       8711  /dev/st2
[5:0:0:4]   tape      IBM       ULTRIUM-TD4       8711  /dev/st3
```

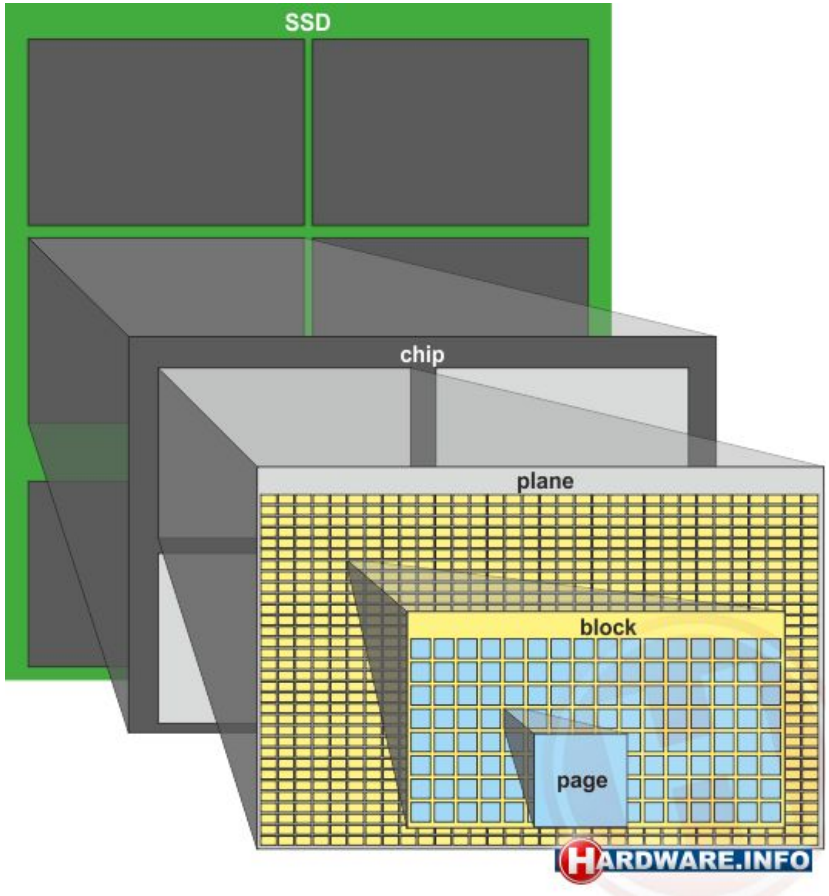
Magnetic Disk Structure

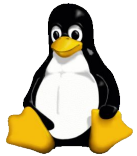
- Rotating disks have
  - Platters, Magnetic surfaces
  - Read/Write heads
  - Tracks, Sectors
  - Cylinders



Solid State Disk Structure

- Solid state disks have
  - Pages/Sectors (4KiB)
  - Erase Blocks (256KiB/512KiB)
  - Planes
  - Chips





Disk Device

- Secondary storage device
- Accessible in units of sectors (512B, 1KiB or 4KiB)
- List all storage devices using `lsscsi`.

```
$ lsscsi
[0:0:1:0]  cd/dvd  TSSTcorp CDW/DVD SH-M522C TS04  /dev/sr0
[2:0:0:0]  disk    ATA      ST3500418AS      CC38  /dev/sda
[3:0:0:0]  disk    ATA      SEAGATE ST330006 NS00  /dev/sdb
[5:0:0:0]  disk    ATA      HITACHI HUA72202 N100  /dev/sdc
```

Disk Partitioning

Block devices (Recap)

- OS creates a logical block device layer on disk, its partitions
- Accessible in units of 512B, 1KiB, 4KiB.
- Main purpose is caching and ordering I/O.
- List all block devices using `lsblk -p`

```
$ lsblk -p
NAME         MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
/dev/sda       8:0    0 465.8G  0 disk
├─/dev/sda1    8:1    0   512M  0 part
├─/dev/sda2    8:2    0    14G  0 part
├─/dev/sda3    8:3    0    15G  0 part /
├─/dev/sda4    8:4    0     1K  0 part
├─/dev/sda5    8:5    0   200G  0 part /home
├─/dev/sda6    8:6    0    16G  0 part [SWAP]
└─/dev/sda7    8:7    0 220.3G  0 part
/dev/sr0      11:0    1  1024M  0 rom
```

NOTE: device names sda, sdb are volatile. May change across reboots.

Disk Partitioning

- Partition is a contiguous region on a hard disk for
  - file system
  - swap space
  - logical volumes

- OS manages information in each partition separately.

- Each partition of a disk is a separate file-system failure domain.
- Where as all partitions of a disk fall in single hardware failure domain.

- Two types of disk partitioning schemes
  - DOS Partition Table (aka Master Boot Record /MBR)
  - GUID Partition Table (GPT)

Best practice: During OS installation keep /, /home, swap as separate partitions

DOS Partition Table (1/2)

- Mostly used format for DOS PCs invented in 1987
- Does not support more than 15 partitions.
  - 4 primary
  - 3 primary + 12 logical
- Supports sector size of 512B only
- Does not support larger than 2TiB disks
- Prone to security issues (rootkit)
- This works with traditional motherboard firmware (BIOS)

DOS Partition Table (2/2)

- List all disks' partitions using `parted -l` or `fdisk -l`
- A sample DOS partition table

```
# parted /dev/sda print
Model: ATA ST3500418AS (scsi)
Disk /dev/sda: 500GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type      File system     Flags
 1      1049kB  538MB   537MB   primary   ext4            boot
 2      538MB   15.6GB  15.0GB  primary   ext4
 3      15.6GB  31.7GB  16.1GB  primary   xfs
 4      31.7GB  500GB   468GB   extended
 5      31.7GB  246GB   215GB   logical   xfs
 6      246GB   264GB   17.2GB  logical   linux-swap(v1)
 7      264GB   500GB   237GB   logical   ext4
```

Creating DOS Partition Table (fdisk)

- Create partition table header using
  - command of `fdisk`
- Reload the partition table using `partprobe`

```
# fdisk /dev/sdb
Welcome to fdisk (util-linux 2.27.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x5b4ad344.

Command (m for help): o
Created a new DOS disklabel with disk identifier 0xb3e8da19.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

* parted command could also be used

Creating DOS Partition (fdisk)

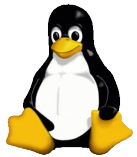
- Create partition using
  `n` command of `fdisk` (parted command could also be used)
- Reload the partition table using `partprobe`

```
# fdisk /dev/sdc
Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-2147483647, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-2147483647, default 2147483647): +10GiB

Created a new partition 1 of type 'Linux' and of size 10 GiB.

Command (m for help): p
Disk /dev/sdb: 1 TiB, 1099511627776 bytes, 2147483648 sectors
...
Device     Boot Start      End  Sectors Size Id Type
/dev/sdb1       2048 20973567 20971520  10G 83 Linux
```

Deleting DOS Partition (fdisk)

- Delete a partition using
  `d` command of `fdisk` (parted command could also be used)
- Reload the partition table using `partprobe`

```
# fdisk /dev/sdc
Command (m for help): d
Partition number (1,2, default 2): 1

Partition 1 has been deleted.
```



GUID Partition Table (GPT) (1/2)

- New format invented in 2005
- Support upto 4 million partitions.
- Supports upto 9.4 ZiB disks
- Supports sector sizes of 512B, 1KiB, 4KiB
- Addresses security issues (rootkit) using Secure boot feature
  - Ubuntu, RHEL support
  - Windows 8.x

  This requires newer motherboard firmware (UEFI)

GUID Partition Table (GPT) (2/2)

- List all disks' partitions using `parted -l`
- A sample GPT disk

```
# parted /dev/sdb print
Model: ATA SEAGATE ST330006 (scsi)
Disk /dev/sdb: 3001GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system     Name          Flags
 1      1049kB  3146kB  2097kB                                bios_grub
 2      3146kB  21.0GB  21.0GB  ext4            fc18          msftdata
 3      21.0GB  41.9GB  21.0GB  ext3            centos        boot, esp
 4      41.9GB  62.9GB  21.0GB  ext4            fc18-gold     msftdata
 5      62.9GB  83.9GB  21.0GB  ext3            centos-gold   msftdata
 6      83.9GB  189GB   105GB   ext3            home          msftdata
 7      189GB   222GB   33.3GB  linux-swap(v1)  swap
 9      222GB   2222GB  2000GB  xfs             MyDrive       msftdata
10      2222GB  3001GB  779GB   xfs             MySpare       msftdata
```

Creating GUID Partition Table (parted)

- Create partition table header using `mklabel` command of `parted`
- Reload the partition table using `partprobe`

```
# parted /dev/sdc
(parted) mklabel msdos

(parted) print
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdc: 2199GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start  End  Size  File system  Name  Flags
```

* fdisk command could also be used

Creating GUID Partition (parted)

- Create partition using `mkpart` command of `parted` (fdisk command could also be used)
- Reload the partition table using `partprobe`

```
# parted /dev/sdc
(parted) mkpart
Partition name?  []? part1
File system type?  [ext2]? xfs
Start? 1
End? 10GiB

(parted) print
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdc: 2199GB
...
Number  Start   End     Size    File system  Name   Flags
 1      1049kB  10.7GB  10.7GB  xfs          part1
```

Delete GUID Partition (parted)

- Delete a partition using `mkpart` command of `parted` (fdisk command could also be used)
- Reload the partition table using `partprobe`

```
# parted /dev/sdc
(parted) rm 1
```

Delete Partition Table

- Delete dos/guid partition table by clobbering the disk contents using `dd` command
- Reload the partition table using `partprobe`

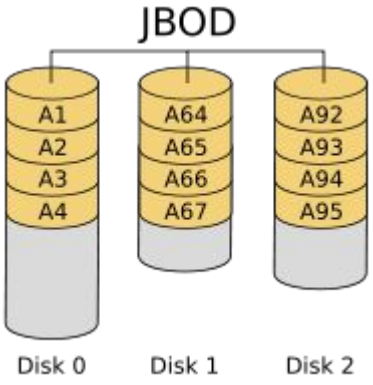
```
# dd if=/dev/zero of=/dev/sdb bs=1M count=1
1+0 records in
1+0 records out
1048576 bytes (1.0 MB, 1.0 MiB) copied, 0.017371 s, 60.4 MB/s

# parted /dev/sdc print
Error: /dev/sdc: unrecognised disk label
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdc: 2199GB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:
```

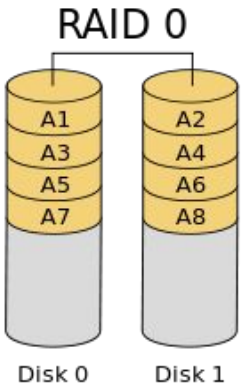
CAUTION: Beware of permanent data loss

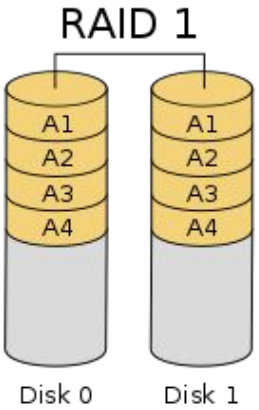
RAID

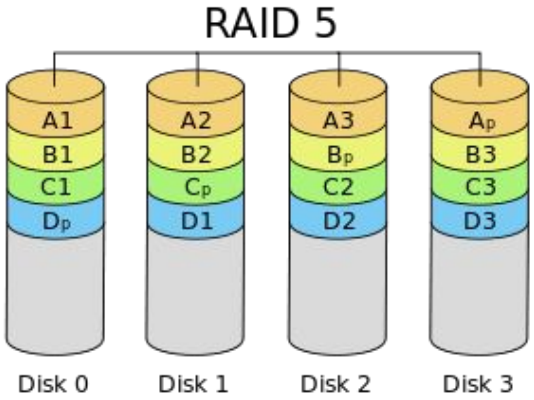
Redundant Array of Independent Disks (RAID)

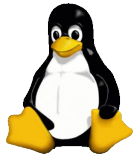
- RAID 0 - Concatenation (aka Just a Bunch Of Disks, JBOD)
- RAID 0 - Striping
- RAID 1 - Mirroring
- RAID 5 - Striping with distributed parity









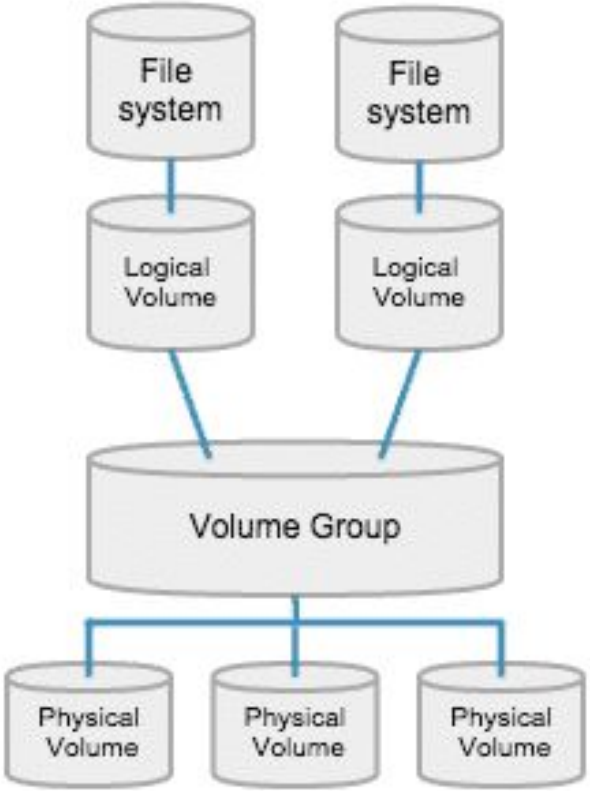


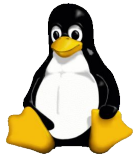
CAUTION: Hardware RAID in motherboard is single point of failure.

Logical Volume Management

Logical Volume Management (LVM)

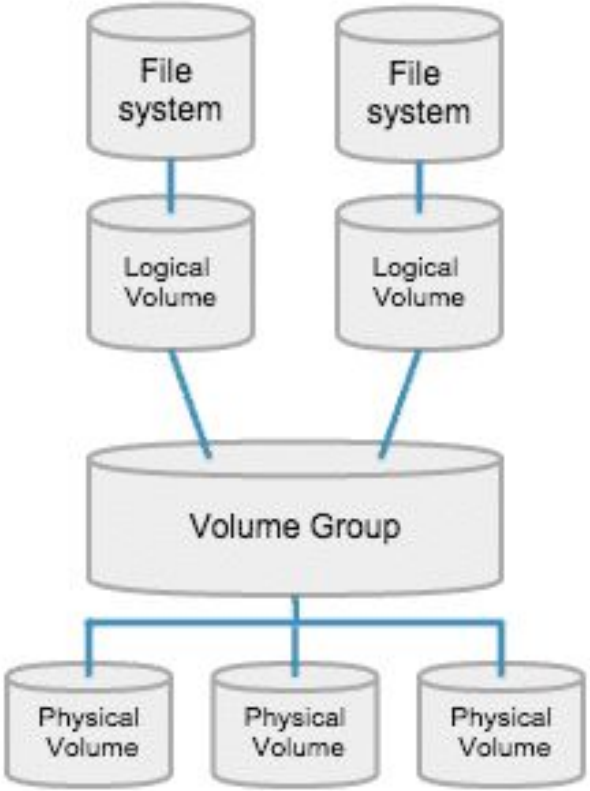
- A subsystem in OS kernel
  - Second layer from bottom in storage stack.
  - Implements Software RAID
  - Creates logical block devices on physical block devices
- Addresses limitations of disk drives using
  - Disk concatenation
  - Redundancy
- Different volume types
  - Concatenated volume (RAID 0 - concatenation)
  - Striped volume (RAID 0 - striping)
  - Mirrored volume (RAID 1)
  - Distributed parity volume (RAID 5)
  - Many many more…





Logical Volume Management (LVM)

- Constructs in LVM
  - Physical volume (PV) : a disk or partition
  - Physical Extent (PE) : a block from physical volume. Disk space is consumed in units of PEs.
  - Volume group (VG) : a group of physical volumes having similar specifications
  - Logical Extent (LE) : a logical block that is mapped to one or more PEs.
  - Logical Volume : A logically contiguous group of extents that can be mapped to one or more PVs.



Physical Volume Operations

- Create physical volume using `pvcreate` command.
- List physical volumes using `pvs` or `pvdisplay` command.
- Delete physical volume using `pvcreate` command.

```
# pvcreate /dev/sd[ghi]
  Physical volume "/dev/sdg" successfully created
  Physical volume "/dev/sdh" successfully created
  Physical volume "/dev/sdi" successfully created

# pvs
  PV         VG    Fmt  Attr PSize PFree
  /dev/sdg         lvm2 ---  1.00t 1.00t
  /dev/sdh         lvm2 ---  1.00t 1.00t
  /dev/sdi         lvm2 ---  1.00t 1.00t

# pvremove /dev/sd[ghi]
  Labels on physical volume "/dev/sdg" successfully wiped
  Labels on physical volume "/dev/sdh" successfully wiped
  Labels on physical volume "/dev/sdi" successfully wiped
```

CAUTION: Beware of permanent data loss

Volume Group Operations

- Create volume group using `vgcreate` command.
- List volume groups using `vgs` or `vgdisplay` command.
- Delete volume group using `vgremove` command.

```
# vgcreate myvg3 /dev/sd[ghi]
  Volume group "myvg3" successfully created

# vgs
  VG       #PV #LV #SN Attr   VSize VFree
  myvg3      3   0    0 wz--n- 3.00t 3.00t

# vgremove myvg3
  Volume group "myvg3" successfully removed
```

Logical Volume Operations

- Create a logical volume using `lvcreate` command.
- List logical volumes using `lvs` or `lvdisplay` or `lsblk` command.
- Delete a logical volume using `lvremove` command.

```
# lvcreate -n mylv01 -L 30GB myvg3
  Using default stripesize 64.00 KiB.
  Logical volume "mylv" created.

# lvs
  LV     VG    Attr       LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  mylv01 myvg3 -wi-a----- 30.00g

# lsblk -p /dev/mapper/myvg3-mylv01
NAME                        MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
/dev/mapper/myvg3-mylv01      253:0     0      30G  0 lvm

# lvremove /dev/mapper/myvg3-mylv01
Do you really want to remove and DISCARD active logical volume mylv01? [y/n]: y
  Logical volume "mylv01" successfully removed
```

Creating a RAID 0 linear volume

- Create a RAID0 linear logical volume using `lvcreate` command.

```
# lvcreate -n myraid0linear1  -L 1.5TB myvg3
  Logical volume "myraid0linear1" created.

# pvs
  PV         VG     Fmt  Attr PSize    PFree
  /dev/sdg   myvg3  lvm2 a--  1024.00g       0
  /dev/sdh   myvg3  lvm2 a--  1024.00g  511.99g
  /dev/sdi   myvg3  lvm2 a--  1024.00g 1024.00g
```

```
# lvdisplay -m /dev/myvg3/myraid0linear1
  ...
  LV Size                1.50 TiB
  --- Segments ---
  Logical extents 0 to 262142:
     Type       linear
     Physical extents    0 to 262142
     ..
  Logical extents 262143 to 393215:
     ...
     Physical extents    0 to 131072
```

Creating a RAID 0 striped volume

- Create a RAID0 striped logical volume using `lvcreate -i` command.

```
# lvcreate -n myraid0striped1  -L 1.5TB -i 3 myvg3
  Using default stripesize 64.00 KiB.
  Logical volume "myraid0striped1" created.

# pvs
  PV         VG     Fmt  Attr PSize    PFree
  /dev/sdg   myvg3  lvm2 a--  1024.00g 512.00g
  /dev/sdh   myvg3  lvm2 a--  1024.00g 512.00g
  /dev/sdi   myvg3  lvm2 a--  1024.00g 512.00g
```

```
# lvdisplay -m /dev/myvg3/myraid0striped1
  ...
  LV Size                1.50 TiB
  --- Segments ---
  Logical extents 0 to 393215:
     Type        striped
     Stripes          3
     Stripe size      64.00 KiB
     Stripe 0:
     ...
     Physical extents     0 to 131071
     Stripe 1:
     ...
     Physical extents     0 to 131071
     Stripe 2:
     ...
     Physical extents     0 to 131071
```

Creating a RAID 1 (mirrored) volume

- Create a RAID1 (mirrored) logical volume using `lvcreate -m` command.
- -m specifies number of mirrors in addition to the original copy.

```
# lvcreate -n myraid0striped1  -L 1.5TB -i 3 myvg3
  Using default stripesize 64.00 KiB.
  Logical volume "myraid0striped1" created.

# pvs
  PV         VG          Fmt  Attr PSize    PFree
  /dev/sdg   myvg3 lvm2 a--  1024.00g 512.00g
  /dev/sdh   myvg3 lvm2 a--  1024.00g 512.00g
  /dev/sdi   myvg3 lvm2 a--  1024.00g 512.00g
```

```
# lvdisplay -m /dev/myvg3/myraid0striped1
  ...
  LV Size                1.50 TiB
  ...
  Logical extents 0 to 393215:
    Type        striped
    Stripes         3
    Stripe size     64.00 KiB
    Stripe 0:
    Physical extents    0 to 131071
    ...
    Stripe 1:
    Physical extents    0 to 131071
    ...
    Stripe 2:
    Physical extents    0 to 131071
    ...
```

Creating a RAID 5 volume

- Create a RAID5 (striping, distributed parity) logical volume using `lvcreate --type raid5` command.

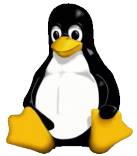
```
# lvcreate -n myraid5dp1 -L 1.5TB --type raid5 myvg3
  Logical volume "myraid5dp1" created.

# pvs
  PV         VG          Fmt  Attr PSize    PFree
  /dev/sdg   myvg3 lvm2 a--  1024.00g 255.99g
  /dev/sdh   myvg3 lvm2 a--  1024.00g 255.99g
  /dev/sdi   myvg3 lvm2 a--  1024.00g 255.99g
```

```
# lvdisplay -m /dev/myvg3/myraid0striped1
  ...
  LV Size                1.50 TiB
  ...
  Logical extents 0 to 393215:
     Type       raid5
     Raid Data LV 0
     ...
```

File Systems

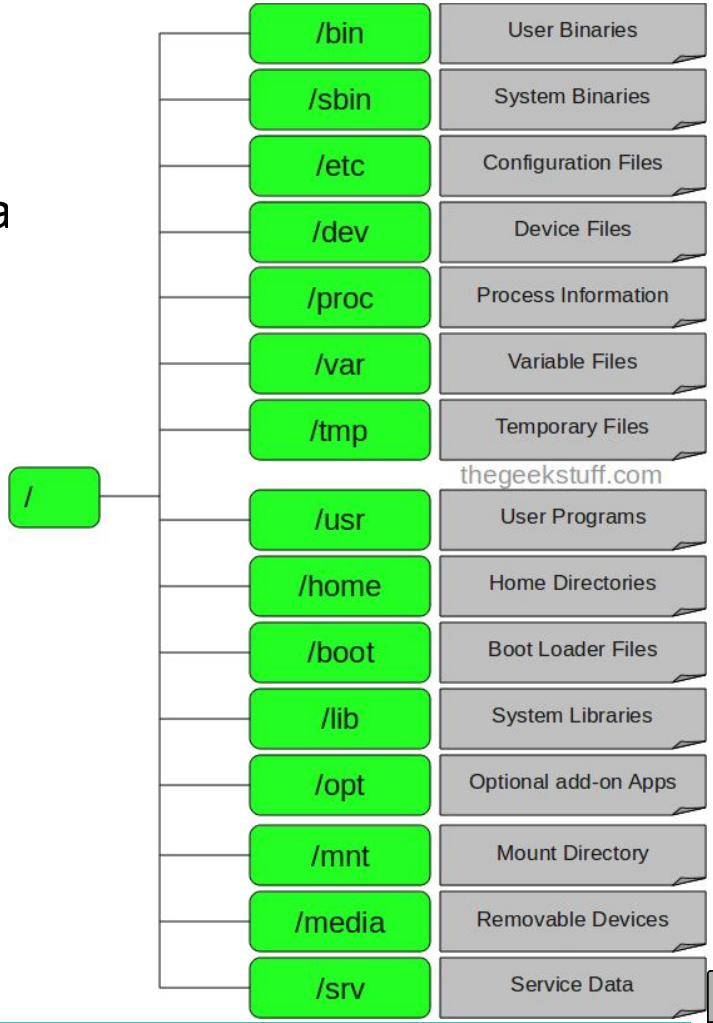
File system (Recap)

- A subsystem in OS kernel
- Logical organization of disk sectors/blocks into files and directories.
- Does accounting of free/used space
- Provides quotas at user/group level
- Provides security using ownership, permissions, access controls (ACL).
- Addresses limitations of disk drives using
  - Logical blocking
  - Caching
- Different file systems types
  - xfs
  - ext4
  - tmpfs
  - iso9660
  - ...



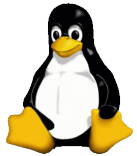
File systems hierarchy (Recap)

- Multiple file systems organized in a tree structure
- Top most directory is / (called root directory)



File system classification

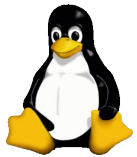
- File systems could be classified into
  - Local file systems (Disk based)
    Eg. ext4, xfs, iso9660, vfat, ntfs, etc.
  - Pseudo file systems (In memory)
    Eg. procfs, sysfs, cgroupfs, encryptfs, tmpfs, etc.
  - Cluster file systems (Tightly coupled + shared storage based)
    Eg. gfs, vxfs, etc.
  - Network file systems (Loosely coupled + network based + not scalable)
    Eg. nfsv3, cifs/smb
  - Distributed systems (Loosely coupled + networked based + scalable)
    Eg. cephfs, hdfs, pnfs (nfsv4.1), lustrefs, glusterfs, googlefs.



File system types

- List all file systems types using `cat /proc/filesystems`

```
$ cat /proc/filesystems
nodev    sysfs
...
nodev    proc
...
nodev    tmpfs
nodev    devtmpfs
...
         ext3
         ext2
         ext4
         Vfat
...
         xfs
...
```



Formatting a file system

- To format a block device with file system use, `mkfs -t <fstype> <blk_device>`
- To find file system type of a block device use, `blkid <blk_device>`

```
# mkfs -t ext4 /dev/nvme0n1p2
# mkfs -t xfs /dev/nvme0n1p4

# blkid
...
/dev/nvme0n1p2: UUID="4c19d94f-0724-4974-b5db-4f39973254d5" TYPE="ext4"
PARTUUID="3737f29f-02"
/dev/nvme0n1p4: UUID="1cddca11-8da7-43eb-9de3-e80c088082c4" TYPE="xfs" PARTUUID="3737f29f-04"
...
```

Mounting a file system

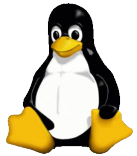
- The process of bringing a file-system online.
- Mount a file system using `mount <blk_device> <mount_point>`
- Command line options generic and specific to file-system type.

```
# mkdir /mnt/nvm1p4
# mount /dev/nvme0n1p4 /mnt/nvm1p4

# df -hT
Filesystem     Type      Size  Used Avail Use% Mounted on
...
/dev/nvme0n1p4 xfs        10G   43M   10G   1% /mnt/nvm1p4

# mount -o remount,ro /dev/nvme0n1p4 /mnt/nvm1p4
```

Auto mounting a file system

- To automatically mount file-systems during boot, add entry to /etc/fstab. Validate fstab using `mount -a` before rebooting.

```
# mkdir /mnt/nvm1p2
# vi /etc/fstab
...
UUID=4c19d94f-0724-4974-b5db-4f39973254d5 /mnt/nvm1p2 ext4  defaults 2 2
...

# mount -a

$ df -hT
Filesystem      Type       Size  Used Avail Use% Mounted on
...
/dev/nvme0n1p2  ext4        25G   44M   24G   1% /mnt/nvm1p2
...
```



Listing file systems

- List all file systems using `df -hT` or `lsblk`

```
$ df -hT
Filesystem      Type       Size  Used Avail Use% Mounted on
/dev/sda1       ext4        98G  5.4G   88G   6% /
...
/dev/nvme0n1p2  ext4        25G   44M   24G   1% /mnt/nvm1p2
/dev/nvme0n1p4  xfs         10G   43M   10G   1% /mnt/nvm1p4
...

# lsblk
NAME         MAJ:MIN  RM   SIZE RO TYPE MOUNTPOINT
nvme0n1      259:0     0   100G  0 disk
├─nvme0n1p4 259:4     0    10G  0 part /mnt/nvm1p4
└─nvme0n1p2 259:2     0    25G  0 part /mnt/nvm1p2
sda            8:0     0   100G  0 disk
├─sda2         8:2     0     1K  0 part
├─sda5         8:5     0   975M  0 part [SWAP]
└─sda1         8:1     0    99G  0 part /
```

Unmounting a file system

- The process of taking a file system offline.
- Unmount a file system using `umount <mount_point>`

```
# umount /mnt/nvm1p4
```

Repairing a file system

- To repair a file system use, `fsck` or `xfs_repair`

```
# fsck -t ext4 /dev/nvme0n1p2
...
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/nvme0n1p2: 11/1638400 files (0.0% non-contiguous), 146849/6553600 blocks

# xfs_repair /dev/nvme0n1p4
Phase 1 - find and verify superblock...
Phase 2 - using internal log
Phase 3 - for each AG...
Phase 4 - check for duplicate blocks...
Phase 5 - rebuild AG headers and trees...
Phase 6 - check inode connectivity...
Phase 7 - verify and correct link counts...
done
```

References

References

- Linux manual pages
- www.wikipedia.org
- Courtesy Google images

Q & A