Linux 2nd Module:

Managing Basic Storage

Partitioning a Disk

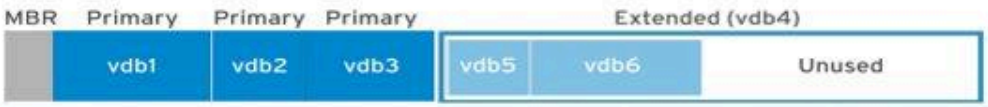
Disk partitioning allows system administrators to divide a hard drive into multiple logical storage units, referred to as partitions. By separating a disk into partitions, system administrators can use different partitions to perform different functions.

For example, disk partitioning is necessary or beneficial in these situations:

- Limit available space to applications or users.
- Separate operating system and program files from user files.
- Create a separate area for memory swapping.
- Limit disk space use to improve the performance of diagnostic tools and backup imaging.

MBR Partitioning Scheme

Since 1982, the *Master Boot Record (MBR)* partitioning scheme has dictated how disks are partitioned on systems running BIOS firmware. This scheme supports a maximum of four primary partitions. On Linux systems, with the use of extended and logical partitions, administrators can create a maximum of 15 partitions. Because partition size data is stored as 32-bit values, disks partitioned with the MBR scheme have a maximum disk and partition size of 2 TiB.



Use the lsblk command to scan the block devices connected to a machine

```
[root@raghav ~]# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda         8:0    0   20G  0 disk
├─sda1      8:1    0    1G  0 part /boot
└─sda2      8:2    0   19G  0 part
  ├─cs-root
  │       253:0    0   17G  0 lvm  /
  └─cs-swap
          253:1    0    2G  0 lvm  [SWAP]
sdb         8:16   0    1G  0 disk
sdc         8:32   0    1G  0 disk
└─sdc1      8:33   0  500M  0 part
sdd         8:48   0    1G  0 disk
sr0        11:0    1  8.1G  0 rom  /run/media/root/CentOS-Stream-9-BaseOS-x86_64
```

Use the lsblk --fs command to scan the block devices connected to a machine and retrieve the file system UUIDs.

```
[root@raghav ~]# lsblk --fs
NAME FSTYPE FSVER LABEL UUID                                 FSAVAIL FSUSE% M
UNTPOINTS
sda
├─sda1
│    xfs               2697663f-ef82-471b-abe4-6764b57a42ce    749.9M    26% /
oot
└─sda2
     LVM2_m LVM2        3M5wpd-TCFY-twHW-RBT8-tU0P-1x9P-NQfdd4
  ├─cs-root
  │  xfs               47e2fe38-e8d3-46d0-a8c4-81d263a17856      2.8G    84% /
  └─cs-swap
     swap   1          5870c342-0abd-4c33-b5fe-5221575a7261                  [
WAP]
sdb  LVM2_m LVM2        R9HiXO-j8Pf-K7dy-LEnb-0YjW-Vg4K-rNRyhc
sdc
└─sdc1
     xfs               26a0cc5a-0585-449f-91e1-f258320b073c
```

Managing Partitions with fdisk

Partition editors are programs which allow administrators to make changes to a disk's partitions, such as creating partitions, deleting partitions, and changing partition types. To perform these operations, administrators can use the Parted partition editor for both the MBR and the GPT partitioning scheme.

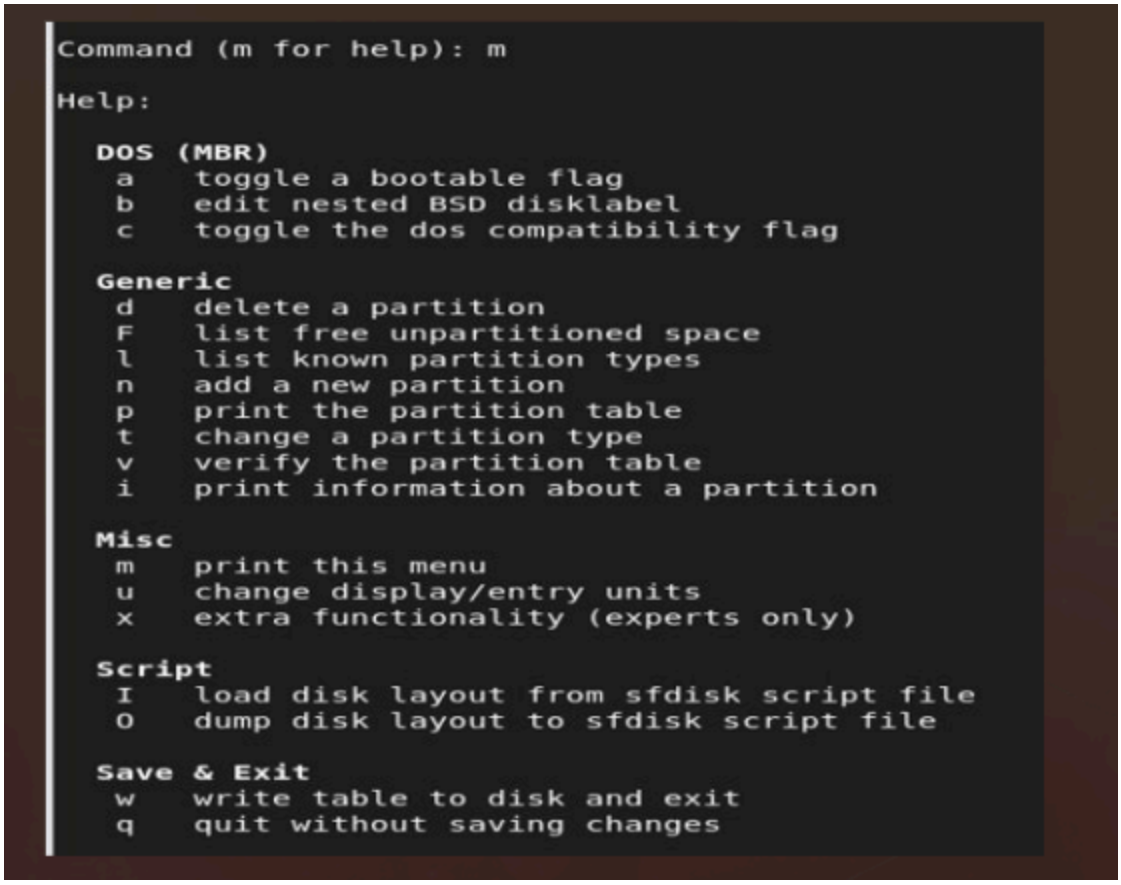
The fdisk command takes the device name of the whole disk as the first argument and one or more subcommands. The following example uses the print subcommand to display the partition table on the /dev/vda disk

FDISK command to create partition

```
[root@raghav ~]# fdisk /dev/sdb

Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

The device contains 'LVM2_member' signature and it will be removed by a write command. See fdisk(8) man page and --wipe option
for more details.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xff4504a8.

Command (m for help):
```

Red Hat Enterprise Linux supports many different file system types, but two common ones are XFS and ext4. Anaconda, the installer for Red Hat Enterprise Linux, uses XFS by default. As root, use the mkfs.xfs command to apply an XFS file system to a block device. For ext4, use mkfs.ext4.

```
[root@host ~]# mkfs.xfs /dev/vdb1
meta-data=/dev/vdb1              isize=512    agcount=4, agsize=60992 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1
```


Command (m for help): m
Help:
DOS (MBR)
a toggle a bootable flag
b edit nested BSD disklabel
c toggle the dos compatibility flag
Generic
d delete a partition
F list free unpartitioned space
l list known partition types
n add a new partition
p print the partition table
t change a partition type
v verify the partition table
i print information about a partition
Misc
m print this menu
u change display/entry units
x extra functionality (experts only)
Script
I load disk layout from sfdisk script file
O dump disk layout to sfdisk script file
Save & Exit
w write table to disk and exit
q quit without saving changes

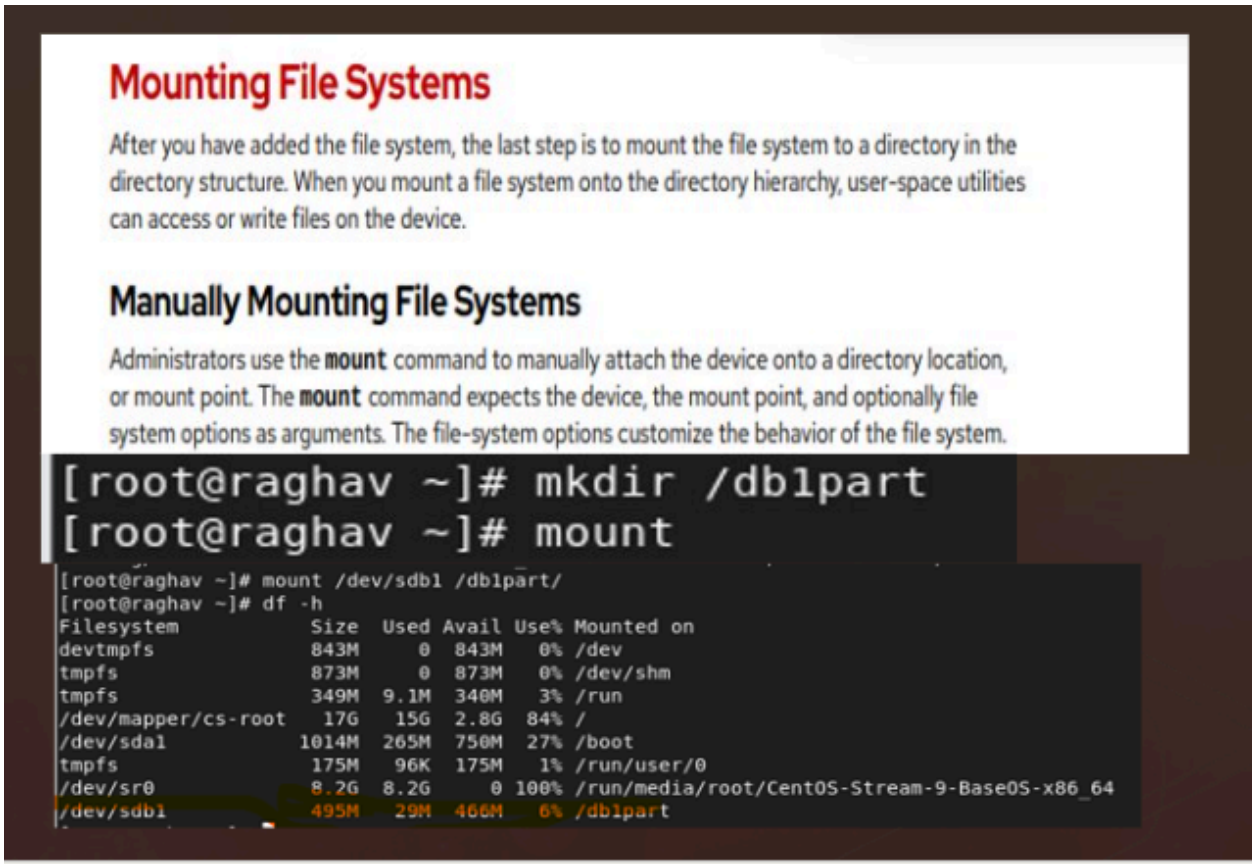
Create partition using fdisk

```
Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (1-4, default 1):
First sector (2048-2097151, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-2097151, default 2097151): +500M

Created a new partition 1 of type 'Linux' and of size 500 MiB.
Partition #1 contains a xfs signature.

Do you want to remove the signature? [Y]es/[N]o: n

Command (m for help): p

Disk /dev/sdb: 1 GiB, 1073741824 bytes, 2097152 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xff4504a8

Device     Boot Start     End Sectors  Size Id Type
/dev/sdb1        2048 1026047 1024000  500M 83 Linux

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

```
[root@raghav ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda           8:0    0   20G  0 disk
├─sda1        8:1    0    1G  0 part /boot
└─sda2        8:2    0   19G  0 part
  ├─cs-root 253:0    0   17G  0 lvm  /
  └─cs-swap 253:1    0    2G  0 lvm  [SWAP]
sdb           8:16   0    1G  0 disk
└─sdb1        8:17   0  500M  0 part
sdc           8:32   0    1G  0 disk
└─sdc1        8:33   0  500M  0 part
sdd           8:48   0    1G  0 disk
sr0          11:0    1 8.1G  0 rom  /run/media/root/CentOS-Stream-9-BaseOS-x86_64
```

```
[root@raghav ~]# mkfs.xfs /dev/sdb1
mkfs.xfs: /dev/sdb1 appears to contain an existing filesystem (xfs).
mkfs.xfs: Use the -f option to force overwrite.
[root@raghav ~]# mkfs.xfs -f  /dev/sdb1
meta-data=/dev/sdb1              isize=512    agcount=4, agsize=32000 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=1 inobtcount=1
data     =                       bsize=4096   blocks=128000, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=1368, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

F


Mounting File Systems
After you have added the file system, the last step is to mount the file system to a directory in the directory structure. When you mount a file system onto the directory hierarchy, user-space utilities can access or write files on the device.
Manually Mounting File Systems
Administrators use the mount command to manually attach the device onto a directory location, or mount point. The mount command expects the device, the mount point, and optionally file system options as arguments. The file-system options customize the behavior of the file system.
[root@raghav ~]# mkdir /db1part
[root@raghav ~]# mount
[root@raghav ~]# mount /dev/sdb1 /db1part/
[root@raghav ~]# df -h
Filesystem Size Used Avail Use% Mounted on
devtmpfs 843M 0 843M 0% /dev
tmpfs 873M 0 873M 0% /dev/shm
tmpfs 349M 9.1M 340M 3% /run
/dev/mapper/cs-root 17G 15G 2.8G 84% /
/dev/sda1 1014M 265M 750M 27% /boot
tmpfs 175M 96K 175M 1% /run/user/0
/dev/sr0 8.2G 8.2G 0 100% /run/media/root/CentOS-Stream-9-BaseOS-x86_64
/dev/sdb1 495M 29M 466M 6% /db1part

Persistently Mounting File Systems

Manually mounting a file system is a good way to verify that a formatted device is accessible and working as expected. However, when the server reboots, the system does not automatically mount the file system onto the directory tree again; the data is intact on the file system, but users cannot access it.

To make sure that the system automatically mounts the file system at system boot, add an entry to the **`/etc/fstab`** file. This configuration file lists the file systems to mount at system boot.

`/etc/fstab` is a white-space-delimited file with six fields per line.

```
[root@raghav ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Sat Dec 17 12:42:50 2022
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/cs-root     /                       xfs     defaults        0 0
UUID=2697663f-ef82-471b-abe4-6764b57a42ce /boot                   xfs     defaults        0 0
/dev/mapper/cs-swap     none                    swap    defaults        0 0
/dev/sdb1       /db1part        xfs     defaults        0 0
```

```
[root@raghav ~]# blkid /dev/sdb1
/dev/sdb1: UUID="f39c8594-0bd0-427f-8d74-090779178c14" BLOCK_SIZE="512" TYPE="xfs" PARTUUID="ff4504a8-01"
```

```
[root@raghav ~]# df -h
Filesystem          Size  Used Avail Use% Mounted on
devtmpfs            843M     0  843M   0% /dev
tmpfs               873M     0  873M   0% /dev/shm
tmpfs               349M  9.1M  340M   3% /run
/dev/mapper/cs-root  17G   15G  2.8G  84% /
/dev/sda1          1014M  265M  750M  27% /boot
tmpfs               175M   96K  175M   1% /run/user/0
/dev/sr0            8.2G  8.2G     0 100% /run/media/root/CentOS-Stream-9-BaseOS-x86_64
/dev/sdb1           495M   29M  466M   6% /db1part
```

fdisk partition remove

```
[root@raghav ~]# fdisk /dev/sdb

Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

This disk is currently in use - repartitioning is probably a bad idea.
It's recommended to umount all file systems, and swapoff all swap
partitions on this disk.


Command (m for help): d

Selected partition 1
Partition 1 has been deleted.
```

```
[root@raghav ~]# lsblk
NAME          MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda             8:0    0    20G  0 disk
├─sda1          8:1    0     1G  0 part /boot
└─sda2          8:2    0    19G  0 part
  ├─cs-root   253:0    0    17G  0 lvm  /
  └─cs-swap   253:1    0     2G  0 lvm  [SWAP]
sdb             8:16   0     1G  0 disk
```

Managing Partitions with Parted

```
[root@raghav ~]# parted /dev/sdb
(parted) mkpart
Partition type?  primary/extended? primary
File system type?  [ext2]? xfs
Start? 500
End? 100
Error: Can't have the end before the start! (start sector=976562 length=-781249)
(parted) mkpart
Partition type?  primary/extended? primary
File system type?  [ext2]? xfs
Start? 500
End? 200
Error: Can't have the end before the start! (start sector=976562 length=-585936)
(parted) mkpart
Partition type?  primary/extended? primary
File system type?  [ext2]? xfs
Start? 2048
Error: The location 2048 is outside of the device /dev/sdb.
(parted) mkpart
Partition type?  primary/extended? primary
File system type?  [ext2]? xfs
Start? 2048s
End? 500MB
(parted) print
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sdb: 1074MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type     File system  Flags
 1      1049kB  500MB   499MB   primary  xfs
```

After creating the parted partition. Continue with creating filesystem and the mount the disk process.

Managing Swap Space

Objectives

After completing this section, you should be able to create and manage swap spaces to supplement physical memory.

Introducing Swap Space Concepts

A *swap space* is an area of a disk under the control of the Linux kernel memory management subsystem. The kernel uses swap space to supplement the system RAM by holding inactive pages of memory. The combined system RAM plus swap space is called *virtual memory*.

When the memory usage on a system exceeds a defined limit, the kernel searches through RAM looking for idle memory pages assigned to processes. The kernel writes the idle pages to the swap area and reassigns the RAM pages to other processes. If a program requires access to a page on disk, the kernel locates another idle page of memory, writes it to disk, then recalls the needed page from the swap area.

Because swap areas reside on disk, swap is slow when compared with RAM. While it is used to augment system RAM, you should not consider swap space as a sustainable solution for insufficient RAM for your workload.

Sizing the Swap Space

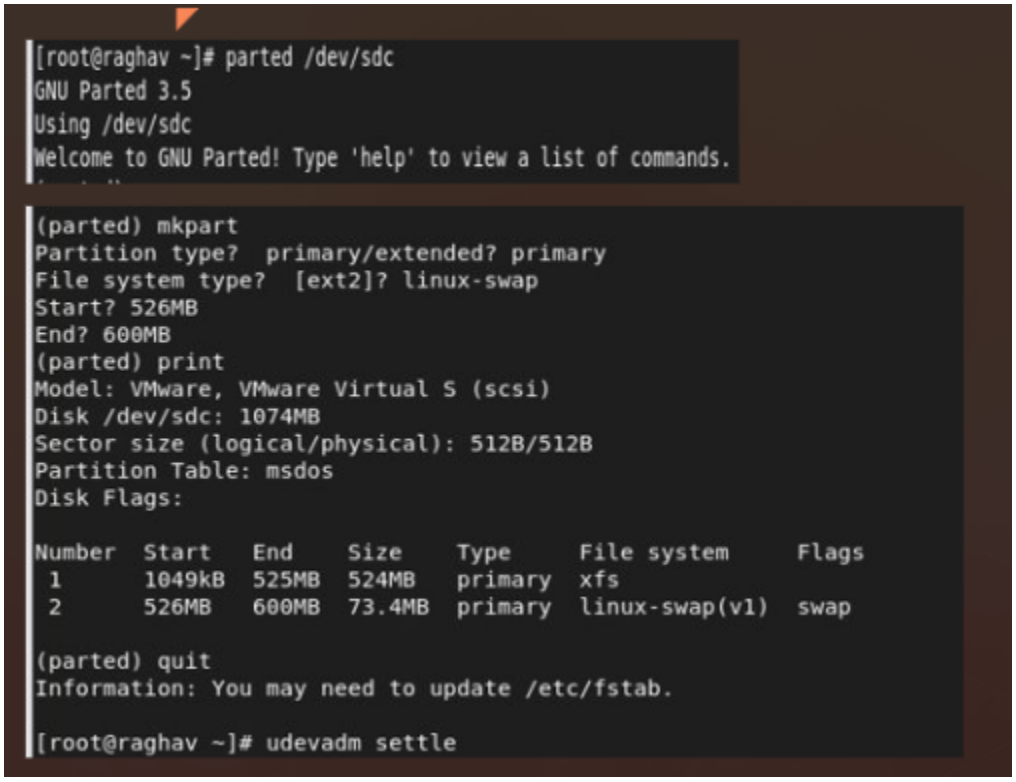
Administrators should size the swap space based on the memory workload on the system. Application vendors sometimes provide recommendations on that subject. The following table provides some guidance based on the total amount of physical memory.

RAM and Swap Space Recommendations

RAM	Swap Space	Swap Space if Allowing for Hibernation
2 GiB or less	Twice the RAM	Three times the RAM
Between 2 GiB and 8 GiB	Same as RAM	Twice the RAM
Between 8 GiB and 64 GiB	At least 4 GiB	1.5 times the RAM
More than 64 GiB	At least 4 GiB	Hibernation is not recommended

The laptop and desktop hibernation function uses the swap space to save the RAM contents before powering off the system. When you turn the system back on, the kernel restores the RAM contents from the swap space and does not need a complete boot. For those systems, the swap space needs to be greater than the amount of RAM.

The Knowledgebase article in the Reference section at the end of this section gives more guidance on sizing the swap space.


[root@raghav ~]# parted /dev/sdc
GNU Parted 3.5
Using /dev/sdc
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) mkpart
Partition type? primary/extended? primary
File system type? [ext2]? linux-swap
Start? 526MB
End? 600MB
(parted) print
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sdc: 1074MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:
Number Start End Size Type File system Flags
1 1049kB 525MB 524MB primary xfs
2 526MB 600MB 73.4MB primary linux-swap(v1) swap
(parted) quit
Information: You may need to update /etc/fstab.
[root@raghav ~]# udevadm settle

```
[root@raghav ~]# udevadm settle
[root@raghav ~]# mkswap /dev/sdc2
Setting up swapspace version 1, size = 70 MiB (73396224 bytes)
no label, UUID=c9d244bb-d6c8-439d-8e20-e6c79ea809c0
[root@raghav ~]# swapon --show
NAME      TYPE      SIZE USED PRIO
/dev/dm-1 partition   2G   0B   -2
/dev/sdb2 partition   3M   0B   -4
/dev/sdb1 partition 477M   0B   -3
[root@raghav ~]# swapon /dev/sdc2
[root@raghav ~]# swapon --show
NAME      TYPE      SIZE USED PRIO
/dev/dm-1 partition   2G   0B   -2
/dev/sdb2 partition   3M   0B   -4
/dev/sdb1 partition 477M   0B   -3
/dev/sdc2 partition  70M   0B   -5
[root@raghav ~]# swapoff /dev/sdc2
[root@raghav ~]# free
               total        used        free      shared  buff/cache   available
Mem:         1786484      900024      274964       23264      611496      703268
Swap:        2588660           0     2588660
[root@raghav ~]# swapon /dev/sdc2
[root@raghav ~]# free
```

```
[root@raghav ~]# vi /etc/fstab
#
# /etc/fstab
# Created by anaconda on Sat Dec 17 12:42:50 2022
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/cs-root     /                       xfs     defaults        0 0
UUID=2697663f-ef82-471b-abe4-6764b57a42ce /boot                   xfs     defaults        0 0
/dev/mapper/cs-swap     none                    swap    defaults        0 0
UUID=c9d244bb-d6c8-439d-8e20-e6c79ea809c0       swap    swap    pri=4   0 0

[root@raghav ~]# swapon -a
[root@raghav ~]# swapon --show
NAME      TYPE      SIZE USED PRIO
/dev/dm-1 partition   2G   0B   -2
/dev/sdb2 partition   3M   0B   -4
/dev/sdb1 partition 477M   0B   -3
/dev/sdc2 partition  70M   0B    4
```

Fdisk - swap

```
Command (m for help): p
Disk /dev/sdc: 3 GiB, 3221225472 bytes, 6291456 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x1d13f796

Device     Boot Start     End Sectors Size Id Type
/dev/sdc1        2048 6291455 6289408   3G 83 Linux

Command (m for help): l

00 Empty            24 NEC DOS          81 Minix / old Lin  bf Solaris
01 FAT12            27 Hidden NTFS Win  82 Linux swap / So  c1 DRDOS/sec (FAT-
02 XENIX root       39 Plan 9           83 Linux            c4 DRDOS/sec (FAT-
03 XENIX usr        3c PartitionMagic   84 OS/2 hidden or   c6 DRDOS/sec (FAT-
04 FAT16 <32M       40 Venix 80286      85 Linux extended   c7 Syrinx
05 Extended         41 PPC PReP Boot    86 NTFS volume set  da Non-FS data
06 FAT16            42 SFS              87 NTFS volume set  db CP/M / CTOS / .
```

```
Command (m for help): t
Selected partition 1
Hex code or alias (type L to list all): 82
Changed type of partition 'Linux' to 'Linux swap / Solaris'.

Command (m for help): p
Disk /dev/sdc: 3 GiB, 3221225472 bytes, 6291456 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x1d13f796

Device     Boot Start     End Sectors Size Id Type
/dev/sdc1        2048 6291455 6289408   3G 82 Linux swap / Solaris

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

Creating Logical Volumes

Logical Volume Management (LVM) Concepts

Logical volumes and logical volume management make it easier to manage disk space. If a file system that hosts a logical volume needs more space, it can be allocated to its logical volume from the free space in its volume group and the file system can be resized. If a disk starts to fail, a replacement disk can be registered as a physical volume with the volume group and the logical volume's extents can be migrated to the new disk.

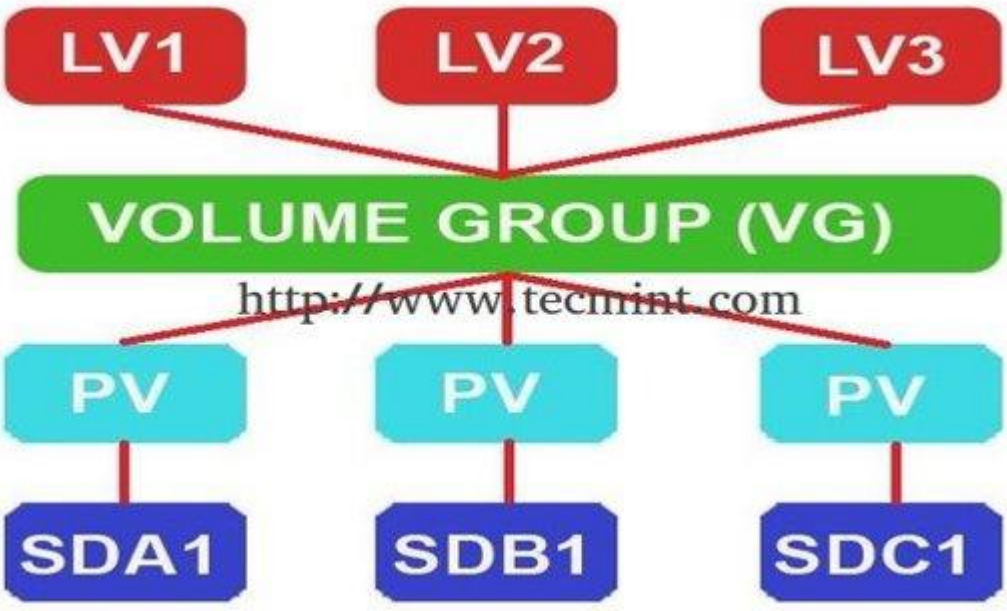
LVM Definitions

Physical devices
Physical devices are the storage devices used to save data stored in a logical volume. These are block devices and could be disk partitions, whole disks, RAID arrays, or SAN disks. A device must be initialized as an LVM physical volume in order to be used with LVM. The entire device will be used as a physical volume.

Physical volumes (PVs)
Physical volumes are the underlying "physical" storage used with LVM. You must initialize a device as a physical volume before using it in an LVM system. LVM tools segment physical volumes into *physical extents (PEs)*, which are small chunks of data that act as the smallest storage block on a physical volume.

Volume groups (VGs)
Volume groups are storage pools made up of one or more physical volumes. This is the functional equivalent of a whole disk in basic storage. A PV can only be allocated to a single VG. A VG can consist of unused space and any number of logical volumes.

Logical volumes (LVs)
Logical volumes are created from free physical extents in a volume group and provide the "storage" device used by applications, users, and the operating system. LVs are a collection of *logical extents (LEs)*, which map to physical extents, the smallest storage chunk of a PV. By default, each LE maps to one PE. Setting specific LV options changes this mapping; for example, mirroring causes each LE to map to two PEs.



Create Physical Volume & display

```
[root@localhost ~]# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda         8:0    0   20G  0 disk
├─sda1      8:1    0    1G  0 part /boot
└─sda2      8:2    0   19G  0 part
  ├─cs-root
  │       253:0    0   17G  0 lvm  /
  └─cs-swap
          253:1    0    2G  0 lvm  [SWAP]
sdb         8:16   0    1G  0 disk
└─sdb1      8:17   0  476M  0 part
sdc         8:32   0    2G  0 disk
sdd         8:48   0    2G  0 disk
sr0        11:0    1  8.1G  0 rom  /run/media/root/CentOS-Stream-9-BaseOS-x86_64
[root@localhost ~]# pv
pvchange   pvcreate   pvmove     pvresize   pvscan
pvck       pvdisplay  pvremove   pvs
[root@localhost ~]# pvcreate /dev/sdc /de
demo/  demo3/ demo4/ dev/
[root@localhost ~]# pvcreate /dev/sdc /dev/
Display all 173 possibilities? (y or n)
[root@localhost ~]# pvcreate /dev/sdc /dev/sdd
  Physical volume "/dev/sdc" successfully created.
  Physical volume "/dev/sdd" successfully created.
```

/

```
[root@localhost ~]# pvdisplay /dev/sdc /dev/sdd
  "/dev/sdc" is a new physical volume of "2.00 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/sdc
  VG Name
  PV Size               2.00 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               WKmHZo-qe0D-vmhh-4dF0-WhxK-VLtN-n9I1p9

  "/dev/sdd" is a new physical volume of "2.00 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/sdd
  VG Name
  PV Size               2.00 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               ifN07d-kOuO-R3cB-yVq8-Lemj-71Jx-YIt23N
```

Volume Group create and display

```
[root@localhost ~]# vgcreate vg01 /dev/sdc /dev/sdd
  Volume group "vg01" successfully created
[root@localhost ~]# vgdisplay vg01
  --- Volume group ---
  VG Name               vg01
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               3.99 GiB
  PE Size               4.00 MiB
  Total PE              1022
  Alloc PE / Size       0 / 0
  Free  PE / Size       1022 / 3.99 GiB
  VG UUID               SLxBRV-rlso-Jbqz-dhnI-Khe0-k4iQ-hTuWZM
```

Logical Volume create & display

```
[root@localhost ~]# lvcreate -n lv01 -L 500M vg01
  Logical volume "lv01" created.
[root@localhost ~]# lvdisplay /dev/
/dev/cs/root    /dev/cs/swap    /dev/vg01/lv01
[root@localhost ~]# lvdisplay /dev/vg01/lv01
  --- Logical volume ---
  LV Path                /dev/vg01/lv01
  LV Name                lv01
  VG Name                vg01
  LV UUID                603VB1-FBM9-d53m-LIfg-YEET-KedH-B5ngSE
  LV Write Access        read/write
  LV Creation host, time localhost.localdomain, 2023-04-01 16:16:53 +0530
  LV Status              available
  # open                 0
  LV Size                500.00 MiB
  Current LE             125
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:2
```

Create File system and mount logical volume

```
[root@localhost ~]# mkfs.xfs /dev/vg01/lv01
meta-data=/dev/vg01/lv01         isize=512    agcount=4, agsize=32000 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=1 inobtcount=1
data     =                       bsize=4096   blocks=128000, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=1368, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[root@localhost ~]# mkdir /lv
[root@localhost ~]# mount /dev/vg01/lv01 /lv
[root@localhost ~]# df -h
Filesystem             Size  Used Avail Use% Mounted on
devtmpfs               843M     0  843M   0% /dev
tmpfs                  873M     0  873M   0% /dev/shm
tmpfs                  349M   15M  335M   5% /run
/dev/mapper/cs-root     17G  5.0G   13G  30% /
/dev/sda1             1014M  265M  750M  27% /boot
tmpfs                  175M  100K  175M   1% /run/user/0
/dev/sr0               8.2G  8.2G     0 100% /run/media/root/CentOS-Stream-9-BaseOS-x86_64
/dev/mapper/vg01-lv01  495M   29M  466M   6% /lv
```

Controlling Access to Files with ACLs

Access Control List Concepts

Standard Linux file permissions are satisfactory when files are used by only a single owner, and a single designated group of people. However, some use cases require that files are accessed with different file permission sets by multiple named users and groups. *Access Control Lists (ACLs)* provide this function.

With ACLs, you can grant permissions to multiple users and groups, identified by user name, group name, UID, or GID, using the same permission flags used with regular file permissions: read, write, and execute. These additional users and groups, beyond the file owner and the file's group affiliation, are called *named users* and *named groups* respectively, because they are named not in a long listing, but rather within an ACL.

Users can set ACLs on files and directories that they own. Privileged users, assigned the **CAP_FOWNER** Linux capability, can set ACLs on any file or directory. New files and subdirectories automatically inherit ACL settings from the parent directory's default ACL, if they are set. Similar to normal file access rules, the parent directory hierarchy needs at least the *other* search (execute) permission set to enable named users and named groups to have access.

File-system ACL Support

File systems need to be mounted with ACL support enabled. XFS file systems have built-in ACL support. Other file systems, such as ext3 or ext4 created on Red Hat Enterprise Linux 8, have the **acl** option enabled by default, although on earlier versions you should confirm that ACL support is enabled. To enable file-system ACL support, use the ACL option with the **mount** command or in the file system's entry in **/etc/fstab** configuration file.

Viewing and Interpreting ACL Permissions

The **ls -l** command only outputs minimal ACL setting details:

```
[user@host content]$ ls -l reports.txt
-rwxrw----+ 1 user operators 130 Mar 19 23:56 reports.txt
```

The plus sign (+) at the end of the 10-character permission string indicates that an extended ACL structure with entries exists on this file.

user:
Shows the *user* ACL settings, which are the same as the standard *user* file settings; **rwx**.

group:
Shows the current ACL *mask* settings, not the *group owner* settings; **rw**.

other:
Shows the *other* ACL settings, which are the same as the standard *other* file settings; no access.

Setacl and getfacl permission for User

```
[root@localhost ~]# mkdir /acl
[root@localhost ~]# getfa
getfacl    getfattr
[root@localhost ~]# getfacl /acl
getfacl: Removing leading '/' from absolute path names
# file: acl
# owner: root
# group: root
user::rwx
group::r-x
other::r-x

[root@localhost ~]# setfacl -m u:user1:rwx /acl
[root@localhost ~]# getfacl /acl
getfacl: Removing leading '/' from absolute path names
# file: acl
# owner: root
# group: root
user::rwx
user:user1:rwx
group::r-x
mask::rwx
other::r-x

[root@localhost ~]# su - user1
[user1@localhost ~]$ cd /acl
[user1@localhost acl]$ touch file
[user1@localhost acl]$ cd
[user1@localhost ~]$ logout
```

Setacl and getfacl permission for Group

```
[root@localhost ~]# setfacl -m group:acl:rwx /acl
[root@localhost ~]# getfacl /acl
getfacl: Removing leading '/' from absolute path names
# file: acl
# owner: root
# group: root
user::rwx
group::r-x
group:acl:rwx
mask::rwx
other::r-x
```

Remove permission for User & Group using -x

```
[root@localhost ~]# setfacl -x u:user1 /acl
[root@localhost ~]# getfacl /acl
getfacl: Removing leading '/' from absolute path names
# file: acl
# owner: root
# group: root
user::rwx
user:user2:rwx
group::r-x
mask::rwx
other::r-x
```

Set Permission using –R for subfiles and dir

```
[root@localhost ~]# setfacl -Rm u:user1:r-x /acl
[root@localhost ~]# getfacl /acl
getfacl: Removing leading '/' from absolute path names
# file: acl
# owner: root
# group: root
user::rwx
user:user1:r-x
user:user2:rwx
group::r-x
mask::rwx
other::r-x

[root@localhost ~]# su - user1
[user1@localhost ~]$ cd /acl
[user1@localhost acl]$ ll
total 0
-rw-r-xr--+ 1 user1 user1 0 Apr  3 22:11 file
```

Remove ACL permission for dir & file using -Rb

```
[root@localhost ~]# setfacl -b /acl
[root@localhost ~]# getfacl /acl
getfacl: Removing leading '/' from absolute path names
# file: acl
# owner: root
# group: root
user::rwx
group::r-x
other::r-x

[root@localhost ~]# ls -ld /acl
drwxr-xr-x. 2 root root 18 Apr  3 22:11 /acl
[root@localhost ~]# setfacl -Rb /acl
[root@localhost ~]# ls -ld /acl
drwxr-xr-x. 2 root root 18 Apr  3 22:11 /acl
[root@localhost ~]# cd /acl
[root@localhost acl]# ll
total 0
-rw-r--r--. 1 user1 user1 0 Apr  3 22:11 file
```

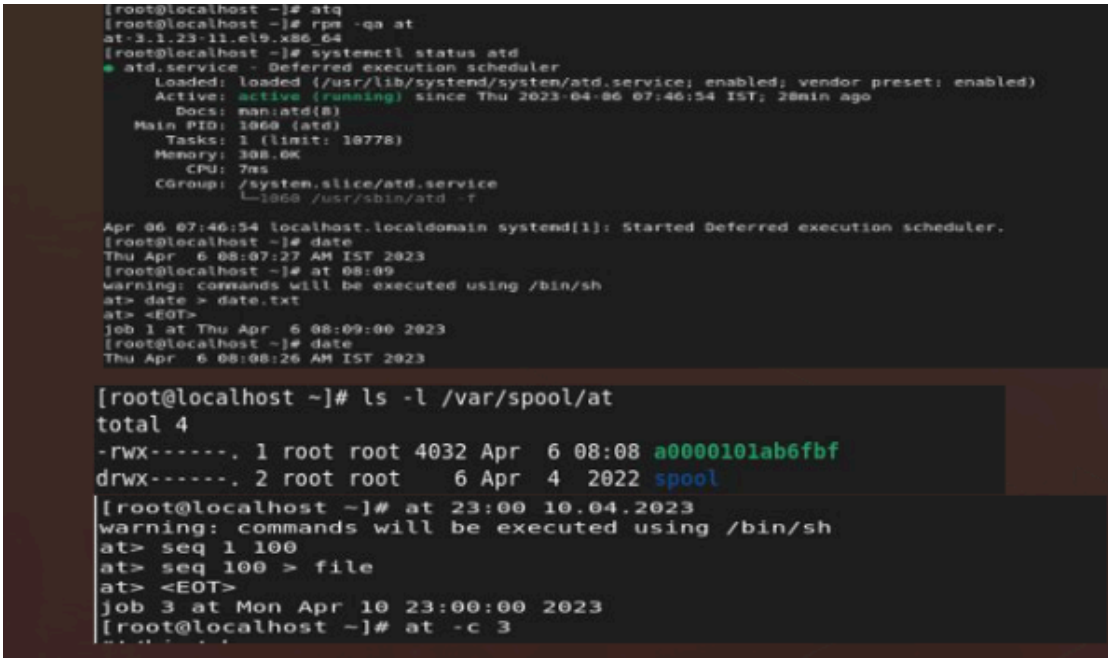
Scheduling a Deferred User Job

Scheduling Deferred User Tasks

Use the **`at`** ***`TIMESPEC`*** command to schedule a new job. The **`at`** command then reads the commands to execute from the **`stdin`** channel. While manually entering commands, you can finish your input by pressing **`Ctrl`**+**`D`**. For more complex commands that are prone to typographical errors, it is often easier to use input redirection from a script file, for example, **`at now +5min <`** ***`myscript`***, rather than typing all the commands manually in a terminal window.

The ***`TIMESPEC`*** argument with the **`at`** command accepts many powerful combinations, allowing users to describe exactly when a job should run. Typically, they start with a time, for example, **`02:00pm`**, **`15:59`**, or even **`teatime`**, followed by an optional date or number of days in the future. The following lists some examples of combinations that can be used.

- **`now +5min`**
- **`teatime tomorrow`** (teatime is **`16:00`**)
- **`noon +4 days`**
- **`5pm august 3 2021`**


[root@localhost ~]# atq
[root@localhost ~]# rpm -qa at
at-3.1.23-11.el9.x86_64
[root@localhost ~]# systemctl status atd
● atd.service - Deferred execution scheduler
Loaded: loaded (/usr/lib/systemd/system/atd.service; enabled; vendor preset: enabled)
Active: active (running) since Thu 2023-04-06 07:46:54 IST; 28min ago
Docs: man:atd(8)
Main PID: 1060 (atd)
Tasks: 1 (limit: 10778)
Memory: 308.0K
CPU: 7ms
CGroup: /system.slice/atd.service
└─1060 /usr/sbin/atd -f
Apr 06 07:46:54 localhost.localdomain systemd[1]: Started Deferred execution scheduler.
[root@localhost ~]# date
Thu Apr 6 08:07:27 AM IST 2023
[root@localhost ~]# at 08:09
warning: commands will be executed using /bin/sh
at> date > date.txt
at> <EOT>
job 1 at Thu Apr 6 08:09:00 2023
[root@localhost ~]# date
Thu Apr 6 08:08:26 AM IST 2023
[root@localhost ~]# ls -l /var/spool/at
total 4
-rwx------. 1 root root 4032 Apr 6 08:08 a0000101ab6fbf
drwx------. 2 root root 6 Apr 4 2022 spool
[root@localhost ~]# at 23:00 10.04.2023
warning: commands will be executed using /bin/sh
at> seq 1 100
at> seq 100 > file
at> <EOT>
job 3 at Mon Apr 10 23:00:00 2023
[root@localhost ~]# at -c 3

Create AT job for different user

```
[root@localhost ~]# su - user1
[user1@localhost ~]$ at 10:00
warning: commands will be executed using /bin/sh
at> date > date.txt
at> <EOT>
job 2 at Thu Apr  6 10:00:00 2023
[user1@localhost ~]$ ls -l /var/spool/at
ls: cannot open directory '/var/spool/at': Permission denied
[user1@localhost ~]$ exit
logout
[root@localhost ~]# cat /var/spool/at
cat: /var/spool/at: Is a directory
[root@localhost ~]# cat /var/spool/at/
cat: /var/spool/at/: Is a directory
[root@localhost ~]# ls -l /var/spool/at/
total 8
-rwx------. 1 root  root  4032 Apr  6 08:44 a0000101b1648e
-rwx------. 1 user1 user1 2824 Apr  6 08:46 a0000201ab702e
drwx------. 2 root  root     6 Apr  6 08:11 spool
```

```
[root@localhost ~]# ls -l /etc/at.deny
-rw-r--r--. 1 root root 1 Apr  4  2022 /etc/at.deny
[root@localhost ~]# ls -l /etc/at.allow
ls: cannot access '/etc/at.allow': No such file or directory
[root@localhost ~]# vi  /etc/at.allow
[root@localhost ~]# su - user1
[user1@localhost ~]$ at 10:00
You do not have permission to use at.
[user1@localhost ~]$ logout
[root@localhost ~]# vi  /etc/at.allow
[root@localhost ~]# su - user1
[user1@localhost ~]$ at 10:00
warning: commands will be executed using /bin/sh
at> date > date.txt
at> ^C[user1@localhost ~]$
logout
[root@localhost ~]# atq
1       Tue Jan  2 10:00:00 2024 a root
2       Thu Apr  6 10:00:00 2023 a user1
```

Commands for scheduling Crontab jobs

```
[root@localhost ~]# cat /etc/cront.ab
cat: /etc/cront.ab: No such file or directory
[root@localhost ~]# cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed

[root@localhost ~]# ls -l /var/spool/cron/
total 0
[root@localhost ~]# crontab -e
no crontab for root - using an empty one
```

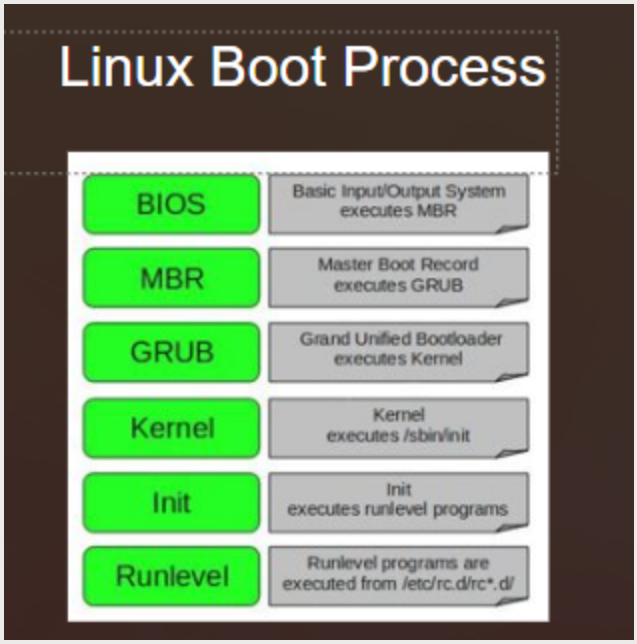
```
[root@localhost ~]# crontab -e
crontab: installing new crontab
[root@localhost ~]# crontab -l
*/5     *       *       *       *       date >> date.txt
*/15    *       *       *       *       cal > cal.txt
00      9-18    *       *       *       date > date1
00      *       31      *       *       seq 1 10 > file
00      *       31      12      *       echo hello > text
00      0       *       *       1-5     ls -l /root > root.txt
```

```
[root@localhost ~]# ls -l /var/spool/cron/root
-rw-------. 1 root root 179 Apr  6 09:26 /var/spool/cron/root
[root@localhost ~]# cat /var/spool/cron/root
*/1     *       *       *       *       date >> date.txt
*/15    *       *       *       *       cal > cal.txt
00      9-18    *       *       *       date > date1
00      *       31      *       *       seq 1 10 > file
00      *       31      12      *       echo hello > text
00      0       *       *       1-5     ls -l /root > root.txt
```

Create AT job for different user, remove, and allow normal user

```
[root@localhost ~]# crontab -eu user1
no crontab for user1 - using an empty one
crontab: no changes made to crontab
[root@localhost ~]# su - user1
[user1@localhost ~]$ crontab -e
no crontab for user1 - using an empty one
crontab: installing new crontab
[user1@localhost ~]$ crontab -l
*/1     *       *       *       *       date > date.txt
[user1@localhost ~]$ exit
```

```
[root@localhost ~]# crontab -lu user1
*/1     *       *       *       *       date > date.txt
[root@localhost ~]# crontab -ru user1
'[root@localhost ~]# '^C
[root@localhost ~]# crontab -lu user1
no crontab for user1
[root@localhost ~]# ls -l /etc/crontab.allow
ls: cannot access '/etc/crontab.allow': No such file or directory
[root@localhost ~]# vi /etc/crontab.allow
[root@localhost ~]# vi /etc/crontab.allow
```



https://operavps.com/boot-process-in-linux-in-6-levels/

https://www.geeksforgeeks.org/run-levels-linux/

Linux Basic commands:

Input/Output Redirection:

< — Input
> — Output
0 — Input
1 —- Output
2 —- Error

1 —- Output

```
[root@raghav newdir]# ls 1>out 2>error
[root@raghav newdir]# cat out
error
new1
out
```

2 —- Error

```
[root@raghav newdir]# ls sample 1>out 2>error
[root@raghav newdir]# cat error
ls: cannot access 'sample': No such file or directory
```

Find command:

```
706  find / -type d -name perm
707  find / -type - -name
708  find / -type f -name test
709  find / -user
710  find / -user user1
711  find / -group shanmug
712  find / -group user1
713  find /var/log -size 5mb
714  find /var/log -size 5MB
715  find /var/log -size +5MB
716  find /var/log -size 5M
717  find /var/log -size 10M
718  find /var/log -size 25M
719  find /var/log -size +25M
720  find /var/log -size +5M
721  find / -size +5M
722  find -type f -empty
723  find / -perm 1777
724  ll
725  cd newdir
726  ll
727  mkdir new1
728  find . -perm 755
```