Storage Management

Unit objectives

After completing this unit, you should be able to:

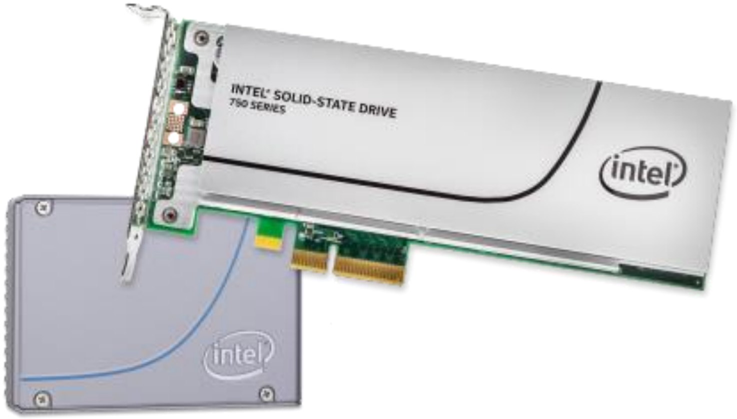
- Review of modern internal hard disk type
- Device naming convention
- Perform Disk partitioning
- Understand what and why need File System
- Create and Mount File System
- Unmount File System

Overview of modern internal hard disk type - 1/2

- Most common device for persistent storage
- Three common types: SATA, SAS, and NVMe
- Integrated Drive Electronics (IDE) - Legacy
  - Some cabling limitations
  - Device naming /dev/hda, hdb, ..., hdh
- Serial Advanced Technology Attachment (SATA)
  - Offers some advantages over ATA/EIDE:
    - Hot swappable, faster transfer speed, smaller cabling
- Small Computer System Interface (SCSI) - Legacy
  - Different subtypes: fast, wide, fast wide, ultra-wide, ...
  - Max 7 or 15 disks on one bus (depends on subtype)
  - Needs correct termination at both ends of bus
  - Generally more expensive
  - Device naming /dev/sda, ..., sdz, sda1, ..., sddx

Overview of modern internal hard disk type – 2/2

- Serial Attached SCSI (SAS)
  - offer better disk speed (10,000 / 15,000 rpm)
  - interface speed of upto 22.5 GB/s
  - Max upto 65,535 disks
- Non-Volatile Memory express (NVMe)
  - interface speed of upto 32 GB/s
  - offers high IOPS, low latency (<=20 microseconds)
  - Two varianets : M.2 and U.2



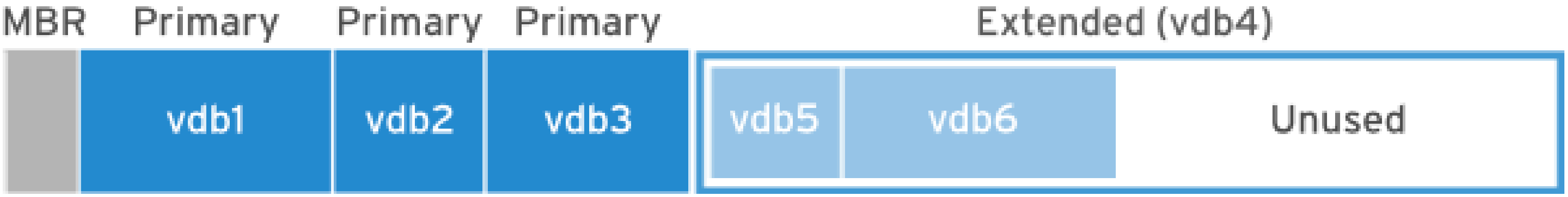
Devices naming convention

- All character devices have a representation in `/dev`

```
# ls -l /dev/{sd,nvme}*
crw-------. 1 root root 243,  0 Sep 27 07:32 /dev/nvme0
brw-rw----. 1 root disk 259,  0 Sep 27 07:32 /dev/nvme0n1
brw-rw----. 1 root disk 259,  1 Sep 27 07:32 /dev/nvme0n1p1
brw-rw----. 1 root disk 259,  2 Sep 27 13:35 /dev/nvme0n1p2
brw-rw----. 1 root disk   8,  0 Sep 27 13:35 /dev/sda
brw-rw----. 1 root disk   8, 16 Sep 27 13:35 /dev/sdb
brw-rw----. 1 root disk   8, 32 Sep 27 13:34 /dev/sdc
brw-rw----. 1 root disk   8, 48 Sep 27 07:32 /dev/sdd
brw-rw----. 1 root disk   8, 64 Sep 27 07:32 /dev/sde
```

Disk partitioning principles : MBR / DOS

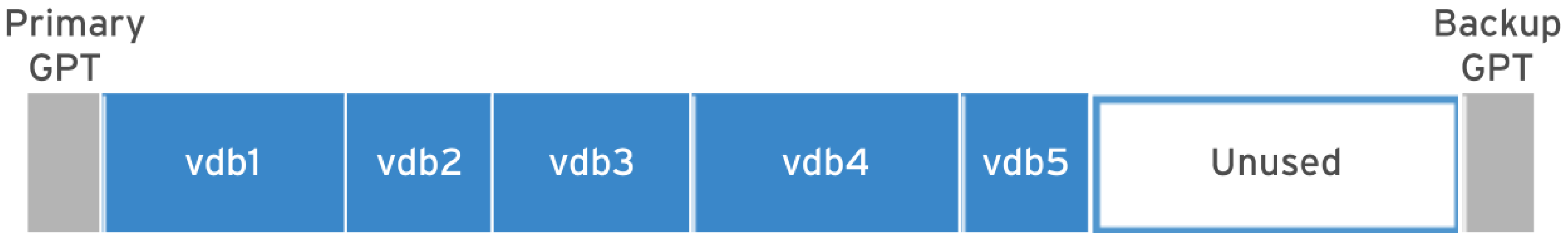
- Four primary partitions, one of which might be an "extended" partition
  - /dev/hda1 - /dev/hda4
- Extended partition: Large number of logical partitions
  - /dev/hda5 and up
- Limitation
  - to 2TB
  - 15 logical partitions



Disk partitioning principles → GPT

- GUID Partition Table
- UEFI firmware standard
- Address limitation in MBR-based scheme
- Encouraged to use!

- Cool stuff
  - max of 128 partitions
  - ZiB or 8 billion TB
  - GUID to identify each disk & partition
  - Redundancy of partition information



Partitioning tools

- fdisk
  - Virtually every PC OS comes with a tool fdisk to create partitions for that OS.
    - Linux, Windows, and so forth
- parted
  - GPLed Linux program, available at www.gnu.org
  - Can create/resize/move/delete partitions
- GParted, QTParted
  - GUI utilities that use GNU Parted
  - Can create/resize/move/delete partitions
- Disk Druid and others
  - Partitioning program integrated in Linux install program
- Cockpit web-based interface

The fdisk tool

View disk configuration

```
# fdisk –l /dev/sda
```

Configure disk's partition

```
# fdisk /dev/sda
```

Display current partition

```
Command (m for help): p
```

Create new partition

```
Command (m for help): n
```

Create delete partition

```
Command (m for help): d
```

The parted tool - interactively

View disk configuration

```
# parted /dev/sda print
```

Configure disk's partition

```
# parted /dev/sda
```

Decide on partition type

```
(parted) mklabel { gpt | msdos }
```

Display current partition

```
(parted) print
```

Create new partition

```
(parted) mkpart
```

Create delete partition

```
(parted) rm
```

Rescue lost partition near start and end

```
(parted) rescue
```

Resize partition

```
(parted) resizepart
```

The parted tool – non-interactively

- Convert disk into GPT

```
# parted /dev/sda mklabel gpt
```

- View disk configuration

```
# parted /dev/sda print
```

- Create first partition

```
# parted /dev/sda mkpart 1 xfs 1 100m
```

- Create next partition

```
# parted /dev/sda mkpart 2 ext4 101m 500m
```

- Create third partition

```
# parted /dev/sda mkpart 3 xfs 501 1g
```

The parted tool – non-interactively

- Delete partition number 2

```
# parted /dev/sda rm 2
```

- Resize parition 3 to 1GB

```
# parted /dev/sda resizepart 3 1g
```

- Make sure kernel detect new partitions and create necessary devices in /dev directory

```
# udevadm settle
```

The parted tool – manage MBR partitions

- Convert disk into MBR

```
# parted /dev/sda mklabel msdos      Warning: This will
destroy data
```

- Resize parition 3 to 1GB

```
# parted /dev/sda resizepart 3 1g
```

Shortcuts to parititioning

- Copy partitions

```
# sfdisk -d /dev/sda | sfdisk /dev/sdb
```

- Quickly delete all partitions in /dev/sda

```
# dd if=/dev/zero of=/dev/sda bs=512 count=1
```

Verify all configured partitions

- Using fdisk

```
# fdisk –l /dev/sda
```

- Using parted

```
# parted /dev/sda print
```

- Using lsblk

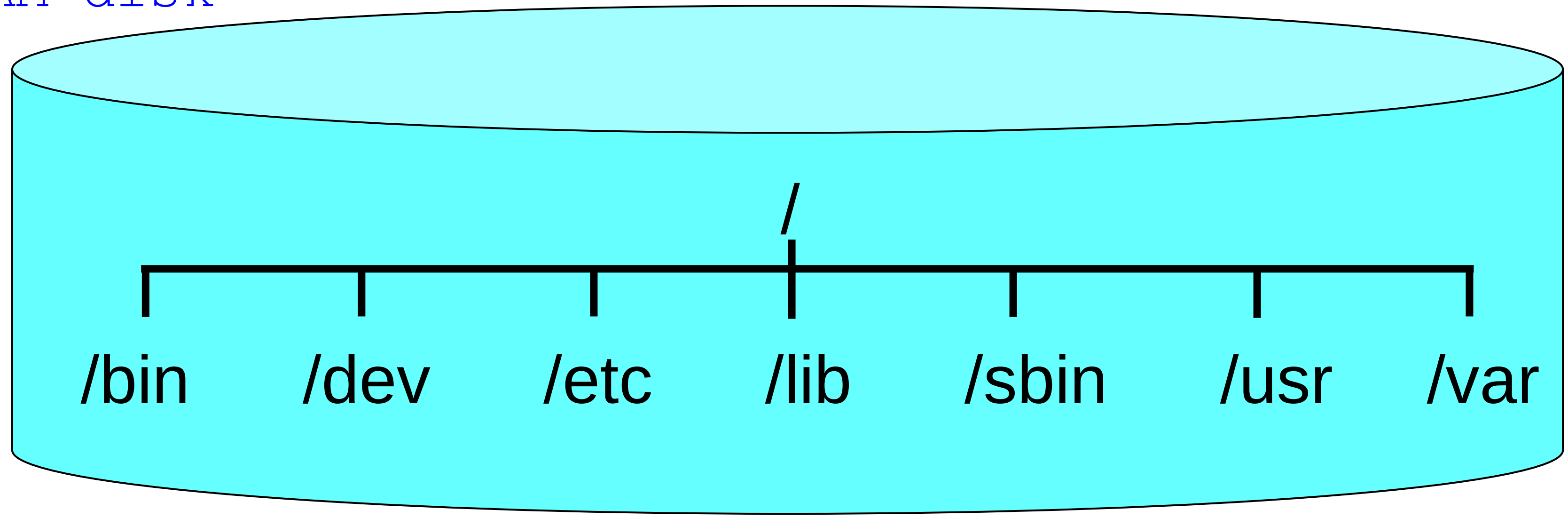
```
# lsblk
```

- Using blkid

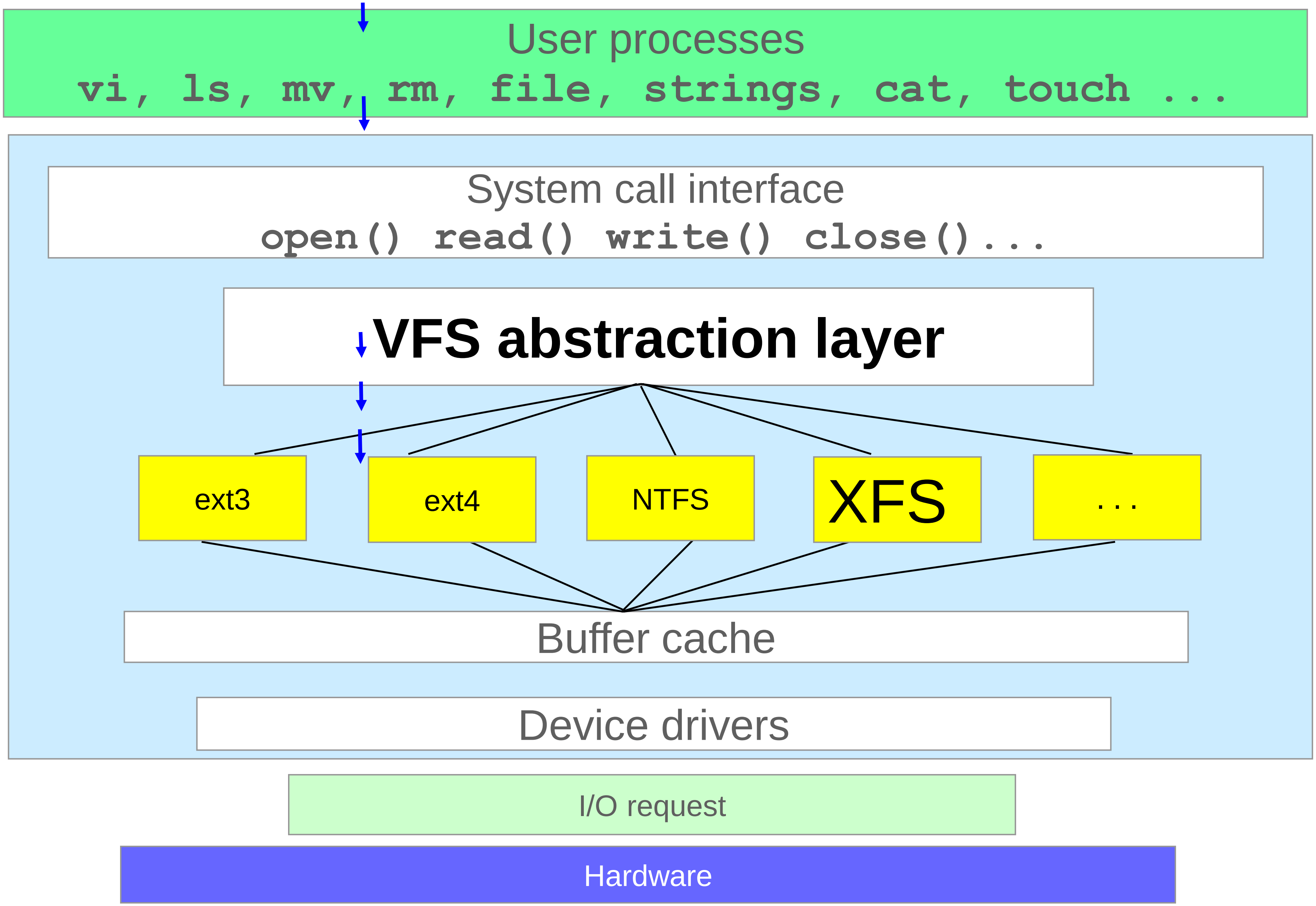
```
# blkid
```

Why need file system?

- Place to store files and refer to them
- Hierarchical structure through use of directories
- A file system can be stored on any block device
  - Floppy disk
  - Hard disk
  - Partition
  - RAID, LVM volume
  - File (for use with a loop device)
  - RAM disk



The virtual file system



© Copyright Trainocate (M) 2020

Format partition with File System

- Traditional: ext2
- Second generation: ext3, ReiserFS, IBM JFS, xfs
- Next generation: ext4, GFS2, Reiser4
- FAT-12, FAT-16, FAT-32, VFAT, NTFS (read-only)
- CD-ROM (ISO 9660)
- NFS (Network File System)
- SMBFS (Windows share), NCPFS (Novell NetWare share)
- `/proc` (for kernel and process information)
- SHMFS (shared memory file system)
- GPFS, Lustre (clustering file systems)
- XFS (high performing 64-bit journaling FS)

Creating file system

- Creating a file system is done with a `mkfs` variant.
  - `mkfs.ext2, mkfs.ext3, mkfs.xfs`
  - `mkfs.msdos, mkfs.fat`
  - `mkfs -t`
- Typical options:
  - `b  blocksize sets blocksize`
  - `i  bytes per i-node sets number of i-nodes`
  - `c  checks disk for bad blocks`
- Example:

```
# mkfs.xfs /dev/sda1
...
Writing inode tables: done
Writing superblocks and filesystem accounting info:
done
...
```

Mounting file system

- Using the `mount` command:
  - Supply device name
  - Supply mount point (empty directory)

```
# mount -t xfs /dev/sda1 /mnt/extra
# mount -o ro,remount /dev/sda1 /mnt/extra
```

Optional: Supply file system type

Optional: Supply other options

Optional: Use different superblock

- To show mounted file systems, use **`mount`** without arguments.

Mounting file systems at system startup

- Add to `/etc/fstab`:

```
/dev/mapper/cl_base-root  /    xfs  defaults          2 1
/dev/mapper/cl_base-swap  swap swap defaults          1 1
/dev/cdrom  /mnt/cdrom    iso9660   noauto,ro,user    0 0
/dev/fd0    /mnt/floppy   msdos     noauto,user       0 0
/dev/sda6   /mnt/extra    ext3      defaults          0 0
```

- Preferrably, using `UUID`:

```
/dev/mapper/cl_base-root /                               xfs     defaults     0 0
UUID=2f209047-45fc-43c3-8ab1-8d270ed12b3c /boot  ext4    defaults     1 2
/dev/mapper/cl_base-swap swap                            swap   defaults     0 0
```

Verify all mounted partitions

- Using commands

```
# mount
# df [ -k | -m | -h ]
# lsblk --fs
```

- Using Files

```
# more /etc/mtab
# more /proc/mounts
```

Unmounting a file system

- Using the `umount` command:
  - disconnect device
  - changing mount options
  - when shutting, temporarily denied access to user

```
# umount /dev/sda1
# umount /mnt/extra
```

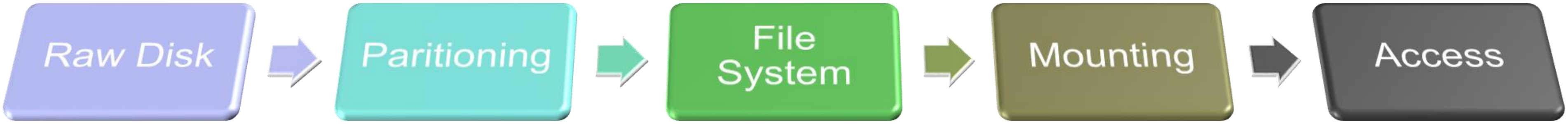
- Unmount all non-os file systems

```
# umount -a
```

- If user still connected

```
# fuser -cu                          <- Inform user
# fuser -ck                          <- Kick user out
# umount -f /mnt
```

The Sequence



- Raw disks : sd | hd | nvme | san luns
- Partitioning : fdisk | parted | gparted
- FS : mkfs | mkfs.xfs | mkfs.ext4
- Mounting : mount /dev/vg/lv /mnt
- Access : cd | touch | mkdir | more | vi | cat ...

Quiz

1. Mount 2nd partition in /dev/sdf to /mnt/data using xfs fs type.

   a) mount xfs /dev/sd2 /mnt/data

   b) mount –t xfs /dev/sd2 /mnt/data

   c) mount xfs /dev/sdf2 /mnt/data

   d) mount –t xfs /dev/sdf2 /mnt/data

Quiz

1. Mount 2nd partition in /dev/sdf to /mnt/data using xfs fs type.
   a) mount xfs /dev/sd2 /mnt/data
   b) mount –t xfs /dev/sd2 /mnt/data
   c) mount xfs /dev/sdf2 /mnt/data
   d) mount –t xfs /dev/sdf2 /mnt/data

Quiz

2. You are currently in /mnt/data, and you need to un mount the file system.

   a) umount –f /mnt/data
   b) kill <pid>; umount /mnt/data
   c) cd; unmount /mnt/data
   d) logout then unmount by umount /mntdata cmd

Quiz

2. You are currently in /mnt/data, and you need to un mount the file system.

   a) umount –f /mnt/data
   b) kill <pid>; umount /mnt/data
   c) cd; unmount /mnt/data
   d) logout then unmount by umount /mntdata cmd

Quiz

3. Why do you need file system?

a) To provide interface to the hardware

b) To provide access to storage

c) To provide interface to the operating system

d) To provide access to mount point

Quiz

3. Why do you need file system?

a) To provide interface to the hardware

b) To provide access to storage

c) To provide interface to the operating system

d) To provide access to mount point

Unit summary

Having completed this unit, you should be able to:

- Understand modern internal hard disk type
- Understand Device naming convention
- Perform Disk partitioning
- Understand what and why need File System
- Create and Mount File System
- Unmount File System