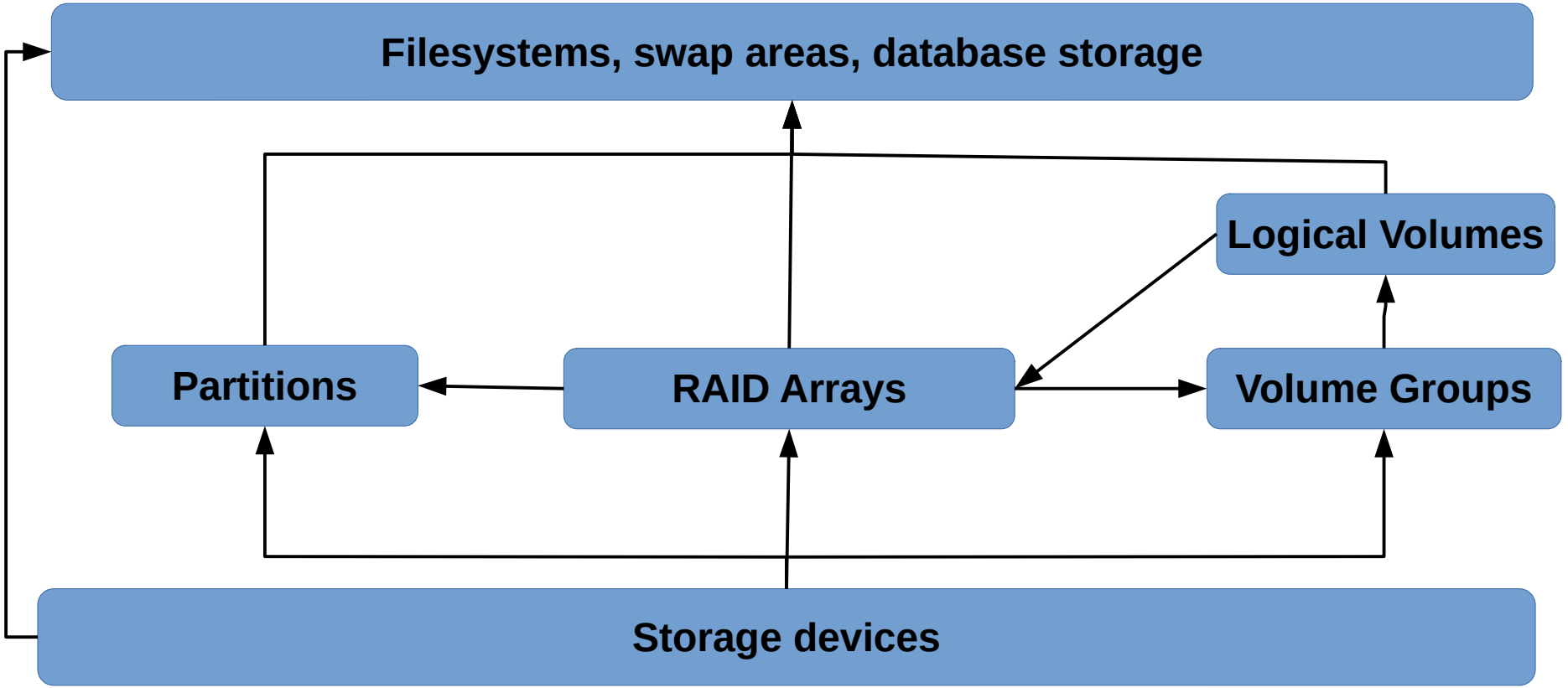
Filesystems

Linux System Administration
Fall 2022

Filesystems

- On Unix/Linux, everything is a file
- Files are organized in a directory tree
- A file system is what stores files
  - File systems integrate into the directory tree
- A disk is a physical storage device
  - Storage blocks are called sectors
  - Disk can be divided into partitions
  - Disks can be grouped in RAID arrays

Storage Management

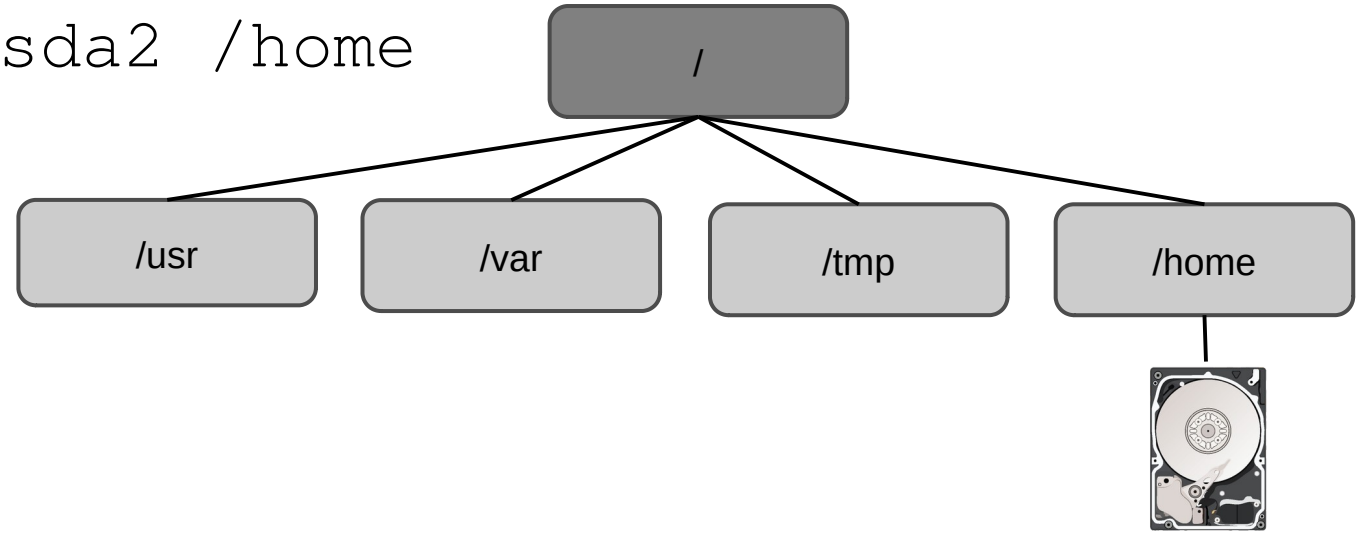


Useful filesystem separation

- `/boot` separate partition for grub
  - `/boot/efi` separate partion for grub/efi
- `/tmp` or `/scratch` temporary storage
- `/home` user direcories
- `/var` contains cache, mail, run, etc
- `/var/log` contains system logs
- `/var/lib/mysql` database storage

Mounting filesystems

- `mount device dir`
  - The mount point must be an existing directory
  - The file system becomes part of the tree
    - Existing sub-tree becomes inaccessible
  - `mount /dev/sda2 /home`



Unix file system concepts

- Derived form the original Fast File System (FFS)
  - Disks are block devices
  - Directory tree structure
  - Directory entries point to inodes
  - inodes contain file characteristics
    - Type, permissions, times, etc.
    - Pointers to data blocks & indirection
  - Superblocks, cylinder groups, fragments
- Journaling added much later

File system features to consider

- Stability & robustness
  - Detecting bit rot
  - Safe handling of power or system failure
- Ability to store very large and very small files
- Performance
  - Read and write speed
  - Recovery time after failure
- Snapshots, replication, compression, encryption, etc.

Popular file systems

- `UFS` Unix File system (FFS)
- `ext2/ext3/ext4` Extended File System
  - `ext3` added journaling (backwards compatible)
- `XFS` extent based file system
- `ZFS` incorporates RAID and volume manager
- `BtrFS` B-tree file system("ZFS lite")
- `tmpfs` volatile memory file system
- `*FAT*` **F**ile **A**llocation **T**able (1977 floppy disks - current USB)

File systems can be mixed

- `vfat` for `/boot/efi` to simplify booting
- `ext4` for `/` good for general purpose use
- `xfs` for `/home` to store large media files
- `BrtFS` for `/var/lib/mysql` for snapshots
- `NFS4` for `/var/mail` network shared storage
- File system type and mount point are seamless
  - `df -T` or `cat /etc/fstab` shows the detail

Device Naming

- `/dev/hdX` Legacy hard drives
- `/dev/sdX` SCSI (SATA, SAS, USB) drives
- `/dev/srX` CD/DVD drives
- `X` is in the order of discovery `a`, `b`, `c`, `d`, ..
- Paritions are appended numerically
  - `/dev/sdc1` First partition (`1`) on third drive (`c`)

Persistent file names

- UUID/GUID **U**niversally/**G**lobally **U**nique **ID**
  - Stored on disk in file system superblock
  - `/dev/disk/by-uuid/*` symbolic link to `/dev/*`
- `blkid` shows UUID and other information
- UUID can be referenced in `/etc/fstab`

/etc/fstab

- Disks to mount on boot (or `mount -a`)
  - **`device mount type options backup check`**
  - `/dev/sda1 / ext4 errors=remount-ro 0 1`
  - **`device`** name of the device
    - `/dev/sda1 UUID=4c3f0e41-9bec-449c-8c57-95b5a4cb9927 /dev/mapper/root`
  - **`mount point`** where this go in the directory tree
    - `/` or `/home` or `/var/lib/mysql`
  - **`type`** file system type
    - `ext4` or `xfs` or `tmpfs` or `nfs4` or `swap`
  - **`options`** mount options
    - `errors=` or `ro` or `rw` or `sw` and may others
  - **`backup`** (obsolete - used by dump - set to zero)
  - **`check`** file system check order
    - `0`=no fsck, `1`=fsck first (file root), `2`=fsck next

Filesystem commands

- `mount` mount a filesystem in the tree
- `umount` unmount a filesystem
- `fsck` file system check
  - Generic front end for for `fsck.ext3`, `fsck.xfs`, etc
- `mkfs` create a new file system (**D.F.I.U.**)
  - Generic front end for `mkfs.ext3`, `mkfs.xfs`, etc
- `badblocks` check drive for bad blocks (low level)

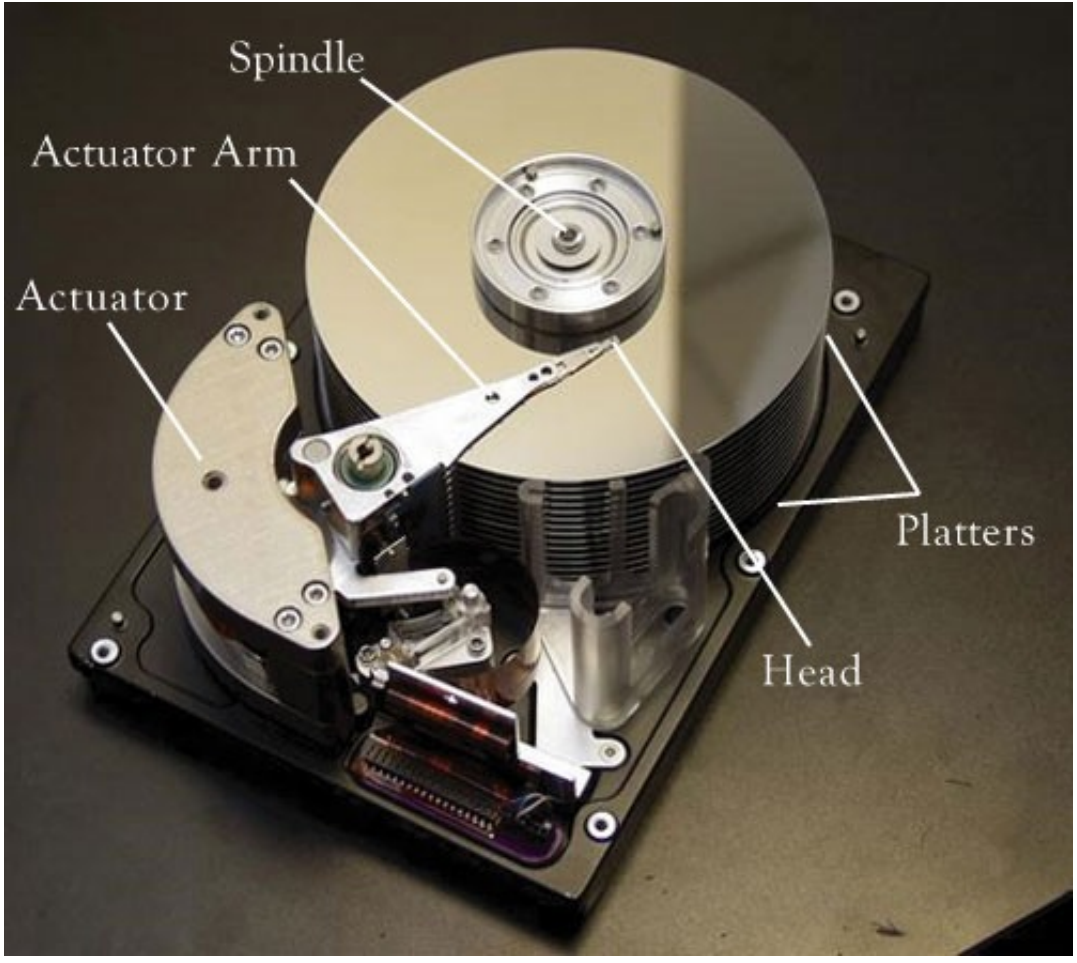
Filesystem specific commands

- ext2/ext3/ext4
  - `tune2fs`
    - set options
    - manage quotas
  - `resize2fs`
    - shrink or expand
  - `dumpe2fs`
    - show filesystem data
- xfs
  - `xfs_admin`
    - set options
  - `xfs_growfs`
    - expand
  - `xfs_info`
    - show filesystem data
  - `xfs_quota`
    - manage quotas

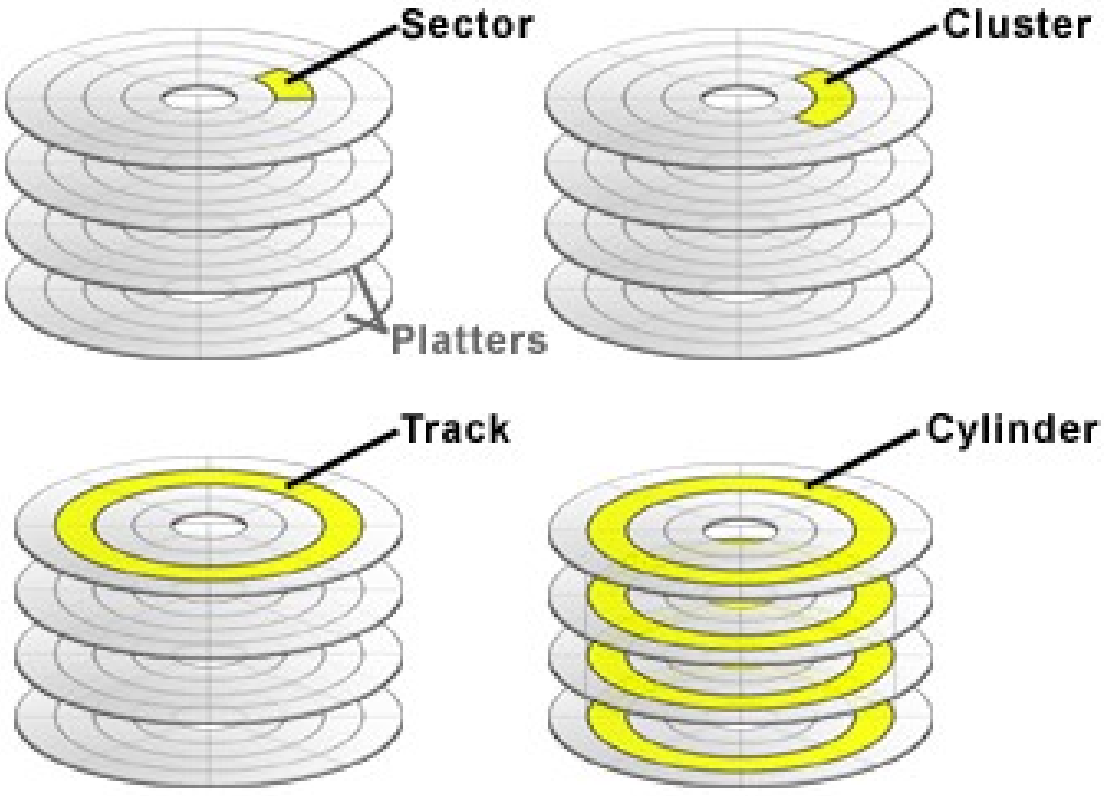
Disk drive types

- Mechanical
  - Slower 4-15 ms
  - Larger capacity
  - Less exensive
  - Sequential access must faster than random
  - Unlimited rewrites
  - Failure often predictable
- Solid State
  - Faster 0.1 ms
  - Lesser capacity
  - More expensive
  - Fast access regardless of pattern
  - Finite rewrites
  - Often fails without warning

Mechanical Hard Drives

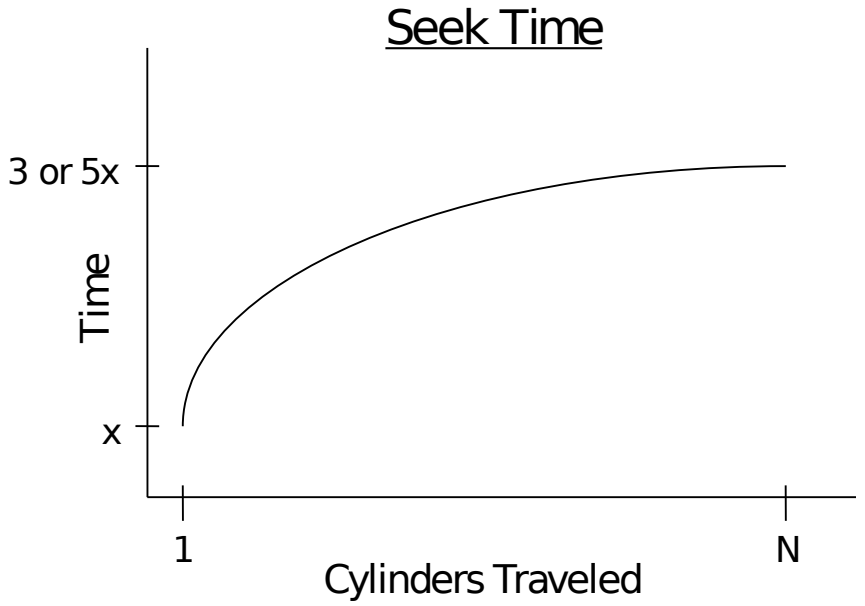


Mechanical Hard Drive Organization



Mechanical Hard Drive Properties

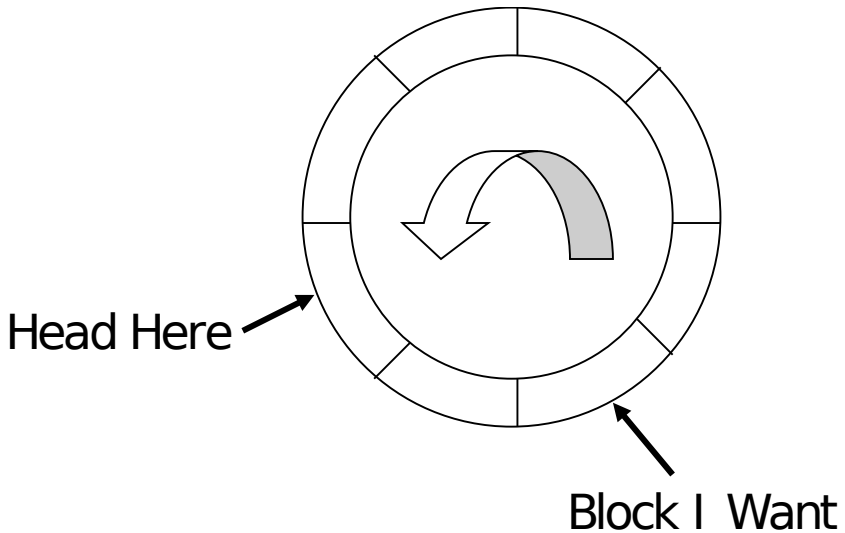
- Access time factors
  - Seek time
    - Acceleration
    - Transit
    - Deceleration
    - Settling
  - Rotational delay
  - Transfer time



Disk scheduling algorithms

- First come, first served
- Shortest Seek Time First
- SCAN (elevator algorithm)
- C-SCAN (one way elevator)
- C-LOOK (stops at first/last requested cylinder)

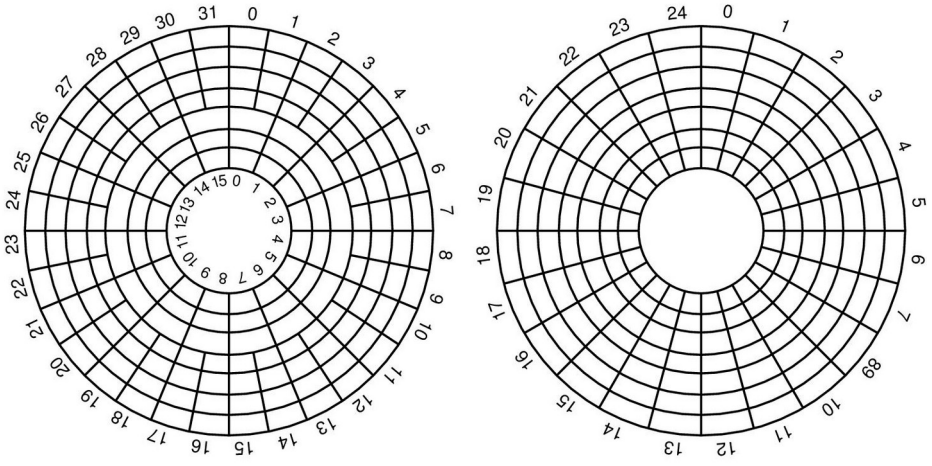
Rotational Delay



HDD Spindle [rpm]	Average rotational latency [ms]
4,200	7.14
5,400	5.56
7,200	4.17
10,000	3.00
15,000	2.00

Transfer Rate

- Transfer rate
  - Speed of media
  - Linear density
  - Faster on the outside
    - Potentially more sectors per track
  - Track total/rotation



Disk Interface Alphabet Soup

- Legacy Interfaces
  - IDE (ATA, PATA) *Integrated Drive Electronics*
  - SCSI *Small Computer System Interface*
- Current Interfaces
  - SATA *Serial ATA* [1.5-6.0 Gb/s]
  - SAS *Serial Attached SCSI* [3-22.5 Gb/s]
  - PCIe (NVMe) *Peripheral Component Interconnect Express*
  - USB *Universal Serial Bus*

Overall performance

- Access time = seek + rotation + transfer
- Write performance may be slower for SSD
  - Erase before write
- Caching
- Read-ahead
- RAID

Disk failures

Why you should have good backups

- MTBF ***M****ean* ***T****ime* ***B****etween* ***F****ailure*
- S.M.A.R.T ***S****elf-****M****onitoring,* ***A****nalysis and* ***R****eporting* ***T****echnology*
- Bad blocks, bad cells and self-healing
  - Low level formatting
- Drive types
  - Value - big and slow
  - Mass-market - fast and hot
  - NAS - robust and cool
  - Enterprise - fast and expensive

Disk Partitions

- Subdivides a disk (or storage system)
  - Presents just like a disk to the file system
    - Simplifies backups
    - Provides a degree of protection
  - Difficult to change later
- Stored on the disk
  - MBR **M**aster **B**oot **R**ecord
  - GPT **G**lobally Unique ID **P**artition **T**able

Managing partitions

- Partition programs
  - fdisk (text UI)
    - text UI
  - parted
    - text UI
    - better for fixing stuff
  - gparted
    - GUI version
- DFIU
  - Make sure you are in the right disk!!!!!!!!!!
  - Moving partitions can destroy file systems
  - Expanding OK, shrinking typically not

Partition Table

- Partition properties
  - Primary/Secondary
  - Bootable flag
  - Start and end sectors
  - Size (sectors)
  - Type
- Partition types (hex)
  - 82 Linux swap
  - 83 Linux
  - 8e Linux LVM
  - a5 FreeBSD
  - fd Linux RAID
  - 86 NTFS

Machine A Partition Table

- **# fdisk -l /dev/sda**

```
Disk /dev/sda: 16 GiB, 17179869184 bytes, 33554432 sectors
Disk model: Virtual disk
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: gpt
Disk identifier: 4329DE0D-371A-4EC7-A2BF-154D2CEF1154

Device       Start      End  Sectors  Size Type
/dev/sda1     2048  1230847  1228800  600M EFI System
/dev/sda2  1230848  3327999  2097152    1G Linux filesystem
/dev/sda3  3328000 33552383 30224384 14.4G Linux LVM
```

 # df -kh

```
Filesystem                         Size  Used Avail Use% Mounted on
/dev/sda1                          599M  6.9M  592M   2% /boot/efi
/dev/sda2                         1014M  159M  856M  16% /boot
/dev/mapper/rl_itd--74--196-root    13G  3.3G  9.6G  26% /
```

RAID

- **R**edundant **A**rray of **I**nexpensive **D**isks
  - Hardware chips or entire subsystems
  - Software RAID kernel module (`md`)
- RAID concepts
  - **Stripe** spread data across devices to improve throughput
  - **Mirror** save redundant information to survive drive failures

Types of RAID

- `JBOD` **J**ust a **B**unch **O**f **D**isks
- `0` Stripes data across 2 disks
- `1` Mirrors data to 2 disks
- `5` Stripe and mirror, survive 1 lost disk (min 3)
- `6` Stripe and mirror, survive 2 lost disks (min 4)
- `0+1` Mirror of stripes (4 disks)
- `1+0` Stripe of mirrors (4 disks)

RAID considerations

- **RAID does not reduce the need for backups**
- Hot spares are nice, but reduces capacity
- RAID5 many writes required hurt performance
- RAID5 vulnerable to desync on power fail
  - Battery backup improves robustness+performance
- Software RAID is inexpensive but slow

Linux Software RAID

- **md** (**m**ultiple **d**isks)
- Devices are physical disks or partitions
- Managed by `mdadm`
- Creating an array

```
mdadm --create /dev/md/name --raid-devices=3
    --level=5 /dev/sdf1 /dev/sdg1 /dev/sdh1
```

- Making the array persistent

```
mdadm --verbose --detail --scan>/etc/mdadm.conf
```

Logical Volume Manager (LVM)

- Physical Volume
  - Disk drives, partititions or RAID subsystems
- Volume Group
  - Collection of physical volumes
- Logical Volume
  - Logical device (partition within volume group)

LVM Pros and Cons

- Software RAID
  - More flexible, but slower than hardware
- Logical volumes can be readily resized
  - Readily add more physical volumes
    - More disks or partitions
  - Most filesystems can expand, some can shrink
- Snapshots

LVM Commands

- Physical Volume
  - `pvcreate`
  - `pvdisplay`
  - `pvchange`
  - `pvck`
  - `pvremove`
  - `pvresize`
- Volume Group
  - `vgcreate`
  - `vgdisplay`
  - `vgchange`
  - `vgck`
  - `vgremove`
  - `vgextend`
  - `vgscan`
- Logical Volume
  - `lvcreate`
  - `lvdisplay`
  - `lvchange`
  - `lvscan`
  - `lvremove`
  - `lvextend`
  - `lvscan`

Simple LVM Recipe

- Label device(s)
  - `pvcreate /dev/sdb1`
- Create volume group
  - `vgcreate SA /dev/sdb1`
- Create logical volumes
  - `lvcreate -n tmp   -L 1G SA`
  - `lvcreate -n mysql -L 4G SA`
- Make file systems
  - `mkfs -t xfs /dev/mapper/SA-tmp`
  - `mkfs -t xfs /dev/mapper/SA-mysql`
- Mount filesystems
  - `mount /dev/mapper/SA-tmp   /tmp`
  - `mount /dev/mapper/SA-mysql /var/lib/mysql`

swap

- Disk blocks that can be used to swap memory blocks to in virtual memory systems
  - Can be useful for large occasional memory hogs
  - Heavily using swap kills performance
- Linux swap commands
  - `mkswap` set up device for swap
  - `swapon` show or enable swap
  - `swapoff` disable swap

Network File System

- NFS is the native Unix Network File System
  - NFSv4 is the current versions
- Allows a server to export a subtree to clients
- Clients mount the subtree like a drive
  - `mount emc:/vol1 /home`
- Automount can mount or unmount on demand