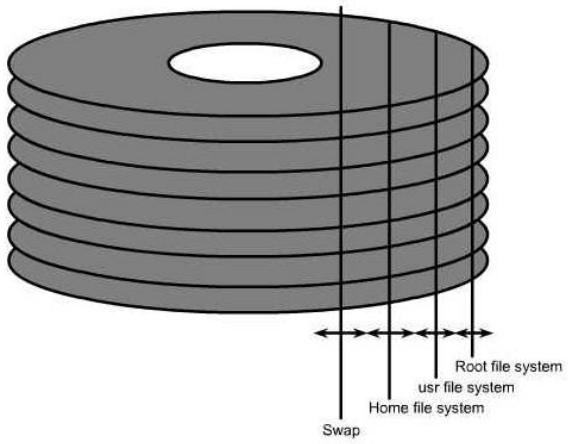
Chapter 18: Partitioning

Understanding Partitioning

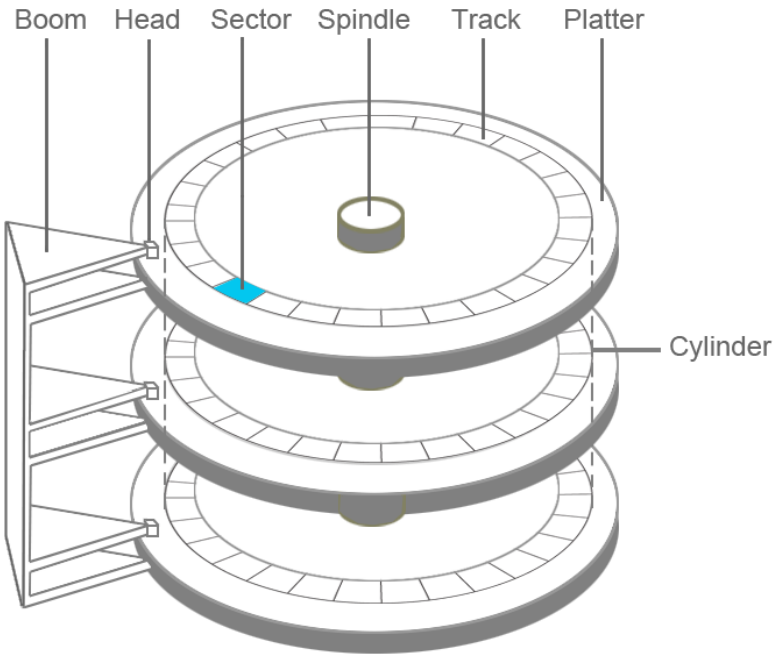
- Optimizes the use of disk space
- Three steps in making and using partitions:
  - Divide the disk into partitions
  - Format the partition with a filesystem
  - Mount the filesystem onto the directory tree



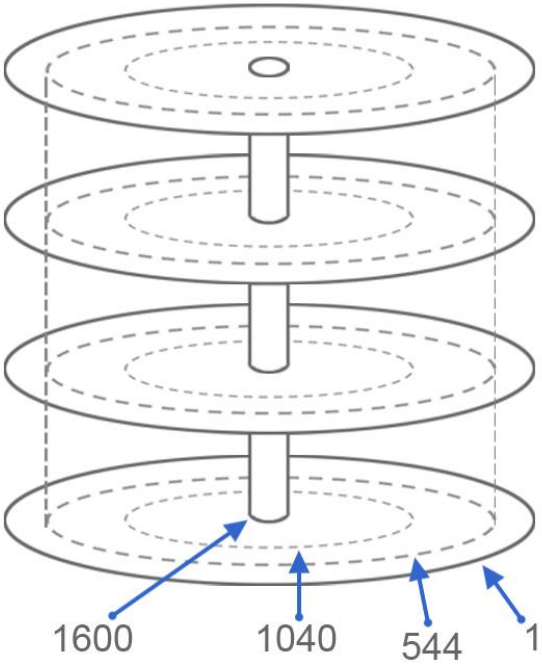
Understanding Partitions

- Physical disk geometry (graphic on next slide)
  - Disks are composed of several platters
  - Platters rotate around a spindle
  - The read/write heads are attached to a boom and moved as a unit
  - Sector: smallest unit on the platter
  - Tracks: one complete rotation around a platter
  - Cylinder: same track on both sides of all platters
  - Partition: a group of multiple consecutive cylinders

Understanding Partitions



Disk Geometry



Example: The first partition would be from cylinder 1 to cylinder 544. The second partition would be from cylinder 545 to cylinder 1040. The last partition would be from cylinder 1041 to 1600.

Partition Naming

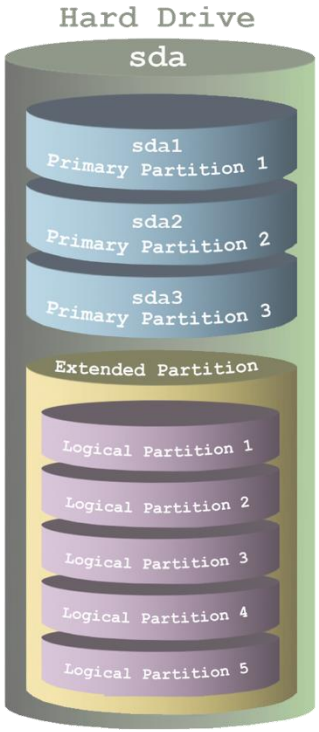
- Each partition has a unique name
- Partition filenames are stored in /dev and are based on the drive where they reside
- Two different types of interfaces
  - /dev/sd* - Drives that start with sd are either SATA (Serial ATA), SCSI (Small Computer System Interface) or USB drives. The first sd drive is called /dev/sda, the second is called /dev/sdb, etc.
  - /dev/hd* - Drives that start with hd are PATA (Parallel ATA), also known as IDE (Integrated Drive Electronics) drives. The first hd drive is called /dev/hda, the second is called /dev/hdb, etc.
- **$** ls /dev/sd*

  sda sda1 sda2 sda3 sda4 sda6 sda7 sda9

Partition Limitations

- Traditional disks using MBR partitioning can have a maximum of four *primary* partitions

- An extended partition acts like a container for additional *logical* partitions



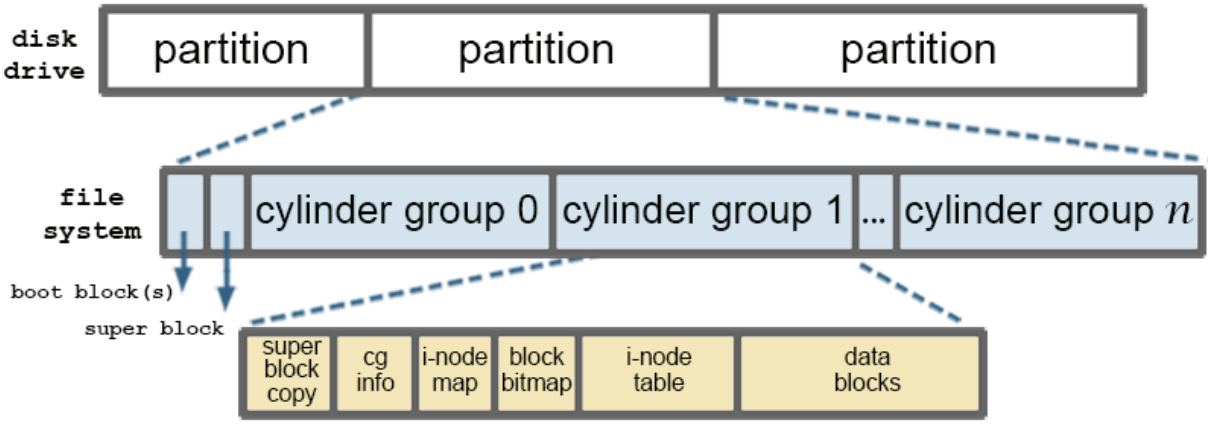
Filesystem

- Terms
  - Filesystem: a structure created on a partition consisting of *tables* defining the location of directories and files
  - Table: database where the filesystem stores *metadata*
  - Metadata: file attributes (i.e. owner, timestamps, data block location, etc.)
  - Inode (identification node): unique number given to each file on the filesystem

Common Filesystems

ext2: Second Extended Filesystem	Advantages: Works well with small and solid-state disk filesystems. Disadvantages: No journaling capability, making it susceptible to data loss in the event of power loss.
ext3: Third Extended Filesystem	Advantages: Can be upgraded from existing ext2 filesystem without data loss. This filesystem performs journaling, which allows for data recovery after a crash. Disadvantages: Writes more to disk than ext2 because of journaling, making it slower. Does not support very large filesystems.
ext4: Fourth Extended Filesystem	Advantages: Support for very large disk volumes and file sizes. Can operate with or without a journal. Backwards compatible with ext3 and ext2. Disadvantages: Not a huge improvement over ext3. No dynamic inode creation.
reiserfs: The Reiser Filesystem	Advantages: The first journaling filesystem for Linux. Works efficiently with small files. Disadvantages: Development of this version has ceased with its successor Reiser4 proceeding slowly without help from the founder.
xfs: Extents Filesystem	Advantages: Works very efficiently with large files. Compatible with the IRIX operating system from SGI. Announced to be the default filesystem for RHEL 7. Disadvantages: The filesystem cannot be shrunk.
vfat: File Allocation Table	Advantages: Supported by almost all operating systems. Commonly used for removable media. Disadvantages: Unable to support very large disks or files. Microsoft's intellectual property claims.
iso: ISO 9660	Advantages: The International Organization for Standardization standard for optical disc media that is supported by almost all operating systems. Disadvantages: Multiple levels and extensions complicate compatibility. Not designed for rewritable media.
udf: Universal Disc Format	Advantages: Designed to replace ISO 9660 and adopted as the standard format for DVDs by the DVD Consortium. Disadvantages: Write support is limited to support revision 2.01 of the standard.

Filesystem Components



- **Superblock:**
  - Area at the beginning of the filesystem used to store important information about the filesystem, including the size of the filesystem, the type of filesystem and which data blocks (where file data is stored) are available

Filesystem Components

- **Group Block**:
  - The filesystem is divided into smaller sections called *groups*. The group block holds data about the group. Each group block also contains a backup copy of the superblock
- **Inode Table**:
  - Each file is assigned a unique *inode* number for the filesystem. This inode number is associated with a table that stores the file's metadata

Introduction to partition schemes

- Before creating partitions, it is important to consider several factors:
  - Will this system support multiple operating systems?
  - How many users will there be on this system?
  - What security policies will be put in place?
  - Will there be a lot of activity on the system?

Supporting Multiple Operating Systems

- Must provide enough space for all operating systems
- If installing Microsoft Windows, install Windows first, leaving room for Linux
- Consider a partition that can be used to share data between operating systems (suggestion: FAT filesystem)

Advantages of Separate Partition for the home Directory

- It is easy to backup or restore an entire partition
- Upgrades of the operating system can be performed much more safely and efficiently
- Because mount options like disk quotas are applied at the partition level, quotas will be preserved on a home partition even when the operating system is replaced or upgraded

Advantages of Separate Partition Writable Directories

- Important to limit how much space regular users can use
- Disk quotas can only be applied on a filesystem by filesystem basis

Security & Usage

- There are many security features that can be applied to individual partitions
- Some directories have a large amount of data written to them consistently, like `/home` and `/var/log` - these should be separate partitions to limit impact on other filesystems

Standard filesystems

Directory	Purpose	Suggested Size
/	The root filesystem holds the files essential to the operation of the system. It must contain the following directories or symbolic links: bin, boot, dev, etc, lib, media, mnt, opt, sbin, srv, tmp, usr, and var.	500MiB-50GiB+ Depends on what is mounted separately
/boot	The /boot directory contains the Linux kernel and the boot loader files.	500MiB-2GB
/home	The /home directory is where user home directories are normally created.	500MiB+ per user
/tmp	The /tmp directory is used to create temporary files for the system and users. If this directory is too small it may prevent applications from functioning correctly.	Minimum of 5 GB + 500MiB+ per user actively logged in

Standard filesystems

Directory	Purpose	Suggested Size
/opt	The /opt directory is where third-party software often is occasionally located. Some examples include: Google Chrome, Google Earth and LibreOffice.	100MiB+ Depends on how many packages are installed
swap	Swap is virtually memory that is not mounted on a directory. This virtual memory is used when actual memory of the system is low. If you have a large amount of memory, you may not need very much swap.	Up to 2 times the physical memory of the system
/usr	The /usr directory contains the bulk of the operating system's files, including most of the commands and system software.	2GiB-10GiB+

Standard filesystems

Directory	Purpose	Suggested Size
/usr/local	This directory is used for locally installed software that should not be upgraded or updated with the operating system.	100MiB+ Size depends on local needs
/var	There are many directories which may have heavy activity under /var for services like mail, ftp, http and printing.	100MiB+ Depending on volume of activity