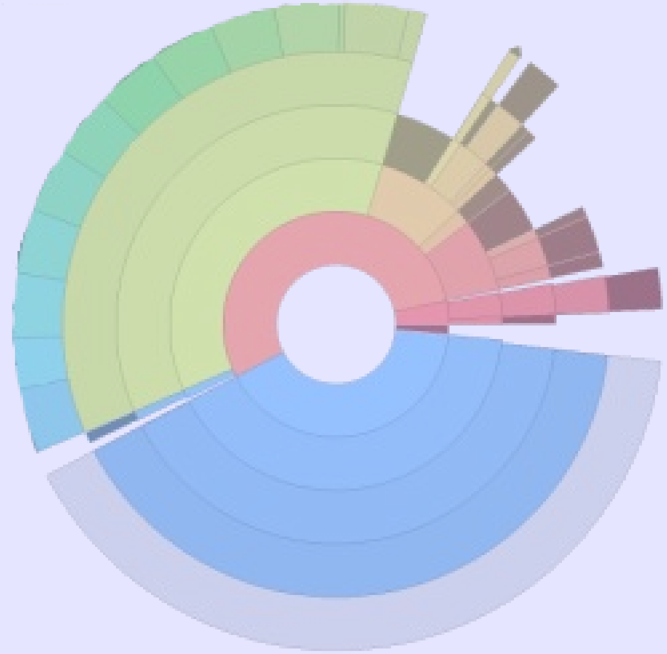
Linux Filesystem Hierarchy and Hard Disk Partitioning



Moreno Baricevic

CNR-IOM DEMOCRITOS
Trieste, ITALY

File System

A file system **is a set of methods and data structures** used to **organize, store, retrieve and manage information** in a permanent storage medium, such as a hard disk.

Its main purpose is to **represent and organize storage resources**.

File System: elements

Name space: is a way to assign names to the items stored and organize them hierarchically.

API: is a set of calls that allow the manipulation of stored items.

Security Model: is a scheme to protect, hide and share data.

Implementation: is the code that couples the logical model to the storage medium.

File Systems: Basic Concepts (1/2)

Disk: A permanent storage medium of a certain size.

Block: The smallest unit writable by a disk or file system. Everything a file system does is composed of operations done on blocks.

Partition: A subset of all the blocks on a disk.

Volume: The term is used to refer to a disk or partition that has been initialized with a file system.

Superblock: The area of a volume where a file system stores its critical data.

File Systems: Basic Concepts (2/2)

Metadata: A general term referring to information that is about something but not directly part of it. For example, the size of a file is very important information about a file, but it is not part of the data in the file. Ownerships, access permissions, creation/modification/access time, are also part of the metadata information pertaining a file.

Journaling: A method of insuring the correctness of file system metadata even in the presence of power failures or unexpected reboots (atomic write).

Attribute: A name and value associated with the name. The value may have a defined type (string, integer, etc.).

Modern File System Features

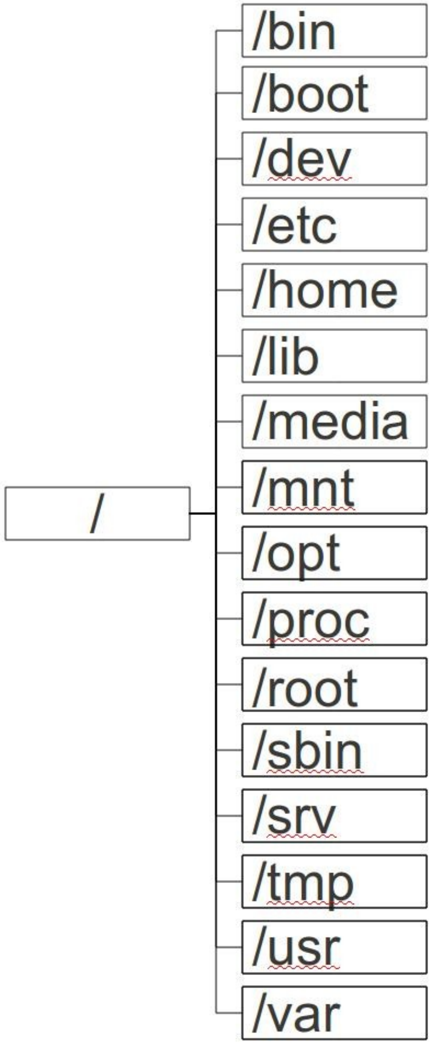
- *Journaling*: write data to journal, commit to file system when complete in *atomic* operation
  - reduces risk of corruption and inconsistency
- Faster file lookups through *balanced tree*
- *Snapshot*: retain status of file system at given point in time by copying metadata and marking object data referred as *copy-on-write*
- *Deduplication*: identify identical storage objects, consolidate and mark them *copy-on-write*

(Local) File Systems: few examples

- FAT
- NTFS
- ext2, ext3, ext4
- Reiserfs
- xfs
- jfs
- ...

File System Hierarchy

UNIX/LINUX Directory Structure



/

/bin
/boot
/dev
/etc
/home
/lib
/media
/mnt
/opt
/proc
/root
/sbin
/srv
/tmp
/usr
/var

In the Filesystem Hierarchy Standard (FHS) all the files and directories appear under the “root directory” “/”.

Directory Structure

- / (root or “slash”)
  - Root directory of the entire hierarchy
- /bin
  - Essential commands: ls, cp, rm, cat...
- /boot
  - Kernel and boot loader files
- /dev
  - Essential devices, such as disk drives, serial ports..

Directory Structure

- /etc
  - System wide configuration files
- /home
  - User's home directories, containing saved files, personal settings...
- /lib
  - Shared libraries needed by the programs on the root filesystem

Directory Structure

- /media
  - Mount points for removable media such as CD-ROMs
- /mnt
  - Mount points for temporary mounts by the system administrator.
- /opt
  - Optional application software packages.

Directory Structure

- /proc
  - Virtual filesystem documenting kernel and process status as text files
- /root
  - Root user home directory
- /sbin
  - Essential system binaries, like init, ifconfig, mount...

Directory Structure

- /usr
  - Contains all commands, libraries, man pages, games and static files for normal operation (User System Resources).

    /usr/bin

    /usr/sbin

    /usr/lib

    /usr/src

    /usr/man

    /usr/share

    /usr/local

    - /usr/local/bin
    - /usr/local/lib

Directory Structure

- /var
  - Variable files (files whose content is expected to continually change during normal operation of the system) such as logs, spool files, and temporary e-mail files.
- /tmp
  - Temporary files, often not preserved between system reboots.

Partitioning

Why partitioning?

- Protect and isolate files
  - A loss of a file system in one partition doesn't affect the others
  - Limits effects of file system full

- Different partitions for different purposes
  - Different filesystems (Encrypted FS for home directories)
  - Software RAID
  - LVM

Partitions Overview

- Primary and Extended partitions.
  - Maximum 4 primary partitions for hard disk.
  - Maximum 1 extended partition.
  - Extended partition can be divided in more logical partitions (>15, depends on partition table).
- One partition must be a contiguous chunk of blocks.
- Once you create a partition you have to live with it (with some exceptions).

Standard Partition Scheme

- /
- /boot
- /home
- /usr
- /tmp

- /var
- /opt
- /scratch
- <swap_area>

Standard Partition Scheme

- /
  - The root directory of the entire file system hierarchy. Contains all the other directories not bound to a specific partition (/etc, /bin, /sbin).
  - 10 GB may be enough.
- /boot
  - Boot loader files (e.g. kernels, initrd, boot loader configuration).
  - At least 100 MB big, 500MB may be appropriate for large ramfs/initrd and multiple kernels. Automatic kernel updates may fill up the space quickly.

Standard Partition Scheme

- /home
  - Users' home directory, containing saved files, personal settings.
  - The size depends on the purpose of the machine (e.g. larger on a desktop, minimal on a print server).
- /usr
  - Contains the majority of (multi-)user utilities and applications.
  - Around 15 to 20 GB large.

Standard Partition Scheme

- /tmp
  - Temporary files, often not preserved between system reboots or periodically removed if unused.
  - 10 GB.
- /var
  - Variable files, files whose content is expected to continually change during normal operation of the system, such as logs, spool files, and temporary e-mail files.
  - 10 GB (not necessarily a separate partition on a desktop installation).

Standard Partition Scheme

- /opt
  - Optional application software packages (usually shared via NFS inside a cluster environment).
  - 15 – 20 GB large.
- /scratch
  - A scratch file system (for temporary data and large files used by scientific applications, pre-processing and post-processing).
  - Maximum available space.

Standard Partition Scheme

- <swap_area>
  - A partition that temporary holds inactive memory pages.
  - The size depends on the amount of RAM, hard drive capacity and typical usage. In the past (small amount of RAM), twice the size of RAM was suggested.
- Other partitions
  - e.g. per-applications reserved partitions
    - One partition for each applications (as apache, mysql...)

Typical question: I have X terabytes of available space, how can I partition it?

There's no easy answer, depends on many factors:

- What kind of services or applications will run on your server?
- What do you need the most on your system: performance or reliability (or a combination of both)?
- Do you need specific type of file system (e.g. encrypted file systems)?

Tips

- Don't allocate all the available space (keep some for future uses).
- Use LVM.
- If you have a raid controller with multiple disks and the overall space isn't an issue, reserve one hard disk as hot-spare.
- ...

Questions?

Resources

- http://www.linuxtopia.org/online_books/linux_beginner_books/linux_filesystem/c23.html