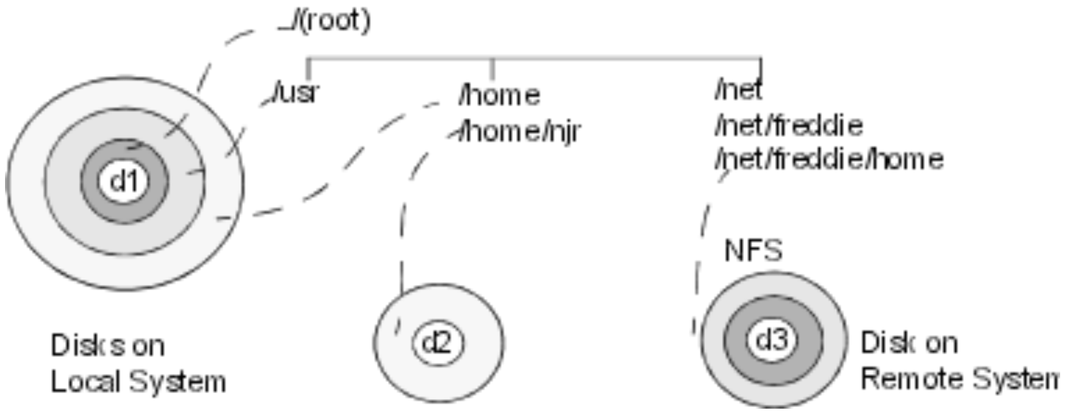
Linux Filesystem Images

Digital Forensics

Linux/UNIX Filesystems

- Linux, with its UNIX heritage, supports a wide variety of filesystems
- The most commonly used filesystem format for Linux is the ext filesystem, usually version 4 simply because that is the default for many OS install programs
- Each filesystem format has a different structure and tries to meet different needs
- A single computer may have multiple filesystems of differing types



http://systemmanager.ru/nbadmin.en/ch29s12s12s03.htm

MacOS

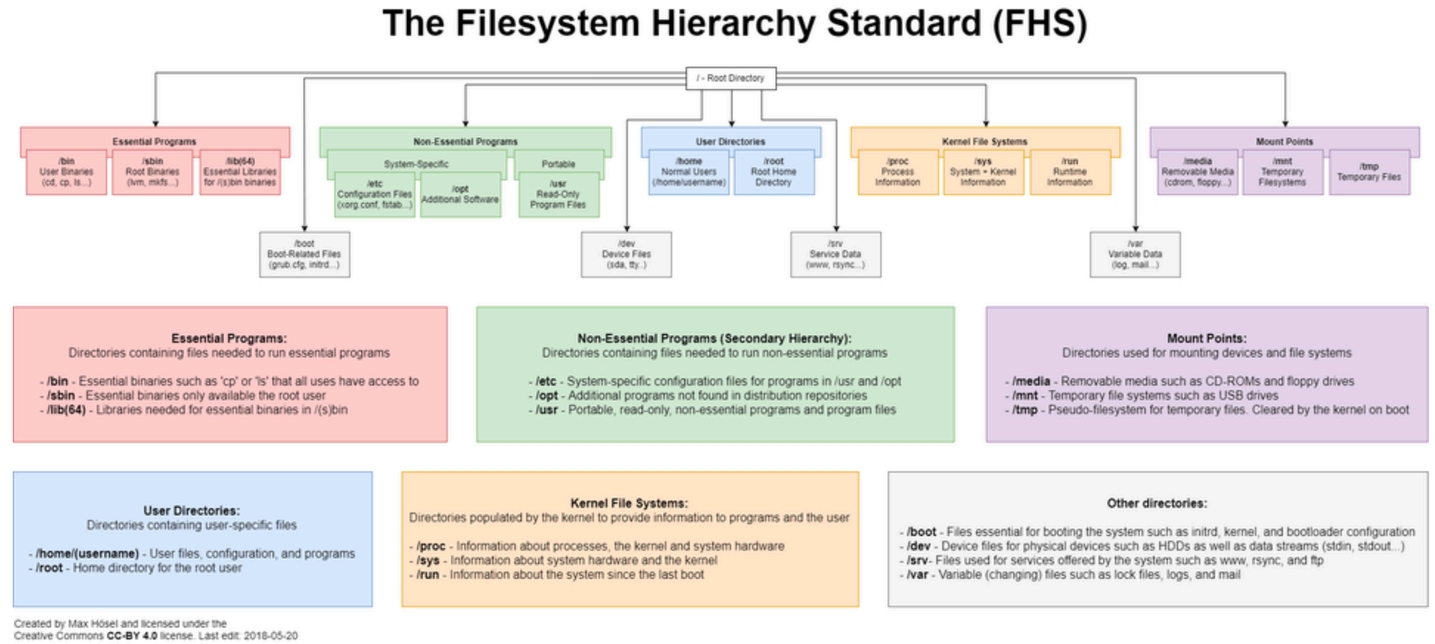
- Apple for most of its early years shipped its operating systems with the HFS filesystem
- To support more file management options and larger system, Apple improved on HFS to produce HFS+
- Recently Apple has moved to another filesystem type called APFS to better support SSDs and made it their default filesystem for new disks
- Other filesystems work with SSDs, but they have kludges and hacks to avoid early death on SSDs - these work but are not a solution designed from the ground up like APFS

Commercial UNIX

- Several vendors produced commercial UNIX systems (e.g. IBM, HP, Oracle, etc.)
- Many of them implemented the UFS filesystem, the ones that are out there now support many different types of filesystems
- ZFS is very popular with file server suppliers
- Many NAS devices run Linux or BSD UNIX and use EXT4 and ZFS respectively

Linux/UNIX Forensic Considerations

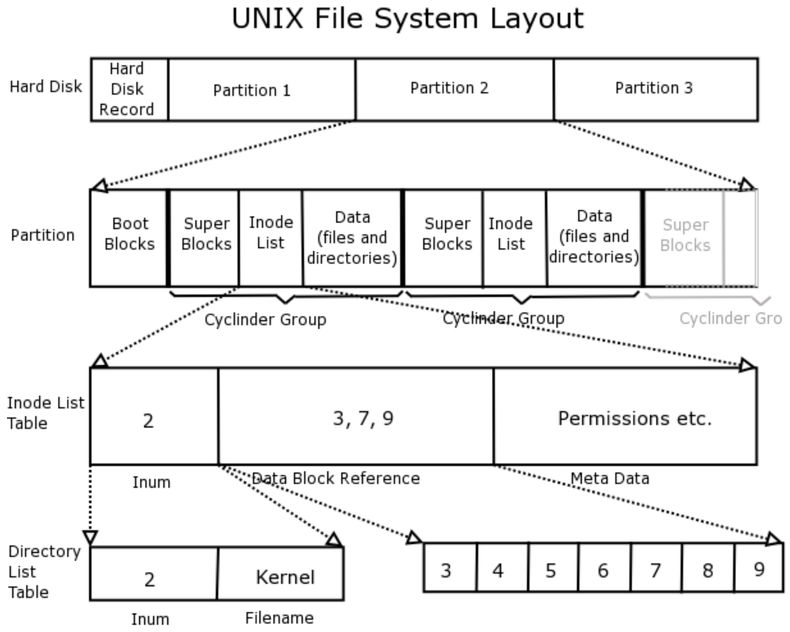
- Linux/UNIX does not use a registry, programs make their own choices about where and how to keep configuration and user data
- Linux/UNIX system files are placed in an organized hierarchy following best practice guidelines (ref: FHS)
- Linux/UNIX systems are designed for real-time simultaneous multi-user use, so filesystem permissions get used more than in Windows to control access to files



https://www.reddit.com/r/linux/comments/8kt99k/the_file_system_hierarchy_standard_visualized_or/

Linux/UNIX Forensic Challenges

- UNIX filesystems typically are not friendly to users who make mistakes or don't know what they are doing
- If it is a desktop system, the GUI tries to encourage users to send files to the recycle bin before deleting, but nothing enforces that
- When most UNIX filesystems delete a file, they remove the filename from the disk, and disconnect the storage blocks from the metadata (called an inode in some filesystems)
- Storage is much more actively managed in UNIX systems, they don’t fragment the same way Windows filesystems do, and data gets overwritten much more rapidly making recovery of deleted file data less likely to succeed



Linux/UNIX Forensic Challenges

- Linux/UNIX systems do not use drive letters in file pathnames, they create a view of filesystems by attaching additional filesystems to existing directory names
- Capturing the files of interest often means capturing all of the filesystems, which may not be local to the computer
- Network-based filesystems using SANs, iscsi, and other technologies is much more common with Linux/UNIX systems than with Microsoft implementations
- The step of identifying storage of interest can be more complex when Linux/UNIX systems are present
- The df and mount commands can be used to help identify storage locations and devices in use, the automounter can mean that storage is dynamic and you must investigate the automounter configuration and logs to determine if additional devices or networked systems might be relevant

Slack Spaces

- Linux does not leave random data in ram slack or file slack
- Similarly, any reasonably modern version of a Microsoft operating system no longer leaves leftovers from previous programs in ram or file slack - they get filled with recognizable patterns
- Partition gaps are still the same issue they have always been
- Ref: https://superuser.com/questions/1058565/file-slack-ram-slack-why-does-windows-write-arbitrary-ram-bytes-to-disk-does

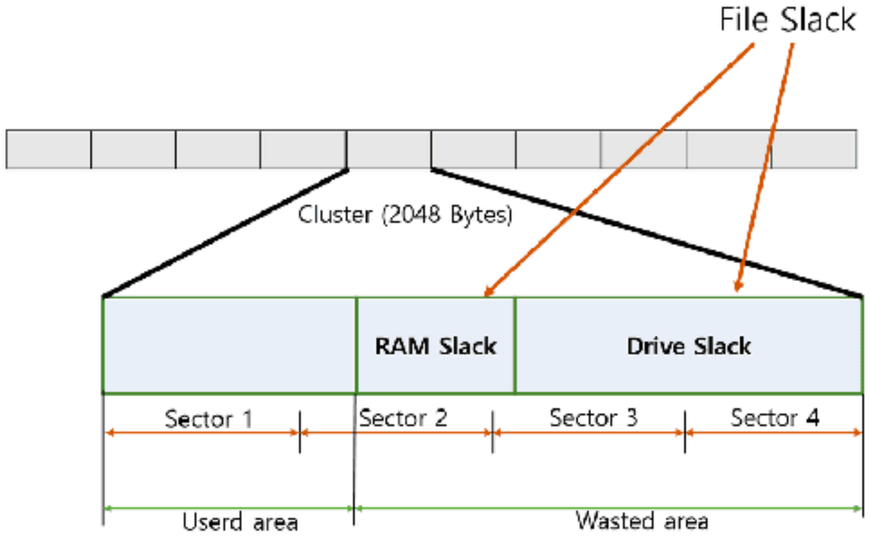


Fig. 1. Slack space in a cluster.

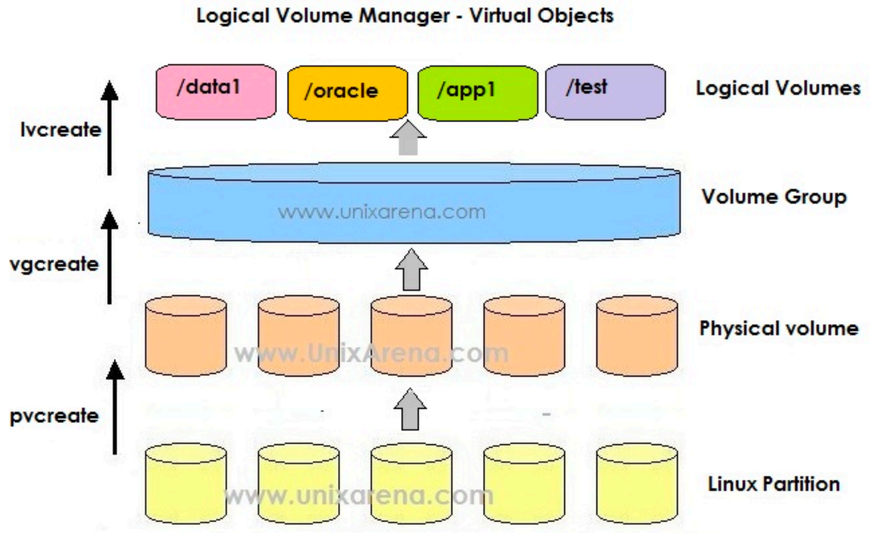
https://www.semanticscholar.org/paper/Data-Hiding-in-Slack-Space-Revisited-Mohan-Thampy/8d29f53300b7df66cdd82442300c3c4a57f0833c

Drive Naming

- Linux/UNIX systems use a convention for drive naming which is fairly universal
- All the device file names live in the /dev directory
- The filenames are commonly 2 or 3 characters to indicate the device driver in use, then a letter to indicate which device of that type you are accessing, then a number to identify the partition you want
- /dev/sda1 is the first SCSI or SATA drive, partition 1
- Leaving off the partition number allows you to access the entire device (i.e. /dev/sda would access the whole drive including any gaps or unallocated space)
- fdisk -l will list all recognized drives and their partitions, as will lsblk in Linux, diskutil is the tool to use for MacOS
- Superuser permissions are required to directly access storage devices using filenames found in the /dev directory

Logical Volumes In Linux

- Linux/UNIX systems are much more likely to use logical volumes which span drives than Microsoft systems
- Linux/UNIX installation programs often create them by default to give the system user more flexibility later to grow the system
- Linux/UNIX systems also commonly create RAID metadisks and storage pools for a number of reasons (e.g. Linux/UNIX is used for larger servers, workstation class desktops, and infrastructure devices)
- Logical volumes are easy to recognize; their device names begin with lvm



https://www.unixarena.com/2013/08/linux-lvm-volume-creation-operation.html/

Linux Image Analysis Tools

- Linux has a wealth of command line tools for finding data in any kind of container, including filesystem images (e.g. grep, sed, awk, etc.)
- They are many forensic tools for Linux systems that can analyze filesystem structures to be more focused in your search, see https://opensource.com/article/18/4/linux-filesystem-forensics for a good example of using some of these tools for an investigation
- See the course github website for more lists of tools and resources for Linux/UNIX filesystem analysis