Linux File Systems

Mac File Systems Too

Objectives

- Describe Linux file structures
- Describe Macintosh file structures
- Use Linux forensics tools*

Linux File Structures

ext4 & more

Linux Commands Warmup

- You can follow along with the activity on pages 310-312 using the **validate** vApp

Linux Filesystem History

- ext: inodes immutable; didn’t last long
  - Improved on by ext2
  - ext3 followed and added journaling
- Fourth Extended File System (ext4)
  - Added support for much large partition (>16TB)
  - Standard file system for most Linux distributions
    - Others: https://en.wikipedia.org/wiki/File_system#LINUX

ext4 Structure

- All things are files
  - Drives / NICs / memory / directories / etc.

- Four core components of structure
  - **Boot Block** – Bootstrap code to start OS is here
  - **Superblock** – metadata; disk geometry & inode tracking
  - **Inode Blocks** – describe data block locations; assigned to each file allocated
  - **Data Blocks -** directories & files stored here

What are index nodes (inodes)?

- Contain file and directory metadata
  - Also link data stored in data blocks
- An assigned inode contains:
  - Mode and type of file or directory
  - Number of links to a file or directory
  - UID and GID of the file or directory’s owner
  - Number of bytes in the file or directory
  - File or directory’s last access time and last modified time
- inode contents (cont.)
  - Inode’s last file status change time
  - Block address for the file data
  - Indirect, double-indirect, and triple-indirect block addresses for the file data
  - Current usage status of the inode
  - Number of actual blocks assigned to a file
  - File generation number of version number
  - Continuation inode’s link

https://en.wikipedia.org/wiki/Inode

What isn’t in an inode?

- Filename
- Path

inode Pointers

- First inode has 13 pointers
  - Pointers 1 to 10 are direct pointers to data storage blocks
- Pointer 11 is an **indirect pointer**
  - Links to 128 pointer inodes and each pointer links directly to 128 blocks
  - Pointer 12 is a **double-indirect pointer**
    - Links 128 inode pointers to 128 inode pointers each
  - Pointer 13 is a **triple-indirect pointer**
    - Links 128 inode pointers to 128 inode pointers, which each point to 128 inode pointers

inode Pointers (cont.)

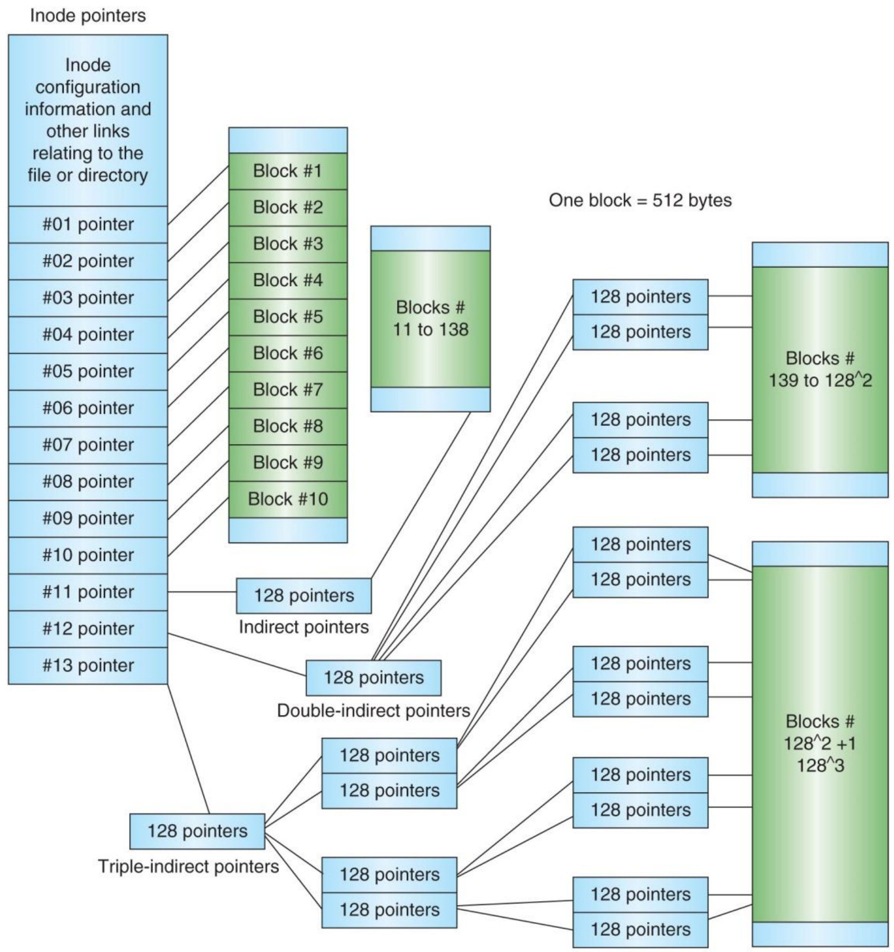


Figure 7-3 Inode pointers in the Linux file system

Bad Block inode

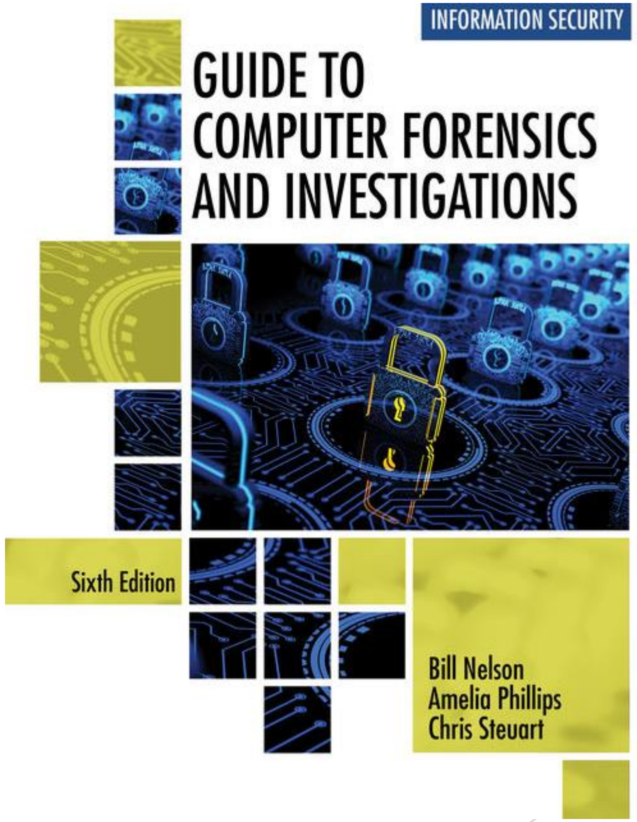
- Tracks bad sectors on a disk
- **`badblocks`** is a linux command to view badblocks
  - **`mke2fs`** & **`e2fsck`** include protections when scanning for badblocks

Links

- **Hard Link:** pointer that allows accessing a file with a different file name
  - `ln` command
  - inodes for hard linked files are identical
  - `.` and `..` are hard links to the current directory and the parent directory
- **Link Count:** Field inside inode that specifies quantity of hard links
  - `ls -ld` to see link count
- **Symbolic Link:** pointers to files not included in link count / can point across drives and are not dependent on inode references
  - AKA: soft link or symlink
  - Dependent on continued existence of what they point to

References

- *Guide to Computer Forensics and Investigations*
  - *ISBN: 9780357688595*



Mac File Structures

HFS / HFS+ / APFS

- Before OS X, **Hierarchical File System (HFS)**
  - Files stored in nested directories (folders)
- **Extended Format File System (HFS+)**
  - Introduced with Mac OS 8.1
  - Supports smaller file sizes on larger volumes, resulting in more efficient disk use
- **Apple File System (APFS)**
  - Introduced in macOS High Sierra
  - When data is written to a device, metadata is also copied to help with crash protection

Mac File Structure Basics

- In Mac, a file consists of two parts:
  - **Data fork** and **resource fork**
  - Stores file metadata and application information
- The data fork typically contains data the user creates
- Resource block contains additional information
  - Such as menus, dialogs, executable code, etc.
- Resource or data block can be blank

macOS File Example

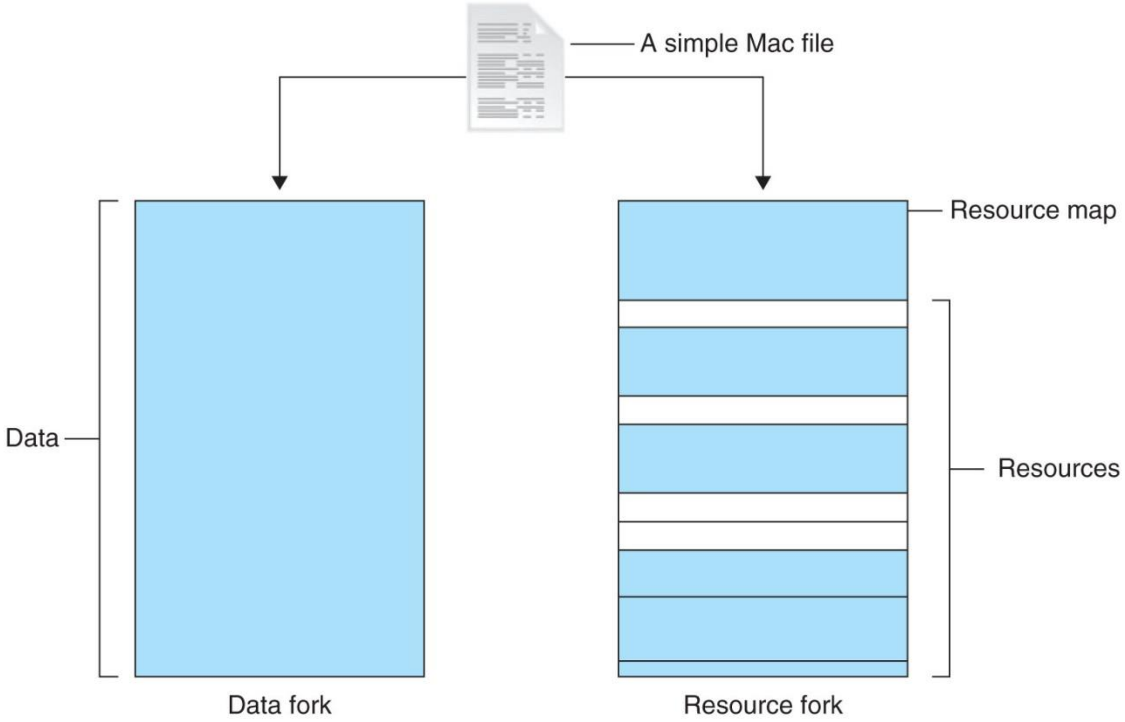


Figure 7-9 The resource fork and data fork in a macOS file

Allocation v. Logical Blocks

- Volumes have **allocation** and **logical blocks**
  - Logical blocks cannot exceed 512 bytes
  - Allocation blocks are a set of consecutive logical blocks

- Two end of file (EOF) descriptors
  - **Logical EOF**
    - Actual ending of the file
  - **Physical EOF**
    - The number of bytes allotted on the volume for a file

Allocation v. Logical Blocks (Cont.)

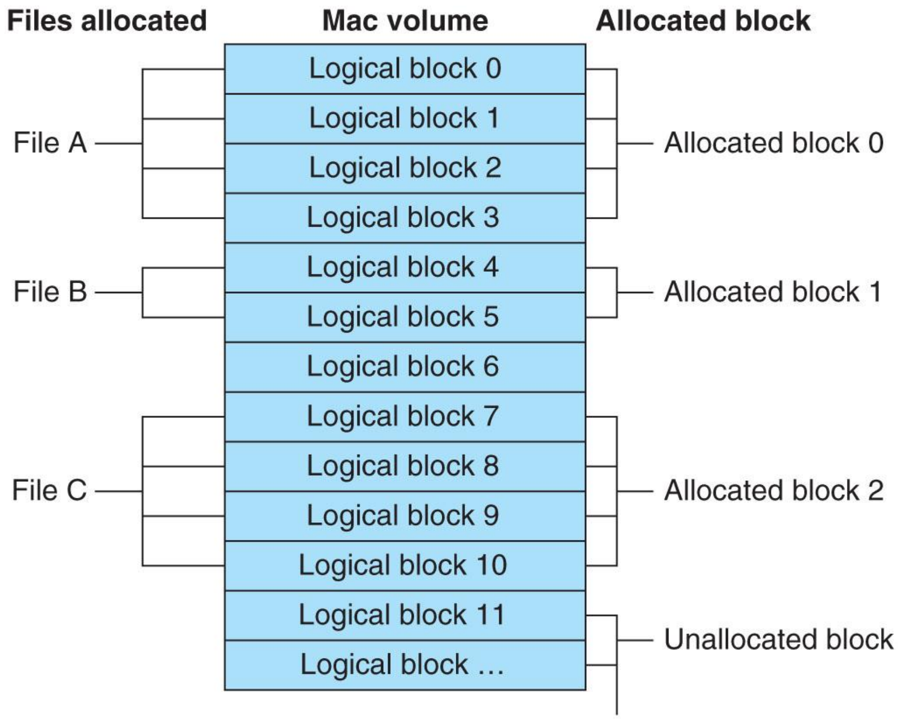

Files allocated
Mac volume
Allocated block
Logical block 0
Logical block 1
Logical block 2
Logical block 3
Logical block 4
Logical block 5
Logical block 6
Logical block 7
Logical block 8
Logical block 9
Logical block 10
Logical block 11
Logical block …
File A
File B
File C
Allocated block 0
Allocated block 1
Allocated block 2
Unallocated block

Figure 7-10 Logical and allocation block structures

Logical EOF v. Physical EOF

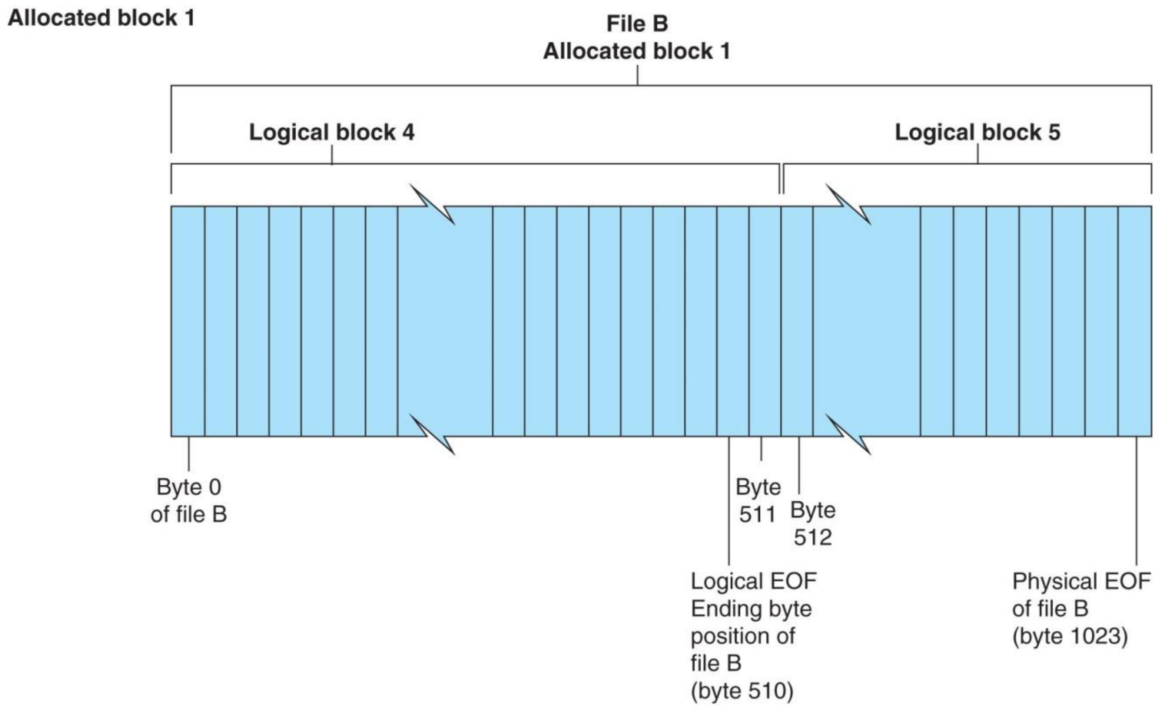


Figure 7-11 Logical EOF and physical EOF

Clumps & Other Core Structures

- **Clumps**
  - Groups of contiguous allocation blocks
  - Reduce fragmentation
- Macintosh OS’s that use HFS use:
  - First two logical blocks, 0 and 1, as boot blocks
  - **Master Directory Block (MDB)** or **Volume Information Block (VIB)**
    - Stores all information about a volume
  - **Volume Control Block (VCB)**
    - Stores information from the MDB when OS mounts
- **Extents overflow file**
  - Stores any file information not in the MDB or a VCB

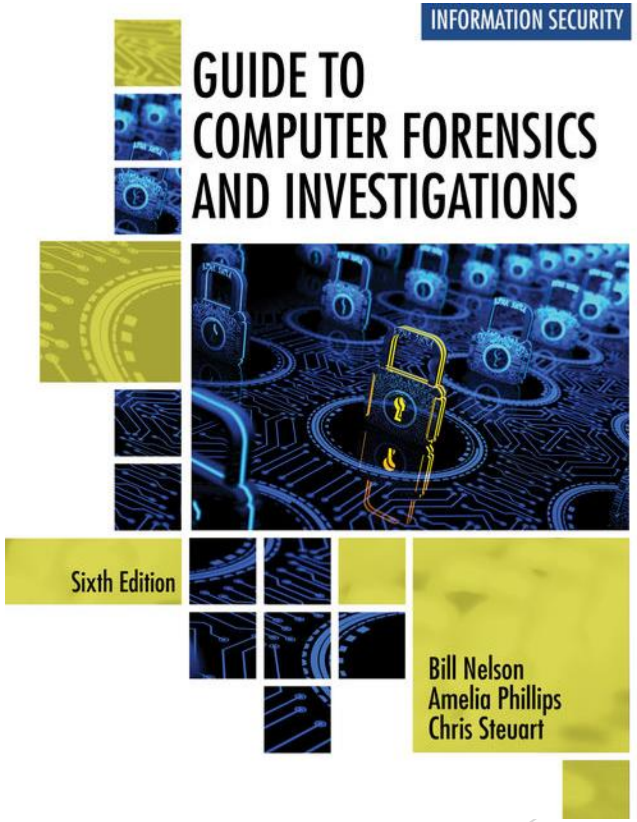
Catalog & B*-tree

- **Catalog**
  - The listing of all files and directories on the volume
  - Maintains relationships between files and directories

- **B*-tree** file system in earlier Mac versions
  - Actual file data is stored on the leaf nodes
  - B*-tree also uses **header**, **index3**
  - +, and **map nodes**

References

- *Guide to Computer Forensics and Investigations*
  - *ISBN: 9780357688595*



Mac Forensic Considerations

Example Differences v. Linux

- Linux has the `/home/username` and `/root` directories
- In macOS, the folders are `/users/username` and `/private/var/root`
- The `/home` directory exists in the macOS but it is empty
- macOS users have limited access to other user accounts’ files and the guest account is disabled

Data Formatting

- Application settings are in three formats:
  - Plaintext, plist files, and the SQLite database
  - **Plist files** are preference files for installed applications on a system

- FileVault is used to encrypt and decrypt a user's `/users` directory

Keychain

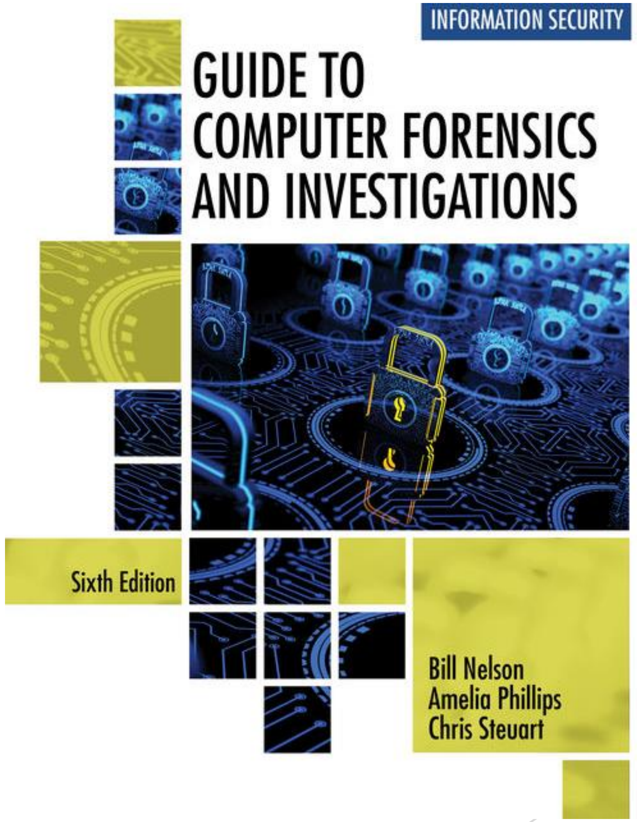
- **Keychains**
  - Files used to manage passwords for applications, Web sites, and other system files
  - The Mac application Keychain Access enables you to restore passwords
- Deleted files are in the Trashes folder
  - If a file is deleted at the command line, however, it doesn’t show up in the trash

Acquisition

- Acquisition Methods in macOS
    - Make an image of the drive
    - Removing the drive from a modern Mac is difficult
        - Attempting to do so without Apple factory training could damage the computer
        - Also difficult for MacBook Air (need special screwdrivers)
    - Use a macOS-compatible forensic boot CD/DVD to make an image
- Several popular vendors listed on pages 325 & 326

References

- *Guide to Computer Forensics and Investigations*
  - *ISBN: 9780357688595*



Linux Forensic Tools

Overview

- Most commercial computer forensics tools can analyze Linux Ext2, Ext3, Ext4, ReiserFS, and Reiser4 file systems
- Freeware tools include Sleuth Kit and its Web browser interface, Autopsy Forensic Browser
- Foremost
  - A freeware carving tool that can read many image file formats
  - Configuration file: foremost.conf
- **Tarball**
  - A data file containing one or more files or whole directories and their contents

Configuring & Running Autopsy (browser) on Linux (Debian)

- `sudo apt-get install –y sleuthkit autopsy`

- `mkdir /home/dsu/evidence`

- `sudo autopsy –d /home/dsu/evidence`

Configuring & Running Autopsy (latest) on Linux (Debian)

- Instructions: https://github.com/sleuthkit/autopsy/blob/develop/Running_Linux_OSX.txt

References

- *Guide to Computer Forensics and Investigations*
  - *ISBN: 9780357688595*