

Guide to Computer Forensics and Investigations Sixth Edition

Chapter 7

Linux and Macintosh File Systems





Objectives

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



Examining Linux File Structures (1 of 2)

- UNIX distributions
 - Silicon Graphics, Inc. (SGI) IRIX, Santa Cruz Operation (SCO) UnixWare, Sun Solaris, IBM AIX, and HP-UX
- Linux distributions
 - Ubuntu, CentOS, Mint, Fedora, and Gentoo
 - Linux is only the core of the OS
- All UNIX-like OSs have a kernel
 - So do all Windows OSs



Examining Linux File Structures (2 of 2)

- Remember that UNIX and Linux commands are case sensitive
 - Wrong capitalization can mean your commands are rejected as incorrect or interpreted as something different



File Structures in Ext4 (1 of 3)

- The early file system standard was **Second Extended File System (Ext2)**
 - **Third Extended File System (Ext3)** replaced Ext2 in most Linux distributions
- **Fourth Extended File System (Ext4)** added support for partitions larger than 16 TB
 - Improved management of large files and offered more flexibility
 - Adoption of Ext4 was slower in some Linux distributions
 - Now considered the standard file system for most distributions



File Structures in Ext4 (2 of 3)

- Everything is a file
 - Files are objects with properties and methods
- UNIX/Linux file system consists of four components
- **Boot block**
 - Contains the bootstrap code
 - UNIX/Linux computer has only one boot block, located on the main hard disk



File Structures in Ext4 (3 of 3)

- **Superblock**

- Specifies disk geometry, available space, and keeps track of all inodes
- Manages the file system

- **Inode blocks**

- First data after the superblock
- Assigned to every file allocation unit

- **Data blocks**

- Where directories and files are stored on a disk drive
- This location is linked directly to inodes



Inodes (1 of 5)

- Contain file and directory metadata
 - Also link data stored in data blocks
- An assigned inode contains the following:
 - Mode and type of file or directory
 - Number of links to a file or directory
 - UID and GID of the file's or directory's owner
 - Number of bytes in the file or directory
 - File's or directory's last access time and last modified time



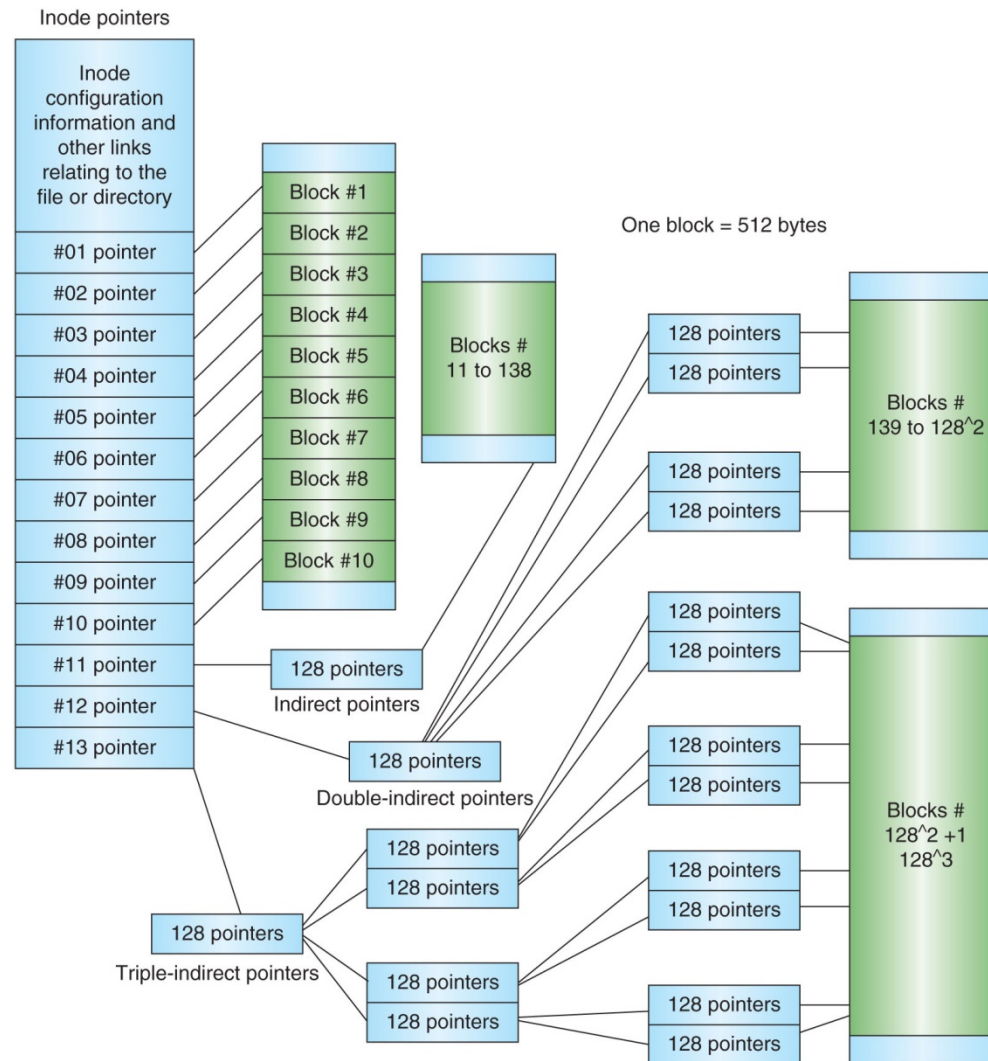
Inodes (2 of 5)

- An assigned inode contains the following (cont'd):
 - 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



Inodes (3 of 5)

- 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**
 - Pointer 13 is a **triple-indirect pointer**





Inodes (5 of 5)

- **Bad block inode**

- Keeps track of disk's bad sectors
- To find bad blocks on a Linux computer, use the following commands
 - `badblocks` - must log in as root to use
 - `mke2fs` and `e2fsck` - include safeguards that prevent them from overwriting important information



Hard Links and Symbolic Links (1 of 6)

- **Hard link**

- A pointer that allows accessing the same file by different filenames
- Use the `ln` command to create a hard link



Hard Links and Symbolic Links (2 of 6)

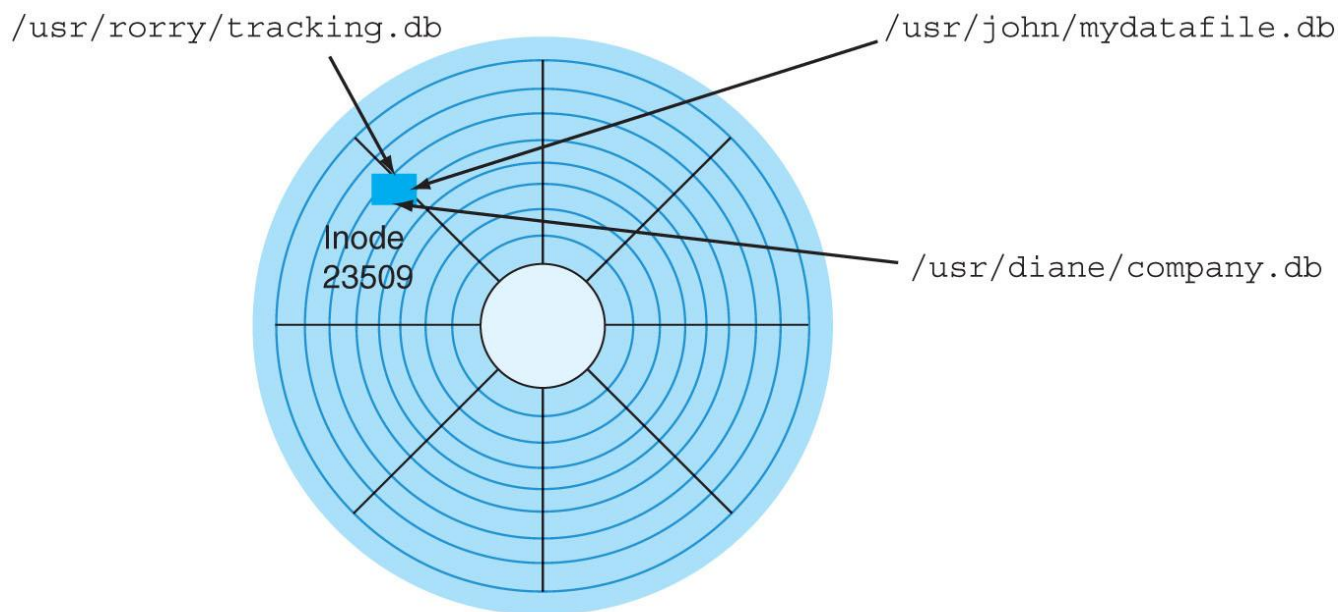


Figure 7-4 Hard-linked files with different filenames



Hard Links and Symbolic Links (3 of 6)

- **Link count**

- A field inside each inode that specifies the number of hard links



Hard Links and Symbolic Links (4 of 6)

```
student@Ubuntu16: ~  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
student@Ubuntu16:~$ ls -a  
..      .config      .gnupg      Pictures    .Xauthority  
..      Desktop     .gnupg      .profile   .xsession-errors  
..      Documents  .ICEauthority Public      .xsession-errors.old  
..      Downloads  .local      Templates  
..      examples.desktop Music        Videos  
student@Ubuntu16:~$
```

The dot-dot indicating the parent directory

The dot indicating the current directory

Figure 7-5 The `ls -a` command showing the dot and dot-dot notation

Source: www.ubuntu.com



Hard Links and Symbolic Links (5 of 6)

- **Symbolic links**

- Pointers to other files and aren't included in the link count
- Also known as “soft links” or “symlinks”
- Can point to items on other drives or other parts of the network
- Have an inode of their own
 - Not the same as the inode of the item they are pointing to
- Depend on the existence of the destination they are pointing to



Hard Links and Symbolic Links (6 of 6)

```
student@Ubuntu16: /tmp
student@Ubuntu16:/tmp/chap07$ cd /tmp
student@Ubuntu16:/tmp$ mkdir testsym
student@Ubuntu16:/tmp$ cd testsym
student@Ubuntu16:/tmp/testsym$ touch test1 test2
student@Ubuntu16:/tmp/testsym$ cd ..
student@Ubuntu16:/tmp$ ln -s /tmp/testsym mysym
student@Ubuntu16:/tmp$ ls -l mysym
test1
test2
student@Ubuntu16:/tmp$ ls -l mysym
lrwxrwxrwx 1 student student 12 Jun 25 22:20 mysym -> /tmp/testsym
student@Ubuntu16:/tmp$
```

Figure 7-8 Creating a symbolic link

Source: www.ubuntu.com



Understanding Macintosh File Structures (1 of 2)

- Mac OS X version 10.13
 - Code-named High Sierra
 - Current version
 - Offers better security, encryption, and performance speeds
- With OS X, Macintosh moved to the Intel processor and become UNIX based



Understanding Macintosh File Structures

(2 of 2)

- 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



An Overview of Mac File Structures (1 of 7)

- 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, such as text or spreadsheets
 - Applications also read and write to the data fork
- Resource block contains additional information
 - Such as menus and dialog boxes
- A volume is any storage medium used to store files
 - It can be all or part of the storage media for hard disks



An Overview of Mac File Structures (2 of 7)

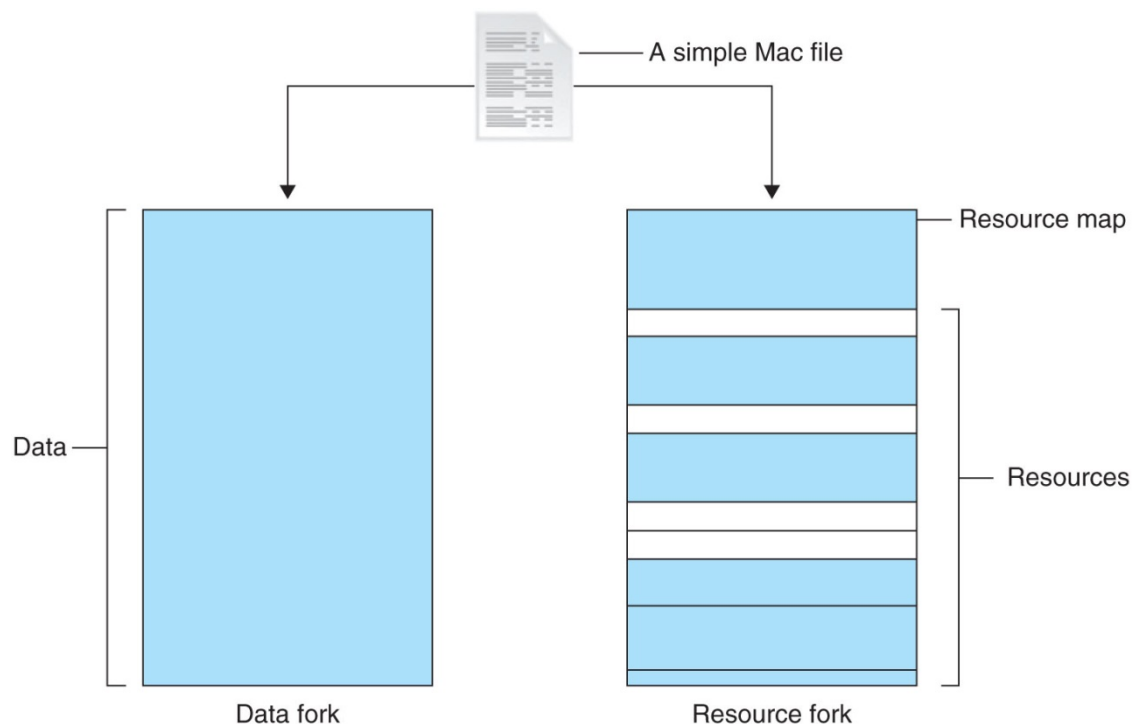


Figure 7-9 The resource fork and data fork in a macOS file



An Overview of Mac File Structures (3 of 7)

- 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



An Overview of Mac File Structures (4 of 7)

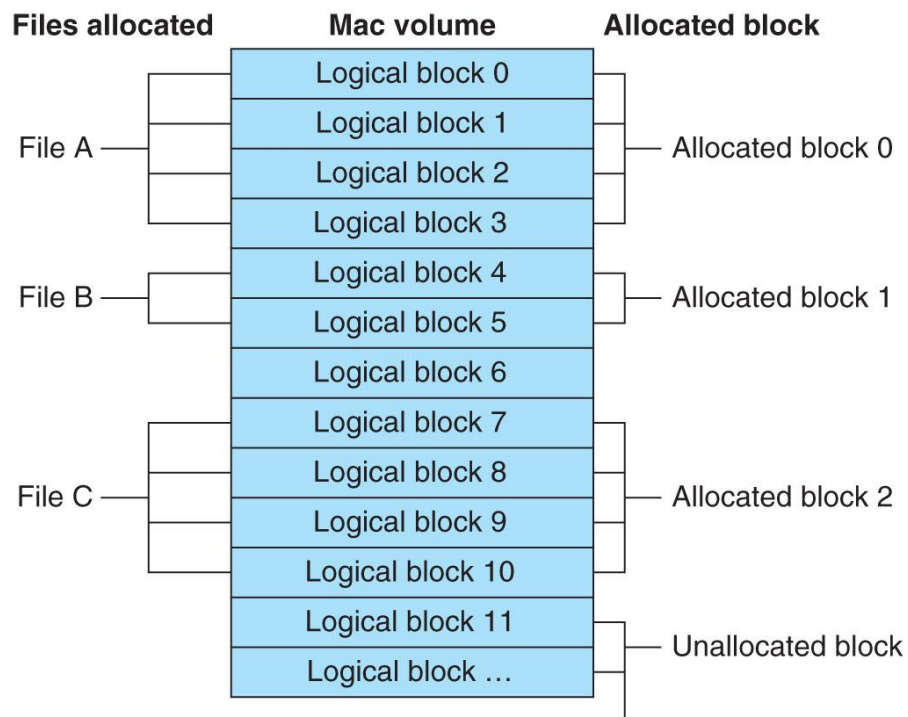


Figure 7-10 Logical and allocation block structures



An Overview of Mac File Structures (5 of 7)

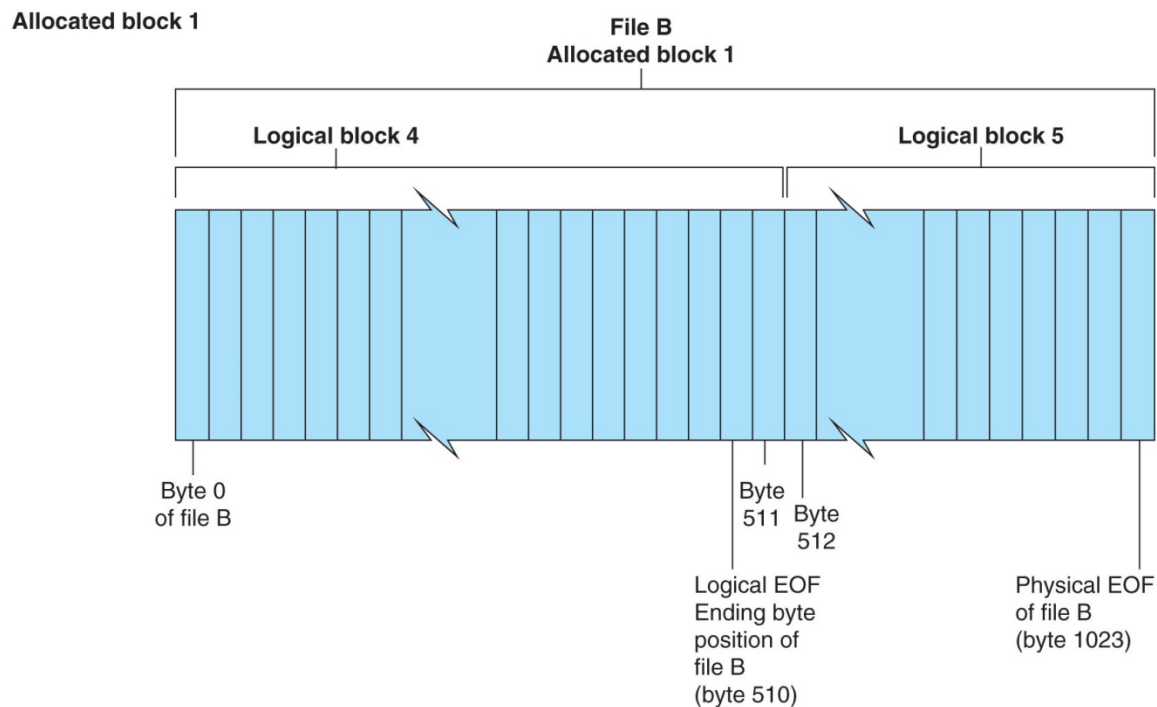


Figure 7-11 Logical EOF and physical EOF



An Overview of Mac File Structures (6 of 7)

- **Clumps**
 - Groups of contiguous allocation blocks
 - Reduce fragmentation
- Older Macintosh OSs 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



An Overview of Mac File Structures (7 of 7)

- **Catalog**

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

- **B*-tree** file system in earlier Mac version

- Actual file data is stored on the leaf nodes
- B*-tree also uses **header**, **index**, and **map nodes**



Forensics Procedures in Mac (1 of 6)

- There are some differences between Linux and macOS file systems
 - 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



Forensics Procedures in Mac (2 of 6)

- For forensics procedures in macOS:
 - You must know where file system components are located and how both files and file components are stored
- 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



Forensics Procedures in Mac (3 of 6)

- **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



Forensics Procedures in Mac (4 of 6)

- Acquisition Methods in macOS
 - Make an image of the drive
 - Removing the drive from a Mac Mini case 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



Forensics Procedures in Mac (5 of 6)

- Acquisition Methods in macOS (cont'd)
 - BlackBag Technologies sells acquisition products specifically designed for OS 9 and OS X
 - MacQuisition is a forensic boot CD that makes an image of a Mac drive
 - After making an acquisition, examine the image of the file system
 - The tool you use depends on the image file format



Forensics Procedures in Mac (6 of 6)

- Acquisition Methods in macOS (cont'd)
 - Tools for working with a raw format image
 - BlackBag Technologies Macintosh Forensic Software
 - SubRosaSoft MacForensicsLab
 - Guidance Software EnCase
 - Recon Mac OS X Forensics with Palladin
 - X-Ways Forensics
 - AccessData FTK
 - First two tools can disable/enable Disk Arbitration
 - Being able to turn off the mount function in macOS
 - Allows you to connect a suspect drive to a Mac without a write-blocking device



Using Linux Forensics Tools

- 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



Installing Sleuth Kit and Autopsy (1 of 3)

- Download the most current source code from www.sleuthkit.org
- To run Sleuth Kit and Autopsy Browser, you need to have root privileges



Installing Sleuth Kit and Autopsy (2 of 3)

```
Ubuntu 16.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal Terminal File Edit View Search Terminal Help
student@Ubuntu16: ~
student@Ubuntu16:~$ autopsy -d /home/student/Documents/Evidence_Locker

=====
                        Autopsy Forensic Browser
                        http://www.sleuthkit.org/autopsy/
                        ver 2.24
=====

Evidence Locker: /home/student/Documents/Evidence_Locker
Start Time: Sun Jun 25 23:04:23 2017
Remote Host: localhost
Local Port: 9999

Open an HTML browser on the remote host and paste this URL in it:

    http://localhost:9999/autopsy

Keep this process running and use <ctrl-c> to exit
```

Figure 7-12 Starting Autopsy in Linux

Source: www.sleuthkit.org



Installing Sleuth Kit and Autopsy (3 of 3)

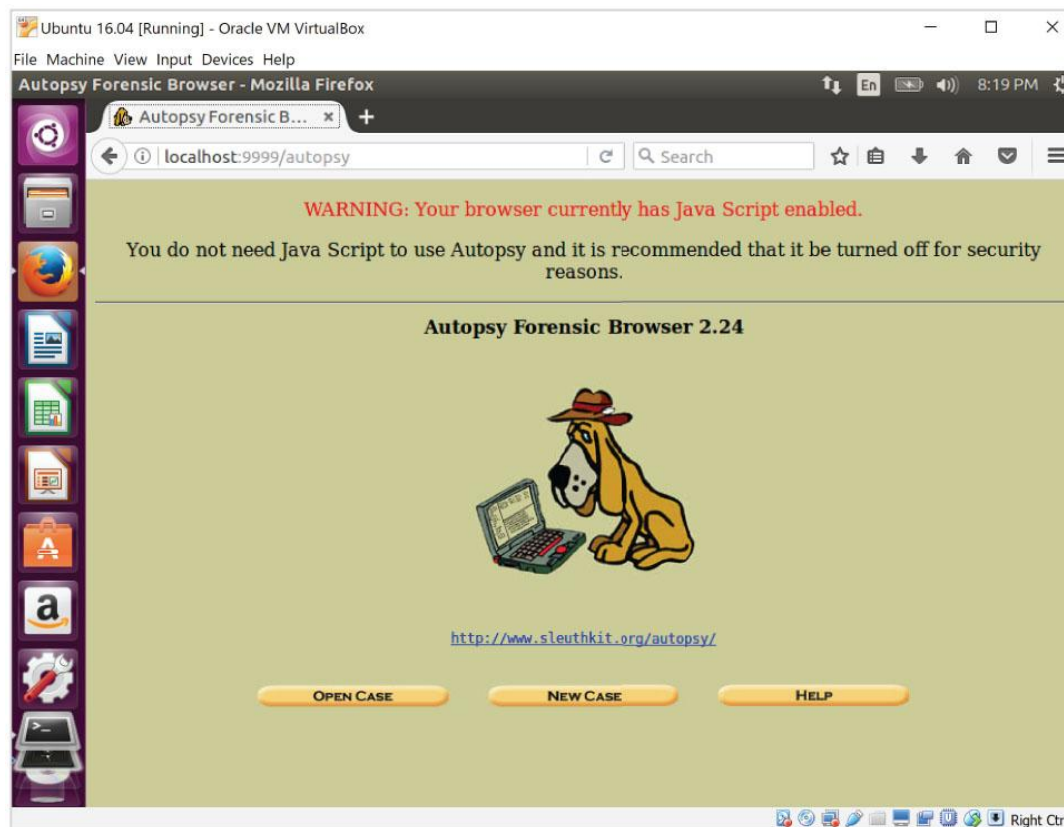


Figure 7-13 The Autopsy main window

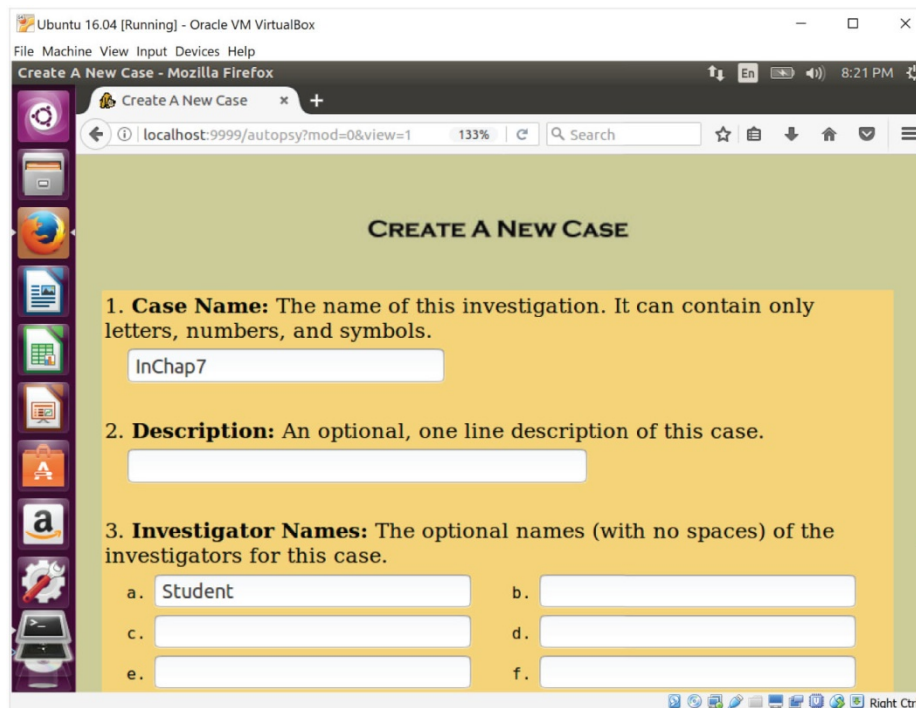
Source: www.sleuthkit.org



Examining a Case with Sleuth Kit and Autopsy (1 of 3)

- Follow instructions to use Sleuth Kit and Autopsy Browser to examine an older Linux file system
 - See Figures 7-14 and 7-15

Examining a Case with Sleuth Kit and Autopsy (2 of 3)



Ubuntu 16.04 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Create A New Case - Mozilla Firefox

localhost:9999/autopsy?mod=0&view=1 133% Search

CREATE A NEW CASE

- Case Name:** The name of this investigation. It can contain only letters, numbers, and symbols.
- Description:** An optional, one line description of this case.
- Investigator Names:** The optional names (with no spaces) of the investigators for this case.
 -
 -
 -
 -
 -
 -

Figure 7-14 The Create a New Case dialog box

Source: www.sleuthkit.org

Examining a Case with Sleuth Kit and Autopsy (3 of 3)

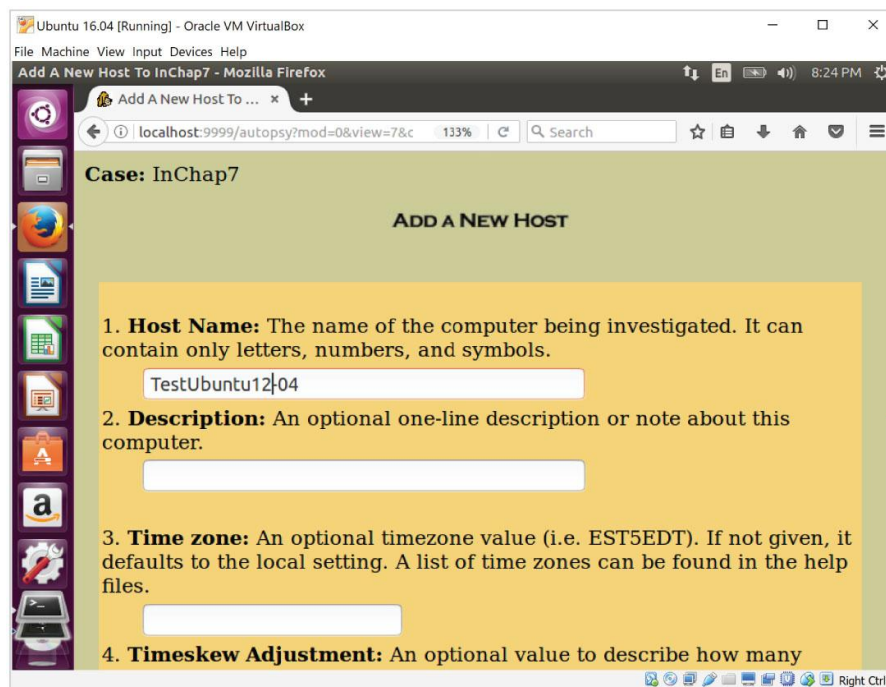


Figure 7-15 The Add a New Host dialog box

Source: www.sleuthkit.org



Summary (1 of 3)

- UNIX was created to be a multiuser, multithreaded, secure OS
- The Linux kernel is usually packaged with other software components, such as a GUI and applications
- Linux supports a wide range of file systems
- UNIX and Linux have four components defining the file system: boot block, superblock, inode block, and data block



Summary (2 of 3)

- In the Linux file system, a hard link is a pointer that allows accessing the same file by different filenames
- Before macOS, the file systems HFS and HFS+ were used
- In older version of macOS, a file consists of two parts: a data fork and a resource fork
- A volume is any storage medium used to store files



Summary (3 of 3)

- Plist files are preference files for installed applications on a macOS system
- In macOS, unified logging has been added for recording log files and includes new utilities to help forensics examiners
- The biggest challenge in acquiring images from macOS systems is often physical access to the drive
- Linux forensic tools are often freeware