CSEC 730 - Advanced Computer Forensics
Lab 1 - Linux Forensic Analysis

Please submit your answers (in PDF format) to the assignment submission folder under *myCourses* > *Assignments* by the deadline.

Objective

This lab will use *Sleuthkit* and *Autopsy (GUI)* to analyze a Linux image. You will practice the Sleuthkit tools at the data layer, metadata layer, file system layer, and file name layer in **part 1**. In **part 2** and **part 3**, you will learn Autopsy, a GUI-based front-end for Sleuthkit's basic features for forensic analysis. Finally, you will provide a short conclusion in **part 4** for your analysis of a given image. The steps provided below are only the guidelines. Please feel free to try a variety of Sleuthkit tools with different options to fully understand this powerful tool.

Case Scenario

Mark Watson works as a Director of Finance at an advertising firm. He suspects that a contractor, Frank Lewis, has read the confidential annual financial report (*Earnings.xls*) to influence his next contract with the firm. The IT administrator informed you that there is a Linux-based file server in the office where all employees save the official documents. Mark and Frank each have their own folders on this server.
You have been given the image of the hard drive, *Linux_Financial_Case.001*, to find any evidence that suggests Frank may have read Earnings.xls.

Evidence File

The acquired image "*Linux_Financial_Case.001.zip" is posted on myCourses > Content > Hands-on Labs > Lab 1.*

After you download and extract the image .zip file, validate both its md5 and sha1 hash values:

- MD5 (Linux Financial Case.001) = 7b39de0ca146c89ad73d1d421c8f7a05
- SHA1 (Linux Financial Case.001) = c7b06f006ff79711e692bd2620aba4cc2a4426d2

Deliverable

Answer all the exercise questions and include screenshots as supporting data if required.

****

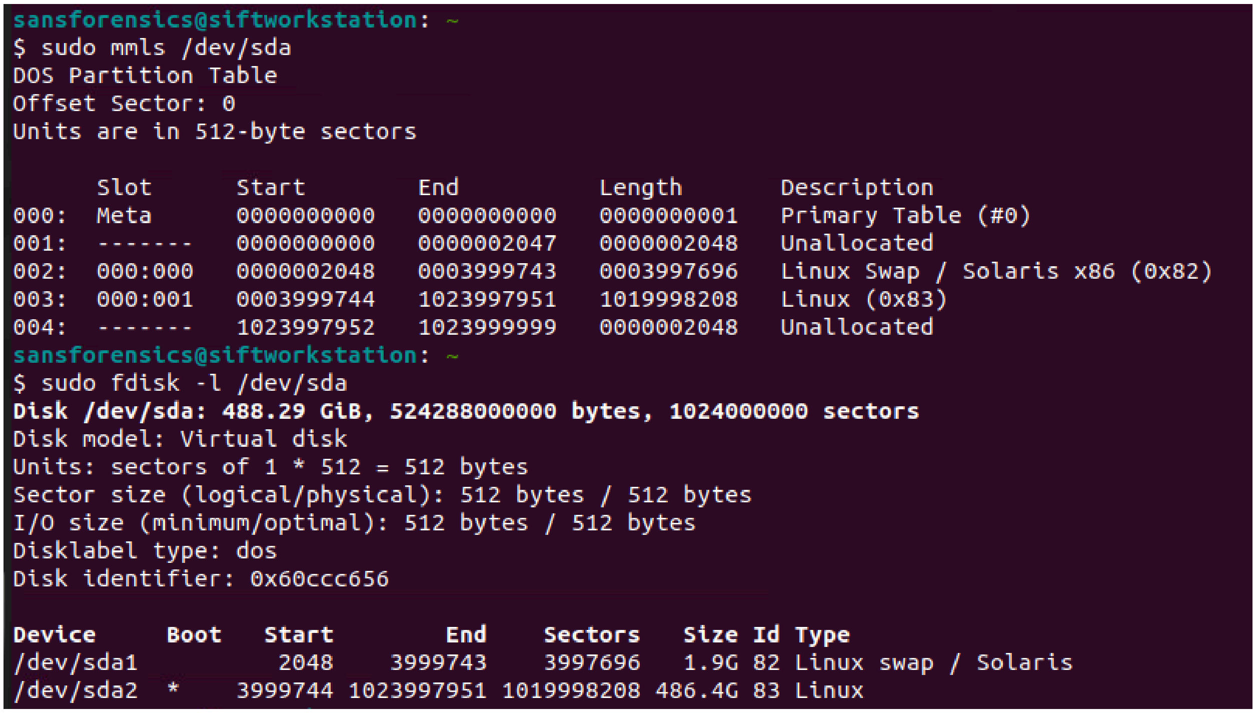
PART 1. Practice the sleuthkit command line tools to analyze the image “*Linux_Financial_Case.001*” (Each question is 2.5 points)

Instructions

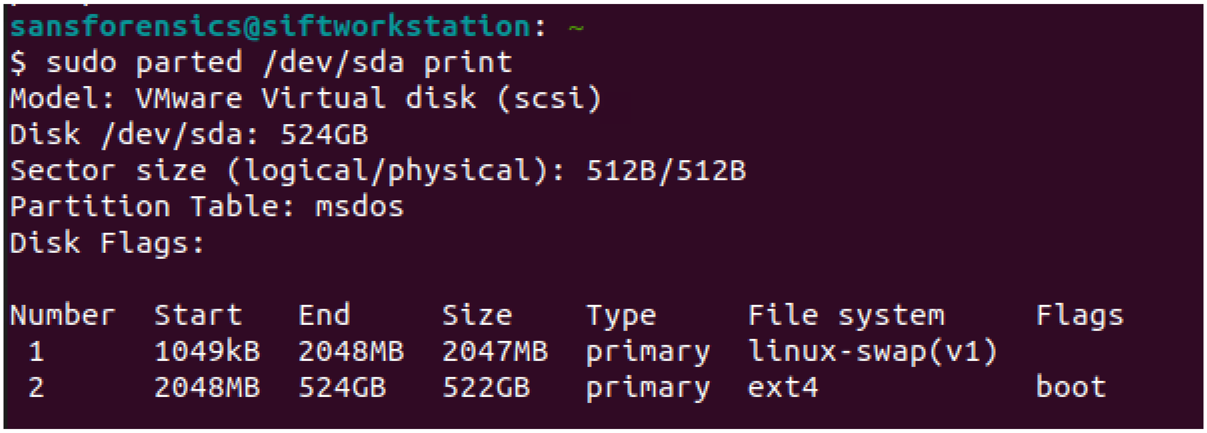
1. Launch the SIFT Workstation VM. The default login username is **sansforensics**, and the default password is **forensics.**

   Question 1. How many partitions do your SIFT VM’s */dev/sda* have? What is the offset of the starting sector for the “Linux” partition? Show the commands and screenshots.

   - There are two partitions, sda1 and sda2.



```
sansforensics@siftworkstation: ~
$ sudo mmls /dev/sda
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
001:  -------   0000000000   0000002047   0000002048   Unallocated
002:  000:000   0000002048   0003999743   0003997696   Linux Swap / Solaris x86 (0x82)
003:  000:001   0003999744   1023997951   1019998208   Linux (0x83)
004:  -------   1023997952   1023999999   0000002048   Unallocated
sansforensics@siftworkstation: ~
$ sudo fdisk -l /dev/sda
Disk /dev/sda: 488.29 GiB, 524288000000 bytes, 1024000000 sectors
Disk model: Virtual disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x60ccc656

Device     Boot   Start        End    Sectors   Size Id Type
/dev/sda1          2048    3999743    3997696   1.9G 82 Linux swap / Solaris
/dev/sda2  *    3999744 1023997951 1019998208 486.4G 83 Linux
```

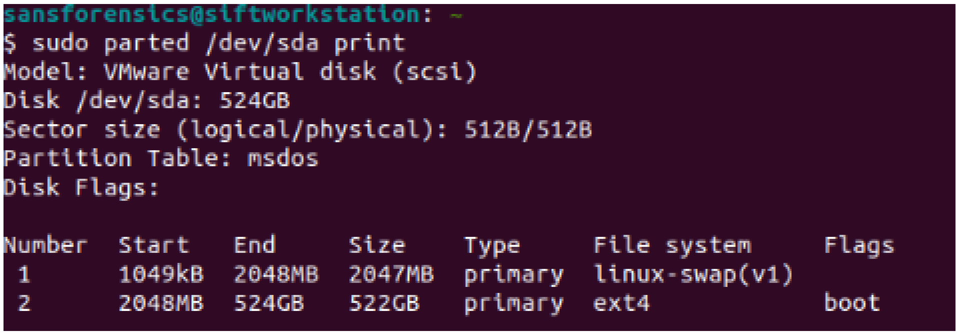


```
sansforensics@siftworkstation: ~
$ sudo parted /dev/sda print
Model: VMware Virtual disk (scsi)
Disk /dev/sda: 524GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type     File system     Flags
 1      1049kB  2048MB  2047MB  primary  linux-swap(v1)
 2      2048MB  524GB   522GB   primary  ext4            boot
```

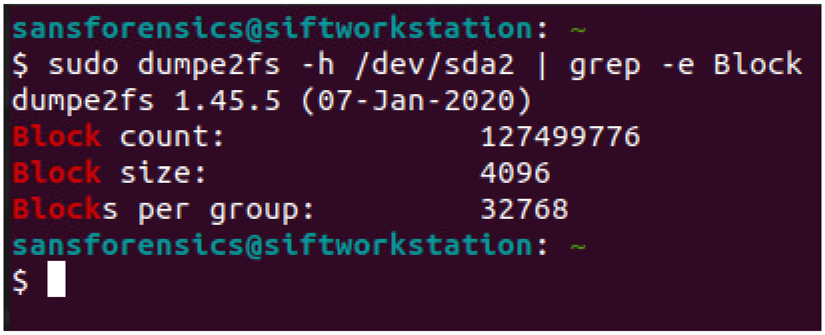
   - According to the fdisk command, the starting sector for the Linux partition is 3999744.

Question 2. Which file system does this “Linux” partition use? What is the block size of this “Linux” partition? Show the commands and screenshots.

- The file system that the Linux partition uses is ext4. The block size is 4096 B



```
sansforensics@siftworkstation: ~
$ sudo parted /dev/sda print
Model: VMware Virtual disk (scsi)
Disk /dev/sda: 524GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type     File system     Flags
 1      1049kB  2048MB  2047MB  primary  linux-swap(v1)
 2      2048MB  524GB   522GB   primary  ext4            boot
```



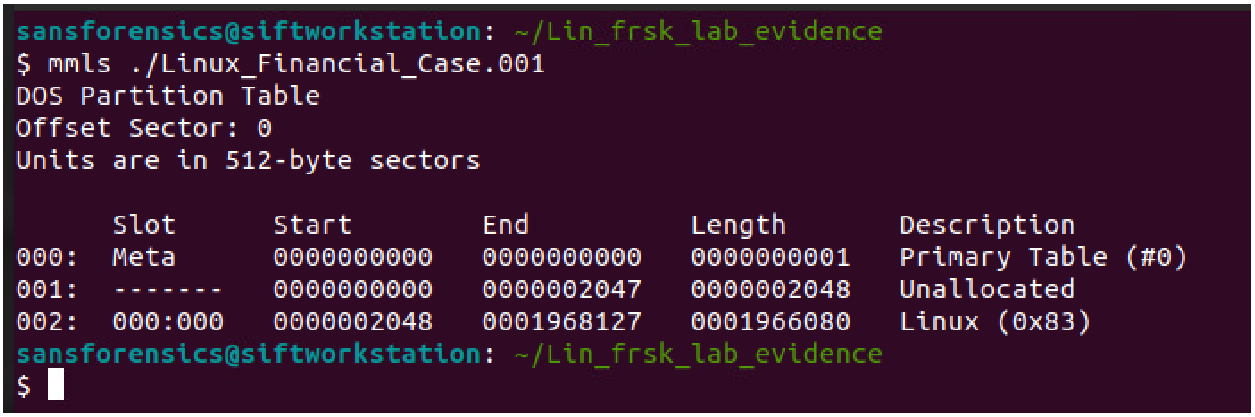
```
sansforensics@siftworkstation: ~
$ sudo dumpe2fs -h /dev/sda2 | grep -e Block
dumpe2fs 1.45.5 (07-Jan-2020)
Block count:              127499776
Block size:               4096
Blocks per group:         32768
sansforensics@siftworkstation: ~
$ 
```

The rest of the questions are related to the image “Linux_Financial_Case.001”

2. The image “Linux_Financial_Case.001” contains one partition. In able to analyze this image, you have to first find the offset of the starting sector for the partition.

   Question 3. What is the command along with the appropriate options you used?

   - The starting sector is 2048.

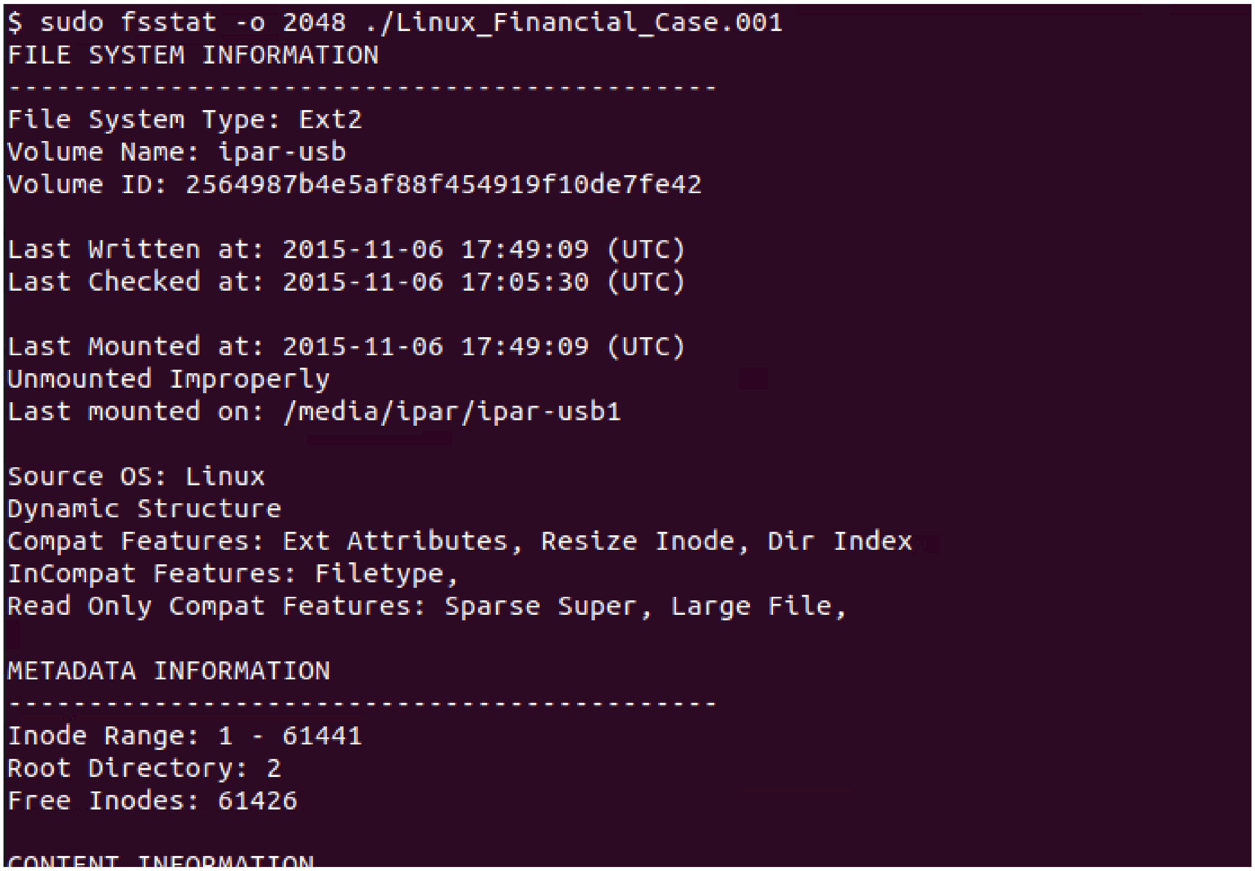


```
sansforensics@siftworkstation: ~/Lin_frsk_lab_evidence
$ mmls ./Linux_Financial_Case.001
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
001:  -------   0000000000   0000002047   0000002048   Unallocated
002:  000:000   0000002048   0001968127   0001966080   Linux (0x83)
sansforensics@siftworkstation: ~/Lin_frsk_lab_evidence
$ 
```

3. Find the image’s file system information (hint: you have to provide the offset you got from step 2.)

Question 4. What is the command along with the appropriate options you used to find the file system?

- The command I used is *“sudo fsstat -o 2048 ./Linux_Financial_Case.001”.*



```
$ sudo fsstat -o 2048 ./Linux_Financial_Case.001
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: Ext2
Volume Name: ipar-usb
Volume ID: 2564987b4e5af88f454919f10de7fe42

Last Written at: 2015-11-06 17:49:09 (UTC)
Last Checked at: 2015-11-06 17:05:30 (UTC)

Last Mounted at: 2015-11-06 17:49:09 (UTC)
Unmounted Improperly
Last mounted on: /media/ipar/ipar-usb1

Source OS: Linux
Dynamic Structure
Compat Features: Ext Attributes, Resize Inode, Dir Index
InCompat Features: Filetype,
Read Only Compat Features: Sparse Super, Large File,

METADATA INFORMATION
--------------------------------------------
Inode Range: 1 - 61441
Root Directory: 2
Free Inodes: 61426

CONTENT INFORMATION
```

Question 5. What type of file system is the image used?

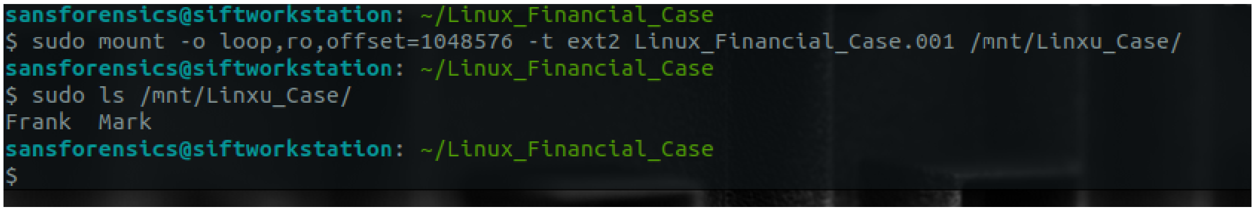
- The file is using Ext2 file system

Question 6. In what scenarios, you do NOT have to use the offset option –o for a sleuthkit command?

- The scenarios where you do NOT have to use to use the offset option -o for a sleuthkit command are:
    1. You are analyzing a single partition image rather than a whole disk image in case the partition starts at the beginning of the image file.
    2. When the filesystem is mounted, the operating system takes care of the offset and the tool works directly with the filesystem without needing to specify the start of the partition.
    3. The tool automatically detects the file system and its starting point. Some sleuthkit tools can auto-detect file system boundaries within an image.

Question 7. Provide the options of *mount* you will run to mount the Linux_Financial_Case.001 image’s partition for forensics investigation? (Show a screenshot of the mounted filesystem)

- sudo mount -o ro,loop,offset=1048576 -t ext2 ./Linux_Financial_Case.001 /mnt/LinuxFinancial

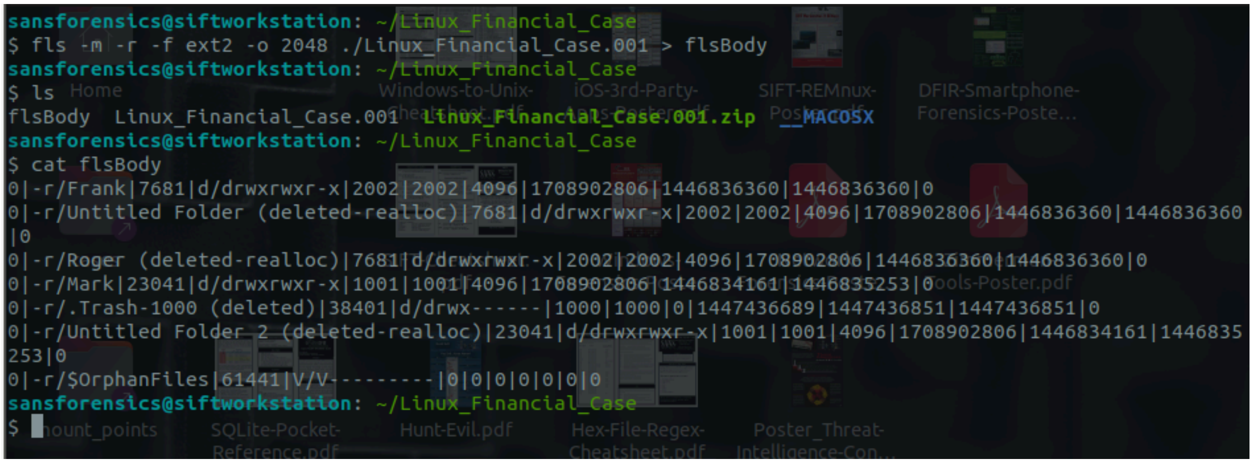


```
sansforensics@siftworkstation: ~/Linux_Financial_Case
$ sudo mount -o loop,ro,offset=1048576 -t ext2 Linux_Financial_Case.001 /mnt/Linxu_Case/
sansforensics@siftworkstation: ~/Linux_Financial_Case
$ sudo ls /mnt/Linxu_Case/
Frank  Mark
sansforensics@siftworkstation: ~/Linux_Financial_Case
$
```

4. Use *fls* to list the deleted files and directories, as a mactime body (-m), and save the file as *flsBody*.

   Question 8. What is the command along with appropriate options you used?

   - The command is fls -m -r -f ext2 -o 2048 ./Linux_Finanical_Case.001 > flsBody

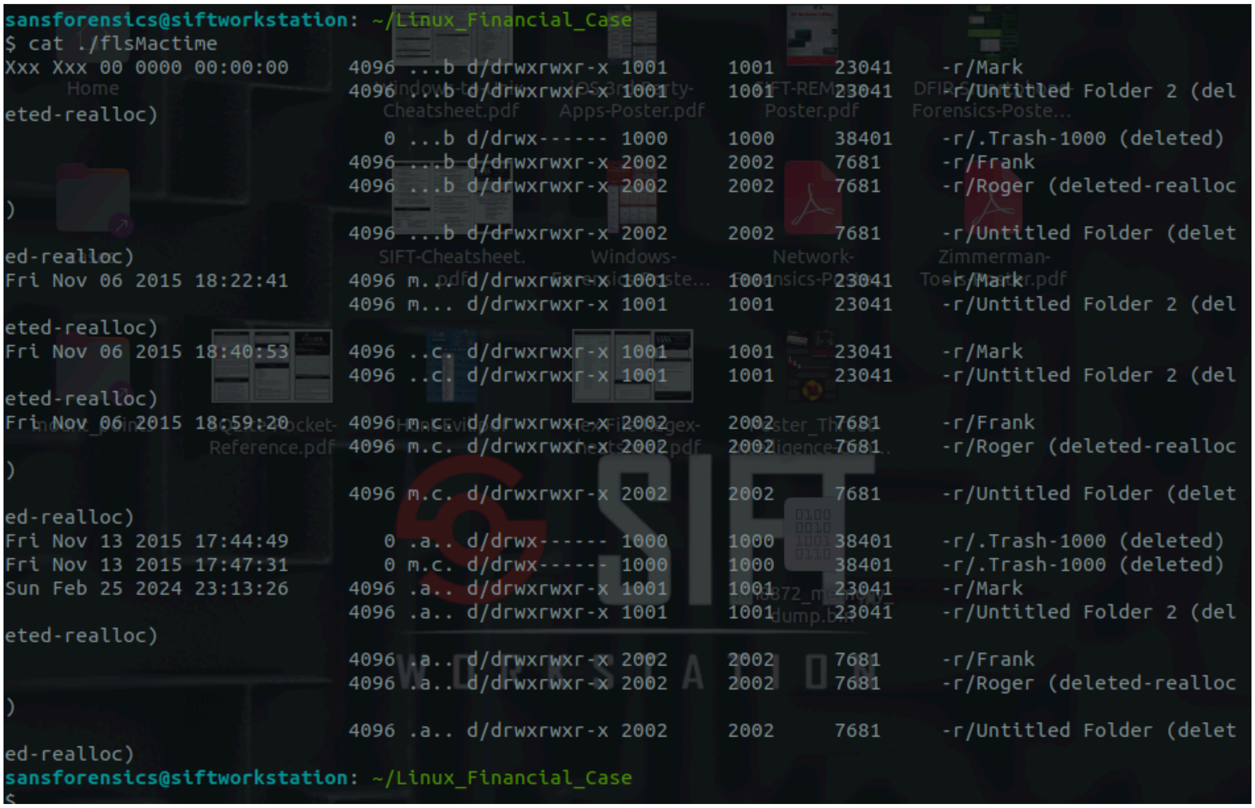


```
sansforensics@siftworkstation: ~/Linux_Financial_Case
$ fls -m -r -f ext2 -o 2048 ./Linux_Financial_Case.001 > flsBody
sansforensics@siftworkstation: ~/Linux_Financial_Case
$ ls
flsBody  Linux_Financial_Case.001  Linux_Financial_Case.001.zip  __MACOSX
sansforensics@siftworkstation: ~/Linux_Financial_Case
$ cat flsBody
0|-r/Frank|7681|d/drwxrwxr-x|2002|2002|4096|1708902806|1446836360|1446836360|0
0|-r/Untitled Folder (deleted-realloc)|7681|d/drwxrwxr-x|2002|2002|4096|1708902806|1446836360|1446836360
|0
0|-r/Roger (deleted-realloc)|7681|d/drwxrwxr-x|2002|2002|4096|1708902806|1446836360|1446836360|0
0|-r/Mark|23041|d/drwxrwxr-x|1001|1001|4096|1708902806|1446834161|1446835253|0
0|-r/.Trash-1000 (deleted)|38401|d/drwx------|1000|1000|0|1447436689|1447436851|1447436851|0
0|-r/Untitled Folder 2 (deleted-realloc)|23041|d/drwxrwxr-x|1001|1001|4096|1708902806|1446834161|1446835
253|0
0|-r/$OrphanFiles|61441|V/V---------|0|0|0|0|0|0|0
sansforensics@siftworkstation: ~/Linux_Financial_Case
$
```

5. Use Sleuthkit's *mactime* to create a timeline of *flsBody*. Save the timeline in a file called *flsMactime* and examine the timeline.

   Question 9. What is the command along with the appropriate options you used? Include a screenshot of a part of the content of *flsMactime.*

   - The command that was used is mactime -b flsBody > flsMactime

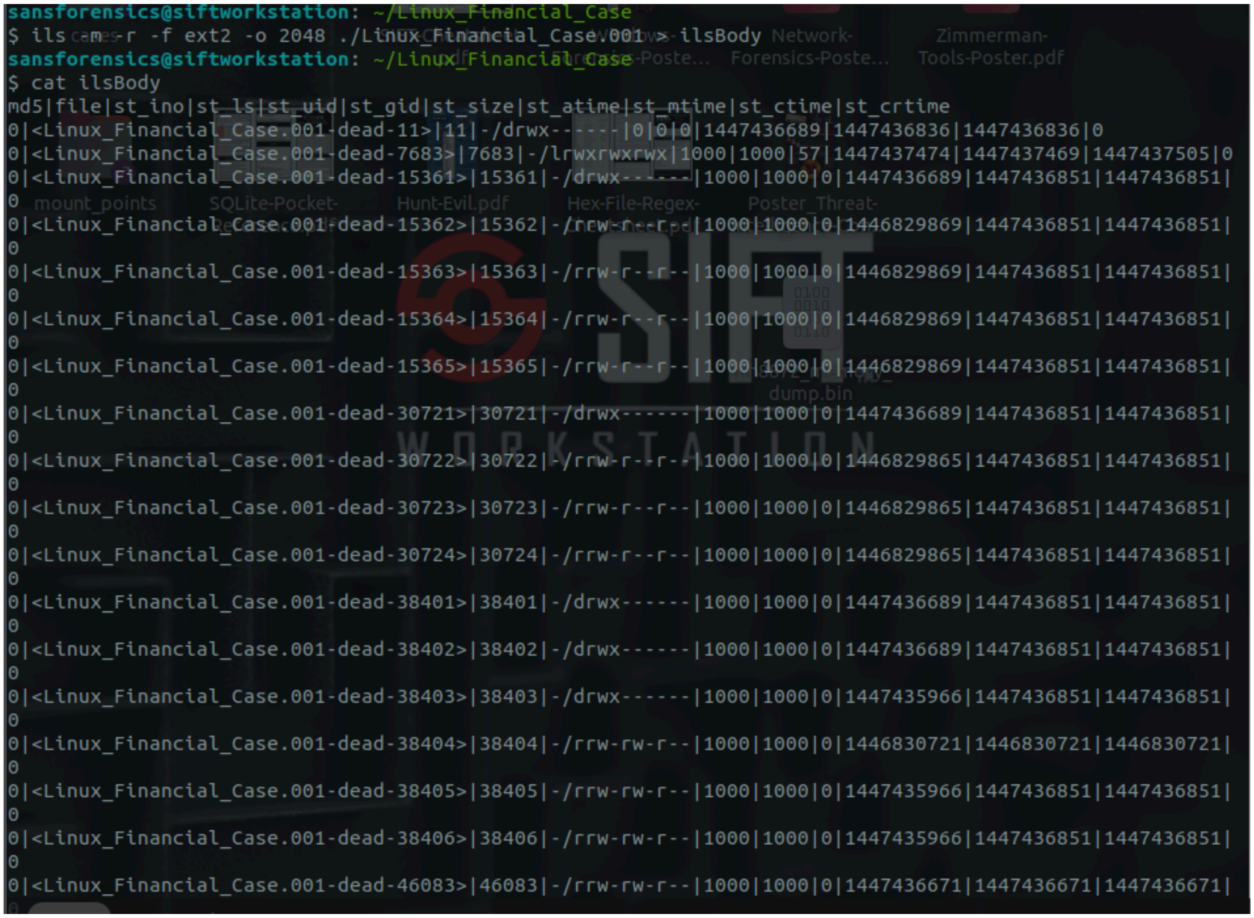


```
sansforensics@siftworkstation: ~/Linux_Financial_Case
$ cat ./flsMactime
Xxx Xxx 00 0000 00:00:00     4096 ...b d/drwxrwxr-x 1001     1001     23041    -r/Mark
                             4096 ...b d/drwxrwxr-x 1001     1001     23041    -r/Untitled Folder 2 (del
eted-realloc)
                                0 ...b d/drwx------ 1000     1000     38401    -r/.Trash-1000 (deleted)
                             4096 ...b d/drwxrwxr-x 2002     2002     7681     -r/Frank
                             4096 ...b d/drwxrwxr-x 2002     2002     7681     -r/Roger (deleted-realloc
)
                             4096 ...b d/drwxrwxr-x 2002     2002     7681     -r/Untitled Folder (delet
ed-realloc)
Fri Nov 06 2015 18:22:41     4096 m... d/drwxrwxr-x 1001     1001     23041    -r/Mark
                             4096 m... d/drwxrwxr-x 1001     1001     23041    -r/Untitled Folder 2 (del
eted-realloc)
Fri Nov 06 2015 18:40:53     4096 ..c. d/drwxrwxr-x 1001     1001     23041    -r/Mark
                             4096 ..c. d/drwxrwxr-x 1001     1001     23041    -r/Untitled Folder 2 (del
eted-realloc)
Fri Nov 06 2015 18:59:20     4096 m.c. d/drwxrwxr-x 2002     2002     7681     -r/Frank
                             4096 m.c. d/drwxrwxr-x 2002     2002     7681     -r/Roger (deleted-realloc
)
                             4096 m.c. d/drwxrwxr-x 2002     2002     7681     -r/Untitled Folder (delet
ed-realloc)
Fri Nov 13 2015 17:44:49        0 .a.. d/drwx------ 1000     1000     38401    -r/.Trash-1000 (deleted)
Fri Nov 13 2015 17:47:31        0 m.c. d/drwx------ 1000     1000     38401    -r/.Trash-1000 (deleted)
Sun Feb 25 2024 23:13:26     4096 .a.. d/drwxrwxr-x 1001     1001     23041    -r/Mark
                             4096 .a.. d/drwxrwxr-x 1001     1001     23041    -r/Untitled Folder 2 (del
eted-realloc)
                             4096 .a.. d/drwxrwxr-x 2002     2002     7681     -r/Frank
                             4096 .a.. d/drwxrwxr-x 2002     2002     7681     -r/Roger (deleted-realloc
)
                             4096 .a.. d/drwxrwxr-x 2002     2002     7681     -r/Untitled Folder (delet
ed-realloc)
sansforensics@siftworkstation: ~/Linux_Financial_Case
$
```

6. Use *ils* to list the inode information for all deleted files, as a mactime body (-m), and save the file as *ilsBody*.

   Question 10. What is the command along with the appropriate options you used?

   - ils -m -r -f ext2 -o 2048./Linux_Financial_Case.001 > ilsBody

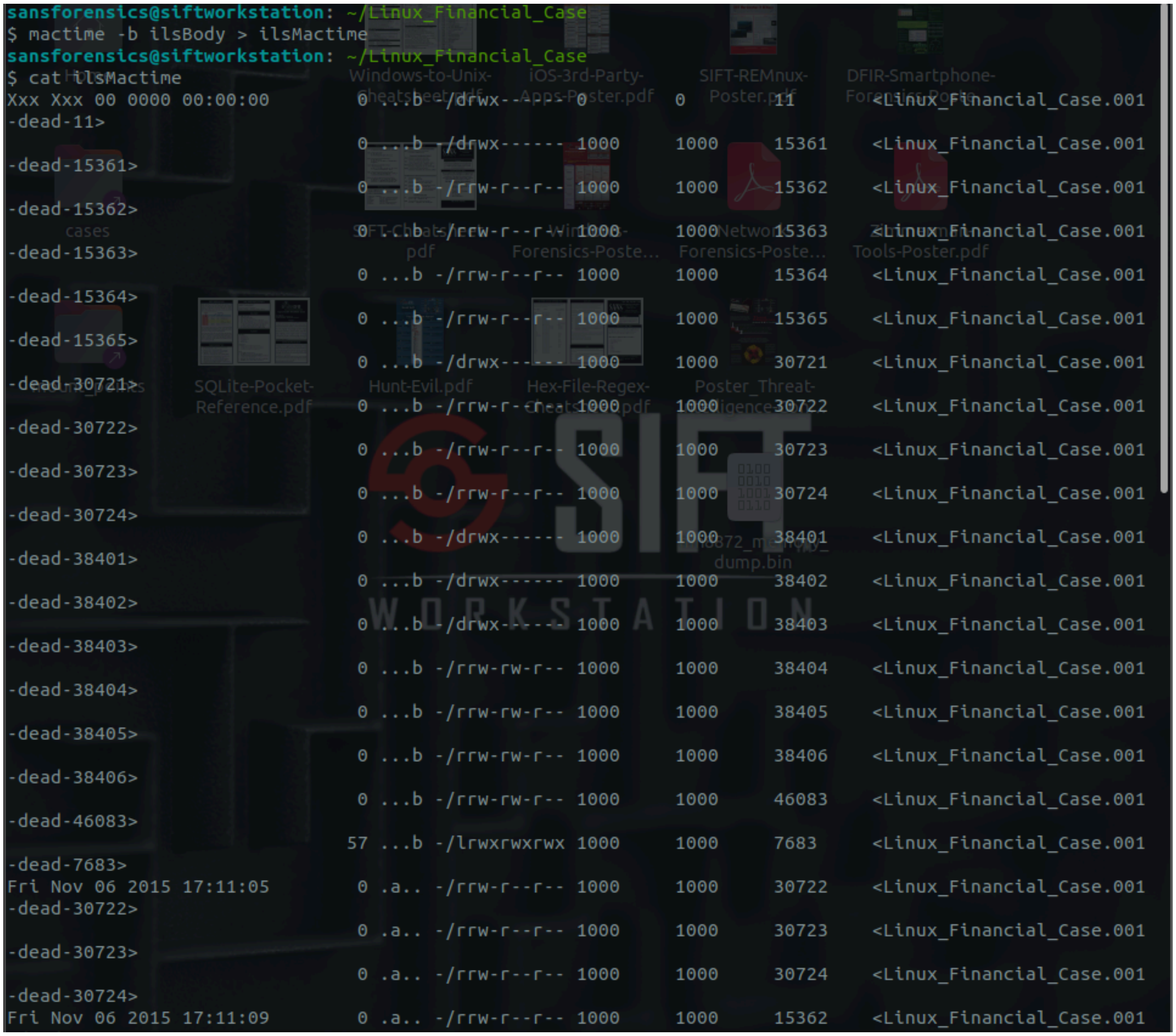


```
sansforensics@siftworkstation: ~/Linux_Financial_Case
$ ils -m -r -f ext2 -o 2048 ./Linux_Financial_Case.001 > ilsBody
sansforensics@siftworkstation: ~/Linux_Financial_Case
$ cat ilsBody
md5|file|st_ino|st_ls|st_uid|st_gid|st_size|st_atime|st_mtime|st_ctime|st_crtime
0|<Linux_Financial_Case.001-dead-11>|11|-/drwx------|0|0|0|1447436689|1447436836|1447436836|0
0|<Linux_Financial_Case.001-dead-7683>|7683|-/lrwxrwxrwx|1000|1000|57|1447437474|1447437469|1447437505|0
0|<Linux_Financial_Case.001-dead-15361>|15361|-/drwx------|1000|1000|0|1447436689|1447436851|1447436851|
0
0|<Linux_Financial_Case.001-dead-15362>|15362|-/rrw-r--r--|1000|1000|0|1446829869|1447436851|1447436851|
0
0|<Linux_Financial_Case.001-dead-15363>|15363|-/rrw-r--r--|1000|1000|0|1446829869|1447436851|1447436851|
0
0|<Linux_Financial_Case.001-dead-15364>|15364|-/rrw-r--r--|1000|1000|0|1446829869|1447436851|1447436851|
0
0|<Linux_Financial_Case.001-dead-15365>|15365|-/rrw-r--r--|1000|1000|0|1446829869|1447436851|1447436851|
0
0|<Linux_Financial_Case.001-dead-30721>|30721|-/drwx------|1000|1000|0|1447436689|1447436851|1447436851|
0
0|<Linux_Financial_Case.001-dead-30722>|30722|-/rrw-r--r--|1000|1000|0|1446829865|1447436851|1447436851|
0
0|<Linux_Financial_Case.001-dead-30723>|30723|-/rrw-r--r--|1000|1000|0|1446829865|1447436851|1447436851|
0
0|<Linux_Financial_Case.001-dead-30724>|30724|-/rrw-r--r--|1000|1000|0|1446829865|1447436851|1447436851|
0
0|<Linux_Financial_Case.001-dead-38401>|38401|-/drwx------|1000|1000|0|1447436689|1447436851|1447436851|
0
0|<Linux_Financial_Case.001-dead-38402>|38402|-/drwx------|1000|1000|0|1447436689|1447436851|1447436851|
0
0|<Linux_Financial_Case.001-dead-38403>|38403|-/drwx------|1000|1000|0|1447435966|1447436851|1447436851|
0
0|<Linux_Financial_Case.001-dead-38404>|38404|-/rrw-rw-r--|1000|1000|0|1446830721|1446830721|1446830721|
0
0|<Linux_Financial_Case.001-dead-38405>|38405|-/rrw-rw-r--|1000|1000|0|1447435966|1447436851|1447436851|
0
0|<Linux_Financial_Case.001-dead-38406>|38406|-/rrw-rw-r--|1000|1000|0|1447435966|1447436851|1447436851|
0
0|<Linux_Financial_Case.001-dead-46083>|46083|-/rrw-rw-r--|1000|1000|0|1447436671|1447436671|1447436671|
```

7. Use Sleuthkit's *mactime* to create a timeline of *ilsBody*. Save the timeline in a file called *ilsMactime* and examine the timeline.

   Question 11. What is the command along with appropriate options you used? Include a screenshot of a part of the content of *ilsMactime.*

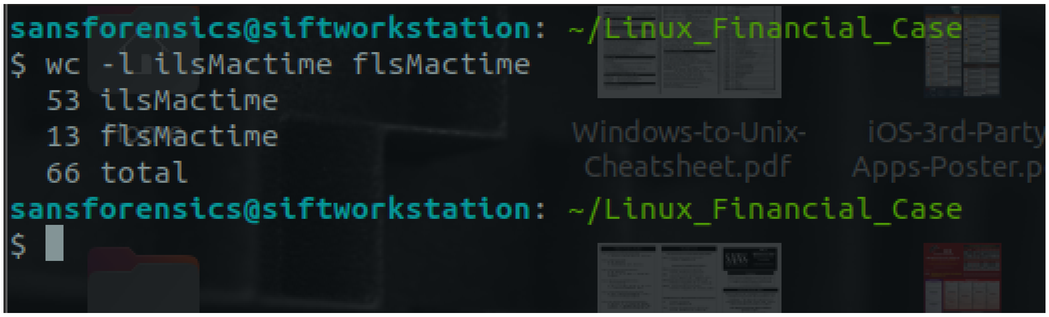
   - mactime -b ilsBody > ilsMactime



```
sansforensics@siftworkstation: ~/Linux_Financial_Case
$ mactime -b ilsBody > ilsMactime
sansforensics@siftworkstation: ~/Linux_Financial_Case
$ cat ilsMactime
Xxx Xxx 00 0000 00:00:00        0 ...b -/drwx------ 0      0      11       <Linux_Financial_Case.001
-dead-11>
                                0 ...b -/drwx------ 1000   1000   15361    <Linux_Financial_Case.001
-dead-15361>
                                0 ...b -/rrw-r--r-- 1000   1000   15362    <Linux_Financial_Case.001
-dead-15362>
                                0 ...b -/rrw-r--r-- 1000   1000   15363    <Linux_Financial_Case.001
-dead-15363>
                                0 ...b -/rrw-r--r-- 1000   1000   15364    <Linux_Financial_Case.001
-dead-15364>
                                0 ...b -/rrw-r--r-- 1000   1000   15365    <Linux_Financial_Case.001
-dead-15365>
                                0 ...b -/drwx------ 1000   1000   30721    <Linux_Financial_Case.001
-dead-30721>
                                0 ...b -/rrw-r--r-- 1000   1000   30722    <Linux_Financial_Case.001
-dead-30722>
                                0 ...b -/rrw-r--r-- 1000   1000   30723    <Linux_Financial_Case.001
-dead-30723>
                                0 ...b -/rrw-r--r-- 1000   1000   30724    <Linux_Financial_Case.001
-dead-30724>
                                0 ...b -/drwx------ 1000   1000   38401    <Linux_Financial_Case.001
-dead-38401>
                                0 ...b -/drwx------ 1000   1000   38402    <Linux_Financial_Case.001
-dead-38402>
                                0 ...b -/drwx------ 1000   1000   38403    <Linux_Financial_Case.001
-dead-38403>
                                0 ...b -/rrw-rw-r-- 1000   1000   38404    <Linux_Financial_Case.001
-dead-38404>
                                0 ...b -/rrw-rw-r-- 1000   1000   38405    <Linux_Financial_Case.001
-dead-38405>
                                0 ...b -/rrw-rw-r-- 1000   1000   38406    <Linux_Financial_Case.001
-dead-38406>
                                0 ...b -/rrw-rw-r-- 1000   1000   46083    <Linux_Financial_Case.001
-dead-46083>
                               57 ...b -/lrwxrwxrwx 1000   1000   7683     <Linux_Financial_Case.001
-dead-7683>
Fri Nov 06 2015 17:11:05        0 .a.. -/rrw-r--r-- 1000   1000   30722    <Linux_Financial_Case.001
-dead-30722>
                                0 .a.. -/rrw-r--r-- 1000   1000   30723    <Linux_Financial_Case.001
-dead-30723>
                                0 .a.. -/rrw-r--r-- 1000   1000   30724    <Linux_Financial_Case.001
-dead-30724>
Fri Nov 06 2015 17:11:09        0 .a.. -/rrw-r--r-- 1000   1000   15362    <Linux_Financial_Case.001
```

8. Compare the number of entries from *ilsMactime* and from *flsMactime*.

Question 12. Do *ilsMactime* and *flsMactme* have the same number of entries? Explain your findings



```
sansforensics@siftworkstation: ~/Linux_Financial_Case
$ wc -l ilsMactime flsMactime
  53 ilsMactime
  13 flsMactime
  66 total
sansforensics@siftworkstation: ~/Linux_Financial_Case
$ 
```

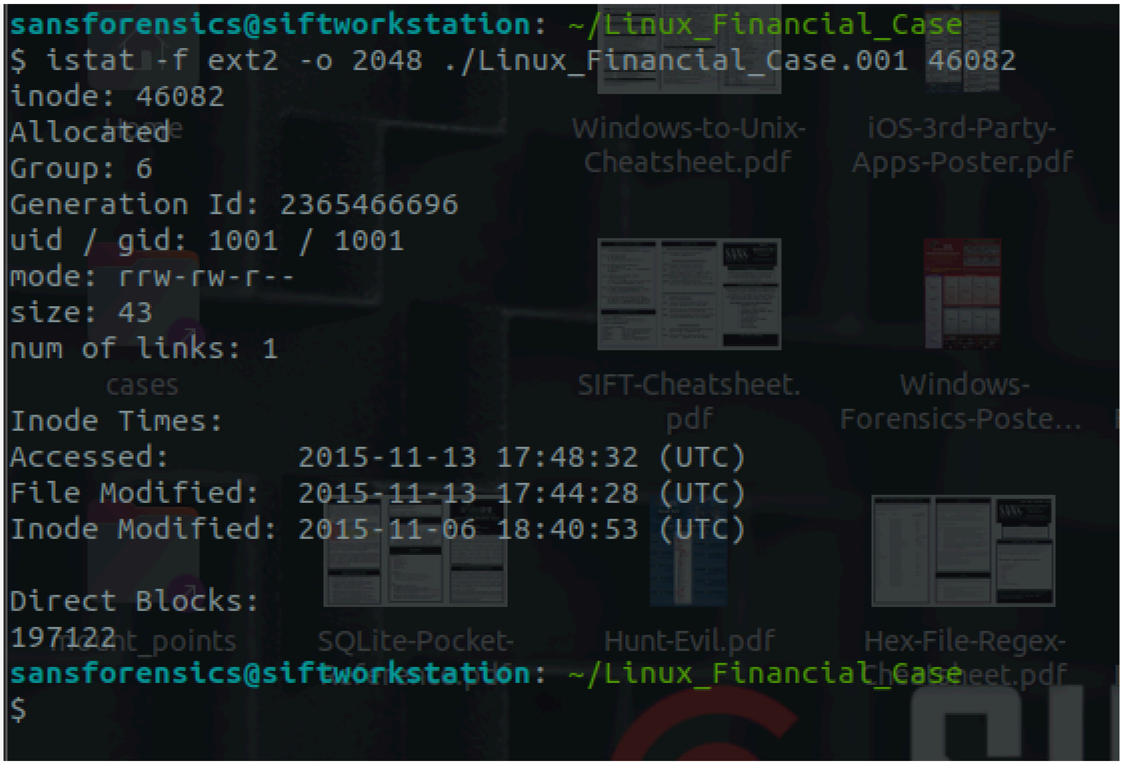
.

- No, they don't have the same number of entries. There are discrepancies between the two files. It may be that two tools might have different mechanism of listing delected files.

9. use *istat* to view the details of the inode 46082.

Question 13. What is the command along with appropriate options you used? Include a screenshot

. istat -f ext2 -o 2048 ./Linux_Financial_Case.001 46082



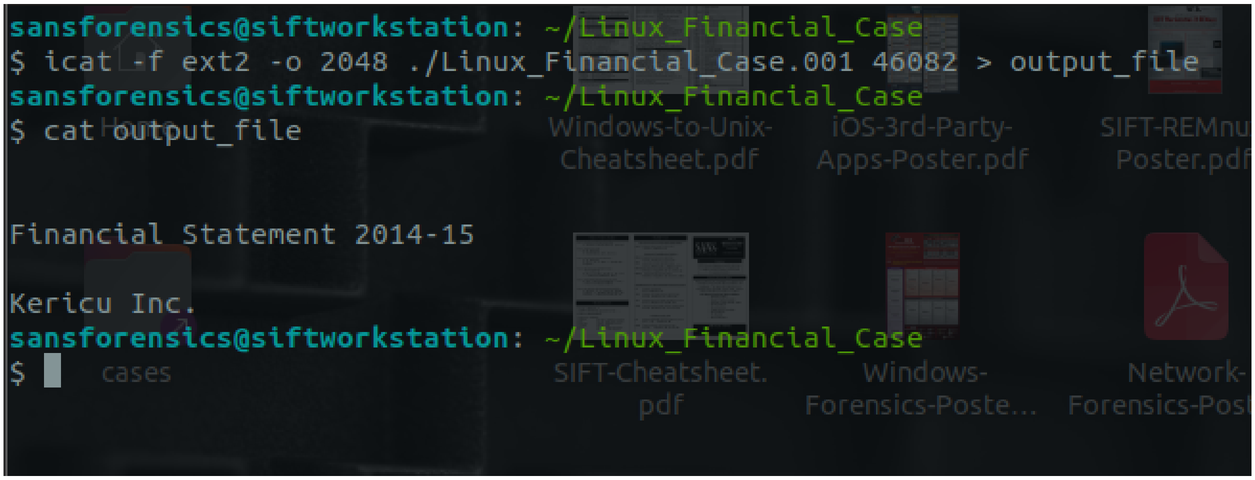
```
sansforensics@siftworkstation: ~/Linux_Financial_Case
$ istat -f ext2 -o 2048 ./Linux_Financial_Case.001 46082
inode: 46082
Allocated
Group: 6
Generation Id: 2365466696
uid / gid: 1001 / 1001
mode: rrw-rw-r--
size: 43
num of links: 1

Inode Times:
Accessed:       2015-11-13 17:48:32 (UTC)
File Modified:  2015-11-13 17:44:28 (UTC)
Inode Modified: 2015-11-06 18:40:53 (UTC)

Direct Blocks:
197122
sansforensics@siftworkstation: ~/Linux_Financial_Case
$
```

Windows-to-Unix-Cheatsheet.pdf
iOS-3rd-Party-Apps-Poster.pdf
cases
SIFT-Cheatsheet.pdf
Windows-Forensics-Poste...
mount_points
SQLite-Pocket-
Hunt-Evil.pdf
Hex-File-Regex-Cheatsheet.pdf

10. use *icat* to dump out data from the inode 46082.

Question 14. What is the command along with appropriate options you used?

- icat -f ext2 -o 2048 ./Linux_Financial_Case.001 46082 > output_file



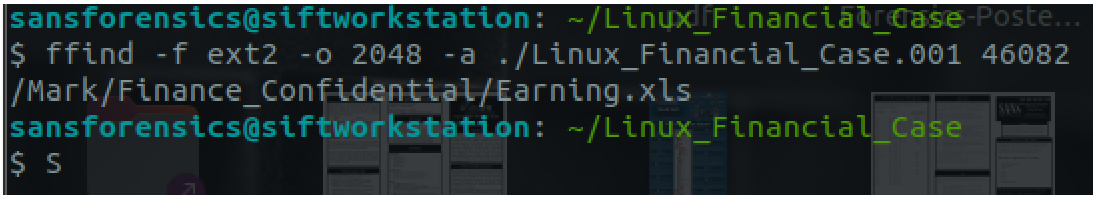
```
sansforensics@siftworkstation: ~/Linux_Financial_Case
$ icat -f ext2 -o 2048 ./Linux_Financial_Case.001 46082 > output_file
sansforensics@siftworkstation: ~/Linux_Financial_Case
$ cat output_file


Financial Statement 2014-15

Kericu Inc.
sansforensics@siftworkstation: ~/Linux_Financial_Case
$ 
```

Windows-to-Unix-Cheatsheet.pdf
iOS-3rd-Party-Apps-Poster.pdf
SIFT-REMnu Poster.pdf
cases
SIFT-Cheatsheet.pdf
Windows-Forensics-Poste...
Network-Forensics-Post

---

11. Use *ffind* to find the file's filename that has the inode 46082.

    Question 15. What is the command along with appropriate options you used? Include a screenshot.



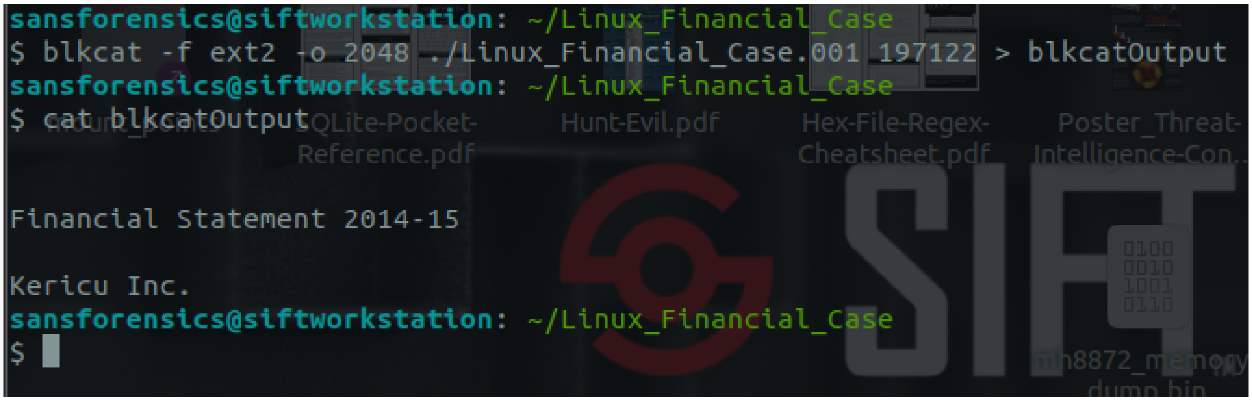
```
sansforensics@siftworkstation: ~/Linux_Financial_Case
$ ffind -f ext2 -o 2048 -a ./Linux_Financial_Case.001 46082
/Mark/Finance_Confidential/Earning.xls
sansforensics@siftworkstation: ~/Linux_Financial_Case
$ S
```

- ffind -f ext2 -o 2048 -a ./Linux_Financial_Case.001 46082

---

12. Use *blkcat* to dump out the data content of the datablock 197122

    Question 16. What is the command along with the appropriate options you used?



```
sansforensics@siftworkstation: ~/Linux_Financial_Case
$ blkcat -f ext2 -o 2048 ./Linux_Financial_Case.001 197122 > blkcatOutput
sansforensics@siftworkstation: ~/Linux_Financial_Case
$ cat blkcatOutput


Financial Statement 2014-15

Kericu Inc.
sansforensics@siftworkstation: ~/Linux_Financial_Case
$ 
```

SQLite-Pocket-Reference.pdf
Hunt-Evil.pdf
Hex-File-Regex-Cheatsheet.pdf
Poster_Threat-Intelligence-Con...
SIFT
0100 0010 1001 0110
mh8872_memory dump.bin

- blkcat -f ext2 -o 2048 ./Linux_Financial_Case.001 197122 > blkcatOutput

---

Question 17. If a file with the inode 100 uses two block addresses, *block 1000* and *block 1001*, will "icat -f ext2 image 100" dump out the same content as the command "blkcat –f ext2 image 1000"? Explain your answer.

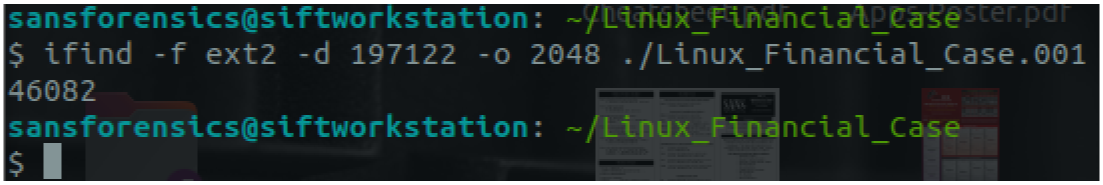
- No they will not because the two commands are used for two different purposes. The icat command is used to display content of a file by inode number, and the command will retrieve the contents of the file associated with indoe 100. The blkcat command is used to directly display the contents of a specific block, so the command for it will only display the contents of the block with address 1000. The data is not interpreted by it as a component of a file's contents. Running blkcat on a single block address will therefore only display the content of that specific block, which might not accurately represent the contents of the full file, if the data in the file spans numerous blocks.

  In conclusion, "blkcat" is used to display the raw data of a particular block, whereas "icat" is used to display the contents of a file. Whereas "blkcat" only shows the content of a single block, which might not adequately represent the contents of the complete file, "icat" will display the concatenated contents of all the blocks in a file that use multiple blocks.

---

13. Use *ifind* to find the inode number that one of its correspondent data blocks is 197122.

Question 18. What is the command along with appropriate options you used? Also provide a case scenario that shows the usefulness of *ifind*.



```
sansforensics@siftworkstation: ~/Linux_Financial_Case
$ ifind -f ext2 -d 197122 -o 2048 ./Linux_Financial_Case.001
46082
sansforensics@siftworkstation: ~/Linux_Financial_Case
$ 
```

- ifind -f ext2 -d 197122 -o 2048 ./Linux_Financial_Case.001.
- Assume for the moment that you are looking into a hacked system forensic case. You discover a suspicious data block with the number 197122 while conducting your analysis. You need to follow this data block back to its accompanying file or directory in order to obtain additional context, even though you fear it may be connected to criminal behavior.

  In this case, the inode number linked to data block 197122 can be found by using ifind to search the file system image. Once you know the inode number, you can extract more details about the file or directory associated to that inode using other forensic tools like istat or icat. You can use this information to further your investigation and comprehend the nature of the suspicious conduct.

PART 2. Use Autopsy to analyze "*Linux_Financial_Case.001*" *case* (Each question is 2.5 points)

Instructions

To Start autopsy:
Start a terminal (go to applications -> Accessories->Terminal) and type in
$ sudo autopsy
While this process is running, open a web browser point it
to the URL indicated – http://localhost:9999/autopsy

Click on “New Case”.

Enter “linux_financial_case” as the case name, you may fill in other optional information, then click “New Case”. Confirm the information and click “OK”. (Names with spaces will not work.)

Click “Add Host”.
Enter “Host1” under “Host Name” and “EST” under “Timezone” and click “Add Host”.

Confirm the information and click “ADD HOST”.
Click “Add Image”.
Click “ADD IMAGE FILE”.

Select “Disk” since this image contains a disk image (vs a partition).

In “Location” type the path to the image file “*Linux_Financial_Case.001*”.

Explore the various “Import Methods”.

Review the options for checking/creating md5’s and select the appropriate entry based on the information you currently have.

Question 1: Which option did you choose and why?

- I choose the copy option because it preserves the original image. Allowing us to use it for future references and eliminates the risk of corruption if a system failure occurs during the import process.

Autopsy identifies the partition and the file system type of this partition.

Question 2: Which Sleuthkit tool does Autopsy use to display the partition table information?

- mmls tools was used.

Question 3: Which Sleuthkit tool does Autopsy use to determine the file system type of this partition?

- Autopsy uses fsstat to determine the file system type of this partition.

Click “Add” to add the image to host 1. And confirm the information and click “OK”.

Now Autopsy should have mounted the partition.

Select the partition, click “Analysis” and choose “FILE ANALYSIS” tab.

In this mode, you can view file and directory metadata and file content.
Click the inode of directory *Mark*, 23041, to see the detail information about Mark’s directory.

Go to *Mark/Finance_Confidential* directory, and click on *Earning.xls* file. In the information window at the bottom, explore the “display”, “report”, “export” links.

Question 4 What information do you get from “display” and “report”? What does “export” do?

- The display tab allows to view the information oand details of the file. The Report generates a comprehensive report based on the analys conducted within Autopsy on the file, for example, details about file system, partitions, file metadata, file contents, and etc. The export allows you to basically save the file from the autopsy from case of analysis.

From here you can recover any of the files shown, including deleted ones if the content has not been overwritten.

Question 5: How can you determine that a file has been deleted?

- You can determine that a file has been deleted by looking at if a file has been flagged as deleted or not. There is a column that says DEL and if there is a check mark there next to a file, then it is delected.

Click “File Type”. Then click “Sort Files by Type”. Then click “OK”.

Question 6: How is the “Sort Files by Type” formation useful in an investigation?

- It is useful because it helps to categorize files based on their types, making it easier for intfigators to navigate through large amounts of data. Identify relevant files, and unexpected files, and priotize the files that needs to be checked.

To view the sorted file, click on “View Sorted Files” and copy/paste the URL into a browser.

Click on “Meta Data” and provide a valid inode number.

Question 7: Knowing an inode number, which Sleuthkit tool does Autopsy use to determine the data blocks referenced by the inode?

- Autopsy uses istat.

Click on the “Image Details” tab and read the information given.

Question 8: What information can you get from this window? Where does Autopsy get this information from?

- Autopsy gets this information form the file system metadata present in the disk image. Autopsy might also get the information from using the tools as well.

Click the “Close” tab to close the “Analyze”, and you will be back to the “Host Manager”.

Select the partition and click “File Activity Timelines”

Click “Create Data File”.
Select the disk partition and click “OK”, and confirm the information.

Question 9: What Sleuthkit command line tool(s) was/were used to generate the body file?

- The fls tool.

Click “OK”.

Now we have the body file, we can sort the body file to generate a timeline.
Autopsy this version has a bug for creating a timeline. To fix it, you will run “sudo cp /usr/bin/mactime /usr/bin/mactime-sleuthkit”.

In the “Create Timeline” window, you can select the starting and ending dates of file activity that you want to see, for example, from Jan. 2015 to Jan. 2016.

Note the sorted information. Click the links at the top to look at other dates.

Question 10: How might this timeline information be useful for forensic investigations?

- The timeline information be useful for forensic investigations because it might help to establishing timeline of events. Identifying Patterns and Anamalies, correlating events, identifying insider threats, support legal proceedings, and etc.

Click “Close”.
Back to “Host Manager”

Explore any other features of Autopsy & Sleuthkit you would like to.

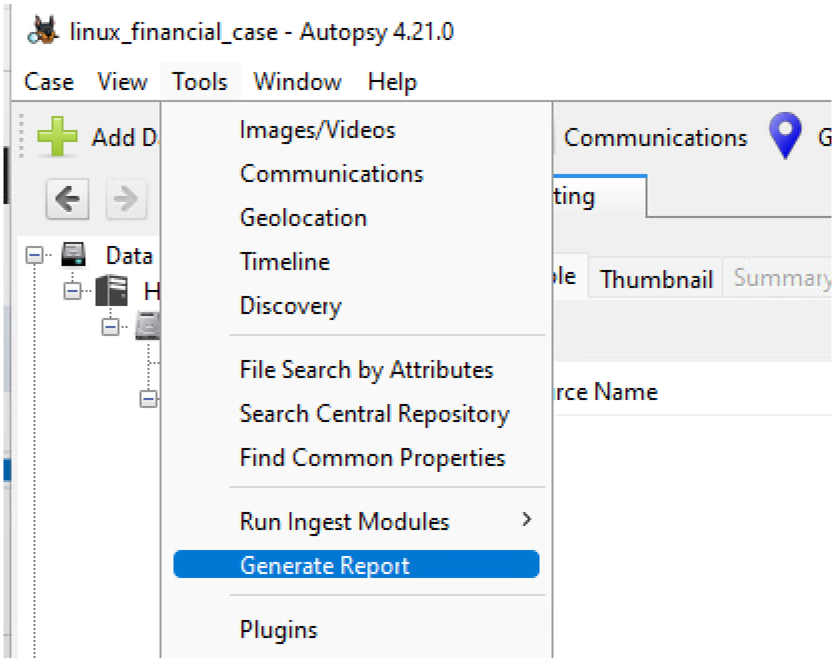
After you are done, close the case by clicking “Close Host” then “Close Case”. You can reopen the case to work on it later if you choose to.

Part 3. Use Windows Autopsy to analyze “*Linux_Financial_Case.001*” *case* (16 points)

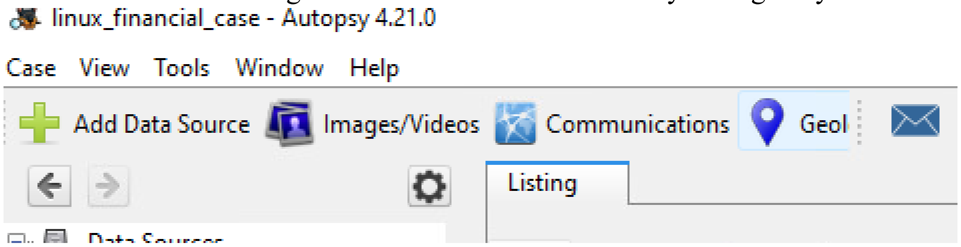
Instruction: Download the latest Autopsy for Windows from https://www.autopsy.com/download/, install it on your Windows Forensic machine, and analyze “Linux_Financial_Case.001”.

1) List at least three features of Windows Autopsy with screenshots. (10 points)

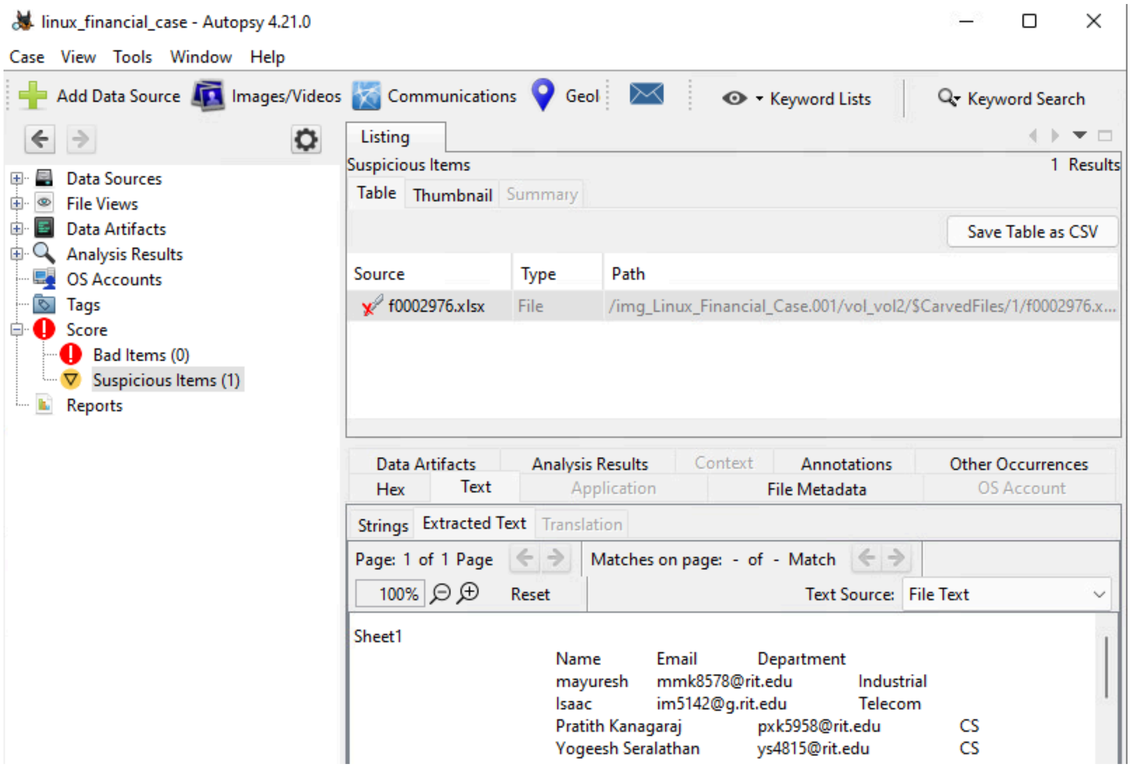
- You can generate report:



- You can see the geolocation if it’s able to find any through any of the file.



- It can show you potential suspicious items or files

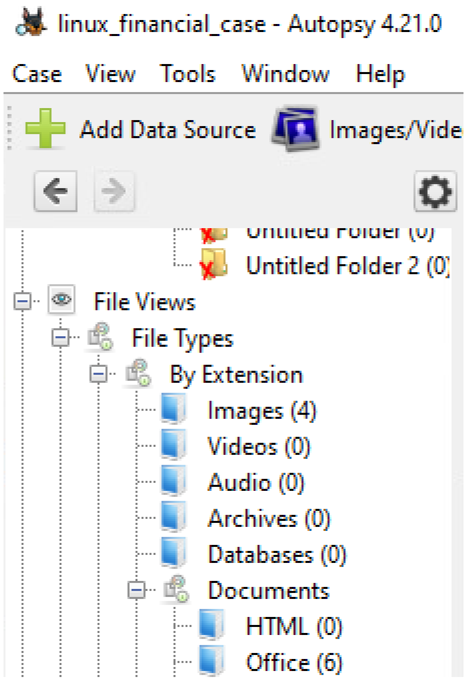


Source	Type	Path
f0002976.xlsx	File	/img_Linux_Financial_Case.001/vol_vol2/$CarvedFiles/1/f0002976.x...

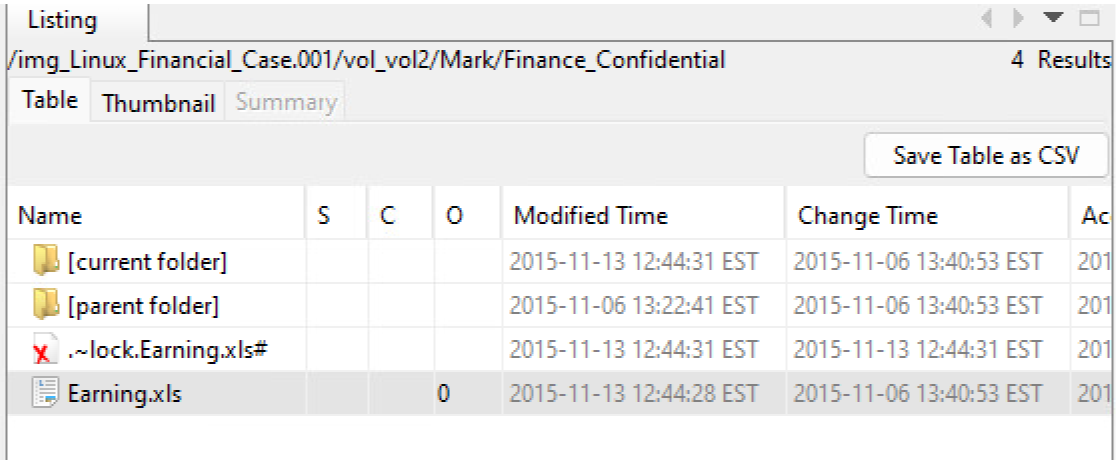
```
Sheet1
                    Name            Email              Department
                    mayuresh        mmk8578@rit.edu              Industrial
                    Isaac           im5142@g.rit.edu             Telecom
                    Pratith Kanagaraj         pxk5958@rit.edu              CS
                    Yogeesh Seralathan        ys4815@rit.edu               CS
```

2) Compare the Windows Autopsy with the Linux Autopsy and provide your comments with two or three sentences. (6 points)

a. Both Linux and Windows autopsy can sort the file type by extension:



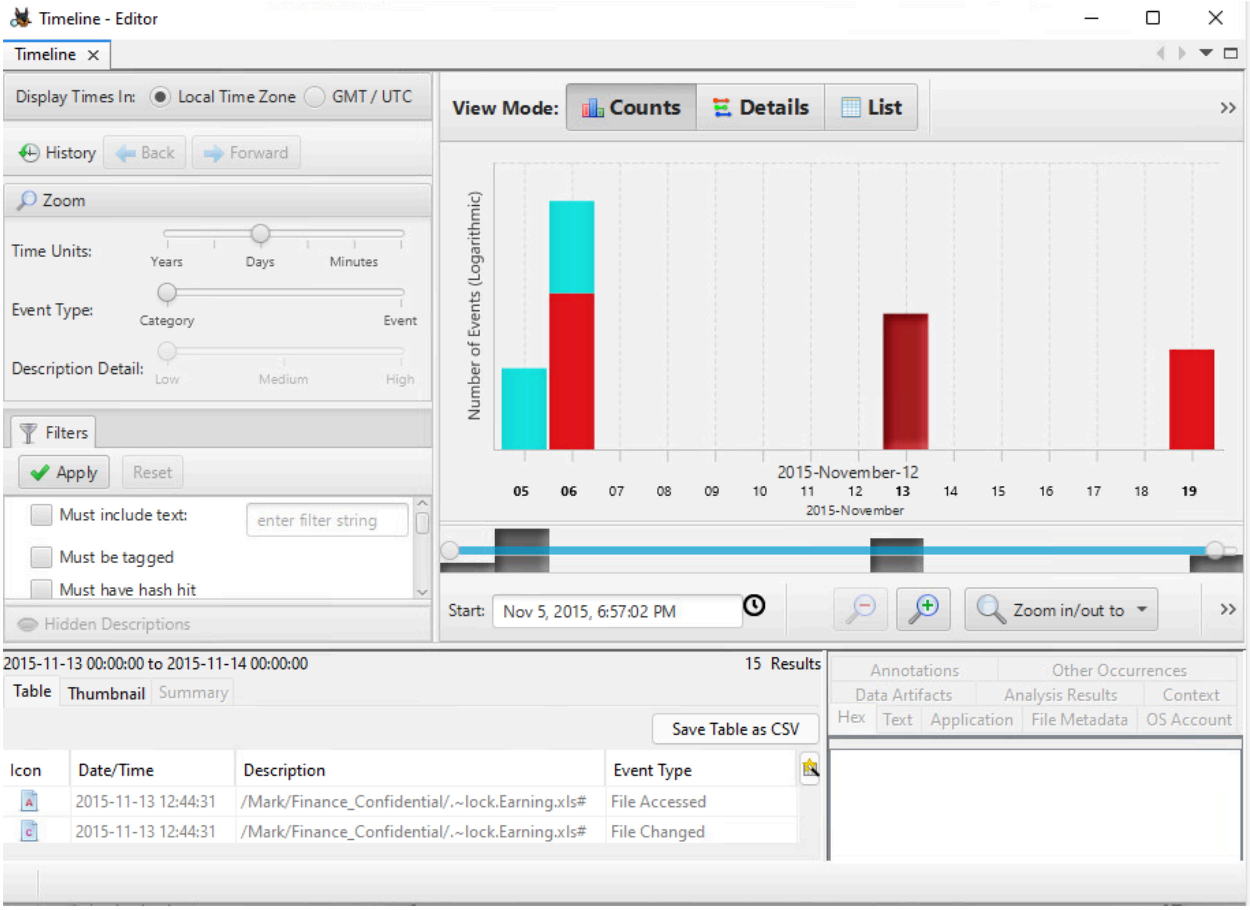
b. Both lets you download the a file in a case:



/img_Linux_Financial_Case.001/vol_vol2/Mark/Finance_Confidential 4 Results

Name	S	C	O	Modified Time	Change Time	Ac
[current folder]				2015-11-13 12:44:31 EST	2015-11-06 13:40:53 EST	201
[parent folder]				2015-11-06 13:22:41 EST	2015-11-06 13:40:53 EST	201
.~lock.Earning.xls#				2015-11-13 12:44:31 EST	2015-11-13 12:44:31 EST	201
Earning.xls			0	2015-11-13 12:44:28 EST	2015-11-06 13:40:53 EST	201

c. In both you can see timeline



Display Times In: ● Local Time Zone ○ GMT / UTC

Start: Nov 5, 2015, 6:57:02 PM

2015-11-13 00:00:00 to 2015-11-14 00:00:00 15 Results

Icon	Date/Time	Description	Event Type
A	2015-11-13 12:44:31	/Mark/Finance_Confidential/.~lock.Earning.xls#	File Accessed
C	2015-11-13 12:44:31	/Mark/Finance_Confidential/.~lock.Earning.xls#	File Changed

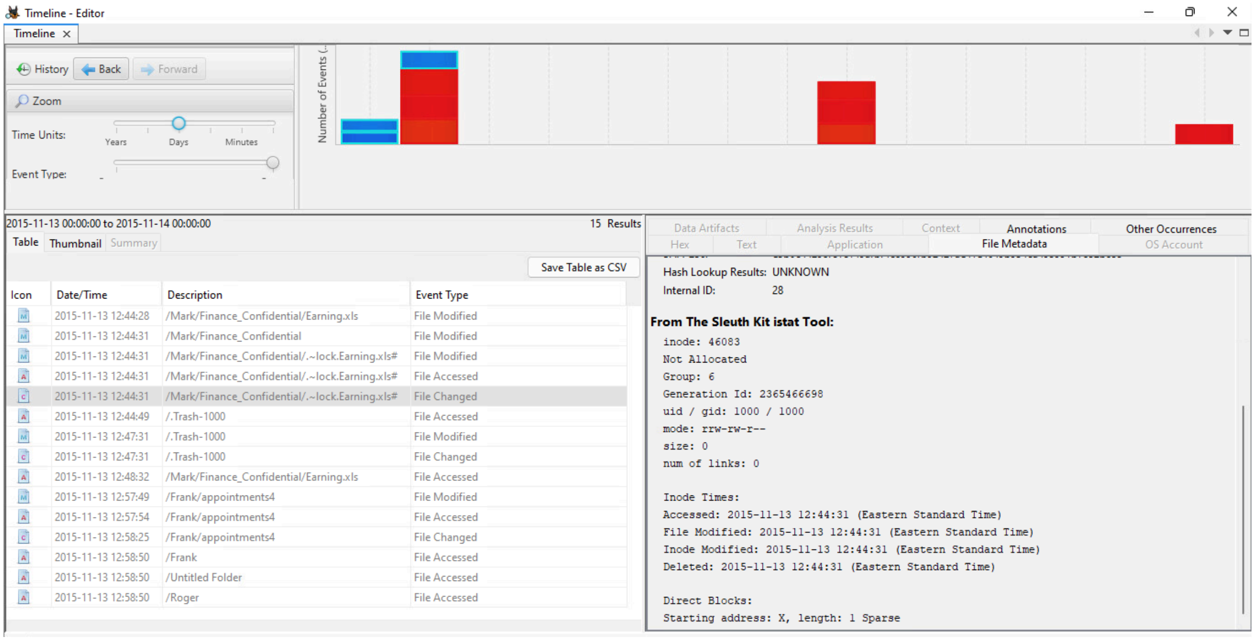
Part 4. Report (14 points)

Read the case scenario again and provide a short report that includes:

1. Your statement and evidence that indicates Frank may have read Earnings.xls. (8 points)
2. Why is Frank able to read a confidential document? (3 points)
3. How do you change the permissions, so that the “Earning.xls” file will not be accessible by others? (3 points)

- The timeline shows that the "Earnings.xls" file within the "/Mark/Finance_Confidential" directory was accessed and modified on 2015-11-13. The presence of a lock file ".~lock.Earnings.xls" also suggests that the file was opened by a user, which typically creates a lock file to prevent concurrent editing. It is important to highlight that there was notable activity from Frank on the same day, close to the time when the "Earnings.xls" file was accessed or modified. This suggests that Frank was active on the system during the relevant timeframe, further supporting the assertion that Frank may have opened or read the "Earnings.xls" file.

  The file permissions (rw-rw-r--) as indicated in the inode data suggest that the file was writable and readable by the owner and group. If Frank was a member of the group with read permissions, or if the directory permissions allowed group read access, he could have accessed the file. To change permissions and restrict access, you could use the chmod command to remove group and other user permissions, the command ***chmod o-rwx,g-rwx*** Earnings.xls This command removes read, write, and execute permissions for both groups and others, ensuring only the file owner can access it.



2015-11-13 00:00:00 to 2015-11-14 00:00:00 15 Results

Icon	Date/Time	Description	Event Type
M	2015-11-13 12:44:28	/Mark/Finance_Confidential/Earning.xls	File Modified
M	2015-11-13 12:44:31	/Mark/Finance_Confidential	File Modified
M	2015-11-13 12:44:31	/Mark/Finance_Confidential/.~lock.Earning.xls#	File Modified
A	2015-11-13 12:44:31	/Mark/Finance_Confidential/.~lock.Earning.xls#	File Accessed
C	2015-11-13 12:44:31	/Mark/Finance_Confidential/.~lock.Earning.xls#	File Changed
A	2015-11-13 12:44:49	/.Trash-1000	File Accessed
M	2015-11-13 12:47:31	/.Trash-1000	File Modified
C	2015-11-13 12:47:31	/.Trash-1000	File Changed
A	2015-11-13 12:48:32	/Mark/Finance_Confidential/Earning.xls	File Accessed
M	2015-11-13 12:57:49	/Frank/appointments4	File Modified
A	2015-11-13 12:57:54	/Frank/appointments4	File Accessed
C	2015-11-13 12:58:25	/Frank/appointments4	File Changed
A	2015-11-13 12:58:50	/Frank	File Accessed
A	2015-11-13 12:58:50	/Untitled Folder	File Accessed
A	2015-11-13 12:58:50	/Roger	File Accessed

Hash Lookup Results: UNKNOWN
Internal ID: 28

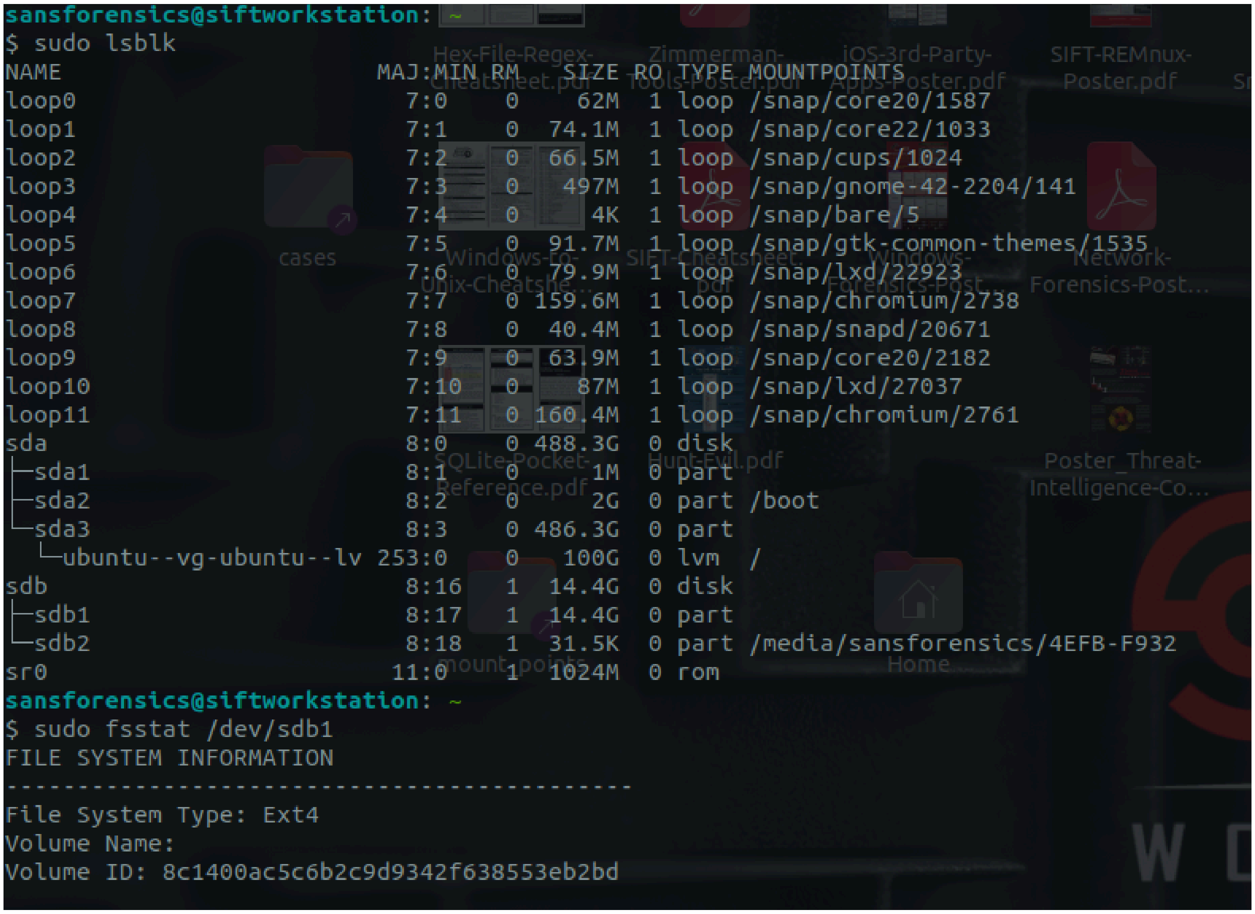
From The Sleuth Kit istat Tool:

```
inode: 46083
Not Allocated
Group: 6
Generation Id: 2365466698
uid / gid: 1000 / 1000
mode: rrw-rw-r--
size: 0
num of links: 0

Inode Times:
Accessed: 2015-11-13 12:44:31 (Eastern Standard Time)
File Modified: 2015-11-13 12:44:31 (Eastern Standard Time)
Inode Modified: 2015-11-13 12:44:31 (Eastern Standard Time)
Deleted: 2015-11-13 12:44:31 (Eastern Standard Time)

Direct Blocks:
Starting address: X, length: 1 Sparse
```

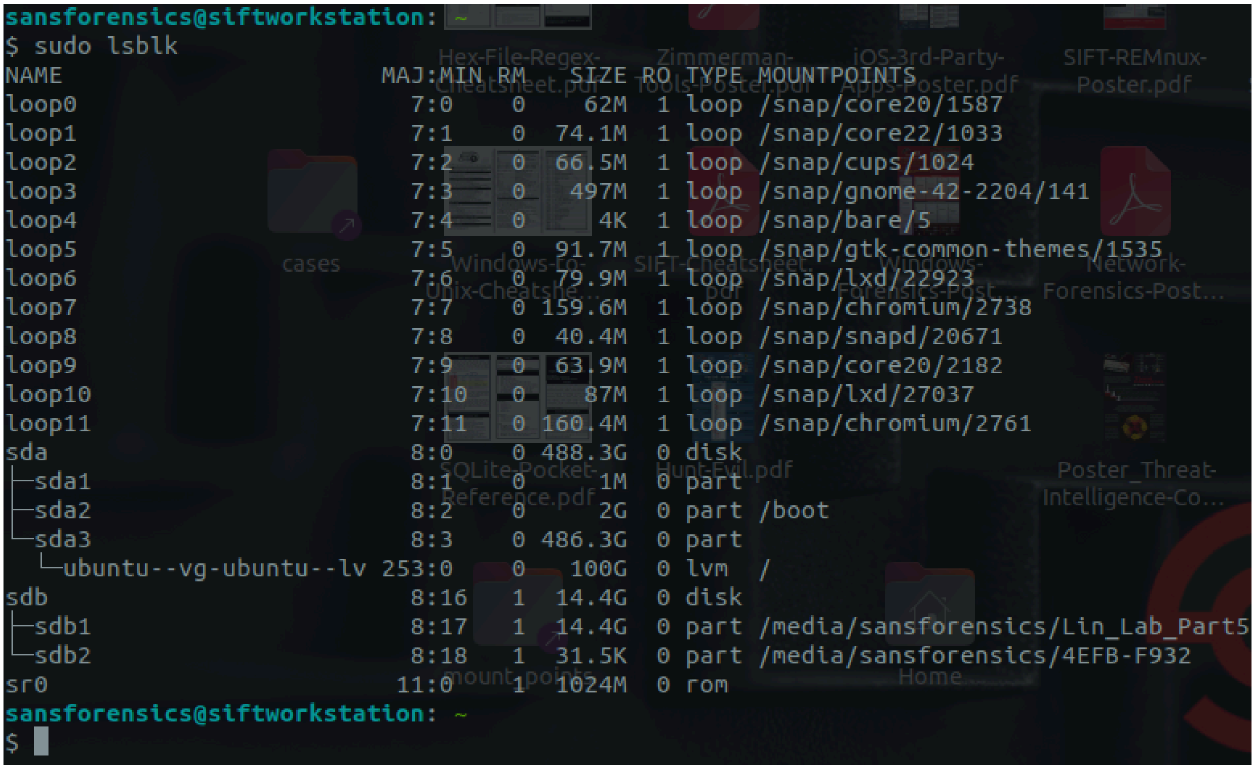
Part 5. Bonus (20 points)

Analyze deleted files in the ext4 filesystem.

1. Create a small ext4 partition on SIFT VM, create some files, a directory, and a couple of files in the directory, and delete some files.

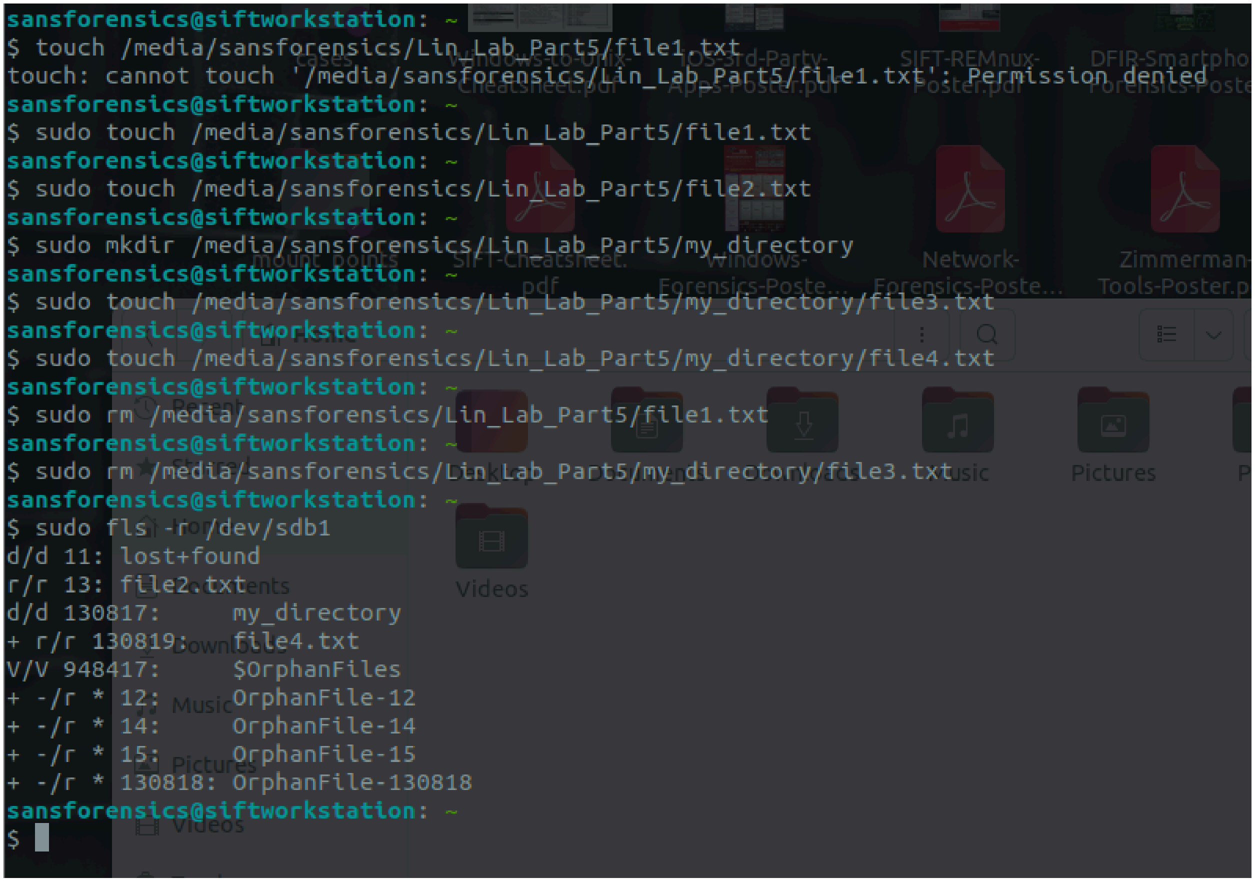


```
sansforensics@siftworkstation: ~
$ sudo lsblk
NAME                       MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0                        7:0    0    62M  1 loop /snap/core20/1587
loop1                        7:1    0  74.1M  1 loop /snap/core22/1033
loop2                        7:2    0  66.5M  1 loop /snap/cups/1024
loop3                        7:3    0   497M  1 loop /snap/gnome-42-2204/141
loop4                        7:4    0    4K   1 loop /snap/bare/5
loop5                        7:5    0  91.7M  1 loop /snap/gtk-common-themes/1535
loop6                        7:6    0  79.9M  1 loop /snap/lxd/22923
loop7                        7:7    0 159.6M  1 loop /snap/chromium/2738
loop8                        7:8    0  40.4M  1 loop /snap/snapd/20671
loop9                        7:9    0  63.9M  1 loop /snap/core20/2182
loop10                       7:10   0    87M  1 loop /snap/lxd/27037
loop11                       7:11   0 160.4M  1 loop /snap/chromium/2761
sda                          8:0    0 488.3G  0 disk
├─sda1                       8:1    0     1M  0 part
├─sda2                       8:2    0     2G  0 part /boot
└─sda3                       8:3    0 486.3G  0 part
  └─ubuntu--vg-ubuntu--lv  253:0    0   100G  0 lvm  /
sdb                          8:16   1  14.4G  0 disk
├─sdb1                       8:17   1  14.4G  0 part
└─sdb2                       8:18   1  31.5K  0 part /media/sansforensics/4EFB-F932
sr0                         11:0    1  1024M  0 rom
sansforensics@siftworkstation: ~
$ sudo fsstat /dev/sdb1
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: Ext4
Volume Name:
Volume ID: 8c1400ac5c6b2c9d9342f638553eb2bd
```



```
sansforensics@siftworkstation: ~
$ sudo lsblk
NAME                       MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0                        7:0    0    62M  1 loop /snap/core20/1587
loop1                        7:1    0  74.1M  1 loop /snap/core22/1033
loop2                        7:2    0  66.5M  1 loop /snap/cups/1024
loop3                        7:3    0   497M  1 loop /snap/gnome-42-2204/141
loop4                        7:4    0    4K   1 loop /snap/bare/5
loop5                        7:5    0  91.7M  1 loop /snap/gtk-common-themes/1535
loop6                        7:6    0  79.9M  1 loop /snap/lxd/22923
loop7                        7:7    0 159.6M  1 loop /snap/chromium/2738
loop8                        7:8    0  40.4M  1 loop /snap/snapd/20671
loop9                        7:9    0  63.9M  1 loop /snap/core20/2182
loop10                       7:10   0    87M  1 loop /snap/lxd/27037
loop11                       7:11   0 160.4M  1 loop /snap/chromium/2761
sda                          8:0    0 488.3G  0 disk
├─sda1                       8:1    0     1M  0 part
├─sda2                       8:2    0     2G  0 part /boot
└─sda3                       8:3    0 486.3G  0 part
  └─ubuntu--vg-ubuntu--lv  253:0    0   100G  0 lvm  /
sdb                          8:16   1  14.4G  0 disk
├─sdb1                       8:17   1  14.4G  0 part /media/sansforensics/Lin_Lab_Part5
└─sdb2                       8:18   1  31.5K  0 part /media/sansforensics/4EFB-F932
sr0                         11:0    1  1024M  0 rom
sansforensics@siftworkstation: ~
$ 
```

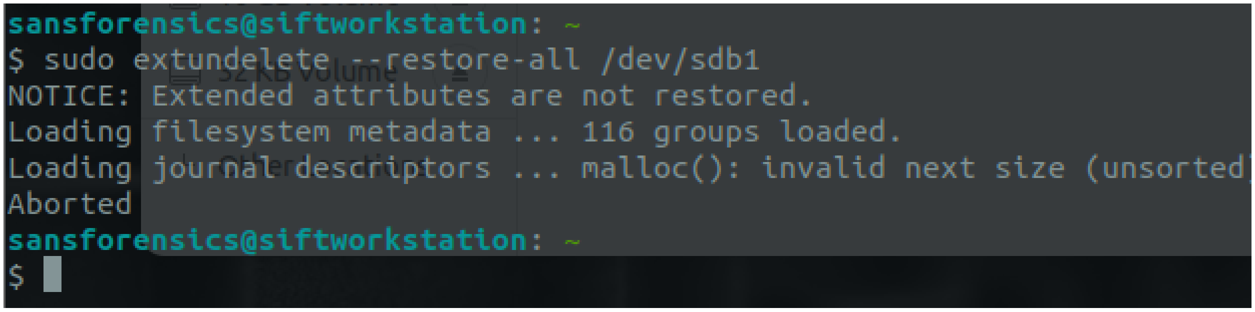
2. Use Sluethkit commands *fls* to list the inode of the deleted file. Show the inode content. Are you able to recover the content of the deleted files? Explain



```
sansforensics@siftworkstation: ~
$ touch /media/sansforensics/Lin_Lab_Part5/file1.txt
touch: cannot touch '/media/sansforensics/Lin_Lab_Part5/file1.txt': Permission denied
sansforensics@siftworkstation: ~
$ sudo touch /media/sansforensics/Lin_Lab_Part5/file1.txt
sansforensics@siftworkstation: ~
$ sudo touch /media/sansforensics/Lin_Lab_Part5/file2.txt
sansforensics@siftworkstation: ~
$ sudo mkdir /media/sansforensics/Lin_Lab_Part5/my_directory
sansforensics@siftworkstation: ~
$ sudo touch /media/sansforensics/Lin_Lab_Part5/my_directory/file3.txt
sansforensics@siftworkstation: ~
$ sudo touch /media/sansforensics/Lin_Lab_Part5/my_directory/file4.txt
sansforensics@siftworkstation: ~
$ sudo rm /media/sansforensics/Lin_Lab_Part5/file1.txt
sansforensics@siftworkstation: ~
$ sudo rm /media/sansforensics/Lin_Lab_Part5/my_directory/file3.txt
sansforensics@siftworkstation: ~
$ sudo fls -r /dev/sdb1
d/d 11: lost+found
r/r 13: file2.txt
d/d 130817:     my_directory
+ r/r 130819:   file4.txt
V/V 948417:     $OrphanFiles
+ -/r * 12:     OrphanFile-12
+ -/r * 14:     OrphanFile-14
+ -/r * 15:     OrphanFile-15
+ -/r * 130818: OrphanFile-130818
sansforensics@siftworkstation: ~
$
```

3. Use *extundelete* to try to recover the deleted content (Note: undelete using *extundelete* is not guaranteed). Show your results and explain how *extundelete* attempts to undelete the content.

- could not undelete using the tool.



```
sansforensics@siftworkstation: ~
$ sudo extundelete --restore-all /dev/sdb1
NOTICE: Extended attributes are not restored.
Loading filesystem metadata ... 116 groups loaded.
Loading journal descriptors ... malloc(): invalid next size (unsorted
Aborted
sansforensics@siftworkstation: ~
$
```

- In an attempt to recover a file's data, extundelete employs a methodical procedure. It starts by looking through the filesystem for inodes that are listed in the inode table but are marked as deleted. The original path of the deleted file is then found by reconstructing the directory structure. After that, extundelete finds and reads the data blocks linked to the inode of the deleted file. The real file data is contained in these data blocks. Lastly, extundelete attempts to maintain the original filename and permissions while writing the recovered file data to a designated output directory. Extundelete may take into account variables like filesystem journaling throughout this procedure to guarantee the consistency and integrity of the recovered file.