

HW- 02 Lab 1

By :- Faraz Ahmed

Using guymager to import and verify an existing disk image

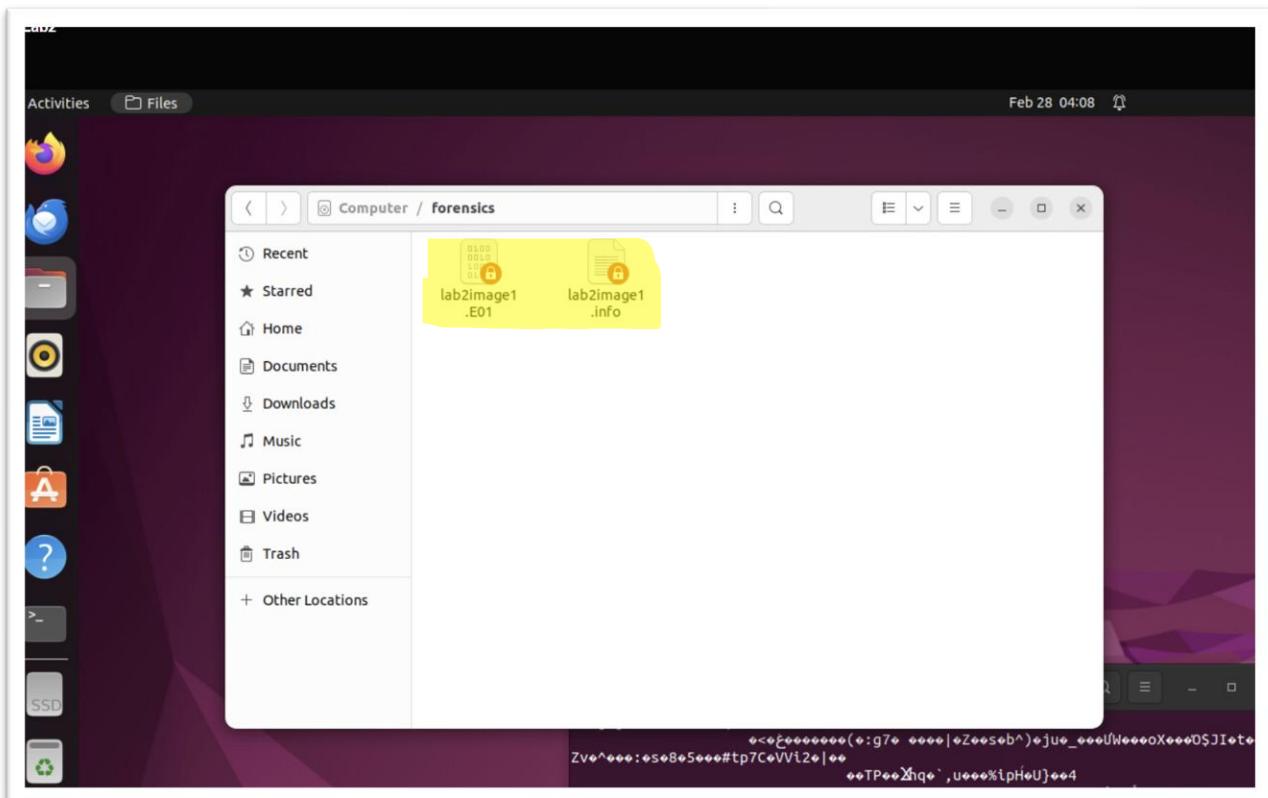
Q7 a. Provide a brief overview of the dd (or raw) file format vs the Expert Witness Format and its significance to a forensic investigator.

Ans7 a.

- **Dd (raw) File Format**- it comes from a Unix command “dd” which creates a bit-for-bit copy of a storage drive or device. So, the copied data doesn’t have any metadata, uncompressed, large file size and simple and widely supported for any forensic tools.
- **Expert Witness Format**- it is developed by EnCase which creates simple copy of storage drive or device. So, the copied data includes compression, metadata and widely supported for organized and crucial forensic investigation.

Q12. What file(s) have been created?

Ans12. As we can observe from the screenshot 1 (highlighted below), there’s two new files “lab2image1” and “lab2image1.info” which are newly created.



Screenshot 1: Screenshot of two new files “lab2image1 and lab2image1” added.

Q13. Open the file ending in “.info”.

- a. What is the MD5 hash?

Ans a. e134ced312b3511d88943d57ccd70c83 is the hash value.

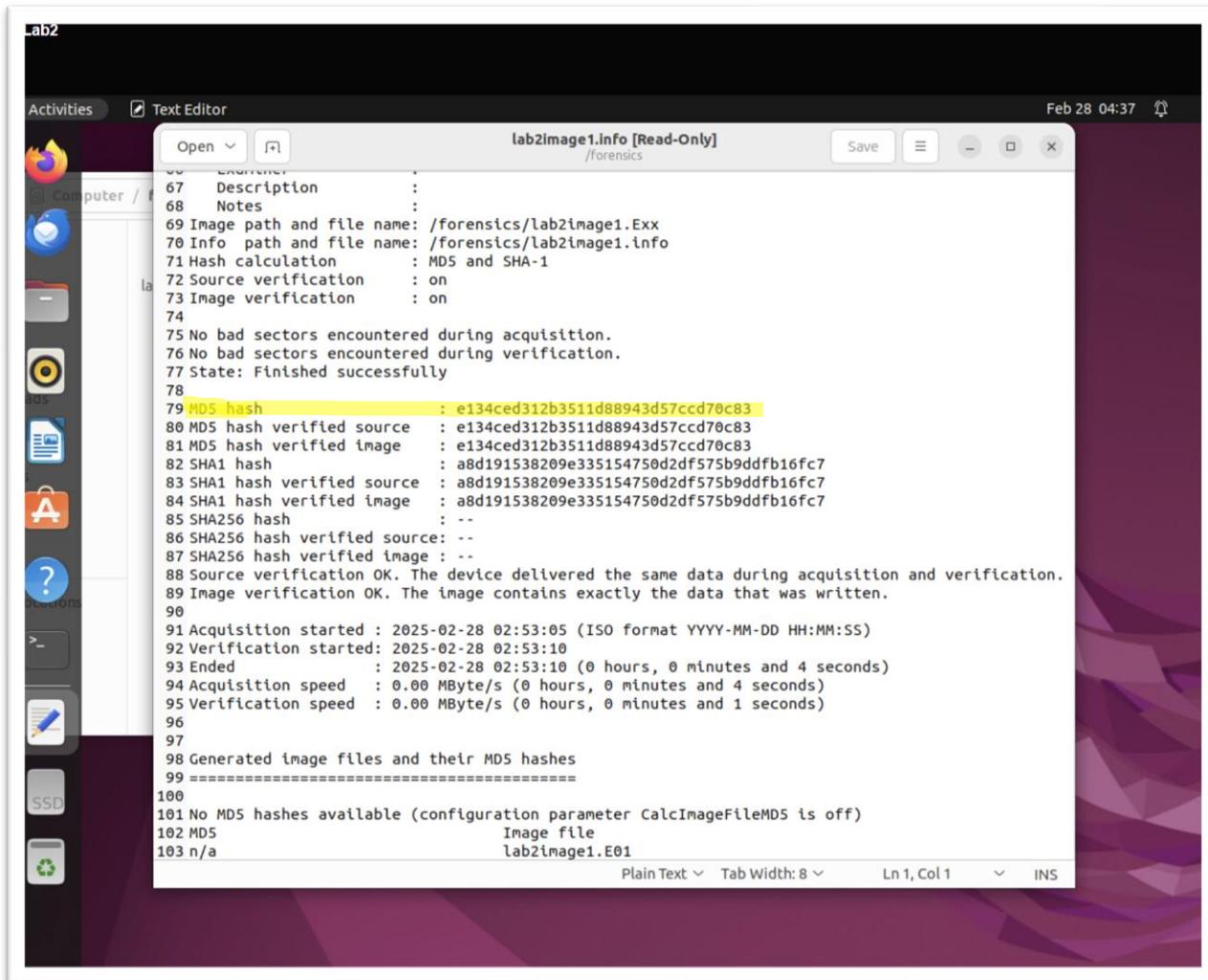
b. What is the MD5 hash verified source?

Ans b. e134ced312b3511d88943d57ccd70c83 is the MD5 hash verified source.

c. Do these two values match? Why is that significant?

Ans c. Yes, these two values match with each other as shown(highlighted) below in screenshot 2. It is significant as:-

- It makes sure that there is data integrity between both copy and original disk.
- It ensures that there was no modification or data lose during coping the data.



```
Lab2
Activities Text Editor Feb 28 04:37
File Edit View Insert Cell Help
Open + lab2Image1.info [Read-Only]
Save X
67 Description :
68 Notes :
69 Image path and file name: /forensics/lab2image1.E01
70 Info path and file name: /forensics/lab2image1.info
71 Hash calculation : MD5 and SHA-1
72 Source verification : on
73 Image verification : on
74
75 No bad sectors encountered during acquisition.
76 No bad sectors encountered during verification.
77 State: Finished successfully
78
79 MD5 hash : e134ced312b3511d88943d57ccd70c83
80 MDS hash verified source : e134ced312b3511d88943d57ccd70c83
81 MDS hash verified image : e134ced312b3511d88943d57ccd70c83
82 SHA1 hash : a8d191538209e335154750d2df575b9ddfb16fc7
83 SHA1 hash verified source : a8d191538209e335154750d2df575b9ddfb16fc7
84 SHA1 hash verified image : a8d191538209e335154750d2df575b9ddfb16fc7
85 SHA256 hash : --
86 SHA256 hash verified source: --
87 SHA256 hash verified image : --
88 Source verification OK. The device delivered the same data during acquisition and verification.
89 Image verification OK. The image contains exactly the data that was written.
90
91 Acquisition started : 2025-02-28 02:53:05 (ISO format YYYY-MM-DD HH:MM:SS)
92 Verification started: 2025-02-28 02:53:10
93 Ended : 2025-02-28 02:53:10 (0 hours, 0 minutes and 4 seconds)
94 Acquisition speed : 0.00 MByte/s (0 hours, 0 minutes and 4 seconds)
95 Verification speed : 0.00 MByte/s (0 hours, 0 minutes and 1 seconds)
96
97
98 Generated image files and their MD5 hashes
99 =====
100
101 No MD5 hashes available (configuration parameter CalcImageFileMD5 is off)
102 MDS Image file
103 n/a lab2image1.E01
```

Screenshot 2: Screenshot of MD5 hash value of “lab2image1.info”.

d. What is the size of each file created?

And d. Space of each file is:-

- Lab2image1.E01- 3.7 kB (3,672 bytes)
- Lab2image1.info- 6.0 kB (5,973 bytes)

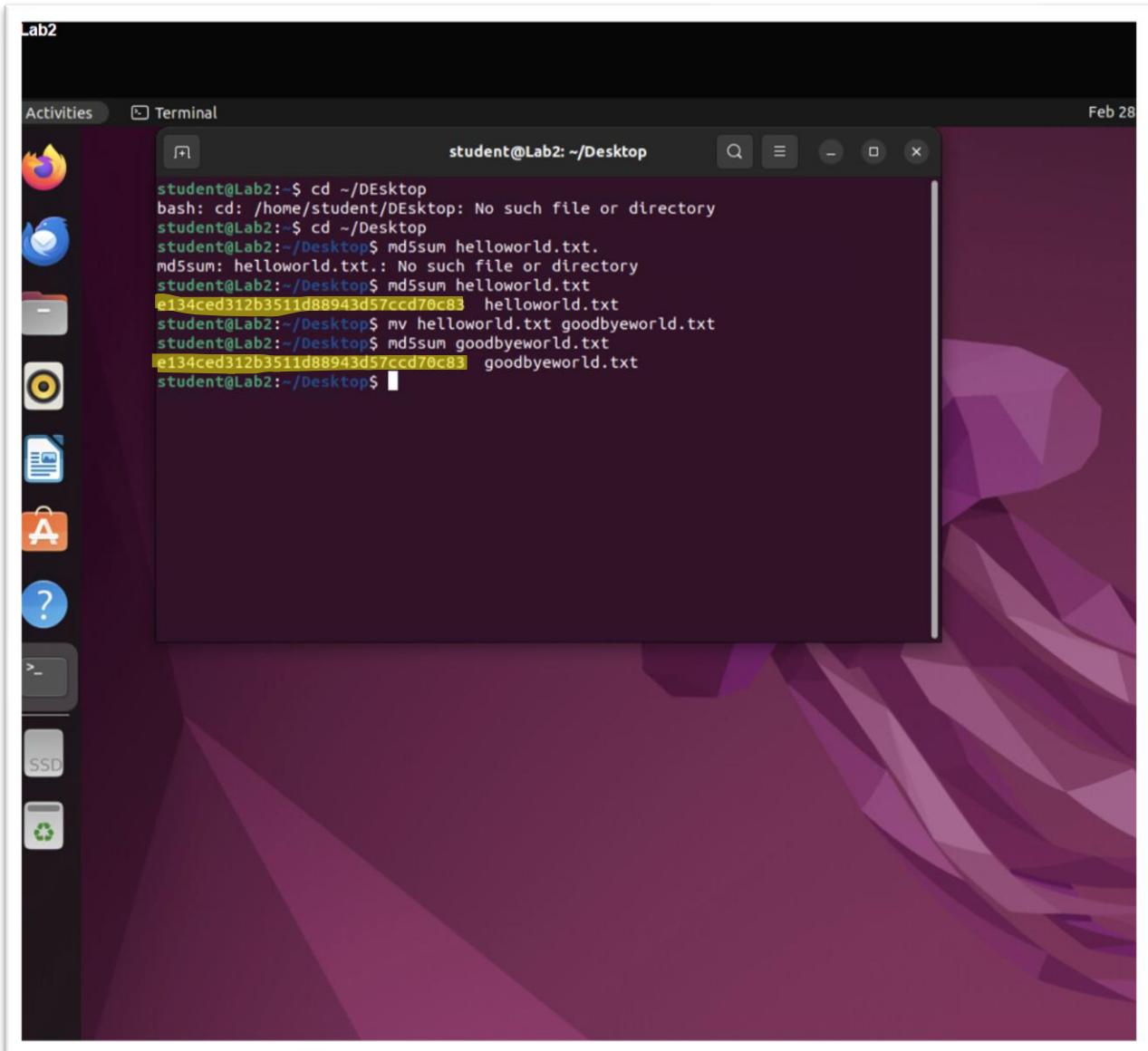
Experimenting With Hashing

Q4. What did this command do?

Ans4. The mv command (move) is a command to rename any file which in this case from helloworld.txt to goodbyeworld.txt but it does not modify or change the contents inside it. So, its same file with just different name.

Q6. Did the hash change? Why (did/did not) the hash change?

Ans6. No, the hash value did not change. It is because MD5 hash value is evaluated based on the contents inside the file, not its name. So, modifying name does not involve changing data include that file hence the hash value remains the same as shown in Screenshot 3.

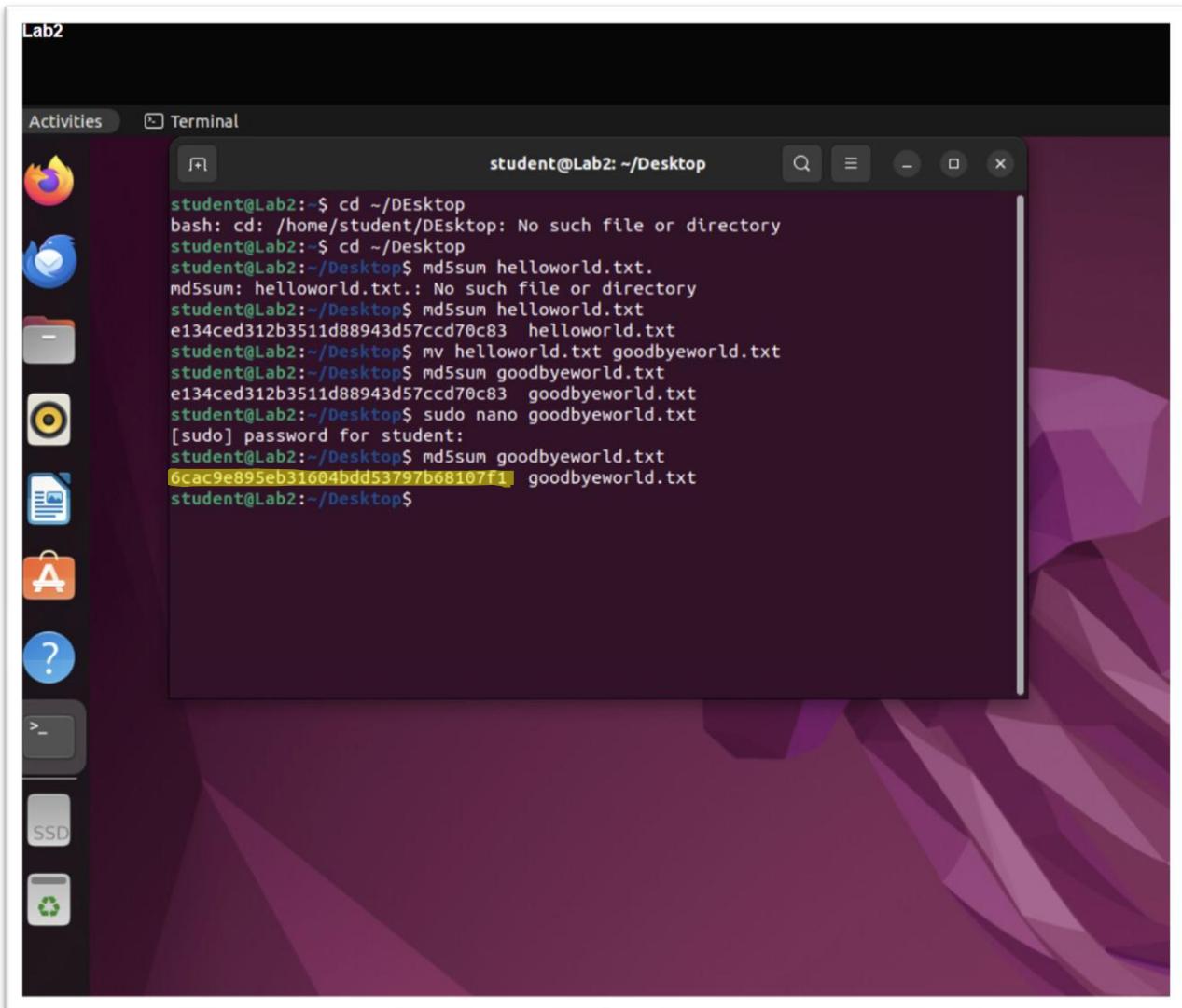
A screenshot of a Linux desktop environment, likely Ubuntu, showing a terminal window titled "student@Lab2: ~/Desktop". The terminal displays a series of commands and their outputs. The user first tries to change directory to "/DEsktop" and then to "/Desktop", both of which fail with "No such file or directory" errors. Next, they run "md5sum helloworld.txt", which outputs a long hex string. They then use the "mv" command to rename "helloworld.txt" to "goodbyeworld.txt". Finally, they run "md5sum goodbyeworld.txt" again, and the output is identical to the previous one, demonstrating that the file's content remained unchanged despite the name change.

```
student@Lab2: ~$ cd ~/DEsktop
bash: cd: /home/student/DEsktop: No such file or directory
student@Lab2: ~$ cd ~/Desktop
student@Lab2: ~/Desktop$ md5sum helloworld.txt
md5sum: helloworld.txt.: No such file or directory
student@Lab2: ~/Desktop$ md5sum helloworld.txt
e134ced312b3511d88943d57cc0c83  helloworld.txt
student@Lab2: ~/Desktop$ mv helloworld.txt goodbyeworld.txt
student@Lab2: ~/Desktop$ md5sum goodbyeworld.txt
e134ced312b3511d88943d57cc0c83  goodbyeworld.txt
student@Lab2: ~/Desktop$
```

Screenshot 3: Screenshot of hash values after renaming the file.

Q9. Did the hash change? Why (did/did not) the hash change?

Ans9. Yes, the hash value changed. This is because MD5 hash values fully depends on inside contents of any file. So, since we are modifying and adding “Goodbye!” inside that file, the hash value will be changed as shown and highlighted in Screenshot 4.

A screenshot of a Linux desktop environment titled "Lab2". On the left is a vertical dock with icons for various applications like a browser, file manager, terminal, and system tools. A terminal window is open in the center, showing a command-line session. The session starts with the user navigating to their desktop directory and attempting to run a non-existent "DEsktop" file. Then, they run "md5sum" on a file named "helloworld.txt", which also does not exist. After creating the file with the command "mv helloworld.txt goodbyeworld.txt", they run "md5sum" again. The output shows the original hash "e134ced312b3511d88943d57ccd70c83" followed by the new hash "6cac9e895eb31604bdd53797b68107f1", which is highlighted in yellow. The terminal window has a dark theme with white text and a black background.

```
student@Lab2:~/DEsktop
bash: cd: /home/student/DEsktop: No such file or directory
student@Lab2:~/DEsktop
student@Lab2:~/DEsktop$ md5sum helloworld.txt.
md5sum: helloworld.txt.: No such file or directory
student@Lab2:~/DEsktop$ md5sum helloworld.txt
e134ced312b3511d88943d57ccd70c83  helloworld.txt
student@Lab2:~/DEsktop$ mv helloworld.txt goodbyeworld.txt
student@Lab2:~/DEsktop$ md5sum goodbyeworld.txt
e134ced312b3511d88943d57ccd70c83  goodbyeworld.txt
student@Lab2:~/DEsktop$ sudo nano goodbyeworld.txt
[sudo] password for student:
student@Lab2:~/DEsktop$ md5sum goodbyeworld.txt
6cac9e895eb31604bdd53797b68107f1  goodbyeworld.txt
student@Lab2:~/DEsktop$
```

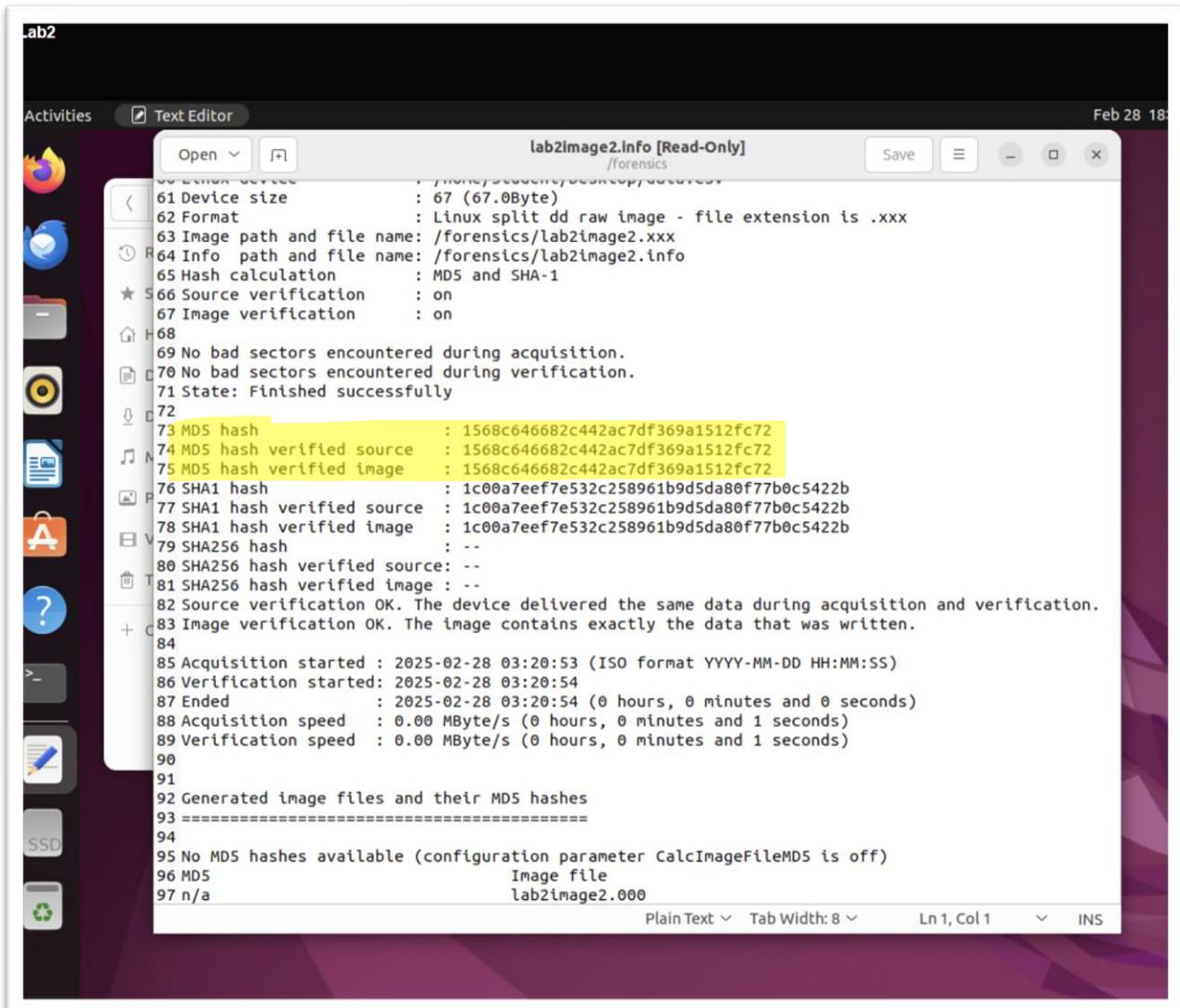
Screenshot 4: Screenshot of hash values after changing contents in that file.

Creating a New Image

Q10. Open the resulting info file and document the MD5 hash values.

Ans10. All of the different MD5 hash values are:-

- MD5 Hash- 1568c646682c442ac7df369a1512fc72
- MD5 Hash Verified Source- 1568c646682c442ac7df369a1512fc72
- MD5 Hash Verified Image- 1568c646682c442ac7df369a1512fc72 (as highlighted below)



```
60 Linux version : /forensics/2025/02/28/03:20:53
61 Device size      : 67 (67.08byte)
62 Format          : Linux split dd raw image - file extension is .xxx
63 Image path and file name: /forensics/lab2image2.xxx
64 Info  path and file name: /forensics/lab2image2.info
65 Hash calculation   : MD5 and SHA-1
66 Source verification  : on
67 Image verification   : on
68
69 No bad sectors encountered during acquisition.
70 No bad sectors encountered during verification.
71 State: Finished successfully
72
73 MD5 hash        : 1568c646682c442ac7df369a1512fc72
74 MD5 hash verified source : 1568c646682c442ac7df369a1512fc72
75 MD5 hash verified image  : 1568c646682c442ac7df369a1512fc72
76 SHA1 hash       : 1c00a7eef7e532c258961b9d5da80f77b0c5422b
77 SHA1 hash verified source : 1c00a7eef7e532c258961b9d5da80f77b0c5422b
78 SHA1 hash verified image  : 1c00a7eef7e532c258961b9d5da80f77b0c5422b
79 SHA256 hash     : --
80 SHA256 hash verified source: --
81 SHA256 hash verified image : --
82 Source verification OK. The device delivered the same data during acquisition and verification.
83 Image verification OK. The image contains exactly the data that was written.
84
85 Acquisition started : 2025-02-28 03:20:53 (ISO format YYYY-MM-DD HH:MM:SS)
86 Verification started: 2025-02-28 03:20:54
87 Ended              : 2025-02-28 03:20:54 (0 hours, 0 minutes and 0 seconds)
88 Acquisition speed   : 0.00 MByte/s (0 hours, 0 minutes and 1 seconds)
89 Verification speed  : 0.00 MByte/s (0 hours, 0 minutes and 1 seconds)
90
91
92 Generated image files and their MD5 hashes
93 =====
94
95 No MD5 hashes available (configuration parameter CalcImageFileMD5 is off)
96 MD5                  Image file
97 n/a                 lab2image2.000
```

Screenshot 5: Screenshot of all MD5 values in "lab2image2.info".

Q11. Document the method and what you suspect is the type of acquisition conducted for these suspect files:

Ans11. The method we used in this was that Guymager basically select and produce a copy of data.csv file rather than take the entire physical drive into consideration. So, this method is known as Logical Acquisition.

So, this is a file-based (logical) forensic acquisition which means that it will copy or image only specific files rather than taking an entire storage device and the file is copied and hashed to verify the integrity and contents properly.

Q12. What is one major drawback of this acquisition method?

Ans12. One major drawback of this acquisition method is it only copies the file and other forensic evidence such as deleted files, hidden space and all system logs so if any criminal or suspect modifies metadata or timestamps of any files, those changes might be missed by it which can influence the investigation.

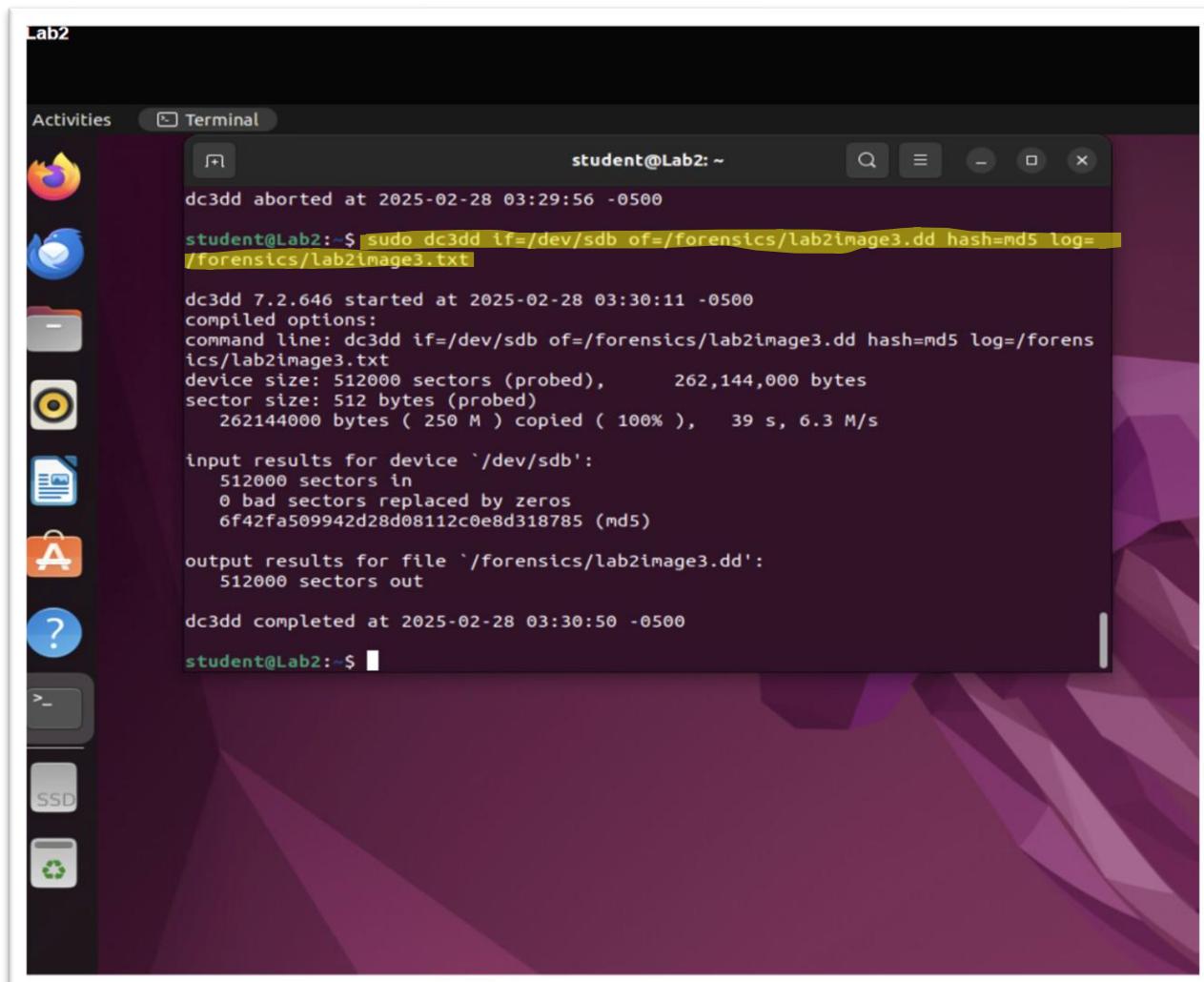
Creating a New Image Using The CLI

Q1 d. Why would you not want to mount the drive?

Ans1 d. Mounting the drive is not good for overall investigation or forensic integrity. Mounting the drive may overwrite the metadata or system logs which will give any suspect to alter with the evidence and destroy forensic integrity.

Q2 b. Do a little bit of research into this command. In 2-5 sentences, describe what the command is doing.

Ans2 b. sudo dc3dd if=/dev/sdb of=/forensics/lab2image3.dd hash=md5 log=/forensics/lab2image3.txt
Where: **dc3dd** is a special forensic version of dd which is optimized for securing all digital evidence. It picks up a specific file from location **if=/dev/sdb** and save output in **of=/forensics/lab2image3.dd** location. Then it calculate an MD5 hash value of that file for data integrity and **log=/forensics/lab2image3.txt** which records all of the forensic details including hash value and other imaging process. (as shown in screenshot 6)



```
student@Lab2: ~
student@Lab2: $ sudo dc3dd if=/dev/sdb of=/forensics/lab2image3.dd hash=md5 log=/forensics/lab2image3.txt

dc3dd 7.2.646 started at 2025-02-28 03:30:11 -0500
compiled options:
command line: dc3dd if=/dev/sdb of=/forensics/lab2image3.dd hash=md5 log=/forensics/lab2image3.txt
device size: 512000 sectors (probed),      262,144,000 bytes
sector size: 512 bytes (probed)
      262144000 bytes ( 250 M ) copied ( 100% ),   39 s, 6.3 M/s

input results for device '/dev/sdb':
  512000 sectors in
  0 bad sectors replaced by zeros
  6f42fa509942d28d08112c0e8d318785 (md5)

output results for file '/forensics/lab2image3.dd':
  512000 sectors out

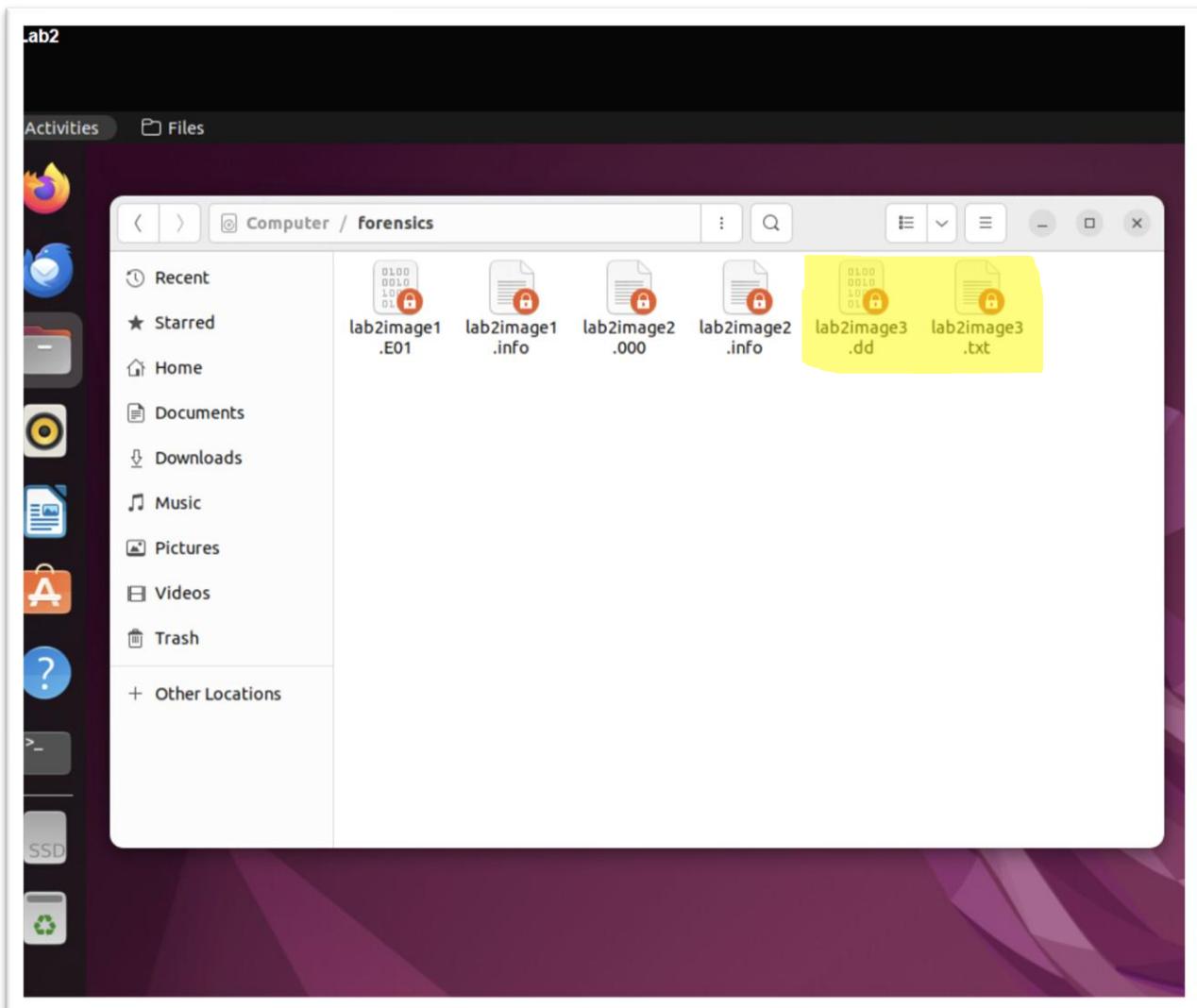
dc3dd completed at 2025-02-28 03:30:50 -0500
student@Lab2: $
```

Screenshot 6: Screenshot of executing “sudo dc3dd if=/dev/sdb of=/forensics/lab2image3.dd hash=md5 log=/forensics/lab2image3.txt”.

Q3. Open the Forensics folder again.

- What file(s) have been created?

Ans a. There are two files created which are “lab2image3.dd” and lab2image3.txt” as shown in screenshot 7.



Screenshot 7: Screenshot of two new files “lab2image3.dd and lab2image3.txt” created.

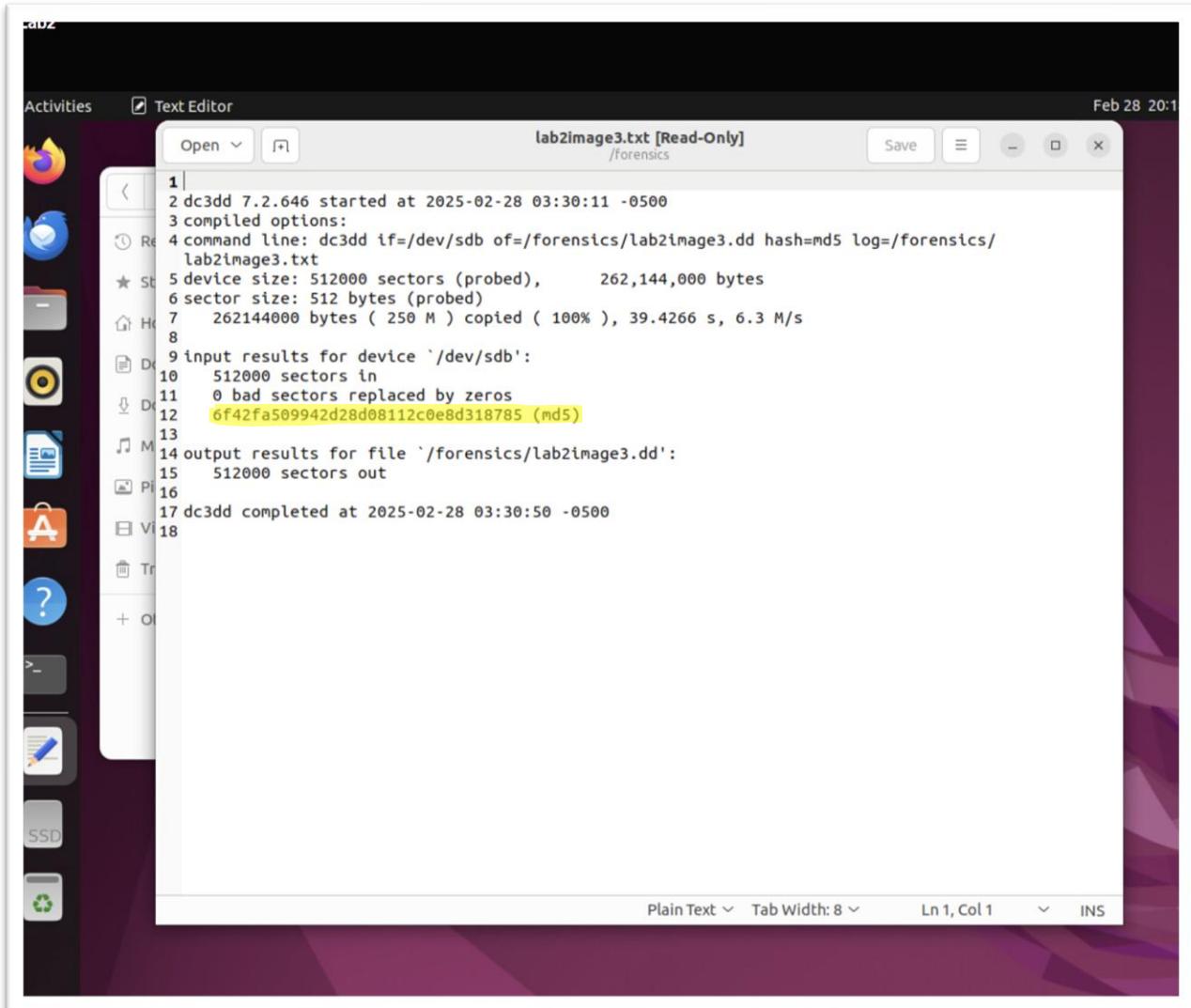
- What two methods can you use to verify the hash of the newly created forensic image?

Ans b. Two different methods to verify hash values are:-

- **Md5sum**- Enter md5sum /forensics/lab2image3.dd and then verify this output with original hash of lab2image3.txt.
- **Dc3dd**- Enter sudo dc3dd if=/forensics/lab2image3.dd hash=md5 and this recalculates the hash values.

- What is the MD5 hash of the new lab2image3.dd file?

Ans c. MD5 hash value is- 6f42fa509942d28d08112c0e8d318785 (highlighted in screenshot 8).



```
1| 2 dc3dd 7.2.646 started at 2025-02-28 03:30:11 -0500
3 compiled options:
4 command line: dc3dd if=/dev/sdb of=/forensics/lab2image3.dd hash=md5 log=/forensics/
lab2image3.txt
5 device size: 512000 sectors (probed),      262,144,000 bytes
6 sector size: 512 bytes (probed)
7    262144000 bytes ( 250 M ) copied ( 100% ), 39.4266 s, 6.3 M/s
8
9 input results for device '/dev/sdb':
10   512000 sectors in
11     0 bad sectors replaced by zeros
12     6f42fa509942d28d08112c0e8d318785 (md5)
13
14 output results for file '/forensics/lab2image3.dd':
15   512000 sectors out
16
17 dc3dd completed at 2025-02-28 03:30:50 -0500
18
```

Screenshot 8: Screenshot of hash value of “lab2image3.txt”.

Additional Questions

Q1. Aside from MD5 and SHA1, what other hashing algorithms can be used to verify digital evidence? List two.

Ans1. Aside from MD5 and SHA1, the other hashing algorithms are:-

- SHA-256- A robust cryptographic hash function with better collision resistance.
- SHA-512- A more and better secure and high integrity robust algorithm with 512-bit hash.

Q2. Would it be useful for a digital forensic investigator to use multiple hashing algorithms to verify digital evidence?

Ans2. Yes, multiple hashing algorithm could be useful as:-

- **Reliability**- If one of the algorithms failed to give correct results, we can depend on other algorithms for accurate outputs.
- **Cross-Validation**- Using multiple hashing algorithm helps us to check if the original data has been altered or not during forensic investigation.

Q3. Research and identify two other forensic drive imaging programs aside from guymager and dc3dd.

Ans3. Two other forensic drive imaging programs are:-

- **FTK Imager**- it is one of the most crucial and widely used forensic tool that helps in creating copy data images, hash values and program logs.
- **Ddrescue**- it is a toll designed specially to recover a lost data from damaged or stolen disks or drivers while minimizing any further data loss.

Q4. One tool we did not get to use during this lab is a write blocker. Describe what a write blocker is and its significance to digital forensic practitioners.

Ans4. A write blocker is a tool used on hardware and software and prevents modification of data inside them during anu crucial forensic investigation. Its significance are:-

- **Data Integrity**- It basically prevents any intentional or accidental modifications on the data which is under investigation, keeping evidence in original state.
- **Legally Practice**- It helps in keeping a healthy legal ethics and practice and also ensure that the evidence remains untouched by anyone.