Charles Wallis
CS4398.001
03/19/2023
Assignment 4

1. First, I created files in size 2MB, 10MB, and 5GB.
2. Then, I used the [ls -lsi] command to list the inode numbers of the files in my directory

```
forensics@forensics:/media/forensics/4e32daa4-4366-4a30-8f06-1b0f951f8da9$ dd if=/dev/urandom of=5Gb_file bs=1G count=5
5+0 records in
5+0 records out
5368709120 bytes (5.4 GB, 5.0 GiB) copied, 52.721 s, 102 MB/s
forensics@forensics:/media/forensics/4e32daa4-4366-4a30-8f06-1b0f951f8da9$ ls -lsi
total 5260488
21   10256 -rw-rw-r-- 1 forensics forensics   10485760 Mar 19 16:09 10mb_file
11    2052 -rw-rw-r-- 1 forensics forensics    2097152 Mar 19 16:09 2mb_file
23 5248012 -rw-rw-r-- 1 forensics forensics 5368709120 Mar 19 16:11 5Gb_file
13     164 -rwxrw-r-- 1 forensics forensics     160368 Mar 17 23:01 inode
22       4 -rw-rw-r-- 1 forensics forensics         11 Mar 19 16:10 text.txt
```

3. In this screenshot, we can see on the very left the inode numbers
    a. 10mb file's inode number is 21
    b. 2mb file's inode number is 11
    c. 5gb file's inode number is 23
    d. Text file's inode number is 22
4. I opened the inode tool, entered [inode 22] to open the text.txt file I created, which gave the following result:

```
Inode Analyser# inode 22
 i_mode = 33204
 i_uid = 1000
 i_size = 10
 i_atime = 1679260232
 i_ctime = 1679260697
 i_mtime = 1679260697
 i_dtime = 0
 i_gid = 1000
 i_links_count = 1
 i_blocks = 8
    i_block[0] = 1903617
    i_block[1] = 0
    i_block[2] = 0
    i_block[3] = 0
```

    a. The i_size is 10 here, and it is made up of only one block, at block number 1903617.
5. I then typed [read 1903617] on the analyser tool, which gave 464F52454E534943530A, Which is the word "FORENSICS" in the ASCII code.

```
Inode Analyser# read 1903617
Contents of Data Block: 1903617 (Block Size: 4096)
------------------------------------------------
46 4F 52 45 4E 53 49 43 53 0A 00 00 00 00 00 00
```

6. Using the 2mb file that I created, I printed [inode 11] and here are the results for that:

```
Inode Analyser# inode 11
 i_mode = 33204
 i_uid = 1000
 i_size = 2097152
 i_atime = 1679260179
 i_ctime = 1679260179
 i_mtime = 1679260179
 i_dtime = 0
 i_gid = 1000
 i_links_count = 1
 i_blocks = 4104
    i_block[0] = 26624
    i_block[1] = 26625
    i_block[2] = 26626
    i_block[3] = 26627
    i_block[4] = 26628
    i_block[5] = 26629
    i_block[6] = 26630
    i_block[7] = 26631
    i_block[8] = 26632
    i_block[9] = 26633
    i_block[10] = 26634
    i_block[11] = 26635
    i_block[12] = 1441
    i_block[13] = 0
    i_block[14] = 0
 i_flags = 0
 i_generation = 2083186988
 i_file_acl = 0
 i_dir_acl = 0
 i_faddr = 0
 i_extra_isize = 32
 i_pad1 = 0
```

   a. This time, we can see that there are 13 blocks being used, from 26624 to 26635 and 1441.
7. I read the block number 13 (i_block[12]), which printed the following:

```
Inode Analyser# read 1441
 Contents of Data Block: 1441 (Block Size: 4096)
 ------------------------------------------------
 0C 68 00 00 0D 68 00 00 0E 68 00 00 0F 68 00 00
 30 46 00 00 31 46 00 00 32 46 00 00 33 46 00 00
 34 46 00 00 35 46 00 00 36 46 00 00 37 46 00 00
 38 46 00 00 39 46 00 00 3A 46 00 00 3B 46 00 00
 3C 46 00 00 3D 46 00 00 3E 46 00 00 3F 46 00 00
 20 4A 00 00 21 4A 00 00 22 4A 00 00 23 4A 00 00
 24 4A 00 00 25 4A 00 00 26 4A 00 00 27 4A 00 00
 28 4A 00 00 29 4A 00 00 2A 4A 00 00 2B 4A 00 00
 2C 4A 00 00 2D 4A 00 00 2E 4A 00 00 2F 4A 00 00
 30 4A 00 00 31 4A 00 00 32 4A 00 00 33 4A 00 00
 34 4A 00 00 35 4A 00 00 36 4A 00 00 37 4A 00 00
 38 4A 00 00 39 4A 00 00 3A 4A 00 00 3B 4A 00 00
 3C 4A 00 00 3D 4A 00 00 3E 4A 00 00 3F 4A 00 00
 60 46 00 00 61 46 00 00 62 46 00 00 63 46 00 00
 64 46 00 00 65 46 00 00 66 46 00 00 67 46 00 00
 68 46 00 00 69 46 00 00 6A 46 00 00 6B 46 00 00
 6C 46 00 00 6D 46 00 00 6E 46 00 00 6F 46 00 00
 70 46 00 00 71 46 00 00 72 46 00 00 73 46 00 00
 74 46 00 00 75 46 00 00 76 46 00 00 77 46 00 00
 78 46 00 00 79 46 00 00 7A 46 00 00 7B 46 00 00
 7C 46 00 00 7D 46 00 00 7E 46 00 00 7F 46 00 00
 80 46 00 00 81 46 00 00 82 46 00 00 83 46 00 00
 84 46 00 00 85 46 00 00 86 46 00 00 87 46 00 00
```

   a. These bytes are written in a pattern, and when converting the decimal values we can see that it is actually the location of other blocks, saved as hex values.

8. Moving onto the 10MB file, I repeated this, printing [inode 21].

```
Inode Analyser# inode 21
 i_mode = 33204
 i_uid = 1000
 i_size = 10485760
 i_atime = 1679260194
 i_ctime = 1679260194
 i_mtime = 1679260194
 i_dtime = 0
 i_gid = 1000
 i_links_count = 1
 i_blocks = 20512
    i_block[0] = 1904129
    i_block[1] = 1904130
    i_block[2] = 1904131
    i_block[3] = 1904132
    i_block[4] = 1904133
    i_block[5] = 1904134
    i_block[6] = 1904135
    i_block[7] = 1904136
    i_block[8] = 1904137
    i_block[9] = 1904138
    i_block[10] = 1904139
    i_block[11] = 1904140
    i_block[12] = 1901569
    i_block[13] = 1901584
    i_block[14] = 0
 i_flags = 0
 i_generation = 2552215817
 i_file_acl = 0
 i_dir_acl = 0
 i_faddr = 0
 i_extra_isize = 32
 i_pad1 = 0
```

   a. This shows that we have one more block than the 2MB file, with 14 total blocks. The blocks are located from 1904129 to 1904140, and there are two pointer blocks, one at 1901569 and one at 1901584.
9. I [read 1901569], which was block number 13, which holds information pointing to other blocks that are holding the actual file data.

```
Inode Analyser# read 1901569
 Contents of Data Block: 1901569 (Block Size: 4096)
 ------------------------------------------------
 0D 0E 1D 00 0E 0E 1D 00 0F 0E 1D 00 10 0E 1D 00
 50 46 00 00 51 46 00 00 52 46 00 00 53 46 00 00
 54 46 00 00 55 46 00 00 56 46 00 00 57 46 00 00
 58 46 00 00 59 46 00 00 5A 46 00 00 5B 46 00 00
 5C 46 00 00 5D 46 00 00 5E 46 00 00 5F 46 00 00
 A0 46 00 00 A1 46 00 00 A2 46 00 00 A3 46 00 00
 A4 46 00 00 A5 46 00 00 A6 46 00 00 A7 46 00 00
 A8 46 00 00 A9 46 00 00 AA 46 00 00 AB 46 00 00
 AC 46 00 00 AD 46 00 00 AE 46 00 00 AF 46 00 00
 B0 46 00 00 B1 46 00 00 B2 46 00 00 B3 46 00 00
 B4 46 00 00 B5 46 00 00 B6 46 00 00 B7 46 00 00
 B8 46 00 00 B9 46 00 00 BA 46 00 00 BB 46 00 00
 BC 46 00 00 BD 46 00 00 BE 46 00 00 BF 46 00 00
 60 4A 00 00 61 4A 00 00 62 4A 00 00 63 4A 00 00
 64 4A 00 00 65 4A 00 00 66 4A 00 00 67 4A 00 00
```

10. I also then read block number 14 [read 1901584], which had a shorter code that is easier to read

```
Inode Analyser# read 1901584
 Contents of Data Block: 1901584 (Block Size: 4096)
 ------------------------------------------------
 31 04 1D 00 0F 09 1D 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

11. I translated this hex to a decimal number, which gave me 1902863, so we know that block 14 is a pointer block to another block.

12. Now moving onto the 5gb file, I first read [inode 23], to notice that it has 15 blocks, with 1073741824 bytes, or roughly 1GB. The blocks continue from 1904145 to 1904156, then it has 3 blocks at the end

```
Inode Analyser# inode 23
 i_mode = 33204
 i_uid = 1000
 i_size = 1073741824
 i_atime = 1679260265
 i_ctime = 1679260274
 i_mtime = 1679260274
 i_dtime = 0
 i_gid = 1000
 i_links_count = 1
 i_blocks = 10496024
    i_block[0] = 1904145
    i_block[1] = 1904146
    i_block[2] = 1904147
    i_block[3] = 1904148
    i_block[4] = 1904149
    i_block[5] = 1904150
    i_block[6] = 1904151
    i_block[7] = 1904152
    i_block[8] = 1904153
    i_block[9] = 1904154
    i_block[10] = 1904155
    i_block[11] = 1904156
    i_block[12] = 1901570
    i_block[13] = 1901571
    i_block[14] = 1934131
 i_flags = 0
 i_generation = 154884685
 i_file_acl = 0
 i_dir_acl = 1
 i_faddr = 0
 i_extra_isize = 32
 i_pad1 = 0
```

13. [read 1901570] at block number 13 gave a pointer again

```
Inode Analyser# read 1901570
 Contents of Data Block: 1901570 (Block Size: 4096)
 ------------------------------------------------
 1D 0E 1D 00 1E 0E 1D 00 1F 0E 1D 00 20 0E 1D 00
 50 4A 00 00 51 4A 00 00 52 4A 00 00 53 4A 00 00
 54 4A 00 00 55 4A 00 00 56 4A 00 00 57 4A 00 00
 58 4A 00 00 59 4A 00 00 5A 4A 00 00 5B 4A 00 00
 5C 4A 00 00 5D 4A 00 00 5E 4A 00 00 5F 4A 00 00
 A0 4A 00 00 A1 4A 00 00 A2 4A 00 00 A3 4A 00 00
 A4 4A 00 00 A5 4A 00 00 A6 4A 00 00 A7 4A 00 00
 A8 4A 00 00 A9 4A 00 00 AA 4A 00 00 AB 4A 00 00
 AC 4A 00 00 AD 4A 00 00 AE 4A 00 00 AF 4A 00 00
 B0 4A 00 00 B1 4A 00 00 B2 4A 00 00 B3 4A 00 00
```

14. [read 1901571] at block 14 gave some more pointers to blocks

```
Inode Analyser# read 1901571
Contents of Data Block: 1901571 (Block Size: 4096
------------------------------------------------
 32 04 1D 00 33 04 1D 00 04 02 1D 00 05 02 1D 00
 06 02 1D 00 07 02 1D 00 04 04 1D 00 05 04 1D 00
 06 04 1D 00 07 04 1D 00 34 04 1D 00 35 04 1D 00
 36 04 1D 00 37 04 1D 00 08 04 1D 00 09 04 1D 00
 0A 04 1D 00 0B 04 1D 00 0C 04 1D 00 0D 04 1D 00
 0E 04 1D 00 0F 04 1D 00 38 04 1D 00 39 04 1D 00
 3A 04 1D 00 3B 04 1D 00 3C 04 1D 00 3D 04 1D 00
 3E 04 1D 00 3F 04 1D 00 10 09 1D 00 11 09 1D 00
 12 09 1D 00 13 09 1D 00 14 09 1D 00 15 09 1D 00
 16 09 1D 00 17 09 1D 00 18 09 1D 00 19 09 1D 00
```

15. The first block number here is 1D 04 32, which is 1901618, and when I tried [read 19017618], it gave me this block, which seems point at another block, 13 21 08.

```
Inode Analyser# read 1901618
 Contents of Data Block: 1901618 (Block Size: 4096)
 ------------------------------------------------
 0C 04 02 00 0D 04 02 00 0E 04 02 00 0F 04 02 00
 10 04 02 00 11 04 02 00 12 04 02 00 13 04 02 00
 14 04 02 00 15 04 02 00 16 04 02 00 17 04 02 00
 18 04 02 00 19 04 02 00 1A 04 02 00 1B 04 02 00
 1C 04 02 00 1D 04 02 00 1E 04 02 00 1F 04 02 00
 20 04 02 00 21 04 02 00 22 04 02 00 23 04 02 00
 24 04 02 00 25 04 02 00 26 04 02 00 27 04 02 00
 28 04 02 00 29 04 02 00 2A 04 02 00 2B 04 02 00
 2C 04 02 00 2D 04 02 00 2E 04 02 00 2F 04 02 00
 30 04 02 00 31 04 02 00 32 04 02 00 33 04 02 00
 34 04 02 00 35 04 02 00 36 04 02 00 37 04 02 00
 38 04 02 00 39 04 02 00 3A 04 02 00 3B 04 02 00
```

16. I [read 132108] from above, and I reached a data block, which is in the 5GB file, created using DD urandom

```
Inode Analyser# read 132108
 Contents of Data Block: 132108 (Block Size: 4096)
 ------------------------------------------------
 EF 3A 28 AE 22 C1 DA 3B D2 C7 DA 4E 81 28 17 EF
 5D 72 85 6A B7 87 1E CC 88 FD 17 36 DE FD 93 48
 2E CB 53 A6 DF 5E 2F 13 0A 34 25 DE 06 3A 23 F7
 FE FA ED C7 3F E2 E8 6C 85 15 AD 5E 80 CB 68 43
 78 B7 89 F8 6D 2C 51 6C 0A 52 FE 21 26 AE 4A 24
 44 82 E7 B7 2E 14 76 73 FE 5A E6 C9 19 F9 AA D0
 F5 BA 00 1B 6F 97 8C 41 84 F7 C5 63 B3 79 FD 61
 35 AB E8 8E C0 19 4F 78 3E 2F 6C 28 08 B5 76 2F
 10 90 A6 DB 6E C3 C8 AC CD 70 23 D8 FF 76 98 94
 26 10 C7 8D EA 7D 23 1C 2A 0D 53 AA D7 41 FB D2
 0C 5E 40 E6 BF 94 86 4D 11 B0 1D 0A 29 38 7A A7
 D2 B3 AF B8 8D 20 B1 49 C9 6D 88 B8 98 15 94 2A
```

17. Then I [read 1934131], which is block number 15, which is where I see 1D 83 34. This is another block.

```
Inode Analyser# read 1934131
 Contents of Data Block: 1934131 (Block Size: 4096)
 ------------------------------------------------
 34 83 1D 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

18. I convert 1D8334 from hex to decimal, which becomes 1934132, so I read it, and I see rows in pattern.

```
Inode Analyser# read 1934132
 Contents of Data Block: 1934132 (Block Size: 4096)
 ------------------------------------------------
 35 83 1D 00 36 83 1D 00 37 83 1D 00 38 83 1D 00
 39 83 1D 00 3A 83 1D 00 3B 83 1D 00 3C 83 1D 00
 3D 83 1D 00 3E 83 1D 00 3F 83 1D 00 40 83 1D 00
 41 83 1D 00 42 83 1D 00 43 83 1D 00 44 83 1D 00
 45 83 1D 00 46 83 1D 00 47 83 1D 00 48 83 1D 00
 49 83 1D 00 4A 83 1D 00 4B 83 1D 00 4C 83 1D 00
 4D 83 1D 00 4E 83 1D 00 4F 83 1D 00 50 83 1D 00
 51 83 1D 00 52 83 1D 00 53 83 1D 00 54 83 1D 00
 55 83 1D 00 56 83 1D 00 57 83 1D 00 58 83 1D 00
 59 83 1D 00 5A 83 1D 00 5B 83 1D 00 5C 83 1D 00
 5D 83 1D 00 5E 83 1D 00 5F 83 1D 00 60 83 1D 00
 61 83 1D 00 62 83 1D 00 63 83 1D 00 64 83 1D 00
```

19. Finally, this is where I reached, and this is a double indirect pointer. So the values here are taking me to another pointer block, and the values inside that pointer block leads me to the actual file data.