CSU, College of Engineering and Computer Science
Department of Computer Science
CSC 153 – Computer Forensics Principles and Practice

Catherine Nguyen
CSC 153
11/23/2020

Activity 6: Recovering Graphics Files

Objectives:

- Split and combine files in Linux.
- Use WinHex to recover graphics files.

Part 1: Split and combine files in Linux.

Since the Winhex evaluation version cannot process a file that is bigger than 200KB, you need to split a file into pieces in order to edit and save a big file. So the first thing we have to do is download 'funpicture' on canvas onto CAINE. The firefox on CAINE wouldn't let me access the files on canvas so I had to download it another way.

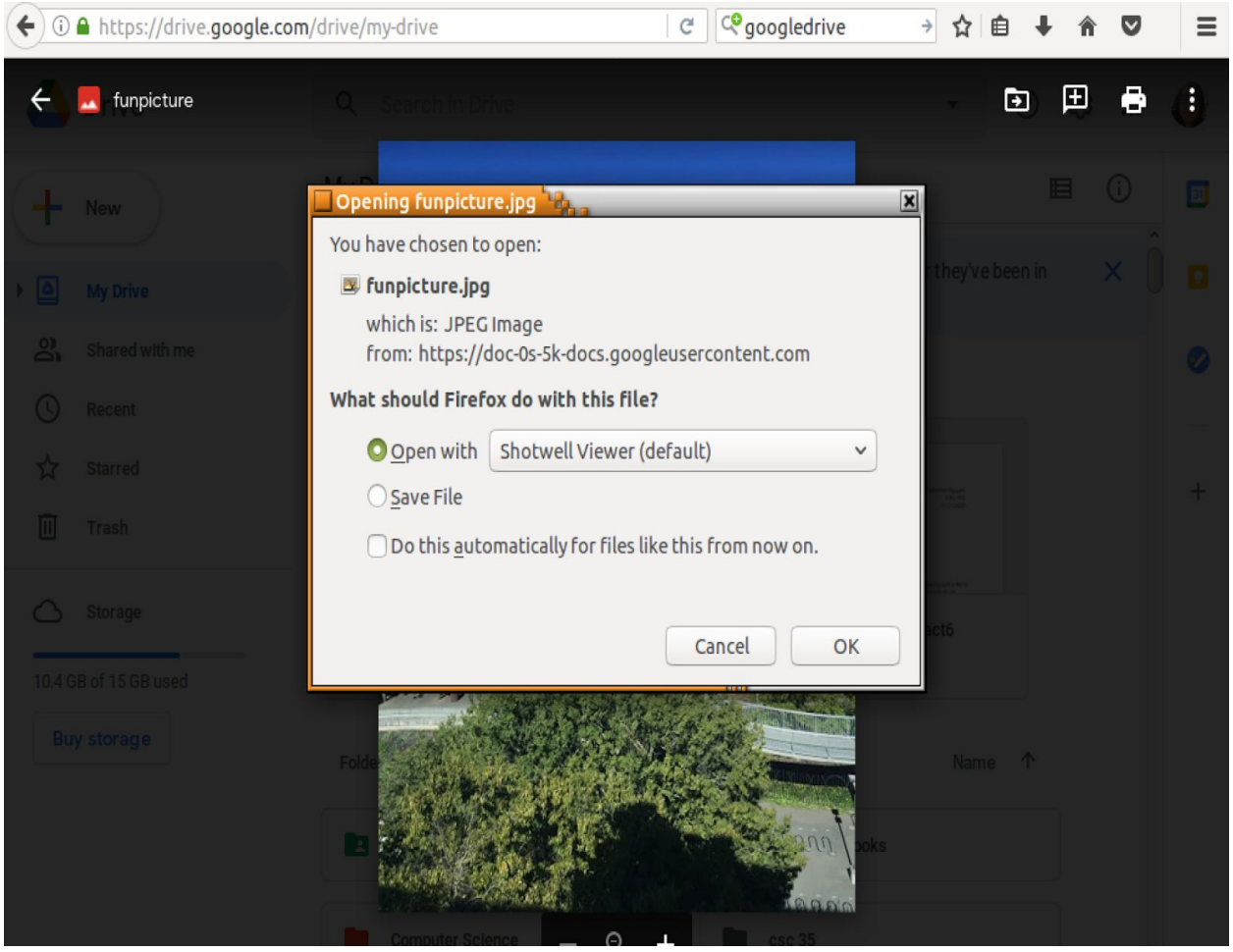


Figure 1. Downloading funpicture.jpg from google drive.

Once the download is complete it is time to open up the terminal and locate the directory in which that file is stored. This will allow us to start our splitting.

```
cainecf@cainecf:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
cainecf@cainecf:~$ cd Downloads
cainecf@cainecf:~/Downloads$ ls
4Basic Forensic Analysis using Autopsy.pdf  image_zero.dd
case.aut                                    investigators.txt
case.log                                    vol1-C..boring.xlxs
funpicture.jpg                              vol1-C..where.were.you.docx
image_zero(1).dd.part
cainecf@cainecf:~/Downloads$
```

Figure 2. Changing the directory to where funpicture.jpg is downloaded.

Now that we are in the designated directory, we must split the file via command ‘split -b 19000 funpicture’. Now because I have to download this file another way I need to add the full name of the file which is funpicture.jpg. Note: I changed the location of the file because I didn’t want the splits to get mixed up.

```
root@cainecf:/home/cainecf/Downloads/fp# ls
funpicture.jpg
root@cainecf:/home/cainecf/Downloads/fp# split -b 19000 funpicture.jpg
root@cainecf:/home/cainecf/Downloads/fp# ls
funpicture.jpg  xai  xar  xba  xbj  xbs  xcb  xck  xct  xdc  xdl  xdu  xed  xem  xev  xfe
xaa             xaj  xas  xbb  xbk  xbt  xcc  xcl  xcu  xdd  xdm  xdv  xee  xen  xew  xff
xab             xak  xat  xbc  xbl  xbu  xcd  xcm  xcv  xde  xdn  xdw  xef  xeo  xex
xac             xal  xau  xbd  xbm  xbv  xce  xcn  xcw  xdf  xdo  xdx  xeg  xep  xey
xad             xam  xav  xbe  xbn  xbw  xcf  xco  xcx  xdg  xdp  xdy  xeh  xeq  xez
xae             xan  xaw  xbf  xbo  xbx  xcg  xcp  xcy  xdh  xdq  xdz  xei  xer  xfa
xaf             xao  xax  xbg  xbp  xby  xch  xcq  xcz  xdi  xdr  xea  xej  xes  xfb
xag             xap  xay  xbh  xbq  xbz  xci  xcr  xda  xdj  xds  xeb  xek  xet  xfc
xah             xaq  xaz  xbi  xbr  xca  xcj  xcs  xdb  xdk  xdt  xec  xel  xeu  xfd
root@cainecf:/home/cainecf/Downloads/fp#
```

Figure 3. Splitting funpicture.jpg file, and listing the results from the split.

After splitting the image we need to combine the file pieces to a new file by typing the command ‘cat x*>newfun’.

```
root@cainecf:/home/cainecf/Downloads/fp# cat x*>newfun
root@cainecf:/home/cainecf/Downloads/fp# ls
funpicture.jpg  xah  xaq  xaz  xbi  xbr  xca  xcj  xcs  xdb  xdk  xdt  xec  xel  xeu  xfd
newfun          xai  xar  xba  xbj  xbs  xcb  xck  xct  xdc  xdl  xdu  xed  xem  xev  xfe
xaa             xaj  xas  xbb  xbk  xbt  xcc  xcl  xcu  xdd  xdm  xdv  xee  xen  xew  xff
xab             xak  xat  xbc  xbl  xbu  xcd  xcm  xcv  xde  xdn  xdw  xef  xeo  xex
xac             xal  xau  xbd  xbm  xbv  xce  xcn  xcw  xdf  xdo  xdx  xeg  xep  xey
xad             xam  xav  xbe  xbn  xbw  xcf  xco  xcx  xdg  xdp  xdy  xeh  xeq  xez
xae             xan  xaw  xbf  xbo  xbx  xcg  xcp  xcy  xdh  xdq  xdz  xei  xer  xfa
xaf             xao  xax  xbg  xbp  xby  xch  xcq  xcz  xdi  xdr  xea  xej  xes  xfb
xag             xap  xay  xbh  xbq  xbz  xci  xcr  xda  xdj  xds  xeb  xek  xet  xfc
root@cainecf:/home/cainecf/Downloads/fp#
```

Figure 4. Combining the image with the cat command.

Finally, we need to rename the combined file to include the .jgp extension, by running the command ‘mv newfile newfun.jpg’

```
root@cainecf:/home/cainecf/Downloads/fp# mv newfun newfun.jpg
root@cainecf:/home/cainecf/Downloads/fp# ls
funpicture.jpg  xah  xaq  xaz  xbi  xbr  xca  xcj  xcs  xdb  xdk  xdt  xec  xel  xeu  xfd
newfun.jpg      xai  xar  xba  xbj  xbs  xcb  xck  xct  xdc  xdl  xdu  xed  xem  xev  xfe
xaa             xaj  xas  xbb  xbk  xbt  xcc  xcl  xcu  xdd  xdm  xdv  xee  xen  xew  xff
xab             xak  xat  xbc  xbl  xbu  xcd  xcm  xcv  xde  xdn  xdw  xef  xeo  xex
xac             xal  xau  xbd  xbm  xbv  xce  xcn  xcw  xdf  xdo  xdx  xeg  xep  xey
xad             xam  xav  xbe  xbn  xbw  xcf  xco  xcx  xdg  xdp  xdy  xeh  xeq  xez
xae             xan  xaw  xbf  xbo  xbx  xcg  xcp  xcy  xdh  xdq  xdz  xei  xer  xfa
xaf             xao  xax  xbg  xbp  xby  xch  xcq  xcz  xdi  xdr  xea  xej  xes  xfb
xag             xap  xay  xbh  xbq  xbz  xci  xcr  xda  xdj  xds  xeb  xek  xet  xfc
root@cainecf:/home/cainecf/Downloads/fp#
```

Figure 5. Renamed newfun to newfun.jpg.

Part 2: Practice recovering graphics files using Winhex.

We start by downloading ‘smallsmallsmall’ from canvas and save it into a folder. The header is a .png file, but its been modified by the suspect.

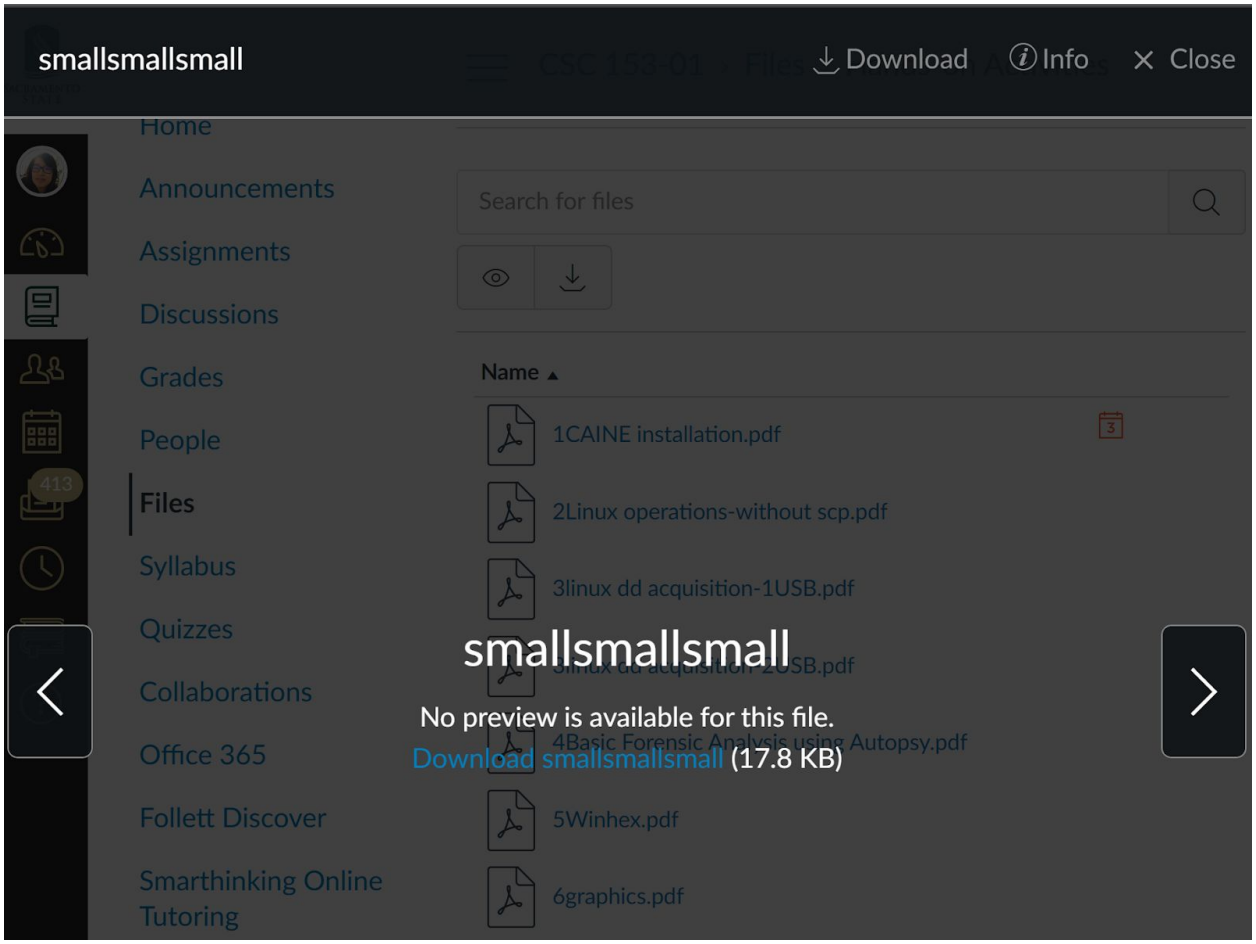


Figure 6. Downloading smallsmallsmall to our local machine.

Now we need to open up WinHex and open it as administrator. As a safety precaution, we can click on options and edit mode. Selecting the Read-Only Mode and click ok.

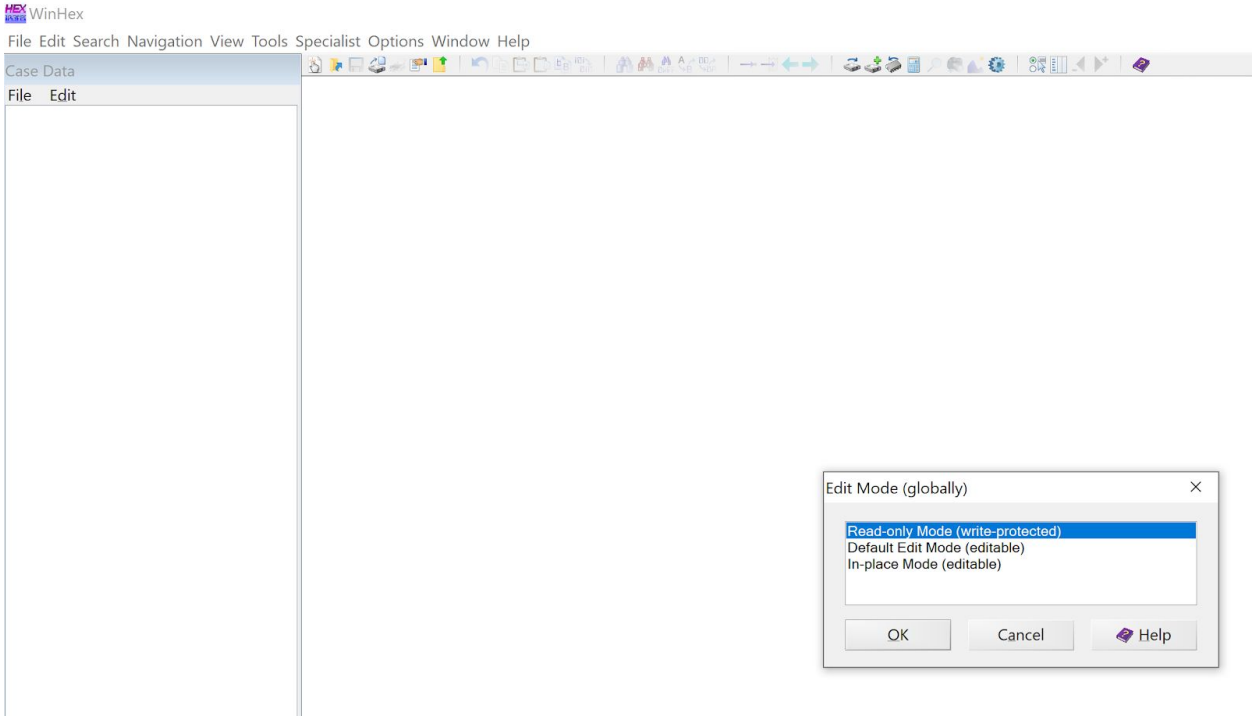


Figure 7. Selecting the read-only mode for safety.

Next, we need to click on Tools and Open Disk. Then we can select the drive in which the file ‘smallsmallsmall’ is located so we can examine it. In this case, the drive will be C:. After clicking ok WinHex will then being traversing the drive. This may take a while depending on the size of the drize.

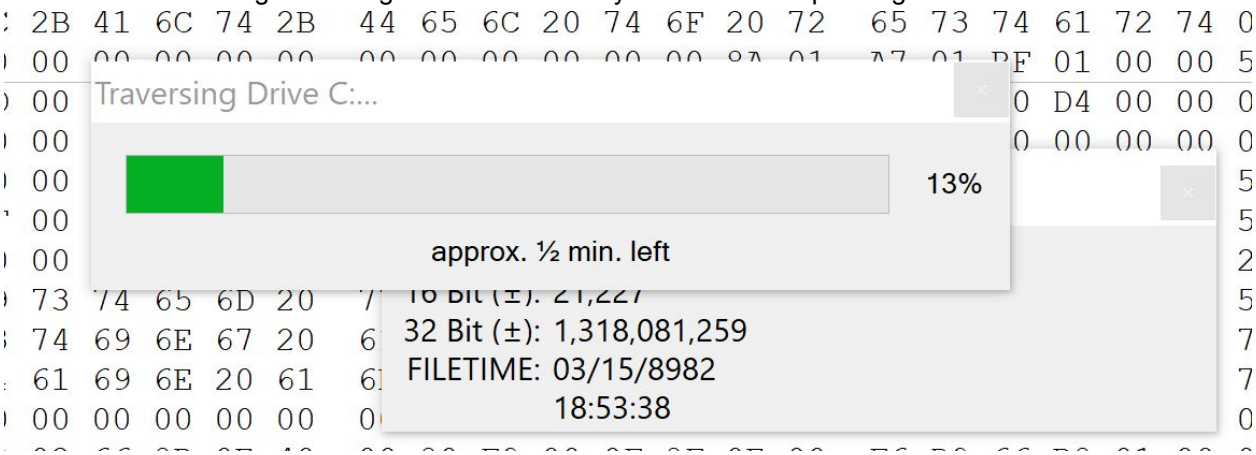


Figure 8. Winhex is traversing the C: drive.

After the traversing is complete we can begin scrolling down to fin the $MFT file, right-click it and choose open. Then open the file in a separate window to examine. This simulates a real investigation because we already know where the file is located so doing this method is preferred for practice.

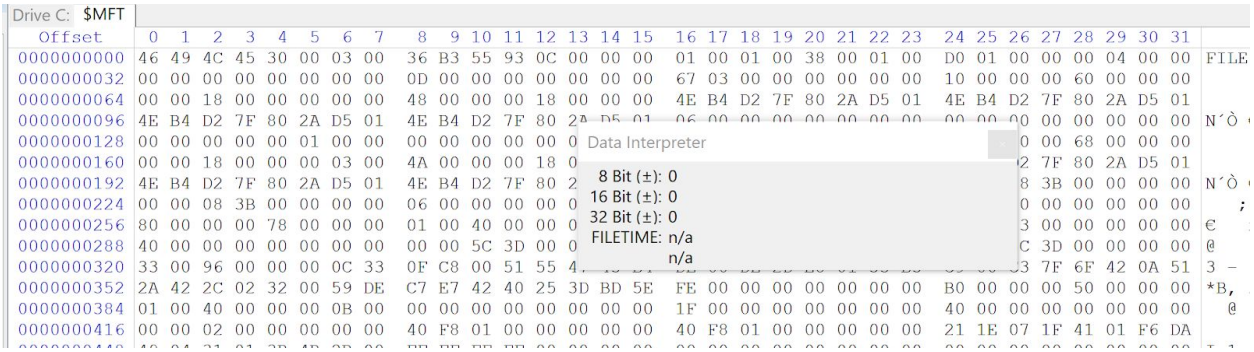


Figure 9. $MFT is open in a separate window.

The next task is to find the smallsmallsmall file. In the $MFT, characters in the file name is usually separated by hexadecimal value 00. The hexadecimal value for the word small is 73 6d 61 6c 6c. Separating each character with 00 it becomes 73 00 6d 00 61 00 6c 00 6c. By repeating this three times to get smallsmallsmall it then becomes 73006D0061006C006C0073006D0061006C006C0073006D0061006C006C00

Knowing this we can then search for the following hex values in the $MFT file. To do so click on Search -> Find Hex Values.

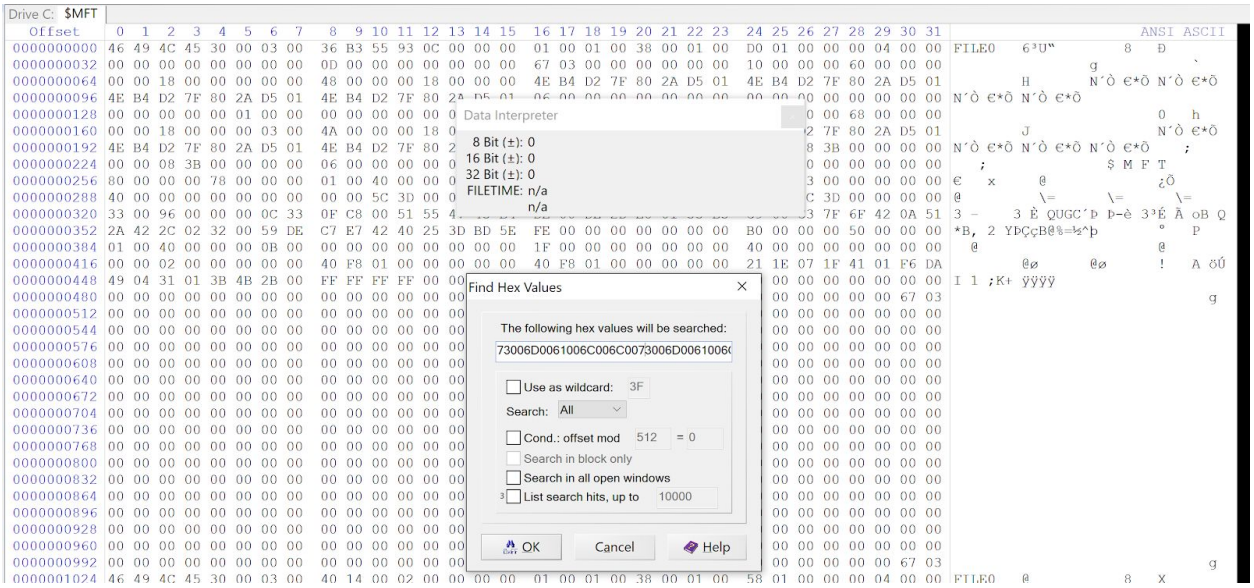


Figure 10. Searching for the hexadecimal title for smallsmallsmall.

After the search, we can see that we got a hit.

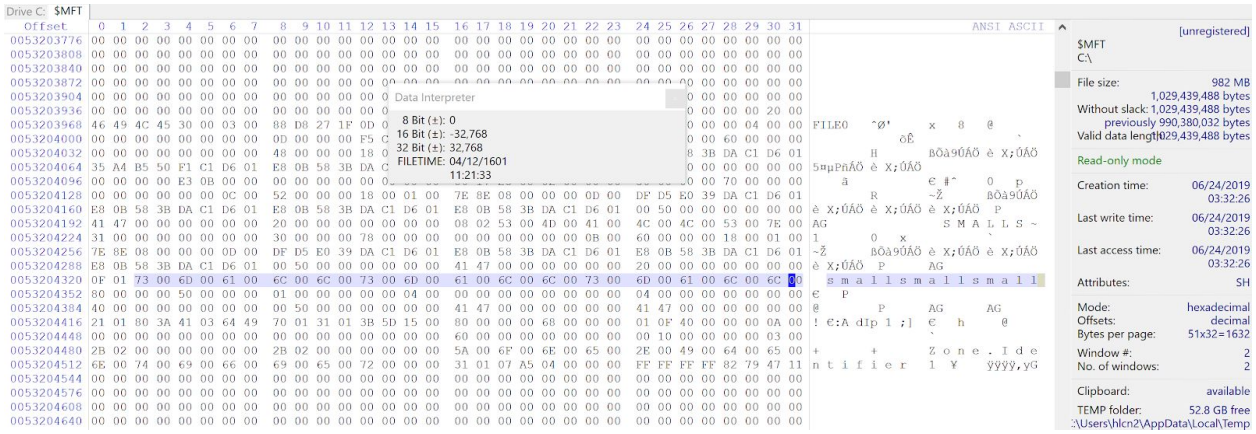


Figure 11. Hit for hexadecimal value.

By following the same methodology as in hands-on activity 5 and lab 2 to examine this $MFT file record. You need to find out the data runs for this file. In the demo, the first data run information is “41 05 37 52 55 04”. This means the starting position of the first data run is “0x 04 55 52 37” and the size is “0x05”. In this case, we can find the start of the data run for this file by looking at offset 0x40 from 0x80.

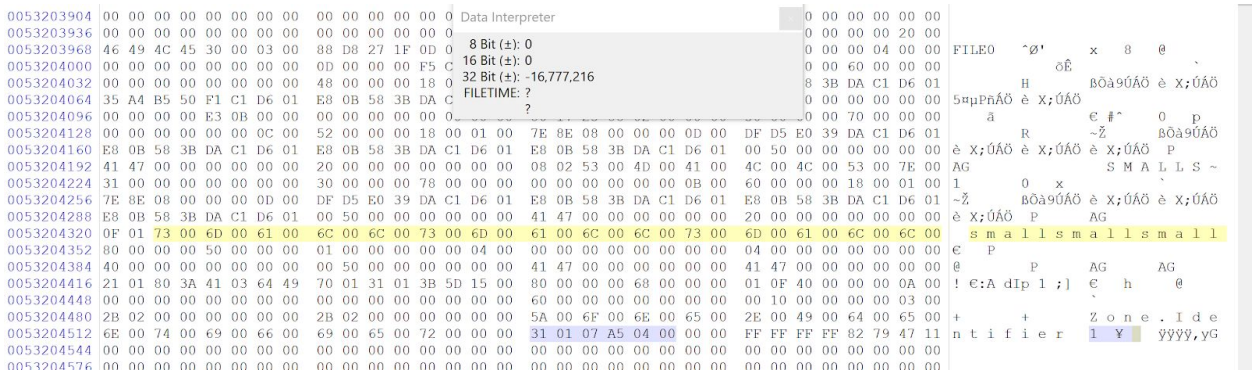


Figure 12. First data run info is 31 01 07 A5 04 00.

The first data run is 31 01 07 A5 04 00, which means the starting position is 0x 00 04 A5 07, and the size 0x05. So next, we open the window for the C: drive once again, and click Navigation then go to offset. We’ll ender the offset appending three zeros to the end of it.

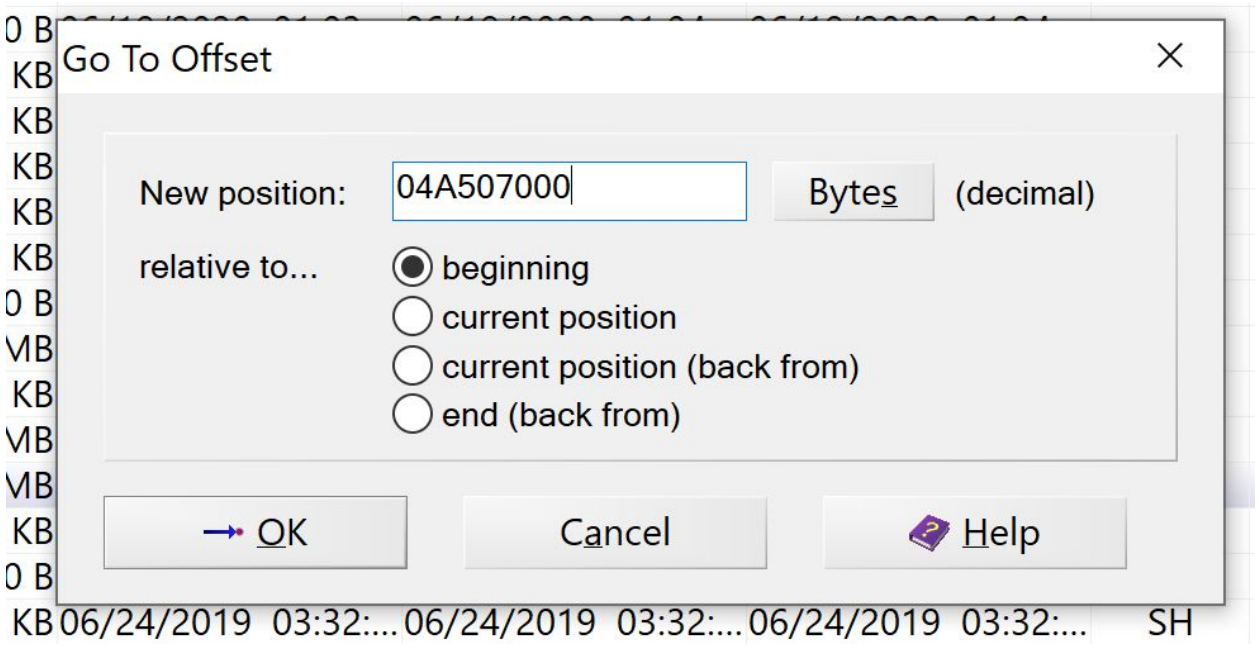


Figure 13. The first data run at offset 04A507000.

Now we're taken to the beginning pg the smallsmallsmall file.

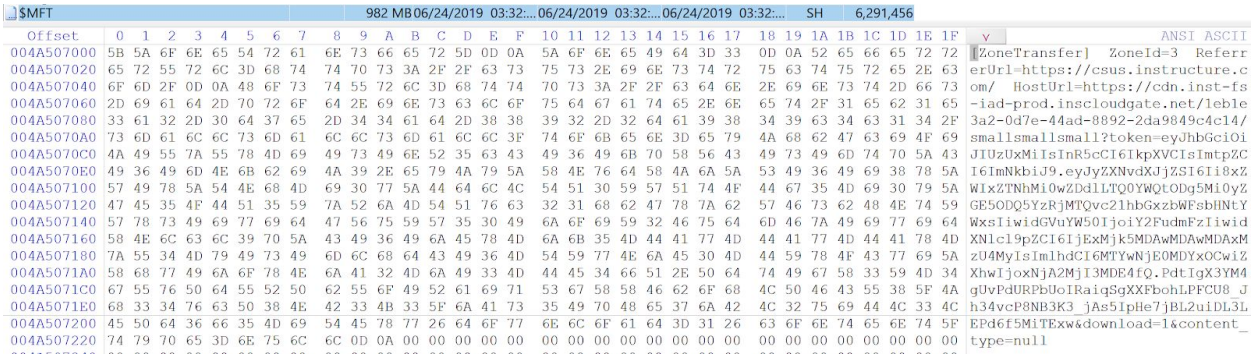


Figure 14. Beginning of smallsmallsmall file.

Since the size of the first data run is 0x05, we can click on Navigation then go to offset again to find the end of the first data run. Setting 05000 as our position and choosing the current position will take us to the ending point for the first data run. In our case, the end of the first data run is at offset 0x004A50C000

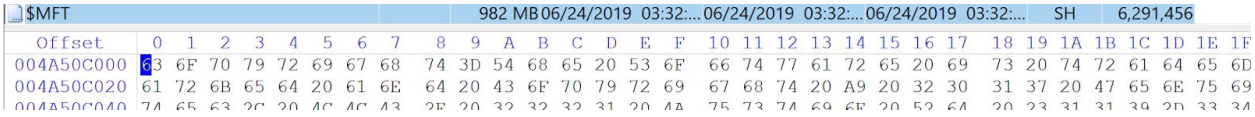


Figure 15. End of the data run.

Starting from the beginning of the file again, we click on the first byte of the file and drag until the offset is 5000, which is the size of the data run.

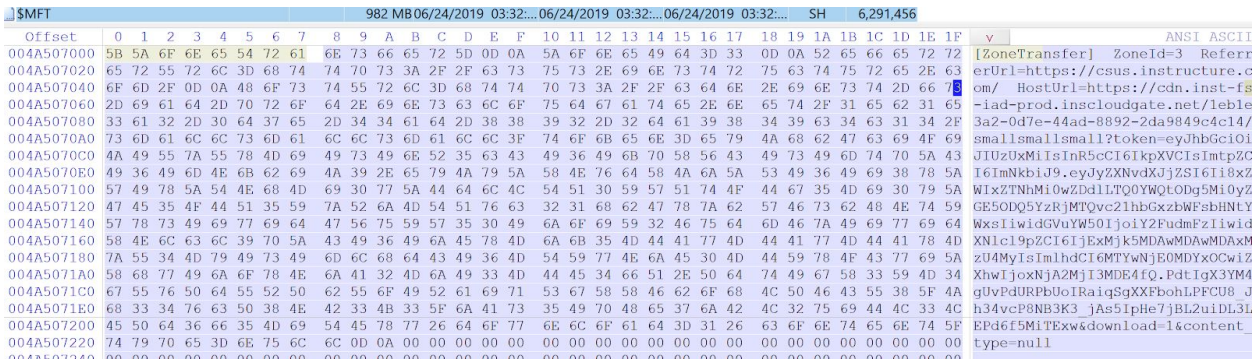


Figure 15. Selecting everything from the first data rune.

Next, we right-click and choose edit copy block into a new file.

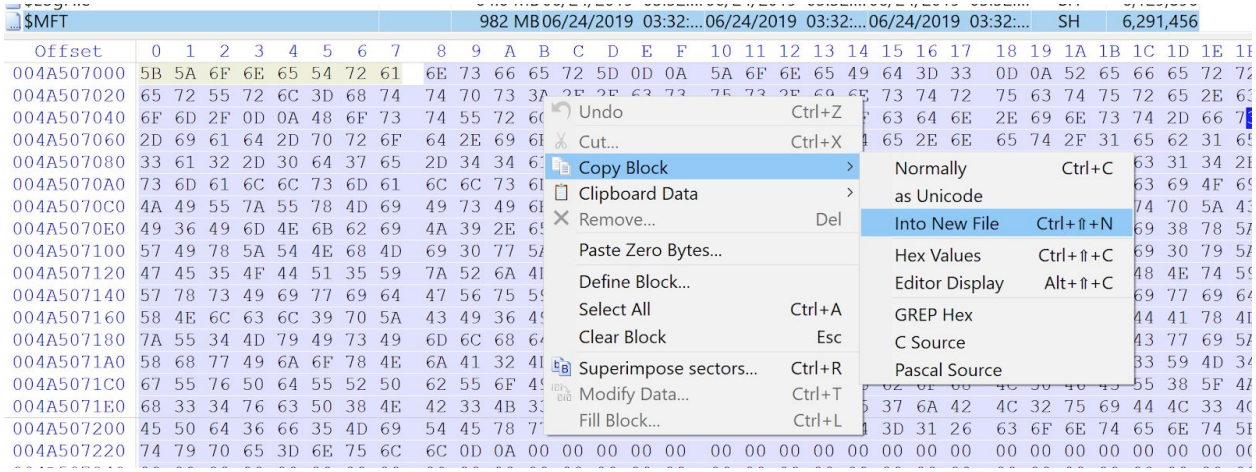


Figure 16. Copy block to a new file.

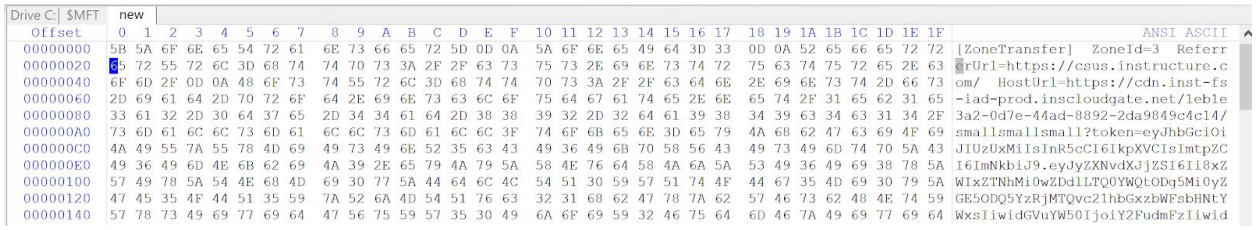
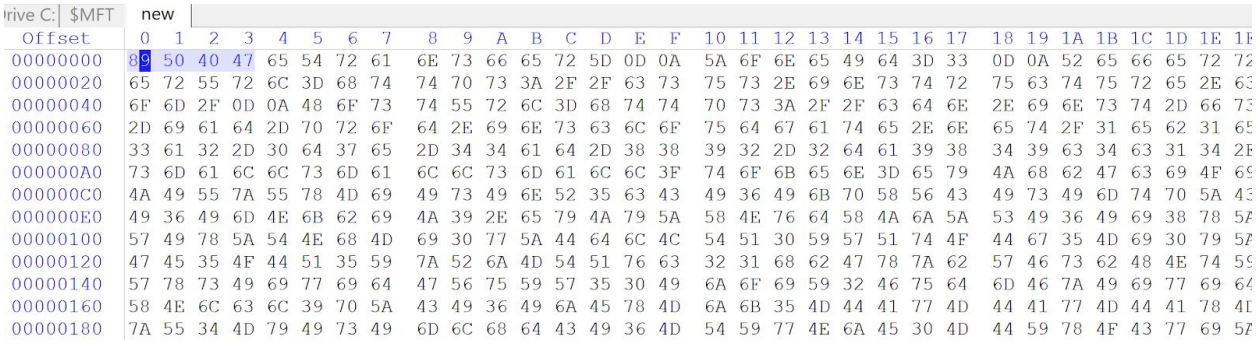


Figure 17. New file from data run copied in figure 20.

The correct header for the PNG file format is 89 50 4E 47, so we change the header of the new file from 5B 5A 6F 6E to 89 50 4E 47, and save it as recovered. In order for me to do this, I had to change the edit mode to allow me to make changes to the file.



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
00000000	89	50	40	47	65	54	72	61	6E	73	66	65	72	5D	0D	0A	5A	6F	6E	65	49	64	3D	33	0D	0A	52	65	66	65	72	72
00000020	65	72	55	72	6C	3D	68	74	74	70	73	3A	2F	2F	63	73	75	73	2E	69	6E	73	74	72	75	63	74	75	72	65	2E	63
00000040	6F	6D	2F	0D	0A	48	6F	73	74	55	72	6C	3D	68	74	74	70	73	3A	2F	2F	63	64	6E	2E	69	6E	73	74	2D	66	73
00000060	2D	69	61	64	2D	70	72	6F	64	2E	69	6E	73	63	6C	6F	75	64	67	61	74	65	2E	6E	65	74	2F	31	65	62	31	65
00000080	33	61	32	2D	30	64	37	65	2D	34	34	61	64	2D	38	38	39	32	2D	32	64	61	39	38	34	39	63	34	63	31	34	2F
000000A0	73	6D	61	6C	6C	73	6D	61	6C	6C	73	6D	61	6C	6C	3F	74	6F	6B	65	6E	3D	65	79	4A	68	62	47	63	69	4F	69
000000C0	4A	49	55	7A	55	78	4D	69	49	73	49	6E	52	35	63	43	49	36	49	6B	70	58	56	43	49	73	49	6D	74	70	5A	43
000000E0	49	36	49	6D	4E	6B	62	69	4A	39	2E	65	79	4A	79	5A	58	4E	76	64	58	4A	6A	5A	53	49	36	49	69	38	78	5A
00000100	57	49	78	5A	54	4E	68	4D	69	30	77	5A	44	64	6C	4C	54	51	30	59	57	51	74	4F	44	67	35	4D	69	30	79	5A
00000120	47	45	35	4F	44	51	35	59	7A	52	6A	4D	54	51	76	63	32	31	68	62	47	78	7A	62	57	46	73	62	48	4E	74	59
00000140	57	78	73	49	69	77	69	64	47	56	75	59	57	35	30	49	6A	6F	69	59	32	46	75	64	6D	46	7A	49	69	77	69	64
00000160	58	4E	6C	63	6C	39	70	5A	43	49	36	49	6A	45	78	4D	6A	6B	35	4D	44	41	77	4D	44	41	77	4D	44	41	78	4D
00000180	7A	55	34	4D	79	49	73	49	6D	6C	68	64	43	49	36	4D	54	59	77	4E	6A	45	30	4D	44	59	78	4F	43	77	69	5A

Figure 18. Replace the file header with the proper header for PNG.

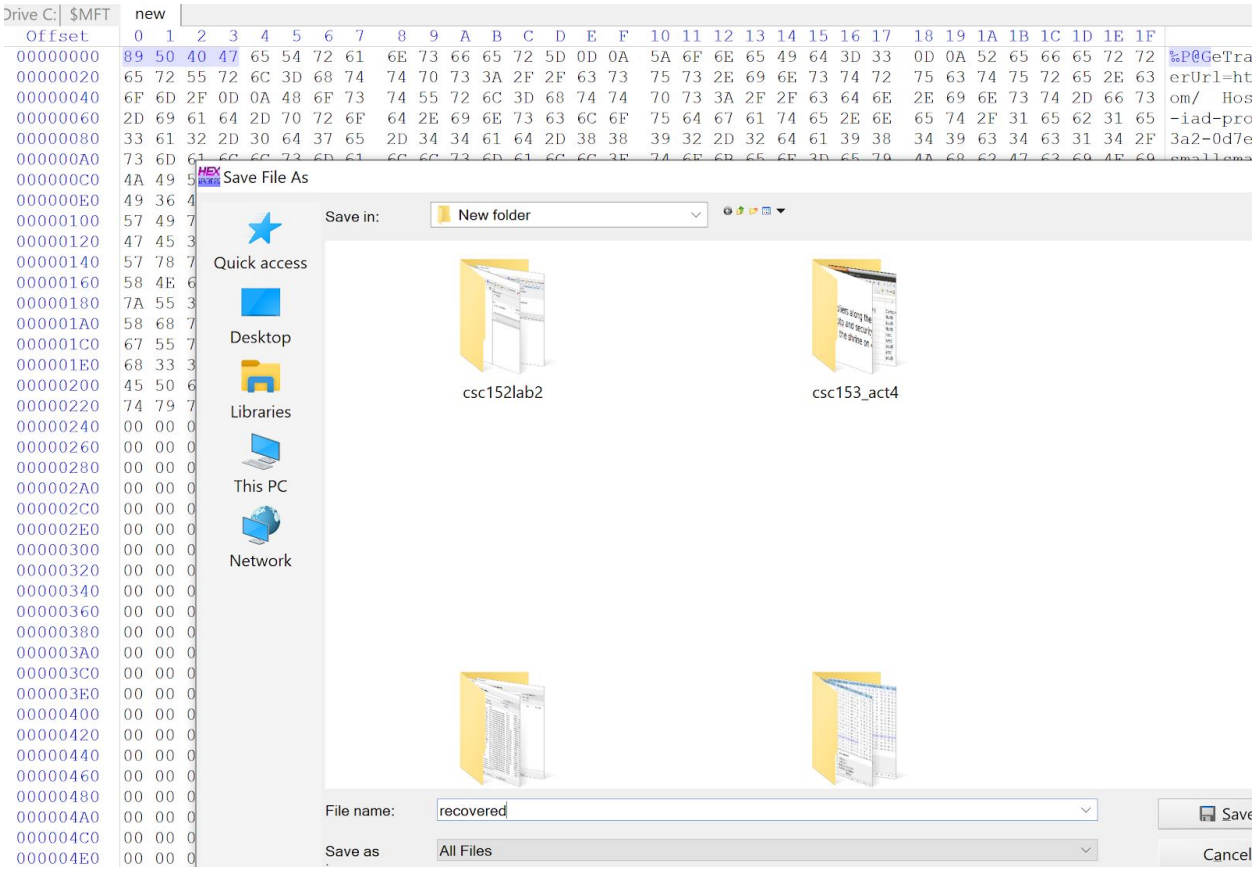


Figure 19. Saving the file as recovered.

Now once we add the .png extension, we can open the file recovered.png and see that it's a 58x75 pixel version of the image from Part 1.

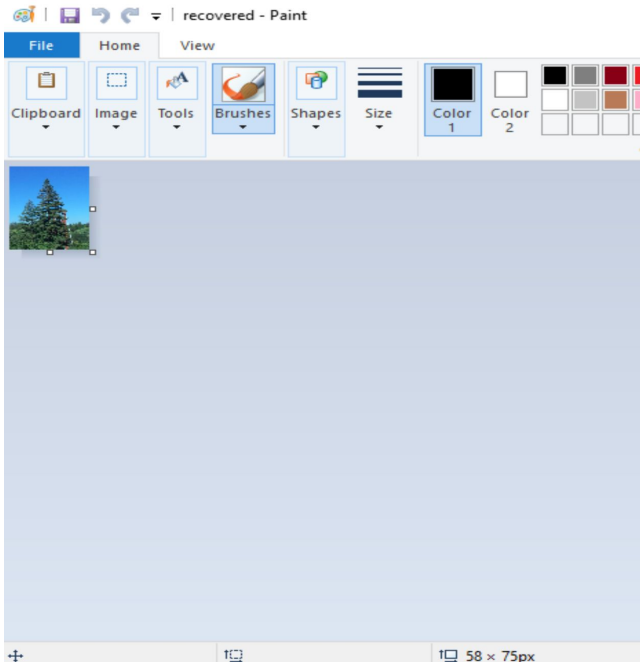


Figure 20. Recovered png.

Questions:

1. **What hexadecimal values did you use to search for the file "smallsmallsmall" in step 13 and 15?**
    a. **The hexadecimal value searched for was** 73006D0061006C006C0073006D0061006C006C0073006D0061006C006C00

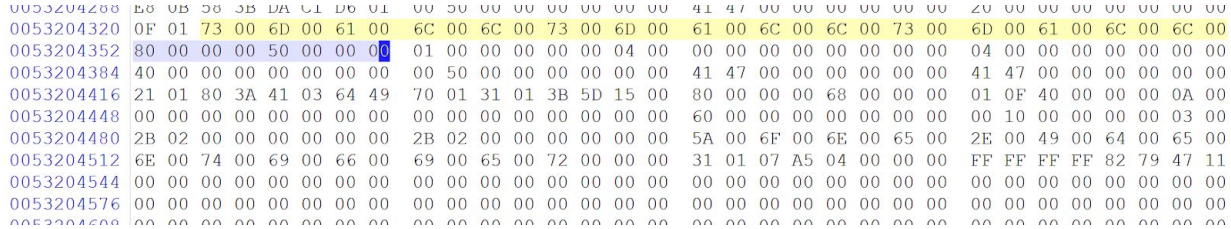


Figure 21. Searching for smallsmallsmall.

2. **Is the file a resident file or a non-resident file? How do you know?**
    a. This is a non-resident file, we can tell by looking at offset 0x08 from attribute 0x80 and see the flag is set to 0x01.

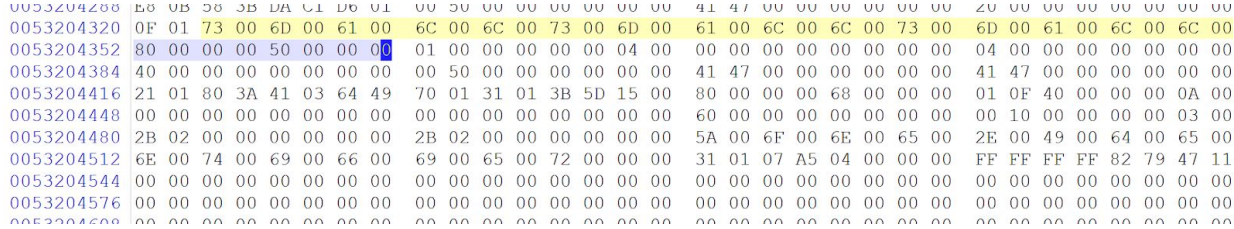


Figure 22. None resident flag.

3. **How many data runs does this file have?**
    a. Only had one data run for the activity,

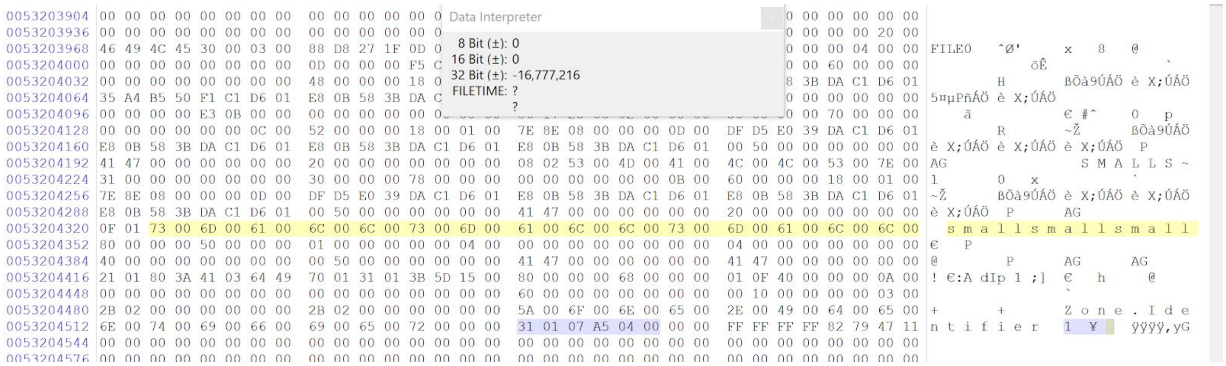
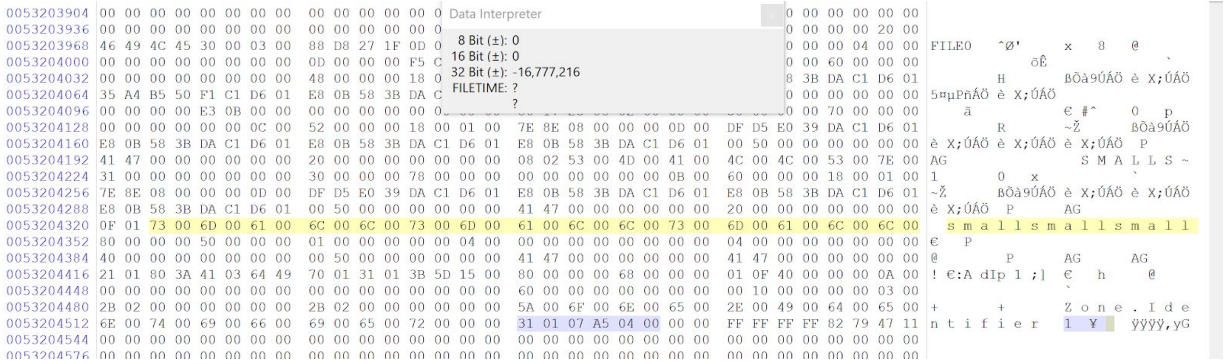


Figure 23. One data run.

4. **What is the starting position for the first data run?**
    a. The starting position for the data run is 0x0004A507



    b.
    c. **Figure 24. The starting position for the first data.**

5. **What is the size of the first data run?**
    a. The size of the first data run is 0x05. On the bottom of WinHex size would be 5000.

6. **In step 20, what is the header of the file "smallsmallsmall" which you downloaded from Canvas? Please provide the first 4 bytes of the hexadecimal values. Please take a screenshot to show the evidence.**
    a. **The header of the file downloaded from canvas was 5B 5A 6F 6E.**

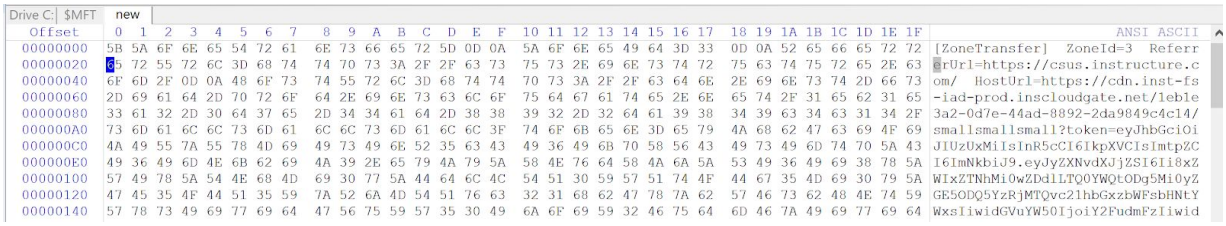


Figure 25. The header for small small small, could be wrong.

7. **What should be the correct header of a PNG file?**
   a. The correct header for a PNG file is 89 50 4E 47.

rive C: | $MFT | new

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
00000000	89	50	40	47	65	54	72	61	6E	73	66	65	72	5D	0D	0A	5A	6F	6E	65	49	64	3D	33	0D	0A	52	65	66	65	72	72
00000020	65	72	55	72	6C	3D	68	74	74	70	73	3A	2F	2F	63	73	75	73	2E	69	6E	73	74	72	75	63	74	75	72	65	2E	63
00000040	6F	6D	2F	0D	0A	48	6F	73	74	55	72	6C	3D	68	74	74	70	73	3A	2F	2F	63	64	6E	2E	69	6E	73	74	2D	66	73
00000060	2D	69	61	64	2D	70	72	6F	64	2E	69	6E	73	63	6C	6F	75	64	67	61	74	65	2E	6E	65	74	2F	31	65	62	31	65
00000080	33	61	32	2D	30	64	37	65	2D	34	34	61	64	2D	38	38	39	32	2D	32	64	61	39	38	34	39	63	34	63	31	34	2E
000000A0	73	6D	61	6C	6C	73	6D	61	6C	6C	73	6D	61	6C	6C	3F	74	6F	6B	65	6E	3D	65	79	4A	68	62	47	63	69	4F	69
000000C0	4A	49	55	7A	55	78	4D	69	49	73	49	6E	52	35	63	43	49	36	49	6B	70	58	56	43	49	73	49	6D	74	70	5A	43
000000E0	49	36	49	6D	4E	6B	62	69	4A	39	2E	65	79	4A	79	5A	58	4E	76	64	58	4A	6A	5A	53	49	36	49	69	38	78	5A
00000100	57	49	78	5A	54	4E	68	4D	69	30	77	5A	44	64	6C	4C	54	51	30	59	57	51	74	4F	44	67	35	4D	69	30	79	5A
00000120	47	45	35	4F	44	51	35	59	7A	52	6A	4D	54	51	76	63	32	31	68	62	47	78	7A	62	57	46	73	62	48	4E	74	59
00000140	57	78	73	49	69	77	69	64	47	56	75	59	57	35	30	49	6A	6F	69	59	32	46	75	64	6D	46	7A	49	69	77	69	64
00000160	58	4E	6C	63	6C	39	70	5A	43	49	36	49	6A	45	78	4D	6A	6B	35	4D	44	41	77	4D	44	41	77	4D	44	41	78	4D
00000180	7A	55	34	4D	79	49	73	49	6D	6C	68	64	43	49	36	4D	54	59	77	4E	6A	45	30	4D	44	59	78	4F	43	77	69	5A

Figure 26. Correct header for PNG.

8. **Take a screenshot to show the recovered file.**

Figure 27. recovered.png