CS-342 Operating Systems

Project 4

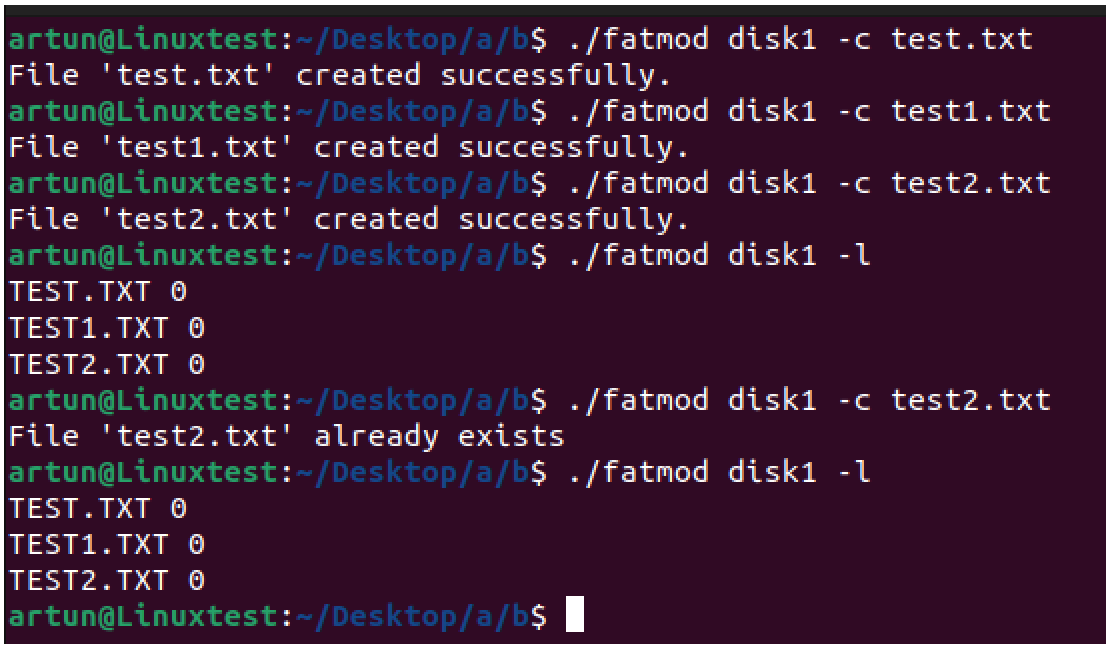
Section 2

Atilla Alp Yavuz 22102191

Tolga Artun Koçak 22102132

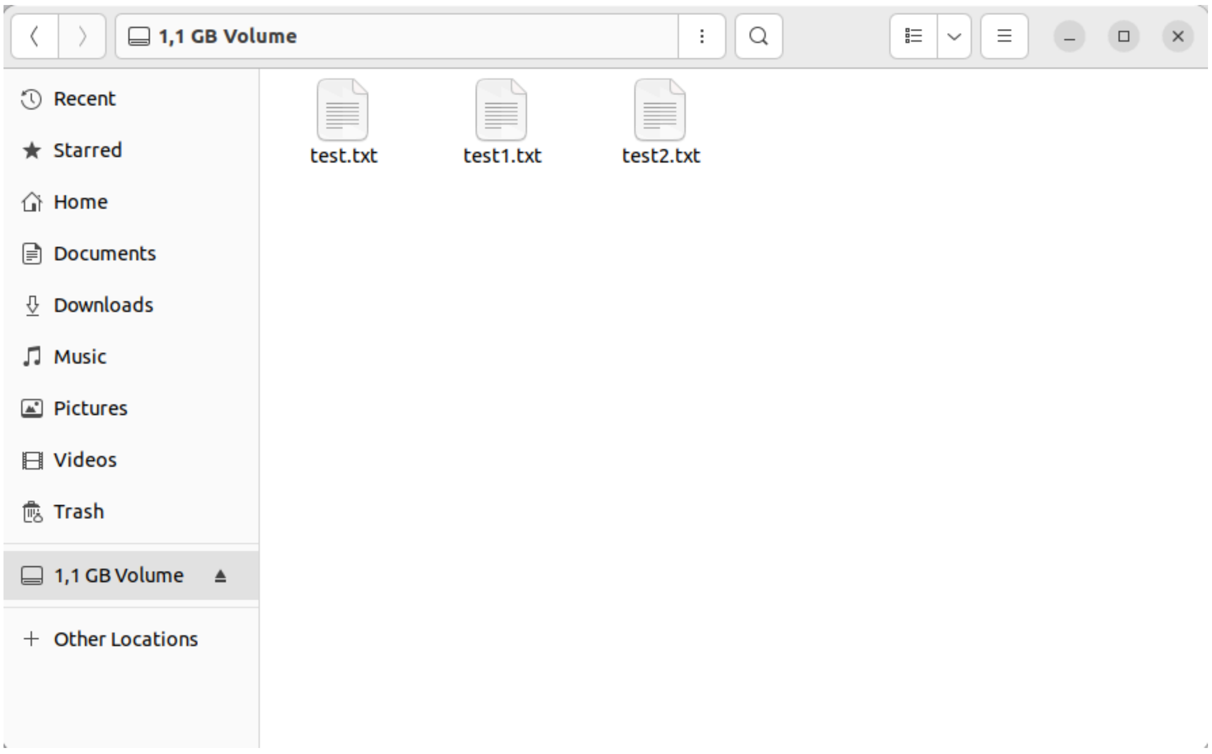
This Project accomplishes every task expected of it during its runtime. These tasks will be lined out in the following tests.

File creation test:

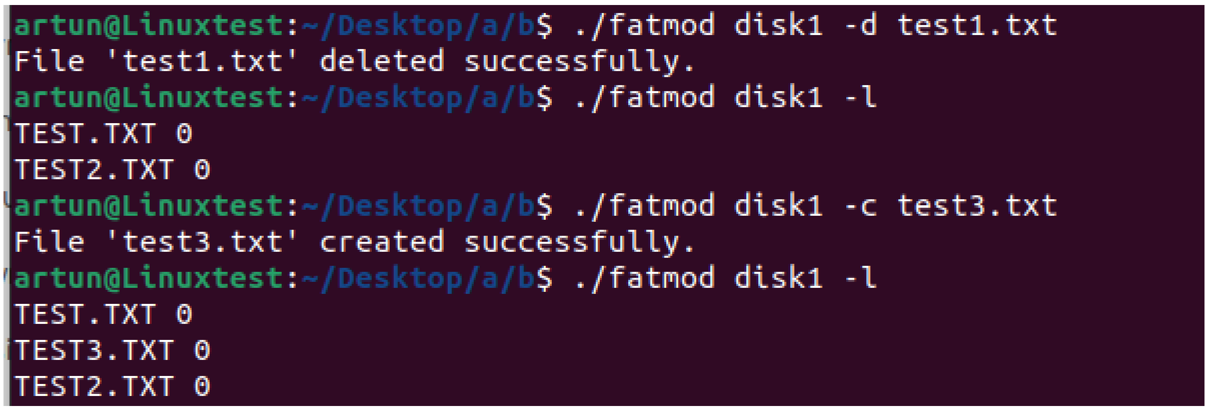


```
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -c test.txt
File 'test.txt' created successfully.
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -c test1.txt
File 'test1.txt' created successfully.
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -c test2.txt
File 'test2.txt' created successfully.
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -l
TEST.TXT 0
TEST1.TXT 0
TEST2.TXT 0
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -c test2.txt
File 'test2.txt' already exists
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -l
TEST.TXT 0
TEST1.TXT 0
TEST2.TXT 0
artun@Linuxtest:~/Desktop/a/b$
```

This test shows that all actions of folder creation can take place using the fatmod library. After unmounting and remounting the partition folder looks like so:

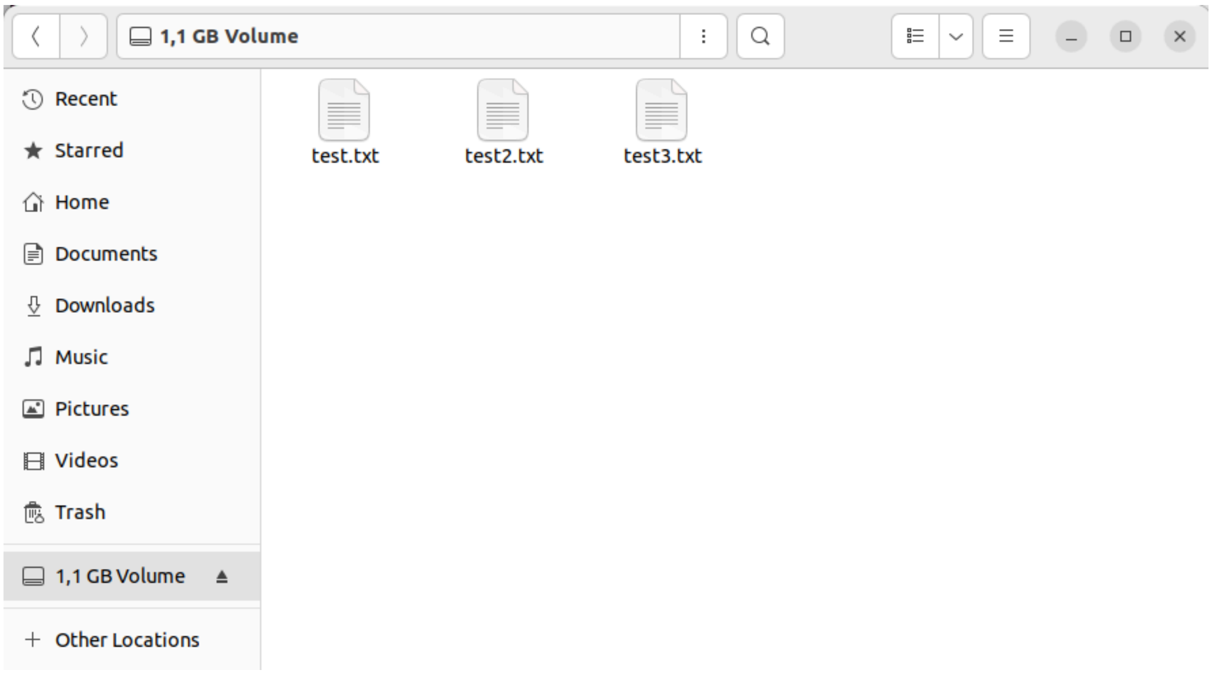


File deletion test:



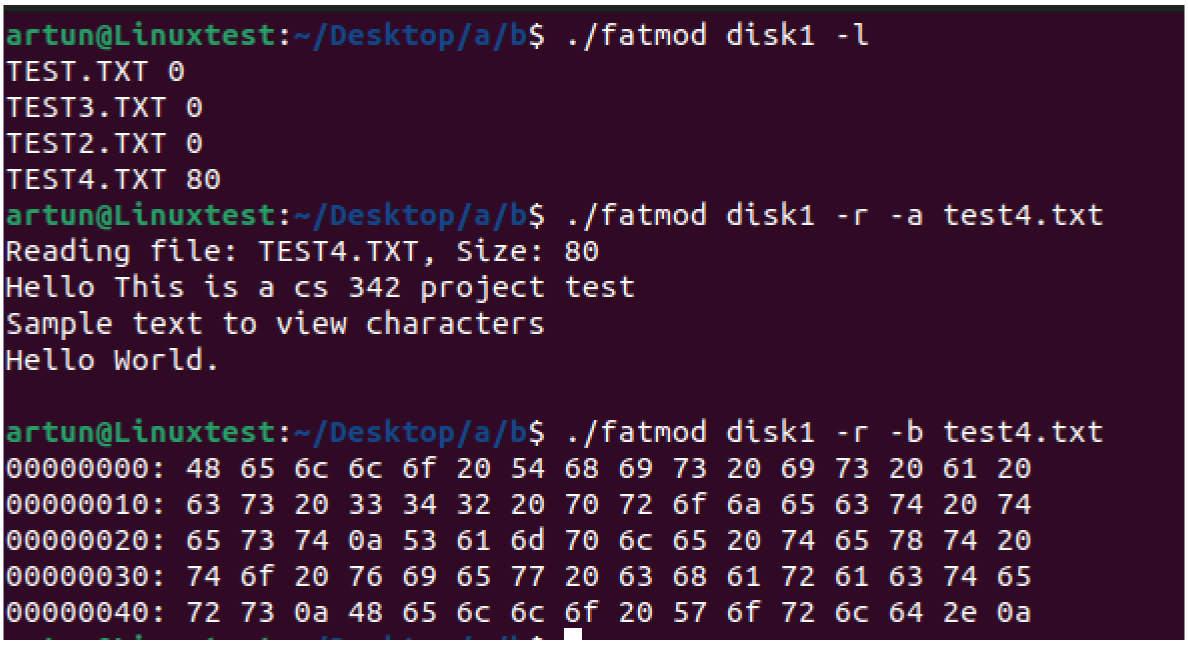
```
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -d test1.txt
File 'test1.txt' deleted successfully.
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -l
TEST.TXT 0
TEST2.TXT 0
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -c test3.txt
File 'test3.txt' created successfully.
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -l
TEST.TXT 0
TEST3.TXT 0
TEST2.TXT 0
```

This test shows that the deletion of folders can occur as well as rewriting ontop of the same root addresses. You can observe that test3 is placed in the place of the previous test1.
The folder structure looks like so after this test:

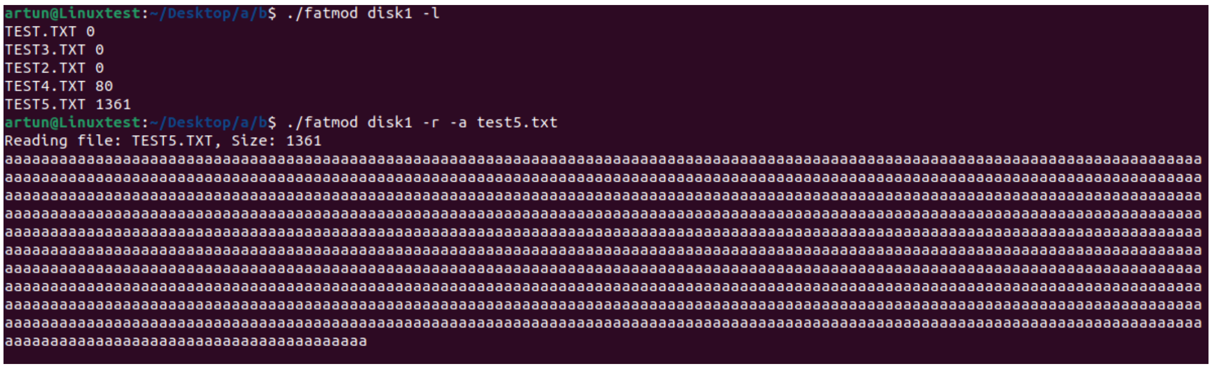


As you can see test1 has been deleted while test3 has also been created.

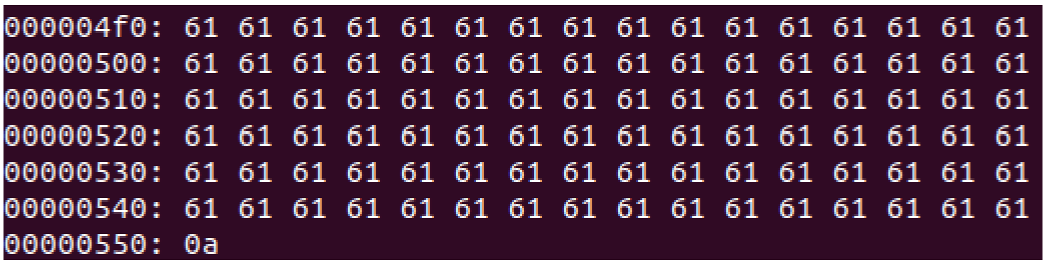
File reading tests:
For these tests, .txt folders created through emacs will be used
the new folder test4.txt can be read in binary and in ascii


```
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -l
TEST.TXT 0
TEST3.TXT 0
TEST2.TXT 0
TEST4.TXT 80
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -r -a test4.txt
Reading file: TEST4.TXT, Size: 80
Hello This is a cs 342 project test
Sample text to view characters
Hello World.

artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -r -b test4.txt
00000000: 48 65 6c 6c 6f 20 54 68 69 73 20 69 73 20 61 20
00000010: 63 73 20 33 34 32 20 70 72 6f 6a 65 63 74 20 74
00000020: 65 73 74 0a 53 61 6d 70 6c 65 20 74 65 78 74 20
00000030: 74 6f 20 76 69 65 77 20 63 68 61 72 61 63 74 65
00000040: 72 73 0a 48 65 6c 6c 6f 20 57 6f 72 6c 64 2e 0a
```

the test5.txt folder is just 1360 instances of the character a. This is to demonstrate how printing can be accomplished when there is more than one cluster allocated to a single file.


```
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -l
TEST.TXT 0
TEST3.TXT 0
TEST2.TXT 0
TEST4.TXT 80
TEST5.TXT 1361
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -r -a test5.txt
Reading file: TEST5.TXT, Size: 1361
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

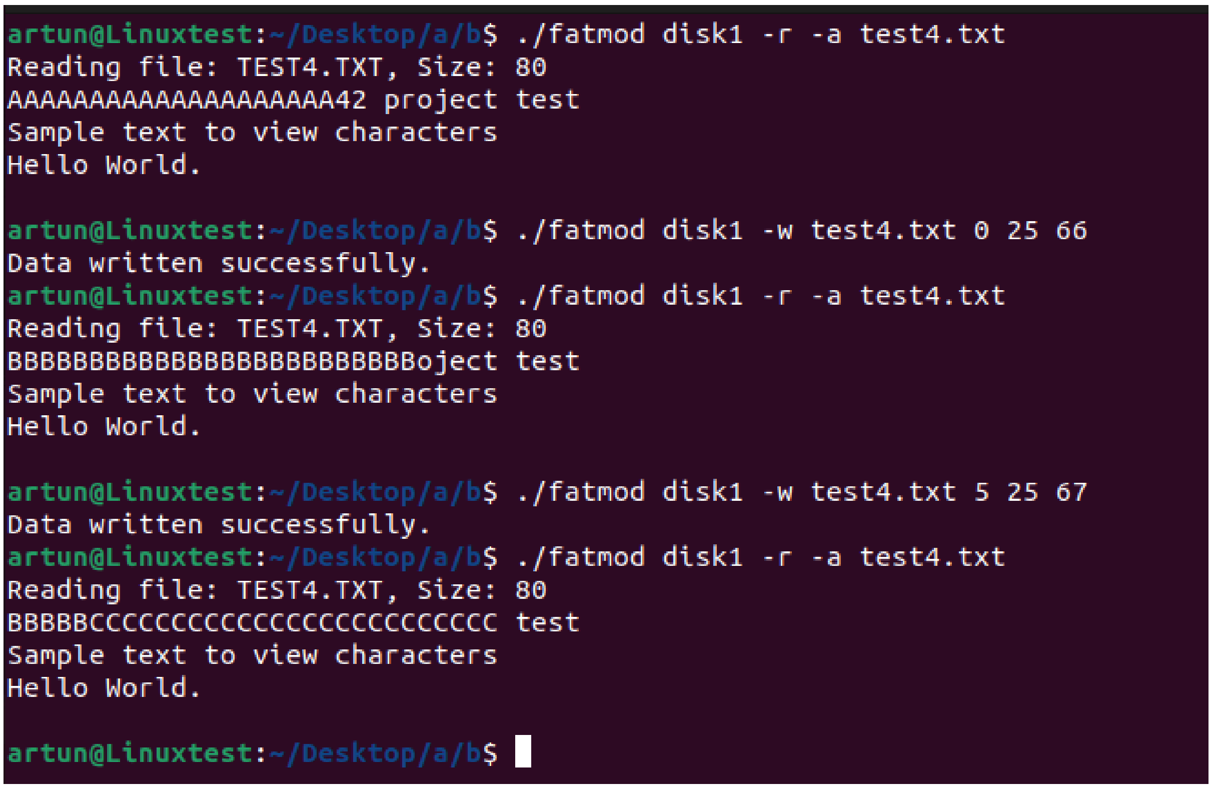
Copying and pasting the output to a notepad app and checking the column position of the last character does indeed prove that the read operation can span multiple clusters. The output of the binary print is too large to paste onto the report so I will paste only the last few lines but be assured that the output is completely the repetition of the hexadecimal value (61) of the lowercase a character.


```
000004f0: 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
00000500: 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
00000510: 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
00000520: 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
00000530: 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
00000540: 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
00000550: 0a
```

you can observe that 550 characters of 61 are repeated and 550 in hexadecimal is equal to 1360 exactly. The newline character is also a part of the initial text file.

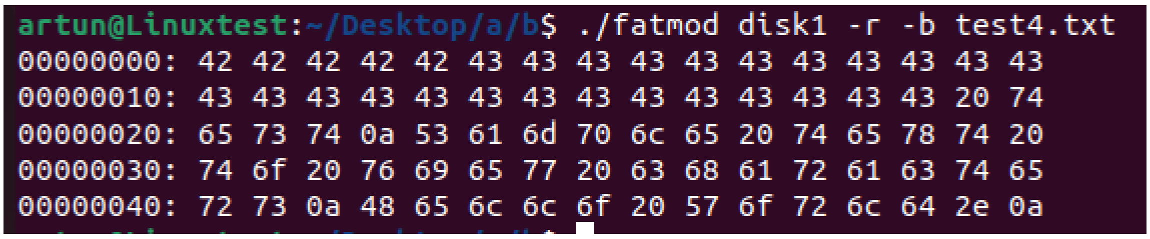
Writing tests:

We can use the file test4 to showcase the writing functionality of the program.



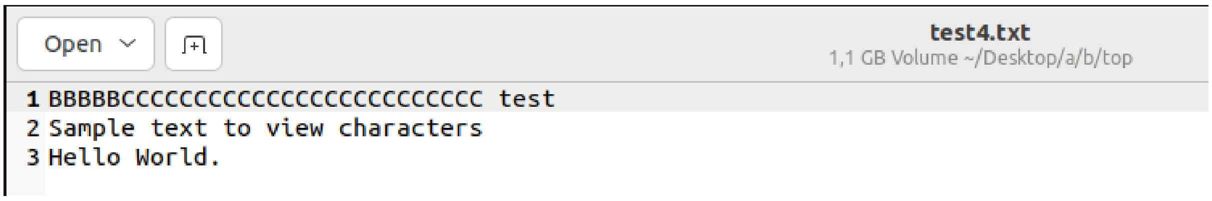
```
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -r -a test4.txt
Reading file: TEST4.TXT, Size: 80
AAAAAAAAAAAAAAAAAAAA42 project test
Sample text to view characters
Hello World.

artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -w test4.txt 0 25 66
Data written successfully.
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -r -a test4.txt
Reading file: TEST4.TXT, Size: 80
BBBBBBBBBBBBBBBBBBBBBBBBBoject test
Sample text to view characters
Hello World.

artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -w test4.txt 5 25 67
Data written successfully.
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -r -a test4.txt
Reading file: TEST4.TXT, Size: 80
BBBBBCCCCCCCCCCCCCCCCCCCCCCCCC test
Sample text to view characters
Hello World.

artun@Linuxtest:~/Desktop/a/b$
```

Firstly, as you can observe, we add 25 instances of the character B over the previously displayed test4.txt and we can see the change it creates. After which we add another 25 instances of the character C, this time with an offset of 5. This results in 5 unchanged B characters in the start. When displaying this file in binary it looks like so:



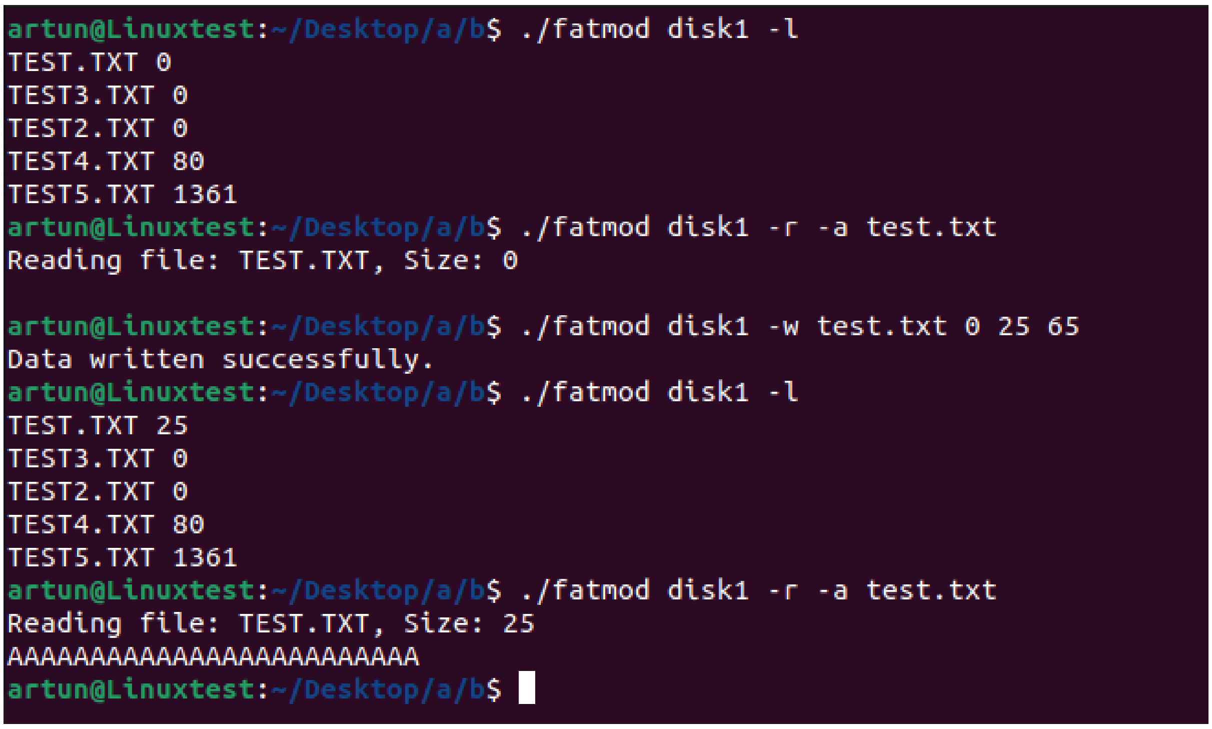
```
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -r -b test4.txt
00000000: 42 42 42 42 42 43 43 43 43 43 43 43 43 43 43 43
00000010: 43 43 43 43 43 43 43 43 43 43 43 43 43 43 20 74
00000020: 65 73 74 0a 53 61 6d 70 6c 65 20 74 65 78 74 20
00000030: 74 6f 20 76 69 65 77 20 63 68 61 72 61 63 74 65
00000040: 72 73 0a 48 65 6c 6c 6f 20 57 6f 72 6c 64 2e 0a
```

you can observe the changes in the binary file as well. Similarly, you can see that the txt folder actually changes by clicking on it independently.



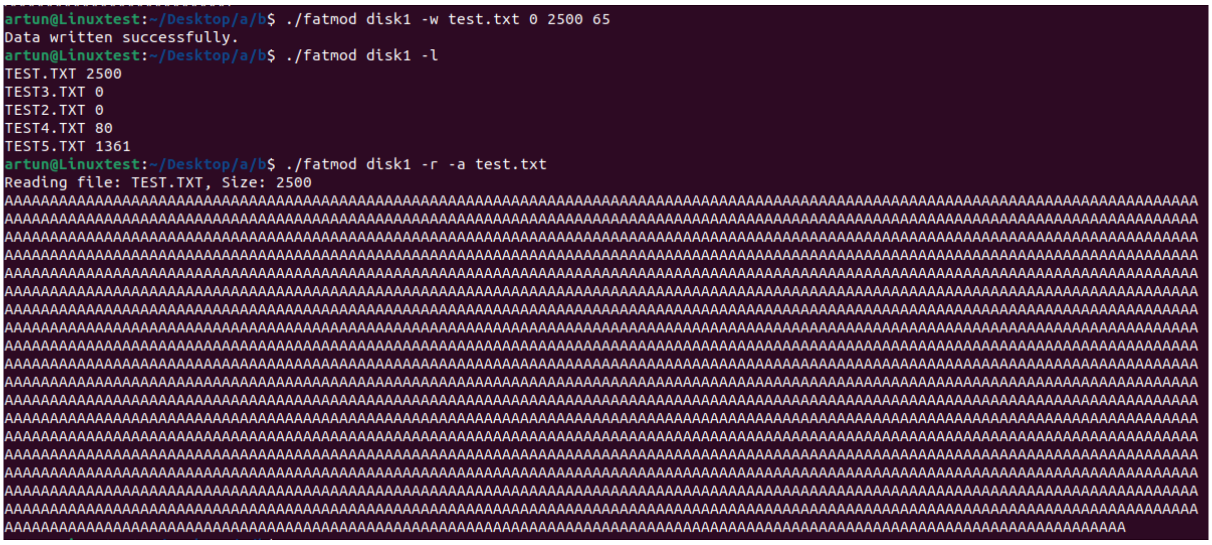
```
1 BBBBBCCCCCCCCCCCCCCCCCCCCCCCCC test
2 Sample text to view characters
3 Hello World.
```

The write function can also alter the size of a file. we will use the previously created file test.txt to demonstrate this:



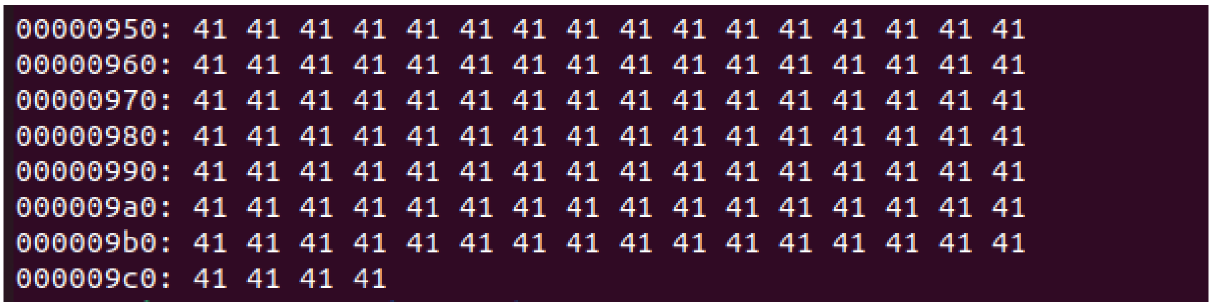
```
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -l
TEST.TXT 0
TEST3.TXT 0
TEST2.TXT 0
TEST4.TXT 80
TEST5.TXT 1361
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -r -a test.txt
Reading file: TEST.TXT, Size: 0

artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -w test.txt 0 25 65
Data written successfully.
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -l
TEST.TXT 25
TEST3.TXT 0
TEST2.TXT 0
TEST4.TXT 80
TEST5.TXT 1361
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -r -a test.txt
Reading file: TEST.TXT, Size: 25
AAAAAAAAAAAAAAAAAAAAAAAAA
artun@Linuxtest:~/Desktop/a/b$
```

A folder of size 0 can increase in size and it will be reflected on the rest of the functions. a txt folder can also have multiple clusters assigned to it if the write function requires it.



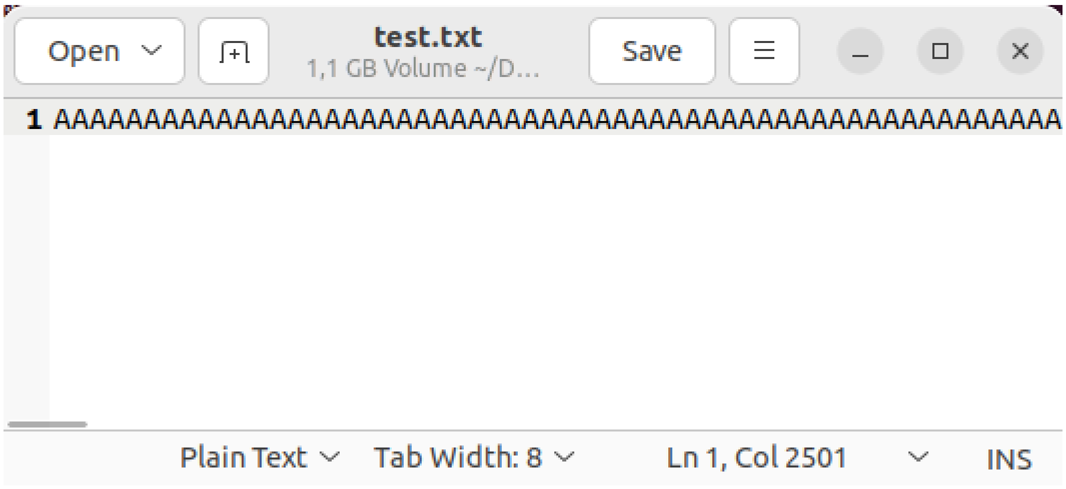
```
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -w test.txt 0 2500 65
Data written successfully.
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -l
TEST.TXT 2500
TEST3.TXT 0
TEST2.TXT 0
TEST4.TXT 80
TEST5.TXT 1361
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -r -a test.txt
Reading file: TEST.TXT, Size: 2500
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

The binary read function will display how many characters there are.



```
00000950: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
00000960: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
00000970: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
00000980: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
00000990: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
000009a0: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
000009b0: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
000009c0: 41 41 41 41
```

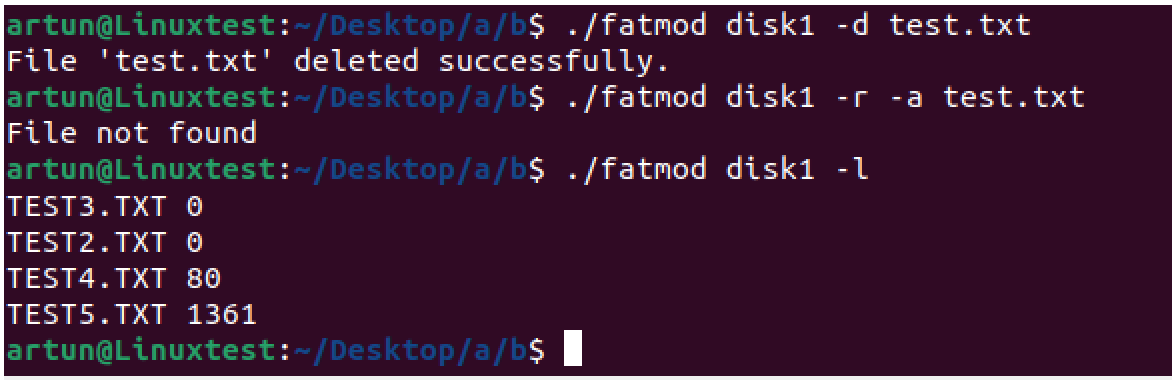
9c0 in hexadecimal is equal to 2496 in decimal with the 4 additional characters, there are exactly 2500 A characters in the file.



```
1 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

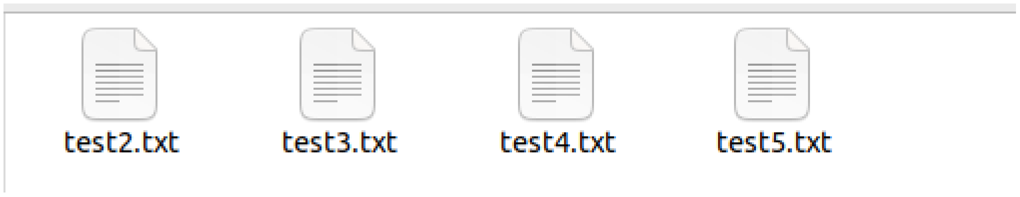
The txt folder when opened also shows that the last character is on the 2500th index and it is all uppercase A's.

We can also delete the test file to demonstrate that deletion can also span multiple clusters.



```
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -d test.txt
File 'test.txt' deleted successfully.
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -r -a test.txt
File not found
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -l
TEST3.TXT 0
TEST2.TXT 0
TEST4.TXT 80
TEST5.TXT 1361
artun@Linuxtest:~/Desktop/a/b$
```

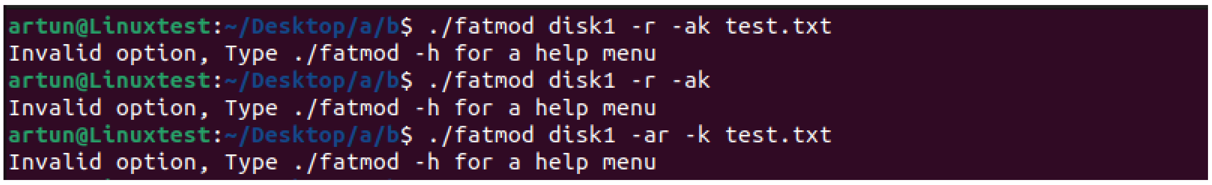
The test file is no longer accessible. Demonstrating that deletion across chunks doesn't cause issues.



This is the last structure of the partition.

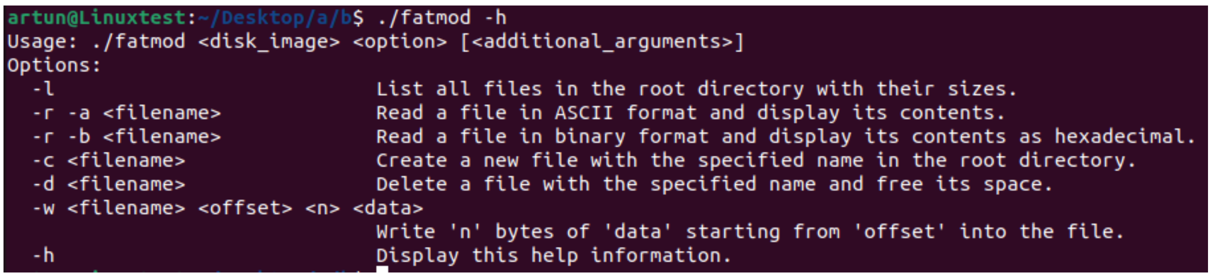
Other tests:

when an invalid command is entered, the system prompts this response



```
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -r -ak test.txt
Invalid option, Type ./fatmod -h for a help menu
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -r -ak
Invalid option, Type ./fatmod -h for a help menu
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -ar -k test.txt
Invalid option, Type ./fatmod -h for a help menu
```

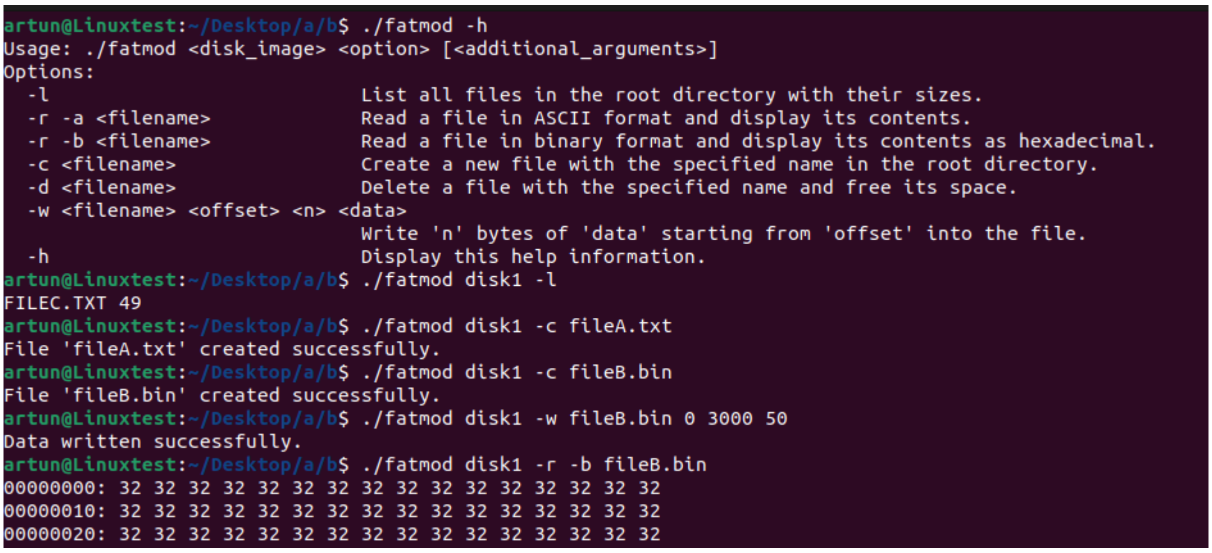
the output for ./fatmod -h is like so:



```
artun@Linuxtest:~/Desktop/a/b$ ./fatmod -h
Usage: ./fatmod <disk_image> <option> [<additional_arguments>]
Options:
  -l                           List all files in the root directory with their sizes.
  -r -a <filename>             Read a file in ASCII format and display its contents.
  -r -b <filename>             Read a file in binary format and display its contents as hexadecimal.
  -c <filename>                Create a new file with the specified name in the root directory.
  -d <filename>                Delete a file with the specified name and free its space.
  -w <filename> <offset> <n> <data>
                               Write 'n' bytes of 'data' starting from 'offset' into the file.
  -h                           Display this help information.
```

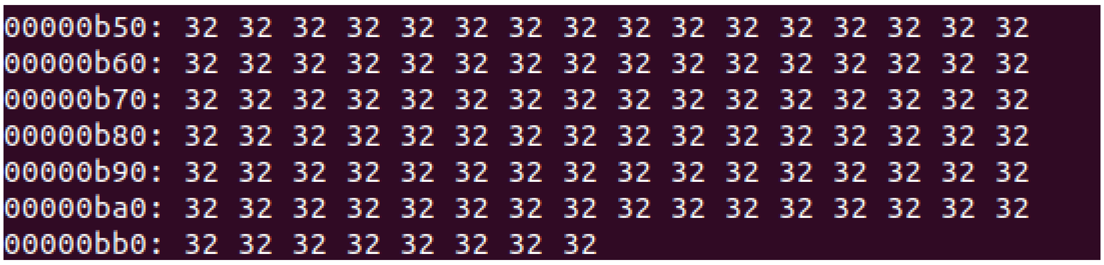
The Complete test:

Running the entirety of the functions in the clarifications page looks like so:



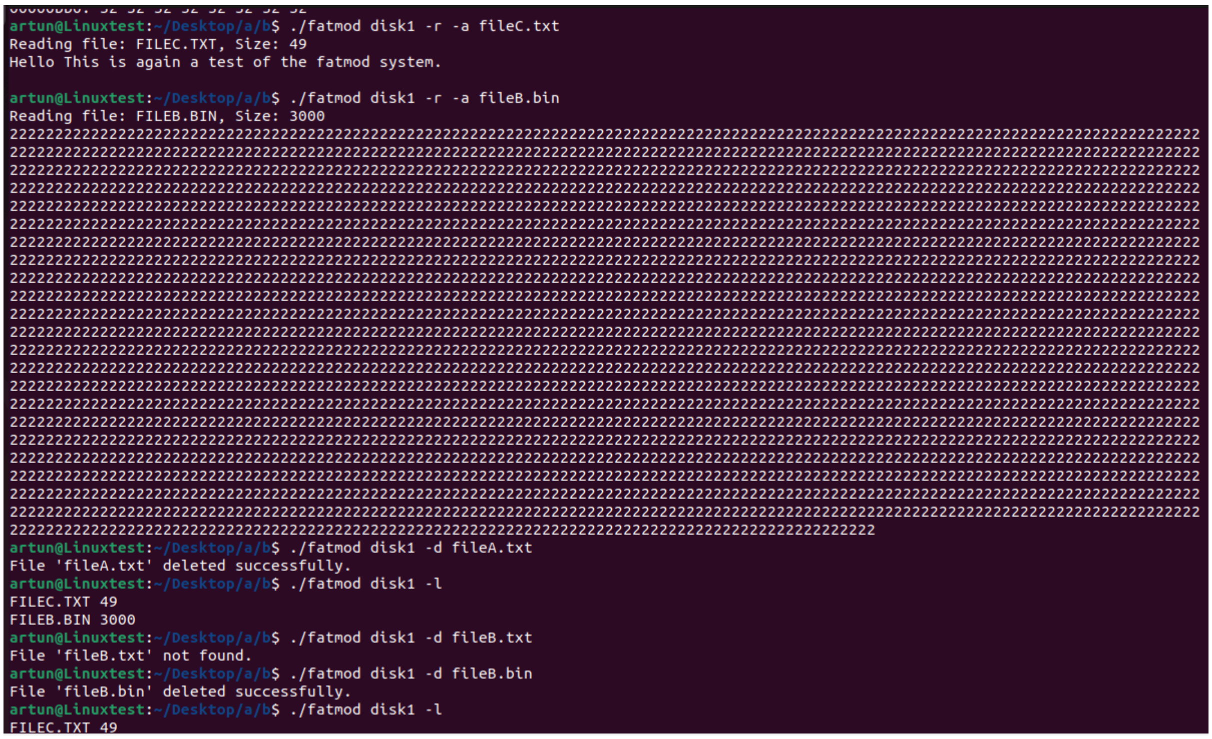
```
artun@Linuxtest:~/Desktop/a/b$ ./fatmod -h
Usage: ./fatmod <disk_image> <option> [<additional_arguments>]
Options:
  -l                              List all files in the root directory with their sizes.
  -r -a <filename>                Read a file in ASCII format and display its contents.
  -r -b <filename>                Read a file in binary format and display its contents as hexadecimal.
  -c <filename>                   Create a new file with the specified name in the root directory.
  -d <filename>                   Delete a file with the specified name and free its space.
  -w <filename> <offset> <n> <data>
                                  Write 'n' bytes of 'data' starting from 'offset' into the file.
  -h                              Display this help information.
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -l
FILEC.TXT 49
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -c fileA.txt
File 'fileA.txt' created successfully.
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -c fileB.bin
File 'fileB.bin' created successfully.
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -w fileB.bin 0 3000 50
Data written successfully.
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -r -b fileB.bin
00000000: 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32
00000010: 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32
00000020: 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32
```

a few -l functions have been added to demonstrate the changes in the filesystem. The following are the last few lines of the -r -b output of fileB.bin



```
00000b50: 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32
00000b60: 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32
00000b70: 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32
00000b80: 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32
00000b90: 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32
00000ba0: 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32
00000bb0: 32 32 32 32 32 32 32 32
```

These are the rest of the functions:



```
00000bb0: 32 32 32 32 32 32 32 32
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -r -a fileC.txt
Reading file: FILEC.TXT, Size: 49
Hello This is again a test of the fatmod system.

artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -r -a fileB.bin
Reading file: FILEB.BIN, Size: 3000
222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -d fileA.txt
File 'fileA.txt' deleted successfully.
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -l
FILEC.TXT 49
FILEB.BIN 3000
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -d fileB.txt
File 'fileB.txt' not found.
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -d fileB.bin
File 'fileB.bin' deleted successfully.
artun@Linuxtest:~/Desktop/a/b$ ./fatmod disk1 -l
FILEC.TXT 49
```