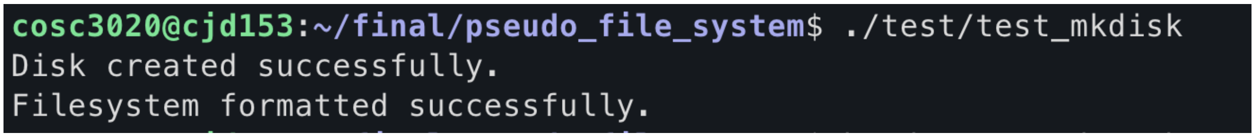
Plan to Run Tests:

test_mkdisk:

Screenshot:

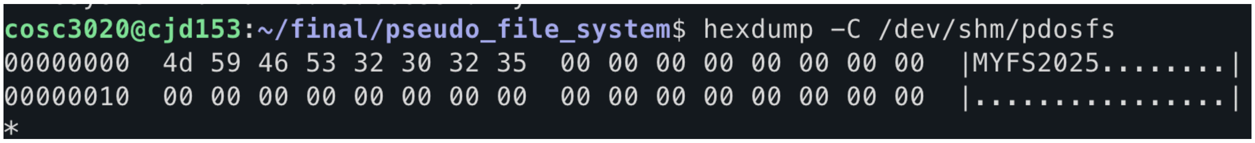


```
cosc3020@cjd153:~/final/pseudo_file_system$ ./test/test_mkdisk
Disk created successfully.
Filesystem formatted successfully.
```

What is being tested, and how:

Create a new disk and file system through pdos_mkdisk() and pdos_mkfs(), printing out messages to see whether they are successful or not.

Hexdump screenshot:

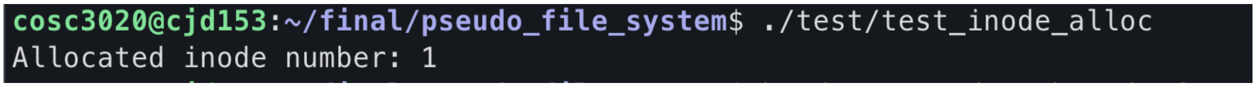


```
cosc3020@cjd153:~/final/pseudo_file_system$ hexdump -C /dev/shm/pdosfs
00000000  4d 59 46 53 32 30 32 35  00 00 00 00 00 00 00 00  |MYFS2025........|
00000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
```

How we know that means it’s working: There now exist rudimentary blocks in the new file system.

test_inode_alloc:

Screenshot:

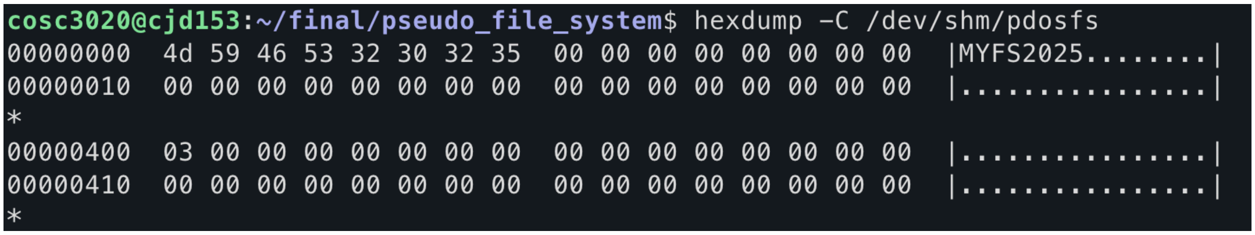


```
cosc3020@cjd153:~/final/pseudo_file_system$ ./test/test_inode_alloc
Allocated inode number: 1
```

What is being tested, and how:

We want to show that inode allocation works; if successful, we output the inode number allocated.

Hexdump screenshot:

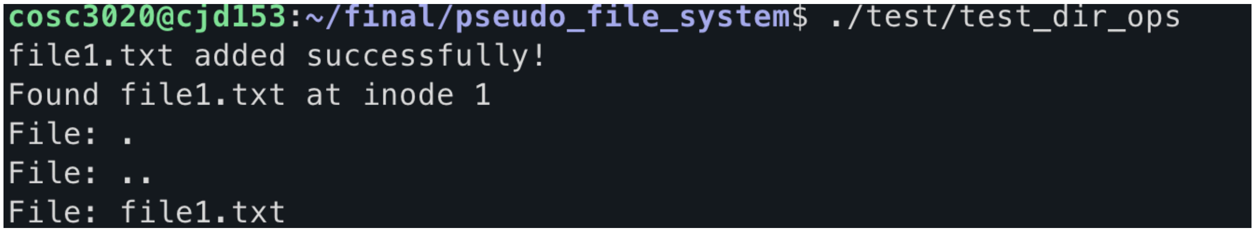


```
cosc3020@cjd153:~/final/pseudo_file_system$ hexdump -C /dev/shm/pdosfs
00000000  4d 59 46 53 32 30 32 35  00 00 00 00 00 00 00 00  |MYFS2025........|
00000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000400  03 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000410  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
```

How we know that means it’s working: The inode bitmap now shows that something has been filled in, corresponding to this inode being ‘taken’ and it shows now that inode ant 000000400 changed from 01 to 03 indicating 0 and 1 inodes are allocated.

test_dir_ops:

Screenshot:



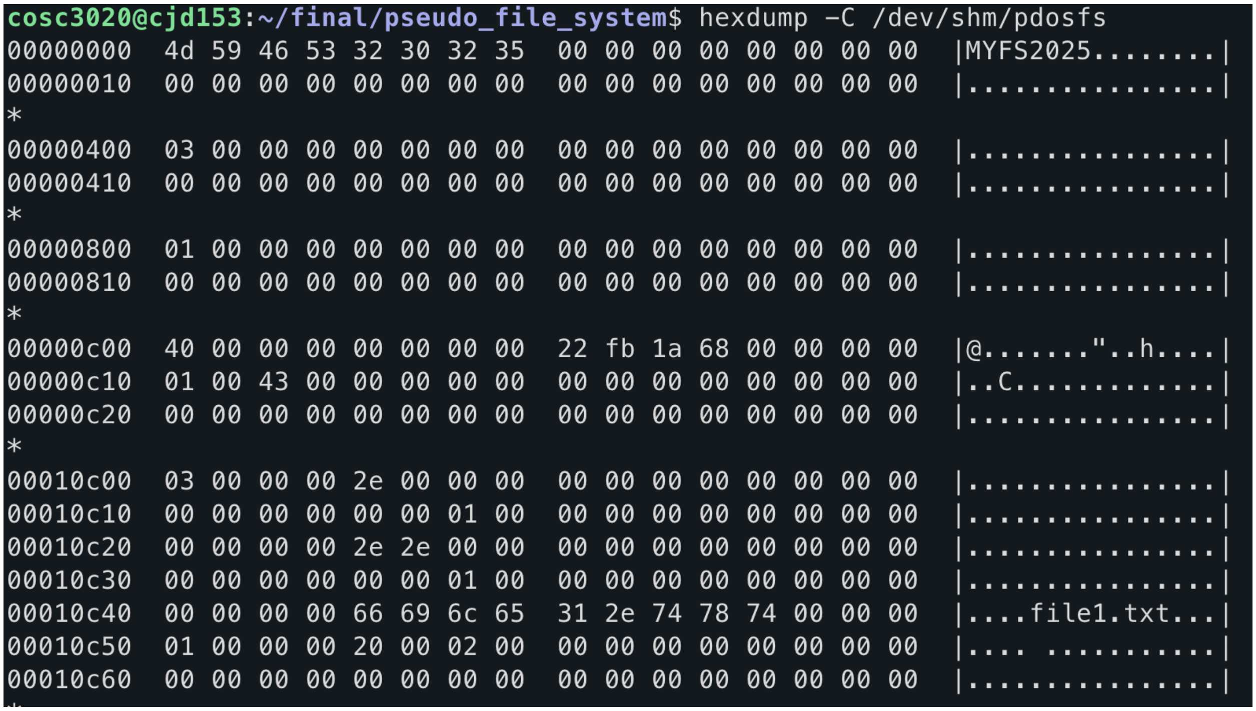
```
cosc3020@cjd153:~/final/pseudo_file_system$ ./test/test_dir_ops
file1.txt added successfully!
Found file1.txt at inode 1
File: .
File: ..
File: file1.txt
```

What is being tested, and how:

We are testing directory functionality using the root directory.

Create a new disk and file system. A file is added into the root directory, and is then searched for. Messages are printed out at each step. The test concludes by printing out the entries in the root directory, which should include the added file.

Hexdump screenshot:

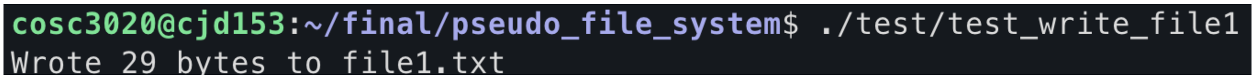


```
cosc3020@cjd153:~/final/pseudo_file_system$ hexdump -C /dev/shm/pdosfs
00000000  4d 59 46 53 32 30 32 35  00 00 00 00 00 00 00 00  |MYFS2025........|
00000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000400  03 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000410  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000800  01 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000810  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000c00  40 00 00 00 00 00 00 00  22 fb 1a 68 00 00 00 00  |@......."..h....|
00000c10  01 00 43 00 00 00 00 00  00 00 00 00 00 00 00 00  |..C.............|
00000c20  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00010c00  03 00 00 00 2e 00 00 00  00 00 00 00 00 00 00 00  |................|
00010c10  00 00 00 00 00 00 01 00  00 00 00 00 00 00 00 00  |................|
00010c20  00 00 00 00 2e 2e 00 00  00 00 00 00 00 00 00 00  |................|
00010c30  00 00 00 00 00 00 01 00  00 00 00 00 00 00 00 00  |................|
00010c40  00 00 00 00 66 69 6c 65  31 2e 74 78 74 00 00 00  |....file1.txt...|
00010c50  01 00 00 00 20 00 02 00  00 00 00 00 00 00 00 00  |.... ...........|
00010c60  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
```

How we know that means it’s working: The modified information in the first inode block corresponds to the file being added; the file itself contains nothing.

test_write_file1:

Screenshot:

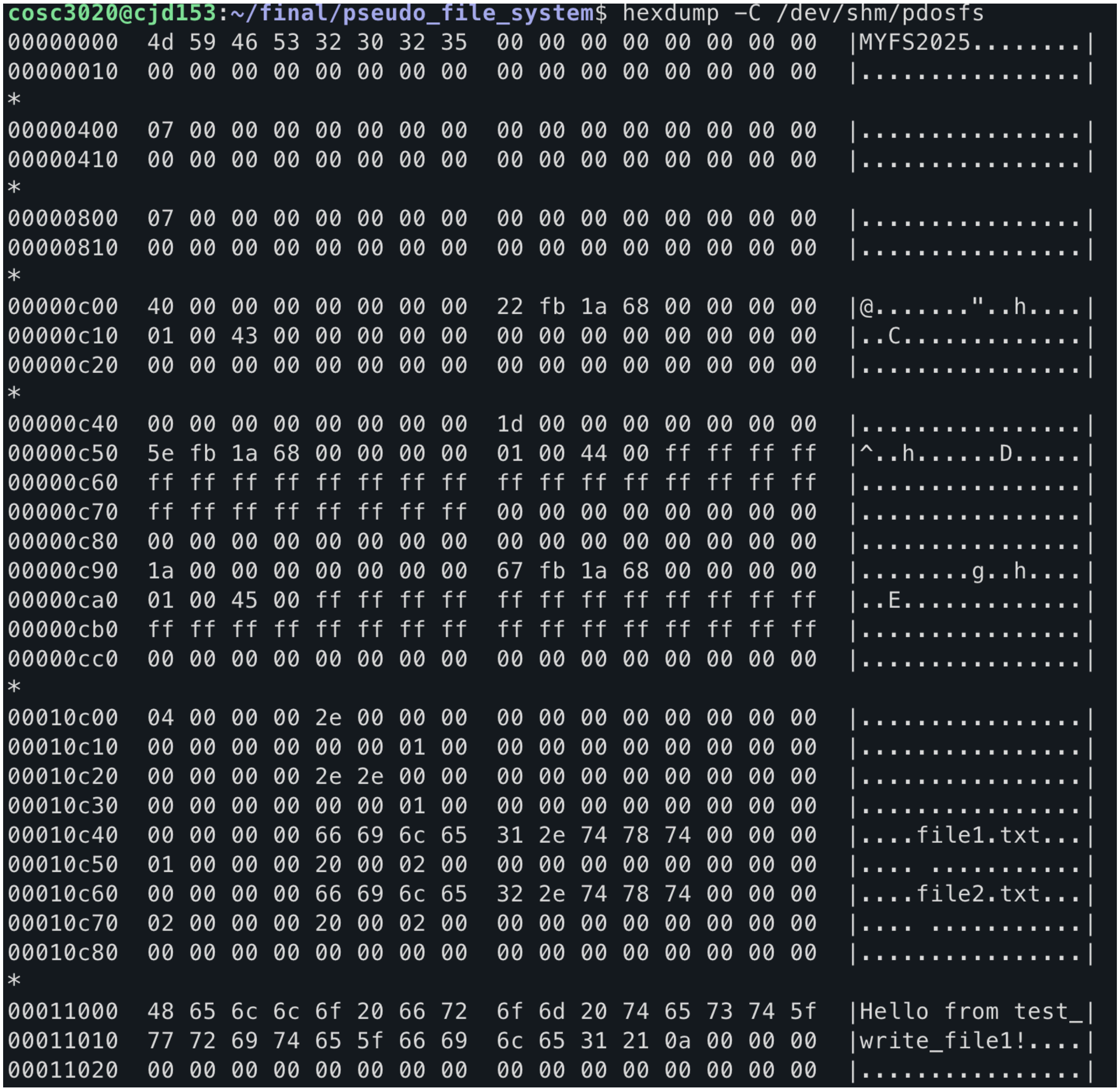


```
cosc3020@cjd153:~/final/pseudo_file_system$ ./test/test_write_file1
Wrote 29 bytes to file1.txt
```

What is being tested, and how:

We want to show that we can write to a file. We make the disk, create a file pointer, and then write in a message.

Hexdump screenshot:

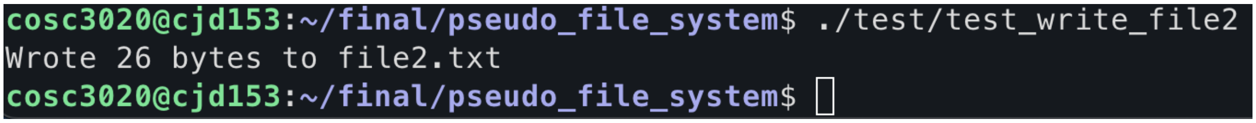


```
cosc3020@cjd153:~/final/pseudo_file_system$ hexdump -C /dev/shm/pdosfs
00000000  4d 59 46 53 32 30 32 35  00 00 00 00 00 00 00 00  |MYFS2025........|
00000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000400  07 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000410  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000800  07 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000810  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000c00  40 00 00 00 00 00 00 00  22 fb 1a 68 00 00 00 00  |@......."..h....|
00000c10  01 00 43 00 00 00 00 00  00 00 00 00 00 00 00 00  |..C.............|
00000c20  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000c40  00 00 00 00 00 00 00 00  1d 00 00 00 00 00 00 00  |................|
00000c50  5e fb 1a 68 00 00 00 00  01 00 44 00 ff ff ff ff  |^..h......D.....|
00000c60  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff  |................|
00000c70  ff ff ff ff ff ff ff ff  00 00 00 00 00 00 00 00  |................|
00000c80  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000c90  1a 00 00 00 00 00 00 00  67 fb 1a 68 00 00 00 00  |........g..h....|
00000ca0  01 00 45 00 ff ff ff ff  ff ff ff ff ff ff ff ff  |..E.............|
00000cb0  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff  |................|
00000cc0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00010c00  04 00 00 00 2e 00 00 00  00 00 00 00 00 00 00 00  |................|
00010c10  00 00 00 00 00 00 01 00  00 00 00 00 00 00 00 00  |................|
00010c20  00 00 00 00 2e 2e 00 00  00 00 00 00 00 00 00 00  |................|
00010c30  00 00 00 00 00 00 01 00  00 00 00 00 00 00 00 00  |................|
00010c40  00 00 00 00 66 69 6c 65  31 2e 74 78 74 00 00 00  |....file1.txt...|
00010c50  01 00 00 00 20 00 02 00  00 00 00 00 00 00 00 00  |.... ...........|
00010c60  00 00 00 00 66 69 6c 65  32 2e 74 78 74 00 00 00  |....file2.txt...|
00010c70  02 00 00 00 20 00 02 00  00 00 00 00 00 00 00 00  |.... ...........|
00010c80  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00011000  48 65 6c 6c 6f 20 66 72  6f 6d 20 74 65 73 74 5f  |Hello from test_|
00011010  77 72 69 74 65 5f 66 69  6c 65 31 21 0a 00 00 00  |write_file1!....|
00011020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
```

How we know that means it's working: We see that now there is some information in the data blocks.

test_write_file2:

Screenshot:

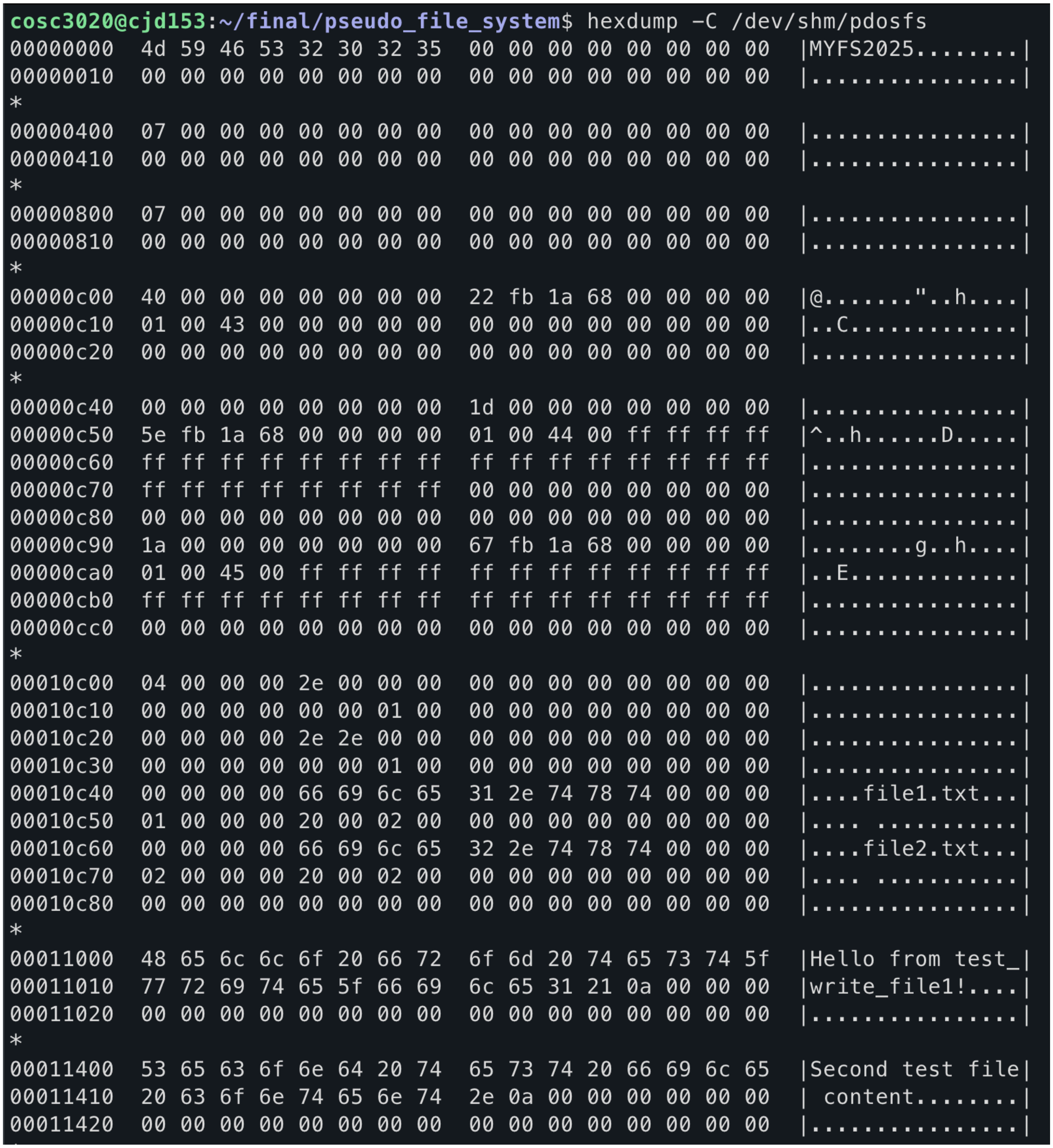


```
cosc3020@cjd153:~/final/pseudo_file_system$ ./test/test_write_file2
Wrote 26 bytes to file2.txt
cosc3020@cjd153:~/final/pseudo_file_system$ 
```

What is being tested, and how:

We want to show that we can write to a second file without compromising the contents of the first. We make the disk, create a file pointer, and then write in a message.

Hexdump screenshot:

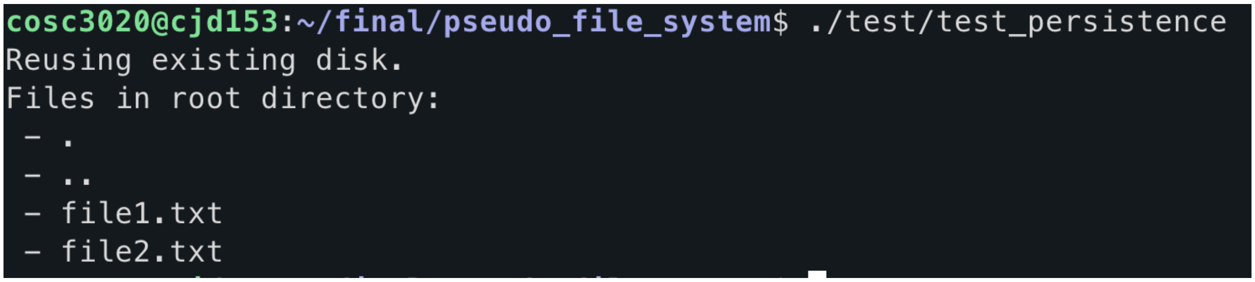


```
cosc3020@cjd153:~/final/pseudo_file_system$ hexdump -C /dev/shm/pdosfs
00000000  4d 59 46 53 32 30 32 35  00 00 00 00 00 00 00 00  |MYFS2025........|
00000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000400  07 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000410  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000800  07 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000810  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000c00  40 00 00 00 00 00 00 00  22 fb 1a 68 00 00 00 00  |@......."..h....|
00000c10  01 00 43 00 00 00 00 00  00 00 00 00 00 00 00 00  |..C.............|
00000c20  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000c40  00 00 00 00 00 00 00 00  1d 00 00 00 00 00 00 00  |................|
00000c50  5e fb 1a 68 00 00 00 00  01 00 44 00 ff ff ff ff  |^..h......D.....|
00000c60  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff  |................|
00000c70  ff ff ff ff ff ff ff ff  00 00 00 00 00 00 00 00  |................|
00000c80  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000c90  1a 00 00 00 00 00 00 00  67 fb 1a 68 00 00 00 00  |........g..h....|
00000ca0  01 00 45 00 ff ff ff ff  ff ff ff ff ff ff ff ff  |..E.............|
00000cb0  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff  |................|
00000cc0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00010c00  04 00 00 00 2e 00 00 00  00 00 00 00 00 00 00 00  |................|
00010c10  00 00 00 00 00 00 01 00  00 00 00 00 00 00 00 00  |................|
00010c20  00 00 00 00 2e 2e 00 00  00 00 00 00 00 00 00 00  |................|
00010c30  00 00 00 00 00 00 01 00  00 00 00 00 00 00 00 00  |................|
00010c40  00 00 00 00 66 69 6c 65  31 2e 74 78 74 00 00 00  |....file1.txt...|
00010c50  01 00 00 00 20 00 02 00  00 00 00 00 00 00 00 00  |.... ...........|
00010c60  00 00 00 00 66 69 6c 65  32 2e 74 78 74 00 00 00  |....file2.txt...|
00010c70  02 00 00 00 20 00 02 00  00 00 00 00 00 00 00 00  |.... ...........|
00010c80  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00011000  48 65 6c 6c 6f 20 66 72  6f 6d 20 74 65 73 74 5f  |Hello from test_|
00011010  77 72 69 74 65 5f 66 69  6c 65 31 21 0a 00 00 00  |write_file1!....|
00011020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00011400  53 65 63 6f 6e 64 20 74  65 73 74 20 66 69 6c 65  |Second test file|
00011410  20 63 6f 6e 74 65 6e 74  2e 0a 00 00 00 00 00 00  | content........|
00011420  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
```

How we know that means it's working: We see that now there is more information in the data blocks; nothing was erased (corresponding to file 1 contents.)

test_persistence:

Screenshot:



```
cosc3020@cjd153:~/final/pseudo_file_system$ ./test/test_persistence
Reusing existing disk.
Files in root directory:
 - .
 - ..
 - file1.txt
 - file2.txt
```

What is being tested, and how:

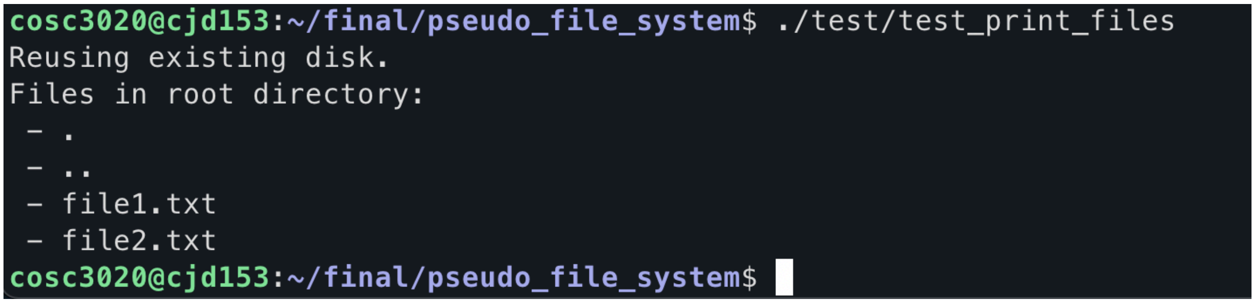
Show that the file system is persistent; that items remain in root directory

Make a disk and file system, print out whether it's fresh, and print out files in the root directory. I like to run this after operations to ensure that nothing has been corrupted in the file system.

How we know that means it's working: If it prints out the existing files that we had before read and write operations we know nothing was corrupted.

test_print_files:

Screenshot:



```
cosc3020@cjd153:~/final/pseudo_file_system$ ./test/test_print_files
Reusing existing disk.
Files in root directory:
 - .
 - ..
 - file1.txt
 - file2.txt
cosc3020@cjd153:~/final/pseudo_file_system$ 
```

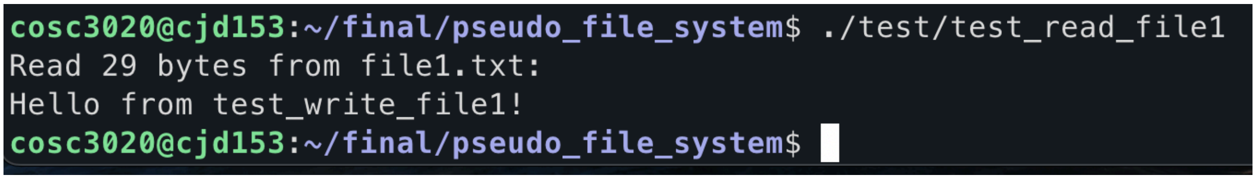
What is being tested, and how:

Create a new disk and file system through pdos_mkdisk() and pdos_mkfs(), printing out messages to see everything in the root directory to make sure directory entries are in there with no data being compromised in the process.

How we know that means it's working: No change in hexdump; we were only reading, not modifying.

test_read_file1:

Screenshot:



```
cosc3020@cjd153:~/final/pseudo_file_system$ ./test/test_read_file1
Read 29 bytes from file1.txt:
Hello from test_write_file1!
cosc3020@cjd153:~/final/pseudo_file_system$ 
```

What is being tested, and how:

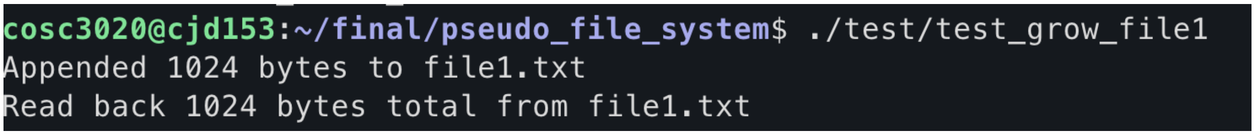
After writing to a file, we want to see that it can be read from and that its data is correctly organized into separate bytes.

We check that the file exists, open it, and read in each byte-character and output the number of bytes read.

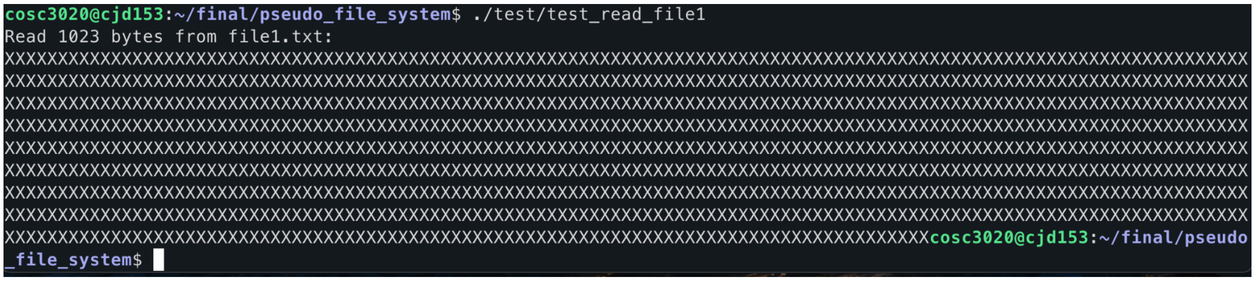
<u>How we know that means it’s working</u>: No change in hexdump; we were only reading, not modifying.

test_grow_file1:

Screenshot:



```
cosc3020@cjd153:~/final/pseudo_file_system$ ./test/test_grow_file1
Appended 1024 bytes to file1.txt
Read back 1024 bytes total from file1.txt
```



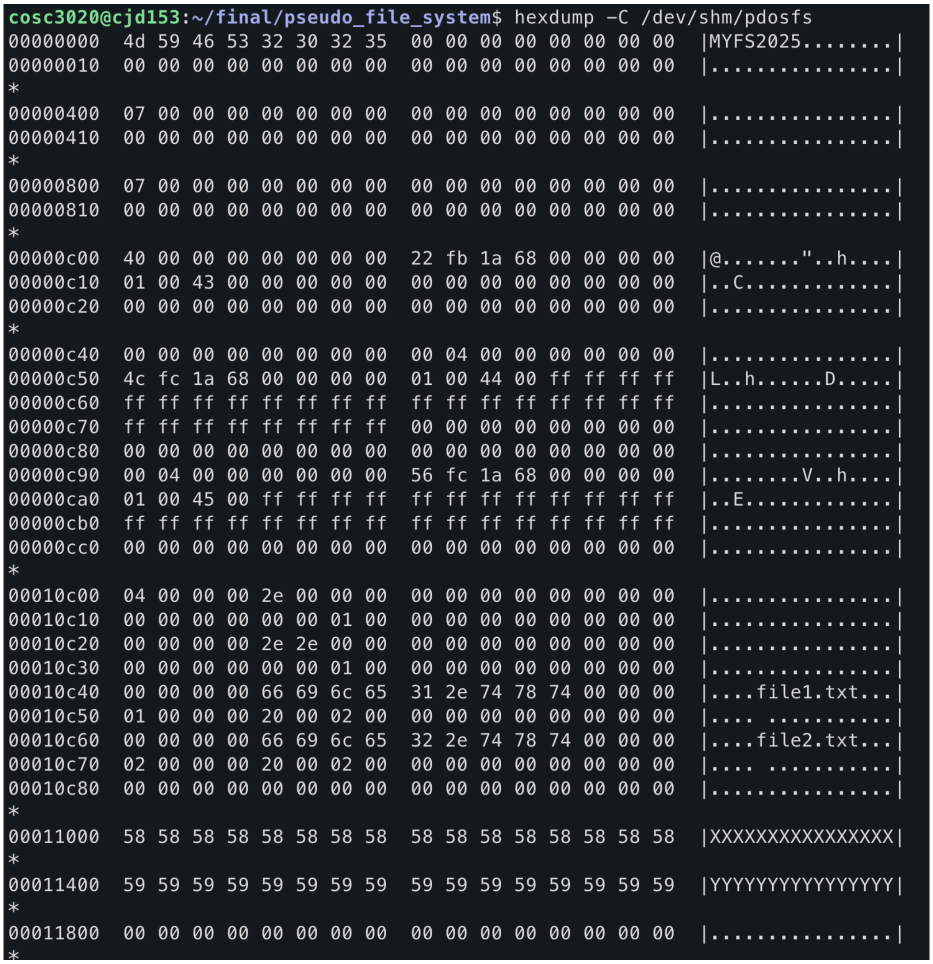
```
cosc3020@cjd153:~/final/pseudo_file_system$ ./test/test_read_file1
Read 1023 bytes from file1.txt:
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXcosc3020@cjd153:~/final/pseudo
_file_system$ 
```

<u>What is being tested, and how:</u>

After test_write_file1, we want to show that adding more characters does not compromise those already present.

We open file 1. We then create a buffer of size 1024 of ‘X’ characters and append it to the file. We then read in each character in file 1 and output the total number of bytes (characters) read, which should be 1024 more than after test_write_file1.

<u>Hexdump screenshot:</u>

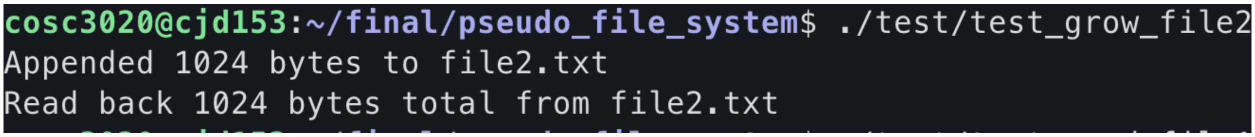


```
cosc3020@cjd153:~/final/pseudo_file_system$ hexdump -C /dev/shm/pdosfs
00000000  4d 59 46 53 32 30 32 35  00 00 00 00 00 00 00 00  |MYFS2025........|
00000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000400  07 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000410  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000800  07 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000810  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000c00  40 00 00 00 00 00 00 00  22 fb 1a 68 00 00 00 00  |@......."..h....|
00000c10  01 00 43 00 00 00 00 00  00 00 00 00 00 00 00 00  |..C.............|
00000c20  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000c40  00 00 00 00 00 00 00 00  00 04 00 00 00 00 00 00  |................|
00000c50  4c fc 1a 68 00 00 00 00  01 00 44 00 ff ff ff ff  |L..h......D.....|
00000c60  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff  |................|
00000c70  ff ff ff ff ff ff ff ff  00 00 00 00 00 00 00 00  |................|
00000c80  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000c90  00 04 00 00 00 00 00 00  56 fc 1a 68 00 00 00 00  |........V..h....|
00000ca0  01 00 45 00 ff ff ff ff  ff ff ff ff ff ff ff ff  |..E.............|
00000cb0  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff  |................|
00000cc0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00010c00  04 00 00 00 2e 00 00 00  00 00 00 00 00 00 00 00  |................|
00010c10  00 00 00 00 00 00 01 00  00 00 00 00 00 00 00 00  |................|
00010c20  00 00 00 00 2e 2e 00 00  00 00 00 00 00 00 00 00  |................|
00010c30  00 00 00 00 00 00 01 00  00 00 00 00 00 00 00 00  |................|
00010c40  00 00 00 00 66 69 6c 65  31 2e 74 78 74 00 00 00  |....file1.txt...|
00010c50  01 00 00 00 20 00 02 00  00 00 00 00 00 00 00 00  |.... ...........|
00010c60  00 00 00 00 66 69 6c 65  32 2e 74 78 74 00 00 00  |....file2.txt...|
00010c70  02 00 00 00 20 00 02 00  00 00 00 00 00 00 00 00  |.... ...........|
00010c80  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00011000  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58  |XXXXXXXXXXXXXXXX|
*
00011400  59 59 59 59 59 59 59 59  59 59 59 59 59 59 59 59  |YYYYYYYYYYYYYYYY|
*
00011800  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
```

How we know that means it's working: We see what already existed in the data blocks, but now more, indicating the newly-added characters. No previously-existing characters in the data blocks have been compromised.

test_grow_file2:

Screenshot:



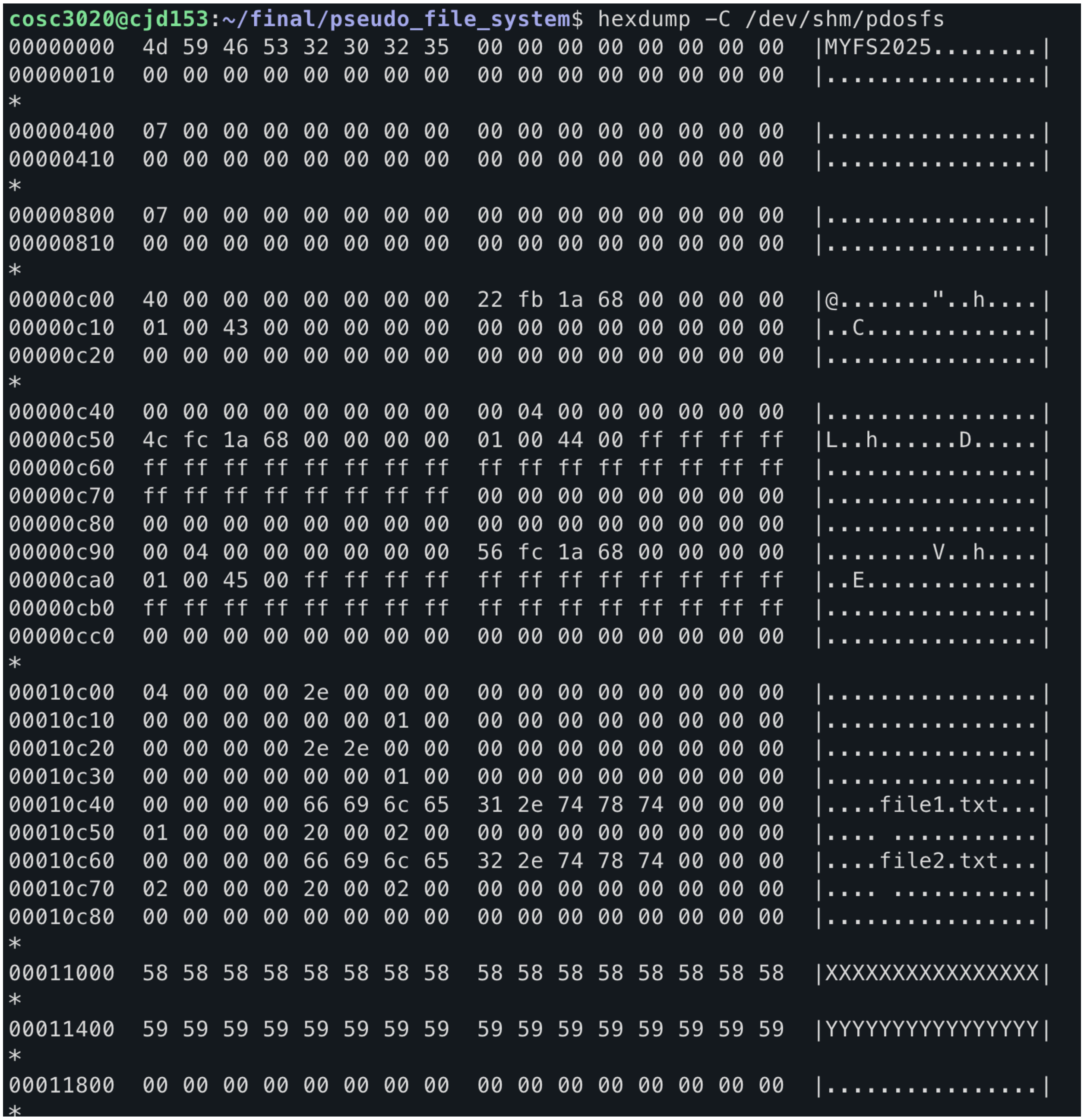
```
cosc3020@cjd153:~/final/pseudo_file_system$ ./test/test_grow_file2
Appended 1024 bytes to file2.txt
Read back 1024 bytes total from file2.txt
```

What is being tested, and how:

After test_write_file2, we want to show that adding more characters does not compromise those already present nor collide with those in file 1.

We open file 2. We then create a buffer of size 1024 of 'X' characters and append it to the file. We then read in each character in file 2 and output the total number of bytes (characters) read, which should be 1024 more than after test_write_file2.

Hexdump screenshot:

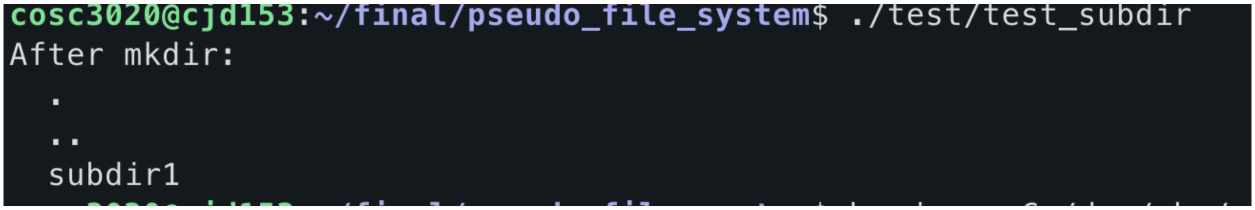


```
cosc3020@cjd153:~/final/pseudo_file_system$ hexdump -C /dev/shm/pdosfs
00000000  4d 59 46 53 32 30 32 35  00 00 00 00 00 00 00 00  |MYFS2025........|
00000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000400  07 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000410  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000800  07 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000810  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000c00  40 00 00 00 00 00 00 00  22 fb 1a 68 00 00 00 00  |@......."..h....|
00000c10  01 00 43 00 00 00 00 00  00 00 00 00 00 00 00 00  |..C.............|
00000c20  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000c40  00 00 00 00 00 00 00 00  00 04 00 00 00 00 00 00  |................|
00000c50  4c fc 1a 68 00 00 00 00  01 00 44 00 ff ff ff ff  |L..h......D.....|
00000c60  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff  |................|
00000c70  ff ff ff ff ff ff ff ff  00 00 00 00 00 00 00 00  |................|
00000c80  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000c90  00 04 00 00 00 00 00 00  56 fc 1a 68 00 00 00 00  |........V..h....|
00000ca0  01 00 45 00 ff ff ff ff  ff ff ff ff ff ff ff ff  |..E.............|
00000cb0  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff  |................|
00000cc0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00010c00  04 00 00 00 2e 00 00 00  00 00 00 00 00 00 00 00  |................|
00010c10  00 00 00 00 00 00 01 00  00 00 00 00 00 00 00 00  |................|
00010c20  00 00 00 00 2e 2e 00 00  00 00 00 00 00 00 00 00  |................|
00010c30  00 00 00 00 00 00 01 00  00 00 00 00 00 00 00 00  |................|
00010c40  00 00 00 00 66 69 6c 65  31 2e 74 78 74 00 00 00  |....file1.txt...|
00010c50  01 00 00 00 20 00 02 00  00 00 00 00 00 00 00 00  |.... ...........|
00010c60  00 00 00 00 66 69 6c 65  32 2e 74 78 74 00 00 00  |....file2.txt...|
00010c70  02 00 00 00 20 00 02 00  00 00 00 00 00 00 00 00  |.... ...........|
00010c80  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00011000  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58  |XXXXXXXXXXXXXXXX|
*
00011400  59 59 59 59 59 59 59 59  59 59 59 59 59 59 59 59  |YYYYYYYYYYYYYYYY|
*
00011800  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
```

How we know that means it’s working: We see what already existed in the data blocks, but now more, indicating the newly-added characters. No previously-existing characters in the data blocks have been compromised.

test_subdir:

Screenshot:

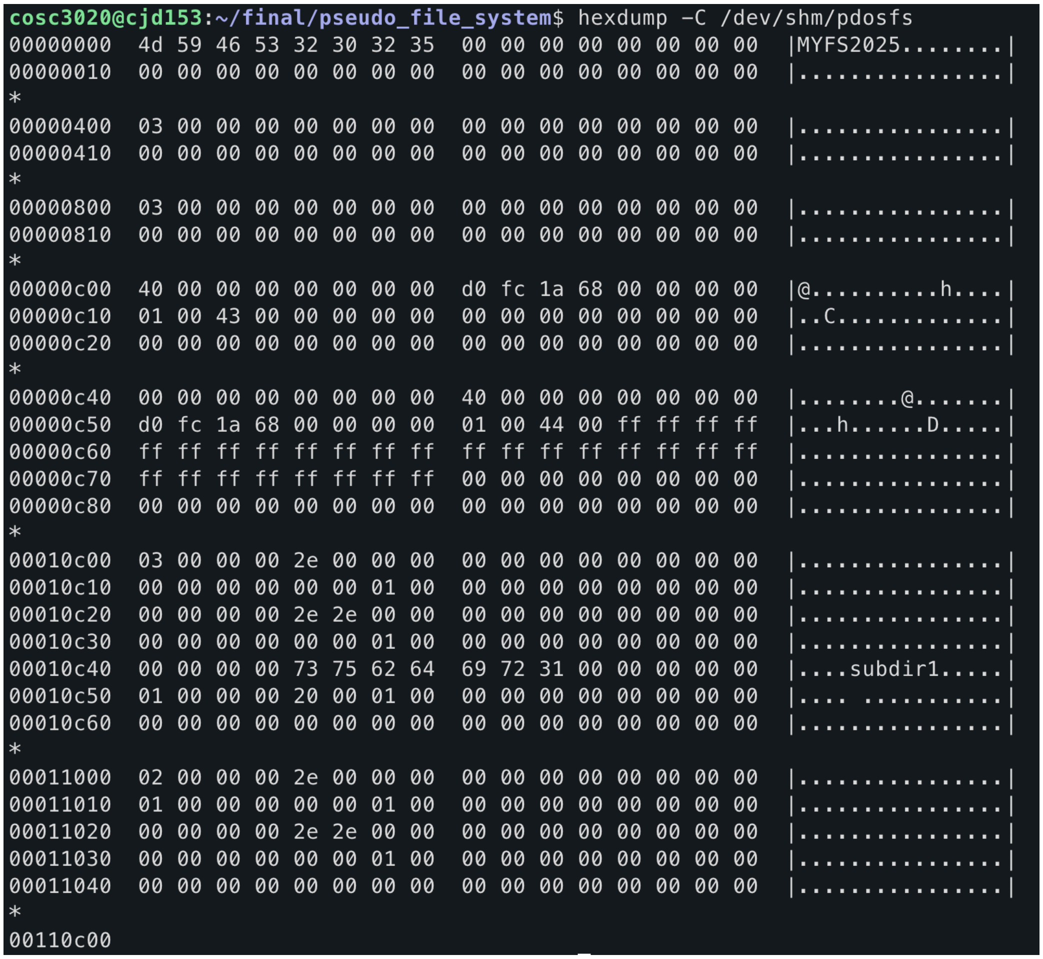


```
cosc3020@cjd153:~/final/pseudo_file_system$ ./test/test_subdir
After mkdir:
  .
  ..
  subdir1
```

What is being tested, and how:

We want to show that we can create a subdirectory within the root directory. We make the disk and file system, then make the subdirectory, and then list root directory contents.

Hexdump screenshot:

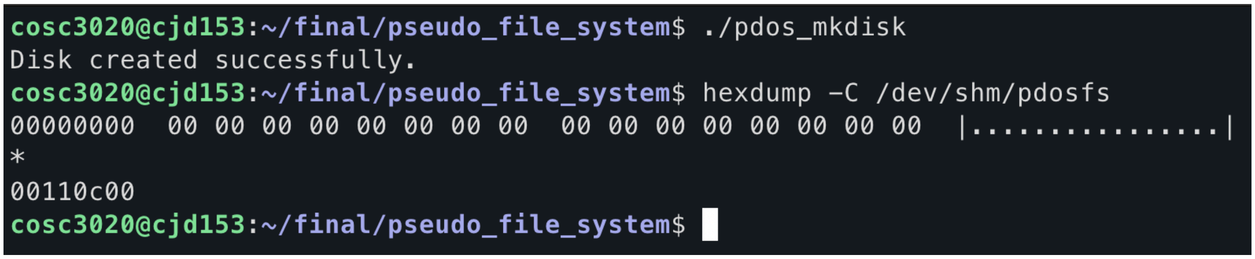


```
cosc3020@cjd153:~/final/pseudo_file_system$ hexdump -C /dev/shm/pdosfs
00000000  4d 59 46 53 32 30 32 35  00 00 00 00 00 00 00 00  |MYFS2025........|
00000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000400  03 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000410  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000800  03 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000810  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000c00  40 00 00 00 00 00 00 00  d0 fc 1a 68 00 00 00 00  |@.........h....|
00000c10  01 00 43 00 00 00 00 00  00 00 00 00 00 00 00 00  |..C.............|
00000c20  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000c40  00 00 00 00 00 00 00 00  40 00 00 00 00 00 00 00  |........@.......|
00000c50  d0 fc 1a 68 00 00 00 00  01 00 44 00 ff ff ff ff  |...h......D.....|
00000c60  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff  |................|
00000c70  ff ff ff ff ff ff ff ff  00 00 00 00 00 00 00 00  |................|
00000c80  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00010c00  03 00 00 00 2e 00 00 00  00 00 00 00 00 00 00 00  |................|
00010c10  00 00 00 00 00 00 01 00  00 00 00 00 00 00 00 00  |................|
00010c20  00 00 00 00 2e 2e 00 00  00 00 00 00 00 00 00 00  |................|
00010c30  00 00 00 00 00 00 01 00  00 00 00 00 00 00 00 00  |................|
00010c40  00 00 00 00 73 75 62 64  69 72 31 00 00 00 00 00  |....subdir1.....|
00010c50  01 00 00 00 20 00 01 00  00 00 00 00 00 00 00 00  |.... ...........|
00010c60  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00011000  02 00 00 00 2e 00 00 00  00 00 00 00 00 00 00 00  |................|
00011010  01 00 00 00 00 00 01 00  00 00 00 00 00 00 00 00  |................|
00011020  00 00 00 00 2e 2e 00 00  00 00 00 00 00 00 00 00  |................|
00011030  00 00 00 00 00 00 01 00  00 00 00 00 00 00 00 00  |................|
00011040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00110c00
```

How we know that means it's working: We see modified information in the data block.

Wrapper files and their output

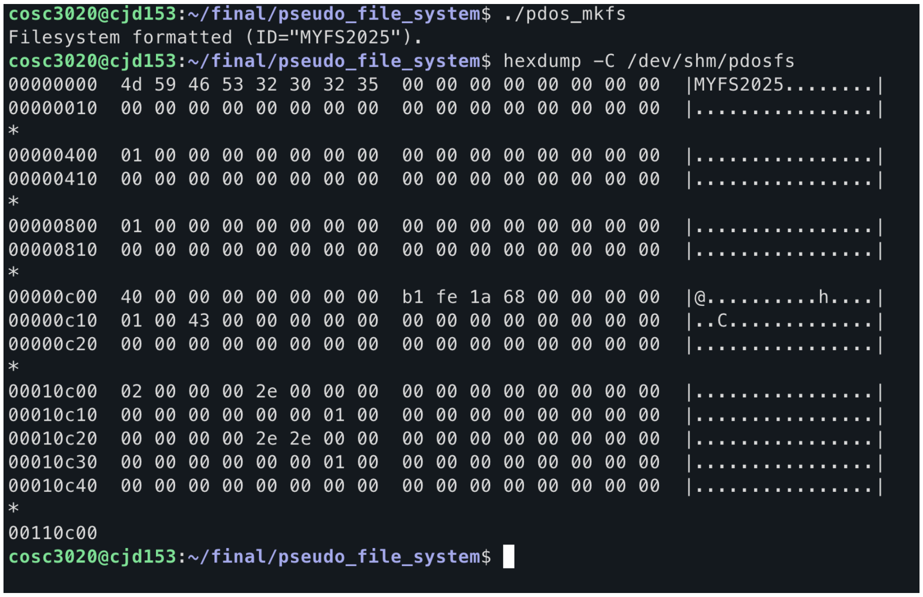
./pdos_mkdisk



```
cosc3020@cjd153:~/final/pseudo_file_system$ ./pdos_mkdisk
Disk created successfully.
cosc3020@cjd153:~/final/pseudo_file_system$ hexdump -C /dev/shm/pdosfs
00000000  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00110c00
cosc3020@cjd153:~/final/pseudo_file_system$ 
```

Here you can see the disk has been created and is empty.

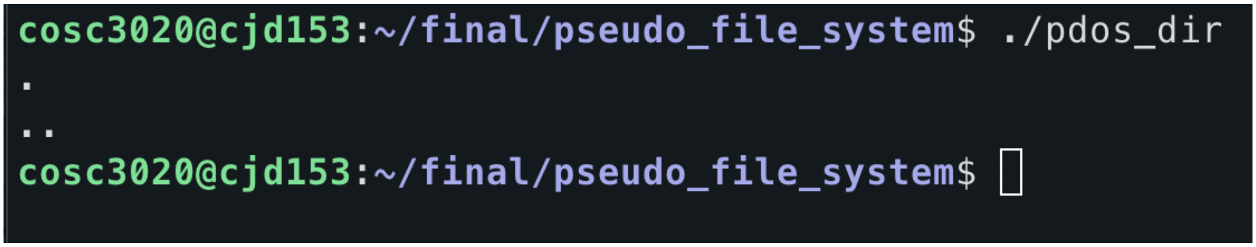
./pdos_mksf



```
cosc3020@cjd153:~/final/pseudo_file_system$ ./pdos_mkfs
Filesystem formatted (ID="MYFS2025").
cosc3020@cjd153:~/final/pseudo_file_system$ hexdump -C /dev/shm/pdosfs
00000000  4d 59 46 53 32 30 32 35  00 00 00 00 00 00 00 00  |MYFS2025........|
00000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000400  01 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000410  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000800  01 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000810  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000c00  40 00 00 00 00 00 00 00  b1 fe 1a 68 00 00 00 00  |@..........h....|
00000c10  01 00 43 00 00 00 00 00  00 00 00 00 00 00 00 00  |..C.............|
00000c20  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00010c00  02 00 00 00 2e 00 00 00  00 00 00 00 00 00 00 00  |................|
00010c10  00 00 00 00 00 00 01 00  00 00 00 00 00 00 00 00  |................|
00010c20  00 00 00 00 2e 2e 00 00  00 00 00 00 00 00 00 00  |................|
00010c30  00 00 00 00 00 00 01 00  00 00 00 00 00 00 00 00  |................|
00010c40  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00110c00
cosc3020@cjd153:~/final/pseudo_file_system$ 
```

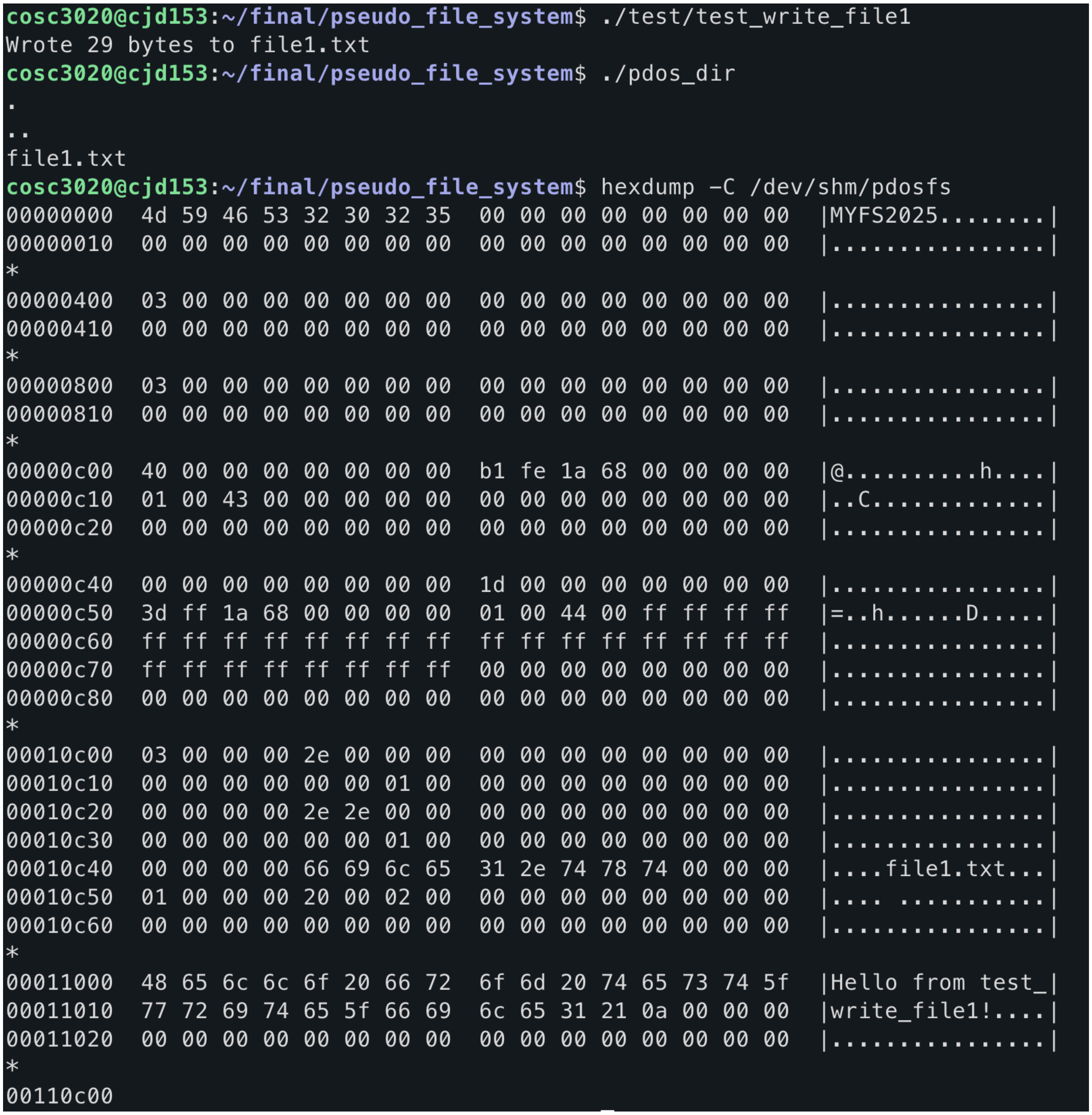
Now you can see that the file system has been created, formatted and named. The " . " and ". ." have been added and the superblock contains file system information.

./pdos_dir



```
cosc3020@cjd153:~/final/pseudo_file_system$ ./pdos_dir
.
..
cosc3020@cjd153:~/final/pseudo_file_system$ 
```

Here we print our the directory and can see that these were added when formatting the file system.



```
cosc3020@cjd153:~/final/pseudo_file_system$ ./test/test_write_file1
Wrote 29 bytes to file1.txt
cosc3020@cjd153:~/final/pseudo_file_system$ ./pdos_dir
.
..
file1.txt
cosc3020@cjd153:~/final/pseudo_file_system$ hexdump -C /dev/shm/pdosfs
00000000  4d 59 46 53 32 30 32 35  00 00 00 00 00 00 00 00  |MYFS2025........|
00000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000400  03 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000410  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000800  03 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000810  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000c00  40 00 00 00 00 00 00 00  b1 fe 1a 68 00 00 00 00  |@..........h....|
00000c10  01 00 43 00 00 00 00 00  00 00 00 00 00 00 00 00  |..C.............|
00000c20  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000c40  00 00 00 00 00 00 00 00  1d 00 00 00 00 00 00 00  |................|
00000c50  3d ff 1a 68 00 00 00 00  01 00 44 00 ff ff ff ff  |=..h......D.....|
00000c60  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff  |................|
00000c70  ff ff ff ff ff ff ff ff  00 00 00 00 00 00 00 00  |................|
00000c80  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00010c00  03 00 00 00 2e 00 00 00  00 00 00 00 00 00 00 00  |................|
00010c10  00 00 00 00 00 00 01 00  00 00 00 00 00 00 00 00  |................|
00010c20  00 00 00 00 2e 2e 00 00  00 00 00 00 00 00 00 00  |................|
00010c30  00 00 00 00 00 00 01 00  00 00 00 00 00 00 00 00  |................|
00010c40  00 00 00 00 66 69 6c 65  31 2e 74 78 74 00 00 00  |....file1.txt...|
00010c50  01 00 00 00 20 00 02 00  00 00 00 00 00 00 00 00  |.... ...........|
00010c60  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00011000  48 65 6c 6c 6f 20 66 72  6f 6d 20 74 65 73 74 5f  |Hello from test_|
00011010  77 72 69 74 65 5f 66 69  6c 65 31 21 0a 00 00 00  |write_file1!....|
00011020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00110c00
```

Now here you can see we created and wrote to a file and tested pdos_dir again. You can see that file was added to the directory and its information is in the hexdump