Project documentation

Problems faced :

- after writing strnum function and trying to use it for the first time I faced an error "undefined reference for strnum", and after a few attempts to find the root of the problem, I discovered that Hfunc.c (where all the functions are implemented) isn't accessed after calling make, so there was no implementation for them.
- figuring out how many blocks are in a file in the dmount function
- used a pointer type with a variable in the first argument of fstat
- One of the most time consuming problems I faced was with strnum function where the makefile wasn't accessing strnum's implementation
- dd needed more than 32bits in dinit implementation
- the problem of undefined reference also faced me in dinit. after diving a bit into the problem it turns out that the Hfunc.c was not compiled with #fPIC ; because I forgot to put the so flags in the compilation in the Makefile
- dd(in pointer) needed long long int 64bits instead of 32bits
- then when testing dinit before attempting to mounting a disk there was a problem with the naming where I had named the experimental drive drive.0 where it should've been disk 1 and used 0,1 numbering instead of 1, 2 while calling dmount which was the reason why it gave zero adress, because BaseFPath in dist.h has disk as the final destination NOT DRIVE
- when executing the dshow function it printed zero as the number of blocks which means it exceeded the variable size, it turns out that I should've put (stat.st_blocks - 1) instead of it without (-1) since it 2bytes can only hold up to 65 536, first thought the filesize was larger than what I anticipated but after dividing it with the block size and getting 65 535 and subtracting one was sure it had something to do with the variable size (needs)

PD 2

- undefined reference to (dmount, format) in Hfunc (hidden symbol / disk is not defined)
solve by: defining the disk declared inside, format public rather than descriptor outside the init function, internal
reason: (search)

- when attempting to execute ./tempexec; it printed: error while loading shared libraries: FSAPI.so: cannot open shared library object: no such file or directory
Solution: replace 1 in the Makefile (lib Forex .so: not found) it didn't compile a file based on a shared obj file .so rather than:
tempexec tempexec.o (FSAPI.so) => Hfunc.o DiskEmu.o Mystd.o Filesystem.o)
which then worked

- when attempting to format a disk it printed a segmentation fault, after using the old debugging method (placing printf statements) I was able to deduce that (finit call) causing the fault
It turns out that when I commented the dmounting calls and the print statements in dinit
- I had forgotten to comment out the dmount function which caused the fault, and left the dmountin for initialisation of file descriptors, when testing a bit it turns out the execution
stops and gives an error if the dmounted descriptor isn't unmounted if
before it reaches the format call it gives a segfault
After searching in format a bit I discovered that after the disk is formatted instead of making the drive number of the filesystem equal to the disk descriptor drive number
I had accidentally assigned the drive number of the disk descriptor of the filesystem equal to the drive number of the disk descriptor rendering my completely ineffective ... لتكون

PD 2

- undefined references to (dmount, format) in Hfunc
  solve by: defining the disk descriptor outside the init function
  reason: (search)
  was when attempting to execute ./tempexec; it printed: error while loading shared libraries: FSAPI.so: cannot open shared library object no such file or directory
  solution: replace 1 in the Makefile (FSAPI.so: cannot find) compile a file based on a shared obj file so rather than: tempexec tempexec.o (FSAPI.so) => Hfunc.o DiskEmio.My std.o Filesystem.o) which then worked.

- when attempting to format the disk it printed a segmentation fault (core dumped), after using the old debugging method (placing printf statements) I was able to deduce that the (dmount) call was causing the fault.

- It turns out that that example when I commented the dmounting calls and the printf statements in dinit, I had forgotten to comment out the dmount function which caused the fault, and left the dmounting for initialization of file descriptors, when testing a bit it turns out the the execution never enters dinit if the dmount call gives an error if the provided descriptor isn't mounted inside the init function before it reaches the format call it gives a seg fault.

After debugging searching in format instead of bit I discovered that after the disk is formatted instead of making the drive number of the filesystem equal to the drive number of the disk descriptor I had accidentally assigned the drive number of the disk descriptor to the drive number of the disk descriptor resulting in my completely ineffective ... لتكون

01 Later hexdumped disk.img to make sure the (number of blocks, inode blocks, inodes, and the magic number) and inode blocks are correct. The super block was fine but

the inode block printed the a random number on the left of the validity byte field (03), which made me make sure of the size of the inodes which turns out to be (34) instead of 32.

although the sum of the sizes of its field is 32 bytes.

- It turns out that I had placed packed at the beginning of the struct rather the end (picked this wrong habit from somewhere). problem solved inode; the DISK IS FORMATTED ☺

(Note: the directory didn't occupy any blocks so I didn't include it in the size variable)

- when attempting for un fs show in format it printed a negative numbers as the num of blocks and the number of inode blocks was the number of blocks. sol: I accidentally placed a comma after the string in the first line (outside the "" marks) which caused this.

- Strace / trace to debug

- another problem with show is that it didn't print info about the inode we had cause, the loop in (search inode) function didn't break out of the outer loop when the (n==idn) condition is met, as it only broke out of the inner one, sol: I set x to the size of inode blocks so the outer loops condition is satisfied no more

- Shared header errors

after finishing show, mountfs, unmountfs, I faced an error of multiple definition of DiskDescriptor caused it once public in fs.h and one extern in filesystem. removed the extern and replaced keyword public with extern in FileSystem.h

And while I was writing the inode functions, I faced an error that no output given when I run execute.com so I traced it and discovered that I had changed the path of "drives" folder without changing the basepath.

after that while attempting to run execute.com after creating multiple inodes it gave a stack smashing error. I traced it and discovered that it was from the dwrite command in the for loop of the reservein function.

I faced a moment when trying to compile after writing show I got at least 5 pages worth of errors reason: else for the wrong if :)

[illegible]

P2 → name should be in quotes (in cmp function in readdir) (add &)

forgot to put pwd [illegible] (in cd) rather than filesystem [illegible] (I copied [illegible] and [illegible] that's why)

the 20 page errors (a [illegible] [illegible] [illegible] syntax errors)