Teletype id:

????????

As of now all the functions work in the way they are supposed to.but write function is still being very stubborn cause we cant write into after writing the first time! :(

Make sure you read all the things changed and the issues .(nitish i have specified what you should start working on i.e implementing ls -t,show all the access times of a file)(pavan start working on persistence ,leave all the sfs edge cases to us for now! I will explain everything to u if you want me to later)

Once the main issues are sorted out , we have to just experiment extensively and repeat the process! :(

Things changed:

para:

* Added many debug print lines in many different functions along with printing the status of the filetable in each function.(its called “showFileTableContents()”)
* Fixed write,read and all functions to make it work at least one time.(read will work everytime provided write succeeds the second time onwards!)
* Added a function called “flushfiletable” which flushes the filetable anytime we delete a particular file as the file descriptor values could still exist
* In the main shell we can also call the function close ,which will remove the first entry for this file in the filetable(cause before, we were closing everytime we read from the file meaning we couldn't write again as that filetable entry would go.so i removed the functionality of closing a file on reading)
* We can only write MAX\_CONTENT\_LIMIT amount of content everytime we want to write
* As of now for debugging purposes the block size is only 2 and not 128 so we should change it in the end after everything works fine
* Changed the way we write now we have to just say write <filename> and then enter the content once there is a prompt(did this as that would only write the first word into the file due the way we split the user input to the shell )
* In the makefile we now have to compile with the flag -lm as i am using <math.h> in write.
* Removed nlink member in inode.(lets not complicate things shall we!)
* Incremented the fdcount member of the inode whenever we open a file !(if that process already has it open i dont increment ,otherwise if another process is adding it then i increment it!)(cant see this working until we implement the CHANGING SHELL WORKAROUND!!(see issues point #5))
* Decremented fdcount whenever we have to remove an entry from the file table.

Nitish:

Pavan:

Issues+things to still do:

para:

* Cant write into the file the second time .(even after unlinking all the datablocks for some reason the datablocks are not NULL!) .the filetable is also not updated
* Cant write into the same file once i close this file,ie in the filetable this entry goes away! But the next time we write it should ideally overwrite from the beginning(but cant do so cause of the same unlinking datablocks error in write)( THIS BUG SEEMS TO BE THE ONLY THING STANDING BETWEEN US AND WORLD DOMINATION!!!!)
* When we write there is no way to currently to write data with newline character as on pressing enter it would start executing sfswrite.
* We should probably change the ls function and include a flag which shows the file times too!(nitish start with this!! Make sure to update the right times in write and read.)
* Our filetable notion could be wrong!.As i was initially thinking of opening multiple shells and show different processes writing into the same file .but the problem is if we create a file say “a” in one shell and write into it. The other shell will never know that “a” was created !!.so ideally everything should have been multiple threads rather than multiple processes .or we can do a very simple workaround and have a global variable called “current shell” or something so in the shell we can manually switch between processes by typing some command.(say something like switch <pid> so any files written will have that pid.)(i think we should do this workaround later as it is very easy!)(NITISH WORK ON THIS AFTER IMPLEMENTING “ls -t” and change all the places we used getpid() with this new global value called “currentshellpid” check in the globals file)

Nitish:

Pavan:

* Bro try looking at how to make it persistent .
* All we have to do is apparently write all the structures content into a file (say by using memcpy) in a specific format.
* Since the current state of the filesystem is working with all the minimum functionalities you should start doing this right away
* Start writing the code in automount.c and dumpfs.c ASAP. for now me and nitish will work on fixing the edge cases,so you start doing this .I will explain sfs code if you need me to anytime
* Start by writing all the prototypes and the globals in the file “globalConstants.h” and change the makefile appropriately