Cameron Bourque UIN: 826004886

Design:

File:

For the file class I started by adding the file id, size, position and disk as private members so the class had access to them. For the constructor, I made it so the file id, size and disk are passed in. The rest of the functions were passed in.

For the constructor, I set up all the private variables.

For the read function, I create a buffer which is the size of a block in the disk. Then I read the block into the buffer. Then from the current position, I read each character into the passed in buffer until either the current position reaches the end of file or n characters were read. Then I return the number of characters read.

For the write function, I create a buffer which is the size of a block in the disk. Then I read the block into the buffer. Then I use the passed in buffer to alter the buffer from the current position until n characters are written. If the current position reaches the end of file then we increment the size and keep going. Once we finish altering the buffer, we write the buffer back into the disk.

For the reset function I change the current position to 0 so it's at the beginning of the file.

For the rewrite function I create a buffer which is the size of a block in the disk and write 0's into it. Then we write the buffer into the disk.

For the EoF function I check whether the current position is equal to the size of the file.

File System:

For the file system class I started by adding disk, size and files array as private variables. The rest of the class I left as is.

For the constructor, I set up all the private variables.

For the mount function, I set the disk and size and then create the files array based on the size of the disk. Then we return true.

For the format function I create a buffer the size of a disk block and fill it with 0's. Then, for each block of the disk I write the empty buffer into it. Then we return true.

For the lookup file function I return the file at the file id index of the files array. This could be null if it doesn't exist.

For the create file function I check if the file at the file id index of the files array is not null or that the disk is null or the file id is not in range to fit in any blocks in the disk. If any of these conditions are true then I return false. Otherwise a new file is created with the file id, size 0, and the disk and placed in the files array at the file id index. Then true is returned.

For the delete file function I check if the disk is null or if the file id is not in range of any blocks in the disk. If any of these conditions are true then we return false. Otherwise we check if the file at the file id index of the files array is null. If so then there is no need to delete so true is returned. Otherwise

we tell the file at the file id index of the files array to rewrite so it is empty. Then we delete the file and set it to null in the files array. Then we return true.