CPSC 351 Project 3 (100 points)

This project can be done individually or in a group of maximum 3 people. For a group of 2 or 3, each of the group members needs to submit. If one group member fails to submit, that person gets 0. Indicate in an additional readme.txt file, the names and email ids of members in the group. If working individually, indicate in the readme.txt, your name and email id.

The file_sys_example.cpp on Titanium opened an existing input file for reading. It also created an output file for writing the data read from the input file. It used the read and write system calls. The input file we used was 67 bytes.

Now we will use memory mapped files instead of the read and write. Also now the input file, inFile.txt will be of size 1 GB.

1) Create an input text file, called inFile.txt of size 1 GB by typing (and then pressing enter) the following command on the terminal:

   ```
   yes This is what my input file has...| head -c 1GB >> inFile.txt
   ```

   Note: The inFile.txt is 1 GB. Don't bother opening to see the file what it contains. Instead run stat command from terminal to check its size. If you are curious to see what the yes command does, try running the following command by creating a smaller file:

   ```
   yes This is what my input file has...| head -c 1KB >> testFile.txt
   ```

   and then open testFile.txt to look at it.

2) Now, start from the given mmap.cpp on Titanium. The timing functionality as demonstrated in class, is already added to this file. The stats struct object is also getting the file size.
   a) Add to this program, the functionality of reading **all data** from inFile.txt and writing to an output File. The output file is to be created from within the pgm using creat. Note that currently in mmap.cpp, only pagesize (4096 bytes) amount of data is being mapped to RAM. Now your file is 1 GB.
   b) Additionally, in mmap.cpp, you are not writing to an output file. Add the functionality of writing the mapped data to an output file, but not using a plain "write" system call.

      Hint: Use a combination of mmap and memcpy for this. Also remember that the OS creates a backup of pages being modified in the memory.

   Compile the program:

   ```
   clang++ mmap.cpp -o test
   ```

Run the program:

```
./test inFile.txt outFile.txt
```

How much time does it take ? You can take an average of several runtimes.

3) Answer the timing question above, in the readme.txt file.

To be submitted

A. mmap.cpp (with your code)
B. Readme.txt

Note

The inFile.txt and outFile.txt are 1 GB each. Don't bother opening to see the files what they contain. Instead run stat command from terminal to check their size. Delete the files, when not running the pgm. They are taking up a sizable amount of space on your disk.

Also, the timing is not really relevant here, as there is no point of comparison. It comes into the picture when several processes may want to read from the mapped region, and you have read the file only once from the disk. But nevertheless, you know how to time a function.

Blurb for your resume

Use your GitHub account as a ready-to-show portfolio of your programming projects to potential employers. Write the description of your project in your own words.

Note: This project involves the concept of memory mapped files within the Linux OS environment.