

CPSC 351 Project 3 (120 points)

Deadline: Friday, May 20, 11:59 PM

The `file_sys_example.cpp` on Canvas 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 68 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 Canvas. The `stats` struct object is 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 program using `creat`. If `creat` does not work, you can create a blank output file beforehand. And open that file in the program.

Note that currently in `mmap.cpp`, only `pagesize` (4096 bytes) amount of data is being mapped to RAM. Now your file is 1 GB. In this project, you should map only 100 pages at a time. You just have to figure out how to map the entire 1 GB file data in chunks of 100 pages. Hint: Use a `for` or `while` loop.

- 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.

- 3) Answer the following briefly in a few sentences in answer.txt.
- a) Are the files in your program sharing the same memory region ?
 - b) Should they ? Justify your answer.

(Note that you should not use read and write system calls for this project. In fact, you should not be using “any other” system calls / functions for this project, except of course the open, creat, mmap, munmap, memcpy.)

Compile the program:

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

Run the program:

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

To be submitted

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

Group Work

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.

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.

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.