CS307 HOMEWORK 4: MEMORY MAPPING & FILE READING

Name: Dilara Nur Memiş

SU ID: 27868

In this homework, I read a file named "loremipsum.txt" and count the occurrences of character "a" with three different methods. The result is 19082160 (same for all methods). In order to compare their performance, I add clock_t objects to the beginning and end of the counting operations and print out the total time in seconds in all methods.

- **First Method (fstream):** I wrote a CPP file in which I used fstream for reading the file. When I run it on flow server, total time for counting stated as 3 seconds.
- **Second Method (fopen):** I wrote a C file and used fopen for reading the file. It tooks 3.57 seconds to find number of a's on flow server.
- **Third Method (mmap):** By using mmap() function, I mapped the file into the virtual memory space of the process. Then I read the file and did necessary operations. It required 1 second to count a's on flow server.

Time-efficiency: As it is stated above, reading the memory-mapped file (3rd method) is almost 3 times faster than the other ones. Reading the file by using fstream and fopen required approximately same amount of time which is around 3 seconds. With mmap(), it only took 1 second. Therefore, in terms of time efficiency, mapping is the most efficient method among these three for reading the file. Mapping a file into memory allows access to data in the file as if that data is in an array in the program. It prevents the OS from allocating, deallocating data buffers owned by the process. If the page size is big (as in the case of loremipsum.txt), using memory mapping instead of fopen(), fstream would be much better for performance. Small files can be read by using fopen(), mmap() without even realizing the difference in terms of time efficiency.

Implementation: In terms of easiness of implementation, C++ file in which I use fstream (1st method) is the most advantageous. It is simpler to use. Besides, when you use fstream, there is no risk of file corruption if you forget to close the file explicitly. In such a case, file is closed automatically when the program terminates. Moreover, you don't need to specify the open mode in contrast to fopen() and mmap() methods. If you do not provide the open mode, the file is opened in read mode by default. Memory mapping method, on the other hand, is really hard to implement compared to the first one. Dealing with pointers and complex syntax makes it easier to make mistakes. In both fopen() and mmap() methods, programmer should close the file (and unmap the region in third method) manually. Otherwise, problems may occur.

To sum up, I prefer to use fstream for reading files that are not so big because it is high-level and easier to implement. If I need to read a big file, I would use memory mapping for fast access and time-efficiency.