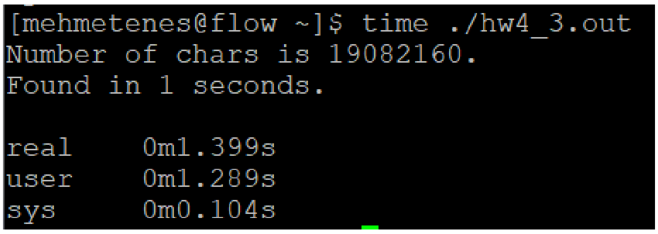
CS307 HW4 REPORT

Mehmet Enes Battal – 26354

In this project counting the specified character in a specified file is implemented using three different methods. In the first method which is in C++ language, fstream is used to get a stream from the specified file. After that using getline() function and looking at the characters in that line for the all lines in the file gives the result. In the second method which is in C language, fopen() function is used to get a pointer to that file and by using the fgetc() function in a loop all the file is read character by character. Third one is also in C language, but the file reading is different. First the specified file is mapped to a process address space using the mmap() function. After this mapping the pointer to the content of the file is obtained and the file can be accessed like an array.

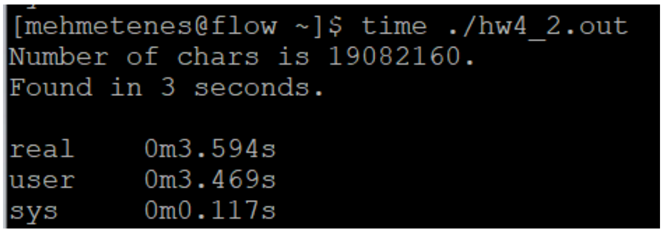
When it comes to the speed of these three methods, the fastest is the third, which uses memory mapping, because file is read from the process address space rather than the disk. There is no time spent to move the file content to address space as the reading progresses. It took 1 second to read and find the result and 1.4 seconds for the whole program to finish.



```
[mehmetenes@flow ~]$ time ./hw4_3.out
Number of chars is 19082160.
Found in 1 seconds.

real    0m1.399s
user    0m1.289s
sys     0m0.104s
```

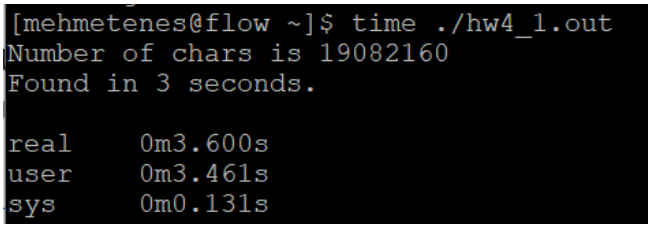
The other two methods are really similar since they first move the data to address space to read and the time they require is also almost the same. The difference is their language. In the tested machine, second method was 0.006 seconds faster than the first one. Second method took 3 seconds to read and 3.594 seconds for the whole program to finish.



```
[mehmetenes@flow ~]$ time ./hw4_2.out
Number of chars is 19082160.
Found in 3 seconds.

real    0m3.594s
user    0m3.469s
sys     0m0.117s
```

The first method also took 3 seconds to read the file and find the result but the whole process took 3.6 seconds on this one.



```
[mehmetenes@flow ~]$ time ./hw4_1.out
Number of chars is 19082160
Found in 3 seconds.

real    0m3.600s
user    0m3.461s
sys     0m0.131s
```

As it can be seen the third method which uses memory mapping is almost more than 2 times faster than the rest. The time complexities of all of them is O(n) where n denotes the number of characters in a file. Or in other words how many bytes the file is. All three methods use a loop to do the necessary operations for all the characters that’s why their time complexities are the same.

To sum up, all methods find the result correctly as 19082160 and it is more than 2 times faster to use memory mapping for reading files. But one must consider the portability issue of the program. Since the third method only woks on Linux it is not usable on other platforms whereas the other two can work on other platforms. The time they spend is proportional to the size of the file. For reference a file which is 266 megabytes takes approximately 1 second for the memory mapped method and approximately 3 seconds for the other methods.