Jerry Perez
CS3013 Project 4 Report
October 13, 2020

Project 4 gave us the opportunity to determine the difference in performance between the read() system call and the mmap() system call. Better yet, it also showed us the consequences of using chunk sizes side by side with the read system call in order to obtain the same end result. Through this project, the three methods should have yielded the same result but via different performance methods. Below will be a brief report highlighting the findings.

The files tested were with chunk sizes of 1 byte, 1k bytes, 4k bytes, and 8k bytes as well as memory mapped I/O (mmap).

I started off by creating 5 test files, each of which incrementing in byte size. The first file size was of byte size 2, the second file size with a byte size of 190, the third with a byte size of 1083, the fourth of 2166, the fifth with 4332, and the final with 24545.

Then, I used ./doit to run each of the files with ./proj4 being careful to specify the correct test file and the chunk size. The following table will demonstrate those findings.

File Size	1 byte	1k bytes	4k bytes	8k bytes	mmap
2	2.204	1.895	1.656	1.875	1.354
190	2.134	1.781	1.676	2.373	0.367
1083	2.851	1.521	1.559	1.509	1.486
2166	3.084	1.544	1.582	1.616	1.55
4332	5.919	1.623	1.379	1.496	1.478
24545	21.039	1.643	1.847	1.45	1.891

The graphical representation of this data is attached as a separate file within the project folder. However, it appears that overall, the 4k and 8k chunk searches give the better performance, especially when the file sizes start to creep up. When file size is low, the performance is relatively similar and won't be too much of a burden for analysis. However, as file sizes get larger, the performance of mmap does the same, and it could very well be because of the fact that mmap maps the contents of the file to memory and then does what it needs to. In addition to this, the 1 byte chunk search gets really slow and isn't the most viable option. I believe that for extremely large file sizes, the best performance is going to be seen with the 8k chunk size with the read command and fairly poor performance with mmap. If file sizes stay small then mmap is going to be the best method to use as it offers the best performance.