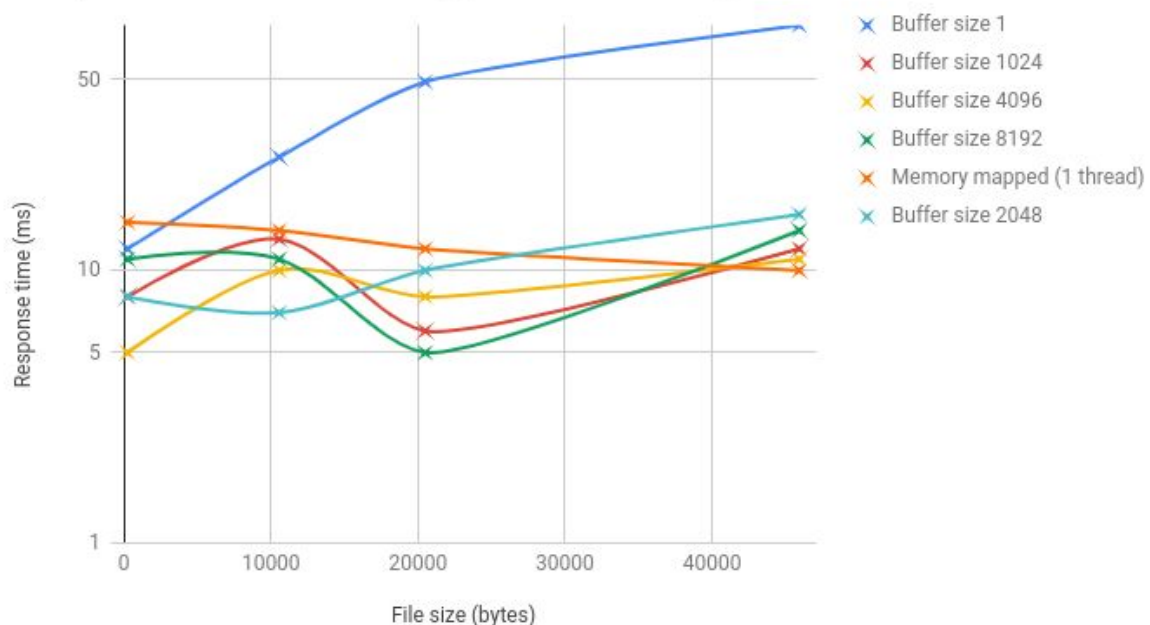


Project 4 - Operating Systems CS 3013 - Prof. Craig Wills
 Name: Pedro de Vasconcellos Oporto
 Username: pdevasconcelloso

****Important note:** Only a virtual machine was available to test the program, which does not have access to a multiprocessor. Additionally, all executions reported 0 major page faults.

Read Syscall vs Memory Mapping

Response time of different types of file reading



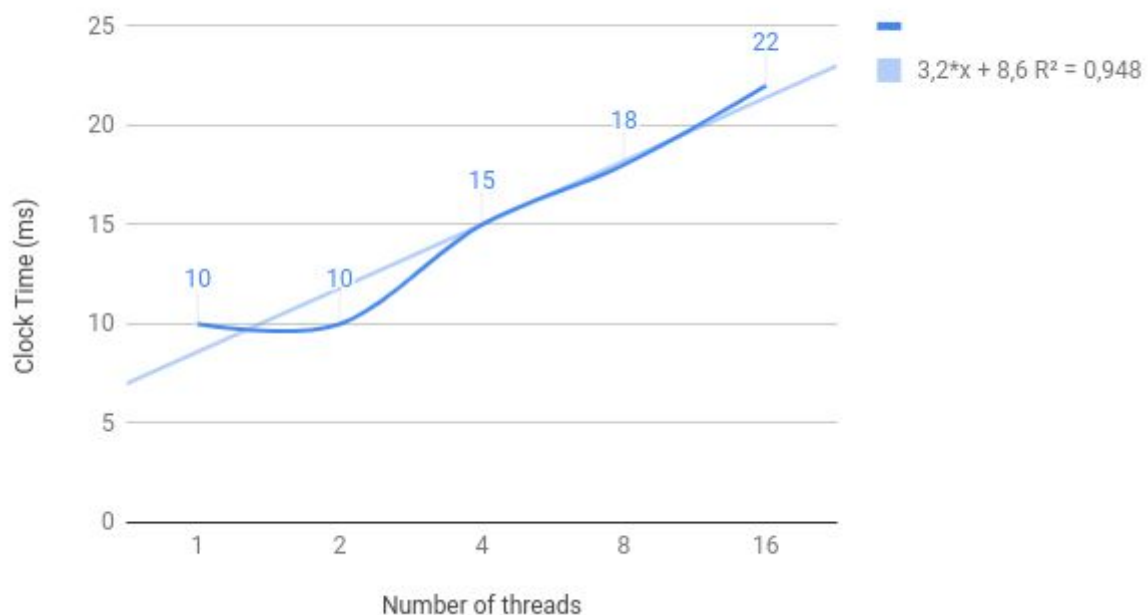
| File name | File size (bytes) | | Buffer size 1 | Buffer size 1024 | Buffer size 2048 | Buffer size 4096 | Buffer size 8192 | Memory mapped (1 thread) |
|-------------------|-------------------|------------|---------------|------------------|------------------|------------------|------------------|--------------------------|
| Makefile | 242 | | 12 | 8 | 8 | 5 | 11 | 15 |
| searchstrings.cpp | 10552 | Clock | 26 | 13 | 7 | 10 | 11 | 14 |
| searchstrings | 20488 | Time (ms): | 49 | 6 | 10 | 8 | 5 | 12 |
| doit | 45944 | | 79 | 12 | 16 | 11 | 14 | 10 |

Above is the result of running the **searchstrings** program on 4 different file sizes in 6 different configurations each. The configurations were using the *read* syscall using buffer of sizes 1, 1024, 2048, 4096, and 8192 bytes, as well as memory mapping the file and reading it in one go. Above are the tabled and graphed results of the different response (clock) times for the different configurations and files (Note that y axis is in a logarithmic scale). Looking at the graph, the response time taken using the read system call increases linearly or

exponentially, beyond 20K bytes file size. Except for when the buffer size is 1 byte, there aren't significant differences between other buffer sizes (from 1Kb to 8Kb). As a contrast, when the whole file is mapped to memory, the time to search the string through the whole file decreased over larger file sizes. It is not indicative that this trend would be true over even larger file sizes but it shows better performance over growth when the file is mapped to memory.

Multithreading Memory Map File

Clock Time for memory mapped file using threads



| Memory mapped threads | Clock Time (ms) |
|-----------------------|-----------------|
| 1 | 10 |
| 2 | 10 |
| 4 | 15 |
| 8 | 18 |
| 16 | 22 |

Above are the results of comparing searching through the memory mapped file using 1, 2, 4, 8, and 16 threads. The file `doit`, with 45944 bytes, was used. Not intuitively, the response (clock) time of the program execution increased linearly regarding to number of threads. The line of best fit has a slope of +3.2 ms/thread, with an R^2 of 0.948. This is likely due to the test not being run in a multiprocessor machine.