

CS 342
FALL 2019
PROJECT #4

PROJECT REPORT

BERK YILDIZ
21502040

I did my experiments with a computer which has processor intel core i5 vPro. The computer is seven years old and it is a slow machine when compared to today's computers. I tested performance of create(), append() and read() functions for different number files and byte sizes.

sfs_create() Stats

Number of Files	Time (ms)
1	48
5	58
10	90
25	147
50	245

Table 1: Results of sfs_open()

The time passed in milliseconds increases as the number of files increases but it is not a linear increasing. As the file number increases, growth rate of time decreases but it still increases. It is seen that creating a file is not a costly process for the computer.

sfs_append() Stats

Bytes	Time in ms (1 file)	Time in ms (2 files)
100	1553	2912
1000	7371	21482
5000	36781	90821
10000	77049	164040

Table 2: sfs_append() stats

It is obviously seen that as the byte to append increase, time increases. For the one file case, especially after appending the 1000 bytes, increase rate is totally linear. The general trend of two files case is more linear than the one file case for all byte sizes. As a result, we can see that writing a file is pretty costly for the computer.

sfs_read() Stats

Bytes	Time in ms (1 file)	Time in ms (2 files)
100	218	462
1000	2333	4143
5000	11690	23821
10000	24575	46910

Table 3: sfs_read() stats

In all cases results of the read function is linear and time increases as the number of bytes read increases. Having more than one file is just duplicating the time by the file number. Reading is more costly than create operation but less costly than appending.