

Lab 1.1 Performance Execution

cat.c

./cat.o test.txt 0.00s user 0.00s system 66% cpu 0.006 total

bytes	cat.c 1	cat.c 2	cat.c 3	cat.c total
974	0.000751	0.000314	0.000261	0.001326
6,539	0.001522	0.001639	0.001812	0.004973
81,923	0.018328	0.017348	0.016386	0.052062
243,811	0.05028	0.050459	0.050336	0.151075

Table 1.1 Run time of 4 tests 3 times for cat.c

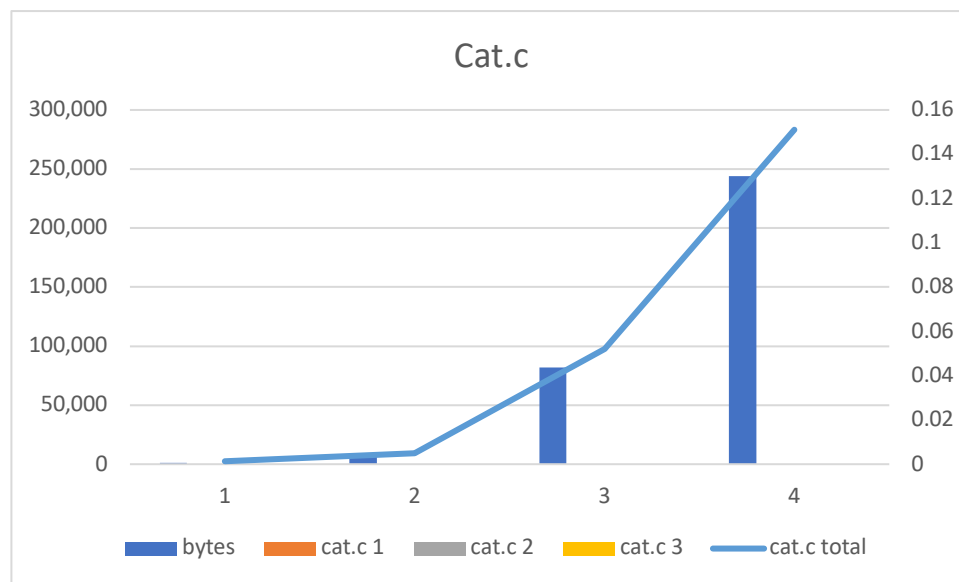


Chart 1.1 Shows the run time grows depending the size of the .txt file

mycat.c

./mycat.o test.txt 0.00s user 0.00 system 61% cpu 0.007 total

bytes	mycat.c 1	mycat.c 2	mycat.c 3	mycat.c total
974	0.000156	0.000197	0.000174	0.000527
6,539	0.000273	0.000424	0.000286	0.000983
81,923	0.000342	0.000216	0.000197	0.000755
243,811	0.000201	0.000177	0.000263	0.000641

Table 2.1 Run time of 4 tests 3 times for mycat.c

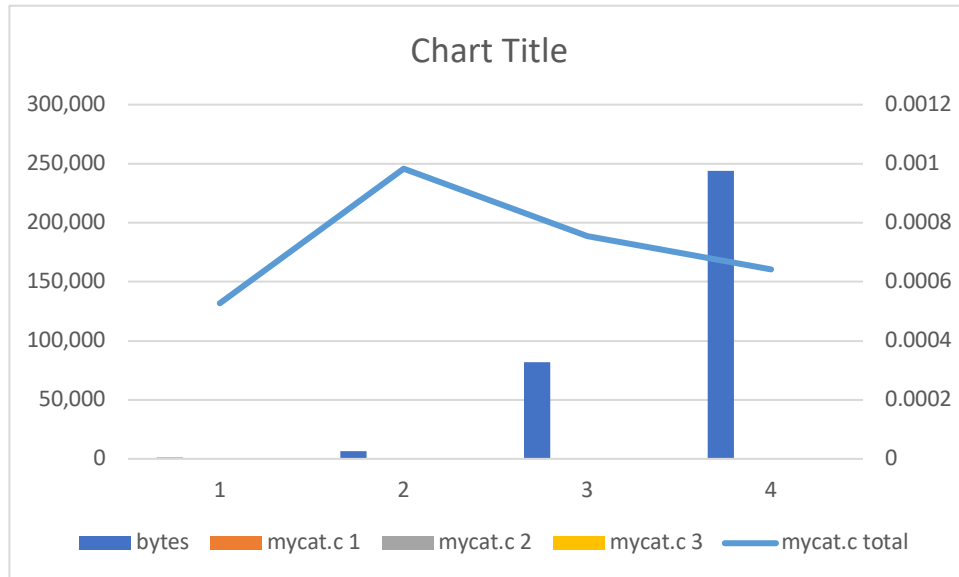


Chart 2.1 Shows the run time grows depending the size of the .txt file

Conclusion

When I add both times into a chart (and even before) it's obvious to see the big difference between the running times. For the cat.c each time you give a bigger file the run time increased. But for the mycat.c which uses *system calls*, it's a lot of time less. And it doesn't seem to get a lot bigger with a bigger input. In the next chart is really visible how the time is similar in small files, but with bigger files System calls are the best option.

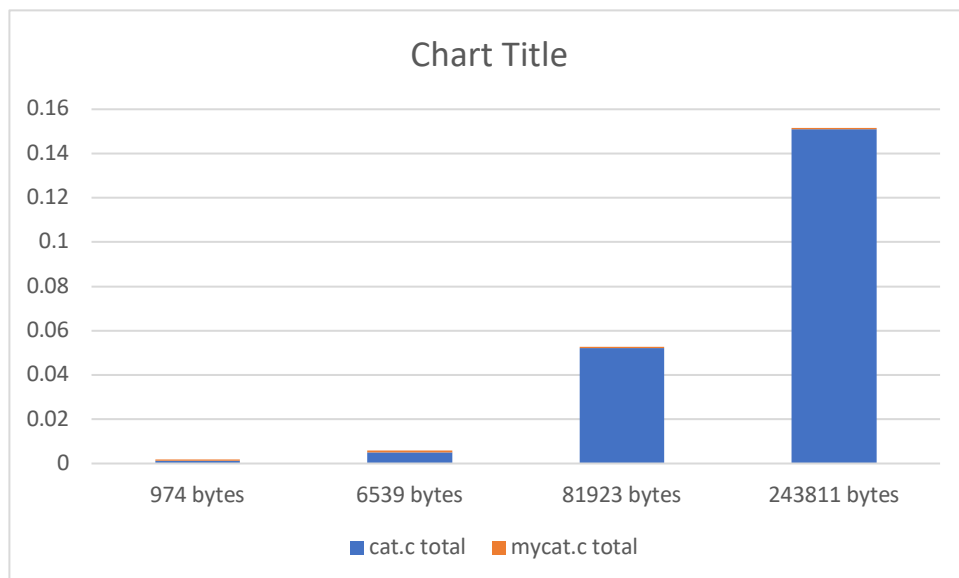


Chart 3.1 Comparison of time depending the size of the bytes in the test.txt file