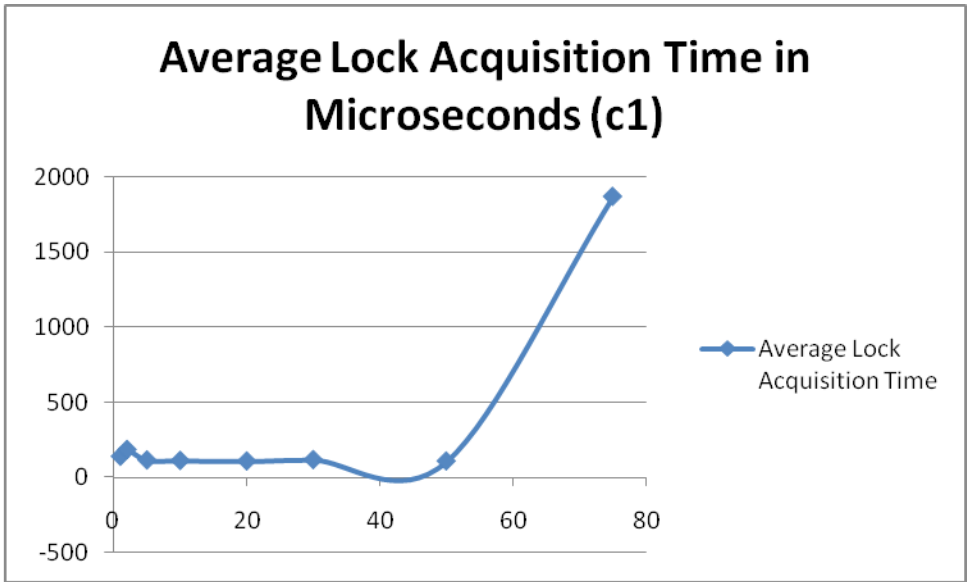
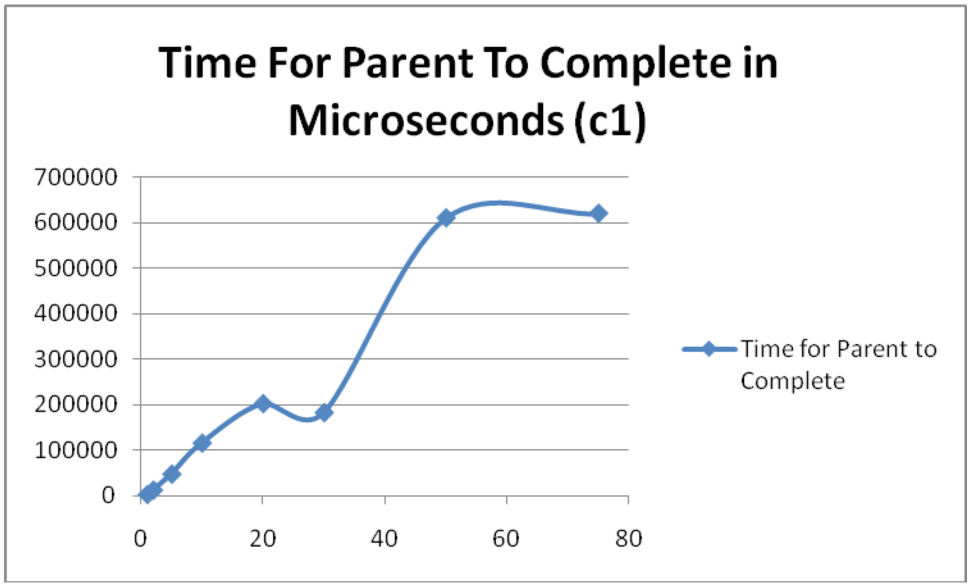
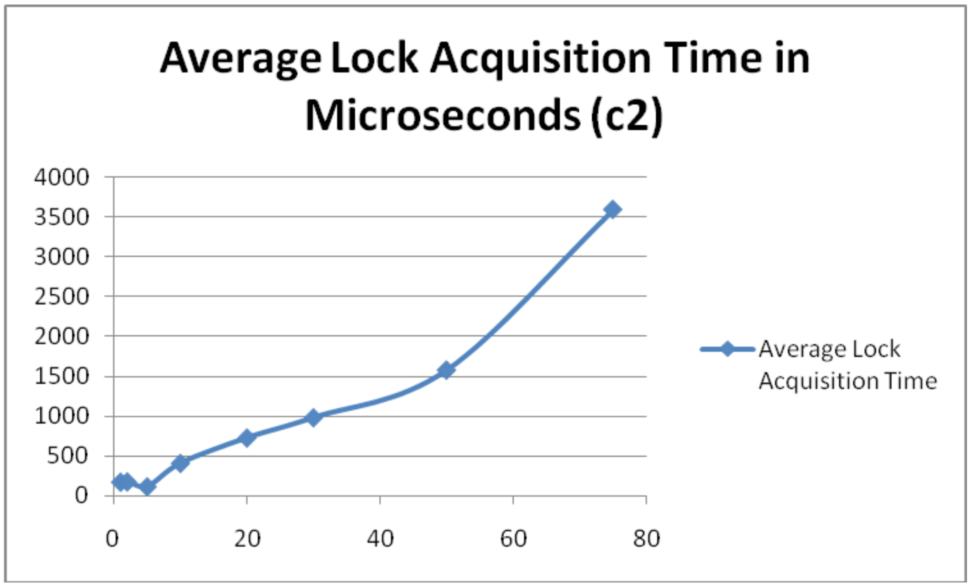
Joseph Schechter cs350 HW#3

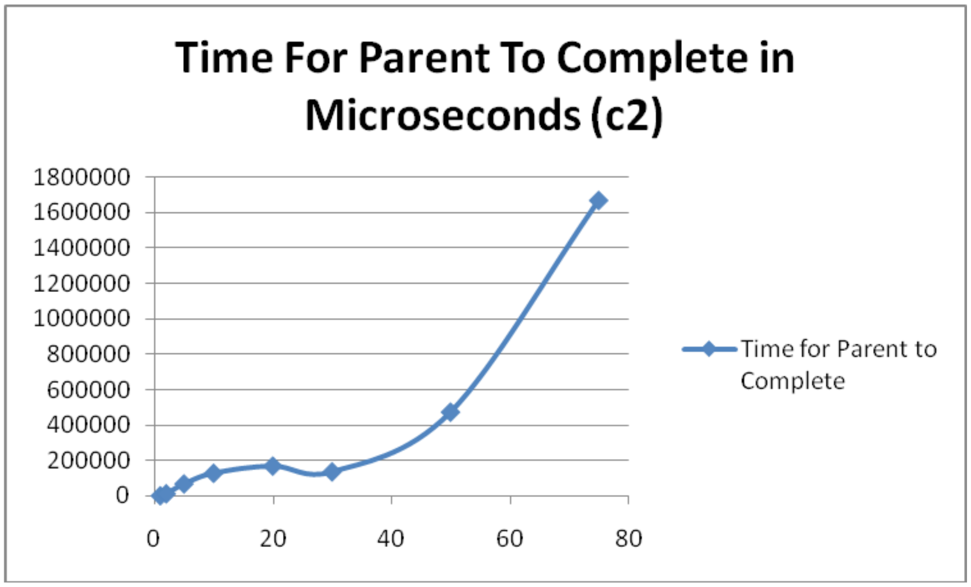




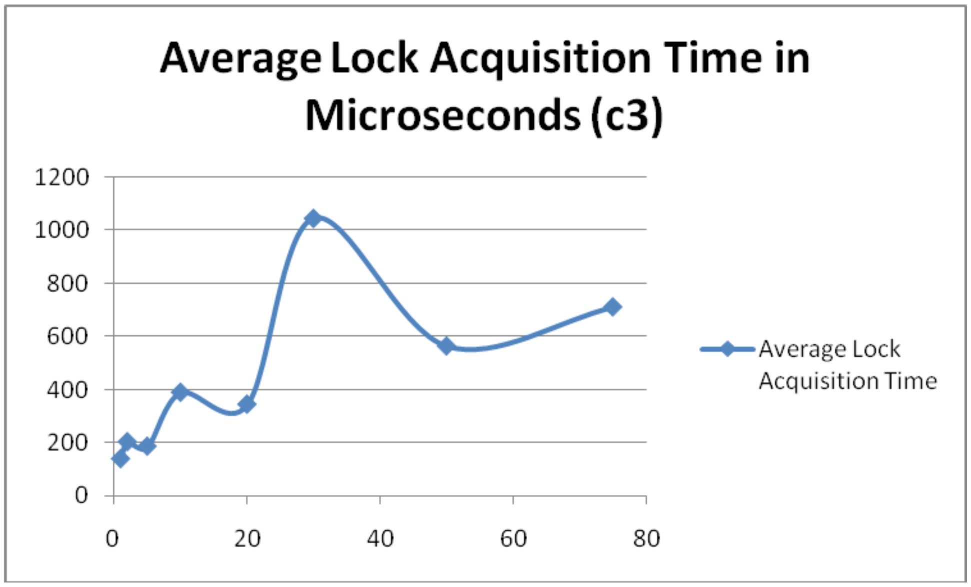
As the amount of processes increases, the time it takes to acquire the lock should go up since there is more competition for the process, but in this case, for numbers less than 50, it stayed roughly the same. The 75 process average wait, however, is much longer, more in line with what I expected.

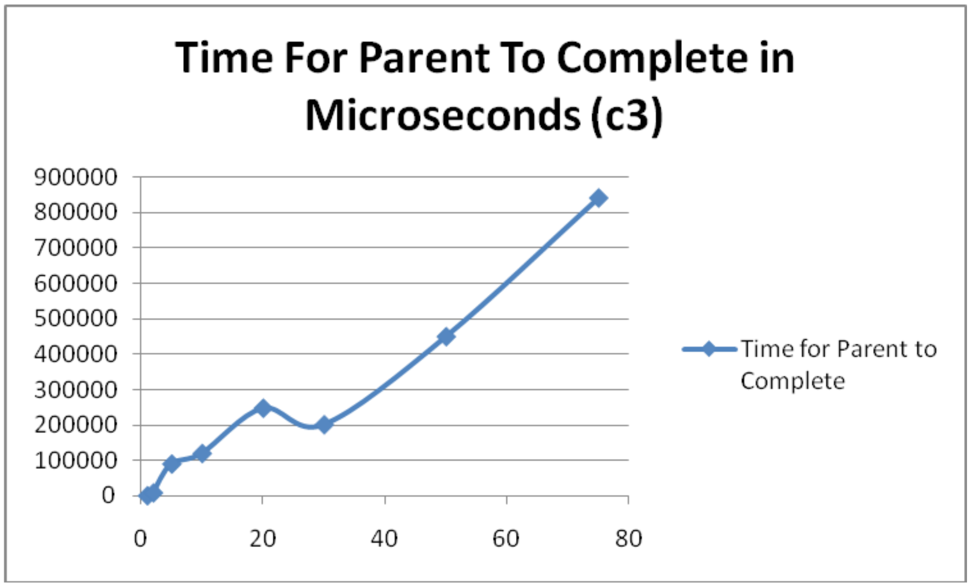
The amount of time it takes for the parent to complete should go up as more processes are added since each of the processes needs to run its course and the parent has to wait() for all of them. There is a big spike with large numbers of processes since they all have to run the first check to get out after the winner has already marked the memory.





These are my most normal values and how I expected these outputs to be. It does not seem to be a linear growth, more of an exponential growth. As more processes are added, there is an exponential increase in the amount of time it takes for the parent to complete as well as the average amount of time for the child to acquire the lock.





There is an outlier for 30 children. For some reason some processes just never got a chance to run until the end, so they skewed the graph. This happens sometimes, and is entirely up to how the cpu picks the next process after I do a sched_yield() and when the process uses all of it's allotted time or blocks. When I see dips in my graph for the amount of time it takes for the parent to complete, I assume it is because there was not a whole lot of waiting done for the children to do their work. They finished adding quickly and did not have a lot of context switching since it has the highest lock acquisition average time.