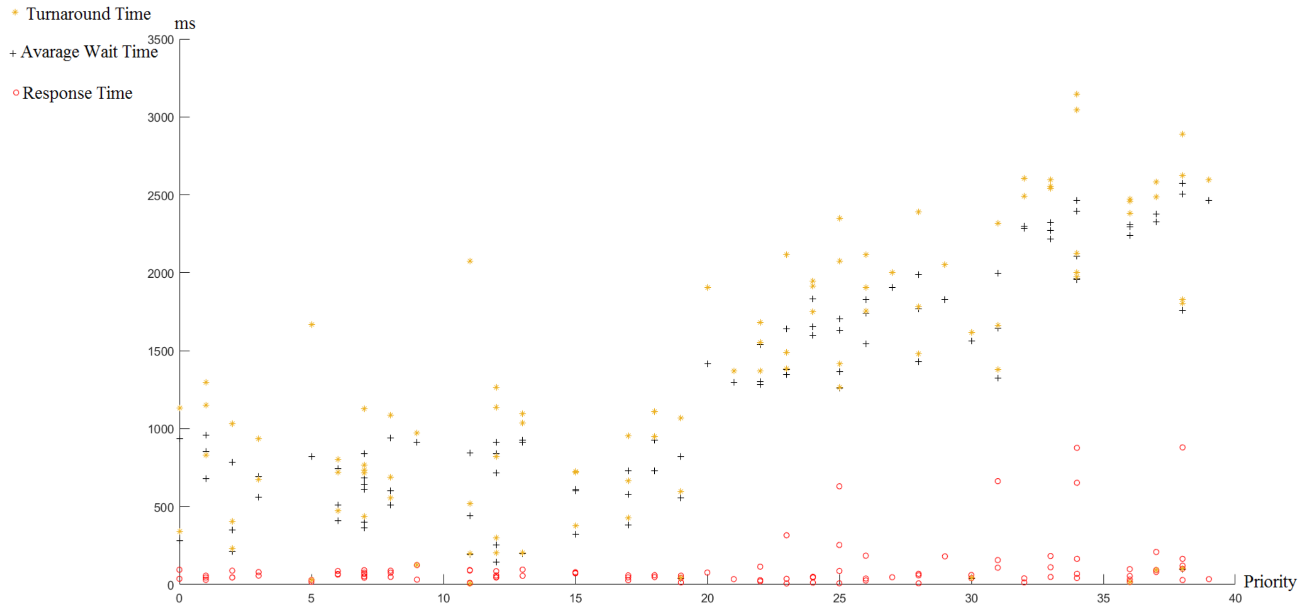
İbrahim Taha Aksu | 21401946

CS-342 Project #2 Analysis and Results

This report is to present the experiments held with the implemented process scheduler: CFS. CFS is an algorithm for selecting the best option for the task to run next. The original algorithm is based on a ready queue that is implemented as a red black tree. The benefit of this very data structure comes from inserting processes in O(log(n)) time even after many deletion operations. For this very project I used sorted linked list since the project aims to simulate the algorithm to try different workloads and efficiency is not the major concern here. I have used a time based simulation for my project rather than a event driven one.

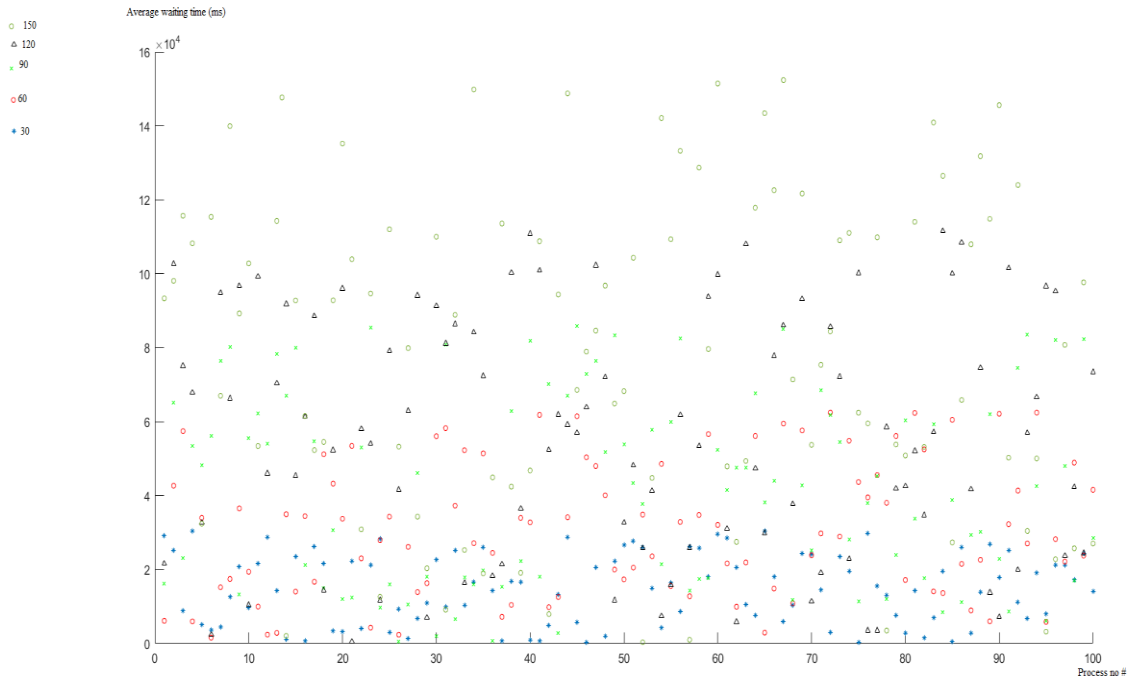
Now let us run some experiments and evaluate the success of the algorithm according to the results we get. The first execution will be done on a work load which is generated by the loadgen program with number of process (N) 100,average start time 50, average number of bursts 10, average cpu length 3, and average i/o length being equal to 20.



The CFS algorithm is designed in a way to let every process run in the cpu whenever they arrive. Since processes usually have longer i/o bursts then their cpus this desicion makes the algorithm much more efficient. By letting all processes use the cpu for small time intervals rather than a single one using it for long time blocks it gives every process ability to do their own i/o while the cpu is never wasting time. This very result could be driven out of the plot too. The red dots in the plot shows response times change with respect to the priority of the process. As the plot shows the response time of a process is not affected by the priority of the process and stays low for any range of priorities.

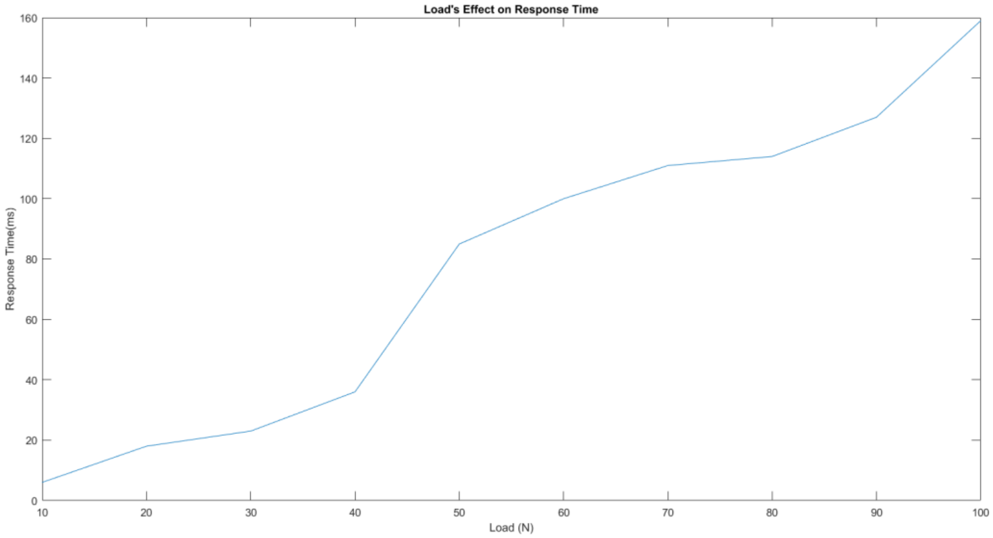
Let us now examine two other points in the plot. Average wait time and turnaround time. The average wait time of a process is defined as the average time that a process waits in ready queue after an i/o operation or when it is first created. Thus it very natural to expect its increase by the increase observed in priority. This is what we see in the plot too. Also, the turnaround time increases with priority increasing since the processes with a lower priority are given more chance to use the cpu and terminate earlier.

This experiment was to show the reader the characteristics of the CFS algorithm by doing an experiment. The next experiment will focus on doing the same thing by changing the generated file and analyzing the results.



For this very experiment there are 5 different generated files to be given as input to CFS algorithm. The parameters used to generate the files are totally same except the average cpu burst time. This experiment is held to see the average cpu times effect on average waiting time. As the cpu bursts increase while the i/o stands still one would be not wrong to expect greater waiting times since the cpu will be busy for longer periods. Just by checking the graph it is possible to see this very result. The shape and average cpu burst time relations are given on the upper left of the graph. As the average cpu burst increases the waiting time of the processes goes up. This experiment is good to make the statement that as the processes that a cpu handles becomes more cpu driven rather than i/o driven the waiting times occur in the ready queue increases more.

The last experiment is done to show the reader the effect of load of processes on the response time. To make the intuition it would be helpful to think of a cpu with only one load of a process. In this case the response time would be equal to 0 since the cpu has to handle only one process and not more. As the number of processes goes up the response time increases since the cpu becomes responsible of more processes. The experiment to check this very case was held again with various generated load files. These files were generated with exactly the same parameters except the number of processes (N). In the end these are fed to CFS algorithm and the average of the response times of processes was calculated and plotted as below:



As it was already argued above, as the load of the cpu increases the average response time to processes increase too. Thus it would not be wrong to conclude that the increase of the load of processes on the cpu has an increasing effect on the average response time to those processes.