Project Report: Shortest Remaining Time vs First Come First Serve

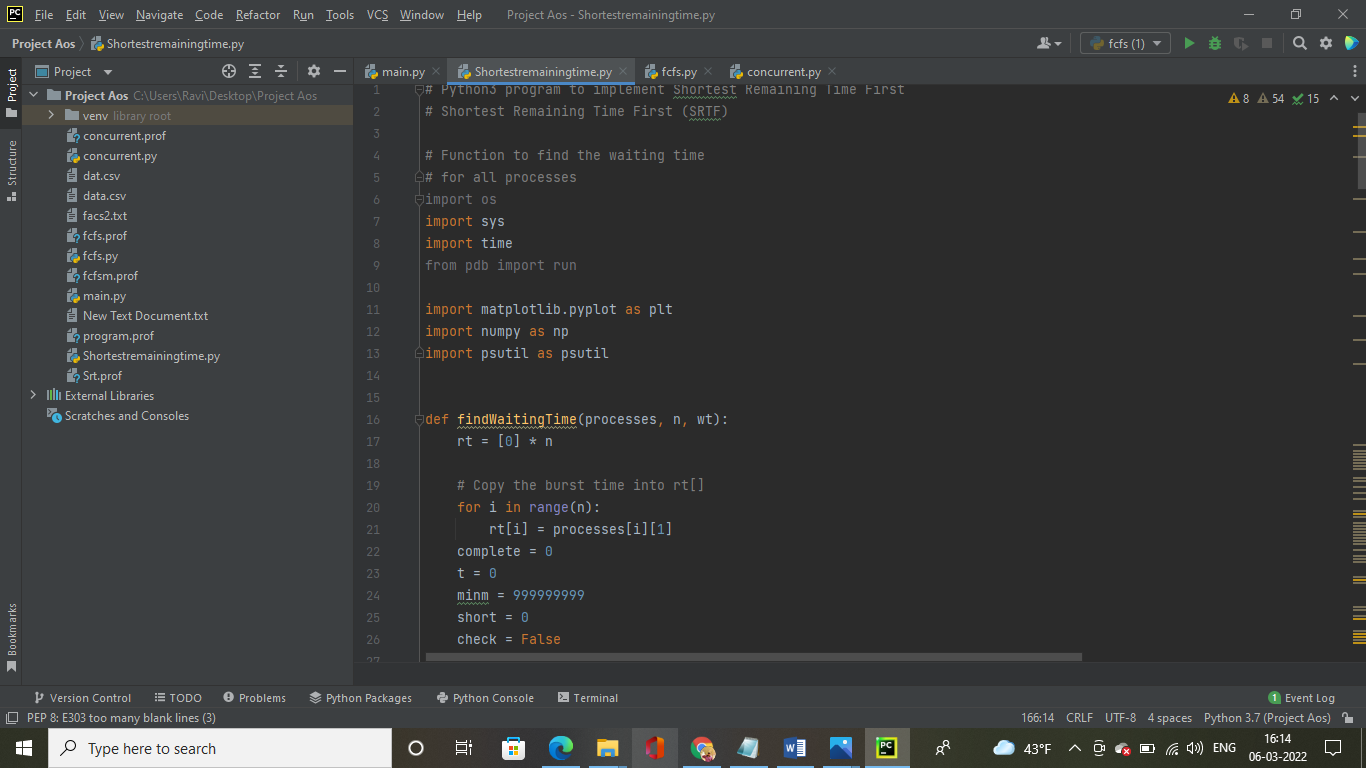
Ravi sankar Gogineni

R11788968

Shortest Remaining Time:

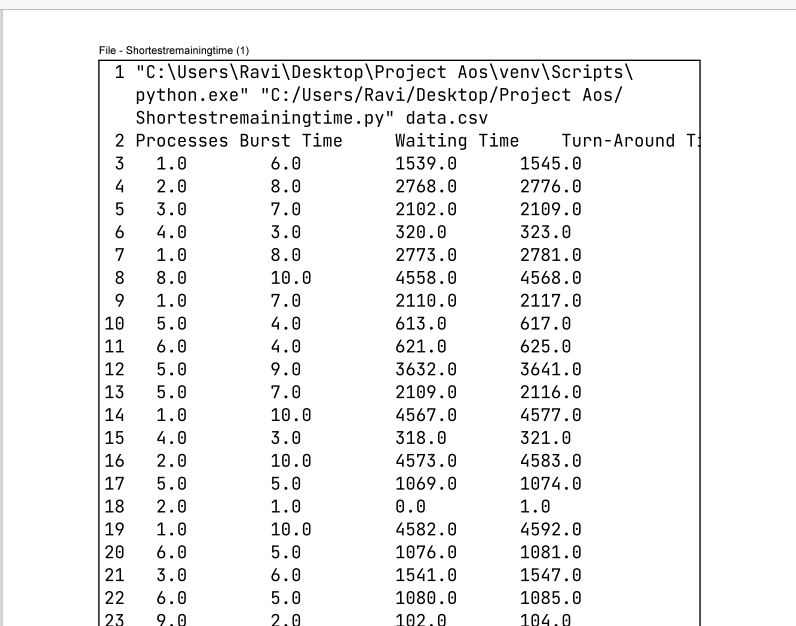
Shortest Remaining Time algorithm is Preemptive.Assumes run times are known. Scheduler picks the process whose remaining run time shortest. Newly arrived jobs are compared to current processes remaining time.

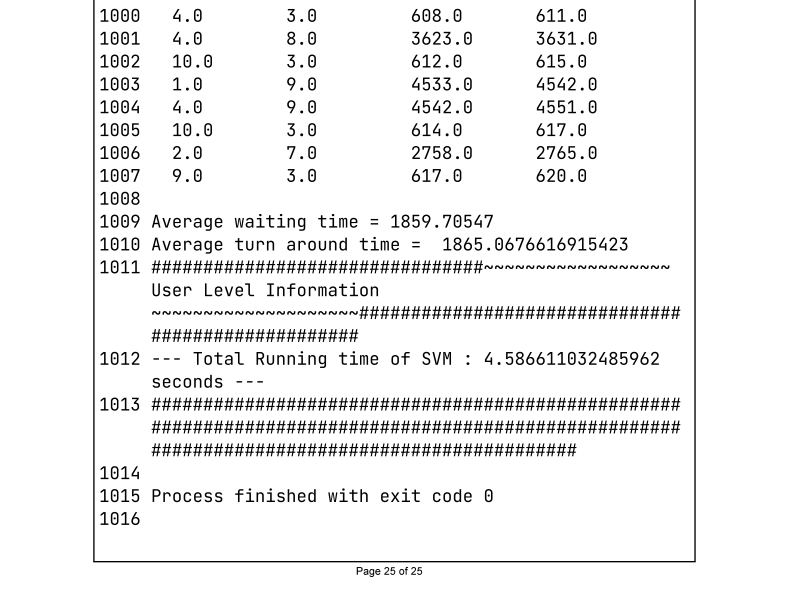
Pycharm id Environment:



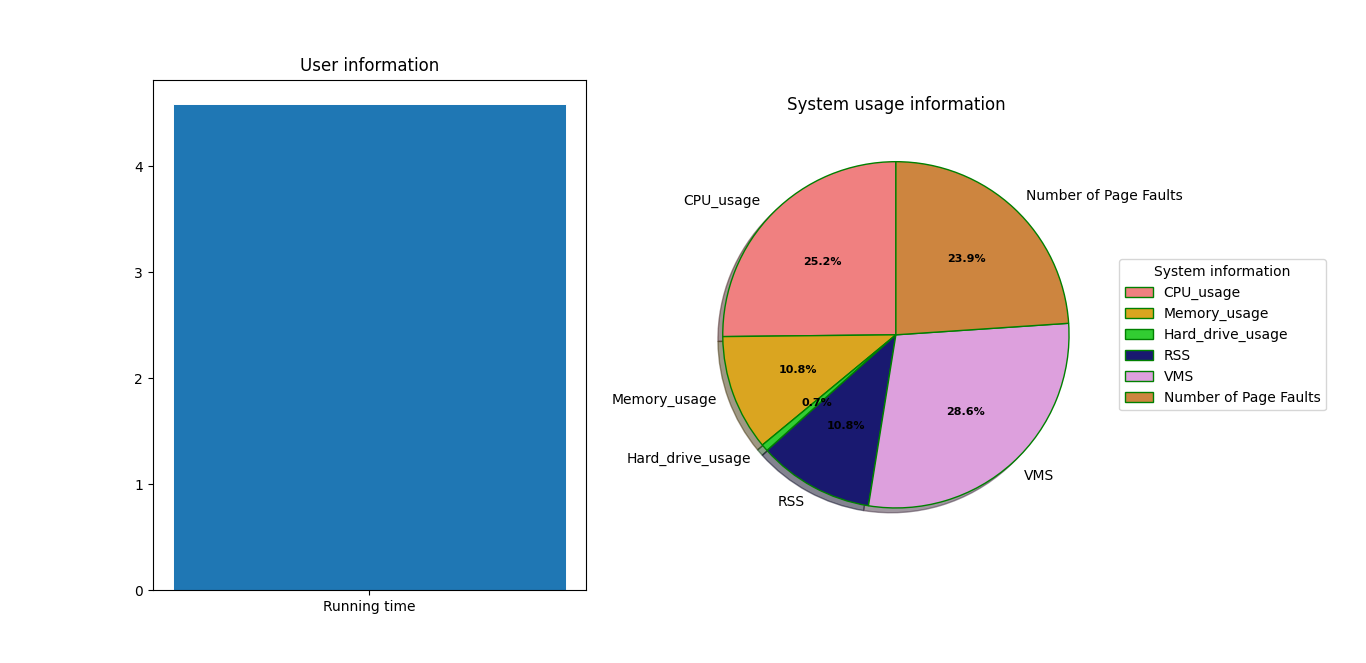
teh

The code is executed Successfully by showing the processes id, Burst time, waiting time,turn around



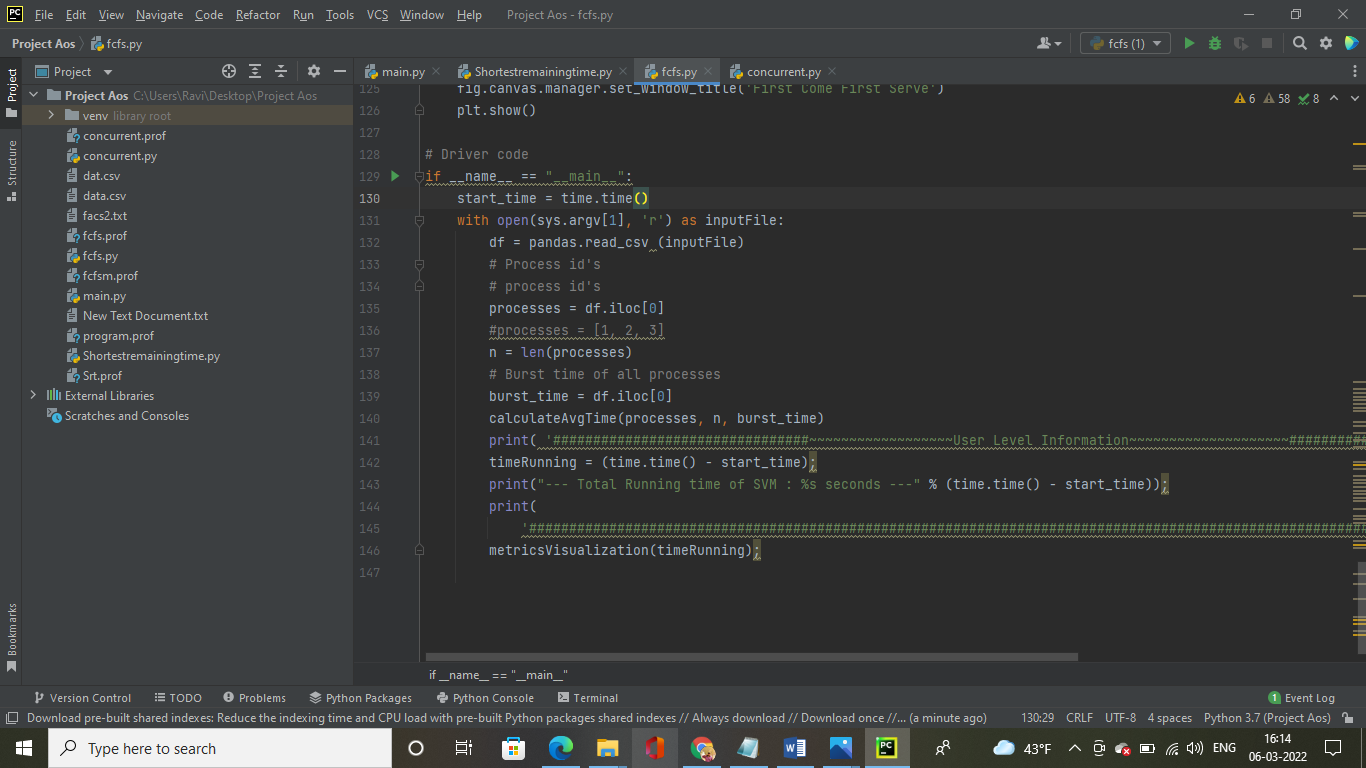


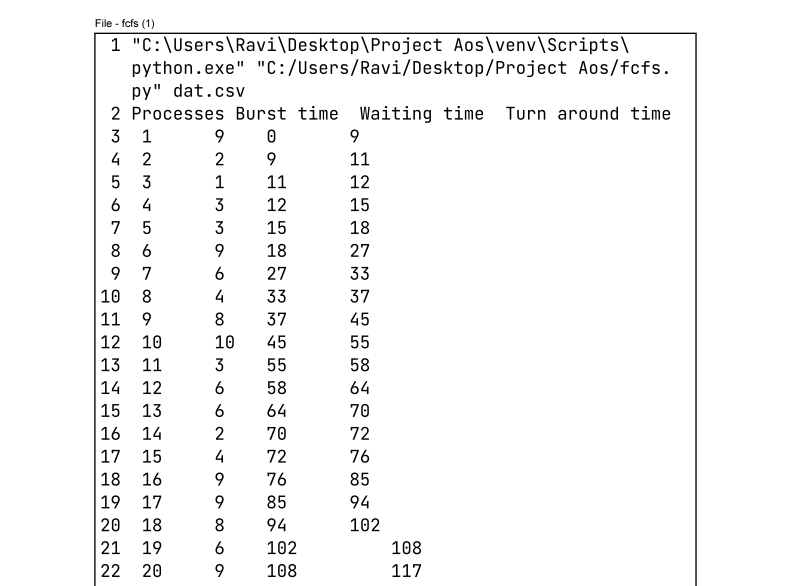
The user level information and System usage information is shown below;

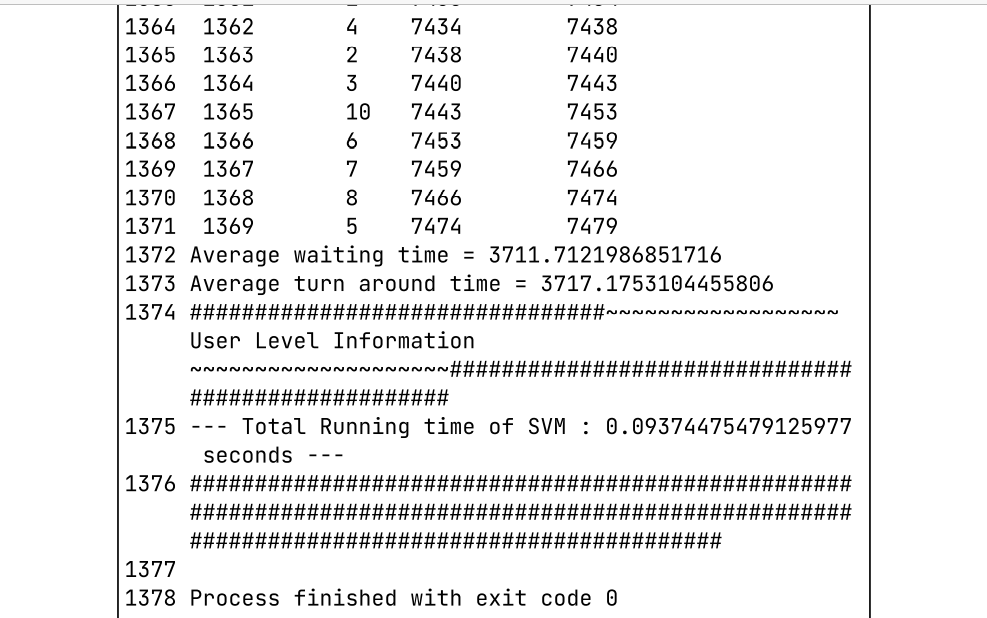


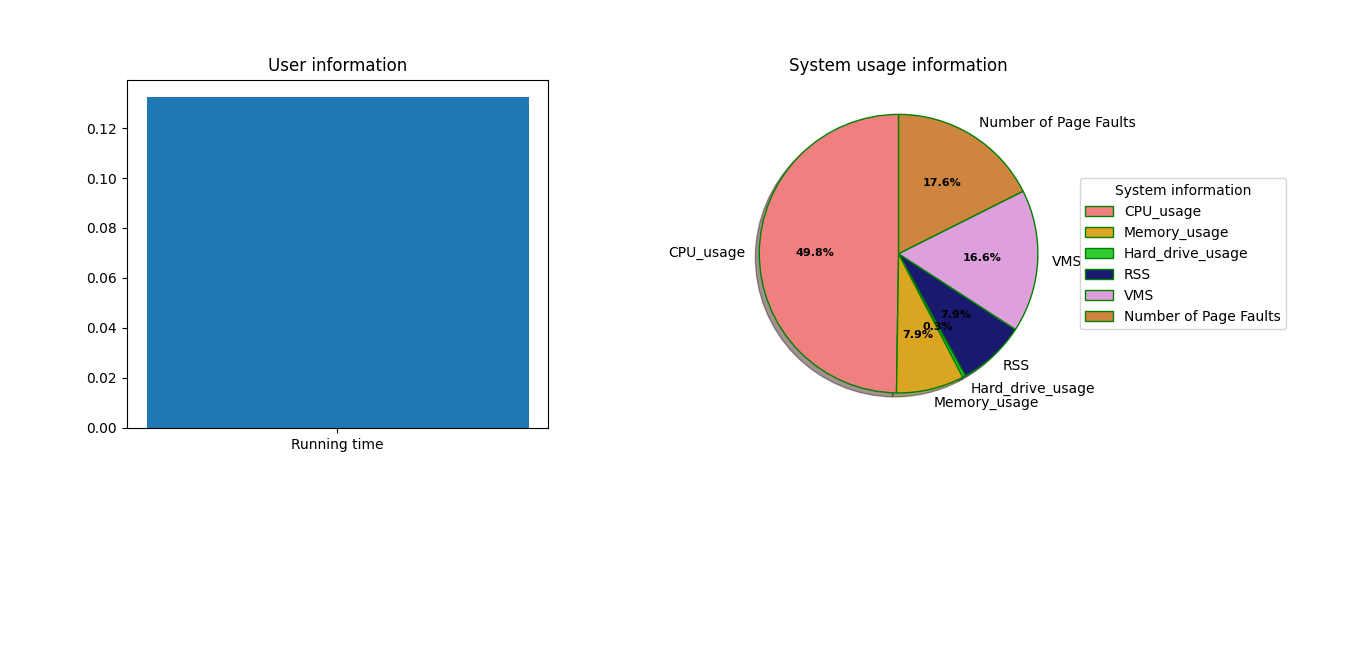
First Come First Serve:

First Come First Serve is the easiest and simplest CPU scheduling algorithm in the operating system that automatically executes processes in order of their arrival. In this type of algorithm, processes which request the CPU first get the CPU for their complete execution first. This method is poor in performance, and the general wait time is high. The simplest form of a CPU scheduling algorithm







The user level information and system usage informationEfficiency Comparison between the two algorithmsEfficiency Comparison between the two algorE

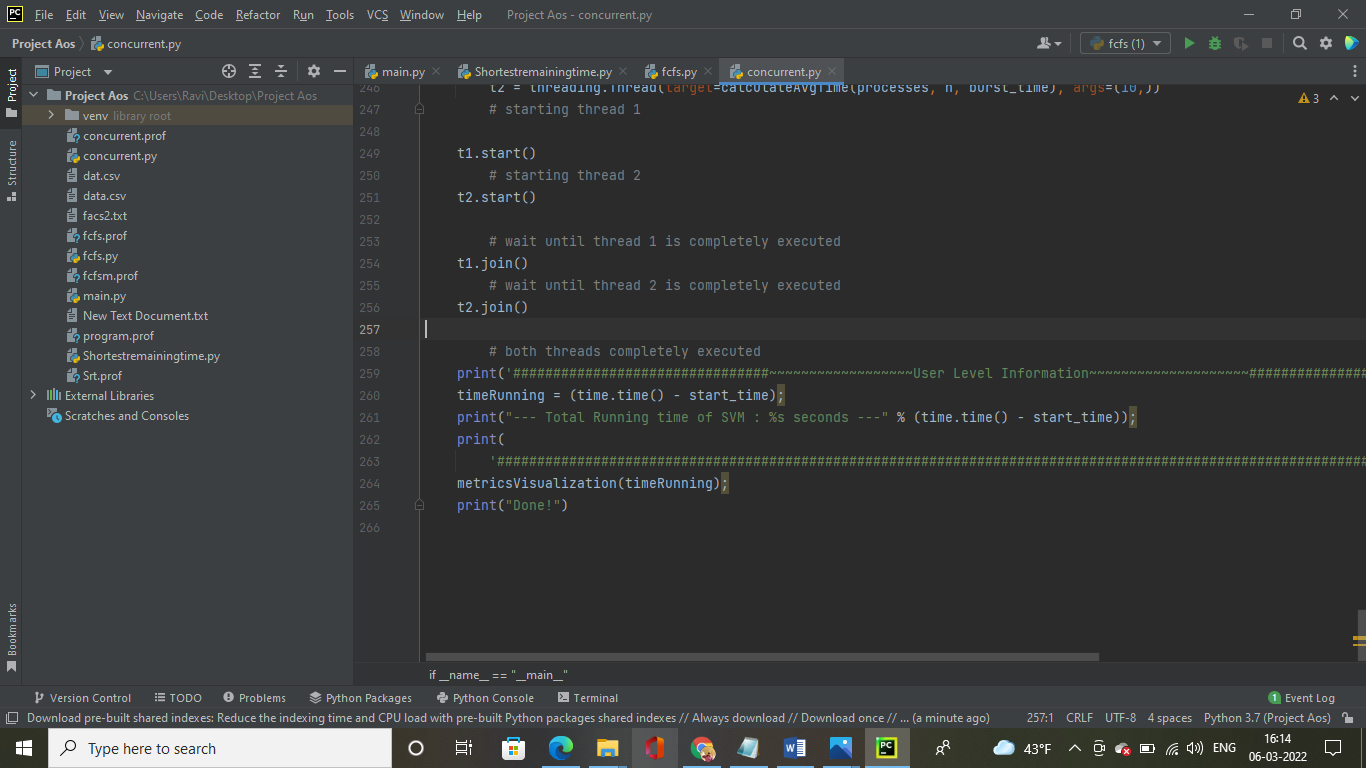
Efficiency comparison between the shortest Remaining Time vs First Come First Serve:

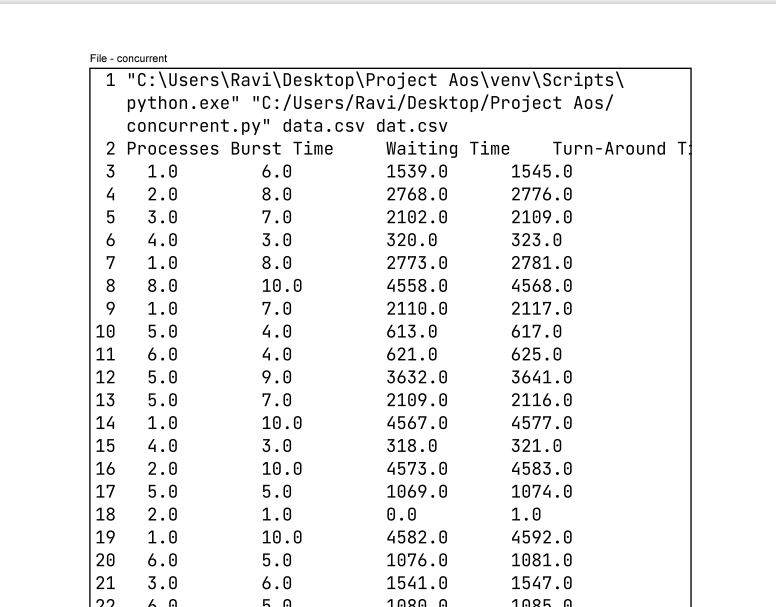
|  |  |  |
| --- | --- | --- |
| System Information: | Shortest Remaining Time | First Come First Serve |
| CPU usage | 25.2% | 49.8% |
| Memory usage | 10.8% | 7.9% |
| Hard drive usage | 0.7% | 0.3% |
| RSS: Resident set size | 10.8% | 7.9% |
| Virtual memory size | 28.6% | 16.6% |
| Number of page faults | 14229 | 20178 |
| User level information |  |  |
| Running Time | 4.6 sec | 0.15sec |

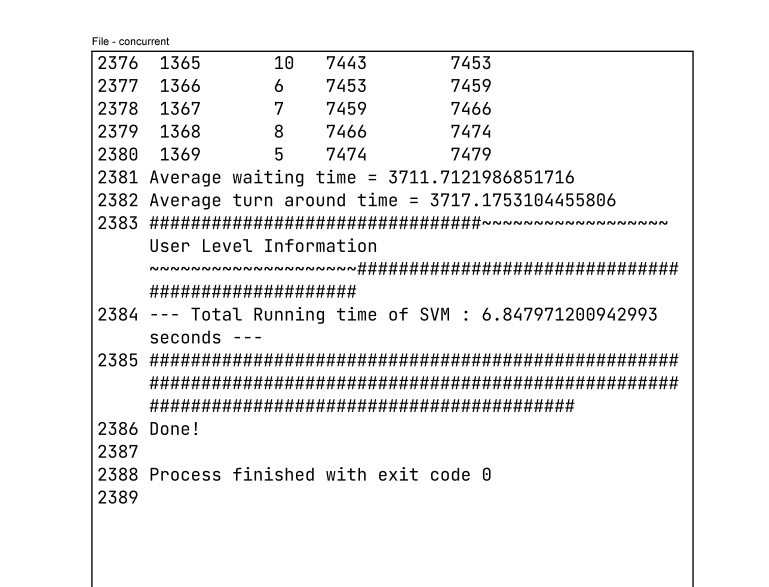
However, the two algorithms perform similar regarding memory and disk usage, with Shortest Remaining time algorithm slightly better. The two algorithms have different number of page faults. All in all, Shortest Remaining Time algorithm is more efficient than First Come First Serve algorithm. Results show that, as expected, the size of the input increases, more resources are consumed and more page fault occurs. Surprisingly, the number of page faults are also lower than First come first serve compare to Shortest remaining time. This might because memory allocator might be smart about allocating memory likely to be used at the same time on the same pages. Below shows monitored CPU and memory usage in different running conditions. Again, CPU and memory are more consumed as the input size increases. Compared to than First come First serve algorithm, Shortest remaining time algorithm performs better in general and becomes about the same when the size increases. Running concurrently confirms to take less CPU but more memory. Shortest Remaining Time algorithm does better than First come first serve algorithm in terms of both time and CPU usage.

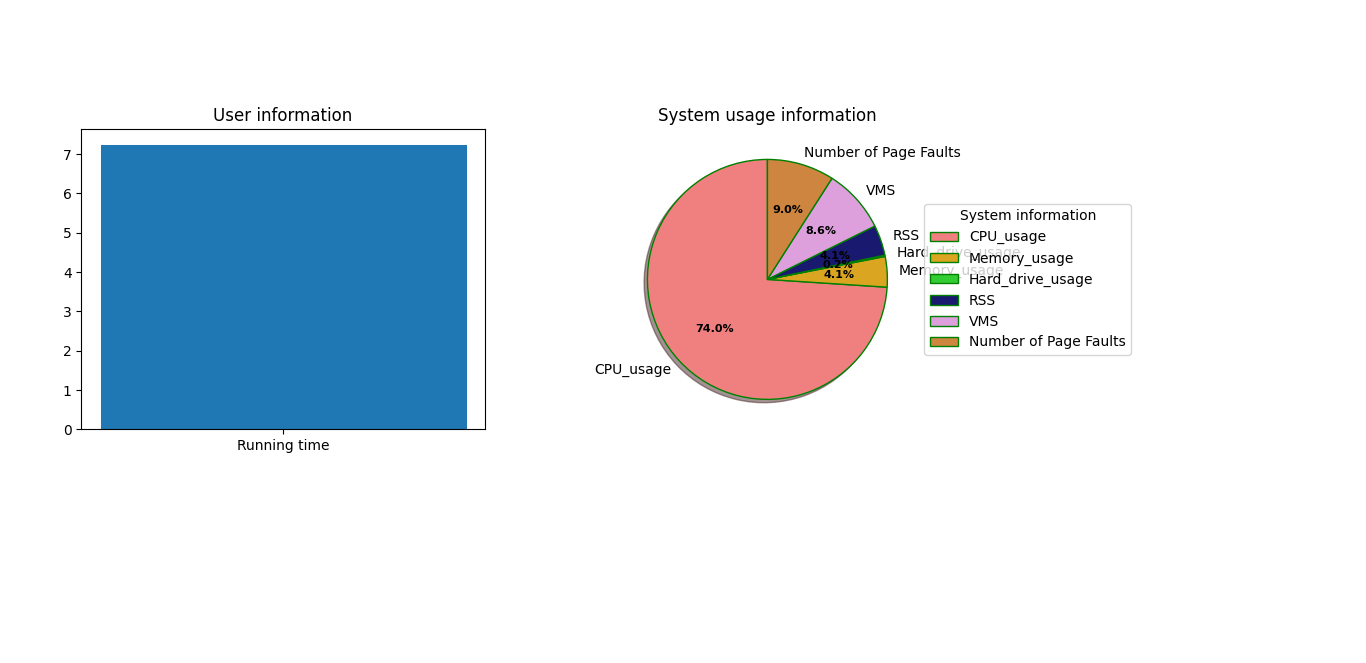
In compare of user level information First come first serve is better than Shortest Remaining Time Because First come first serve takes less time to execute than Shortest Remaining Time.

Concurrent of Shortest Remaining Time vs First Come First Serve:









Performance comparison between 3 running conditions

|  |  |  |  |
| --- | --- | --- | --- |
| System Information: | Shortest Remaining Time | First Come First Serve | Concurrent(SRT VS FCFS) |
| CPU usage | 25.2% | 49.8% | 74% |
| Memory usage | 10.8% | 7.9% | 4.1% |
| Hard drive usage | 0.7% | 0.3% | 0.2% |
| RSS: Resident set size | 10.8% | 7.9% | 4.1% |
| Virtual memory size | 28.6% | 16.6% | 8.6% |
| Number of page faults | 14229 | 20178 | 20303 |
| User level information |  |  |  |
| Running Time | 4.6 sec | 0.15sec | 7.2 sec |

Results show that, as expected, the size of the input increases, more resources are consumed and more page fault occurs. When running both programs concurrently, results showed that CPU and disk usage are consumed less than running an algorithm individually. Surprisingly, the number of page faults are also lower than the other two conditions. This might because memory allocator might be smart about allocating memory likely to be used at the same time on the same pages. If CPU usage is better with percentage is between the 40% to 90% in comparing between the three programs CPU usage is less in Shortest Remaining Time compared to other two. Page faults are the slowest possible thing that can happen to computer very good if you don't crash very very slow when have to page by this we can say that number of page faults are less in shortest Remaining Time. Virtual Memory Size for an algorithm this should be lower! because if it is higher than latency is higher, by comparing between the three the concurrent has less virtual memory. Memory Usage: for an algorithm this should be higher! if it is higher than latency to retrieve algorithm is lower! Memory usage is similar to the Virtual memory. Resident set size is how much memory a process is consuming in physical RAM, to load all of its pages after its execution. Concurrent has best RSS percentage compared to other two algorithms. In comparison between three algorithms in system user information Concurrent program persons worst only in CPU usage but in all other expects its better than Shortest Remaining Time and First come First serve. In user level information the first come first serve takes less time. But in comparing between all the three Shortest remaining Time is best in system user information and in user level information the First come first serve is best.

Efficiency between the three algorithms Shortest Remaining Time is Best.