Team Meatball  
Andres Chorro-Dahlgren  
Dora Do  
Jannette Pham-Le  
CS 149 – 02  
16 February 2015

**Assignment #2: Report**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **Average Turn Around Time** | **Average Wait Time** | **Average Response Time** | **Average Throughput** |
| **FCFS** | | 19.516472 | 14.334636 | 14.334636 | 17.475729 |
| **RR** | | 40.49018 | 34.95706 | 0.84167665 | 16.56051 |
| **SJF** | | 10.555662 | 5.8876715 | 5.8876724 | 19.23077 |
| **SRT** | | 9.755663 | 5.0876718 | 3.9876716 | 19.23077 |
| **HPF Non-preemptive** | **All Priorities** | 14.67557 | 9.246853 | 9.246853 | 16.82243 |
| **Priority 4** | 9.538634 | 3.3893995 | 3.3893988 | 9 |
| **Priority 3** | 16.675577 | 10.88209 | 10.882089 | 5 |
| **Priority 2** | 31.091827 | 27.118698 | 27.118698 | 3 |
| **Priority 1** | 1.6592023 | 0.17222607 | 0.17222613 | 1 |
| **HPF Preemptive** | **All Priorities** | 24.284647 | 18.630604 | 5.1306024 | 16.260162 |
| **Priority 4** | 9.983078 | 3.8338442 | 0.8338431 | 9 |
| **Priority 3** | 16.075577 | 10.282089 | 6.8820887 | 5 |
| **Priority 2** | 38.76163 | 33.30496 | 12.1049595 | 5 |
| **Priority 1** | 121.6592 | 120.172226 | 0.17222613 | 1 |

Although we thought Round Robin would have the largest average throughput within a period of 100 quanta, there was a tie between Shortest Job First and Shortest Remaining Time.

As expected, the shortest average turnaround time belonged to Shortest Remaining Time, with Shortest Job First coming in second by being within one quanta of each other.

The shortest average wait time belonged to Shortest Remaining Time as well, with Shortest Job First coming in second by being within a little less than one whole quanta of each other.

Finally, the algorithm with the shortest average response time was by far, Round Robin, which was not surprising since Round Robin essentially allows all the processes to run one burst at a time.

Overall, from our statistics we find that the Shortest Remaining Time algorithm is the better algorithm to handle multiple processes, since it has the average shortest turnaround time and shortest average wait time. Because all our processes were created with randomized arrival times, run times, and priority levels, we do not expect these algorithms to continue to have the best average turnaround time, response time, wait time, or throughput. For instance, on a separate run for Highest Priority Preemptive and Nonpreemptive, many of the processes with priority level 1 could not run, so we did not have averages for that queue. This resulted in lower and much better averages for those overall algorithms. However, because Shortest Remaining time holds the shortest time for two of these criteria, we can assume it is the “best” algorithm.