A Statistics-Based Approach to Determine Optimum Time Quantum for Round Robin Scheduling Algorithm

Adison Kleinsasser
Department of Computer Science
University of South Dakota
Adison.Kleinsasser@usd.edu

Vrinda Pareek
Department of Computer Science
University of South Dakota
Vrinda.Pareek@coyotes.usd.edu

Jesse Scott
Department of Computer Science
University of South Dakota
Jesse.Scott@coyotes.usd.edu

***Abstract*— In this paper, a statistics-based approach is implemented to assign optimum time quantum to the processes waiting in the ready queue and designing a self-adjusting algorithm that would change the quantum value dynamically by examining the workload and burst value of processes fed to the system. Our algorithm chooses the best quantum for the test set by calculating the optimum μ, using training sets composed of all combinations of four sets of burst times and four sets of arrival times. The algorithm chooses the optimum μ, considering waiting time, and fairness.**

This paper intends to consider waiting time, turnaround time and fairness for evaluation purposes and expects reduced waiting time and context switches.

Keywords—Statistics, Round Robin, Time quantum, Waiting time, Fairness

I. Introduction

Process scheduling is one of the most critical tasks performed by the operating system. The Round Robin (RR) algorithm is among the most common scheduling algorithms used. The processes waiting in the ready queue are assigned a time quanta/time slices in equal portions for which they seize on the processor in a circular order (aka cyclic executive). If the time quantum is extremely large, it causes a poor response time and behaves like the first-come, first-serve (FCFS) scheduling algorithm. On the other extreme, if the time quantum is too small, it causes multiple context switches and effects CPU efficiency.

II. Motivations

Round Robin is one of the fairest algorithms, as it allows for reduced waiting time, while preventing starvation of longer processes. The quantum time is one of the fundamental parts of round robin scheduling. Changing it can have a drastic effect on wait time, turnaround time, and execution time. Insufficient length of a quantum results in long turnaround times, while a quantum of too much length results in an RR algorithm that behaves like a FCFS algorithm, rather than a round robin algorithm. In this project, we prepare an algorithm to determine a fair, yet efficient, quantum time for the simple round robin scheduling algorithm.

With each new set of processes, the burst times and arrival times may change. Because of this, the RR scheduling algorithm must be able to change its optimum quantum time along with each new set. This requires a method of allowing the code to choose the quantum without the manual aid of a human operator. This is of great interest in the Information Technology field.

III. Challenges

Our main goal is to enable the scheduler to choose the quantum based on the burst time of the processes in the introduced testing set. The main challenge, then, is producing an algorithm that will identify the proximity of a set of processes to each of the training sets, in order to select the most-similar training set, and therefore determine the quantum according to the best quantum for the training set matched with the testing set.

One of the challenges we had to meet was determining the proper measurement of performance. There are many different and distinct measurements for a scheduler, including waiting time, turnaround time, response time, and throughput. While all of these can be measured and have an effect on the algorithm's use and performance, ultimately, the ML algorithm must choose a quantum based primarily on one parameter.

IV. Proposed Solution

An optimized version of the Round Robin algorithm has been introduced. The algorithm is modified to choose an optimal time quantum using a statistics-based algorithm. Sixteen work sets are taken into consideration while preparing the training set. The training set is made of four distinct sets of burst times, combined with four distinct sets of arrival times.

Burst times:

1. 2-10
2. 100-110
3. 100-110, with 2-10 mod 3
4. 2-100

Allowed space between arrival times:

1. 1-10
2. 1-100
3. 1-50
4. 1-5

This produced sixteen total combinations (hereafter known as combos).

Our first step in writing the algorithm to choose the optimum quantum for a testing set was to find the optimum time quantum for each of the combos. To this effect, we first calculated the waiting time for each of the sixteen combos, for each potential quantum.

After this, the fairness of each quantum was calculated for each of the combos. Fairness was calculated as follows:

First, using the set of waiting times for each process using the quantum, the largest difference between consecutive waiting times was found. This difference was subtracted from the average waiting time, and the absolute value of this number was considered the quantum's fairness score for the combo.

difference = largest difference between consecutive wait times
fairness = | average wait time - difference |

Fig. 1 through fig. 16 depict the plot of average waiting time and fairness for each of the sixteen work sets (combos) under consideration.

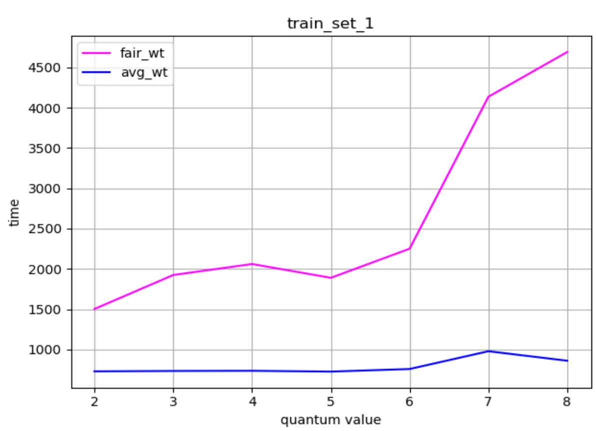

Fig. 1. Plot of average waiting time and fairness score for train set 1: a combination of burst time set 1 and arrival time set 1.

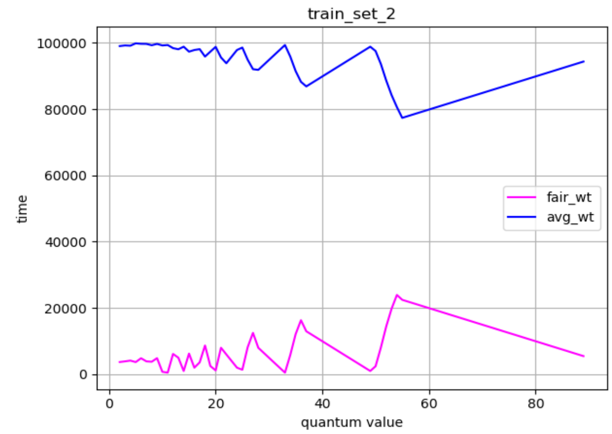

Fig. 2. Plot of average waiting time and fairness score for train set 2: a combination of burst time set 2 and arrival time set 1

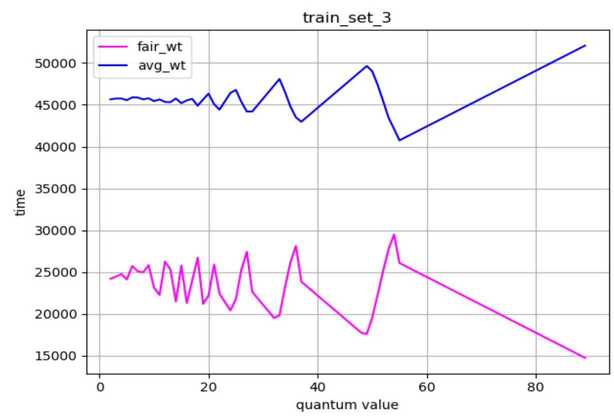

Fig. 3. Plot of average waiting time and fairness score for train set 3: a combination of burst time set 3 and arrival time set 1

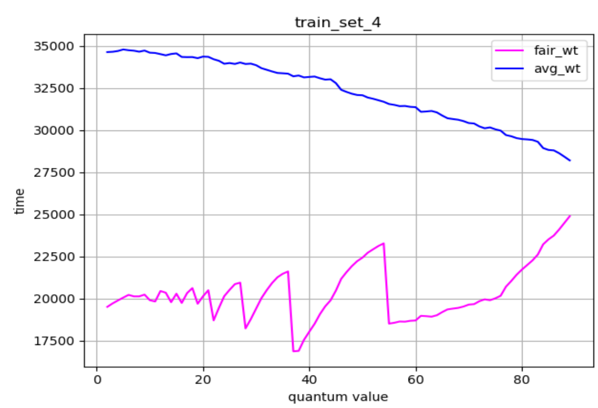

Fig. 4. Plot of average waiting time and fairness score for train set 4: a combination of burst time set 4 and arrival time set 1

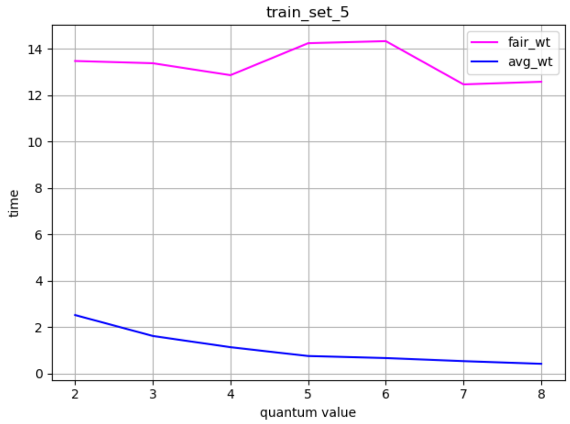

Fig. 5. Plot of average waiting time and fairness score for train set 5: a combination of burst time set 1 and arrival time set 2

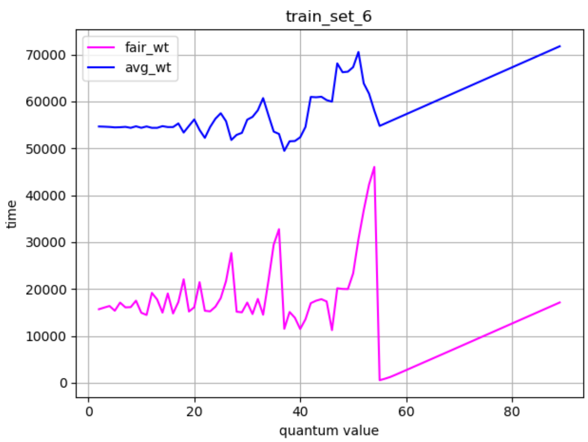

Fig. 6. Plot of average waiting time and fairness score for train set 6: a combination of burst time set 2 and arrival time set 2

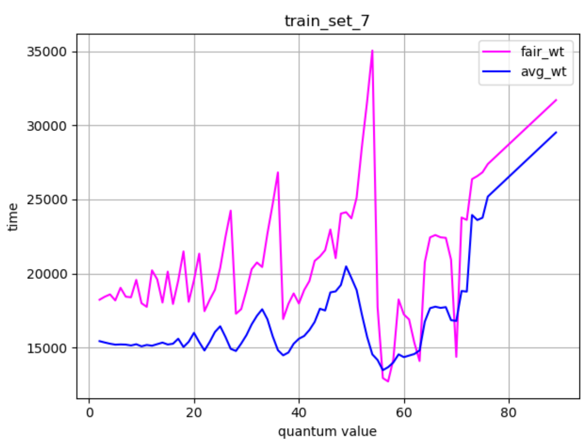

Fig. 7. Plot of average waiting time and fairness score for train set 7: a combination of burst time set 3 and arrival time set 2

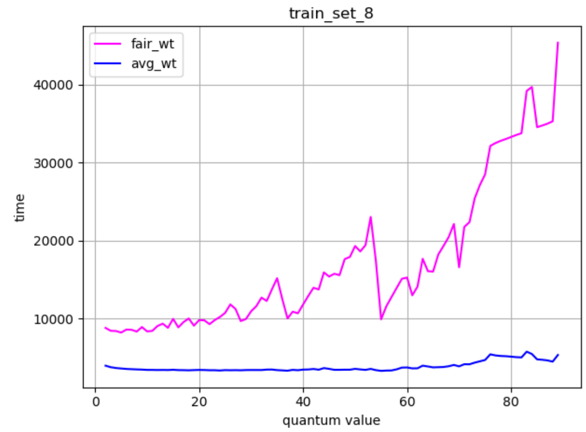

Fig. 8. Plot of average waiting time and fairness score for train set 8: a combination of burst time set 4 and arrival time set 2

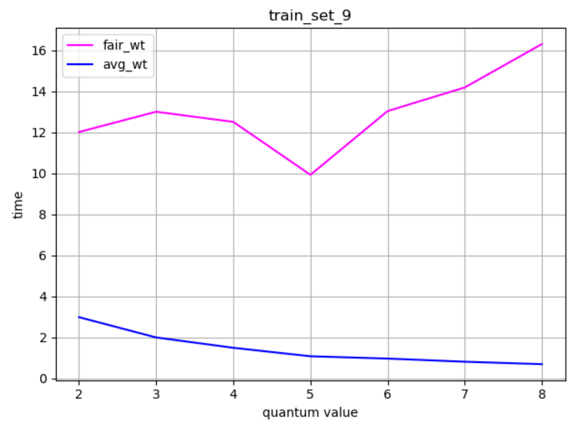

Fig. 9. Plot of average waiting time and fairness score for train set 9: a combination of burst time set 1 and arrival time set 3

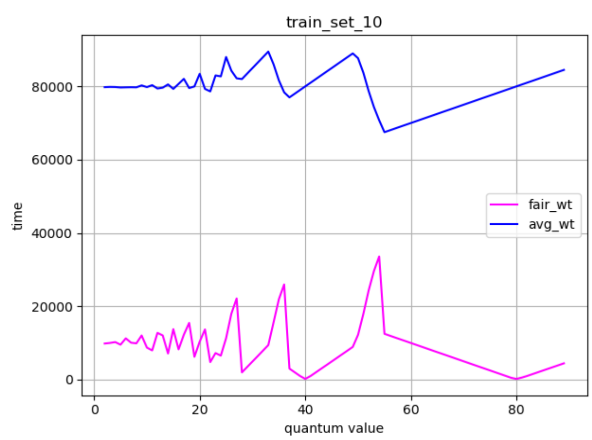

Fig. 10. Plot of average waiting time and fairness score for train set 10: a combination of burst time set 2 and arrival time set 3

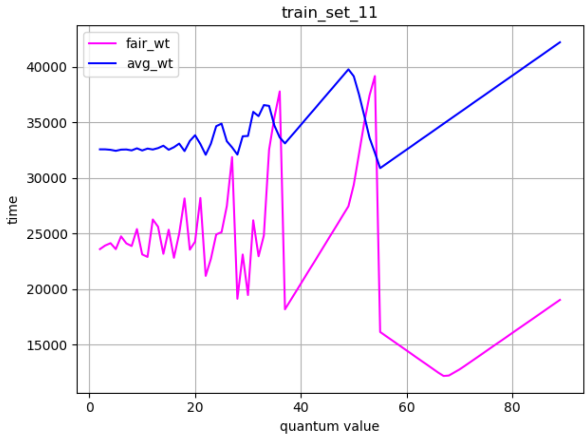

Fig. 11. Plot of average waiting time and fairness score for train set 11: a combination of burst time set 3 and arrival time set 3

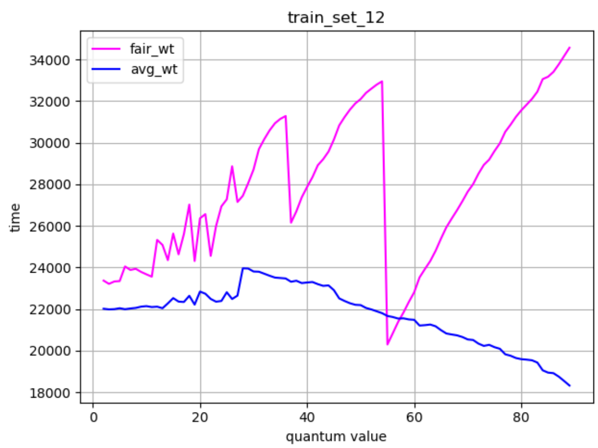

Fig. 12. Plot of average waiting time and fairness score for train set 12: a combination of burst time set 4 and arrival time set 3

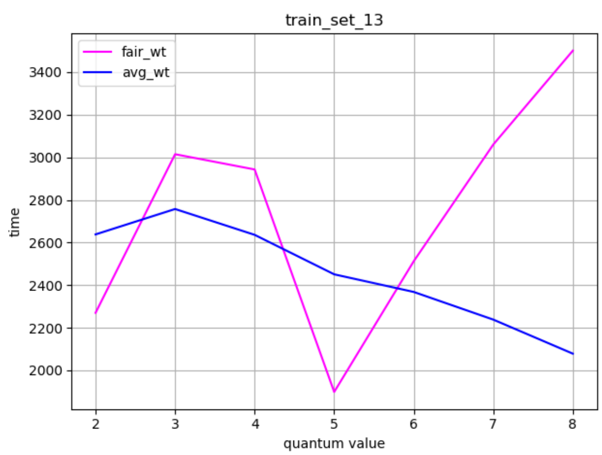

Fig. 13. Plot of average waiting time and fairness score for train set 13: a combination of burst time set 1 and arrival time set 4

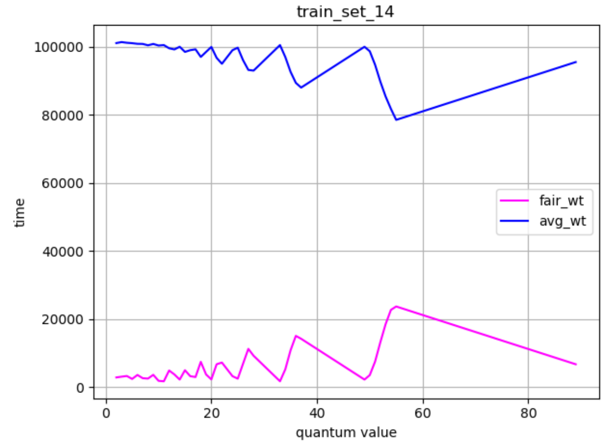

Fig. 14. Plot of average waiting time and fairness score for train set 14: a combination of burst time set 2 and arrival time set 4

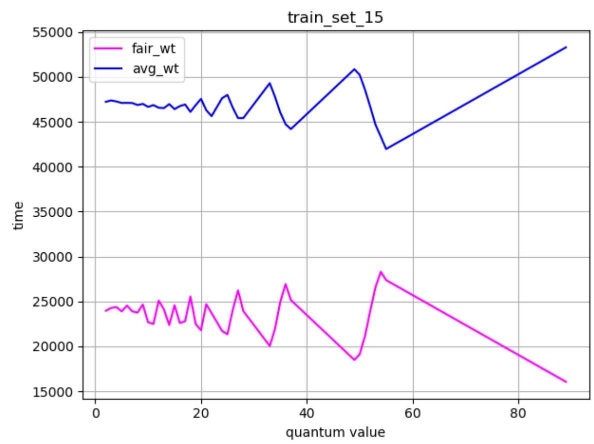

Fig. 15. Plot of average waiting time and fairness score for train set 15: a combination of burst time set 3 and arrival time set 4

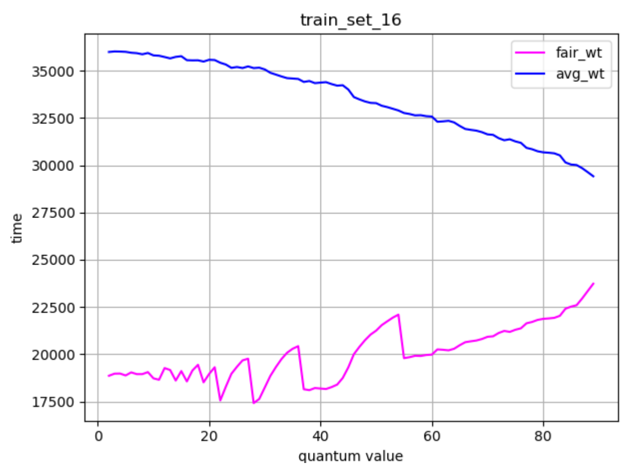

Fig. 16. Plot of average waiting time and fairness score for train set 16: a combination of burst time set 4 and arrival time set 4

After the fairness of each potential quantum was calculated for a given combo, the combo's optimum quantum was assigned as the quantum with the lowest fairness score.

As shown in Table 1, an optimal value of time quantum has been associated with the training set for individual work set.

TABLE I.

Train set	***Average turnaround time (μs)***	***Average waiting time (μs)***	***Fairness score***	***Quantum***
train_set_1	735.469	729.555	1503.445	2.0
train_set_2	99430.045	99325.135	461.865	33.0
train_set_3	52138.965	52067.017	14757.983	89.0
train_set_4	33237.439	33182.288	16863.712	37.0
train_set_5	6.440	0.526	12.474	7.0
train_set_6	54899.117	54794.207	552.793	55.0
train_set_7	13747.722	13675.774	12712.226	57.0
train_set_8	3638.740	3583.589	8198.411	5.0
train_set_9	6.983	1.069	9.931	5.0
train_set_10	80074.207	79969.297	208.703	80.0
train_set_11	34947.627	34875.679	12182.321	67.0
train_set_12	21728.950	21673.799	20301.201	55.0
train_set_13	2456.579	2450.665	1898.335	5.0
train_set_14	100650.831	100545.921	1752.079	33.0
train_set_15	53359.751	53287.803	16048.197	89.0
train_set_16	35190.817	35135.666	17419.334	28.0

Table 1. Time quantum allocation for training set

The second step to the algorithm was to pair the testing set to the combo most similar to it. For this, the mean of the burst times was calculated for each set of burst times and for the testing set. The difference between the means of the testing set and each of the training sets were calculated.

Then, the same was done for the arrival times. The mean of arrival times for each training set and the testing set were calculated, and the difference between the mean of the testing set to each of the training sets was calculated.

In order to provide visualization of the burst times and arrival times of our test data as compared to our training burst and arrival times, Figure 17 shows a box plot of the four sets of burst times used in the training data, as well as the burst times of the test data.

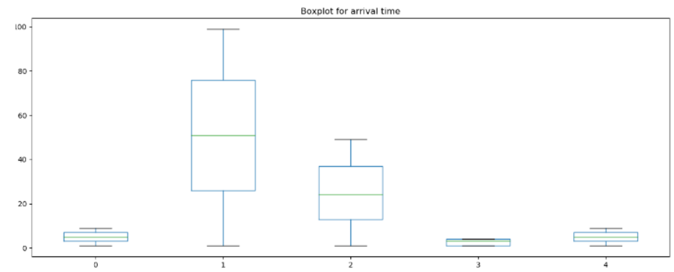

Fig. 17 Box plot to depict the distribution of burst times of 4 training sets (0-3) with test set (4) under consideration.

Likewise, Figure 18 shows the four sets of arrival times for the training data, and for the testing data.

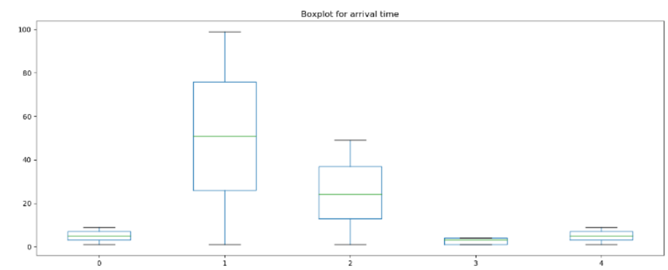

Fig. 18 Box plot to depict the distribution of arrival times of 4 training sets (0-3) with test set (4) under consideration.

The testing set was then matched with the closest training burst time set and training arrival time set based on smallest difference calculated. This produced a match between the testing set and the most-similar combo.

Once the testing set was matched with a combo (a training arrival time and burst time), the optimum quantum for the matched combo was then assigned to the testing set.

V. CONCLUSION

In this study, we have produced a statistics-based algorithm for determining the optimum time quantum for a testing set. This determination is based on its similarity to one of sixteen combinations of four sets of arrival times, and four sets of burst times.

VI. FUTURE WORK

Future work on this project includes forming k-means clusters from each of the combos. A challenge encountered was difficulty producing k-means clusters for use in the machine learning algorithm. Originally, we attempted to show four k-means clusters based on the burst times of each of the training sets. However, this did not produce four distinct clusters, in fact, much of some of the sets overlapped considerably. This was not done taking arrival time into consideration, and future work would be defining a method of calculating k-means clustering involving both burst time and arrival time.

A second future addition to this study is the addition of more training sets, both sets of burst times and sets of arrival times. In choosing sets of burst times, we attempted to utilize a range of burst time sets that would provide as broad a scope as possible with the limited number of sets used, however, the current sets may not accurately represent all potential sets of data in the real world. For example, one set of burst times includes both high and low burst times but is biased toward high burst times (100-110, with 2-10 mod 3). Another set containing the reverse (2-10 with 100-110 mod 3) could be used as well. Sets of real-world burst times could also be used to replace the artificial sets.

In choosing sets of arrival times, the sets were produced in increasing allowance of time gaps between arriving processes. Future work would include increasing the allowed time gaps between processes. Like real-world burst times, sets of real-world arrival times could also be added.

A third future addition would be the inclusion of execution time as another parameter in finding the optimum quantum time. This would produce a quantum that is not only fair, but also more efficient.

REFERENCES

[1] A. Noon, A. Kalakech and S. Kadry, "A New Round Robin Based Scheduling Algorithm for Operating Systems: Dynamic Quantum Using the Mean Average", International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 1, May 2011.

[2] R. Mohanty, H. S. Behera, K. Patwari and M. R. Das, "Design and Performance Evaluation of a New Proposed Shortest Remaining Burst Round Robin (SRBRR) Scheduling Algorithm", In Proceedings of International Symposium on Computer Engineering & Technology (ISCET), Vol. 17, 2010.

[3] R. Mohanty, M. Das, M. L. Prasanna, Sudhashree, "Design and Performance Evaluation of a New Proposed Fittest Job First Dynamic Round Robin (FJFDRR) Scheduling Algorithm", International Journal of Computer Information Systems, Vol. 2, No. 2, 20