Convoy Effect Elimination in FCFS Scheduling

Sambath M., K. Padmaveni, Linda Joseph, Ravi S., J. Thangakumar, D. John Aravindhar

Abstract: One of the important activities of operating systems is process scheduling. There are many algorithms available for scheduling like First Come First Served, Shortest Job First, Priority Scheduling and Round Robin. The fundamental algorithm is First Come First Served. It has some drawback of convoy effect. Convoy effect occurs when the small processes are waiting for lengthy process to complete. In this paper novel method is proposed to reduce convoy effect and to make the Scheduling optimal which reduces average waiting time and turnaround time.

Keywords: convoy effect, scheduler, throughput, turnaround time.

I. INTRODUCTION

CPU Scheduling is the process of assigning the process to the processor for execution. In memory multiple programs are stored and ready for execution. The performance of the processor depends on how the programs are dispatched to the processor for execution. In First Come First Served (FCFS) algorithm the processes are dispatched in the order of arrival to the processor. Scheduling is done by the scheduler[1]. The scheduling algorithm reflects the overall performance of the processor. There are three types of schedulers: [1]

1. Long-term scheduler (Job scheduler): It is responsible for loading the programs from the memory into the ready queue.
2. Short-term scheduler (CPU Scheduler): Schedules the selected process for execution and gives control to the CPU[2].
3. Medium-term scheduler: Time sharing systems uses this kind of scheduler. The process is preempted for some reason and resumed back after sometime. [9]

The performance of the scheduling algorithm is estimated based on various criteria like CPU utilization, Throughput, Turnaround time, waiting time and response time.[3]

1. CPU utilization indicates how effectively CPU is utilized.[8]
2. Throughput denotes the number of processes completed per unit time.[4]
3. Turnaround Time is the time needed for the process to complete its execution.[5]
4. Waiting Time is the amount of time a process waiting in the queue.
5. Response Time is the time between submission of request for execution and till the first response is produced not the output.

II. FCFS SCHEDULING

The processes are loaded from memory into ready queue. From the ready queue the processes are taken based on their arrival time and assigned to the processor for execution[6] as shown in Figure 1.

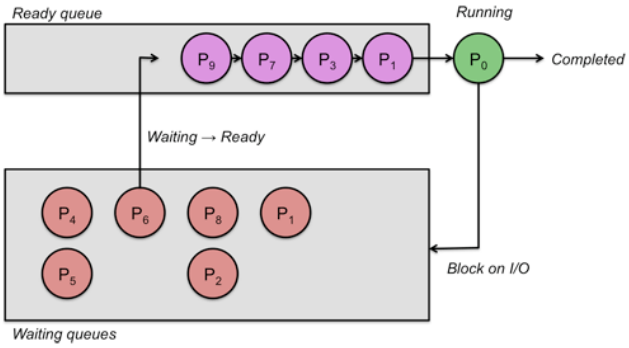


Figure 1 FCFS Scheduling

The table 1 lists the processes and their execution time (burst time). The FCFS scheduling (Figure 2) is applied and the average waiting time and turnaround time are calculated and it is listed in table 2.

Table 1 Process and burst time

S.No.	Process	Burst Time
1	P1	15
2	P2	19
3	P3	35
4	P4	40
5	P5	50
6	P6	58
7	P7	3
8	P8	5
9	P9	7
10	P10	6

Revised Manuscript Received on February 24, 2020.
***** Correspondence Author
Sambath M., Assistant Professor, Department of CSE, Hindustan Institute of Technology and Science, Chennai, India.
Email: msambath@hindustanuniv.ac.in
Dr. K. Padmaveni, Associate Professor**,** Department of CSE, Hindustan Institute of Technology and Science, Chennai, India.
Email: kpadmaveni@hindustanuniv.ac.in
Linda Joseph, Assistant Professor, Department of CSE, Hindustan Institute of Technology and Science, Chennai, India.
Email: lindaj@hindustanuniv.ac.in
Ravi S., Assistant Professor, Department of CSE, Hindustan Institute of Technology and Science, Chennai, India.
Email: sravi@hindustanuniv.ac.in
Dr. J. Thangakumar, Associate Professor, Department of CSE, Hindustan Institute of Technology and Science, Chennai, India.
Email: tkumar@hindustanuniv.ac.in
Dr. D. John Aravindhar, Professor, Department of CSE, Hindustan Institute of Technology and Science, Chennai, India.
Email: jaravindhar@hindustanuniv.ac.in

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
0	15	34	69	109	159	217	220	225	232 238

Figure 2 Gantt Chart using FCFS

Table 2 Waiting time and turnaround time using FCFS

Process	Burst Time	Waiting time	Turnaround time
P1	15	0	15
P2	19	15	34
P3	35	34	69
P4	40	69	109
P5	50	109	159
P6	58	159	217
P7	3	217	220
P8	5	220	225
P9	7	225	232
P10	6	232	238
	Average	**128**	**151.8**

The average waiting time is 128 unit time and average turnaround time is 151.8 unit time. From the table 2 it is observed that the short processes are available after the lengthy processes[7]. Due to this small processes are waiting till the completion of lengthy processes. This leads to increase in waiting time as well as turnaround time. This effect is called as convoy effect. To reduce this effect novel approach – BinFCFS algorithm is proposed.

III. BINFCFS ALGORITHM

1. Find out the minimum and maximum in burst time.
2. Create the number of bins depends on the range of burst time.
3. Group the processes and put it in the bins based on the range of burst time.
4. Take the processes from the bins in the order and execute.

For the above scenario, the processes are arranged into the bins based on their burst time as shown in Figure 3. Start from Bin 0 executes P7, P8, P9 and P10. Then continue with Bin1, Bin2 and so on.

For the above scenario, the processes are arranged into the bins based on their burst time as shown in Figure 3. Start from Bin 0 executes P7, P8, P9 and P10. Then continue with Bin1, Bin2 and so on.

The waiting time and turnaround time for each process is calculated using the proposed method – BinFCFS algorithm(Figure 4) and is given in the table 3.

Bin 0 (0-9)	**Bin 1 (10-19)**	**Bin 2 (20-29)**	**Bin 3 (30-39)**	**Bin 4 (40-49)**	**Bin 5 (50-59)**
P7 P8 P9 P10	P1 P2		P3	P4	P5 P6

Figure 3 Bins holding the processes

P7	P8	P9	P10	P1	P2	P3	P4	P5	P6
0	15	34	69	109	159	217	220	225	232 238

Figure 4 Gantt Chart using BinFCFS algorithm

Table 3 Waiting time and turnaround time using BinFCFS

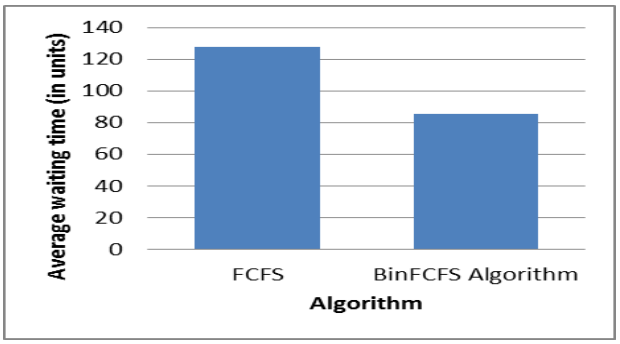
Process	Burst Time	waiting time	Turnaround time
P7	3	0	3
P8	5	3	8
P9	7	8	15
P10	6	15	21
P1	15	21	36
P2	19	36	55
P3	35	55	90
P4	40	90	130
P5	50	130	180
P6	58	180	238
	Average	**85.33**	**121.5**

IV. RESULT AND DISCUSSION

The average waiting time and turnaround time for the above problem using FCFS algorithm is 128 and 151.8 as given in table 4. It is reduced to 85.33 and 121.5 respectively using the proposed BinFCFS algorithm.

Table 4 Performance Comparison

Algorithm	Average waiting time	Average Turnaround time
FCFS	**128**	**151.8**
BinFCFS Algorithm	**85.33**	**121.5**



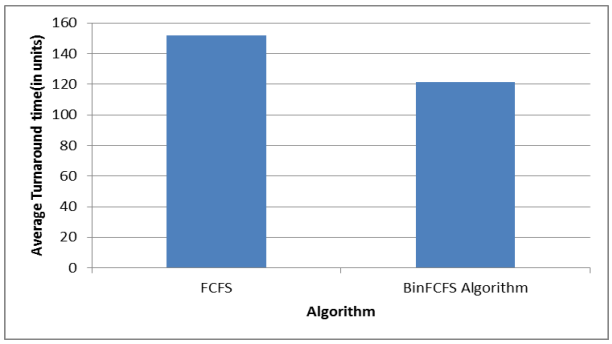


Figure 5 a) Performance comparison for Average waiting time b) Performance comparison of average turnaround time

V. CONCLUSION

The FCFS algorithm schedules the processes in the order of arrival irrespective of whether it is a short process or long process. Due to this convoy effect is occurred. This effect is avoided by the proposed BinFCFS algorithm by grouping the short processes and executing them in the first.

REFERENCES

1. Siahaan, Andysah Putera Utama. "Comparison analysis of CPU scheduling: FCFS, SJF and Round Robin." International Journal of Engineering Development and Research 4.3 (2016): 124-132
2. Dave, Badal, Surendra Yadav, and Manish Mathuria. "Customary Methods for CPU Scheduling: A Review." (2017).
3. Rinku, Dhruva R., and M. Asha Rani. "Evaluation of Scheduling Algorithms on Linux OS." Advances in Decision Sciences, Image Processing, Security and Computer Vision. Springer, Cham, 2020. 210-217.
4. Chandra Shekar, N., and V. Karthik. "Analysis of Priority Scheduling Algorithm on the Basis of FCFS & SJF for Similar Priority Jobs." International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) 4.3 (2017).
5. Aslam, Naila. Designing a Model for improving CPU Scheduling by using Machine Learning. Diss. 2017.
6. Sowmya, G., et al. "A Comparison Of Scheduling Algorithm For Best Utilization Of Memory." International Journal of Pure and Applied Mathematics 120.6 (2018): 3563-3570.
7. Sankar, A., and G. Ananthnath. "Improved Dynamic Time Slice Priority Scheduling Algorithm with Unknown Burst Time." (2018).
8. Joon, Monika, and Neetu Sharma. "Optimization Of Scheduling Algorithms For Load Balancing Using Hybrid Approach." (2017).
9. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, "Operating System Concepts", Ninth Edition, 2013.

AUTHORS PROFILE

Mr. Sambath M., is graduated in Computer Science & Engineering in the year 2001 and obtained his Master degree in Computer Science & Engineering from Anna University in the year 2008 and currently pursuing his Ph.D. in the area of Image Processing at Hindustan Institute of Technology and Science, Chennai, India. His area of interest includes Image processing, Algorithms, Operating systems.

Dr. K. Padmaveni, received Ph.D. degree in cloud computing from Hindustan Institute of Technology and Science. She is currently an Associate Professor in Hindustan Institute Of Technology and Science and her main interest are in virtualization and scheduling in cloud.

Linda Joseph, is an assistant professor at Hindustan Institute of Technology and Science, Chennai, Tamil Nadu, India. She did her masters at Noorul Islam College of Engineering, Anna University, Tamil Nadu. She is now with the Department of Computer Science and Engineering. Her research interests are Network Security, Artificial Intelligence, Virtualization and Cloud Security.

Ravi Sridharan, has received his MCA degree from SRM Engineering College, Chennai .He obtained his M.Phil Degree from Periyar University, Salem, Tamilnadu, India. He obtained his M.E., Computer Science & Engineering from Sree Sastha Institute of Engineering & Technology , Chennai .He is presently pursuing his Ph.D. at Hindustan Institute of Technology & Science , Chennai, Tamil Nadu, India. He is a member of ACM, IET, IAENG ,Toast Master Club. His area of interests is Mobile Adhoc Networks, Internet of Things & Software Engineering.

Dr. Thangakumar Jeyaprakash, has been working as an Associate Professor at Hindustan Institute of Technology and Science, Padur, Chennai in the Department of Computer Science for over Fifteen years. He has finished his Ph.D. in Computer Science and Engineering from our Institution and his research area is “Mobile Adhoc Networks and Network Security”. He is professionally very competent and has the ability to motivate the students in various academic and co-curricular activities.

Dr. D. John Aravindhar, received Ph.D. degree in Data mining from Hindustan Institute of Technology and Science. He is currently a Professor in Computer Science Department of Hindustan Institute of Technology and Science. His area of interest are Data mining and cloud computing.