IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 1, January 2022

DOI: 10.17148/IJARCCE.2022.11105

Analysis of Priority Preemptive Scheduling Algorithm: Case Study

Tri Dharma Putra

Department of Informatics, Faculty of Computer Science,

University of Bhayangkara Jakarta Raya, Jalan Perjuangan Bekasi Utara, West Java, Indonesia

Abstract: There are priority preemptive scheduling algorithm and priority non-preemptive scheduling algorithm in operating system. When the newly arrived process is compared its priority to the process that is running and if its priority is higher it will executed by the CPU. This is known as priority preemptive scheduling algorithm where when the recently arrived process is positioned at the head of the queue and cannot be interrupted, this is known as priority non-preemptive scheduling algorithm. This priority algorithm is based on execution of process over a priority value. The higher the value, the higher the priority. Every process has its own priority. The first process to be executed is the process with higher priority, then continue until all the processes finish. In this journal we discuss the priority preemptive scheduling algorithm. With the priority number from 0 until 10, where the 0 the the lowest priority and 10 is the highest priority. Two case studies are discussed here.

Keywords: Priority, Scheduling, Priority Preemptive Scheduling Algorithm, Priority Non-Preemptive Scheduling Algorithm

I. INTRODUCTION

Operating System (OS) is software which its role is as an interface between a user and the computer hardware. OS is known as a resource manager because its main duty is to manage the resources of computer system. Scheduling is one fundamental and most important design. Scheduling refers to set of rules, policies and mechanism that govern the order in which resources is allocated to various processes and the work is to be done. The scheduling is a methodology of managing many queues of processes in order to make delay minimum and to make performance optimal of the system. A scheduler is a module in operating system that implements the scheduling policies. Its main objective is to make systems performance's optimal that match with the criteria set by the system designer [7].

Scheduling is a prime concept in multiprocessing and multitasking of OS design and in real-time operating system design by arranging switching in the CPU among process. Priority Scheduling Algorithm is a well known algorithm in CPU processing.

For system of single processor, when multiple process comes, then one process can be execute at a time and other process remain in waiting state until the CPU becomes idle or can be scheduled again. To expand the CPU usage, the goal of multiprogramming is to have some procedure running at all times. CPU scheduling manages the issue of choosing which of the procedures in the ready queue is to be assigned the CPU. Operating system usually performs scheduling of processes which is major task of a system [8].

Priority preemptive scheduling algorithm is usually use among various other algorithms for scheduling CPU, however it can make the problem of starvation which happens when processes with priority which is lower are not given any chance of CPU utilization due to prolong CPU usage by processes with higher priorities [4].

II. OVERVIEW OF EXISTING ALGORITHMS

There are several scheduling algorithms proposed by experts, namely:

1. **Shortest Job First (SJF)**. In this scheduling algorithm, the system will choose the smallest burst time of process to be executed first. So it is why it is named Shortest Job First (SJF). The weakness of this algorithm is there could be starvation, the long process will never be executed because there are still shorter jobs exist in the system. If shorter job keeps coming in then starvation will occur [7]. SJF algorithm is possibly optimal. It executes the short process before the long process and thus reduces the waiting time for short process more than increases waiting time for long process. Which finally ends up with minimum average waiting time compared to the other scheduling algorithm [1].

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 1, January 2022

DOI: 10.17148/IJARCCE.2022.11105

2. **First Come First Serve (FCFS).** The process which arrives first will be executed first. This algorithm sometimes is called First In First Out (FIFO). This is the simplest scheduling algorithm. While the processes in the ready queue will occupy the CPU in the order of their arrival to the ready queue. The process which enters first in the ready queue will occupy the CPU first and the process which enters afterwards will occupy CPU sequentially in the arrival order. This scheduling algorithm is non-preemptive. Once the process get allocated the CPU it will not leave the CPU until it get terminated. In this algorithm the ready queue is implemented using FIFO list. The newly arrival processes are added to the rear end and the CPU is allocated to a process at front end or head. The drawback in this algorithm is the high average waiting time [6] [10] [3].
3. **Round Robin (RR)**. Round robin algorithm is a real-time scheduling algorithm in operating system. The RR CPU scheduling algorithm is cited as standard RR and it is a preemptive type that allocates a slice of time named TQ which stands for time quantum for every process in ready queue. Whenever TQ completes the current process is preempted and put in the rear of ready queue. RR is usually applied in real-time and time-sharing operating system because it provides every process an average share of time to utilize the CPU and gives a small responds time. Moreover, the standard RR algorithm has many weaknesses such as small throughput and big turnaround time as well as the bit waiting time and also huge context switches number [3]. Dynamic time slice is one solution to find the effective algorithm running in the system. The intelligent time slice (quantum) for round robin architecture for real time operating systems is a modified version of simple round robin scheduling [11].
4. **Priority Scheduling**. For Priority Scheduling, the basic idea is straightforward, each process is assigned a priority. Priority processes which are equal are scheduled in FCFS order. SJF is one example. SJF has the same idea as Priority Scheduling. Namely, the longer the burst of CPU, then it makes the lower the priority and the smaller the burst time, the higher the priority. Priority can be defined internally or externally. Internally defined priorities use some measurable quantities or qualities to computer priority of a process [7]. Priority scheduling algorithm manages processes in its queue based on its priority. Something else that gives priority on running state is preemption [5].

III. ORGANIZATION STRUCTURE

This journal divides into six chapters. The first chapter is introduction. Here we discussed the story behind the scheduling algorithms in operating system. In the second chapter we discuss the overview of existing algorithms, Discussion about algorithm like shortest job first, first come first serve, round robin and priority algorithm in general. The third chapter is about organization structure of this journal. The fourth chapter is about priority preemptive scheduling algorithm in detail, Chapter five is about case study analysis for priority scheduling algorithm. We discuss two case studies and the calculation of average turnaround time and average waiting time. The last chapter is conclusion.

IV. PREEMPTIVE PRIORITY SCHEDULING ALGORITHM

Let we start with priority scheduling algorithm. Every process has a priority and the CPU is allocated based on the highest priority. The important of any process is based on a range of values. Any process entering the ready queue will be allocated its importance based on priority. This priority value defined the decision of allocating the CPU to a process in a way that the process with high value of priority will occupy the CPU first or next. There are two version, one is preemptive, another one is non-preemptive. We will discuss the preemptive one.

In this algorithm, the higher the value of the priority, the higher the priority. Let's say the priority is 3 compared with priority 1, so the priority 3 will get the CPU first, since its priority is higher. And in the preemptive system, the process can be interrupted by a new process in ready queue if the new process has higher priority than the running one. Starvation can occur in this system if the value of priority is always higher than the one in the waiting queue.

Processes with the same priority are executed based on FCFS. Priority can be defined based on time requirements, memory requirements of any other resources requirements [2]. Priority are indicated by some fixed range of numbers , such 0-7, in this analysis we use priority number from 1 to 5 [9].

V. CASE STUDY ANALYSIS

5.1. Case Study 1

Here we have four processes, in table 1, namely A, B, C, and D. Also given here different arrival times, with different priorities each.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 1, January 2022

DOI: 10.17148/IJARCCE.2022.11105

TABLE 1. CASE STUDY 1

Process	Arrival Time	Burst Time	Priority
A	0	4	1
B	1	3	3
C	2	5	4
D	4	4	5

Process A arrives at 0, with burst time 4 and priority 1, but it is interrupted at 1 by B with higher priorities, namely 3. Then B occupies the CPU for one milli second since C, at 2 with priority 4 interrupts B. C with priority 4 higher than B, then at 4 mili seconds, D interrupts C for 4 mili seconds since D has the highest priority, namely 5. Then C occupies CPU for 3 mili seconds since C has the higher priority than the rest. Then B for 2 mili seconds then finally A occupies CPU for 3 mili seconds.

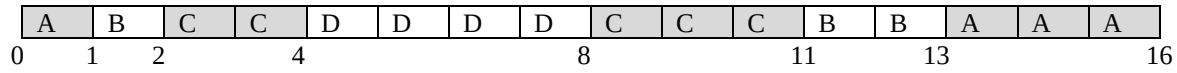

Fig 1. Gantt Chart of Case Study 1

From the analysis above we have gantt chart as follows (16 miliseconds) as in fig 1 above.

TABLE 2. CALCULATION OF AVERAGE TURNAROUND TIME AND AVERAGE WAITING TIME

Process	Arrival Time	Burst Time	Starting Time	Finish Time	Turnaround Time	Waiting Time
A	0	4	0	16	16-0 = 16	13-1=12
B	1	3	1	13	13-1=12	11-2=9
C	2	5	2	11	11-2=9	8-4=4
D	4	4	4	8	8-4=4	0
				average	10.25	6.25

From table above we get average turnaround time 10.25 and average waiting time 6.25.

5.2. Case Study 2

In this case study 2, table 3, we have four processes, namely A, B, C, and D. Also given here different arrival times, with different priorities each.

TABLE 3. CASE STUDY 2

Process	Arrival Time	Burst Time	Priority
A	0	5	1
B	1	3	3
C	3	3	4
D	4	4	5

Process A arrives at 0, with burst time 5 and priority 1, then A is interrupted by B at 1 mili seconds since B has higher priority namely 3. After 2 mili seconds, C interrupted B at 3, for 1 mili second. C has higher priority namely 4 than B which is 3. D interrupts C at 4 with the highest priority, namely 5 occupies the CPU for 4 mili seconds, after that C occupies the system for 2 mili seconds since C has the highest priority than the rest in ready queue. Then B gets in for 1 mili second, Finally A occupies the CPU for 4 mili seconds.

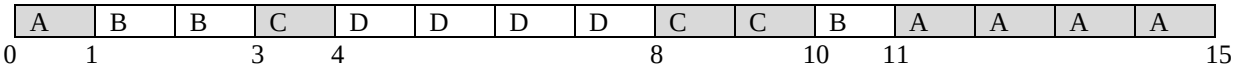

Fig 2. Gantt Chart of Case Study 2

From the analysis above we have gantt chart as follows (15 mili seconds) as figure 2 above.

TABLE 4. CALCULATION OF AVERAGE TURNAROUND TIME AND AVERAGE WAITING TIME

Process	Arrival Time	Burst Time	Starting Time	Finish Time	Turnaround Time	Waiting Time
A	0	5	0	15	15-0 = 15	11-1=10
B	1	3	1	11	11-1=10	10-3=7

DOI: 10.17148/IJARCCE.2022.11105

C	3	3	3	10	10-3=7	8-4=4
D	4	4	4	8	8-4=4	0
				average	9.0	5.25

From table above we get average turnaround time 9.0 and average waiting time 5.25.

VI. CONCLUSION

The objective of this journal is to analyse Priority Preemptive Scheduling Algorithm in CPU. The calculation of two case studies show that prioriy process scheduling has different average waiting time. For case study 1 we get average turnaround time 10,25 and average waiting time 6,25 mili seconds. While in case study 2 we get average turnaround time 9.0 and average waiting time 5.25 mili seconds. These two case studies give understanding about the priority process scheduling more thoroughly. The preemptive priority scheduling will interrupt the process if the running process has lower priority than the ready process. Case study 1 finishes after 16 mili seconds where case study 2 finishes after 15 mili seconds.

REFERENCES

[1]. Asma Joshita Trisha, S. B. (2019). A Combined Preemptive SJF and Preemptive Priority Algorithm to Enhance CPU Utilization. International Journal of Computer Applications, 177(19), 26–30.

[2]. Chandra Shekar N, K. V. (2017). Analysis of Priority Scheduling Algorithm on the Basis of FCSF & SJF for Similar Priority Jobs. International Journal of Engineering Research in Computer Science and Engineering, 4(3), 73–76.

[3]. Hoger K. Omar, Kamal H. Jihad, S. F. H. (n.d.). Comparative Analysis of the Essential CPU Scheduling Algorithms. Bulletin of Electrical Engineering and Informatics, 10(5), 2742–2750.

[4]. Kunal Chandiramani, Rishabh Verma, S. M. (2019). A Modified Priority Preemptive Algorithm for CPU Scheduling. International Confierence on Recent Trends in Advanced Computing 2019, ICRTAC 2019, 363–369.

[5]. Ledina Hoxha Karteri, A. S. (2015). Preemptive and Non- Preemptive Priority Scheduling. International Journal of Computer Science & Management Studies, 19(01), 1–5. www.ijcsms.com

[6]. Pawan Singh, Amit Pandey, A. M. (2015). Varying Response Ratio Priority: A Preemptive CPU Scheduling Algorithm (VRRP). Journal of Computer and Communications, 3(April), 40–51.

[7]. Putra, T. D. (2020). Analysis of Preemptive Shortest Job First (SJF) Algorithm in CPU Scheduling. IJARCCE, 9(4), 41–45. https://doi.org/10.17148/ijarcce.2020.9408

[8]. Rukhsar Khan, G. K. (2015). Analysis of Priority Scheduling Algorithm on the Basis of FCFS & SFJ for Similar Priority Jobs. International Journal of Computer Science and Mobile Computing, 4(9), 324–331.

[9]. Saraswathi Seemakuthi, Verikat Alekhya Siriki, E. L. L. (2015). A Review on Various Scheduling Algorithms. International Journal of Scientific & Engineeering Research, 6(12), 769–777.

[10]. Siahaan, A. P. U. (2016). Comparison Analysis of CPU Scheduling: FCFS, SJF and Round Robin. International Journal of Engineering Development and Research, 4(3), 124–131.

[11]. Tri Dharma Putra, A. F. (2021). Comparison Between Simple Round Robin and Intelligent Round Robin Algorithms in CPU Scheduling. International Journal of Advanced Research in Computer and Communication Engineering, 10(4), 86–90.

