Information Engineering Express
International Institute of Applied Informatics
2019, Vol. 5, No.1, 11-18

An Adjustable Round Robin Scheduling Algorithm in Interactive Systems

Samih M. Mostafa*, Hirofumi Amano†

Abstract

CPU scheduling is considered as the basic job within the operating system. Scheduling criteria including waiting time, context switches and others have been suggested for comparing CPU scheduling algorithms. In this paper, a modified version of Round Robin algorithm is introduced as an attempt to combine the advantageous of low scheduling overhead of Round Robin and favor short process to minimize the average waiting time and number of context switches of running processes in interactive (time-shared) systems. A threshold is considered to determine whether the running process will be interrupted because of the expiration of its time slice specified by the Round Robin policy or will continue execution until termination. Derived results show that the suggested modification minimizes the average waiting time and number of context switches compared to Round Robin algorithm.

Keywords: CPU scheduling, interactive system, time-sharing, Round Robin.

1 Introduction

CPU scheduling problem decides which of the processes in the ready queue to be allocated the CPU [1,2]. Scheduling algorithms are the mechanism by which a resource is assigned to a client. In this paper, the concept of a resource is restricted to CPU time and clients to processes. The decision of scheduling refers to the concept of selecting the next process for execution.

There are many different CPU-scheduling algorithms which have different properties, and the choice of a particular algorithm may favor one class of processes over another, therefore, selecting the appropriate algorithm is an issue. To choose which algorithm to be used in a particular situation, the properties of the various algorithms must be considered. Defining the criteria to be used in the selection is the first and important aspect to be considered [2–7]. Waiting time (i.e., the total time the process spent in the ready queue) is an important criterion to be minimized.

It is necessary to saving the context of a running process when it is interrupted or waits for an I/O device. This means that the current context of a process always resides in its process control block (PCB). For simplicity, whenever the running process stops, its context is saved in its PCB and the context of other scheduled process from its PCB is loaded. This is known as context switching.

*Mathematics Department, Faculty of Science, SVU University, Qena, Egypt
†Research Institute for Information Technology, Kyushu University, Japan

[3] S. M. Mostafa and S. Kusakabe, “Effect of Thread Weight Readjustment Scheduler on Scheduling Criteria,” Inf. Eng. Express, Jan. 2015.

[4] S. M. Mostafa and S. Kusakabe, “Towards Maximizing Throughput for Multithreaded Processes in Linux,” Int. J. New Comput. Archit. their Appl., vol. 4, no. 4, pp. 70–78, 2014.

[5] A. Singh, P. Goyal, and S. Batra, “An optimized round robin scheduling algorithm for CPU scheduling,” Int. J. Comput. Sci. Eng., vol. 02, no. 07, pp. 2383–85, 2010.

[6] S. Elmougy, S. Sarhan, and M. Joundy, “A novel hybrid of Shortest job first and round Robin with dynamic variable quantum time task scheduling technique,” J. Cloud Comput., vol. 6, no. 1, pp. 0–12, 2017.

[7] S. M. Mostafa, “Proportional Weighted Round Robin: A Proportional Share CPU Scheduler in Time Sharing Systems,” Int. J. New Comput. Archit. their Appl., vol. 8, no. 3, pp. 142–47, 2018.

[8] M. S. Iraji, “Time Sharing Algorithm with Dynamic Weighted Harmonic Round Robin,” J. Asian Sci. Res., vol. 5, no. 3, pp. 131–42, 2016.

[9] T. Helmy and A. Dekdouk, “Burst round robin as a proportional-share scheduling algorithm,” Jan. 2007.

[10] M. M. Tajwar, M. N. Pathan, L. Hussaini, and A. Abubakar, “CPU scheduling with a round robin algorithm based on an effective time slice,” J. Inf. Process. Syst., vol. 13, no. 4, pp. 941–50, 2017.

[11] B. Caprita, W. C. Chan, J. Nieh, C. Stein, and H. Zheng, “Group ratio round-robin: O (1) proportional share scheduling for uniprocessor and multiprocessor systems,” Proc. Annu. Conf. USENIX Annu. Tech. Conf., no. 1, pp. 36–36, 2005.

[12] B. Caprita, J. Nieh, and W. C. Chan, “Group Round Robin: Improving the Fairness and Complexity of Packet Scheduling,” Proc. 2005 ACM Symp. Archit. Netw. Commun. Syst., pp. 29–40, 2005.

[13] L. Abeni, G. Lipari, and G. Buttazzo, “Constant bandwidth vs. proportional share resource allocation,” no. July 1999, pp. 107–11, 2003.

[14] A. Silberschatz, G. Gagne, and P. B. Galvin, Operating Systems Concepts. 2012.

[15] J. Sunil, V. G Anisha Gnana, and V. T Karthija, Fundamentals of Operating Systems Concepts. 2018.