**Application of Shortest Job First in Utilizing Multi-level Queue Scheduling Algorithm for CPU Scheduling**

Adrian A. Cabural, Erl Jean S. Recile, Michael L. Orbino

College of Information Technology and Computer Science

University of the Cordilleras

**Abstract – Many algorithms are available for CPU scheduling including Multi-level Queue (MLQ) and Shortest Job First (SJF) scheduling algorithm. It has been observed that the SJF reduces the average waiting time (AWT), Average Turn-around Time (ATAT) and Burst Time (BT) to the minimum, in a single CPU environment. The main objective of this paper is to provide a complete visual representation of the performance for the scheduling algorithm of MLQ and SJF with the combination of both by applying SJF scheduling algorithm in each level of queues. We, conclude, therefore that (SJFMLQ) Application of SJF in utilizing MLQ reduces the average waiting time, the average turn-around time and the burst time of each queue priority.**

**I. Introduction**

CPU scheduling is the basic operation of the operating system. It’s a procedure, which is used by the processes threads; user process or data gain permissions to the system given assets. The purpose of these scheduling techniques is to increase the system running performance and fair used or available resources. The basic idea of scheduling is to share computer assets among the available number of process to request for system resources. Largely the computer resources are schedule before they use. CPU is the one of most important resource of the computer system resources. It is more important in operating system design to effectively and efficiently to schedule this operating system resource.

CPU scheduling decides which process perform when their execution and acquire how many and which resources. CPU scheduling is essential because it performs a major role in efficient resource utilization and it positively affects the overall system performance. If there are multiple process ready for their execution then it decides which process perform his execution when.

Multi-level queue and Shortest Job First are one of those scheduling algorithms for CPU scheduling. In introducing Multi-level Queue Scheduling, A common division is made between foreground (Interactive) processes and background (batch) processes. These two types of processes have different response-time requirements and so may have different scheduling needs. In addition, foreground processes may have priority (externally defined) over background processes. A scheduling algorithm for multilevel queue partitions the ready queue into several separate queues. The processes are permanently assigned to one queue, generally based on some process property, such as memory size, process priority, or process type. Each queue has its own scheduling algorithm, In this case we will apply shortest job first in both processes.

Shortest job first is a scheduling algorithm in which the process with the smallest execution time is selected for execution next. Shortest job first can be either preemptive or non-preemptive. Owing to its simple nature, shortest job first is considered optimal. It also reduces the average waiting time for other processes awaiting execution. Additionally, the processes that are part of the interactive and batch processes will be sorted when applied in multilevel queue partitions.

**II. Related Literature**

A large number of papers are published on the topic of operating system algorithms in the recent years. Many researchers develop their own scheduling algorithms, which are published in their articles. These different algorithms have worked differently according to their model, which is used to develop these algorithms.

To order to ensure better performance metrics, users should understand the characteristics of these algorithms. Several requirements for evaluating CPU scheduling algorithms have been proposed. Some of the criteria are discussed below.

**Burst Time:** Time required by a process for CPU execution.

**Turnaround Time:** From the submission time of a process to its completion time is turnaround time. It refers to the total time period spends waiting to get into memory, waiting in the ready queue, executing in the CPU and doing input/output operations.

***Turnaround Time = Completion Time – Arrival Time***

***Average Turnaround Time = total of all Turnaround Time / no. of processes***

**Waiting Time:** The scheduling algorithm for the CPU does not affect the amount of time the system runs or inputs / outputs throughout. It affects only the amount of time that a process spends waiting in the ready queue. Waiting time refers to the sum of time periods spends waiting in the ready queue.

***Waiting Time = Turnaround Time – Burst Time***

***Average Waiting Time = total of all Waiting Time/ no. of processes.***

**Context Switch:** The process of storing the state of a process or of a thread, so that it can be restored and execution resumed from the same point later. This allows multiple processes to share a single CPU, and is an essential feature of a multitasking operation system

The first come first served is one of the easiest scheduling techniques. Ready queue of this algorithm is like ready queue present in First In First Out (FIFO). The process selects from the head of ready queue and execution starts. When the running process completes execution the process is terminated and removes from the process list. The new process is select for execution from the ready queue.

In smallest jobs scheduled first (SJF) scheduling algorithm the ready queue is arranged in order to burst duration of CPU. In these algorithms the activities with shortest burst time are placed in front or the queue. On the basis of this algorithm the process is select and transmits to CPU for execution. When the running process completes its execution the process is terminated and removes from the process list. The new process is selected for execution from the ready queue head.

Existing literature shows that SJF is an optimal scheduling algorithm as its average waiting time is less than other algorithms. Its throughput is quite better than the other scheduling algorithms. Besides it numerous advantages, it shows the problem of high context switching. Context switching is also an overhead. It decreases the system performance. So there is a need to minimize the context switching to maximize the system performance.

**III. Proposed Algorithm**

1. The proposed algorithm is hybrid-scheduling algorithm. This algorithm consists of two algorithms; the first is multi-level queue algorithm (MLQ) and the other one is the shortest job first (SJF) algorithm.

Pre-initialized conditions:

* Time quantum is directly initialized to 4 in the code. It can be changed if any other time quantum requirement popup.
* All the arrays like completion time, waiting time, burst time are initialized with 500.
* Arrival time is taken as 0 for all the process.
* Priority contains 0 and 1.1 Indicates high priority and 0 indicates low priority.

Pseudo code of SJFMLQ algorithm:

* Burst time and priority of each process is read from the excel sheet and saved into two arrays of size 500.
* Based on the priority the all the process is moved into 2 queues.
* The process with high priority (1) is moved to queue 1 and all the process with low priority be moved to queue 2.
* SJF are performed on all queues, queue 1 will first then move to queue 2.
* Take number of elements to be inserted.
* Select process that has shortest burst time among all process will execute first.
* If processes have same burst time length then FCFS (First come First Serve) scheduling algorithm used.
* Make average waiting time length of next process.
* Start with first process, selection as above and other processes are to be in queue.
* After the execution of both the queues is completed then we calculate waiting time and turnaround time of each process.

**B.** Illustration of the SJFMLQ algorithm

To further illustrate how the algorithm works, we present a simulation.

Input:

* Process [0,1,2,3,4]
* Burst time [3,4,7,2,4]
* Priority [0,1,1,0,1]
* Arrival time =0

Example:

* Q1-----Process [1,2,4]
* Q2-----Process [0,3]
* SJF on both queue

Final Gant chart is

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P1 | P4 | P2 | P3 | P0 |
| 0 4 | 4 8 | 8 15 | 15 17 | 17 20 |

Q1 executed first, P1, P2, and P4 are part of the Q1 or interactive process. According to the burst time of each process P1 and P4 have the same burst time length then FCFS scheduling algorithm was used. Then P2 was processed with burst time length of 7. After the Q1 finished executing, Q2 was processed. Since P3 has a shorter burst time length it was processed first before P0.

Average Waiting time:

(0+4+8+15+17)/5 = 8.8

Average Turnaround time:

(4+8+15+17+20)/5 = 12.8

**IV. Test Cases**

**Tables and Charts**

A. Test Case 1 (MLQ)

B. Test Case 2 (SJF)

C. Test Case 3 (MLQSJF)

Assumed 5 process with the same burst time that will have a visual representation on each algorithm.

**V. Discussions**

A. About the MLQ and SJF comparison

B. About the comparison of MLQ and SJF scheduling algorithm with MLQSJF algorithm.

**References:**

[1]University of Cincinnati (2018, May 31). Multi-Level Queue Scheduling Algorithm

[2]Muhammad Akhtar, Bushra Hamid, Inayat ur-Rehman, Mamoona Humayun, Maryam Hamayun and Hira Khurshi. (2015, October 30). An Optimized Shortest Job First Scheduling Algorithm for CPU Scheduling. *Journal of Applied Environmental and Biological Sciences* ISSN: 2090-4274.

<https://www.researchgate.net/publication/286452857_An_Optimized_Shortest_job_first_Scheduling_Algorithm_for_CPU_Scheduling>