Scientific Journal of Faculty of Education, Misurata University-Libya, Vol. 8, No. 20, Sep. 2022
Published online in September
المجلة العلمية لكلية التربية، جامعة مصراتة، ليبيا، السنة الثامنة، المجلد الثامن، العدد العشرون، سبتمر 2022
Issn :2710- 4141
تاريخ النشر 2022/10/11 تاريخ الاستلام 2022/**07**/15

Control Processing Unit Scheduling

Raheel Mhdil Baka
College of Education-Misurata University
R..baka@edu.misurata
Mofida Mohammed Elobeidi
College of Technical Sciences/Misurat
Mof.716.g@gmail.com
Marwa Ismal Edurrat
Faculty of Information Technology- Misurata University
m.eldurrat@it.misurata

Abstract:

This paper is a general survey for the central processing unit (CPU) scheduling, illustrates the types of scheduling and how to deal with operations executed by the processor, besides; the analysis of scheduling algorithms necessary to ideally use the processor in executing operations existing in waiting queue.

Keywords: Job Scheduling, CPU Scheduling, Multilevel feedback queue, Process Scheduling, Time quantum, turn-around-time, waiting- time, context-switching, MLFQS (Multilevel-Feed Back Queue Scheduling) and MLQS (Multilevel Queue Scheduling).

جدولة وحدة المعالجة المركزية

راحيل المهدي بكة
كلية التربية- جامعة مصراتة

مفيدة محمد العبيدي
كلية العلوم التقنية- مصراتة

مروة اسماعيل الضراط
كلية تقنية المعلومات- جامعة مصراتة

الملخص:

هذا البحث عبارة عن دراسة عام لجدولة وحدة المعالجة المركزية (CPU)، ويوضح أنواع الجدولة وكيفية التعامل مع العمليات التي ينفذها المعالج بالإضافة إلى تحليل خوارزميات الجدولة اللازمة لاستخدام المعالج بشكل مثالي في تنفيذ العمليات الموجودة في قائمة الانتظار.

الكلمات المفتاحية:

جدولة الوظائف، جدولة وحدة المعالجة المركزية، قائمة انتظار متعددة المستويات، جدولة العملية، كم الوقت، وقت الدوران، وقت الانتظار، تبديل الحالة (السياق)، MLFQS (متعدد المستويات- تغذية رجوع جدولة قائمة الانتظار) وMLQS (جدولة قائمة انتظار متعددة المستويات).

Introduction:

Processor is the core and mastermind of computer, responsible of execution all operations related to processing (arithmetic and logic), in addition to coordinate and organize execution of these operations according to the CPU scheduling algorithms. CPU is comprised registers where data required for carrying out arithmetic operations stored temporarily to be processed, and controlled by CPU. In single-kernel processor computers, or even multi-kernel, there is one or many processors must interfere to perform various operations in short time without being overlapped or the outlet of first operation is the inlet of the second one [2][10] .

CPU scheduling is viewed as setting a plan to arrange operations to access the processor as one operation accesses it each time exploiting processor's time. Scheduler undertakes arrangement entering of operations to the processor.

CPU scheduling is defined as an operation of choosing the appropriate operation from the ready queue, then and put it in execution [2].

This paper aims at illustrating types and standards of scheduling, and policies followed in organizing execution of operations in the processor which called algorithms. The area of this paper summarized in one CPU with many operations, and one operation employ a CPU in a particular time, one task allocated for one operation.

We review here which algorithms ideally exploit the processor, execute its operations in an organized way, which is by studying the classifications of scheduling algorithms, their advantages and disadvantages by means of their own criteria.

2- Literature Review

Mechean (2011) and SurananWarats (2007) indicated that, in Multi-Task operating system the problem of scheduling is a main problem, and that there are criteria that specify the algorithms that have priority in execution the processed operations.

Silbersdiatz (2009) indicated that scheduled algorithms is a technique used to distribute resources within sections, instantaneously and incompatibly, and to be used in operating system to share CPU time of operations, threads, Disk drives (I/O) scheduling.

Abrelmona (2000) explained that Short Time Scheduler (STS) algorithms are used to determine which available operation is to be executed later on in the processor. Long Time Scheduler (STS) algorithms apply limitations on the systems and system users in any time.

Consequently, we indicated, in this paper, the algorithms that will occupy the processor time in an ideal way, which gives operation the priorities to be executed in an organized way, besides; the criteria necessary to identify the optimum algorithm.

3- Objective of CPU Scheduling

Scheduling refers to the set of policies and mechanisms that and OS support for deterring the order of execution of pending jobs and processes A scheduler is an OS module , that determines the next pending job to be admitted into the system for execution or next ready process to dispatched to RUN state.

The basic idea of scheduling is to keep the CPU busy as much as possible by executing a (user) process until it must wait for an event, and then switch to another process.

One processor in a certain computer system means that only one operation will be performed, in which; upon execution, a chain of I/O burst-CPU burst is formed. Burst is what processor usually does when executing a set of orders, or what input/output devices do in order to carry out input and output operations [2][7][10], where the processor is void and performing no operation, but waiting until input and output is finished.

In this case the processor is an idle, where the exploitation of the processor is not deemed to be optimum.

As a step to realize a better exploitation of processor, some operations should set ready to be applied in the main memory, that is called Multi Programming[2], where more than one operation are performed simultaneously, which is CPU Scheduling to finish the work of the processors in as short time as possible [2].

4- Scheduling Criteria

Scheduling is most significant features of operations functioning, as it organize input which will be performed in the CPU. It depends on several criteria that determine the operation to be performed[2][4][6][7], these criteria represent in the following :

1- Utilization of CPU: Where processing unit is busied to be used for better utilization.

2- Used up Time: It is the time from starting performance of the operation up to the end.

3- Compatible Data Amount: It indicates the rate of operations done within a certain time.

4- Waiting Time: It represents the time any operation needs to wait in the queue, before accessing the processor to be executed.

5- Response Time: The time required by the program in order to start the actual work.

6- Equivalence: The degree in which each operation has an equal chance to be performed, no operation is deprived of being not executed.

The previous criteria should be observed in scheduling operation [2][4], there are some oppositions to these criteria, as follows :

1- Processor exploitation criterion opposed response time criterion. To abiding by the former results in lost the later [2].

2- In the event of applying equivalence criterion, it optimally apposes processor exploitation criterion, that increases the waiting time, leading to an opposition with it, and that reduces the number of executed operations during time unit, and apposes flowing data amount.

In the oppositions we already mentioned, these criteria should be observed and combined in scheduling algorithms.

5. Types of Scheduling

After operation entered into the processor to be executed, it is to be put in job queue by means of queue routing, in order to provide a necessary space in the memory, accordingly; the scheduler chooses a task to and loads it in the main memory, and places it in ready queue.

In many systems, scheduling activity is divided into separate functions :

5.1 Long-term scheduling

This scheduling determines which operation will access the queue, and which one leave the system, and which operations will be postpone.
It transports the operation from New case to Ready one [2], this is depend on the capacity of main memory, as the bigger is the capacity the lesser is the time, because it requires no waiting in order to allow sites in the memory [2][4].This scheduling is important in the actual time.

5.2 middle-term scheduling

This type is existent in some operating systems, it removes operations from the memory according to the priority and space available in the memory. i.e. it determines the tasks that should be postponed or resumed [2][4].

5.3 Short-term scheduling

It specifies the operations that should be processed and occupied the resources of processing and the time period of each operation.

This scheduling transport the ready operations from the memory to the processor to be executed, observing that the operation to be fast [2][4].

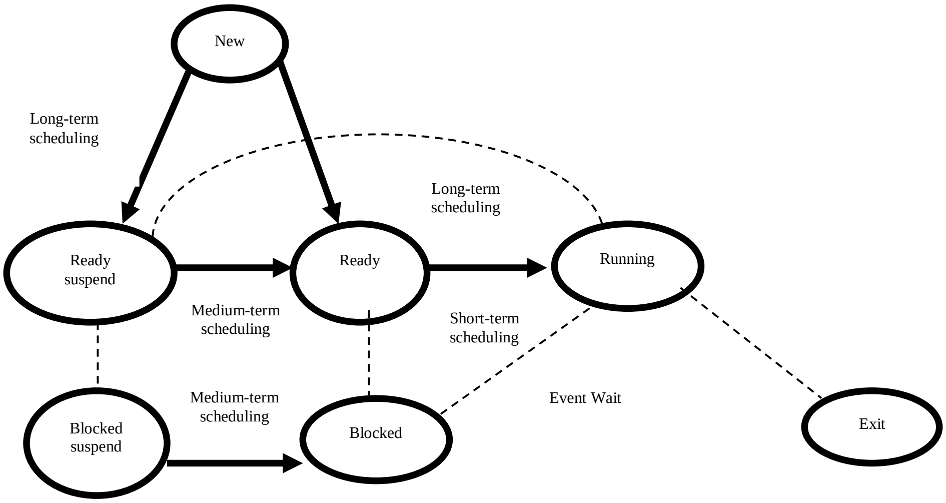

Figure (1): Types of Scheduling

6. Classification of Scheduling

Processors scheduling is used in managing the operations intended to be executed in CPU, it could be classified into two types as indicated later.

6.1 Preemptive Scheduling

It is called (preemptive scheduling) when it gives the operation to the processor, and the processor can pre-empt this operation even if it is in the midway of execution and converts the control to another operation. For instance, when a new operation and to its execution time is lesser than the time remained for execution of the current operation, so the current operation is pre-empted and a new operation allowed to access the CPU [2].

This operation is an optimum, i.e.; it gives lesser value for waiting time average of a number of operations, but its estimations are not efficient for the execution time of the operation.

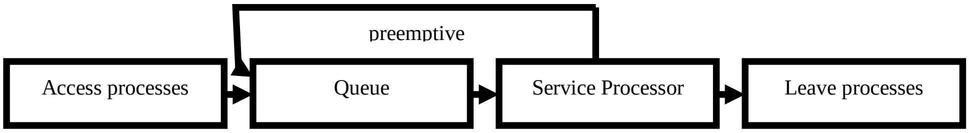

Figure (2) : Preemptive Scheduling

6.2 Non_Preemptive Scheduling

It is called non-preemptive scheduling when the processor is given an operation. i.e., the a short operation is selected from waiting queue, it doesn't leave CPU unless the its execution time finished [2][4], the most features of this type of scheduling:

1- Short operations kept waiting by long ones.
2- Response times are expected to be more because operations that have greater priority cannot replace the waiting operations.

This scheduling needs specific programs such as a timer and a kernel that has no jurisdictions to take the operation out of CPU.

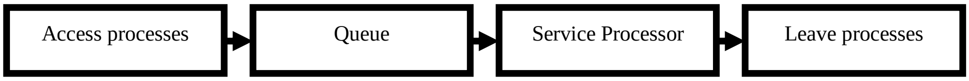

Figure (3) : Non_ Preemptive Scheduling

7. Dispatcher

The dispatcher is the module that gives control of the CPU to the process selected by the scheduler. This function involves:

1- Switching context.
2- Switching to user mode.
3- Jumping to the proper location in the newly loaded program.

The dispatcher needs to be as fast as possible, as it is run on every context switch. The time consumed by the dispatcher is known as dispatch latency.

8. Scheduling Algorithms Analysis

The main idea of scheduling algorithms is the increase of processors production rate, the benefits, reducing waiting time, response time, returning, achieving fairness between operations, as well as reducing scheduling length.

This part of paper comprises a review of basics and general concepts of scheduling related to multi processors. Owing to the great evolution of computers (multi processors are coming out), priority of executing operations should be set, additionally; more than one operation could be performed simultaneously, as the scheduling issue manifests in accomplishing this priority to carry out a certain operation from many one, allocating a processor to it from many processors, then completing execution after setting a time for each operation, a short time possible [2], that is; no more than one processor can be allocated to one operation, i.e., each processor is allocated to execute on operation, or else it will stay unemployed.

8.1 First Come First Serve Algorithm (FCFS)

It is also called first in first out (FIFO), it depends on the queue notion, as no operation can performed before the operation that comes first. Functions are arranged according to the priority of its arrival and its status in the waiting queue [1][2][4][7][10].

It characterizes with that the waiting time is not necessarily to be the shorter, the ideal exploitation of the processor, besides; its easiness in terms of programming and execution. Its imperfections represented in that any operation takes long time in order to enter which many consume all the time of CPU, and accordingly; a range to execute the short-time operation should be available.

To solve this problem a time should be set for process the operation needless to shift it with other operations, this time is called (time slice).

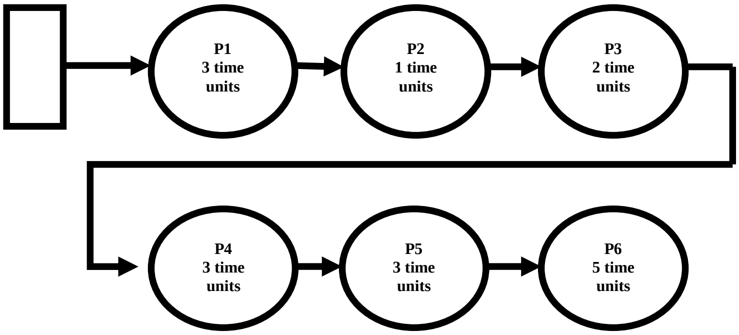

Figure (4) : First Come First Serve Algorithm (FCFS)

Example:

Process	Arrival time	Service time
1	0	8
2	1	4
3	2	9
4	3	5

FCFC

P_1	P_2	P_3	26

0 8 12 21 26

Average wait = ((8-0)+(12-1)+(21-2)+(26-3))/4=
61/4 =15.25

8.2 Shortest-Job-First Algorithm (SJF)

Every operation is associated with the time required to for the execution, after that; the short-time operation is selected [1][4][7][10]. That is so obvious in preemptive and non-preemptive scheduling, as we have already stated.

Example:

Process	Arrival time	Service time
1	0	8
2	1	4
3	2	9
4	3	5

Preemptive Shortest Job First

P_1	P_2	P_4	P_1	P_3

0 1 5 10 17 26

Average wait = ((5-1)+(10-3)+(17-0)+(26-2))/4=
52/4 =13.0

8.3 Round Robin (RR) Algorithm

This algorithm allows a specific execution time to all operations within CPU, which is called (quantum), its value ranges between 10-100 millisecond, upon completion of this time, the operation goes to enter waiting queue. It is worth noting that if quantum value is raised, then it will shift to a

FCFS, and if we reduced the quantum value processing time will end up with context time [1][2][3][6][10].

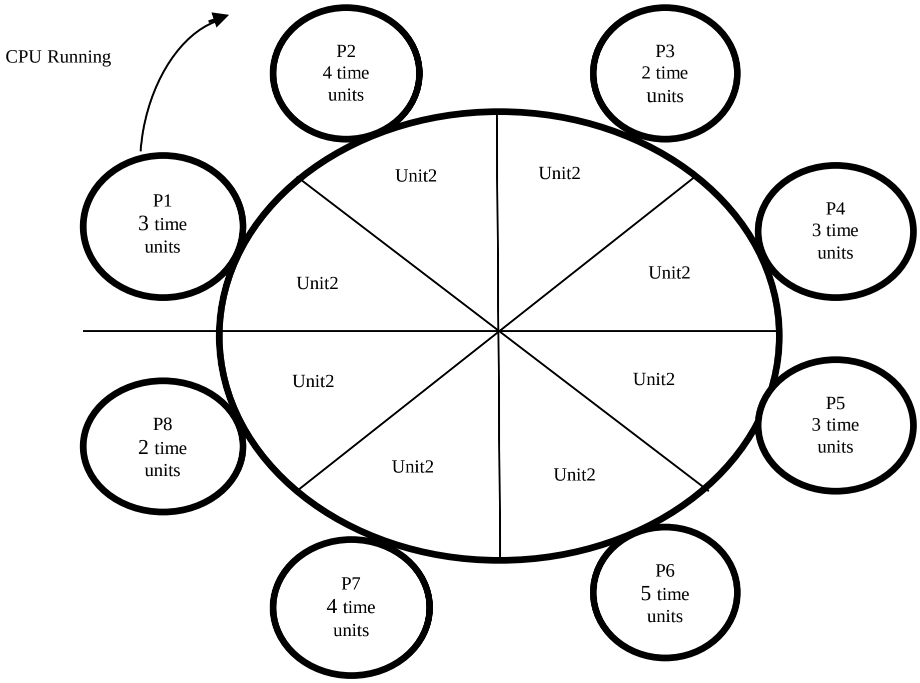

Figure (5) : RR Algorithm

Example:

Process	Arrival time	Service time
1	0	8
2	1	4
3	2	9
4	3	5

Round Robin, Quantum = 4

No priority-based preemption

P_1	P_2	P_3	P_4	P_1	P_3	P_4	P_3

0 4 8 12 16 20 24 25 26

Average wait = ((20-0) + (8-1)+(26-2) + (25-3))/4= 73/4=18.25

8.4 Priority Algorithm

In this algorithm priority degree is to be distributed to each operation. Top-priority operations are executed first [2], and there are two types of operation. The first type is by preemption that allows the currently executed operation to be aborted, and the other type is that when top-priority operated accessed to the queue, the processor seizes functioning with the current operation and begins executing the arrival operation which has top priority [4][7].

8.5 Multi Level Feedback Queue

Here the next operation has the priority in execution, which has high priority with all allocating a very short time, i.e., it depends on priority and time sharing [1]. If the operation completed the allocated time, its priority drops by one degree, and it get a double time next time, where it achieves the criterion of response speed as the processor will be unoccupied, this algorithm divides the ready queue to several different queues. And by depending on many criteria (memory size, priority of job, type of job) the job will be vested in the appropriate queue, and each queue shall has an algorithm to follow in scheduling the job that vested to it [4].

Here the job cannot shift to any other queue but the multi level queue with the possibility of transfer [4]. Scheduling algorithm between these queues follows one of the following methods :

1- Fixed Priority Method :

It vests a fixed priority in each queue, and begins executing operations of the queue that holding high priority, upon completing all these operations the queue moves to the operation next in priority.

2- Time Slice Method :

It sets a fixed amount that gives the first queue a time of the processor, then moves the next queue applying as same as the first queue, i.e., the processor time is divided between the queues.

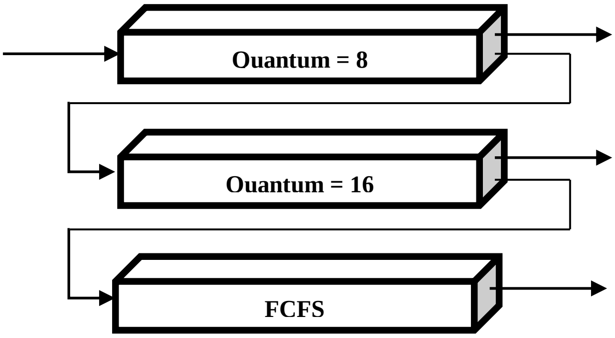

Figure (6) : Multilevel feedback queue

8.6 Multi-Level Queue

The notion of this algorithm is identical to the notion of multi-level feedback queue. The queue is divided into many queues with the difference that here we can move from one queue to another [4].

The system is to be considered as a set of queues, each one has a priority and follows a certain algorithm. This method characterizes with that the short operation takes a long time till the operations that takes long time completed, while the operations that their turn comes more than once without completing, they reach to the end of queue without completing and without effecting the short jobs.

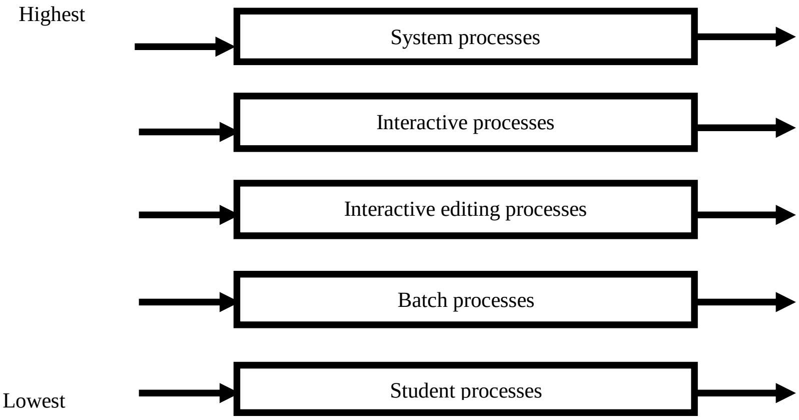

Figure (7) : Multilevel queue

8.7 Real Time Scheduling Algorithm

Real-Time system controls the external events that have their own timing requirements, therefore, real time system should be designed to respond to these events [1][9]. OS is designed to support the momentum, interaction, and sharing time which is differ for real time (RT) [5].

This system linked to a final appointment that is determined whether by onset time or completing time. The operation is classified as :

1- Hard Real Time: It should abide by the final appointments.

2- Soft Real Time: This requires that critical processes receive priority over less fortunate [1][9].

8.8 Two-Level Scheduling

The previous algorithms suppose that operations are existed in the main memory, but in case that there is no sufficient space for the operations, then algorithms is placed in the storing disc wholly or partially. Thus, the operation will be slower due to the increase of context switch time. For this problem solve to be solved, this algorithm is used. The scheduler selects to execute the next operation from the operations stored in the main memory.

High level scheduler removes operations that stayed for long time in the main memory, then shift it to the storing disc, vice versa [2]. When processing this change, low level scheduler undertakes scheduling of operations existing in the memory, and selecting the next operation to be executed.

Some variables should be observed when high level scheduler undertakes determining operations, and shifts them to the disc [2] [5] [10], these variables are :

1- Time Period the operation takes from its last swapping operation.
2- Time period it takes from CPU time.
3- Operation size.
4- Operation priority.

8.9 Multi-Processing Scheduling Algorithm

Operations scheduling is done on many processors, but scheduling of the operation becomes complex. The operations existing in the ready queue uses the available processor. There are some limitations on scheduling, for example; if there is an input and output unit linked to a processor by means of a single-carrier, we will find that operations use this unit should by scheduled in the processor that linked to this unit [5]. The way of scheduling this algorithm :

1- **Asymmetric Multiprocessing:**
Where one of the processors distributes jobs between other processors, this processor is called Master Server, the other processors process user's operations.

2- **Symmetric Multiprocessing:**
All user's operations are processed by one processor, they may placed in the ready queue which allocated to each processor.

3- **Processor Affinity:**
An operation could be processed once more in another processor, consequently; it does not make use of data existing in the cache. It is better this operation remains to be executed in the same processor where it begins, this is called Processor Affinity [7].

9. Summary

In this paper first of all we have discussed about scheduling and then various types of scheduling with the examples. Principally, algorithms scheduling can be arbitrary, with powerful impacts on system, efficiency and response time. The best of algorithms scheduling is to give high priority operations that require less time of execution.

10. Recommendations

On the light of this paper, we suggest a combination is to be done between two algorithms, which are SJF algorithm and RR algorithm in order to produce a new algorithm, seeking to treat the imperfections of both of them. The resultant algorithm gives us rapidly a processing in a lesser waiting time.

11. References

1. Abielmona,R.(2000).*Scheduling Algorithmic Rresearch*,12-15 .
2. Bajpai,A.,Guta,D.,Srivastava,H.,&Chawla,K.(2009),*cpu scheduling Algorithm* .
3. Bibi,S.,Azam,F.,&Chaudhry,Y.(2010).*Combinatory CPU Scheduling Algorithm*,International Journal of Computer Science and Information Security,Vol. 8, No. 7 .
4. Chand, P. S., Chaudhary, M. T., & Kumar, M. (2011). *Extended RR-scheduling algorithm*, *9*(5), 5–9.
5. Jeet,A.,&Ahmed,u.*Cpu scheduling Algorithm and their implementation*.
6. Meehean,J.T(2011).*Towards Transparent cpu Algorithms and their implementation*.1-17 .
7. Mostafa,S.M.,Rida,S.Z.,&Hamad,S.H.(2010).*Finding Time Quantum Of Round Robin Cpu Scheduling Algorithm In General Computing Systems Using Integer Programming*, *5*(October), 64–71.
8. Oyetunji,E.O.,&Oluleye,A.E.(2009).*Performance Assessment of Some CPU Scheduling Algorithms*, 1(1), 22–26.
9. Sattar,I.,Shahid, M. and Yasir,N.(2014).*Multi-Level Queue with Priority and TimeSharing for Real Time Scheduling*, International Journal Of Multidisciplinary Sciences And Engineering, Vol. 5, No.
10. Silberschatz,A.,&galvin,p.B. *operating system concepts*,13-17 .