SINGLE TIME-UNIT OPERATION SCHEDULING ALGORITHM

[1]Ibraheem A. Kateeb, Yasser A. Ahmed, [2]Amjad A. Jourishi
[1]Department of Computer Engineering, Qassim University, Qassim, 51452, Saudi Arabia
[2]Computer Science Department, Amman Arab University, Amman, Jordan
[1].kateeb@qu.edu.sa,,[2] hsanien@qu.edu.sa

ABSTRACT*- This research includes a study and analyzing scheduling algorithms for independent processors as an algorithm Longest Processing Time scheduling algorithm (LPT) and Shortest Processing Time scheduling (SPT), then compare it with the proposed algorithm Single Unit Operation (SUO) based on the measuring length scheduling of each of algorithms*.

Index terms-LPT, SPT, Scheduling Algorithm

I. INTRODUCTION

Process management within processors is one of the most important processes in computer systems. These processes appear to exist when you run a command or program, which gives each operation a number. These operations are carried out by the Central Processing Unit (CPU), one after another, by dividing the time between these processes and servicing each operation as necessary and its importance.

In general, processor scheduling is used to manage and perform tasks that are developed for a task. It is included in many statistical and mathematical positions, categorized into two types: the first, scheduling algorithms with no possibility of interrupting operations (no preemptive) and second scheduling algorithms with the possibility of interrupting operations (preemptive) [1].

The basic idea of scheduling algorithms is to increase the throughput and utilization of processors, reduce waiting time, return time, response time, achieves fairness between operations and reduces the length of the schedule [2].

This research includes the presentation of the general fundamentals and concepts of the scheduling algorithms for multiple processors, in particular, the independent process scheduling algorithms and their comparison with the proposed algorithm. Different cases will be studied and discussed to compare between the three algorithms.

II. Scheduling Problem

There is an important feature in the operating systems within the computers. Scheduling is optimization issues. The process is how to manage paths, and most of the solutions are got by using the process research methods, statistical and mathematical methods. Scheduling issues have been extensively studied in the fields of operations research and computer science through scheduling algorithms, due to the significant and rapid development of the computers, which its processors prioritize the implementation of operations as well as the implementation of multiple processes. The scheduling issue in how to prioritize a specific process from multiple processors, customize the wizard from multiprocessor and end execution after setting a time for each operation at the minimum possible time. A single operation cannot allocate more than one processor at a time, because each processor is assigned a process for executing it or remains idle.

Single-processor computer systems (Uniprocessor System) are the foundation for creating multiprocessor computing systems. Single-processor computing systems mean a single operation, and the rest of the operations are waiting in readiness row until the first operation is completed. In multi-processor computing systems, several operations are performed at the same time. This is done by distributing processes to existing processors and implementing them according to the precedence established through multiple processor scheduling algorithms [3]. The use of a large number of processors reduces the length of the schedule.

III. Scheduling Criteria

The processor scheduling algorithms have a different property, which prompts to favor an algorithm on the other, and there are bases and properties used to compare in the definition of a good and efficient algorithm and these exponents are [4]:

1. CPU Utilization: means how to make a processor permanently exploited and become busy all the time in performing operations as much as possible.
2. CPU Throughput: represents the number of fully implemented operations within a time unit.
3. Turnaround Time: represents the length of time that the operator enters the system and until the operation is completed.
4. Waiting Time: represents the time it takes to stay in the ready queue in order for the processor to be assigned for execution.
5. Response Time: represents the time period since the operation entered the system until the initial response in implementation.
6. Scheduling Length: is the time period that operations need to execute, in the case of multiple processors, the scheduling optimization is based on the length of the schedule.

Scheduling length is the most important measure to determine the efficacy of algorithms in the scheduling of multiple processors.

IV. Independent Tasks Scheduling Algorithms

We will show an algorithm of scheduling algorithms for a set of independent operations, with an optional number of processors and execution time for each set of process.

1. Longest Processing Time Scheduling Algorithm (LPT)

This algorithm is one of the basic algorithms in the scheduling of independent processes [5]. In this algorithm, the process with the longest execution time and all operations at one level determine the precedence of operations. The main steps of the algorithm are:

1. Start of the algorithm
2. Determine the priority of the process based on the greater implementation time.
3. Process Selection
4. Choose the appropriate processor to perform the operation
5. Customizing the wizard for the process
6. Allocating the wizard for the process

7. When all operations are not scheduled, go to step 3 or
8. Terminate

2. Shortest Processing Time Scheduling Algorithm (SPT)

This algorithm is the reverse of the LPT algorithm in the scheduling of the operations. In this algorithm, the process with the shortest execution time and all operations in one level [6] determines the precedence of operations. The main steps of the algorithm are:

1. Start of the algorithm
2. Determine the priority of the process based on the smallest implementation time.
3. Process Selection
4. Choose the appropriate processor to perform the operation
5. Customizing the wizard for the process
6. Allocating the wizard for the process
7. When all operations are not scheduled, go to step 3 or
8. Terminate

3. The proposed Algorithm: Single Unit Operation (SUO)

The new proposed algorithm is to schedule independent operations on an optional number of multiple processors; it will be used on a set of independent processes and execution times for equal or different operations. Processors begin with operations with significant implementation times and the processor's specialization with the possibility of interrupting the process and implementing a unit one of them then runs a comparison of all operations and implements the largest execution time of operations as well as implements one unit and so on.

The proposed algorithm steps are as below:

1. Start of the algorithm
2. Select the process that has the greatest time to implement and allocate the idle processor and implement one unit of it.
3. Compare the remaining processes and select the process that has the largest execution time of all the processes and allocates the processor for it.
4. When all operations are not scheduled, go to step 3 or
5. Terminate

The difference between the proposed algorithm and the algorithms (LPT) and (SPT) is that the second and third are concentrated whereas in the proposed algorithm, implements a Single Unit Operation (SUO) the operation can be interrupted during execution and no idle time for the processor, while other algorithms carry out the whole process, the operation cannot be interrupted.

Two cases will be discussed to compare LPT and SPT algorithms with the proposed algorithm, where the first case is divided into two parts, the first part containing two processors, where the second part containing three processors. The second case, we will change the execution times for the operations and at the same time perform with two processors and three processors as well.

V. DISCUSSION AND CASES

1. The first case (two processors):

I. Two CPUs and five different times' processes

In this case, we have two processors and five processes at different times as in table 1, LPT and STP will be applied when compared to SUO algorithm

Table 1: Process and execution order

Process	Time Unit	Execution order LPT	Execution order SPT
P1	5	3	3
P2	6	2	4
P3	4	4	2
P4	2	5	1
P5	7	1	5

Table 2: LPT algorithm (two processors)

CPU1	P5	P5	P5	P5	P5	P5	P5	P3	P3	P3	P3	P4	P4
CPU2	P2	P2	P2	P2	P2	P2	P1	P1	P1	P1	P1	Φ	Φ
	1	2	3	4	5	6	7	8	9	10	11	12	13
	Time unit												

Table 3: SPT algorithm (two processors)

CPU1	P4	P4	P1	P1	P1	P1	P1	P5	P5	P5	P5	P5	P5	P5
CPU2	P3	P3	P3	P3	P2	P2	P2	P2	P2	P2	Φ	Φ	Φ	Φ
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	Time unit													

First, LPT algorithm is applied to two processors, according to table 1; the Gantt chart for the schedule will be as in table 2.

The first processor (CPU_1) is assigned to P_5, which is the longest process, while the second processor (CPU_2) is assigned to P_2, which has a shorter execution time than P_5, which is the second online to be executed. P_1 is allocated to CPU_2 after the completion of P_2 because CPU_1 is still busy with P_5. (CPU_1) is assigned to P_3, after the completion of P_1 and P_3, P_4 is assigned to (CPU1) and (CPU_2) is static idle (Φ), as in table 2.

When STP algorithm is applied to the same process to two processors, the Gantt chart for the schedule will be as in table 3.

The first processor (CPU_1) is assigned to P_4, which is the shortest process, while the second processor (CPU_2) is assigned to P3, which has a longer execution time than P_4, which is the second online to be executed. P_1 is allocated to CPU_1 after the completion of P_4 because CPU_2 is still busy with P_3. (CPU_2) is assigned to P_2, after the completion of P_1 and P_2, P_5 is assigned to (CPU_1), and (CPU_2) is static idle (Φ), as in table 3.

Now, applying the proposed SUO algorithm to two processors, the Gantt chart for the schedule will be as in table 4.

Table 4: SUO algorithm (two processors)

CPU1	P5	P5	P5	P5	P2	P2	P5	P5	P2	P4	P4	P2
CPU2	P2	P2	P1	P1	P3	P3	P1	P1	P3	P3	P1	P5
	1	2	3	4	5	6	7	8	9	10	11	12
	Time unit											

1. The implementation begins with (CPU_1) assigned to P_5, as it has the longest execution time and implements one-time unit, (CPU_2) is allocated to P_2 (the second longest time) and implements one-time unit.
2. Then allocated (CPU_1) to P_5 again and the one-time unit is implemented, and (CPU_2) is assigned to P_2 or P_1 because they have the second largest (equal time) implementation time and the one-time unit is implemented. (Assuming that the (CPU_2) was assigned for P_2).
3. Assign (CPU_1) to P_5 and (CPU_2) to P_1, implements one-time unit each.
4. The processes (1, 2, 3, 5) have the greatest implementation time (equal) and assuming that the (CPU_1) is allocated P_2 and (CPU_2) to P_3.
5. Assign (CPU_1) to P_1 and (CPU_2) to P_5 then implements a one-time unit for both.
6. All processes have equal implementation time, assuming that (CPU_1) is allocated to P_1 and implements a one-time unit, (CPU2) is dedicated to P_5, and one-time unit is implemented.
7. Processes (4, 3, 2) have the longest implementation time (equal) and assuming that (CPU_1) is allocated to P_2 and (CPU_2) to P_3.

Thus, all operations continue to be carried out as in table 4.

II. Two CPUs and five processes-two the same time

In this case, we have two processors five processes with different times but two are the same value as in table 5, LPT and STP will be applied when compared to SUO algorithm.

Table 5: Process and execution order

Process	Time Unit	Execution order LPT	Execution order SPT
P1	5	2	3
P2	3	4	1
P3	4	3	2
P4	6	1	4
P5	3	4	1

When we apply the algorithm (LPT) to two processors, the Gantt chart for the schedule will be as in table 6.

Table 6: LPT algorithm (two processors)

CPU1	P4	P4	P4	P4	P4	P4	P2	P2	P2	P5	P5	P5
CPU2	P1	P1	P1	P1	P1	P3	P3	P3	P3	Φ	Φ	Φ
	1	2	3	4	5	6	7	8	9	10	11	12
	Time unit											

Table 7: SPT algorithm (two processors)

CPU1	P2	P2	P2	P3	P3	P3	P3	P4	P4	P4	P4	P4	P4
CPU2	P5	P5	P5	P1	P1	P1	P1	P1	Φ	Φ	Φ	Φ	Φ
	1	2	3	4	5	6	7	8	9	10	11	12	13
	Time unit												

When we apply the algorithm (SPT) to two processors, the Gantt chart for the schedule will be as in table 7.
When we apply the algorithm (SUP) to two processors, the Gantt chart for the schedule will be as in table 8.

Table 8: SUP algorithm (two processors)

CPU1	P4	P4	P4	P4	P1	P5	P5	P3	P3	P1	P5
CPU2	P1	P1	P3	P3	P2	P2	P1	P4	P4	P2	Φ
	1	2	3	4	5	6	7	8	9	10	11
	Time unit										

2.The second case (three processors)

I. Two CPUs and five different times' processes

In this case, we have three processors and five processes with different times as in table 9; LPT and STP will be applied when compared to SUO algorithm.

Table 9: Process and execution order

Process	Time Unit	Execution order LPT	Execution order SPT
P1	5	1	1
P2	6	1	2
P3	4	2	1
P4	2	2	1
P5	7	1	2

Appling LPT, SPT and SUO algorithms to three processors, the Gantt chart for the schedule will be as in tables 10, 11, and 12, respectively:

Table 10: LPT algorithm (three processors)

CPU1	P5	P5	P5	P5	P5	P5	P5	Φ	Φ
CPU2	P1	P1	P1	P1	P1	P1	P4	P4	Φ
CPU3	P2	P2	P2	P2	P2	P3	P3	P3	P3
	1	2	3	4	5	6	7	8	9
	Time unit								

Table 11: SPT algorithm (three processors)

CPU1	P4	P4	P2	P2	P2	P2	P2	P2	Φ	Φ	Φ
CPU2	P3	P3	P3	P3	P5	P5	P5	P5	P5	P5	P5
CPU3	P1	P1	P1	P1	P1	Φ	Φ	Φ	Φ	Φ	Φ
	1	2	3	4	5	6	7	8	9	10	11
	Time unit										

Table 12: SUO algorithm (three processors)

CPU1	P5	P5	P5	P5	P5	P5	P4	P4
CPU2	P2	P2	P2	P2	P1	P1	P1	P3
CPU3	P1	P1	P3	P3	P3	P2	P2	P5
	1	2	3	4	5	6	7	8
	Time unit							

II. Two CPUs and five processes-two the same time

In this case, we have three processors five processes with different times but two are the same value as in table 13, LPT and STP will be applied then compared to SUO algorithm.

Table 13: Process and execution order

Process	Time Unit	Execution order LPT	Execution order SPT
P1	5	2	3
P2	3	4	1
P3	4	3	2
P4	6	1	4
P5	3	4	1

Appling LPT, SPT and SUO algorithms to three processors, the Gantt chart for the schedule will be as in tables 14, 15, and 16, respectively:

Table 14: LPT algorithm (three processors)

CPU1	P2	P2	P2	P1	P1	P1	P1	P1	Φ
CPU2	P5	P5	P5	P4	P4	P4	P4	P4	P4
CPU3	P3	P3	P3	P3	Φ	Φ	Φ	Φ	Φ
	1	2	3	4	5	6	7	8	9
	Time unit								

Table 15: SPT algorithm (three processors)

CPU1	P3	P3	P2	P2	P2	P2	P2	P2	Φ	Φ
CPU2	P4	P4	P5	P5	P5	P5	P5	P5	P5	P5
CPU3	P1	P1	P1	P1	Φ	Φ	Φ	Φ	Φ	Φ
	1	2	3	4	5	6	7	8	9	10
	Time unit									

Table 16: SUP algorithm (three processors)

CPU1	P4	P4	P4	P4	P4	P3	P3
CPU2	P1	P1	P1	P5	P5	P5	P5
CPU3	P3	P3	P2	P2	P1	P1	P4
	1	2	3	4	5	6	7
	Time unit						

VI. RESULTS OF ALL CASES:

The results of the comparison in both cases, different times and different times but two are the same, are shown in table 17 and 18, respectively:

Table 17: Results of different times

	LPT	SPT	SUO
Two CPUs	13	14	12
Three CPUs	9	11	8

Table 18: Results of different times but two are the same

	LPT	SPT	SUO
Two CPUs	12	13	11
Three CPUs	9	10	7

It is very clear that SUO has the least time in all cases discussed.

VII. CONCLUSION

1. The length of the schedule for SUP algorithm is less than (STP) and (LPT) algorithms.
2. The efficiency of SUP algorithm was not affected if the execution times of the independent operations were changed.
3. We note that the increase in the number of processors helps to reduce the length of the scheduling for all algorithms.

Based on the results shown above, it is recommended that we use the proposed algorithm if we have independent and preempt operations (preemptive).

REFERENCES

[1] E.G. Coffman, "Ideal preemptive schedules on two processors" J. Sethuraman2, V.G. Timkovsky, Digital Object Identifier (DOI) 10.1007/s00236-003-0119-6, Canada, 2003.

[2] Hwang, K. and Briggs, F.A., "Computer Architecture And Parallel Processing" , Mc Graw-Hill, Inc., New York, 1984.

[3] Oliver Sinnen," *Task Scheduling For Parallel Systems*", John Wiley , Canada,(2007).

[4] Vincent T'kindt and Jean-Charles Billaut, Translator Henry Scott, " *Multicriteria Scheduling*" ,Springer , Germany, (2006) .

[5] Peng,D. T, Shin, K.G. and Abdel Zaher ,T. F., "*Assignment and scheduling Communicating Periodic Tasks in Distributed Real-Time Systems*", IEEE ,Transaction on Software Engineering,Vol.23,No.12, 1997

[6] David A. Patterson, John L. Hennessy, "*Computer Organization and Design: the Hardware/Software Interface*", Morgan Kaufmann. 2014