International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

September 2023 Issue 56 Volume 15 Number 3 Pages 77-85

STUDY AND STATISTICAL ANALYSIS OF SINGLE PROCESSING OPERATING SYSTEM: EFFECTIVENESS AND SIMULATION

M.A. Mohammed S.K. Mohsin S.J. Mohammed

Mechanical Engineering Department, University of Technology, Baghdad, Iraq
maysoon.a.mohammed@uotechnology.edu.iq, Shaymaa.k.mohsin@uotechnology.edu.iq,
arah.j.mohammed@uotechnology.edu.iq

Abstract- The processor system considers as the most important units in the operating systems on which many processes depend. Therefore, the efficiency of the operating system is related to the efficiency of the processor system. There are different types of processor systems, one of which is single and multi-processor systems. In this paper, we focused on the single processor system vs multi-processor system in terms of studying, analyzing, and simulating the environment of a single and multi-processor system. The purpose of this research is to analyze and evaluate the work of a single processor system statistically and simulate an environment related to it to show the importance of such systems. The results show that, the single processing operating system can withstand the overhead more than the other systems. The statistical analysis is done using SPSS software and for verification, validation and tracking the result theoretically, simulations is made out using Arena simulation software.

Keywords: Single Processing System, Operating System, Processors, Central Processing Unit, Simulation.

1. INTRODUCTION

A processor is a small chip or a logic circuit which performs tasks on an external stream of data source such as memory data or other external data. The processor can be called as a microprocessor or Central processing unit (CPU), however, processor generally understood as CPU [1]. A processor is the main unit in the computer which responsible of manipulating the main commands that derive to the computer [2], The processor contains the main and basic units of hardware which are arithmetic and I/O units, these units perform and allocate instructions for other components executing on the computer [3]. The processor works as an integrated system to process instructions and perform various tasks in the operating system. A processor system can contain either one processor or more than one processor. The number of processors present in the system determines the number of tasks that are executed in the operating system. So, there are single-processor or multi-processor systems, and each type of these systems has importance in being responsible for completing and executing tasks.

In this research, we focused on the single-processor system to give it its importance as the multi-processor is important, also because we did not find much research or articles in this field on the internet. A single processor system has one processor; therefore, the system selects only one process from the ready queue (even if there are multiple processes) to be executed at a time and then returns to select another process and so on. Most computers with general purposes have single-processor systems as they are generally used [4]. Figure 1 illustrates the system of a single processor [5].

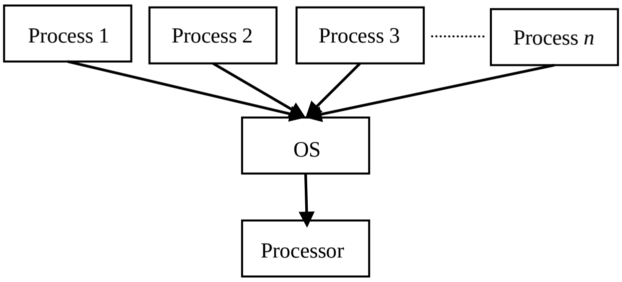

Figure 1. Design of Single Processor System

In Figure 1, multi-processes that must be executed, but for the system has only one processor; therefore, every time just one process can be worked on. The partitions of this paper will be as follow:

The next section illustrates the related work, then the section follow is to the methodology of this research, after, analysis and simulation section to shows the analytical and simulation results, finally, presents the conclusions in the last section.

2. RELATED WORK

Single processor system is an essential operating system routine, which is the process of occupied the processor to a specific mission for a specific amount of time [6]. Studying the effect of the single processor system needs focus and attention due to the lack of research on the Internet that has been discussing this topic recently. In the last decades, many researchers produce articles to investigate with the system of single processor to find solutions for the problems with this system.

Most of researchers focused on the multi-processor system in their articles, therefore, in this research, we preferred to give the preference to the single processor system than multi-processor system by presenting a study, analysis and simulation regarding this system.

Proposed a single processor to export a tenet set for utility elliptic curve cryptography (ECC) in view of excess marked digit portrayal with the aid of appearing an unengaged point augmentation utilizing focuses in relative direction in much less time [7]. In article [8], the researchers produced a mechanism to synchronize multithreading with a real time collector of garbage in a single processor system. The mechanism used in the above research guaranteed to maintain the precise roots data, to apply an effective write-barriers, to increment the scan of root and also to ensure rapid pre-emption time of garbage collector effectiveness. A scheduling algorithm to give priority to processes that occupied the single processor using weighted round-robin has been shown in [6]. This algorithm has been enhanced the classic round-robin performance. However, the use of this algorithm with multi-processor system would be more efficient.

Evaluated the danger of the secret processor channels between virtual machines [7]. They created a framework called CCCV to make a secret channel to transfer information depending on the loads of the single processor. In [10], researchers have also moved towards scheduling tasks in a single-processor system where they analyze the effects of different degradation schemes and devise optimal scheduling policies that reduce the expected range of waiting time and, for some models, policies that reduce configuration variance.

In report presented by [11], they investigated the lattice Boltzmann method for each "off-the-shelf" computer systems and high-performance computing (HPC) systems that have been specifically tailored to meet specific needs, to improve the performance of the characteristic of the single processor system for different data layouts. The findings of their study indicated that vector systems have the potential to perform remarkably well on COTS architectures, surpassing them by a factor of ten. A multi-processor system is used in [12] to present the factors measurements of sprite operating system. They need to five processors to achieve their goal which is, increase the throughput of the system linearly with the size of the system. They justified new designs to measure the performance of two different ports of the Choices object-oriented operating system. They validated their work using a TCP communication application.

An investigation on the single CPU scheduling algorithms is made by [13]. The goal of their study is to examine the CPU scheduler and how it impacts on the scheduling algorithms. In simulation and analysis part, an excel software and user defined simulation used to analyze and simulate a system proposed by [14]. In [15], they used calculations and construction simulations to investigate the 3D printing by robotic cooperation. The [16] used a simulator called Contiki/Cooja to simulate the risks of attacking a router that can damage or effect on Internet of Things (IoT) network. The next section is to provide an analysis of single processor system to evaluate the performance of this system to manipulate different data types. This analysis done by using SPSS software.

3. METHODOLOGY AND EXPERIMENTAL PART

As we mention in the last sections, single processor system responsible to perform one task from the ready queue at a time, while the multi-processor system can perform more than one task at a time because it contains more than one processor. From this, we can conclude that the performance of the multi-processor system is higher than single-processor system where the preformance rate will be higher too. This is true when only one single processor system is used versus multi-processor system. When comparing the performance of single and multi-processor systems, it is important to remember that a multiprocessor system can outperform a single processor system by a factor of *n*. However, this advantage will be less than *T* (where *T* is the preformance rate of the individual processors in the multiprocessor system). [17]. In this part, we made an experiment using two single processor system versus one multi-processor system contains two processors, a First Come First Server algorithm is applied in this experiment and the data used is taken from internet [18] and personal data. By considering 6 processes with different arrival-times and burst-times are shown in Table 1. Tables 2 and 3 show the distribution of the processes to single and multi-processor systems.

Table 1. Entry processes

Processes	Arrival-Times	Burst-Times
A	0	5
B	10	7
C	15	2
A1	0	3
B1	5	7
C1	7	3

Table 2. Execution of processes in two single processor systems

Sys	Process	Arrival Times	Burstt Times	Start	End	Waiting Times
Sys 1	A	0	21	0	21	0
	F	20	10	21	31	1
Sys 2	B	5	3	5	8	0
	C	10	6	10	16	0
	D	15	2	16	18	1
	E	18	5	18	23	0
	G	22	3	23	26	1

Table 3. Execution of processes in multi-processor systems with two processors

Process	Arrival Times	Burst Times	Processor 1	Processor 2	Start	End	Waiting Times
A	0	5	A		0	5	0
A1	0	3		A1	0	3	0
B1	5	7	B1		5	12	0
C1	7	3		C1	7	10	0
B	10	7		B	10	17	0
C	15	2	C		15	17	0

To calculate the turnaround time and waiting time for both systems [19]:

$$\text{Turnaround Time} = \text{End of Service} - \text{Arrival Time} \quad (1)$$

$$\text{Waiting Time} = \text{Turnaround Time} - \text{Burst Time} \quad (2)$$

From Equations (1) and (2) [19], the average turnaround time from both systems is 5 ms and there is no waiting time for all tasks.

The results are similar because the number of tasks is small, the arrival time is far apart, and the burst time is low, but this does not negate the fact that when the number of tasks increases with a close arrival time and a large burst time (Table 4), it causes overhead on one of the processors in the multi-processor system and thus reduces its throughput because (as shown the next example) when the burst time for a particular task is very large, this leads to one processor being busy with this task and leaving the rest of the tasks to the other processor, and therefore this preoccupation with sharing resources for all these tasks causes overhead on the processor. Whereas in the case of a single processor system, it was basically designed to withstand this overhead as it does not accept only one task at a time.

Table 4. Processes with different arrival and burst times

Process	Arrival Times	Burst Times
A	0	21
B	5	3
C	10	6
D	15	2
E	18	5
F	20	10
G	22	3

Table 5. Execution of processes in two single processor systems

Sys	Process	Arrival Times.	Burstt Times	Start	End	Waiting Times
Sys1	A	0	5	0	5	0
	B1	5	7	5	12	0
	C	15	2	15	17	0
Sys2	A1	0	3	3	3	0
	C1	7	3	10	10	0
	B	10	7	17	17	0

Table 6. Execution of processes in multi-processor systems with two processors

Process	Arrival Times	Burst Times	Processor 1	Processor 2	Start	End	Waiting Times
A	0	21	A		0	21	0
B	5	3		B	21	31	1
C	10	6		C	5	8	0
D	15	2		D	10	16	0
E	18	5		E	16	18	1
F	20	10	F		18	23	0
G	22	3		G	23	26	1

Tables 5 and 6 show the distribution of the processes to single and multi-processor systems. In this example the average turnaround and waiting time for both system is equal which is 8 ms and 0.4 ms respectively. However, despite the equal times, the overhead on the processor 2 in the multiprocessor system is greater because it works within one system with the processor 1, while each single processor system works as a single unit designed to withstand the overhead. We can note that the average-turnaround-time and average-waiting-time are the same for both systems, this because we use the same number of tasks, same burst time and same arrival time for both systems. Hence, we conclude that increasing the tasks number with a large burst time and close arrival time reduces the gain of the processor and thus reduces the throughput of the multi-processor system.

Thus, the performance rate of n single processor systems is greater than of a multi-processor system. The coming section to Analyze these results statistically.

4. STATISTICAL ANALYSIS RESULTS

To test the results of the experiments above, a significant level is needed to compare with the results. The significance level at which the Wilcoxon signed-rank test is used is 0.05. This statistic can be used to assess whether differences between two populations' means are significant. The Wilcoxon Signed Rank test was selected as the data is in pairs and comes from the same population. By using the Wilcoxon test, we assume that all measures for comparison, including turnaround-time and waiting-time in this analysis, have a symmetric shape that is not normally distributed. Two hypotheses are used for the purpose of comparison: the alternative hypothesis, known as H_1, and the null hypothesis, known as H_0, as shown in Equations (3) and (4) [20].

$$H_0 : \mu_1 - \mu_2 = 0 \tag{3}$$

$$H_1 : \mu_1 - \mu_2 \neq 0 \tag{4}$$

where, μ_1 is the median of results for single processor systems, and μ_2 is the median of results for multi-processor system.

If the results between both systems have no significant difference that means H_0=0. In contrast, H_1≠0 indicates that there is a significant difference between the two systems results as shown in Tables 7, 8 and 9 which are the Wilcoxon Rank Test, hypothesis test summary for average turnaround time and hypothesis test summary for waiting time respectively between single processor and multi-processor systems while Table 10 is the correlations between average turnaround and waiting times after made different experiments with different processes, different arrival and burst times.

Table 7. Wilcoxon Rank Test

Wilcoxon Signed Ranks Test			
Ranks	*N*	Mean Rank	Sum of Ranks
TATMSP/TATSPS-Negative Ranks -Positive Ranks -Ties -Total	2^a 4^b 4^c 10	3.50 3.50	7.00 14.00
WTMSP/WTSPS-Negative Ranks -Positive Ranks -Ties -Total	2^D 4^e 4^f 10	3.50 3.50	7.00 14.00
Test Statistics[a1]	TATMSP TATSPS	WTMSP WTSPS	
Z Asymp. Sig.(2-tailed)	$-.816^{b1}$ .414	$-.816^{b1}$ .414	

where,

a. TATMSP<TATSPS

b. TATMSP>TATSPS

c. TATMSP=TATSPS

d. WTMSP<WTSPS

e. WTMSP>WTSPS

f. WTMSP=WTSPS

a1. Wilcoxon Signed Ranks Test

b1. Based on Negative Ranks

Table 8. Hypothesis summary for average turnaround time

Nonparametric Tests			
Hypothesis Test Summary			
Null Hypothesis	Test	Sig.	Decision
The median of differences between TATSPS and TATMSP equals 0.	Related-Samples Wilcoxon Signed Rank Test	.414	Retain the null hypothesis.
Asymptotic significances are displayed. The significance level is .05.			

Table 9. Hypothesis summary for average waiting time

Nonparametric Tests			
Hypothesis Test Summary			
Null Hypothesis	Test	Sig.	Decision
The median of differences between WTSPS and WTMSP equals 0.	Related-Samples Wilcoxon Signed Rank Test	.414	Retain the null hypothesis.
Asymptotic significances are displayed. The significance level is .05.			

Table 10. Correlations Statistics

Correlations				
Descriptive Statistics	Mean	Std. Deviation	N	
TATSPS	59.1000	33.03012	10	
TSTMSP	59.3000	33.20994	10	
WTSPS	48.5000	30.94524	10	
WTMSP	48.7000	31.12716	10	
Correlations	TATSPS	TSTMSP	WTSPS	WTMSP
TATSPS-Pearson Correlation -Sig.(2-tailed) -N	1 10	1.000** .000 10	.999** .000 10	.999** .000 10
TATMSP-Pearson Correlation -Sig.(2-tailed) -N	1.000** .000 10	1 10	.999** .000 10	.999** .000 10
WTSPS-Pearson Correlation -Sig.(2-tailed) -N	.999** .000 10	.999** .000 10	1 10	1.000** .000 10
WTMSP-Pearson Correlation -Sig.(2-tailed) -N	.999** .000 10	.999** .000 10	1.000** .000 10	1 10
** Correlation is Significant at the 0.01 level (2-tailed)				

The retain hypothesis is because the negative ranks of Wilcoxon Test, where is the median differences between both system is 0, that means no significant differences between both systems in the statistical analysis. This statistical analysis is done to analyze the results that produced from the experiments not to analyze the overhead or throughputs of both systems. These measurements known with real experiments.

The average-waiting-time and average-turnaround-time are the chosen comparative measures,table 10 shows the correlations between average turnaround and waiting times after made different experiments with different processes, different arrival and burst times. The software that used to produce the analysis results is SPSS software.

5. SIMULATION RESULTS

To validate the results that conducted from the previous section, a simulation method is named Discrete Event Simulation (DES) is chosen as the validation process. This simulation is chosen due to its characteristic that model an event in discrete time using queueing theory. Such process is similar to processor system. This simulation made using Arena simulation software. Simulation is considered a standard benchmarking and analytical technique. In several situations and instances practitioners such as [21] stated that simulation is one of the most widely use techniques used in mimicking scientific and engineering ideas or concepts.

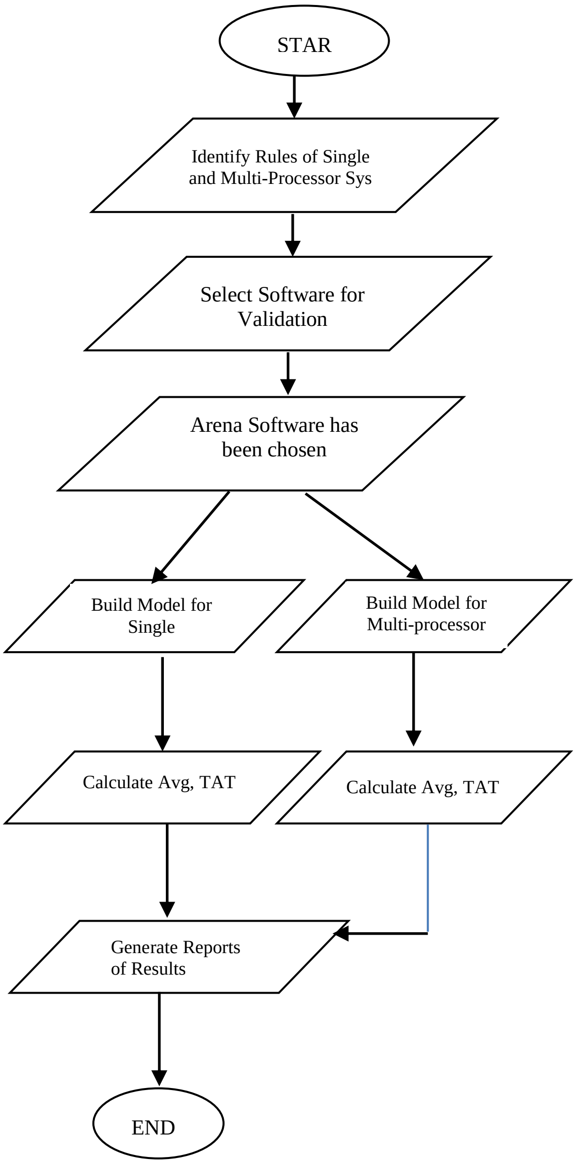

Figure 2. Framework of Arena Software Validation

For instance, Arena software is DES and automation software developed by Rockwell Automation. For the validation purposes, two simulation models were developed using Arena software version 19: single-processor-system and multi-processor-system. The purpose of developing these models is to understand the performance of the systems, validate the results of both systems and to visualize the process, so that it be easy for the reader to contain the idea of this study. Each simulation model consists of an arrival process; two systems with single processor (resource) and one system with two processors (resources). After developing the two simulation models, black box testing is done to ensure that the developed model is correct. The conceptual framework of Arena software for single and multi-processor system is presented in Figure 2. The conceptual models have been depended on for single and multi-processor systems in this study are shown in Figures 3 and 4, respectively.

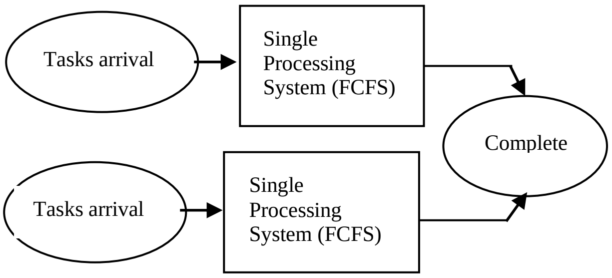

Figure 3. Conceptual Model of Single Processor System

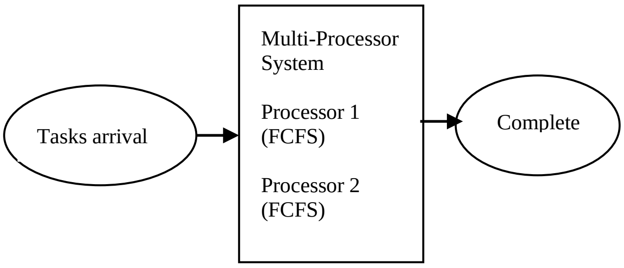

Figure 4. Conceptual Model of Multi-Processor System

Figures 5 and 6 illustrate the model of single and multi-processor systems respectively in Arena software. The figures are visualization to the work of both systems where Figure 5 shows two single systems with one processor, the processes arrive randomly with burst times, these processes occupy the idle processor, then after the processing is complete the process leave the system. The processes executed with First come First serve algorithm, which means the first process in the ready-queue will be chosen by the system for processing, then the second one and so. The principles of the work for both systems are the same, same processes, same arrival times, same burst times, and same algorithm.

After three replications of simulation using Arena, the results are conducted for number in and out of tasks that executed in the processor and usage of resources as shown in Tables 11 and 12 which illustrated the number in and number out of processes for single and multi-processor systems, respectively. Tables 13 and 14 illustrated the usage of resources for single and multi-processor systems respectively. Figures 7, 8, 9 and 10 shows the charts number in, number out and usage of resources for single and multi-processor systems, respectively.

Table 11. Number in and number out of processes for single processor system

Number In				
	Average	Half Width	Min Average	Max Average
Single Processor 1	30.0000	4.30	29.0000	32.0000
Single Processor 2	36.0000	7.45	33.0000	39.0000
TAT	19.6667	1.43	19.0000	20.0000
Number Out				
	Average	Half Width	Min Average	Max Average
Single Processor 1	8.0000	2.48	7.0000	9.0000
Single Processor 2	11.6667	1.43	11.0000	12.0000
TAT	6.6667	11.74	3.0000	12.0000

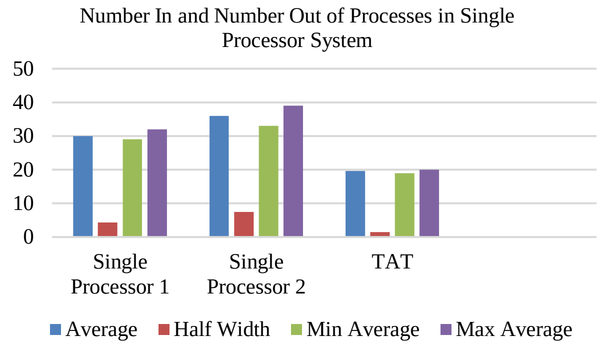

Figure 7. Number in and number out for single processor system

Table 12. Number in and number out of processes for multi-processor system

Number In				
	Average	Half Width	Min Average	Max Average
Multi-Processor	66.0000	4.30	65.0000	68.0000
TAT	19.3333	2.87	18.0000	20.0000
Number Out				
	Average	Half Width	Min Average	Max Average
Multi-Processor	19.6666	1.43	19.0000	20.0000
TAT	8.3333	7.99	6.0000	12.0000

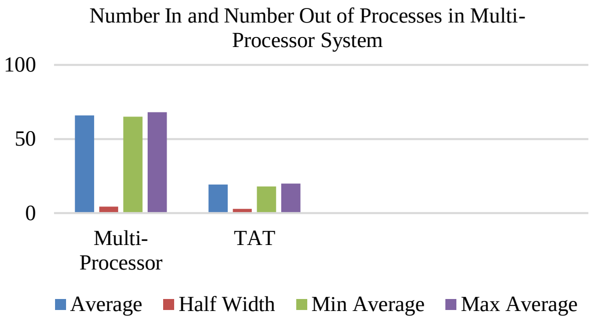

Figure 8. Number in and number out for multi-processor system

From Tables 13 and 14, it is obvious that there is a convergence in the number of tasks in and out to the processors in both systems, with a very small difference of 1 for the single processor system, and this is due to the overhead on the multi-processor system, which reduced its throughput. The usage of resources for both systems almost the same.

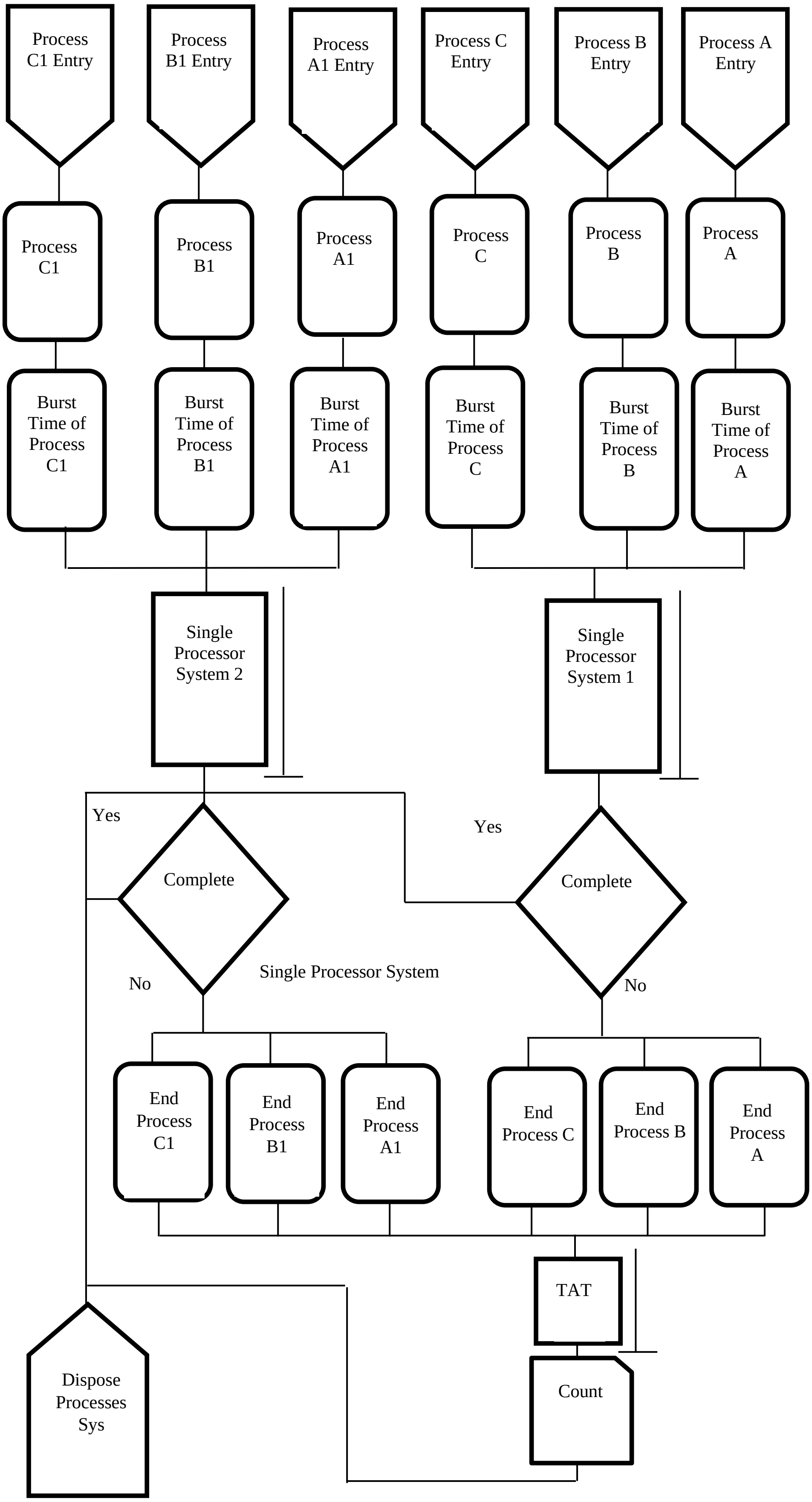

Figure 5. Arena Model of Single Processor System

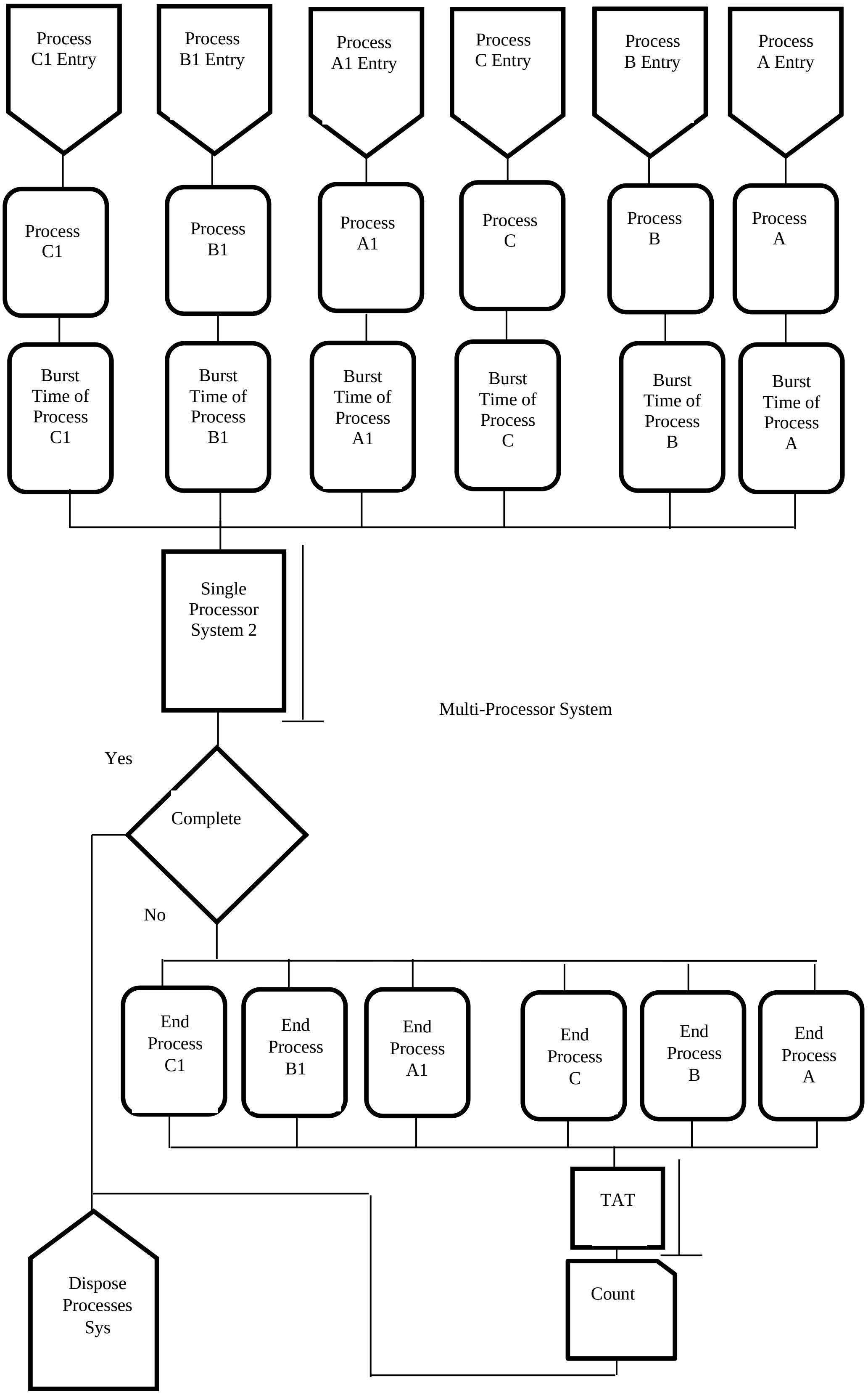

Figure 6. Arena Model of Multi-Processor System

Table 13. Usage of resources of single processor system

Resource of single processor
Usage
Total Number Seized

	Average	Half Width	Min Average	Max Average
Resource 1	7.6667	11.74	4.0000	13.0000
Resource 2	20.66667	1.43	20.0000	21.0000

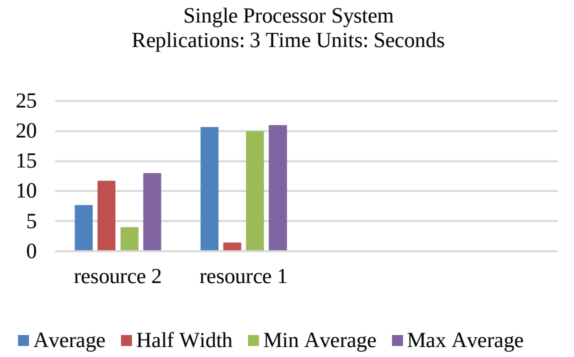

Figure 9. Resource usage in single processor system

Table 14. Usage of resources of multi-processor system

Resource of multi-processor				
Usage				
Total Number Seized				
	Average	Half Width	Min Average	Max Average
Processor 1	20.6667	1.43	20.0000	21.0000
Processor 2	20.6667	1.43	20.0000	21.0000
Resource 2	9.3333	7.99	7.0000	13.0000

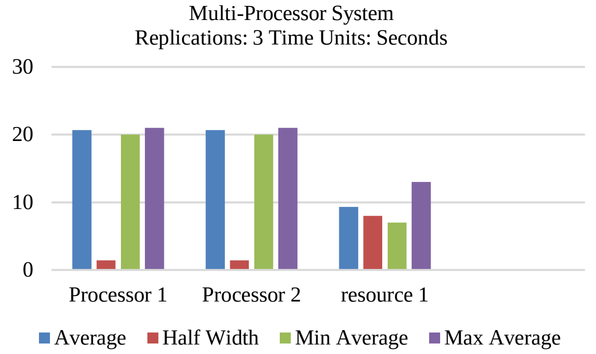

Figure 10. Resource usage in multi-processor system

6. CONCLUSION

In this research, a study was made about the importance of the single and multi-processor system in handling tasks in the operating system. After conducting some experiments for different missiones with different arrival times and different burst times and doing statistical analysis and simulation for the single and multi-processor systems environment, we concluded that the single-processor system can implement tasks and share resources completely and independently. If more than one single-processor-system is used versus a multi-processor system, this will Increases throughput of single processor system versus multi-processor system.

In general, the results obtained from the different experiments are almost similar with very little difference for the single processor system. The difference between the two systems which is the overhead affected the performance of multi-processor system where the processors in multi-processor system cooperate to perform tasks and share resources at the same time [22]. When the first task has high burst time then the first processor will be busy to perform this task while the other one performs the other tasks especially if the total burst times of these tasks is less than the first one.

The second processor will be under overhead of performing processes and sharing resources. This overhead occurs because the multi-processor system designed to cooperate, while in n single processor system such overhead will not occur because each system designed to work as a complete unit.

REFERENCES

[1] D. Anderson, T. Shanley, "Pentium Processor System Architecture", Addison-Wesley Professional, p. 464, USA, 1995.
[2] G.M. Amdahl, "Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities", The Spring Joint Computer Conference, pp. 483-485, 18-20 April 1967.
[3] G. Tomsho, "Guide to Operating Systems", Cengage Learning, p. 688, USA, 2020.
[4] A. Silberschatz, P. Baer Galvin, G. Gagne, "Operating System Concepts 10Edition", Wiley, p. 992, USA 2018.
[5] A.S. Tanenbaum, H. BOS, "Modern Operating Systems", PEARSON, p. 1136, UK, 2015.
[6] M.A. Mohammed, "Priority and Dynamic Quantum Time Algorithms for Central Processing Unit Scheduling", Computer Systems and Software Engineering, Vol. Doctorate, p. 220, 2018.
[7] D. Bochu, I. Prakash, K. Jyothi Jeevana, "ECC Processor for Design and Implementation of High-Speed RSD", Turkish Journal of Computer and Mathematics Education (TURCOMAT), Vol. 10, pp. 787-795, 2019.
[8] F. Siebert, "Real-Time Garbage Collection in Multi-Threaded Systems on a Single Processor", The 20th IEEE Real-Time Systems Symposium (Cat. No. 99CB37054), pp. 277-278, 1999.
[9] K.L. Flower, N. Sharma, "CPU Load Based Threat Prevention Framework for Covert Channel Attacks", Turkish Journal of Computer and Mathematics Education (TURCOMAT), Vol. 11, pp. 866-878, 2020.
[10] S. Browne, U. Yechiali, "Scheduling Deteriorating Jobs on a Single Processor", Operations Research, Vol. 38, pp. 495-498, 1990.
[11] G. Wellein, T. Zeiser, G. Hager, S. Donath, "On the Single Processor Performance of Simple Lattice Boltzmann Kernels", Computers and Fluids, Vol. 35, pp. 910-919, 2006.
[12] J.H. Hartman, J.K. Ousterhout, "Performance Measurements of a Multiprocessor Sprite Kernel", USENIX Summer, pp. 279-288, 1990.
[13] P.P. Acharjya, S. Koley, S. Barman, R. Mukherjee, "A Study of CPU Scheduling Techniques in Comparison", Turkish Journal of Computer and Mathematics Education (TURCOMAT), Vol. 11, pp. 896-905, 2020.
[14] A.K. Ibrahim, R.I. Saeed, R.A. Ali, "Partially Shading A PV Module: Analysis and Design with Simulation", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 51, Vol. 14, No. 2, pp. 124-129, June 2022.
[15] A. Bahani, E.H. Ech Chhibat, H. Samri, H.A. El Attar, "Cooperation Simulation of Two 6-Axis Robots in an Additive Concrete Construction Application",

International Journal on Technical and Physical Problems of Engineering, (IJTPE), Issue 53, Vol. 14, No. 4, pp. 156-162, June 2022.
[16] A. Krari, A. Hajami, E. Jarmouni, “Study and Analysis of RPL Performance Routing Protocol Under Various Attacks”, International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 49, Vol. 13, No. 4, pp. 152-161, December 2021.
[17] A. Onsman, ”Single Processor Systems”, www.tutorialspoint.com/Single-Processor-Systems, 2018.
[18] L. Williamsm, “FCFS Scheduling Algorithm: What is, Example Program”, www.guru99.com/fcfs-scheduling.html, 2021.
[19] A. Silberschatz, P.B. Galvin, G. Gagne, “Operating System Concepts, Windows XP Update”, John Wiley and Sons, p. 976, USA, 2006.
[20] T. Dickhaus, “Rank Tests”, Theory of Nonparametric Tests, Springer International Publishing, pp. 47-69, Cham, Switzerland, 2018.
[21] W.D. Kelton, R.P. Sadowski, N.B. Zupick, “Simulation with Arena”, Mcgrow Hill, p. 656, NY, USA, 2014.
[22] G.C. Fox, R.D. Williams, P.C. Messina, “Parallel Computing Works!”, Morgan Kaufmann, Burlington, p. 977, Massachusetts, USA, 2014.

BIOGRAPHIES

Name: **Maysoon**
Middle Name: **Abdulameir**
Surname: **Mohammed**
Birthday: 08.12.1977
Birth Place: Baghdad, Iraq
Bachelor: Computer Sciences, Computer Sciences Department, University of Technology, Baghdad, Iraq, 1999
Master: Computer Sciences, Computer Sciences Department, University of Technology, Baghdad, Iraq, 2005
Doctorate: Computer Sciences, Faculty of Computer and Software Engineering, University Malaysia Pahang, Pahang, Malaysia, 2017
The Last Scientific Position: Lecturer, Coordinator of Scientific Committee, Mechanical Engineering Department, University of Technology, Baghdad, Iraq, Since 2000
Research Interests: Operating System, Data Information, Simulation and Modeling
Scientific Publications: 12 Papers

Name: **Shaymaa**
Middle Name: **Kadhim**
Surname: **Mohsin**
Birthday: 14.05.1978
Birth Place: Baghdad, Iraq
Bachelor: Computer Sciences, Computer Sciences Department, University of Technology, Baghdad, Iraq, 2007
Master: Computer Sciences, Information Security Systems Department, Kazan National Research Technical University, Kazan, Tatarstan, Russia, 2014
The Last Scientific Position: Assist. Lecturer, Mechanical Engineering Department, University of Technology, Baghdad, Iraq, Since 2010
Research Interests: Data Security Information
Scientific Publications: 5 Papers

Name: **Sarah**
Middle Name: **Jasim**
Surname: **Mohammed**
Birthday: 22.03.1986
Birth Place: Baghdad, Iraq
Bachelor: Computer Sciences, Computer Sciences Department, University of Technology, Baghdad, Iraq, 2007
Master: Computer Sciences, Computer Sciences Department, University of Technology, Baghdad, Iraq, 2015
The Last Scientific Position: Assist. Lecturer, Mechanical Engineering Department, University of Technology, Baghdad, Iraq, Since 2010
Research Interests: Information Technology, Image Recognition, AI
Scientific Publications: 6 Papers