Performance Analysis of STS-NP Algorithm with variation in Context Switching Time

[1]Sheikh Muhammad Bilawal, [2]Muhammad Adnan Khan

[1]AIMS Institute of Management Sciences, DHA campus, Lahore, Cantt
[2]School of Engineering and Applied Sciences, ISRA University, Islamabad Campus

Abstract
Short Term Scheduling (STS) algorithms are available which is very helpful for operating system to select the processes for execution. Each process has its own arrival time and execution time (burst time). A burst time is a time period, which indicates the total time taken by a process for its execution. A STS algorithm tries to maximize the CPU utilizations. One factor that effect CPU utilization is context switching time. This paper presents that, with Non-Preemptive (STS) scheduling algorithms if context switching time varies then it produced impact upon the performance of CPU. Furthermore, this is finished by utilizing deterministic model and queuing model.
Keywords:
STS, NP, PCB, Burst time, Deterministic model, Execution Time, context switching, Arrival time

1. Introduction

A system has distinctive states amid its execution and a program in its execution state is known as procedure [1]. Furthermore, Context exchanging is a system in which a CPU leaves the as of now executing undertaking and chooses the following errand from prepared line; likewise stack it into memory for execution. At the point when a CPU changes to the following assignment it first spare the condition of as of now running errand in its Process Control Block (PCB) and afterward stack the PCB of next undertaking from the ready queue.

In the event that a PC is working then there is a probability of event of numerous jobs [3]. In Non-Preemptive scheduling calculation if a procedure executing its Burst time and another procedure arrived, the at present executing job will proceeded regardless of the possibility that the procedure complete its undertaking. At that point it picks the following job as per chose calculation method. Case in point in the event that it is utilizing FCFS Scheduling calculation with Non-Preemptive procedure then, that specific occupation will be chosen whose arrival time is least, and on the off chance that it is utilizing SJF with Non-Preemptive system then it check Burst time, and in the event that it is Priority with Non-Preemptive strategy then it will break down need number.

It investigates that because of variety (increment or abatement) connection exchanging time it will specifically influence the general execution of CPU planning calculations. Context exchanging time is considered as overhead. Context exchanging times are mostly wards upon an equipment support [2]. A Job essentially possesses distinctive stages for its consummations [1].

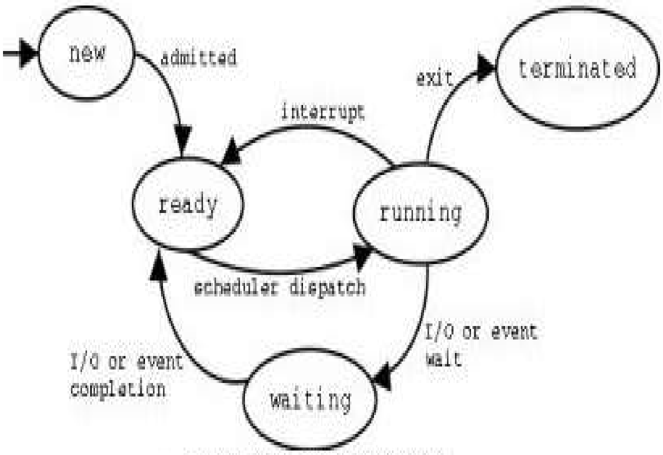

Figure 1: Process State Model

1.1 Importance of context switching in short term scheduling algorithm with Non-Preemptive technique

Short Term Scheduling algorithms are useful for job scheduler or swapper in the choice of jobs from job queue. Job scheduler otherwise called long term scheduler and its fundamental intention is to choose the proper occupations from various employment lines and give a blended occupation to execution, for example, CPU bound and I/O bound jobs. It is likewise used to control the level of multi-programming [5]. Swapper is additionally called medium term scheduler, and it is utilized to diminish the level of multi-programming. Also, it swaps those employments whose are in holding up state and make space in memory for new jobs [12]. In a unit amount of time if a certain amount of work is completed it is known as throughput. Our motive is to complete more jobs in less time, means to increase throughput. [11]

Manuscript received September 5, 2026
Manuscript revised September 20, 2026
https://doi.org/10.22937/IJCSNS.2026.26.9.8

There are many scheduling algorithm techniques are available with different characteristics. However, there are some properties and characteristics of algorithms on which comparison of different techniques is based. [8][9] Commonly the criteria for scheduling are based upon Utilization or efficiency, Turnaround time, waiting time, throughput time, and Response time etc.

2. Short-term Scheduling Algorithm with Non-Preemptive Technique

1. First Come First Serve (Non-Preemptive)
2. Shorter Job First (Non-Preemptive)
3. Priority (Non-Preemptive)

2.1 First Come First Serve (Non-Preemptive)

The technique of First Come First Serve of Non-Preemptive algorithm is simple and straight forward it is also known as First In First Out algorithm. Those processes which come first for execution in ready queue, will deal first and the processes will deal one by one in a sequence in which those processes will arrived. Because this algorithm is non-preemptive [7] therefore it will execute a current process completely and then it will move to next one. If more than one tasks arrive at the same time and waiting for their execution in ready queue and currently executing job is slower and have a large burst time is using the CPU, and all other jobs has lesser burst time, but all of these jobs must waits for unnecessarily longer period of time. [6] This situation is known as convoy effect. Therefore First Come First Serve (Non-Preemptive) algorithm is the troublesome for those systems which are time sharing.

2.2 Shortest Job First (Non-Preemptive)

Shortest Job First algorithm with Non-Preemptive technique focus on the smaller burst time. If multiple tasks arrive at the same time and place in a ready queue for executions then according to this algorithm it will pick that particular task from job pool has a smaller burst time. But if two different tasks have the same burst time then FCFS technique uses to select the appropriate task. [14] When a new task arrive in ready queue for execution while another task is currently executing, then the burst time of newly coming task is compared to the currently running task if the burst time of newly coming task is less than currently executing task, the currently executing task will complete first and then pick the other because it is non-preemptive algorithm

2.3 Priority (Non-Preemptive)

In Priority algorithm, a priority number is also defined with each task, and this priority number decide which task will execute, if the priority number is greater; then it has less priority, and if priority number is smaller than priority of that process will greater. For instance a task having priority number 1 has higher priority, than a task having priority number 2. So if a process is executing and a process of higher priority occurs then the running process will complete first then it will perform context switching and select the task according to their priority number, from currently available tasks in ready queue. As it is non-preemptive algorithm, therefore if a task is selected then it will not free the CPU before its completion. A situation may arise that coming task has higher priority than existing task then aging is required [10].

3. Models of Analytic Evaluation

3.1 Deterministic model

This model is purely mathematical model in which the output is identified with readily available values. In this model Randomization element for values is not included. [15] Due to this reason, always same output will be generated for a given input. Consider the following example of data, by using deterministic model on all Non-Preemptive algorithms to determine average Response time, average Waiting time, and average Turnaround time.

Table 1: Examples of Times

Jobs	Burst Time	Arrival Time	Priority
J1	6.6	2.4	7
J2	2.8	1.3	5
J3	5.4	0.8	4
J4	1.3	2.3	6
J5	0.3	0.2	1
J6	2.3	0.2	2
J7	0.7	0.5	3
J8	0.2	3.1	9
J9	2.2	4.4	10
J10	4.3	2.5	8

(Time measured in µSeconds)

Table 2: First Come First Serve (FCFS)
Contact Switching Time = 0.0 µs

Jobs	Burst Time	Arrival Time
J1	6.6	2.4
J2	2.8	1.3
J3	5.4	0.8
J4	1.3	2.3
J5	0.3	0.2
J6	2.3	0.2
J7	0.7	0.5
J8	0.2	3.1
J9	2.2	4.4
J10	4.3	2.5

Table 3: Gantt chart

Free	J5	J6	J7	J3	J2
0	0.2	0.5	2.8	3.5	8.9

J4	J1	J10	J8	J9	J10
0	0.2	19.6	23.9	24.1	26.3

Response Time (RT) = (1st Time CPU Allocation of Ji) – Arrival Time
RT of J1 = 13.3 – 2.4 = 10.9 µs
RT of J2 = 8.9 – 1.3 = 7.6 µs
RT of J3 = 3.5 – 0.8 = 2.7 µs
RT of J4= 11.7 – 2.3 = 9.4 µs
RT of J5 = 0.2 – 0.2 = 0 µs
RT of J6 = 0.5 – 0.2 = 0.3 µs
RT of J7 = 2.8 – 0.5 = 2.3 µs
RT of J8 = 23.9 – 3.1 = 20.8 µs
RT of J9 = 24.1 – 4.4 = 19.7 µs
RT of J10 = 19.4 – 2.5 = 16.9 µs
Average RT = (10.9+7.6+2.7+9.4+0 +0.3+2.3+20.8+19.7+16.9) / 10
Average RT = (90.6)/10 = 9.06 µs

Turnaround Time (TAT) = (BT of Ji) + Waiting Time (WT)
TAT of J1 = 6.6 + 10.9 = 17.5 µs
TAT of J2 = 2.8 + 7.6 = 10.4 µs
TAT of J3 = 5.4 + 2.7 = 8.1 µs
TAT of J4 = 1.3 + 9.4 = 10.7 µs
TAT of J5 = 0.3 + 0 = 0.3 µs
TAT of J6 = 2.3 + 0.3 = 2.6 µs
TAT of J7 = 0.7 + 2.3 = 4.0 µs
TAT of J8 = 0.2 + 20.8 = 21.0 µs
TAT of J9 = 2.4 + 19.7 = 22.1 µs
TAT of J10 = 4.3 + 16.9 = 21.2 µs
Average TAT= (17.5+10.4+8.1+10.7+0.3+2.6+4.0+21.0+22.1+21.2) /10
Average TAT = 117.9 / 10 = 11.79 µs

Table 4: First Come First Serve (FCFS)
Context Switching Time = 0.2 µs

Free	J5	CS	J6	CS	J7
0	0.2	0.5	0.7	3.0	3.2

CS	J3	CS	J2	CS	J4
3.9	4.1	9.5	9.7	12.5	12.7
CS	J1	CS	J10	CS	J8
14.0	14.2	20.8	21.0	25.3	25.5

CS	J9	
25.7	25.9	28.1

Response Time (RT) = (1st Time CPU Allocation of Ji) – Arrival Time
RT of J1 = 14.2 – 2.4 = 11.8 µs
RT of J2 = 9.7 – 1.3 = 8.4 µs
RT of J3 = 4.1 – 0.8 = 3.3 µs
RT of J4= 12.7 – 2.3 = 10.4 µs
RT of J5 = 0.2 – 0.2 = 0 µs
RT of J6 = 0.7 – 0.2 = 0.5 µs
RT of J7 = 3.2 – 0.5 = 2.7 µs
RT of J8 = 25.5 – 3.1 = 22.4 µs
RT of J9 = 25.9 – 4.4 = 21.5 µs
RT of J10 = 21.0 – 2.5 = 18.5 µs
Average RT = (11.8+8.4+3.3+10.4+0+0.5+2.7+22.4+21.5+18.5) / 10
Average RT = (99.5)/10 = 9.95 µs

Turnaround Time (TAT) = (BT of Ji) + Waiting Time (WT)
TAT of J1 = 6.6 +11.8 = 18.4 µs
TAT of J2 = 2.8 + 8.4 = 11.2 µs
TAT of J3 = 5.4 + 3.3 = 8.7 µs
TAT of J4 = 1.3 + 10.4 = 11.7 µs
TAT of J5 = 0.3 + 0 = 0.3 µs
TAT of J6 = 2.3 + 0.5 = 2.8 µs
TAT of J7 = 0.7 + 2.7 = 3.4 µs
TAT of J8 = 0.2 + 22.4 = 22.6 µs
TAT of J9 = 2.2 + 21.5 = 23.7 µs
TAT of J10 = 4.3 + 18.5 = 22.8 µs
Average TAT= (18.4+11.2+8.7+11.7+0.3+2.8+3.4+22.6+23.7+22.8) /10
Average TAT = 125.6 / 10 = 12.56 µs

Table 3: First Come First Serve (FCFS)
Contact Switching Time = 0.5 µs

Free	J5	CS	J6	CS	J7
0	0.2	0.5	10.	3.3	3.8

CS	J3	CS	J2	CS	J4
4.5	5.0	10.4	10.9	13.7	14.2

CS	J1	CS	J10	CS	J8
15.5	16.0	22.6	23.1	27.4	27.9

CS	J9	
28.1	28.6	30.8

Response Time (RT) = (1st Time CPU Allocation of Ji) – Arrival Time
RT of J1 = 16.0 – 2.4 = 13.6 µs
RT of J2 = 10.9 – 1.3 = 9.6 µs
RT of J3 = 5.0 – 0.8 = 4.2 µs
RT of J4= 14.2 – 2.3 = 11.9 µs
RT of J5 = 0.2 – 0.2 = 0 µs
RT of J6 = 1.0 – 0.2 = 0.8 µs

RT of J7 = 3.8 – 0.5 = 3.3 µs
RT of J8 = 27.9 – 3.1 = 24.8 µs
RT of J9 = 28.6 – 4.4 = 24.2 µs
RT of J10 = 23.1 – 2.5 = 20.6 µs
Average RT = (13.6+9.6+4.2+11.9+0+0.8+3.3+24.8+24.2+ 20.6) / 10
Average RT = (113.0)/10 = 11.30 µs

Turnaround Time (TAT) = (BT of Ji) + Waiting Time (WT)
TAT of J1 = 6.6 + 13.6 = 20.2 µs
TAT of J2 = 2.8 +9.6 = 12.4 µs
TAT of J3 = 5.4 + 4.2 = 9.6 µs
TAT of J4 = 1.3 + 11.9 = 13.2 µs
TAT of J5 = 0.3 + 0 = 0.3 µs
TAT of J6 = 2.3 + 0.8 = 3.1 µs
TAT of J7 = 0.7 + 3.3 = 4.0 µs
TAT of J8 = 0.2 + 24.8 = 25.0 µs
TAT of J9 = 2.2 + 24.2 = 26.4 µs
TAT of J10 = 4.3 + 20.6 = 24.9 µs
Average TAT = (20.2 + 12.4 + 9.6 + 12.3 + 0.3 + 3.1 + 4.0 + 25.0 + 26.4 + 24.9) /10
Average TAT = 139.1/ 10 = 13.91 µs

Table 4: Shortest Job First Serve (SJFS)
Context Switching Time = 0.0 µs

Jobs	Burst Time	Arrival Time
J1	6.6	2.4
J2	2.8	1.3
J3	5.4	0.8
J4	1.3	2.3
J5	0.3	0.2
J6	2.3	0.2
J7	0.7	0.5
J8	0.2	3.1
J9	2.2	4.4
J10	4.3	2.5

Gantt chart

Free	J5	J7	J6	J8	J4
0	0.2	0.5	1.2	3.5	3.7

J9	J2	J10	J3	J1	
5.0	7.2	10.0	14.3	19.7	26.1

Response Time (RT) = (1st Time CPU Allocation of Ji) – Arrival Time
RT of J1 = 19.7 – 2.4 = 17.3 µs
RT of J2 = 7.2 – 1.3 = 5.9 µs
RT of J3 = 14.3 – 0.8 = 13.5 µs
RT of J4= 3.7 – 2.3 = 1.4 µs
RT of J5 = 0.2 – 0.2 = 0 µs
RT of J6 = 1.2 – 0.2 = 1.0 µs
RT of J7 = 0.5 – 0.5 = 0 µs
RT of J8 = 3.5 – 3.1 = 0.4 µs
RT of J9 = 5.0 – 4.4 = 0.6 µs
RT of J10 = 10.0 – 2.5 = 7.5 µs
Average RT = (17.3+5.9+13.5 +1.4 + 0 +1.0 +0 +0.4 +0.6 + 7.5) /10
Average RT = (47.6) /10 = 4.76 µs

Turnaround Time (TAT) = (BT of Ji) + Waiting Time (WT)
TAT of J1 = 6.6 + 17.3 = 23.9 µs
TAT of J2 = 2.8 + 5.9 = 8.7 µs
TAT of J3 = 5.4 + 13.5 = 18.9 µs
TAT of J4 = 1.3 + 1.4 = 2.7 µs
TAT of J5 = 0.3 + 0 = 0.3 µs
TAT of J6 = 2.3 + 1.0 = 3.3 µs
TAT of J7 = 0.7 + 0 = 0.7 µs
TAT of J8 = 0.2 + 0.4 = 0.6 µs
TAT of J9 = 2.2 + 0.6 = 2.8 µs
TAT of J10 = 4.3 + 7.5 = 11.8 µs
Average TAT= (23.9 +8.7+18.9 +2.7+0.3 +3.3+0.7+0.6+2.8+11.8) /10
Average TAT = 73.7 / 10 = 7.37 µs

Table 5: Shortest Job First Serve (SJFS)
Context Switching Time = 0.2 µs
Gantt chart

Free	J5	CS	J7	CS	J6
0	0.2	0.5	0.7	1.4	1.6

CS	J8	CS	J4	CS	J9
3.9	4.1	4.3	4.5	5.8	6.0

CS	J2	CS	J10	CS	J3
8.2	8.4	11.2	11.4	15.7	15.9

CS	J1	
21.3	21.5	28.1

Response Time (RT) = (1st Time CPU Allocation of Ji) – Arrival Time
RT of J1 = 21.5 – 2.4 = 19.1 µs
RT of J2 = 8.4 – 1.3 = 7.1 µs
RT of J3 = 15.9 – 0.8 = 15.1 µs
RT of J4= 4.5 – 2.3 = 3.7 µs
RT of J5 = 0.2 – 0.2 = 0 µs
RT of J6 = 1.6 – 0.2 = 1.4 µs
RT of J7 = 0.7 – 0.5 = 0.2 µs
RT of J8 = 4.1 – 3.1 = 1.0 µs
RT of J9 = 6.0 – 4.4 = 1.6 µs
RT of J10 = 11.4 – 2.5 = 8.9 µs
Average RT = (19.1+ 7.1+ 15.1+ 3.7+ 0+ 1.4+ 0.2+ 1.0+1.6 + 8.9) /10

Average RT = (58.1)/10 = 5.81µs

Turnaround Time (TAT) = (BT of Ji) + Waiting Time (WT)
TAT of J1 = 6.6 + 19.1 = 25.7 µs
TAT of J2 = 2.8 + 7.1 = 9.9 µs
TAT of J3 = 5.4 + 15.1 = 20.5 µs
TAT of J4 = 1.3 + 3.7 = 5.0 µs
TAT of J5 = 0.3 + 0 = 0.3 µs
TAT of J6 = 2.3 + 1.4 = 3.7 µs
TAT of J7 = 0.7 + 0.2 = 0.9 µs
TAT of J8 = 0.2 + 1.0 = 1.2 µs
TAT of J9 = 2.2 + 1.6 = 3.8 µs
TAT of J10 = 4.3 + 8.9 = 13.2 µs
Average TAT = (25.7+9.9+20.5 + 5.0+0.3+3.7+0.9+1.2 +3.8+13.2) /10
Average TAT = 84.2 / 10 = 8.42 µs

Table 6: Shortest Job First Serve (SJFS) Context Switching Time = 0.5 µs

Gantt chart

Free	J5	CS	J7	CS	J6
0	0.2	0.5	1.0	1.7	2.2

CS	J8	CS	J4	CS	J9
4.5	5.0	5.2	5.7	7.0	7.5

CS	J2	CS	J10	CS	J3
9.7	10.2	13.0	13.5	17.8	18.3

CS	J1	
23.7	24.2	30.8

Response Time (RT) = (1^{st} Time CPU Allocation of Ji) – Arrival Time
RT of J1 = 24.2 – 2.4 = 21.8 µs
RT of J2 = 10.2 – 1.3 = 8.9 µs
RT of J3 = 18.3 – 0.8 = 17.5 µs
RT of J4= 5.7 – 2.3 = 3.4 µs
RT of J5 = 0.2 – 0.2 = 0 µs
RT of J6 = 2.2 – 0.2 = 2.0 µs
RT of J7 = 1.0 – 0.5 = 0.5 µs
RT of J8 = 5.0 – 3.1 = 1.9 µs
RT of J9 = 7.5 – 4.4 = 3.1 µs
RT of J10 = 13.5 – 2.5 = 11.0 µs
Average RT = (21.8 + 8.9 + 17.5+3.4+0+2.0+0.5+1.9+3.1+11.0) / 10
Average RT = (70.1)/10 = 7.01 µs

Turnaround Time (TAT) = (BT of Ji) + Waiting Time (WT)
TAT of J1 = 6.6 + 21.8 = 28.4 µs
TAT of J2 = 2.8 + 8.9 = 11.7 µs
TAT of J3 = 5.4 + 17.5 = 22.9 µs
TAT of J4 = 1.3 + 3.4 = 4.7 µs
TAT of J5 = 0.3 + 0 = 0.3 µs
TAT of J6 = 2.3 + 2.0 = 4.3 µs
TAT of J7 = 0.7 + 0.5 = 1.2 µs
TAT of J8 = 0.2 + 1.9 = 2.1 µs
TAT of J9 = 2.2 + 3.1 = 5.3 µs
TAT of J10 = 4.3 + 11.0 = 15.3 µs
Average TAT = (28.4+11.7+22.9+4.7+0.3+4.3+1.2+2.1+5.3+15.3) /10
Average TAT = (96.2) / 10 = 9.62 µs

Table 7: Priority Context Switching Time = 0.0 µs

Example:

Jobs	Burst Time	Arrival Time	Priority
J1	**6.6**	**2.4**	**7**
J2	**2.8**	**1.3**	**5**
J3	**5.4**	**0.8**	**4**
J4	**1.3**	**2.3**	**6**
J5	**0.3**	**0.2**	**1**
J6	**2.3**	**0.2**	**2**
J7	**0.7**	**0.5**	**3**
J8	**0.2**	**3.1**	**9**
J9	**2.2**	**4.4**	**10**
J10	**4.3**	**2.5**	**8**

Gantt chart

Free	J5	J6	J4	J7	J9
0	0.2	0.5	2.8	4.1	4.8

J10	J3	J2	J1	J8	
7.0	11.3	16.7	19.5	26.1	26.3

Response Time (RT) = (1^{st} Time CPU Allocation of Ji) – Arrival Time
RT of J1 = 19.7 – 2.4 = 17.1 µs
RT of J2 = 16.7 – 1.3 = 15.4 µs
RT of J3 = 11.3 – 0.8 = 10.5 µs
RT of J4= 2.8 – 2.3 = 0.5 µs
RT of J5 = 0.2 – 0.2 = 0 µs
RT of J6 = 0.5 – 0.2 = 0.3 µs
RT of J7 = 4.1 – 0.5 = 3.6 µs
RT of J8 = 26.1 – 3.1 = 23.0 µs
RT of J9 = 4.8 – 4.4 = 0.4 µs
RT of J10 = 7.0 – 2.5 = 4.5 µs
Average RT = (17.1+15.4+10.5+0.5+0+0.3+3.6+23.0 + 0.4 + 4.5) / 10
Average RT = (75.3)/10 = 7.53 µs

Turnaround Time (TAT) = (BT of Ji) + Waiting Time (WT)
TAT of J1 = 6.6 + 17.1 = 23.7 µs
TAT of J2 = 2.8 + 15.4 = 18.2 µs
TAT of J3 = 5.4 + 10.5 = 15.9 µs
TAT of J4 = 1.3 + 0.5 = 1.8 µs
TAT of J5 = 0.3 + 0 = 0.3 µs

TAT of J6 = 2.3 + 0.3 = 2.6 µs
TAT of J7 = 0.7 + 3.6 = 4.3 µs
TAT of J8 = 0.2 + 23.0 = 23.2 µs
TAT of J9 = 2.2 + 0.4 = 2.6 µs
TAT of J10 = 4.3 + 4.5 = 8.8 µs
Average TAT = (23.7+18.2+15.9+1.8+0.3+2.6+4.3+23.2+2.6+8.8) /10
Average TAT = (101.4) / 10 = 10.14 µs

Table 8: Priority Context Switching Time = 0.2 µs
Gantt chart

Free	J5	CS	J6	CS	J4
0	0.2	0.5	0.7	3.0	3.2

CS	J9	CS	J7	CS	J10
4.5	4.7	6.9	7.1	7.8	8.0

CS	J3	CS	J2	CS	J1
12.3	12.5	17.9	18.1	20.9	21.1

CS	J8	
27.7	27.9	28.1

Response Time (RT) = (1st Time CPU Allocation of Ji) – Arrival Time
RT of J1 = 21.1 – 2.4 = 18.7 µs
RT of J2 = 18.1 – 1.3 = 16.8 µs
RT of J3 = 12.5 – 0.8 = 11.7 µs
RT of J4= 3.2 – 2.3 = 0.9 µs
RT of J5 = 0.2 – 0.2 = 0 µs
RT of J6 = 0.7 – 0.2 = 0.5 µs
RT of J7 = 7.1 – 0.5 = 6.6 µs
RT of J8 = 27.9 – 3.1 = 24.8 µs
RT of J9 = 4.7 – 4.4 = 0.3 µs
RT of J10 = 8.0 – 2.5 = 5.5 µs
Average RT = (18.7+16.8+11.7+0.9+0+0.5+6.6+24.8+0.3+5.5) / 10
Average RT = (85.8)/10 = 8.58 µs

Turnaround Time (TAT) = (BT of Ji) + Waiting Time (WT)
TAT of J1 = 6.6 + 18.7 = 25.3 µs
TAT of J2 = 2.8 + 16.8 = 19.6 µs
TAT of J3 = 5.4 + 11.7 = 17.1 µs
TAT of J4 = 1.3 + 0.9 = 2.2 µs
TAT of J5 = 0.3 + 0 = 0.3 µs
TAT of J6 = 2.3 + 0.5 = 2.8 µs
TAT of J7 = 0.7 + 6.6 = 7.3 µs
TAT of J8 = 0.2 + 24.8 = 25.0 µs
TAT of J9 = 2.2 + 0.3 = 2.5 µs
TAT of J10 = 4.3 + 5.5 = 9.8 µs
Average TAT = (25.3+19.6+17.1+2.2+0.3+2.8+7.3+25.0+2.5 + 9.8) /10
Average TAT = (111.9) / 10 = 11.19 µs

Table 3: Priority Context Switching Time = 0.5 µs
Gantt chart

Free	J5	CS	J6	CS	J4
0	0.2	0.5	1.0	3.3	3.8

CS	J9	CS	J7	CS	J10
5.1	5.69	7.8	8.3	9.0	9.5

CS	J3	CS	J2	CS	J1
13.8	14.3	19.7	20.2	23.0	25.5

CS	J8	
30.1	30.6	30.8

Response Time (RT) = (1st Time CPU Allocation of Ji) – Arrival Time
RT of J1 = 23.5 – 2.4 = 21.1 µs
RT of J2 = 20.2 – 1.3 = 18.9 µs
RT of J3 = 14.3 – 0.8 = 13.5 µs
RT of J4= 3.8 – 2.3 = 1.5 µs
RT of J5 = 0.2 – 0.2 = 0 µs
RT of J6 = 1.0 – 0.2 = 0.8 µs
RT of J7 = 8.3 – 0.5 = 7.8 µs
RT of J8 = 30.6 – 3.1 = 27.5 µs
RT of J9 = 5.6 – 4.4 = 1.2 µs
RT of J10 = 9.5 – 2.5 = 7.0 µs
Average RT = (21.1+18.9 +13.5+1.5+0+0.8+7.8+27.5+ 1.2+7.0) / 10
Average RT = (99.3)/10 = 9.93 µs

Turnaround Time (TAT) = (BT of Ji) + Waiting Time (WT)
TAT of J1 = 6.6 + 21.1 = 27.7 µs
TAT of J2 = 2.8 + 18.9 = 21.7 µs
TAT of J3 = 5.4 + 13.5 = 18.9 µs
TAT of J4 = 1.3 + 1.5 = 2.8 µs
TAT of J5 = 0.3 + 0 = 0.3 µs
TAT of J6 = 2.3 + 0.8 = 3.1 µs
TAT of J7 = 0.7 + 7.8 = 8.5 µs
TAT of J8 = 0.2 + 27.5 = 27.7 µs
TAT of J9 = 2.2 + 1.2 = 3.4 µs
TAT of J10 = 4.3 + 7.0 = 11.3 µs
Average TAT =(27.7+21.7+18.9+2.8+0.3+3.1+8.5+27.7+3.4+11.3)/10
Average TAT = (125.4) / 10 = 12.54 µs

---

4. Analysis Charts

Context Switching Time = 0.0µs

Sr#	Algorithm	Avg RT (µs)	Avg TAT (µs)
1	FCFS(NP)	9.06	11.97
2	SJF(NP)	4.76	7.37
3	Priority(NP)	7.53	10.14

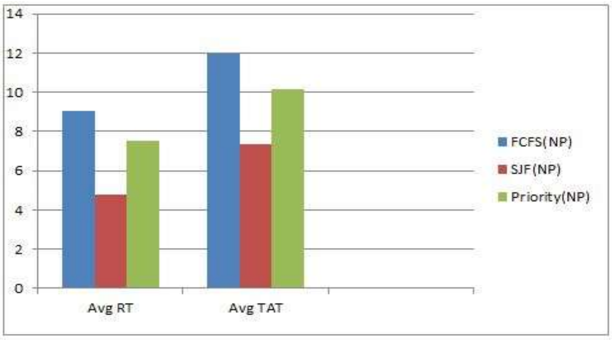

Figure 2: Comparison Graph between Average Response Time (µs) and Average Turnaround Time (µs) of scheduling algorithms

Context Switching Time = 0.2 µs

Sr#	Algorithm	Avg RT (µs)	Avg TAT(µs)
1	FCFS(NP)	9.95	12.56
2	SJF(NP)	5.81	8.42
3	Priority(NP)	8.58	11.19

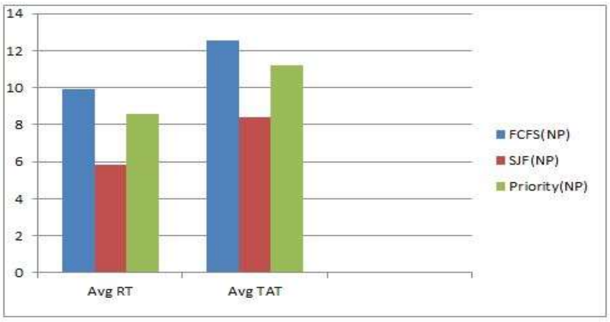

Figure 3: Comparison Graph between Average Response Time (µs) and Average Turnaround Time (µs) of scheduling algorithms

Context Switching Time = 0.5 µs

Sr#	Algorithm	Avg RT (µs)	Avg TAT (µs)
1	FCFS(NP)	11.30	13.91
2	SJF(NP)	7.01	9.62
3	Priority(NP)	9.93	12.54

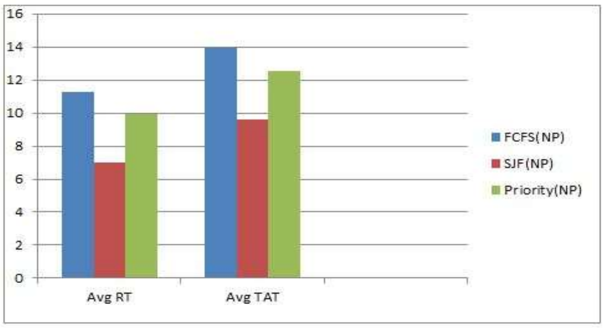

Figure 4: Comparison Graph between Average Response Time (µs) and Average Turnaround Time (µs) of scheduling algorithms

Queuing model

In this model, waiting queues are analyzed. This model is built for predict the lengths and waiting times of a queue. [13]
The formula used for this purpose is:

$$n = \lambda w$$

Here in this formula:
n = Average queue length
λ = Average arrival time
w = Average waiting time

(Assuming λ = 0.7)

If Context Switching Time = 0.0µs

Sr #	Algorithm	Average waiting time (w)	Average queue length (n)
1	FCFS(NP)	9.06	6.342
2	SJF(NP)	4.76	3.332
3	Priority(NP)	7.53	5.271

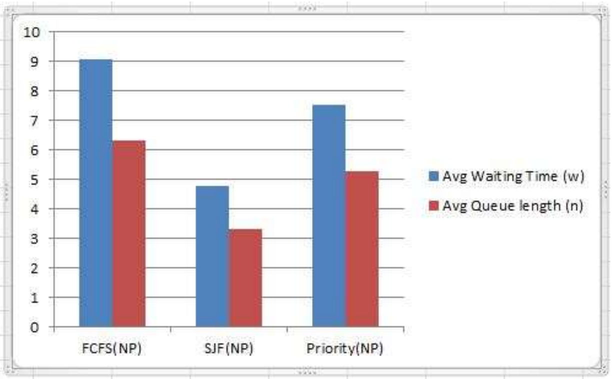

Figure 5: Comparison Graph between Average Waiting Time (µs) and Average Queue length (µs) in Non-Preemptive algorithm, with context switching Time 0.0 µs

If Context Switching Time = 0.2 µs

Sr#	Algorithm	Average waiting time (w)	Average queue length (n)
1	FCFS(NP)	9.95	6.965
2	SJF(NP)	5.81	4.067
3	Priority(NP)	8.58	6.006

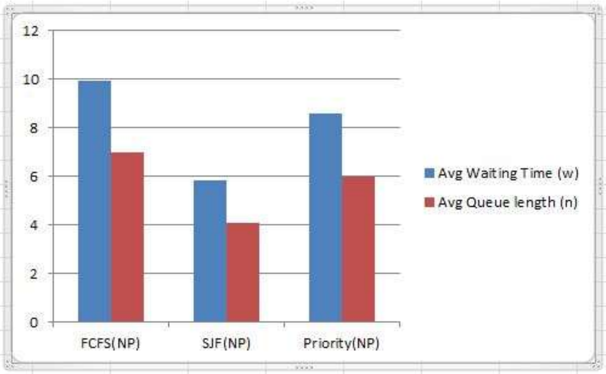

Figure 6: Comparison Graph between Average Waiting Time (µs) and Average Queue length (µs) in Non-Preemptive algorithm, with context switching Time 0.2 µs

If Context Switching Time = 0.5 µs

Sr#	Algorithm	Average waiting time (w)	Average queue length (n)
1	FCFS(NP)	11.30	7.910
2	SJF(NP)	7.01	4.907
3	Priority(NP)	9.93	6.951

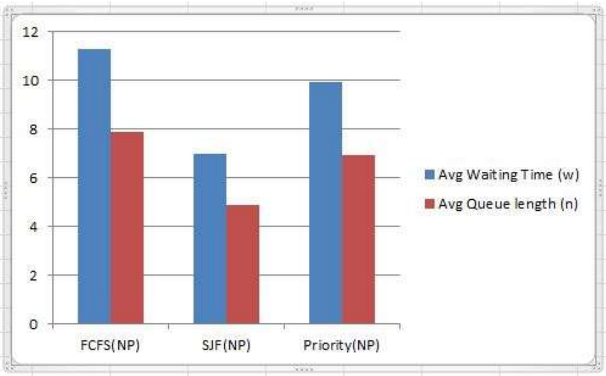

Figure 7: Comparison Graph between Average Waiting Time (µs) and Average Queue length (µs) in Non-Preemptive algorithm, with context switching Time 0.5 µs

5. Conclusion

An investigation of Context Switching time in Short Term Scheduling calculation with Non-Preemptive (STS-NP) methods has been displayed in this paper. The calculations have been investigated by utilizing queuing model and deterministic model. From counts clearly STS-NP has least Response Time, Waiting Time, Turnaround Time and boosts the CPU usage. Thus the proposed variation of STS-NP gives better results when contrasted with other Non-Preemptive planning calculations.

References

[1] Silberschatz A, Galvin PB, Gagne G. Operating System Concepts 2012, 8th edition, Wiley India.
[2] Milan Milenkovic, "Operating Systems Concepts and Design", McGRAM-HILL, Computer Science Series, second edition.
[3] Sabrian F, CD Nguyen, S Jha D Platt, F Safaei. Processing resource scheduling in programmable networks. Computer communication 2005. 28:676-87
[4] http://www.tutorialspoint.com/operating_system/os_process_scheduling.htm
[5] Geol N. A Comparative Study of CPU Scheduling Algorithms. International Journal of Graphics & Image Processing 2012. 245:251.
[6] Saleem U, Javed MY. Simulation of CPU Scheduling Alogrithm. 0-7803-6355-8/00/$10.00@2000 IEEE.
[7] Tanenbaum AS. Modern Operating System 2008. Prentice Hall.
[8] Dhore A. Operating Systems. Technical Publications.
[9] Bandarupalli SB, Nutulapati NP, Varma PS. A Novel CPU Scheduling Algorithm–Preemptive & Non-Preemptive. International Journal of Modern Engineering Research (IJMER) 2012. 6:4484-90.
[10] http://www.geekinterview.com/Interview-Questions/Operating-System/Windows-Unix
[11] Shrivastava M. Analysis and Review of CPU Scheduling Algorithms. IJSER 2014. 3:132-34
[12] http://www.tutorialspoint.com/operating_system/os_process_scheduling.htm
[13] http://www.cs.uic.edu/~jbell/CourseNotes/OperatingSyste ms/5_CPU_Scheduling.html
[14] Kumar A, Rohil H, Arya S. Analysis of CPU Scheduling Policies through Simulation. IJRCSSE 2013. 5:1158-62.