Analysis of CPU Scheduling Algorithms

Operating Systems
CSD-222

Submitted by:
Kashish Srivastava (185014)
Dipesh Kumar (185015)
Akash Rana (185034)

CSE (4 Year) : 4th Semester

Under the guidance of

Dr. Pradeep Singh
Assistant Professor
Department of Computer Science and Engineering
National Institute of Technology, Hamirpur

Introduction

This document is a report for the individual project "Simulation of various CPU scheduling algorithms and analysing the behaviour of each scheduler". CPU scheduling is a process which allows one process to use the CPU while the execution of another process is on hold(in waiting state) due to unavailability of any resource like I/O etc, thereby making full use of CPU. The aim of CPU scheduling is to make the system efficient, fast and fair.

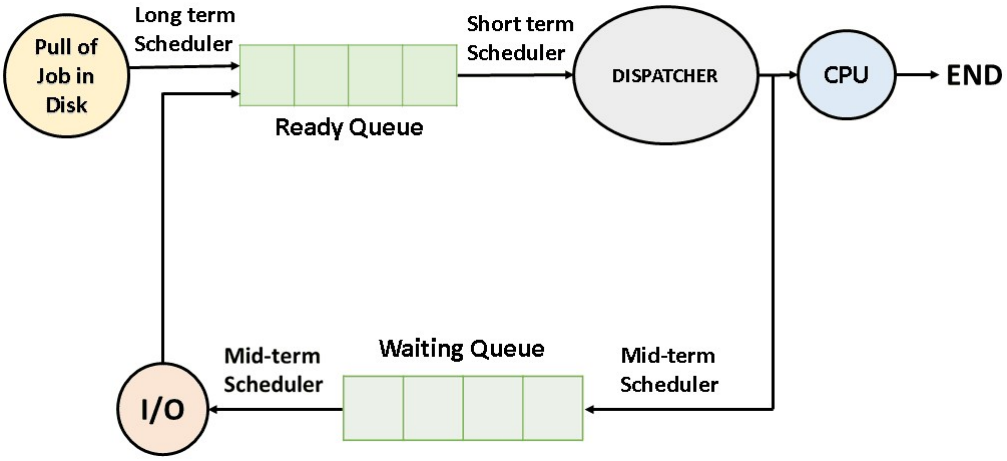

Figure 1: Cpu scheduling

The project is an attempt to differentiate the behaviour of each scheduler with the statistical approach. In this project we performed each CPU scheduling and compared them on the basis of their completion time,throughput,average job elapsed time and average job waiting time during the process. It also discusses the complexity of the written code as instructed.

Software requirement and specification

Python-dateutil(2.8.1)

The dateutil module provides powerful extensions to the standard datetime module, available in Python.

matplotlib(3.2.2)

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

cycler(0.10.0)

A single entry Cycler object can be used to easily cycle over a single style.

Kiwisolver(1.2.0)

Kiwisolver is an efficient C++ implementation of the Cassowary constraint solving algorithm. Kiwi is an implementation of the algorithm based on the seminal Cassowary paper.

numpy(1.19.0)

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices).

pyparsing(2.4.7)

Pyparsing is a mature, powerful alternative to regular expressions for parsing text into tokens and retrieving or replacing those tokens.

six(1.15.0)

Six provides simple utilities for wrapping over differences between Python 2 and Python 3. It is intended to support codebases that work on both Python 2 and 3 without modification.

Project Work

The major aspect of this project work has been to elobrate the main differences that exists between the different aglorithms used in CPU scheduling.
The different algorithms and their throghput, average waiting time, average elapsed time etc are plotted using matplotlib and decribed below using graphs and shapes:-

FCFS

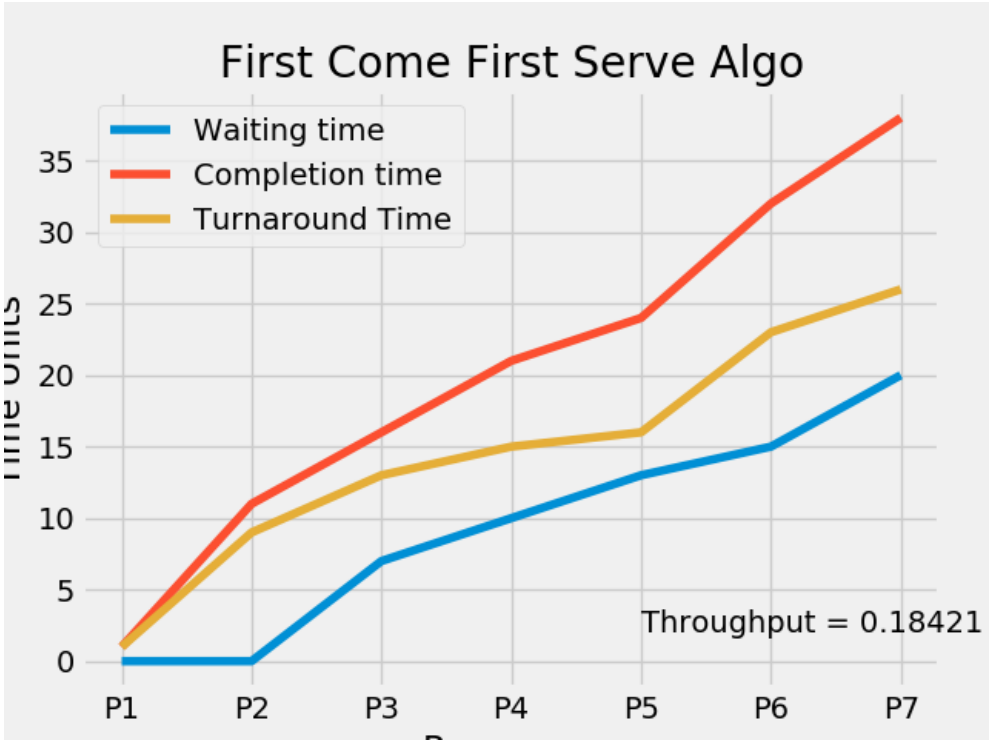

Figure 2: FCFS output values.

SJF(Preemptive)

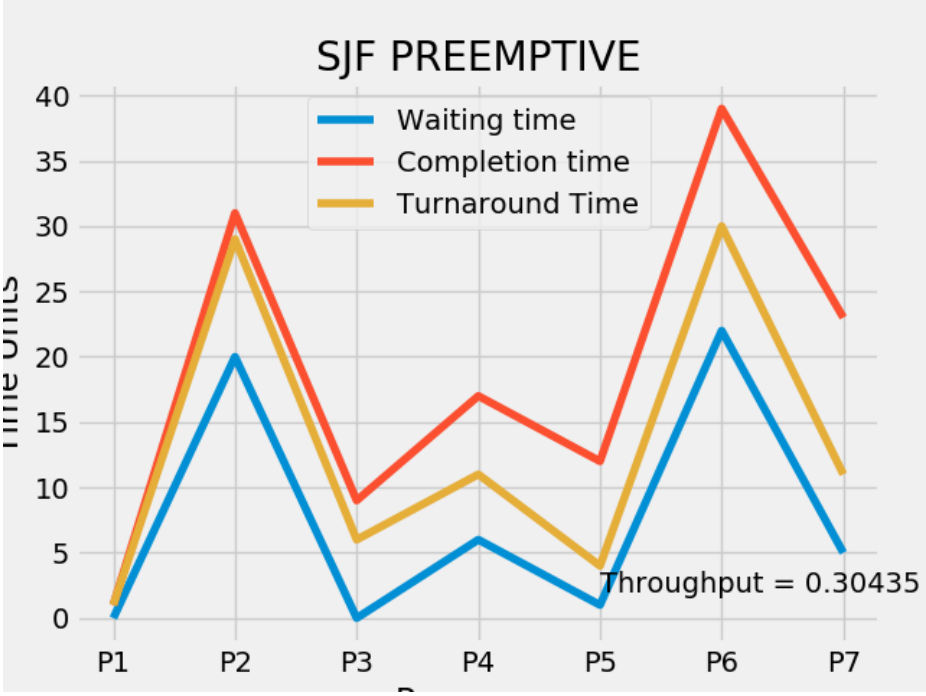

Figure 3: SJF(Preemptive) output values.

SJF(Non preemptive)

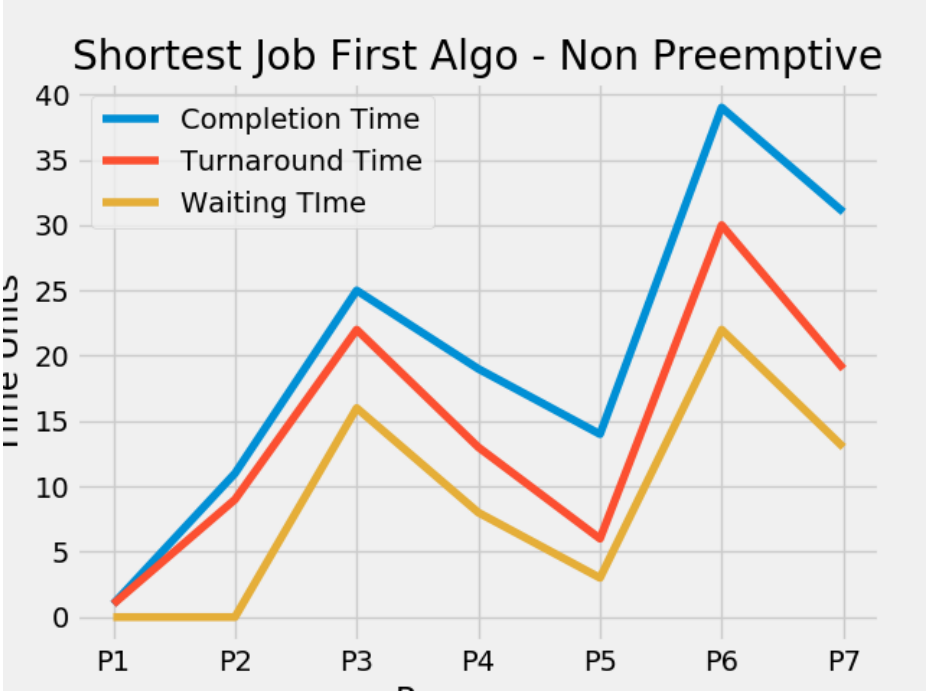

Figure 4: SJF(Non preemptive) output values.

Priorty CPU scheduling(Preemptive)

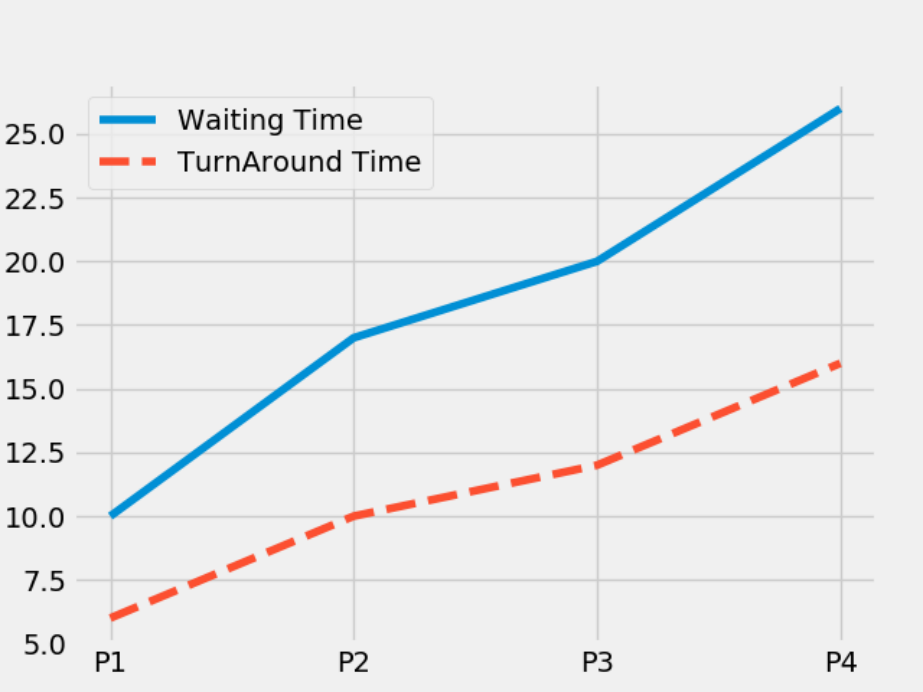

Figure 5: Priorty cpu scheduling(Preemptive) output values.

Priorty CPU scheduling(Non preemptive)

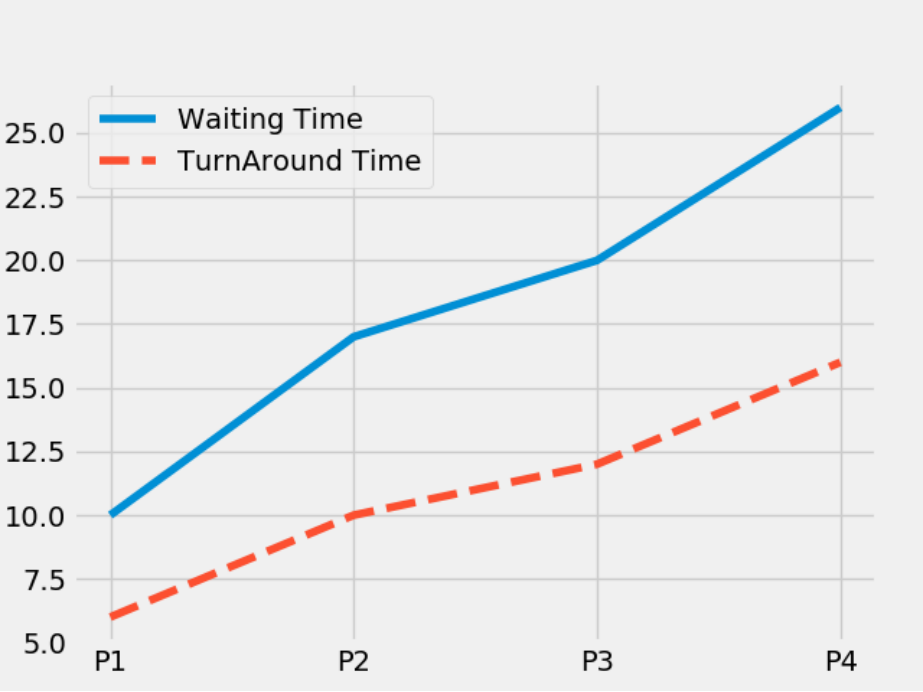

Figure 6: Priorty cpu scheduling(Non Preemptive) output values.

Round robin CPU scheduling

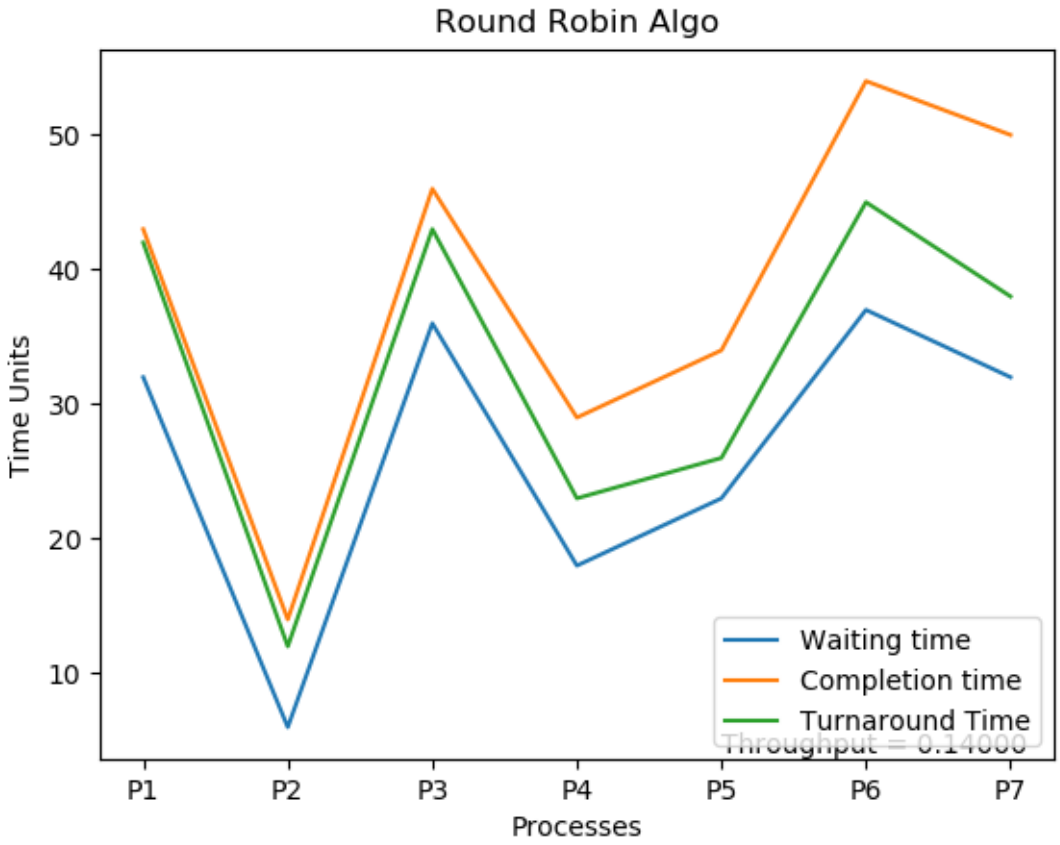

Figure 7: Round Robin cpu scheduling output values.

Conclusion

From the given project work we have concluded that the difference between the turn around time,average waiting time,average elapsed time and completion time for same input of different cpu scheduling is as follows:-

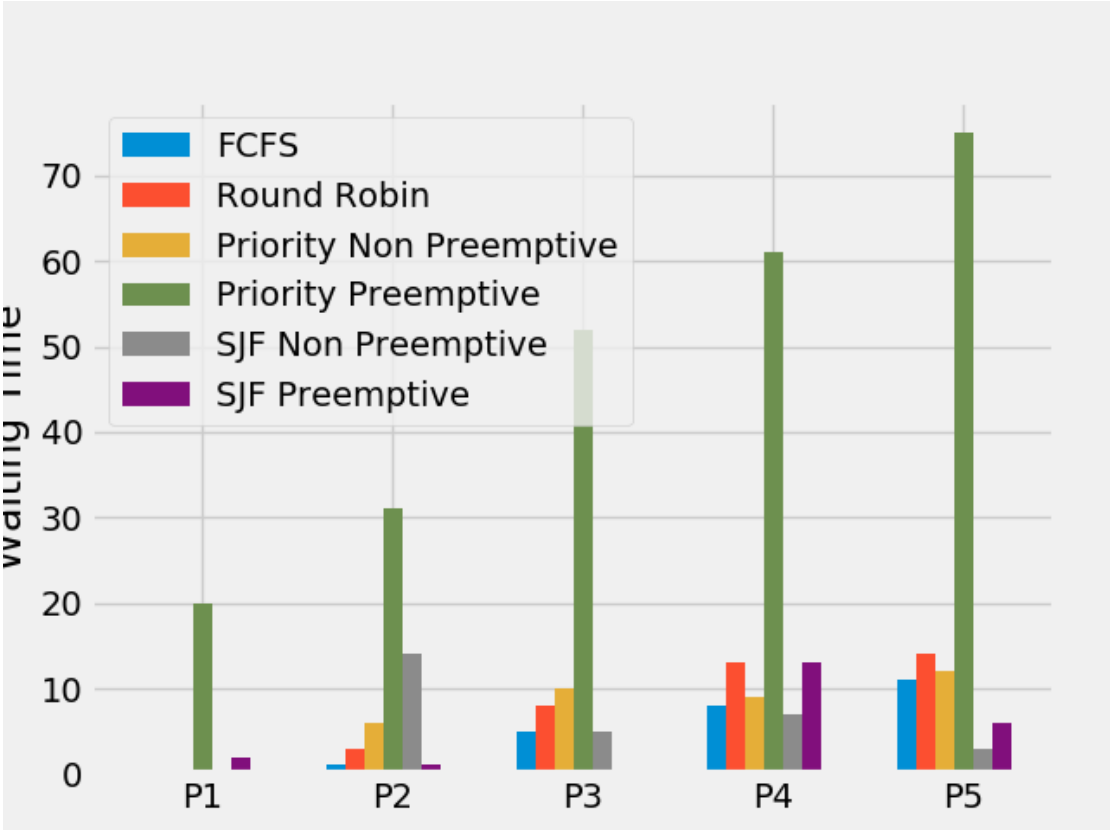

Figure 8: comparison between CPU scheduling.

from the above graph we came to know that every algorithm works better on the significant problem as the fcfs is better for a small burst time.The sjf is better if the process comes to processor simultaneously and round robin, is better to adjust the average waiting time desired and the priorty works better where the relative important of each process may be precisely defined.

References

The source code of the project can be found at:-
`https://github.com/cannibalcheeseburger/cpu-scheduling-simluation.git`
And the other references we used for this project are given:-

- A. Dusseau, R. H. dan A. C., Operating Systems: Three Easy Pieces, Arpaci-Dusseau Books, 2014.
- Operating System Principles – Galvin
- Tanenbaum, Modern Operating Systems, Pearson Education, Inc., 2008.
- `http://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/5_CPU_Scheduling.html`
- `http://codex.cs.yale.edu/avi/os-book/OS8/os8c/slide-dir/PDF-dir/ch5.pdf`