**SCHEDULING SMARTER: EXPLORING FIRST COME FIRST SERVE AND**

**SHORTEST JOB FIRST ALGORITHMS**

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

## Bachelor of Technology/Master of Technology

In

## Computer Science and Engineering School of Engineering and Sciences

Submitted by

## Candidate Name (APXXXXXXXXXXX)



Under the Guidance of

## (Supervisor Name)

**SRM University–AP**

**Neerukonda, Mangalagiri, Guntur Andhra Pradesh – 522 240 [Month, Year]**

# Certificate

Date: 5/18/2023

This is to certify that the work present in this Project entitled “**Scheduling Smarter: Exploring First Come First Serve and Shortest Job First Algorithms**” has been carried out by **[Name of the Candidate]** under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in **School of Engineering and Sciences**.

## Supervisor

(Signature)

Prof. / Dr. [Name] Designation,

Affiliation.

## Co-supervisor

(Signature)

Prof. / Dr. [Name] Designation,

Affiliation.

# AIM OF PROJECT:

# Implementation of Scheduling Smarter: Exploring First Come First Serve and Shortest Job First Algorithms

# DISCRIPTION:

# Scheduling is the process of assigning resources to tasks in a way that meets the requirements of the tasks and the resources. There are many different scheduling algorithms, each with its own advantages and disadvantages.

# Two of the simplest and most commonly used scheduling algorithms are first come first serve (FCFS) and shortest job first (SJF).

# First Come First Serve

# FCFS is a simple algorithm that schedules tasks in the order in which they arrive. This means that the first task to arrive is the first task to be scheduled. FCFS is a fair algorithm, as all tasks are given an equal chance of being scheduled first. However, FCFS can be inefficient, as it may not always schedule the shortest tasks first.

# Shortest Job First

# SJF is a more efficient algorithm than FCFS. SJF schedules the shortest task first, regardless of when it arrived. This means that SJF can minimize the amount of time that tasks spend waiting to be scheduled. However, SJF can be unfair, as tasks that arrive early may have to wait longer to be scheduled than tasks that arrive later.

# Implementation of FCFS and SJF

# FCFS and SJF can be implemented in a variety of ways. One way to implement FCFS is to use a queue. A queue is a data structure that stores tasks in the order in which they arrived. The first task in the queue is the first task to be scheduled.

# SJF can be implemented using a priority queue. A priority queue is a data structure that stores tasks in the order of their priority. The highest priority task in the queue is the first task to be scheduled.

**ROLES AND RESPONSIBILITIES:**

**Resources Management : (Team member name)**

(Team member ) collected information which is required to the ppt and report and helped in preparing ppt and report.

**Code Implementation: (Team member name)**

(Team member ) implemented python script to handle keyboard interrupt

Using some native modules which is available in python.

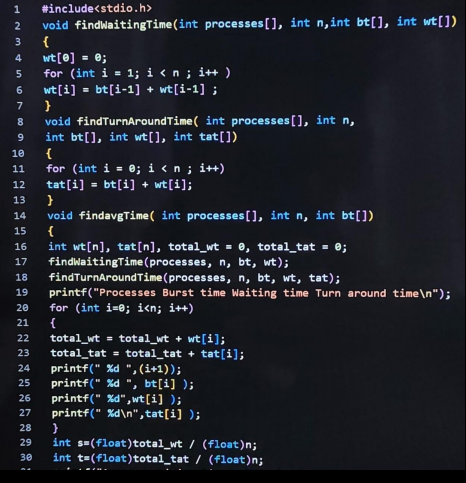
**Report and PPT Preparation: (Team member name)**

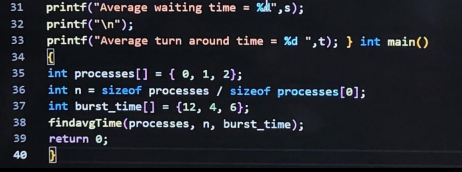
(Team member ) worked on ppt and report which is a refence of our project.

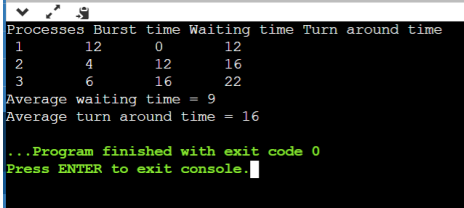
.

**CODE:**

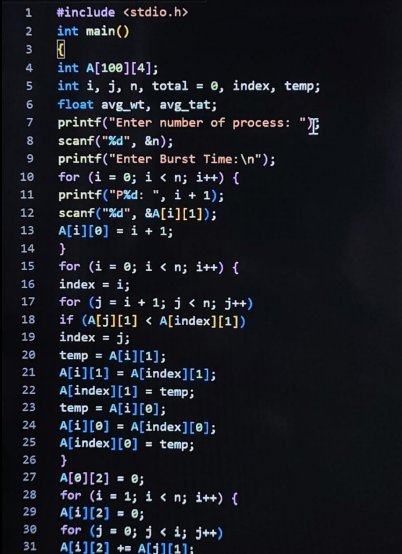
First come First Serve (FCFS):

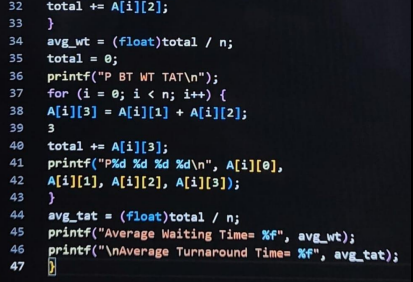


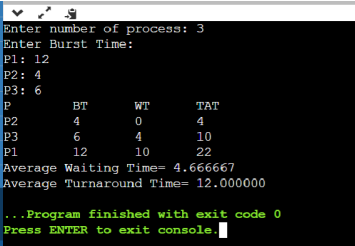




Shortest job First(SJF):







**EXPLANATION:**

First come first serve (FCFS) is a CPU scheduling algorithm that schedules processes in the order in which they arrive at the ready queue. This means that the process that has been waiting in the ready queue the longest is the next process to be scheduled on the CPU.

FCFS is a simple and fair scheduling algorithm. It is simple to implement and understand, and it ensures that all processes are given an equal chance to be scheduled on the CPU. However, FCFS can be inefficient, as it may not always schedule the shortest processes first.

Here are the steps involved in FCFS scheduling:

Processes arrive at the ready queue and are placed in order of arrival.

The process at the front of the queue is scheduled on the CPU.

The process executes on the CPU until it finishes or is interrupted by a higher-priority process.

The process is removed from the ready queue and placed in the finished queue.

The next process in the ready queue is scheduled on the CPU.

Here is an example of FCFS scheduling:

Process A arrives at the ready queue.

Process B arrives at the ready queue.

Process C arrives at the ready queue.

Process A is scheduled on the CPU.

Process A finishes executing.

Process B is scheduled on the CPU.

Process B finishes executing.

Process C is scheduled on the CPU.

Process C finishes executing.

FCFS scheduling is a simple and fair scheduling algorithm, but it can be inefficient. If you have a system with a lot of short processes, FCFS scheduling can lead to a lot of context switching, which can slow down the system.

Here are some of the advantages of FCFS scheduling:

Simple to implement and understand

Fair to all processes

Here are some of the disadvantages of FCFS scheduling:

Can be inefficient

Can lead to a lot of context switching

The shortest job first (SJF) scheduling algorithm is a CPU scheduling algorithm that schedules processes in the order of their estimated execution time. This means that the process with the shortest estimated execution time is the next process to be scheduled on the CPU.

SJF is a simple and efficient scheduling algorithm. It is simple to implement and understand, and it ensures that the processes are scheduled in the order that will minimize the average waiting time for all processes. However, SJF can be unfair, as processes that arrive early may have to wait longer to be scheduled than processes that arrive later.

Here are the steps involved in SJF scheduling:

Processes arrive at the ready queue and are placed in order of estimated execution time.

The process at the front of the queue is scheduled on the CPU.

The process executes on the CPU until it finishes or is interrupted by a higher-priority process.

The process is removed from the ready queue and placed in the finished queue.

The next process in the ready queue is scheduled on the CPU.

Here is an example of SJF scheduling:

Process A arrives at the ready queue and has an estimated execution time of 10 milliseconds.

Process B arrives at the ready queue and has an estimated execution time of 20 milliseconds.

Process C arrives at the ready queue and has an estimated execution time of 5 milliseconds.

Process C is scheduled on the CPU.

Process C finishes executing after 5 milliseconds.

Process A is scheduled on the CPU.

Process A finishes executing after 10 milliseconds.

Process B is scheduled on the CPU.

Process B finishes executing after 20 milliseconds.

SJF scheduling is a simple and efficient scheduling algorithm, but it can be unfair. If you have a system with a lot of short processes, SJF scheduling can lead to a lot of context switching, which can slow down the system.

Here are some of the advantages of SJF scheduling:

Simple to implement and understand

Efficient

Minimizes average waiting time

Here are some of the disadvantages of SJF scheduling:

Can be unfair

Can lead to a lot of context switching

**APPLICATIONS:**

A real-life example of the FCFS method is buying a movie ticket on the ticket counter. In this scheduling algorithm, a person is served according to the queue manner.Theperson who arrives first in the queue first buys the ticket and then the next one. This will continue until the last person in the queue purchases the ticket. Using this algorithm,the CPU process works in a similar manner.

FCFS

FCFS is often used in batch processing systems, where a large number of jobs are submitted to the system and scheduled in the order in which they arrive.

FCFS can also be used in real-time systems, where it is important to ensure that all processes are given an equal chance to be scheduled.

For example, FCFS could be used to schedule the execution of tasks in a factory, where it is important to ensure that all tasks are completed in a timely manner.

SJF

SJF is often used in systems where it is important to minimize the average waiting time for all processes.

For example, SJF could be used to schedule the execution of tasks in a web server, where it is important to ensure that all requests are responded to in a timely manner.

SJF can also be used in systems where it is important to ensure that the most important tasks are completed first.

For example, SJF could be used to schedule the execution of tasks in an operating room, where it is important to ensure that the most critical patients are treated first.

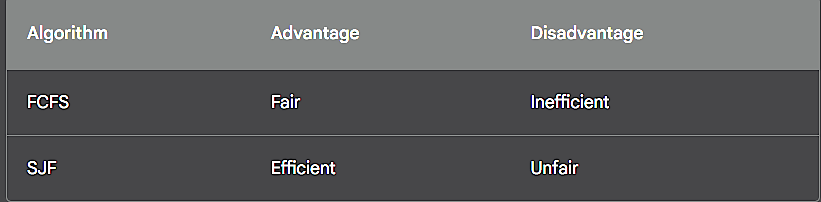
**CONCLUSION:**

Sure, here is a conclusion statement for FCFS and SJF scheduling algorithms:

FCFS and SJF are two simple and commonly used scheduling algorithms. FCFS schedules processes in the order in which they arrive, while SJF schedules processes in the order of their estimated execution time. FCFS is fair, as all processes are given an equal chance to be scheduled first. SJF is efficient, as it minimizes the average waiting time for all processes. However, SJF can be unfair, as processes that arrive early may have to wait longer to be scheduled than processes that arrive later.

The best algorithm to use depends on the specific requirements of the processes and the resources. If fairness is important, then FCFS may be the best option. If efficiency is important, then SJF may be the best option. In some cases, a hybrid algorithm may be the best option. A hybrid algorithm combines the features of two or more algorithms. For example, a hybrid algorithm could use FCFS for tasks that are not time-sensitive and SJF for tasks that are time-sensitive.

Here is a table summarizing the advantages and disadvantages of FCFS and SJF scheduling algorithms:

****