Dharmsinh Desai University

Academic Year 2022-23

Department:

Faculty of Management and information science

Subject:

Operating System and Linux Programming

Full Name: Sutariya Savankumar Sureshbhai

Roll No.: MA065

ID No.: 22MAPOG030

Submitted to: Prof. Narayan Joshi | MCA Department

Student sign. **Professor sign.**

Term work details

a. Problem Definition:

The goal of this project is to develop a web-based process scheduler simulator that can help users understand and evaluate different scheduling algorithms and their effectiveness under different workloads. The simulator should provide a user-friendly interface that allows users to select different scheduling algorithms and set various parameters(time quantum, priority etc.), and should be able to generate visualizations of the scheduling results, such as a Gantt chart or table showing the sequence of processes and their execution times.

b. Desired Outcomes & Output:

The desired outcome of this project is a fully-functional web-based process scheduler simulator that can handle a range of different scheduling algorithms and parameters. The simulator should be able to generate visualizations of the scheduling results, such as Gantt charts or tables, and provide users with the ability to compare the performance of different algorithms under different workloads.
Algorithms will be covered : FCFS

c. Methodology:

Using JavaScript to create a web-based simulator that can handle different scheduling algorithms and parameters. Designing the user interface using HTML and CSS. The final output of the project will be a deployed web-based process scheduler simulator that can be accessed on any web browser.

About CPU Scheduling Algorithms

The CPU Scheduling is the process by which a process is executed by the using the resources of the CPU. The process also can wait due to the absence or unavailability of the resources. These processes make the complete use of Central Processing Unit.

- **FCFS (First Come First Served):**
  This algorithm schedules processes in the order they arrive. It is non-preemptive and has a simple implementation. However, it may result in long waiting times for processes with long burst times. A visualization of this algorithm shows that the CPU runs each process to completion before starting the next process.

  - **Characteristics:**
    - Implementation is simple.
    - Does not cause any causalities while using
    - It adopts a non pre emptive and pre emptive strategy.
    - It runs each procedure in the order that they are received.
    - Arrival time is used as a selection criterion for procedures.

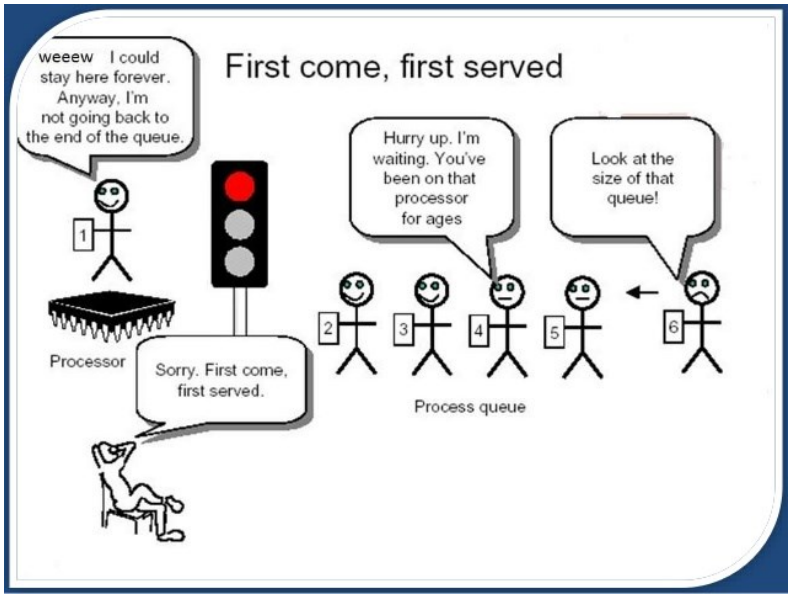
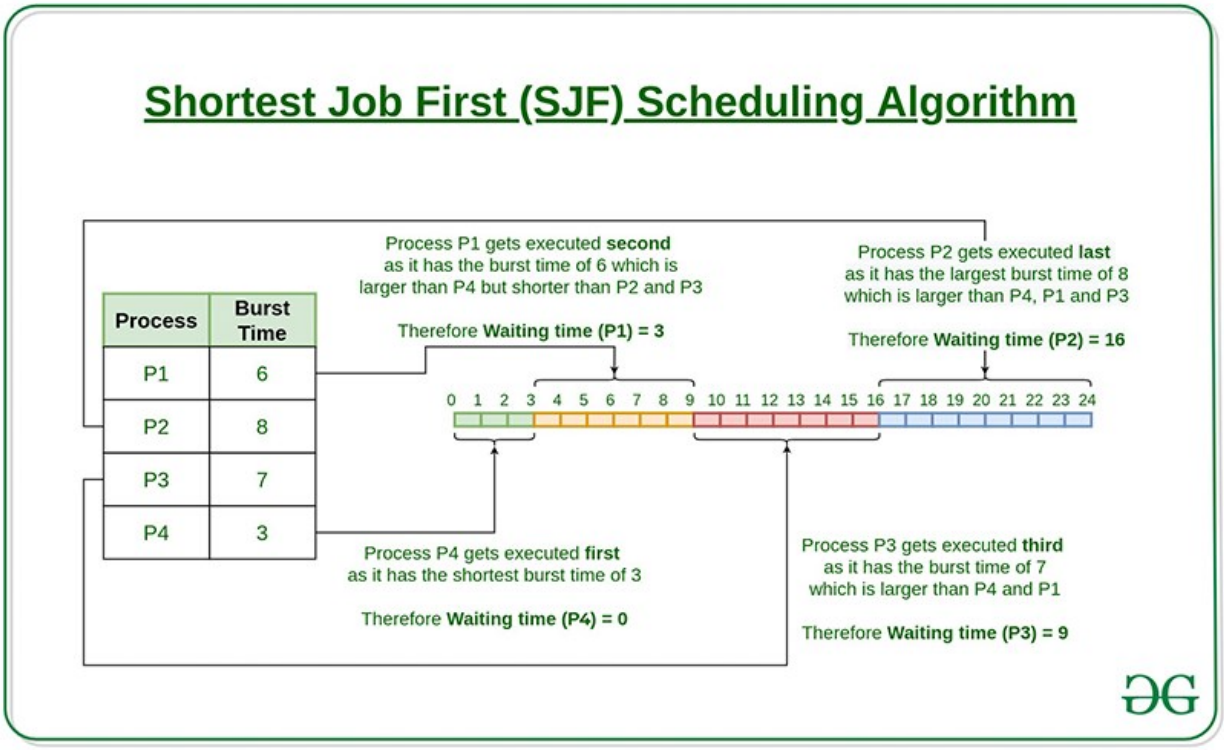


Figure (1: First Come First Served)

- **SJF (Shortest Job First):**

  The shortest job first (SJF) or shortest job next, is a scheduling policy that selects the waiting process with the smallest execution time to execute next. SJN, also known as Shortest Job Next (SJN)

  - **Characteristics:**
    - Associated with every job as it requires a unit of time for a job to complete.
    - Helpful for batch-type processing.
    - Improves process throughput as shorter jobs execute first, further reducing the turnaround time.
    - Offers shorter jobs that improve job output.



- **SRTF (Shortest Remaining Time First):**

  Shortest Remaining Time First (SRTF) is the preemptive version of Shortest Job Next (SJN) algorithm, where the processor is allocated to the job closest to completion.
  This algorithm requires advanced concept and knowledge of CPU time required to process the job in an interactive system, and hence can't be implemented there. But, in a batch system where it is desirable to give preference to short jobs, SRT algorithm is used.

- **Characteristics:**
  - Processes with a shorter burst time are executed quickly.
  - Offers shorter jobs that improve job output.
  - Preemptive

Solution (Process Scheduler Simulator)

The purpose of this app is to provide a CPU scheduling algorithm visualization tool on a webpage. It allows users to simulate and visualize the behavior of different CPU scheduling algorithms and their effects on the CPU and waiting queue.

Algorithms covers (currently)

- **FCFS**(First Come First Served)
- **SJF**(Shortest Job First)
- **SRTF**(Shortest Remaining Time First)

Features

- Insert custom processes into the process table.
- Generate random values into the process table to save the time.
- Select Algorithm to execute.
- Adjust the speed of execution and animations.
- Live flow of process execution from process table to terminated table.
- Average turnaround time and average waiting time.
- Execution log.

Usage

Open this URL https://freshhminds.me/CSAVisualizer/

Process Scheduler Simulator

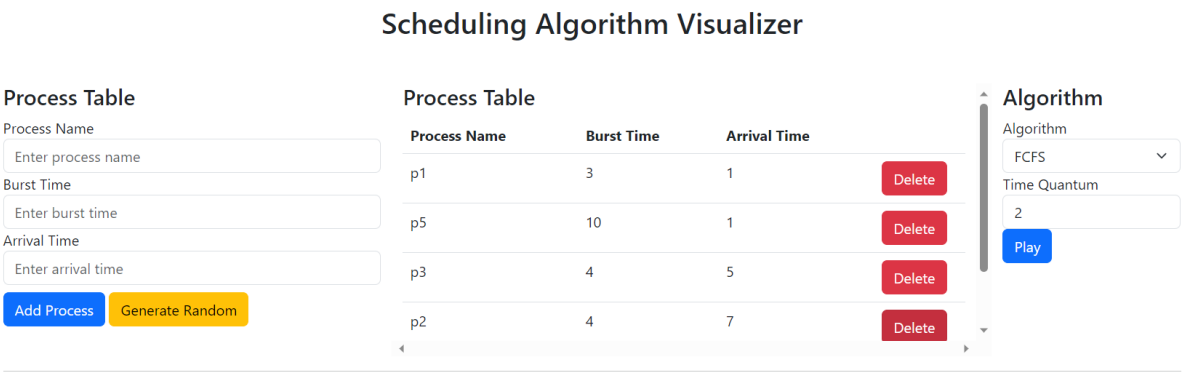


Figure (2.1: Input processes to visualize and select the algorithm)

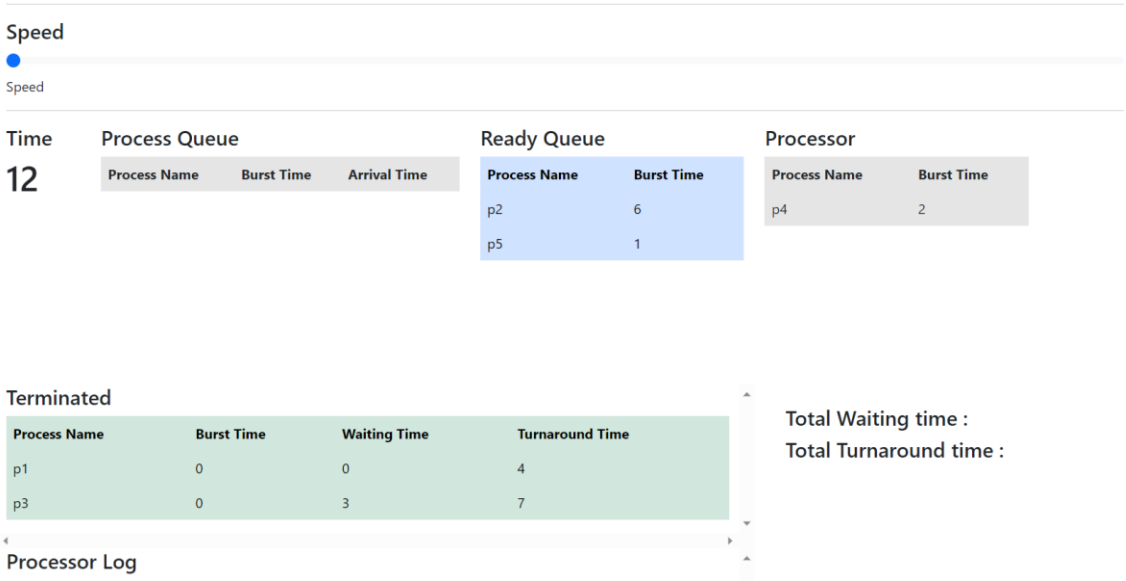


Figure (2.2: Control of simulation speed and view how a process executes)

Figure (2.3: Finally results and processor log will be displayed)

- # Technical Details

Programming language: JavaScript
User Interface: HTML with bootstrap
Tools and services:

- Git/GitHub (Version control)
- GitHub pages (Hosting)
- Selenium IDE (Testing)

- # Files
  - **index.html**
    - Contains the html structures including insertion forms/buttons, relevant tables and results.
    - It uses bootstrap grid system to organize the components.

- **script.js**
  - Contains User input functions and validations to ensure proper entry of data.

- **Algorithms.js**
  - It is the entry point to execute the simulation
  - Contains these functions:
    - **executeAlgorithm() ->** Executes selected algorithm
    - **getRandomData(n) ->** Generates n number of random data to save inserting time.
    - **FCFS()**
      - Which executes First Come First Served method simulation.
      - It iterates according to the selected speed by the user.
      - Tasks performed
        - Take process from process table to readyQueue according to arrival time
        - Send the process from queue to processor if processor is idle.
        - Calculate waiting and turnaround time for each process.

- **ui.js**
  - It basically controls and manages the changes in HTML file during the execution of the algorithm (Animations).

References

- https://www.geeksforgeeks.org/first-come-first-serve-cpu-scheduling-non-preemptive/