Interactive Visual Scheduler Simulator

Student: Lang, Yanwu; Ye, Zhentong
Advisor: Dr. Hamza Djigal Computer Science, College of Science Mathematics and Technology
温州肯恩大学 WENZHOU-KEAN UNIVERSITY

Introduction

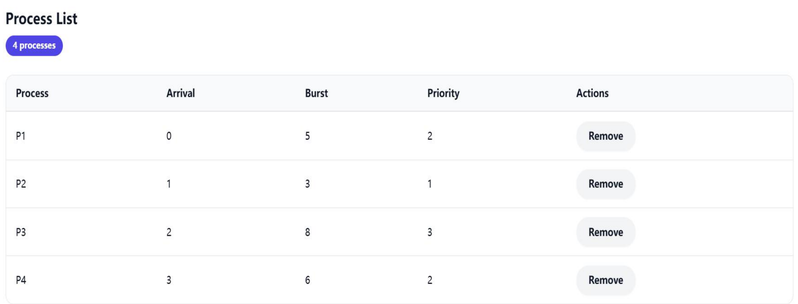
To enhance teaching, learning, and experimentation for professors and students, this project developed a web-based CPU scheduling visualization simulator.
This system supports six classic scheduling algorithms (FCFS, SJF, SRTF, Priority Scheduling, HRRN, RR and HEFT) and visualizes the CPU scheduling process through dynamic Gantt graphs, real-time CPU and ready queue animations, and step-by-step analysis and explanation.

System Architecture

I. Front-end framework: HTML + CSS + JavaScript
II. Page modules: Setup / Simulate / Compare / HEFT
III. State management: localStorage persistence
IV. Scheduling core: script_new.js handles all algorithm logic
V. Presentation core: Dynamic DOM rendering + CSS animations
VI. Deployment method: GitHub Pages /docs/index.html

Key Features

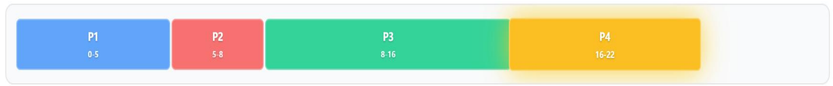
1. Setup Module
Used to initialize data, providing the Process List data to be scheduled (providing basic examples of various types), and selecting the CPU scheduling method to be used.



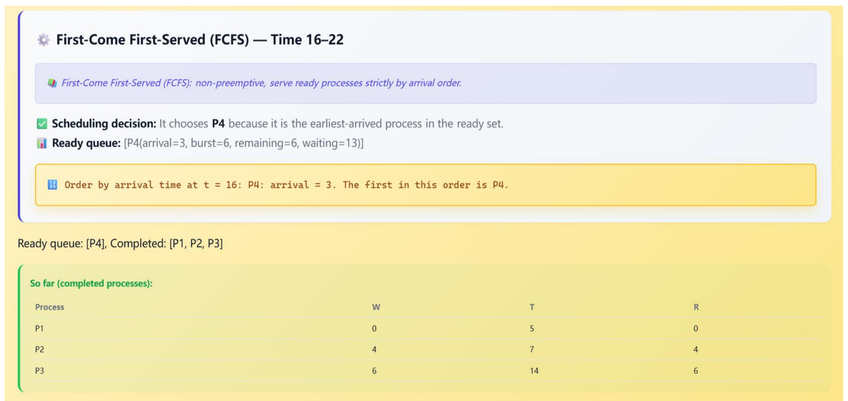
2. Scheduling Method Simulate
The Simulate section allocates the user's process list using the selected CPU scheduling method and visualizes the process. Users can control the animation to stop or continue using simulation controls, visualize the Ready Queue, and view the Gantt Chart. Each step is explained using an algorithm, and finally, the decision log and performance metrics are displayed. The FCFS algorithm will be used as an example next.

CPU & Ready Queue

Gantt Chart



Algorithm Explanation

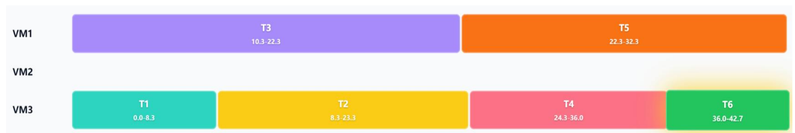


3. Algorithm Comparison
The Algorithm Comparison section compares different CPU scheduling methods under the same process list.

4. HEFT Scheduling for Cloud/Distributed Systems
The system also deploys visualizations of the HEFT scheduling method, including DAG, Gantt chart, step-by-step interpretation algorithm, and decision log.

HEFT Task Graph (DAG)

HEFT Gantt Chart (Multi-Processor)



Conclusions

This project successfully built an interactive, deployable, and educational CPU scheduling simulator, which can help to better understand the operation of various CPU scheduling methods and more conveniently analyze scheduling problems in real projects.