

CPU Scheduler Project – Complete Phases

Phase 1: Process Input and Selection of Scheduling Algorithm

The user can input the name, arrival time, burst time, and optionally the priority of each process. An algorithm is selected from FIFO, SJF, SRTF, Priority (cooperative/preemptive), and Round Robin.

Phase 2: Execution Queue and Step-by-step Explanation

The logic and steps taken by the algorithm are shown, detailing the waiting and response times. The final execution queue is also visualized.

Phase 3: Gantt Chart and Result Table

A dynamic Gantt chart is rendered with colors by process and support for download. A results table is also shown with average waiting and response time calculations.

Phase 4: Automated Testing and CI/CD

Unit tests implemented using pytest. CI pipeline configured with GitHub Actions.

Phase 5: Efficiency Metrics

Metrics such as context switches, CPU usage percentage, and total wait time are calculated. Comparative bar charts between algorithms are generated.

Phase 6: Automatic Algorithm Recommendation

A rules-based engine suggests the most appropriate algorithm based on patterns in the input data, such as similar bursts, staggered arrivals, or varying priorities.

Phase 7: Step-by-step Simulation

Interactive simulation where users can play, pause, step, or restart execution. Each quantum or execution is visualized with dynamic feedback and animations.

Phase 8: Decision Panel

For each step, a panel explains why the chosen process was selected based on the algorithm's criteria (e.g., shortest burst, lowest priority, or arrival order). This is ideal for education and debugging.

Phase 9: Save and Load Results

Full simulation results (data, metrics, Gantt, recommendation, configuration) can be exported and reloaded from a JSON file. This allows sharing and continuing analysis later.

Phase 10: Persistent Configurations

Users can save their preferred settings (language, algorithm, number of processes, quantum). A JSON configuration file can be downloaded or reloaded in future sessions.