OS Project

Cpu scheualing

Rawan sous

Rivan jaradat

Dana ismail

Project describtion:

CPU Scheduling Simulator

Design and implement various CPU scheduling algorithms, such as Round Robin, Priority

Scheduling, or Shortest Job First (SJF). Allow users to configure scheduling parameters,

simulate task arrival and completion, and evaluate scheduling performance metrics (e.g.,

turnaround time, waiting time).

Programming language used :python

Workflow:

We divided the step to three parts, each will be assigned to manage one part,but also communicate and

Debug corrunntely to insure the proper working of the functionalities and the alghorithims

Person 1:

Scheduling Algorithm Implementation

Implement the Round Robin scheduling algorithm.

Determine the order in which processes or threads are selected for execution.

Scheduling Parameters Configuration:

Implement the functionality to configure the time quantum for Round Robin.

Provide an interface or commands to set and adjust this parameter dynamically.

Task Arrival and Completion Simulation:

Simulate the arrival of tasks or processes into the system.

Implement the mechanism to generate a sequence of tasks or processes with their associated arrival times and burst times.

Person 2:

Scheduling Algorithm Implementation:

Implement the Priority Scheduling algorithm.

Determine the order in which processes or threads are selected for execution based on priority levels.

Scheduling Parameters Configuration:

Implement the functionality to configure priority levels for Priority Scheduling.

Provide an interface or commands to set and adjust these parameters dynamically.

Task Arrival and Completion Simulation:

Simulate the completion of tasks based on their burst times or execution time.

Person 3:

Scheduling Algorithm Implementation:

Implement the Shortest Job First (SJF) scheduling algorithm.

Determine the order in which processes or threads are selected for execution based on burst time.

Performance Metrics Evaluation:

Calculate and evaluate performance metrics such as turnaround time, waiting time, and response time for all implemented scheduling algorithms.

Update and track these metrics as tasks or processes are executed and completed.

Visualization and Reporting:

Develop visualizations, such as Gantt charts, to showcase the scheduling activities and execution timeline of tasks or processes.

Generate reports or summaries that present the performance metrics for each scheduling algorithm.

Implement the functionality to compare the performance of different scheduling algorithms based on the metrics obtained.