CMPS442 Operating Systems

Course Project

The goal of your CMPS442 course project is to test the operations of the Operating Systems Scheduler Simulator. You should execute the requirements of the project and submit a professional report that contains the complete requirements stated below.

Requirements

You should submit a report that should contain projects from the sections stated below. So, your report should include **two projects**. Choose **one** project from **Part A**. In addition, the other project should be **Part B, project number 7 (mandatory)**. The length of the Report should be **between 12 to 15 pages**. You should use the font: **Times New Roman** with font size equal to 12 and **line spacing equal to 1.5**. The report should contain the following sections:

- **Introduction**: An Introduction that describes the simulator you are using and its details, and the objectives of your project.
- **Problem Statement and Experimentation**: Details of the scheduling simulation that you did. The steps of the simulation, the screenshots that you took during the simulation, and the various results that you obtained. You MUST show a screenshot and the result of the execution of each step: From the beginning of the simulation until all processes are executed. Also, you must show the scheduling output as a diagram similar to the scheduling diagrams that we took in the class.
- **Discussion**: How do you describe and interpret the results that you obtained? How do you describe what happened in the simulation, and what do you deduce from what happened in the simulation? How do you compare the performance of the simulator that you studied in your experiment with what we have taken in class?
- **Conclusion**: A summary of what you did and the conclusions that you reached, in addition to any ideas you propose or points of view that you suggest.

Evaluation

The grade of the project will be divided as follows:

Correct setup of the experiment .... (20%)
Calculation of all Results .... (30%)
Correctness of the Results...(20%)
General format and appearance of the report .... (15%)
Discussion of Results .... (15%)

Projects

Part A (choose only one)

Project Number 1:

Generate the following simulation:

- Number of Processes = 10
- DOM = 20
- Number of Bursts = 5
- CPU Bursts Range: Minimum = 10, Maximum = 30
- I/O Bursts Range: Minimum = 50, Maximum = 200
- Priority: Minimum = 1, Maximum = 5
- Arrival time: Minimum = 0, Maximum = 30
- Context Switch = 2
- Alpha = 0.5, Tau = 50

- Use the "Priority" Scheduling algorithm

- Show the configuration of the processes before you start.
- Move the simulator step by step.
- Whenever a change occurs to one of the processes, save the change and take a screenshot if necessary.
- When all processes finish execution, examine and analyze the results that you obtained and discuss them and whether they were exactly as you expected and what are the new things that you learned from the simulation.
- Draw the Output of the simulation as a scheduling diagram similar to what we have taken in class, showing the processes and the times at which each process was executed.

Project Number 2:

Generate the following simulation:

- Number of Processes = 8
- DOM = 20
- Number of Bursts = 10
- CPU Bursts Range: Minimum = 5, Maximum = 25
- I/O Bursts Range: Minimum = 30, Maximum = 150
- Priority: Minimum = 1, Maximum = 3
- Arrival time: Minimum = 0, Maximum = 20
- Context Switch = 2
- Alpha = 0.5, Tau = 50

- Use the “Shortest Job First” Scheduling algorithm

- Show the configuration of the processes before you start.
- Move the simulator step by step.
- Whenever a change occurs to one of the processes, save the change and take a screenshot if necessary.
- When all processes finish execution, examine and analyze the results that you obtained and discuss them and whether they were exactly as you expected and what are the new things that you learned from the simulation.
- Draw the Output of the simulation as a scheduling diagram similar to what we have taken in class, showing the processes and the times at which each process was executed.

Project Number 3:

Generate the following simulation:

- Number of Processes = 12
- DOM = 20
- Number of Bursts = 15
- CPU Bursts Range: Minimum = 5, Maximum = 20
- I/O Bursts Range: Minimum = 20, Maximum = 100
- Priority: Minimum = 1, Maximum = 7
- Arrival time: Minimum = 0, Maximum = 25
- Context Switch = 2
- Alpha = 0.5, Tau = 50

- Use the “FCFS” Scheduling algorithm

- Show the configuration of the processes before you start.
- Move the simulator step by step.

- Whenever a change occurs to one of the processes, save the change and take a screenshot if necessary.
- When all processes finish execution, examine and analyze the results that you obtained and discuss them and whether they were exactly as you expected and what are the new things that you learned from the simulation.
- Draw the Output of the simulation as a scheduling diagram similar to what we have taken in class, showing the processes and the times at which each process was executed.

Project Number 4:

Generate the following simulation:

- Number of Processes = 9
- DOM = 20
- Number of Bursts = 7
- CPU Bursts Range: Minimum = 10, Maximum = 40
- I/O Bursts Range: Minimum = 100, Maximum = 300
- Priority: Minimum = 1, Maximum = 5
- Arrival time: Minimum = 0, Maximum = 35
- Context Switch = 2
- Alpha = 0.5, Tau = 50

- Use the "Round Robin" Scheduling algorithm with Time Quantum equal to 5

- Show the configuration of the processes before you start.
- Move the simulator step by step.
- Whenever a change occurs to one of the processes, save the change and take a screenshot if necessary.
- When all processes finish execution, examine and analyze the results that you obtained and discuss them and whether they were exactly as you expected and what are the new things that you learned from the simulation.
- Draw the Output of the simulation as a scheduling diagram similar to what we have taken in class, showing the processes and the times at which each process was executed.

Project Number 5:

Generate the following simulation:

- Number of Processes = 15
- DOM = 20
- Number of Bursts = 6
- CPU Bursts Range: Minimum = 10, Maximum = 15
- I/O Bursts Range: Minimum = 20, Maximum = 75
- Priority: Minimum = 1, Maximum = 10
- Arrival time: Minimum = 0, Maximum = 45
- Context Switch = 2
- Alpha = 0.5, Tau = 50

- Use the “Shortest Remaining Job First” Scheduling algorithm

- Show the configuration of the processes before you start.
- Move the simulator step by step.
- Whenever a change occurs to one of the processes, save the change and take a screenshot if necessary.
- When all processes finish execution, examine and analyze the results that you obtained and discuss them and whether they were exactly as you expected and what are the new things that you learned from the simulation.
- Draw the Output of the simulation as a scheduling diagram similar to what we have taken in class, showing the processes and the times at which each process was executed.

Project Number 6:

Generate the following simulation:

- Number of Processes = 13
- DOM = 20
- Number of Bursts = 5
- CPU Bursts Range: Minimum = 10, Maximum = 25
- I/O Bursts Range: Minimum = 10, Maximum = 100
- Priority: Minimum = 1, Maximum = 5
- Arrival time: Minimum = 0, Maximum = 30
- Context Switch = 2
- Alpha = 0.5, Tau = 50

- Use the “Round Robin” Scheduling algorithm with Time Quantum equal to 10

- Show the configuration of the processes before you start.
- Move the simulator step by step.

- Whenever a change occurs to one of the processes, save the change and take a screenshot if necessary.
- When all processes finish execution, examine and analyze the results that you obtained and discuss them and whether they were exactly as you expected and what are the new things that you learned from the simulation.
- Draw the Output of the simulation as a scheduling diagram similar to what we have taken in class, showing the processes and the times at which each process was executed.

Part B (Mandatory)

Project Number 7:

Generate the following simulation:

- Number of Processes = 4
- Number of Resources = 3
- Max Matrix:

	R1	R2	R3
P1	7	5	3
P2	3	2	2
P3	9	0	2
P4	2	2	2

- Allocation Matrix:

	R1	R2	R3
P1	0	1	0
P2	2	0	0
P3	3	0	2
P4	2	1	1

- Available Vector: R1=3 ; R2=3 ; R3=4
- Context Switch = 2
- Alpha = 0.5, Tau = 50

- Use the "Bankers Algorithm" to find the Need Matrix and to check if the system is in safe state.
- Show the configuration of the processes before you start.
- Move the simulator step by step.

- Whenever a change occurs to one of the processes, save the change and take a screenshot if necessary.
- Write the safe sequence of the algorithm similar to what we have taken inclass, showing the processes ordered in a safe order.
- Try implementing several scenarios like killing processes and requesting more resources and show the changes by taking screenshots.
- Check again if the system will remain in safe state.