RTS LAB Assignment 1

- **Name:** Shubham Kishor Vispute
- **BITS ID:** 2024HT01026

- **Course Name:** Real Time Systems
- **Date of Submission:** 27/10/2024

PART A

Question 1: Rate Monotonic Scheduling (RMS) Analysis

Task Set:

Task	Period (ms)	Execution Time (ms)	Utilization (U = Execution Time / Period)
T1	4	1	U1 = 1/4 = 0.25
T2	5	2	U2 = 2/5 = 0.40
T3	10	3	U3 = 3/10 = 0.30

Total utilization Utotal = U1 + U2 + U3 = 0.25 + 0.40 + 0.30 = 0.95

Schedulability Check: For RMS, the utilization threshold for **n** tasks is

Uthreshold = n(2^(1/n)−1).

For 3 tasks, the threshold is:

Uthreshold = 3(2^(1/3)−1)= 0.77976 ≈ 0.78

Since Utotal = 0.95 > Uthreshold = 0.78, the task set is **not schedulable** under RMS.

Question 2: Earliest Deadline First (EDF) Scheduling

Response Times:

- **T1:** Since T1 has the earliest deadline (4 ms), it will be executed first. Its execution time is 1 ms, so its response time is:

  RT1 = 1 ms

- **T2:** After T1 finishes, T2 will start. T2 has a deadline of 5 ms and needs 2 ms to complete. Since T1 finishes at 1 ms, T2 starts at 1 ms and finishes at 3 ms. Thus, its response time is:

  RT2 = 3 ms

- **T3:** After T2 finishes, T3 will start. T3 has a deadline of 6 ms and needs 1 ms to complete. Since T2 finishes at 3 ms, T3 starts at 3 ms and finishes at 4 ms. Therefore, its response time is:

  RT3 = 4 ms

Task	Period (ms)	Execution Time (ms)	Deadline (ms)	Response Time (ms)
T1	4	1	4	1
T2	5	2	5	3
T3	6	1	6	4

All tasks meet their deadlines under EDF scheduling:

- RT1 = 1 ms (meets the deadline of 4 ms)
- RT2 = 3 ms (meets the deadline of 5 ms)
- RT3 = 4 ms (meets the deadline of 6 ms)

Thus, the task set is **schedulable under EDF**.

Question 3: Task Response Time Calculation

Worst-Case Response Time Calculation:

- R1 = C1 = 2 (no higher-priority tasks)
- R2 = C2 + I1 = 3 + 2 = 5
- R3 = C3 + I1 + I2 = 1 + 2 + 3 = 6

Therefore, the response times are:
R1=2 ms, R2=5 ms, R3=6 ms.

Task	Priority	Period (ms)	Execution Time (ms)	Response Time (ms)
T1	1	6	2	2
T2	2	8	3	5
T3	3	10	1	6

- The **worst-case response time** for each task has been calculated, accounting for preemption by higher-priority tasks.
- Each task meets its deadline since the response times are less than or equal to the task periods:

  $RT1 = 2 \leq 6$

  $RT2 = 5 \leq 8$

  $RT3 = 6 \leq 10$

Therefore, the task set is **schedulable** under the given priority-based scheduling algorithm.

Question 4: Context Switching Impact

Task Set:

Task	Period (ms)	Execution Time (ms)	Context Switches	Effective Execution Time (ms)
T1	5	2	1	2 + 1 = 3
T2	10	3	1	3 + 1 = 4
T3	15	4	2	4 + 2 = 6

Effective Execution Times:

- T1 = 3 ms
- T2 = 4 ms
- T3 = 6 ms

Question 5: Missed Deadlines in a Hyperperiod

Step 1: Calculate the Hyperperiod

The **hyperperiod** is the least common multiple (LCM) of the periods of the tasks.

- Period of T1 = 6 ms
- Period of T2 = 10 ms
- Period of T3 = 12 ms

The **LCM** of 6, 10, and 12 is **60 ms**. So the full hyperperiod is **60 ms**, but the problem asks for the number of deadlines missed in **30 ms**.

Step 2: Calculate the Number of Instances (Deadlines) in 30 ms

To calculate the number of deadlines for each task within 30 ms, we divide the 30 ms by the period of the task.

- T1:
  30 / 6 = 5 deadlines

- T2:
  30 / 10 = 3 deadlines

- T3:
  30 / 12 = 2 deadlines

The number of deadlines in 30 ms is:

- T1: 5 deadlines
- T2: 3 deadlines
- T3: 2 deadlines

Step 3: Calculate the Missed Deadlines

To calculate the missed deadlines, simulate the execution of tasks within the 30 ms window and check if any task's execution finishes after its deadline.

1. **T1 (Period = 6 ms, Execution Time = 2 ms):**
    - The task needs to complete every 6 ms.
    - In 30 ms, there are 5 periods: [0-6], [6-12], [12-18], [18-24], [24-30].
    - Each instance of T1 requires 2 ms to execute, so it will always finish before the next deadline. **No deadlines are missed**.
2. **T2 (Period = 10 ms, Execution Time = 3 ms):**
    - The task needs to complete every 10 ms.
    - In 30 ms, there are 3 periods: [0-10], [10-20], [20-30].
    - Each instance of T2 requires 3 ms to execute, so it will also finish before each deadline. **No deadlines are missed**.

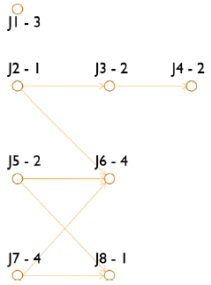
3. **T3 (Period = 12 ms, Execution Time = 4 ms):**
    - The task needs to complete every 12 ms.
    - In 30 ms, there are 2 periods: [0-12], [12-24].
    - Each instance of T3 requires 4 ms to execute, so it will finish before its deadlines. **No deadlines are missed**.

Task	Period (ms)	Execution Time (ms)	Number of Deadlines in Hyperperiod (30 ms)	Deadlines Missed
T1	6	2	5	0
T2	10	3	3	0
T3	12	4	2	0

PART B

Question1 . For the task graph show below show that a priority driven schedule with non-preemption will produce a better schedule than priority driven schedule with pre-emption

All tasks are aperiodic with a deadline of 12. All tasks except for J5 release at 0. J5 releases at 4. The priority of jobs is such that J1 has a higher priority when compared to J2 has a higher.



Solution : The objective of this experiment is to demonstrate that a priority-driven, non-Preemptive schedule can yield a better outcome than a Preemptive schedule for a specific set of aperiodic tasks. The Cheddar scheduling analysis tool is used to simulate and compare both configurations.

Task Set and Configuration

Task Characteristics

- **Number of Tasks**: 5 (labelled J1 to J5)
- **Deadline**: All tasks have a common deadline of 12 units.
- **Release Times**: All tasks are released at time 0, except for J5, which is released at time 4.
- **Priorities**: Tasks are prioritized as follows:

  J1>J2>J3>J4>J5
- **Processor Configuration**: The system has dual identical cores, and jobs are allowed to migrate across cores.

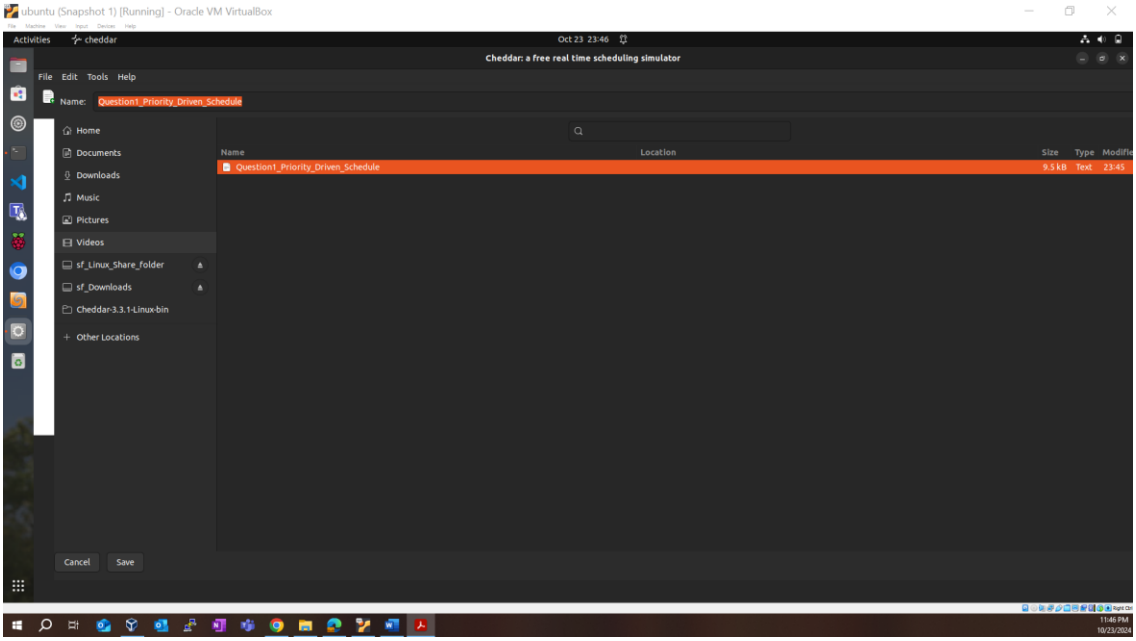
Scheduling Algorithms

1. **Preemptive Priority Scheduling**: Higher-priority tasks can preempt lower-priority tasks.
2. **Non-Preemptive Priority Scheduling**: Tasks are scheduled based on priority but without preemption, meaning a task runs to completion once it starts.

Actual Priority Scheduling Implementation.

1. **Preemptive Priority Scheduling :**

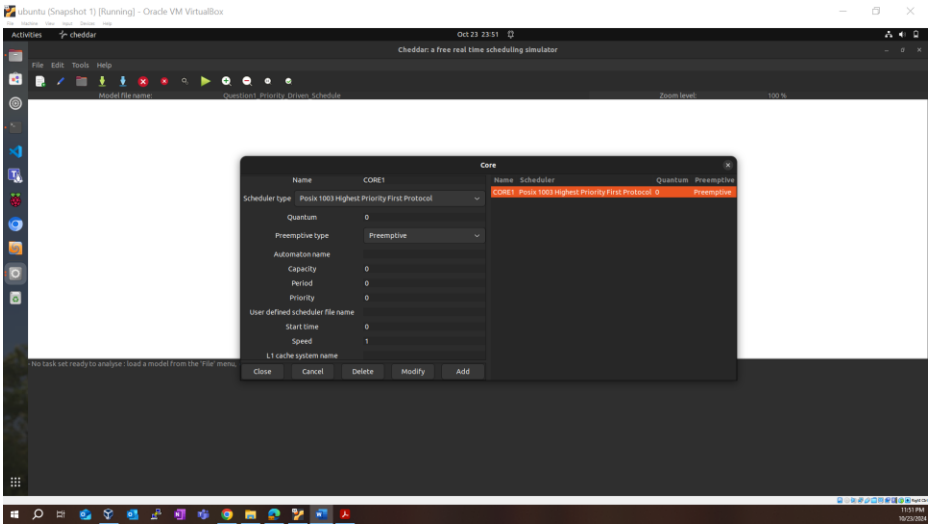
Saved the file =>

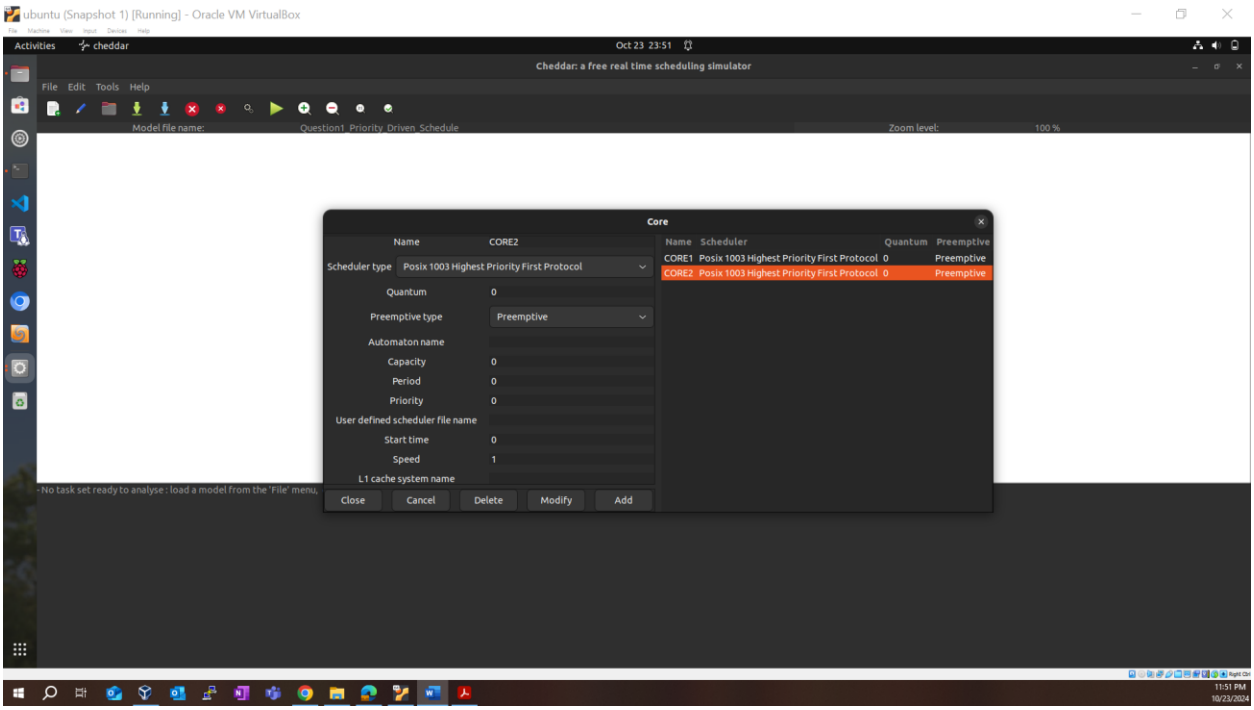


Added CORE1 & CORE2

Schedular type POSIX 1003 Highest Priority First Protocol

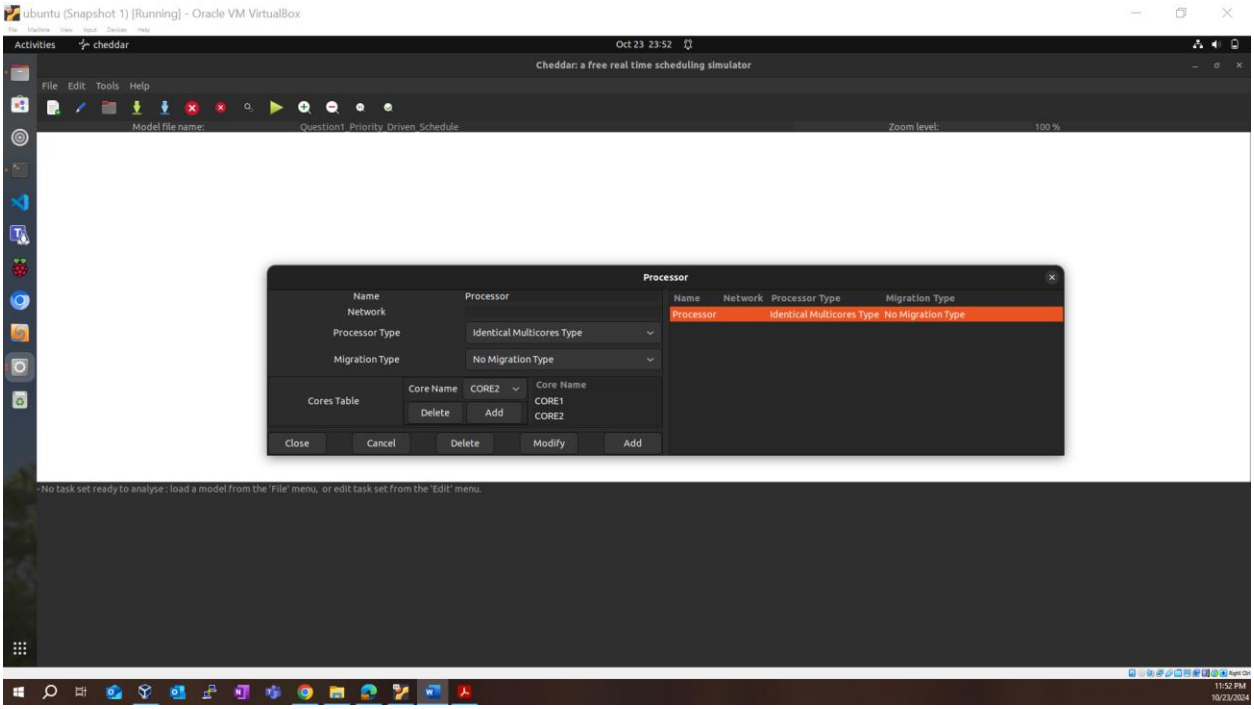
Preemption type : Preemptive =>



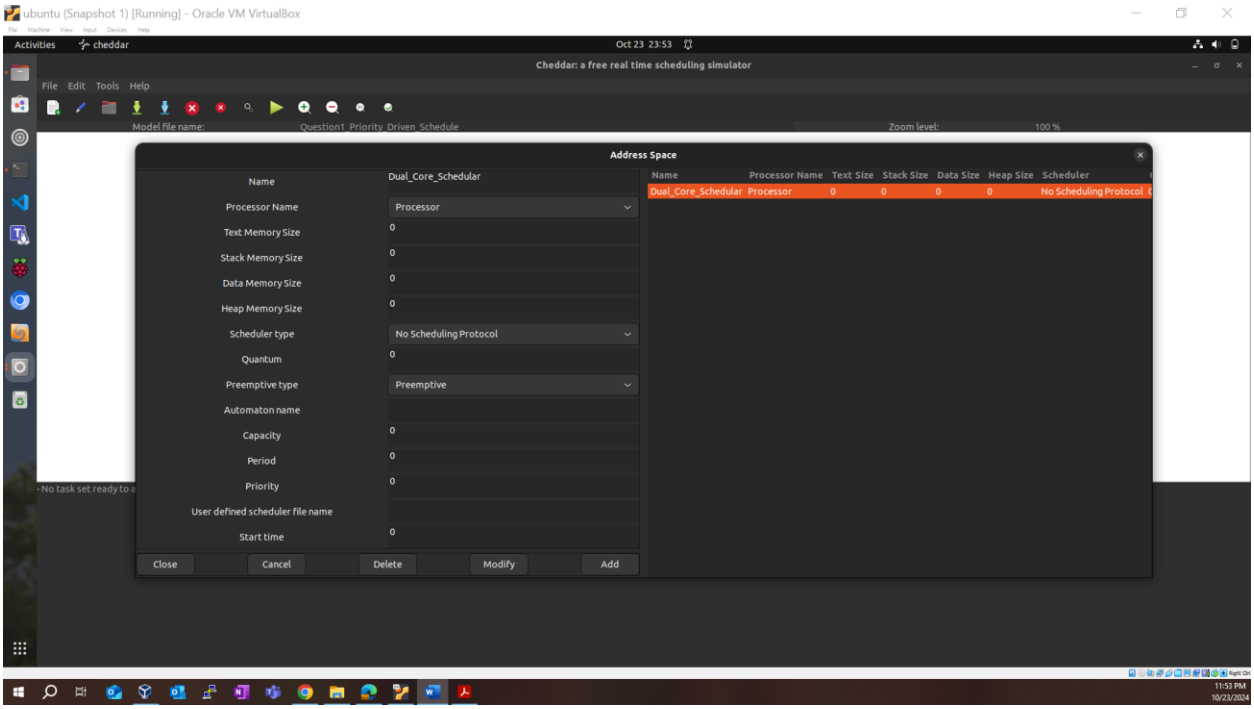


Added Processor

Processor Type : Identical Multicore Type
Cores Table : CORE1,CORE2

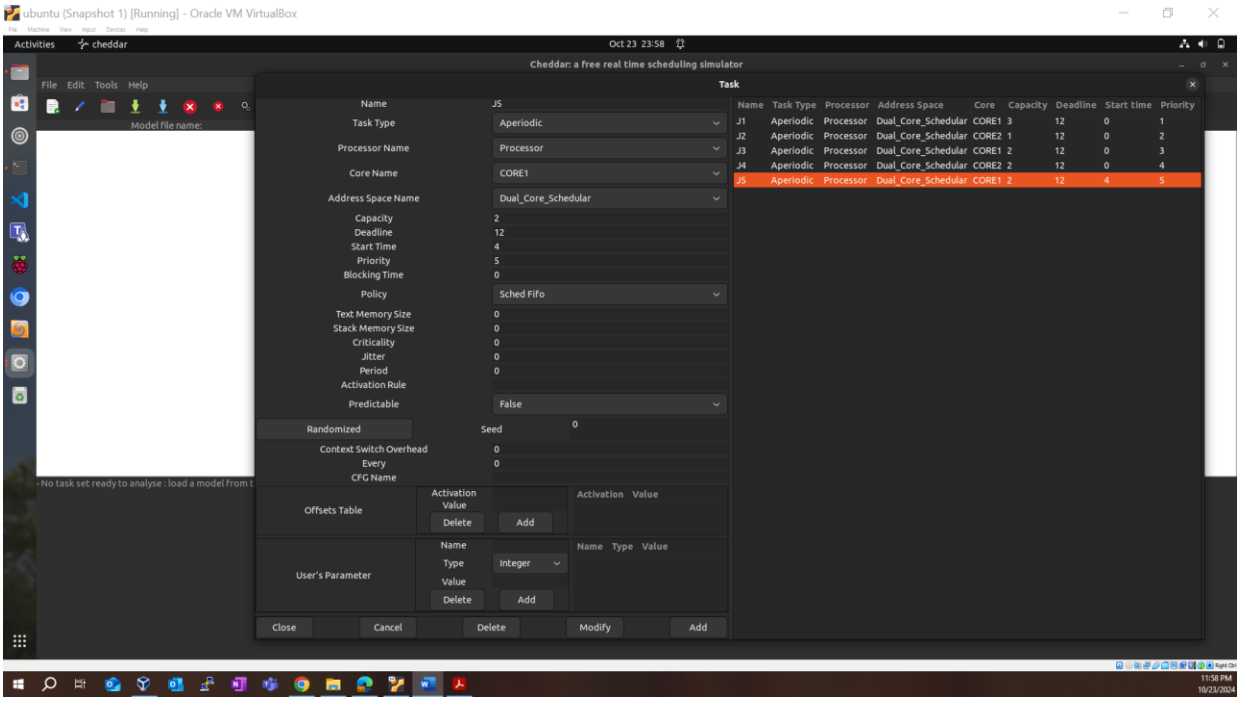


Added Address Space

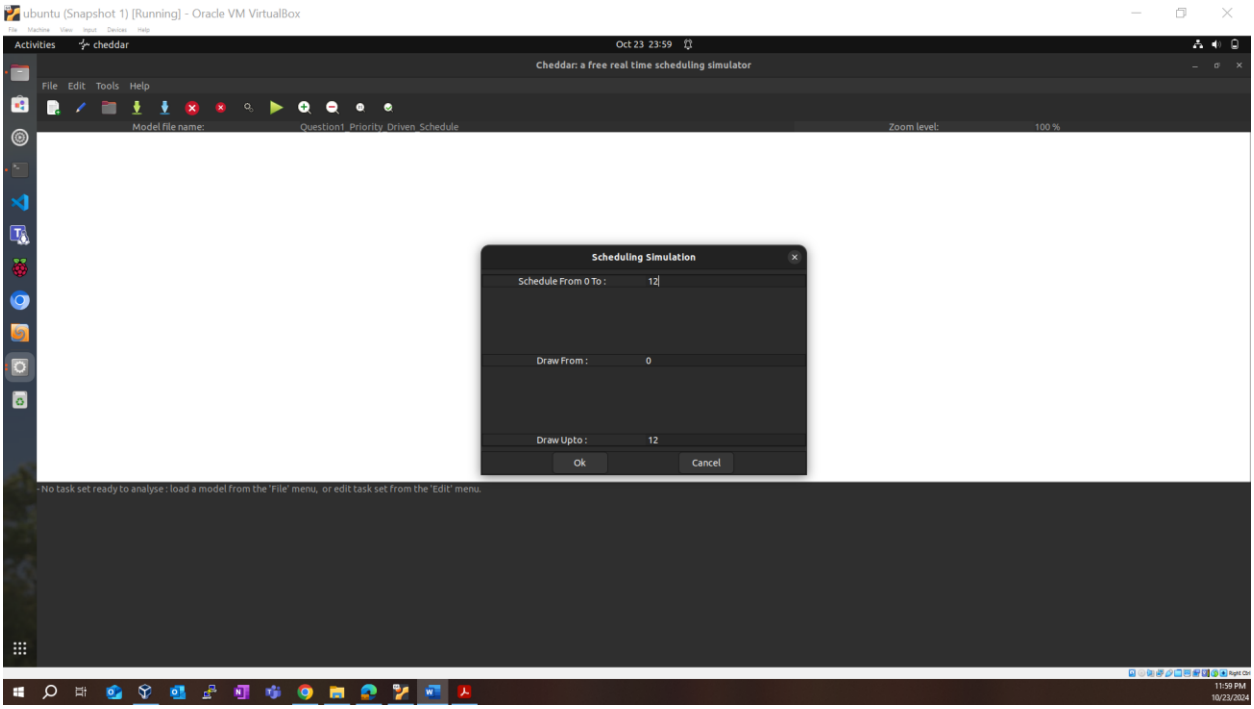


Add Task : Add J1,J2,J3,J4,J5

J1,J3,J5 is assigned to CORE1 of Processor & J2,J4 is assigned to CORE2 of Processor

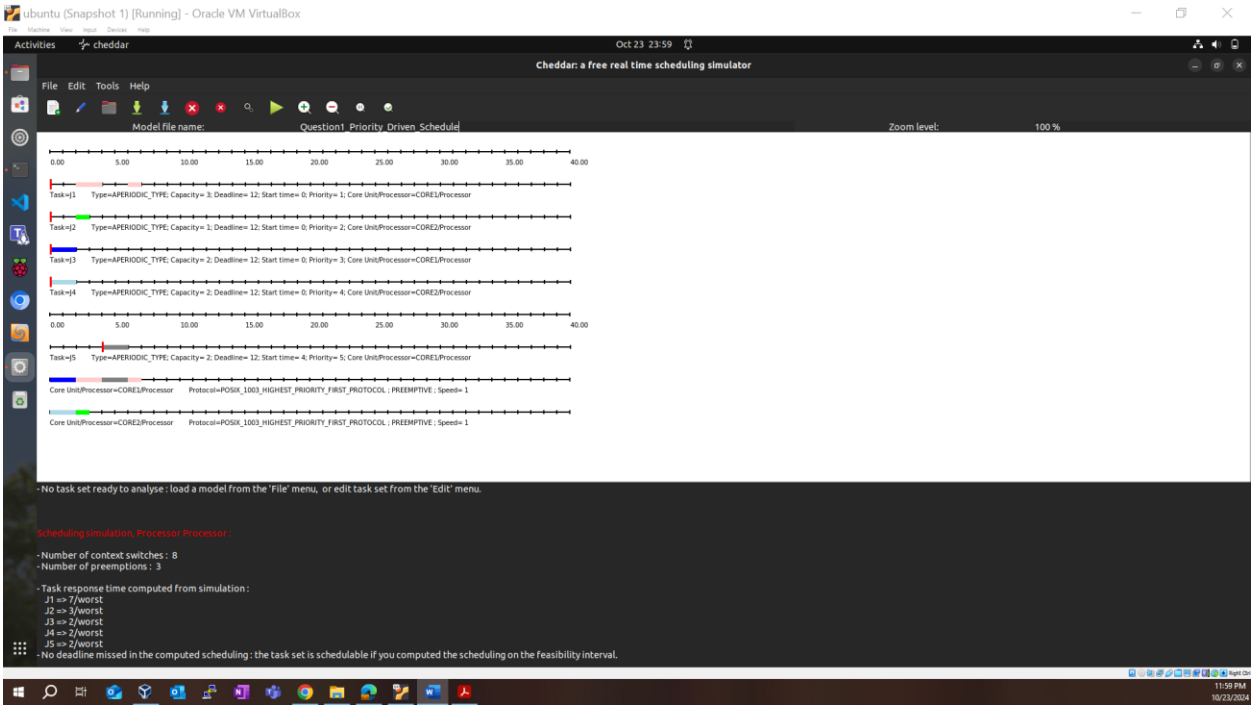


Run Simulation Schedule from 0 to 12ms =>



Preemptive Scheduling Results

- **Screenshot:**



- **Observations:**

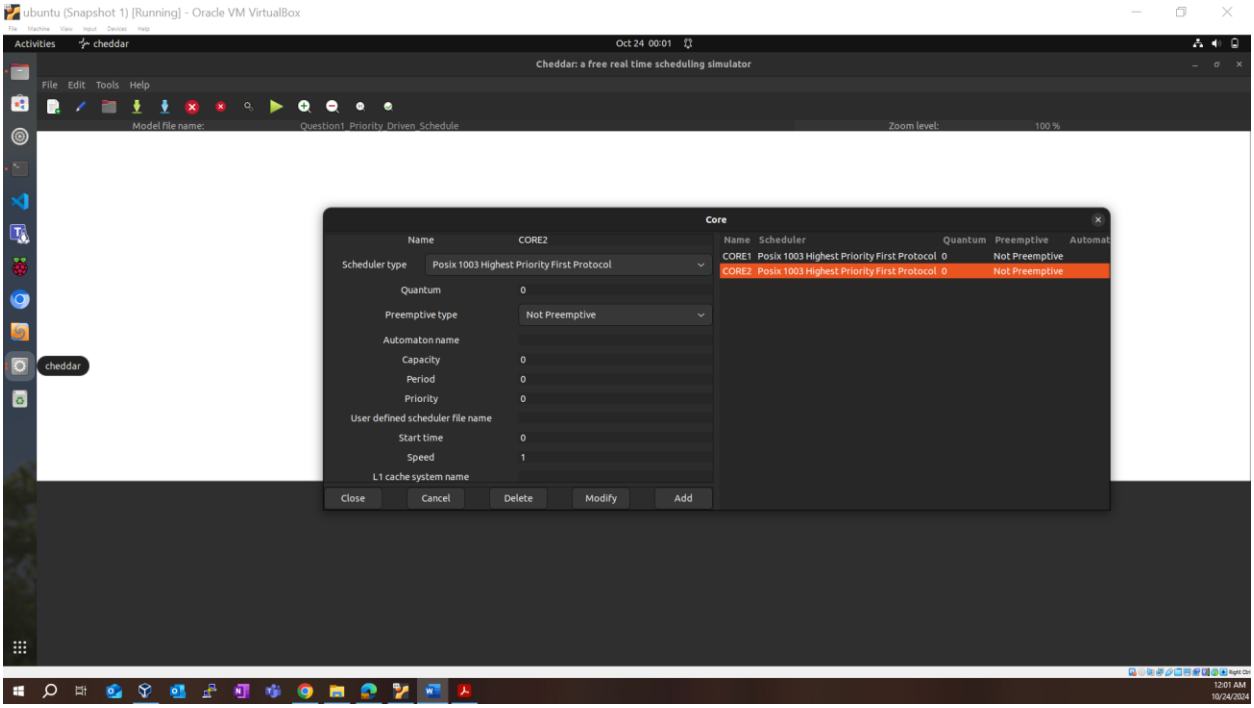
Frequent preemptions increased context switching overhead.

Some lower-priority tasks experienced delays due to interruptions by higher-priority tasks.

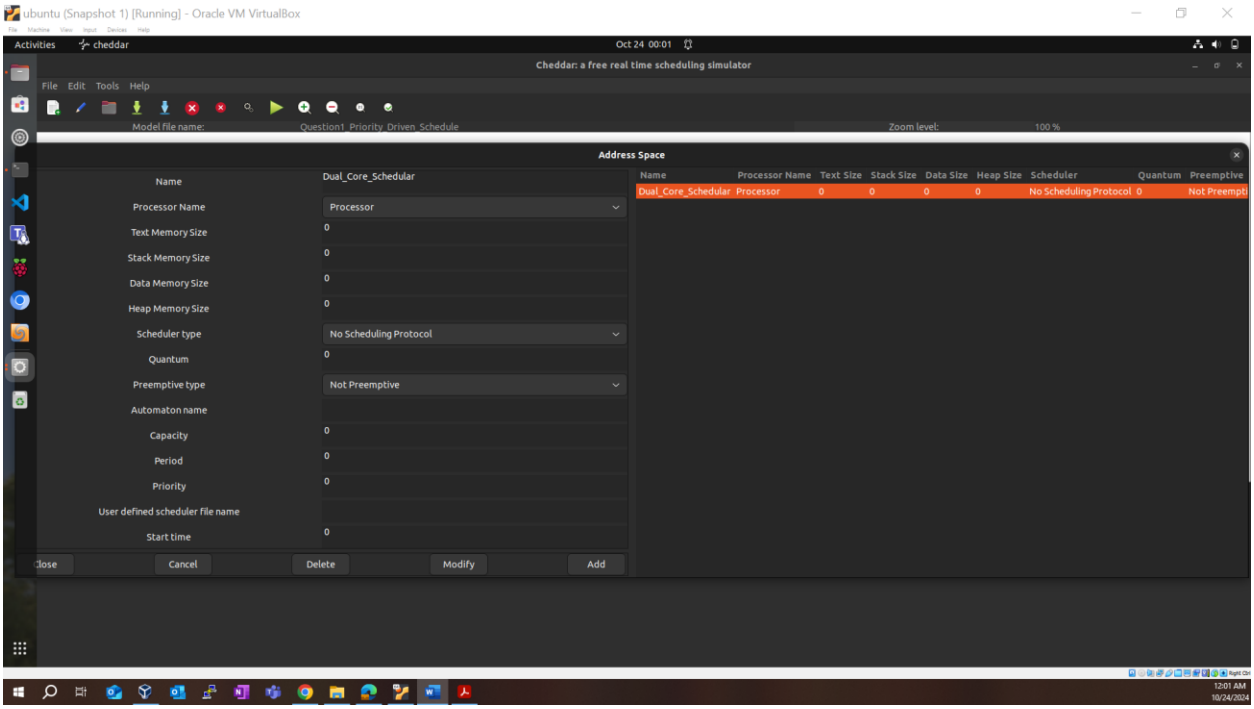
Task	Release Time	Execution Time	Deadline
J1	0	3	12
J2	0	1	12
J3	0	2	12
J4	0	2	12
J5	4	2	12

2. NON-PREEMPTIVE Priority Driven Scheduling :

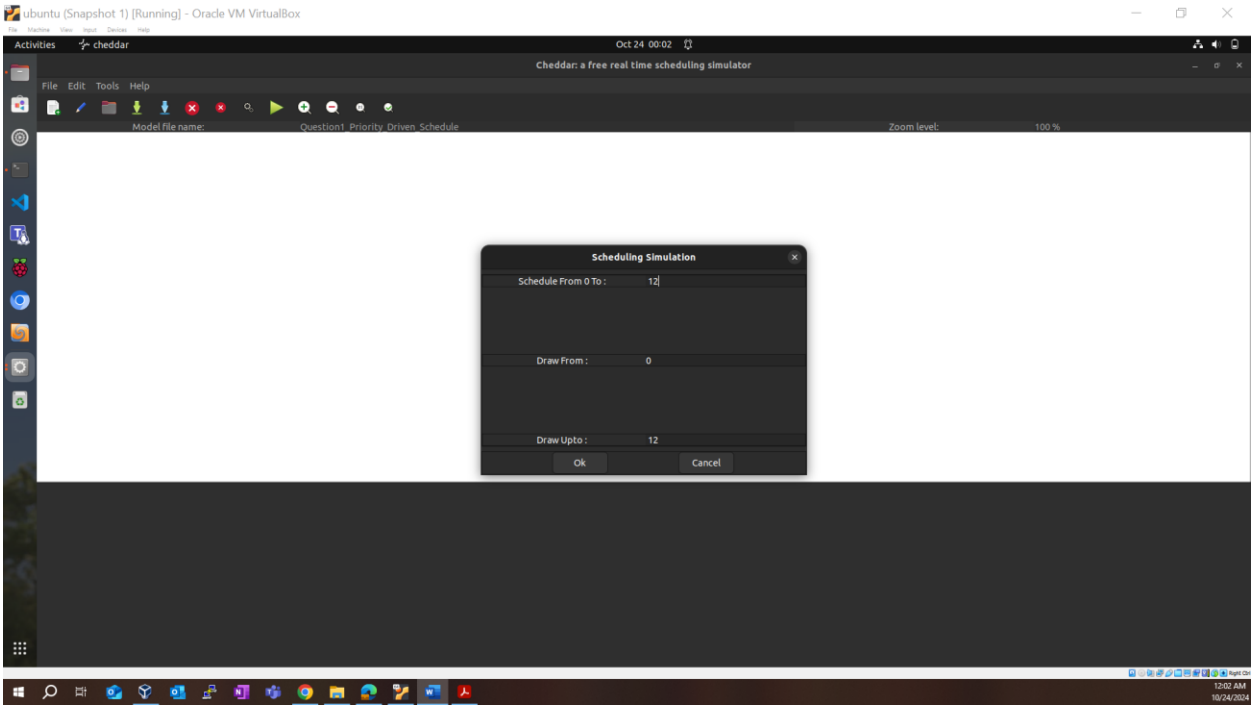
Modify CORE1 and CORE2 Preemption type to Not Preemptive



Modify the Address Space to Not Preemptive

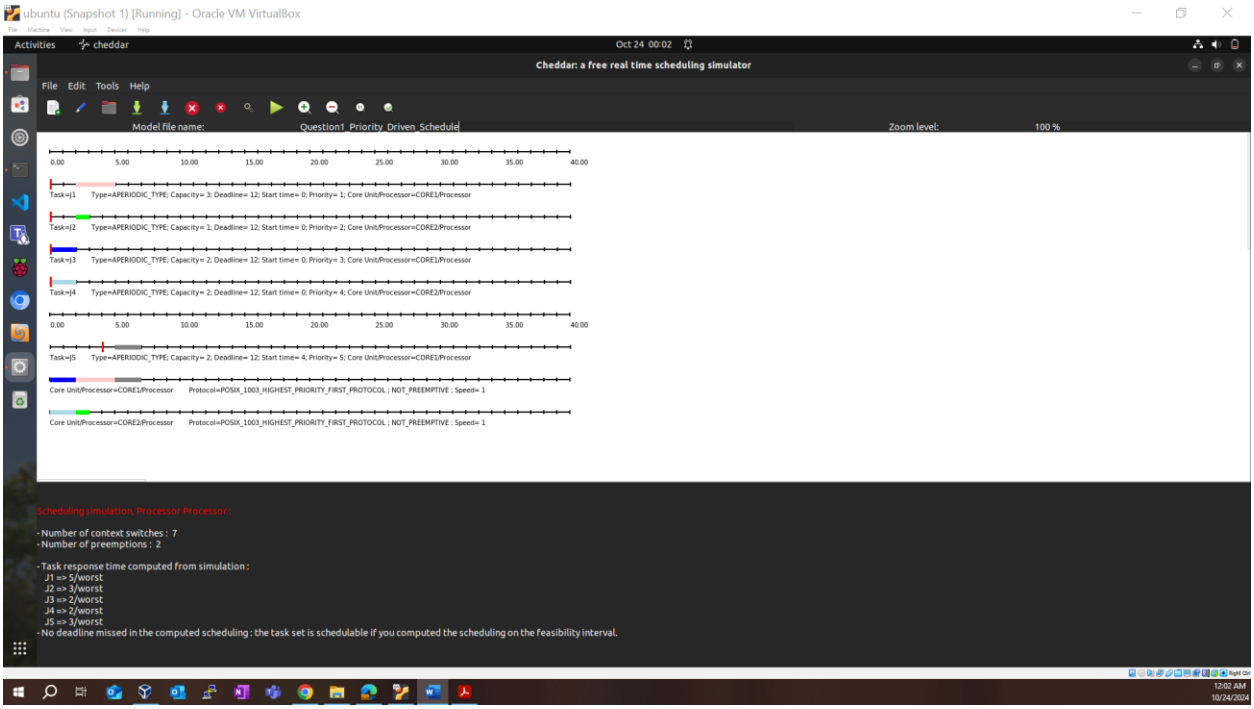


Again Run Simulation Schedule from 0 to 12ms



Non-Preemptive Scheduling Results

- **Screenshot**:



- **Observations**:

All tasks completed within their deadlines.

Minimal context switches due to the absence of preemption, reducing overhead.

Task	Release Time	Execution Time	Deadline
J1	0	3	12
J2	0	1	12
J3	0	2	12
J4	0	2	12
J5	4	2	12

Question 3 : Show that the following tasks table is schedulable using LLF but not using EDF on a dual-core processor. All tasks are aperiodic.

	J1	J2	J3
r	0	0	0
e	1	1	5
d	1	2	5

Solution : The objective of this experiment is to show that a task set is **schedulable using Least Laxity First (LLF) scheduling** but **not using Earliest Deadline First (EDF) scheduling** on a dual-core processor. This analysis will be conducted using Cheddar, a real-time scheduling analysis tool.

Task Set and Configuration

Task Characteristics

- **Number of Tasks**: 3 (labelled J1, J2, and J3)
- **Release Times**:

  J1, J2, and J3 are all released at time 0.
- **Execution Times (e)**:

  J1=1, J2=1, J3=5
- **Deadlines (d)**:

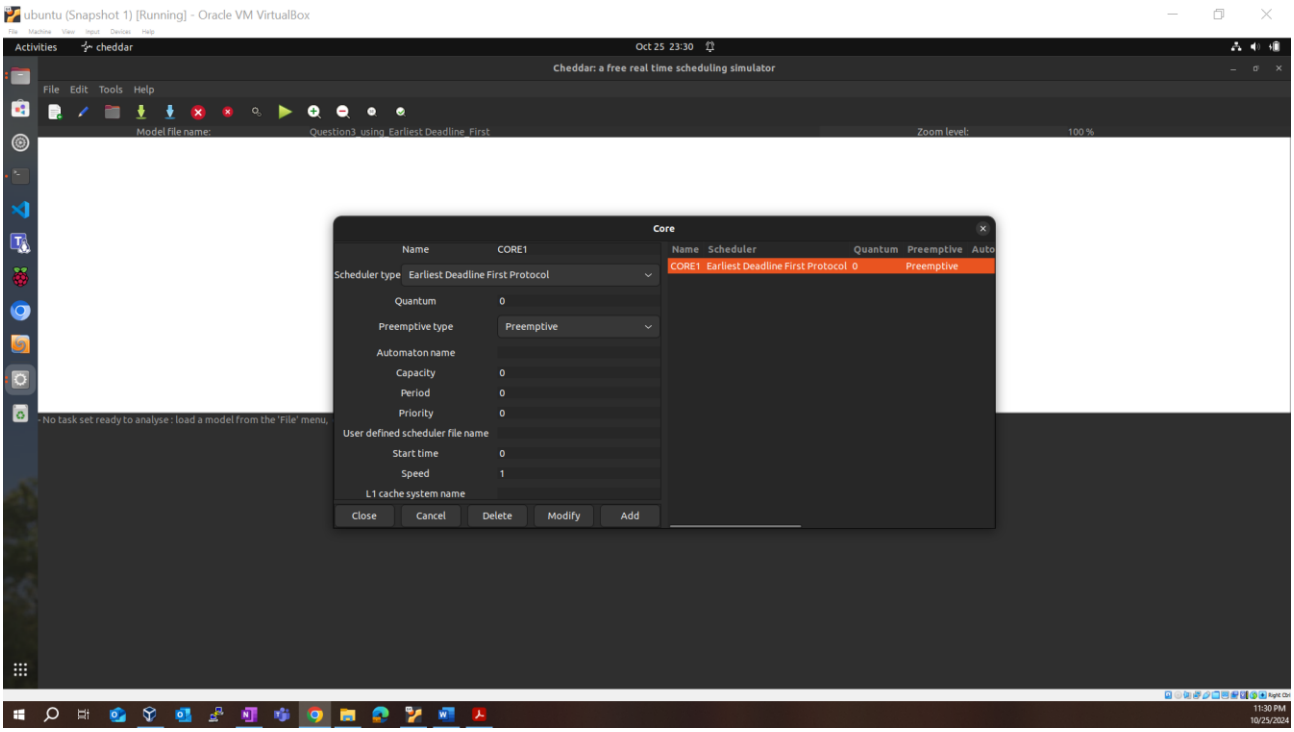
  J1=1, J2=2, J3=5
- **Processor Configuration**: Dual-core processor setup.

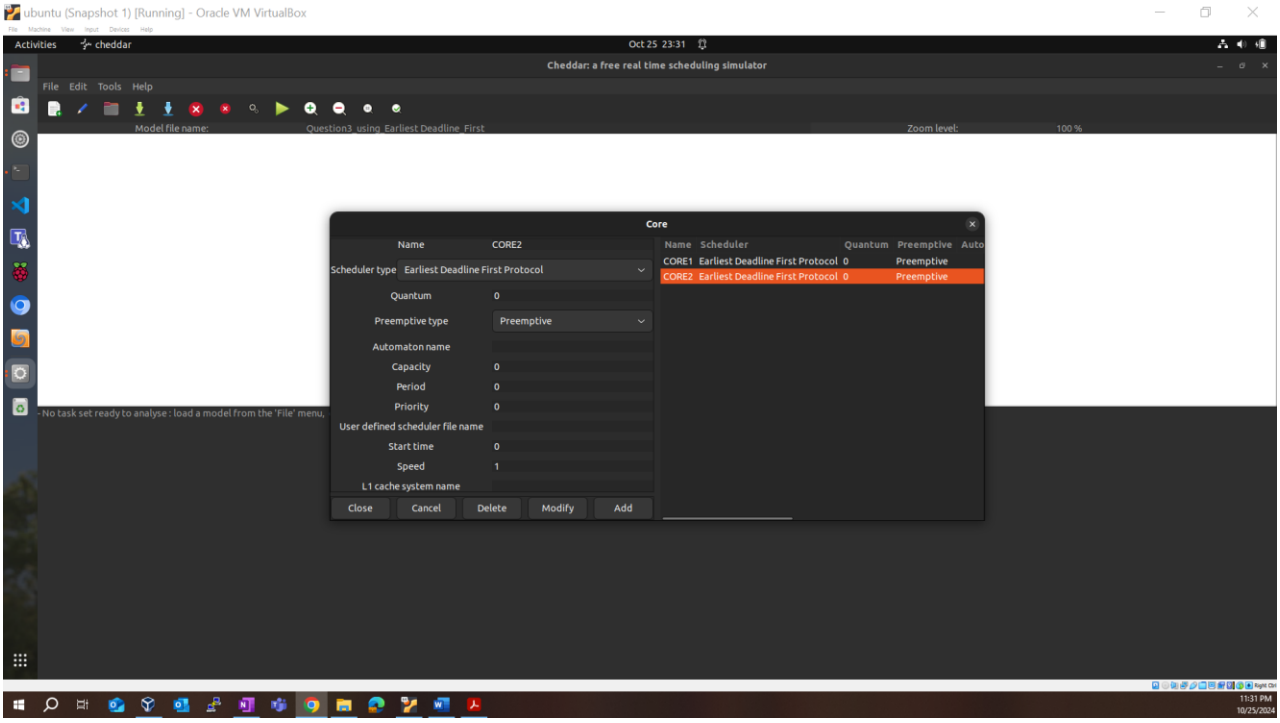
Scheduling Algorithms

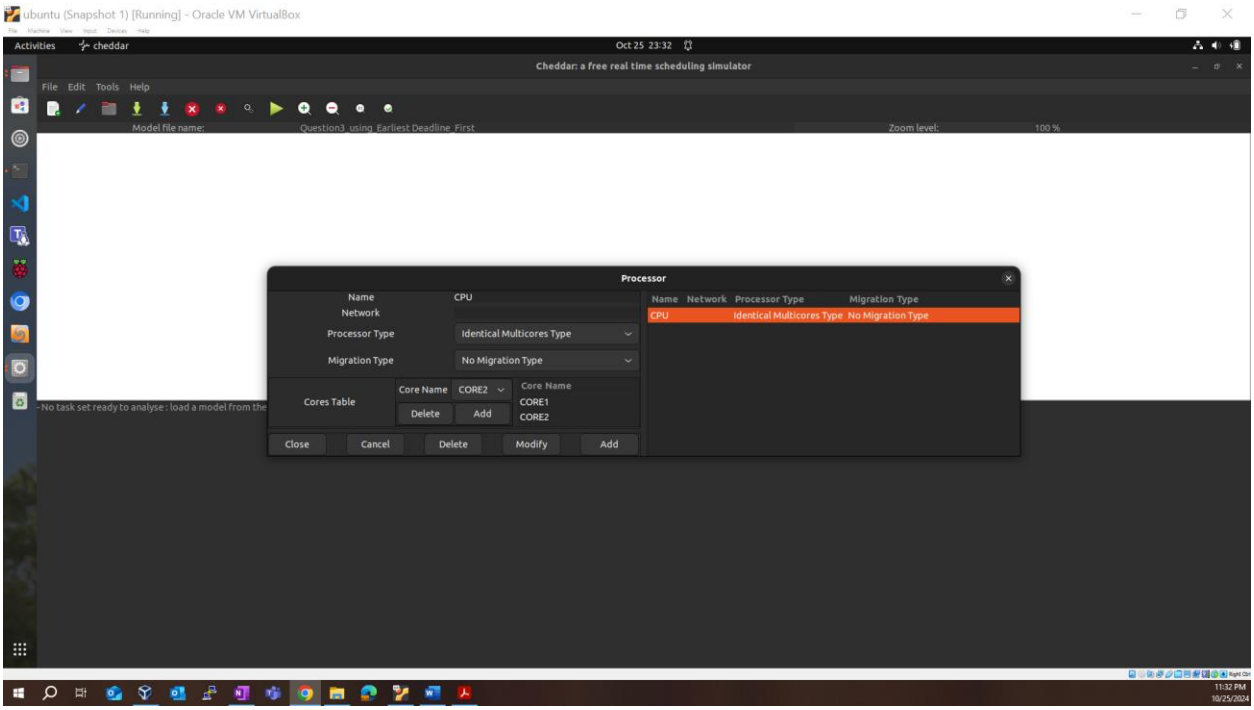
1. **Earliest Deadline First (EDF)**: Assigns dynamic priorities based on deadlines.
2. **Least Laxity First (LLF)**: Assigns priorities based on the smallest laxity (the difference between the time remaining until the deadline and the execution time remaining).

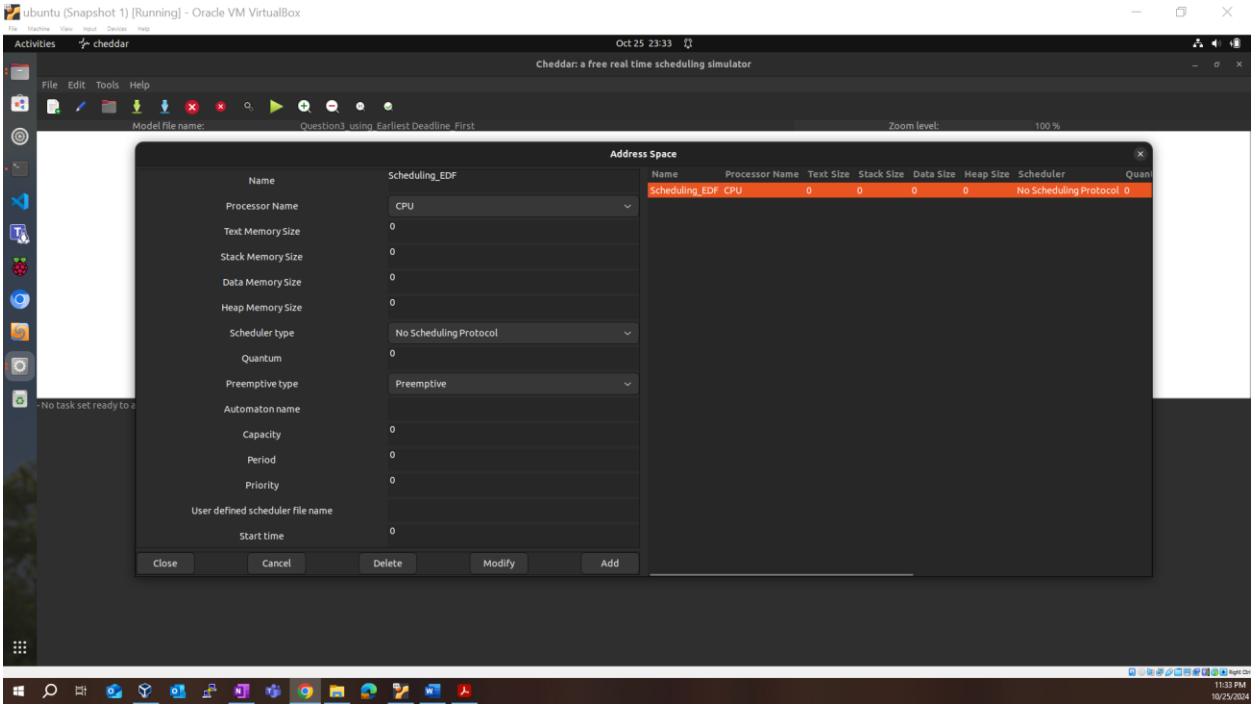
1. **Earliest Deadline First (EDF)**

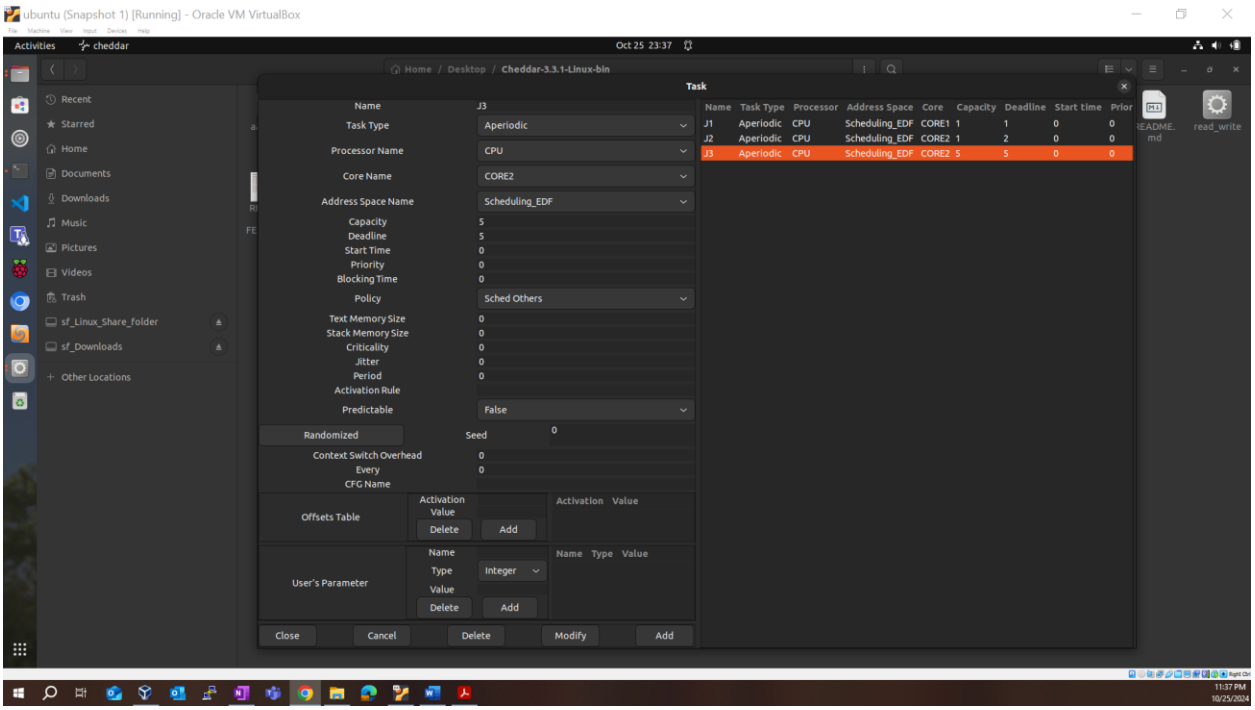
Using PREEMPTIVE :

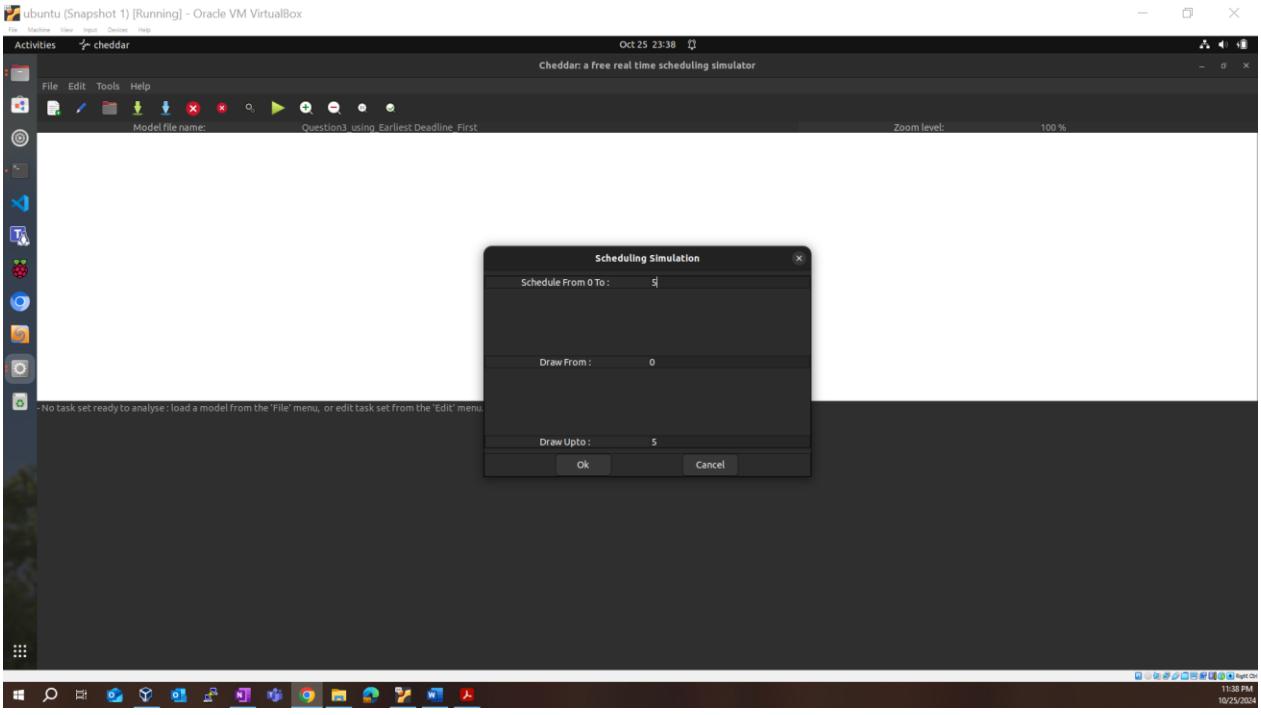





ubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
Activities
cheddar
Oct 25 23:32
Cheddar: a free real time scheduling simulator
File Edit Tools Help
Model file name:
Question3_using_Earliest Deadline_First
Zoom level:
100 %
Processor
Name
CPU
Network
Processor Type
Identical Multicores Type
Migration Type
No Migration Type
Cores Table
Core Name
CORE2
Delete
Add
Core Name
CORE1
CORE2
Close
Cancel
Delete
Modify
Add
Name Network Processor Type Migration Type
CPU Identical Multicores Type No Migration Type
-No task set ready to analyse : load a model from the
11:32 PM
10/25/2024

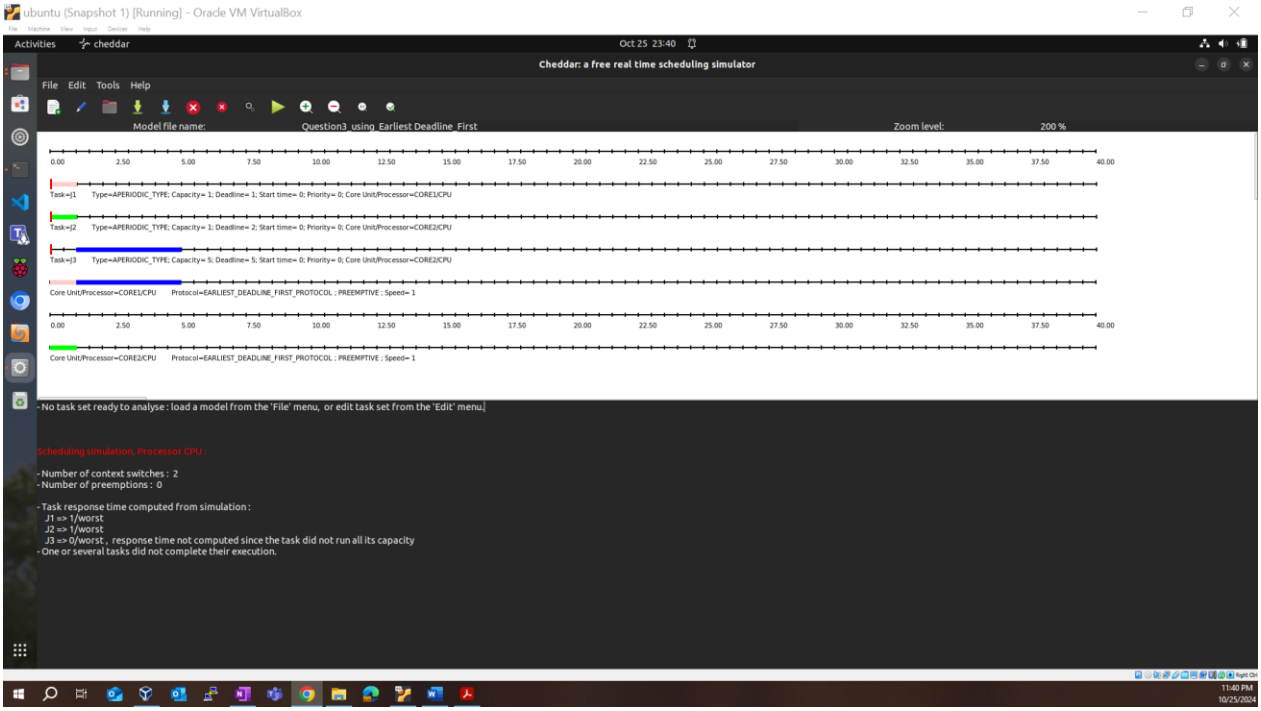

ubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
Activities
cheddar
Oct 25 23:33
Cheddar: a free real time scheduling simulator
File Edit Tools Help
Model file name:
Question3_using_Earliest Deadline_First
Zoom level:
100 %
Address Space
Name
Scheduling_EDF
Processor Name
CPU
Text Memory Size
Stack Memory Size
Data Memory Size
Heap Memory Size
Scheduler type
No Scheduling Protocol
Quantum
Preemptive type
Preemptive
Automaton name
Capacity
Period
Priority
User defined scheduler file name
Start time
Close
Cancel
Delete
Modify
Add
Name Processor Name Text Size Stack Size Data Size Heap Size Scheduler
Scheduling_EDF CPU 0 0 0 0 No Scheduling Protocol 0
-No task set ready to a
11:33 PM
10/25/2024


ubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
Activities
cheddar
Oct 25 23:37
Home / Desktop / Cheddar-3.3.1-Linux-bin
Task
Name J3
Task Type Aperiodic
Processor Name CPU
Core Name CORE2
Address Space Name Scheduling_EDF
Capacity 5
Deadline 5
Start Time 0
Priority 0
Blocking Time 0
Policy Sched Others
Text Memory Size 0
Stack Memory Size 0
Criticality 0
Jitter 0
Period 0
Activation Rule
Predictable False
Randomized
Seed 0
Context Switch Overhead 0
Every 0
CFG Name
Offsets Table
Activation Value
Delete
Add
Activation Value
User's Parameter
Name
Type Integer
Value
Delete
Add
Name Type Value
Close
Cancel
Delete
Modify
Add
Name Task Type Processor Address Space Core Capacity Deadline Start time Prior
J1 Aperiodic CPU Scheduling_EDF CORE1 1 1 0 0
J2 Aperiodic CPU Scheduling_EDF CORE2 1 2 0 0
J3 Aperiodic CPU Scheduling_EDF CORE2 5 5 0 0
README.md
read_write
Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
Trash
sf_Linux_Share_Folder
sf_Downloads
Other Locations
11:37 PM
10/25/2024

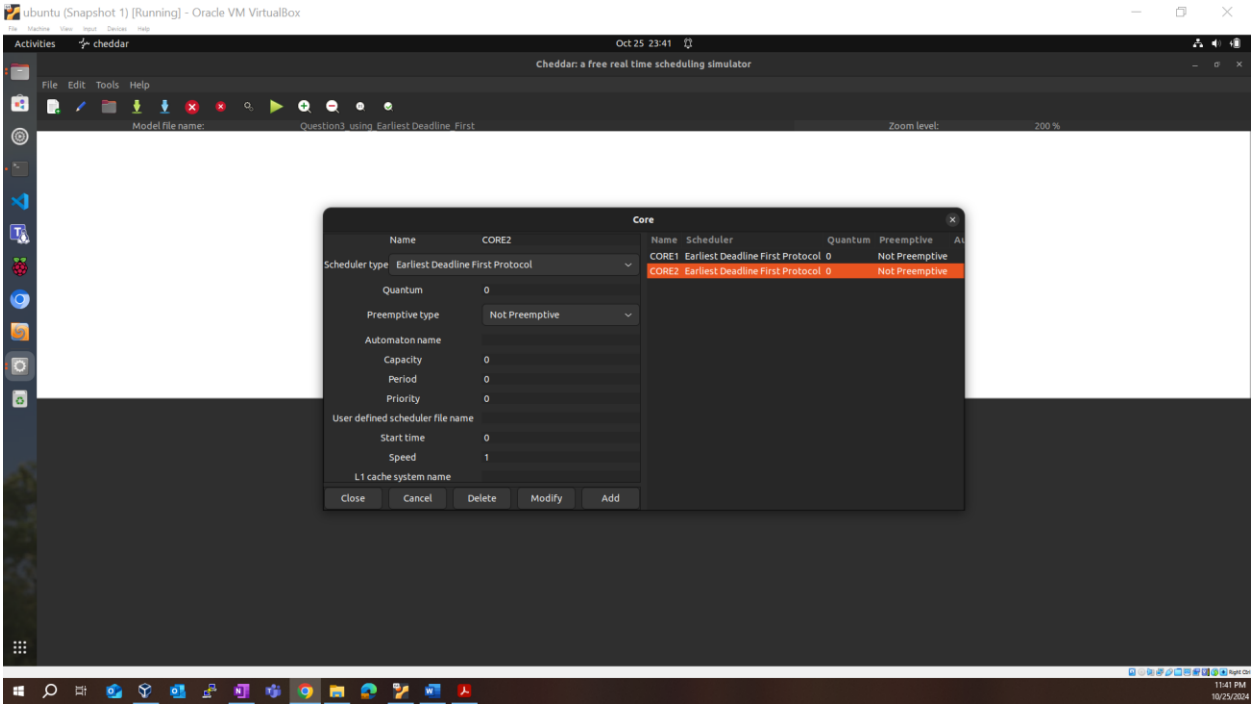

ubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
Activities
cheddar
Oct 25 23:38
Cheddar: a free real time scheduling simulator
File Edit Tools Help
Model file name: Question3_using_Earliest Deadline_First
Zoom level: 100 %
Scheduling Simulation
Schedule From 0 To : 5
Draw From : 0
Draw Upto : 5
Ok
Cancel
- No task set ready to analyse : load a model from the 'File' menu, or edit task set from the 'Edit' menu.
11:38 PM
10/25/2024

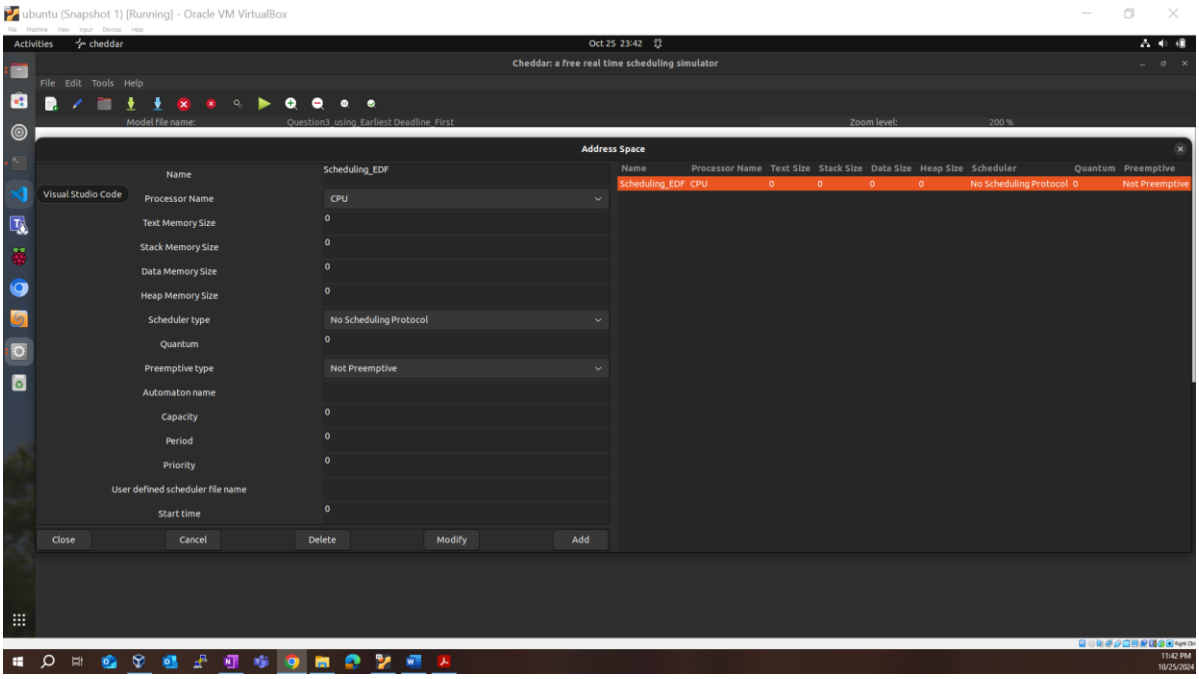
Earliest Deadline First (EDF) Scheduling (PREEMPTIVE) Results

- **Screenshot:**



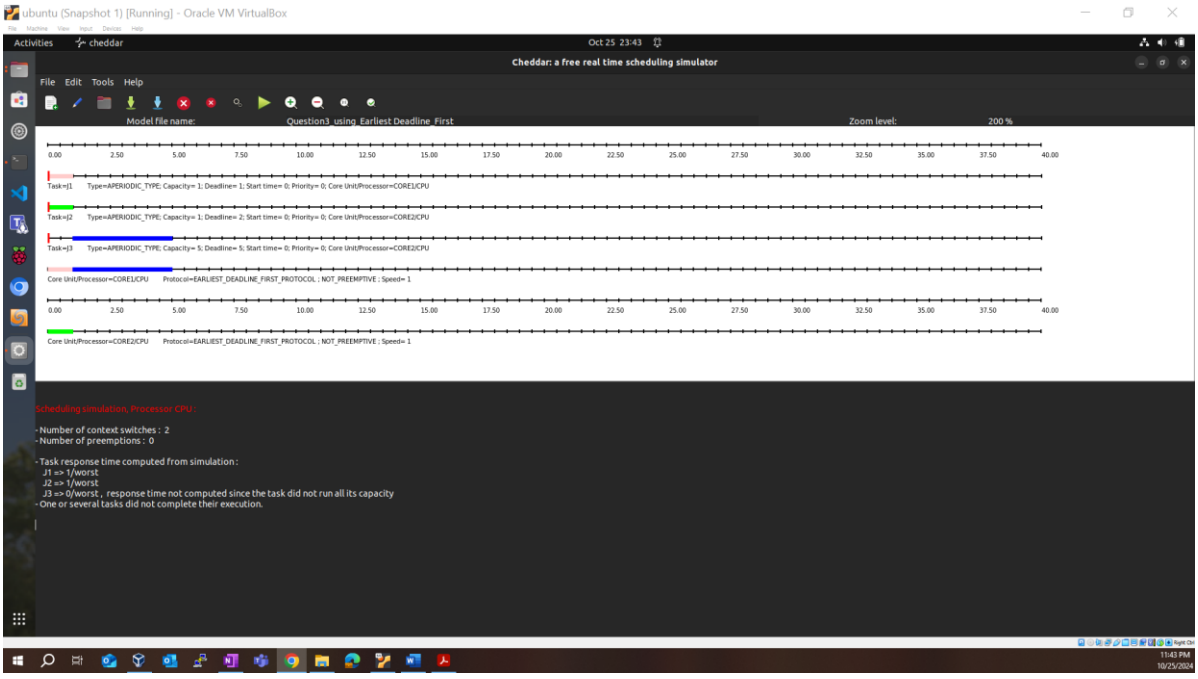
Using NOT PREEMPTIVE EDF :





Earliest Deadline First (EDF) Scheduling Not Preemptive Results

- **Screenshot:**

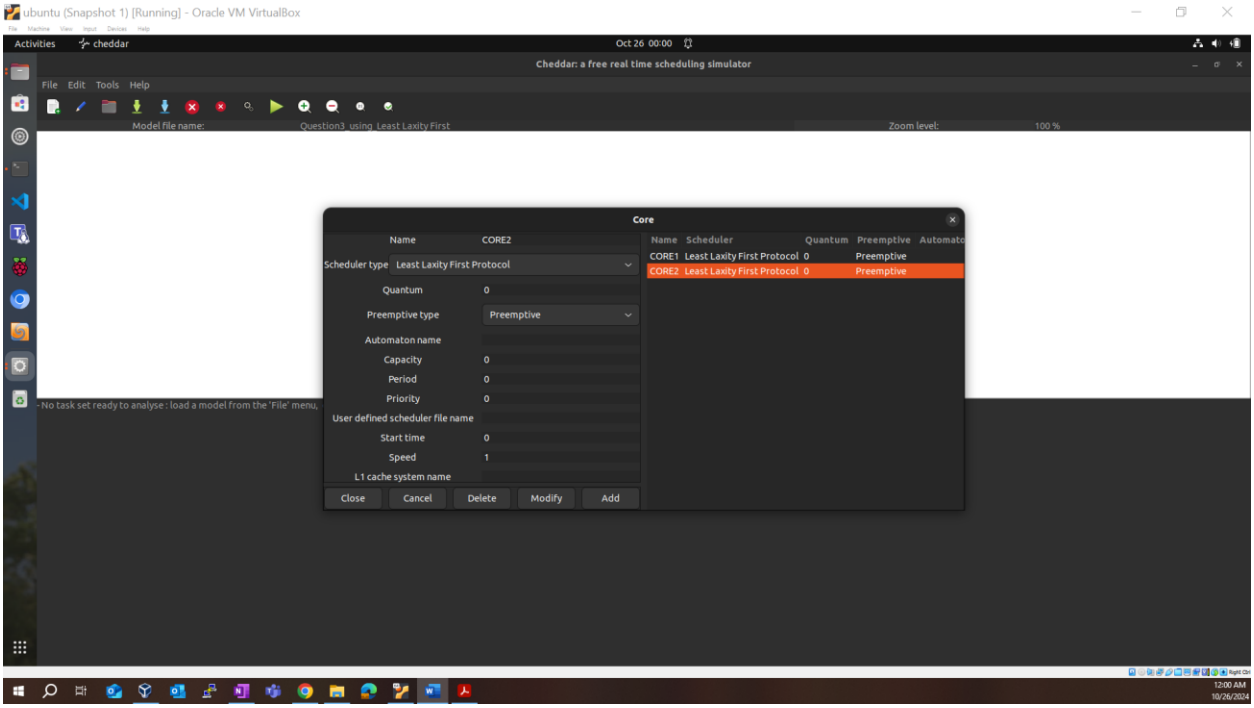


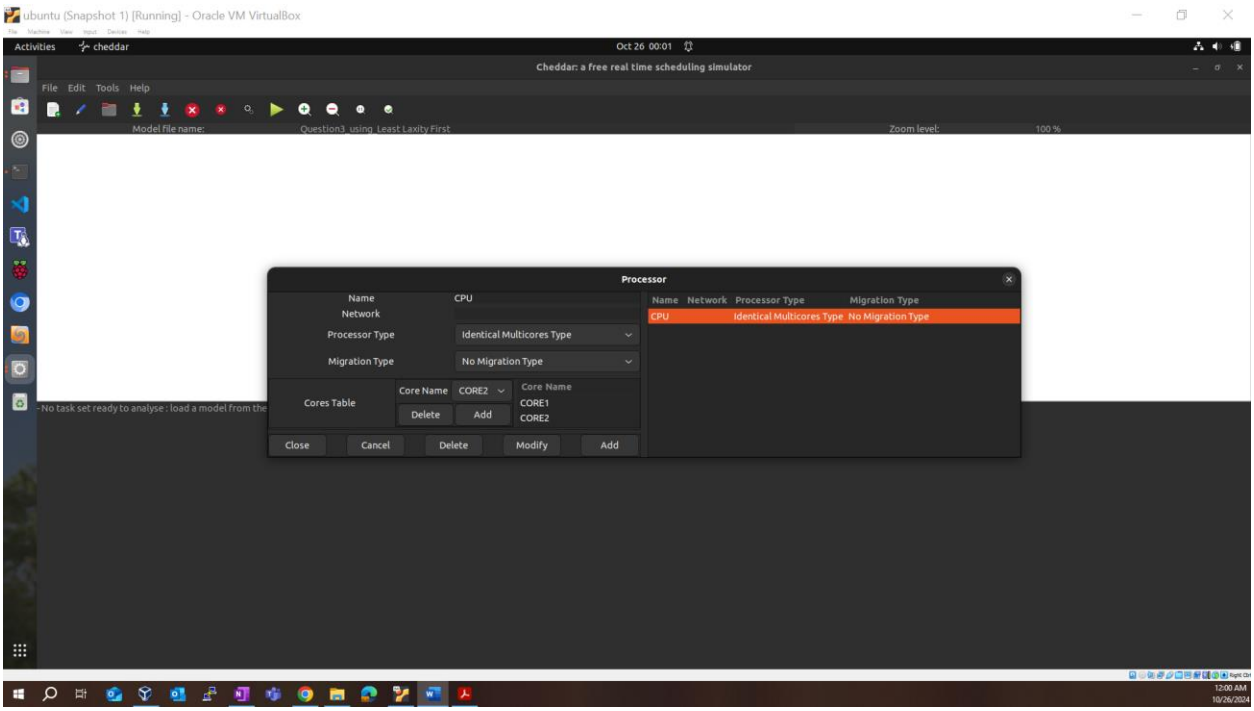
Observations:

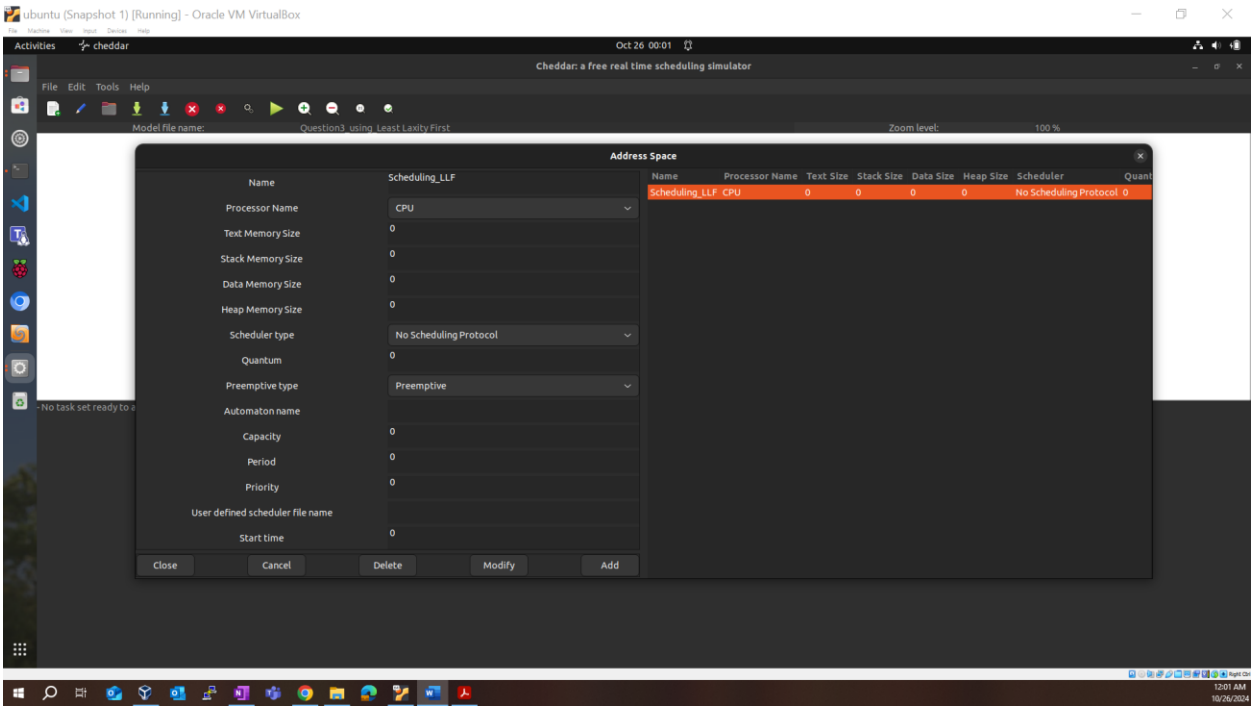
Not all tasks complete within their deadlines under EDF scheduling, particularly due to EDF's prioritization of tasks solely based on deadlines, which doesn't account for laxity.

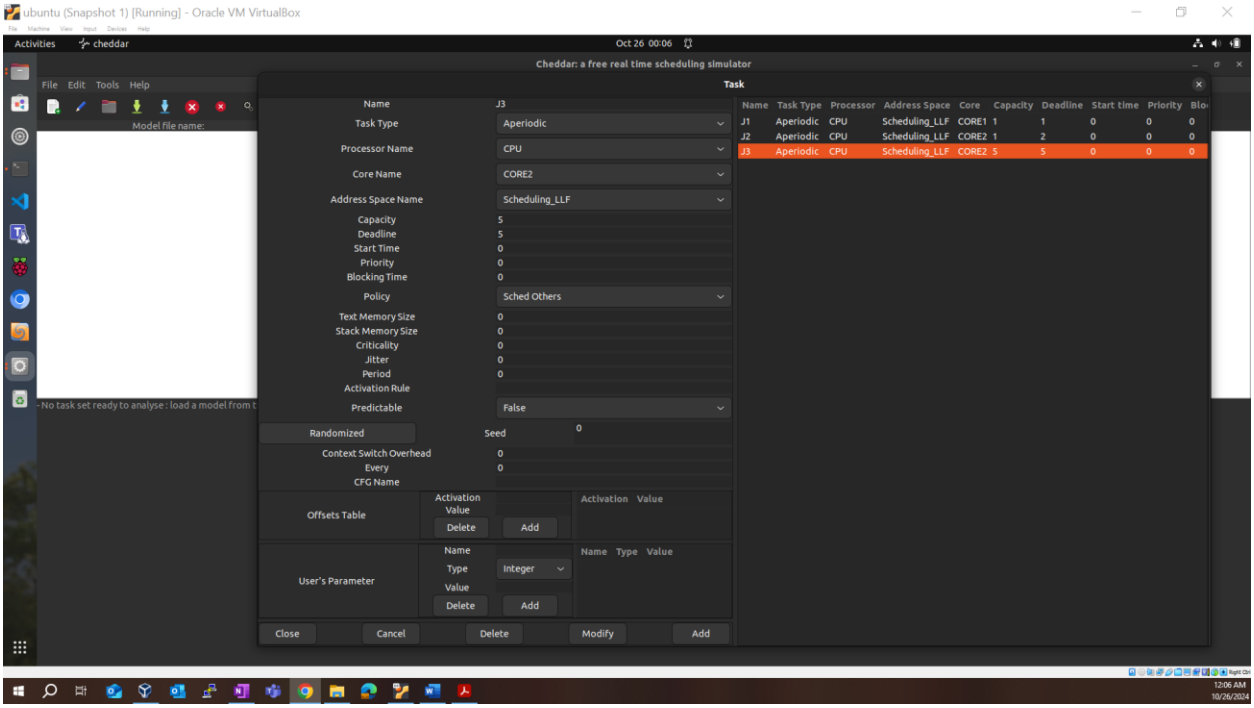
2. Least Laxity First (LLF)

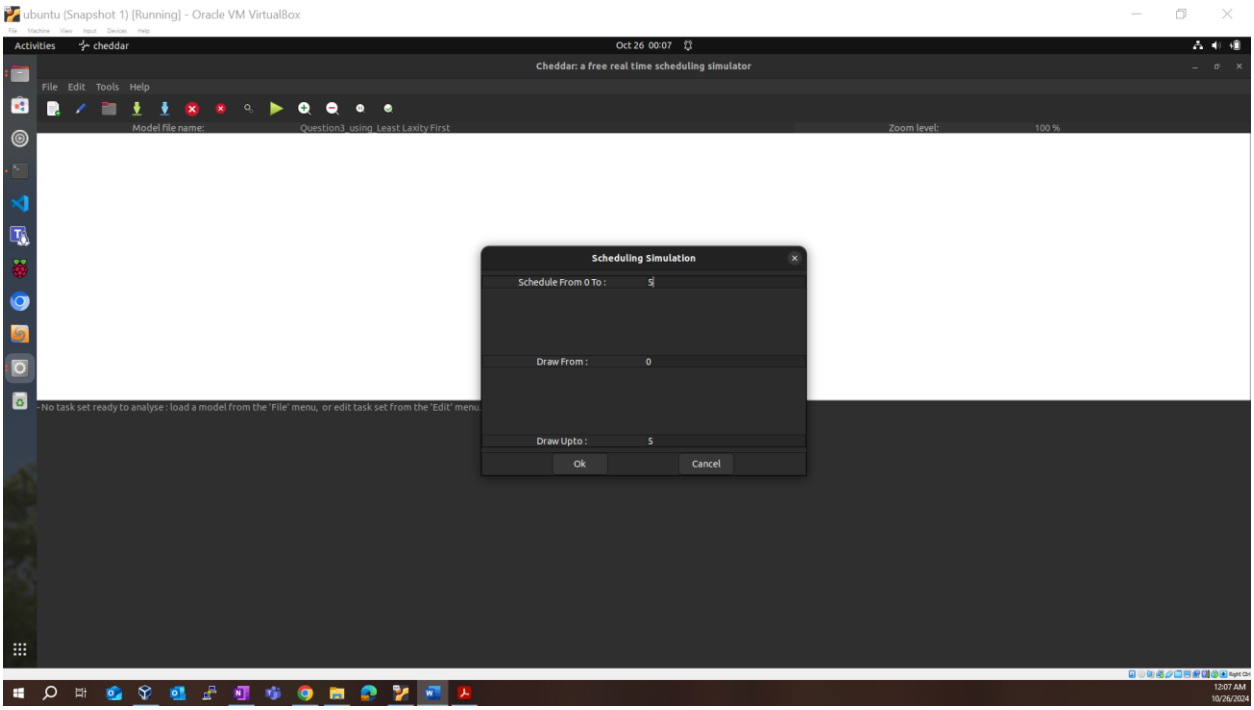
Using PREEMPTIVE :





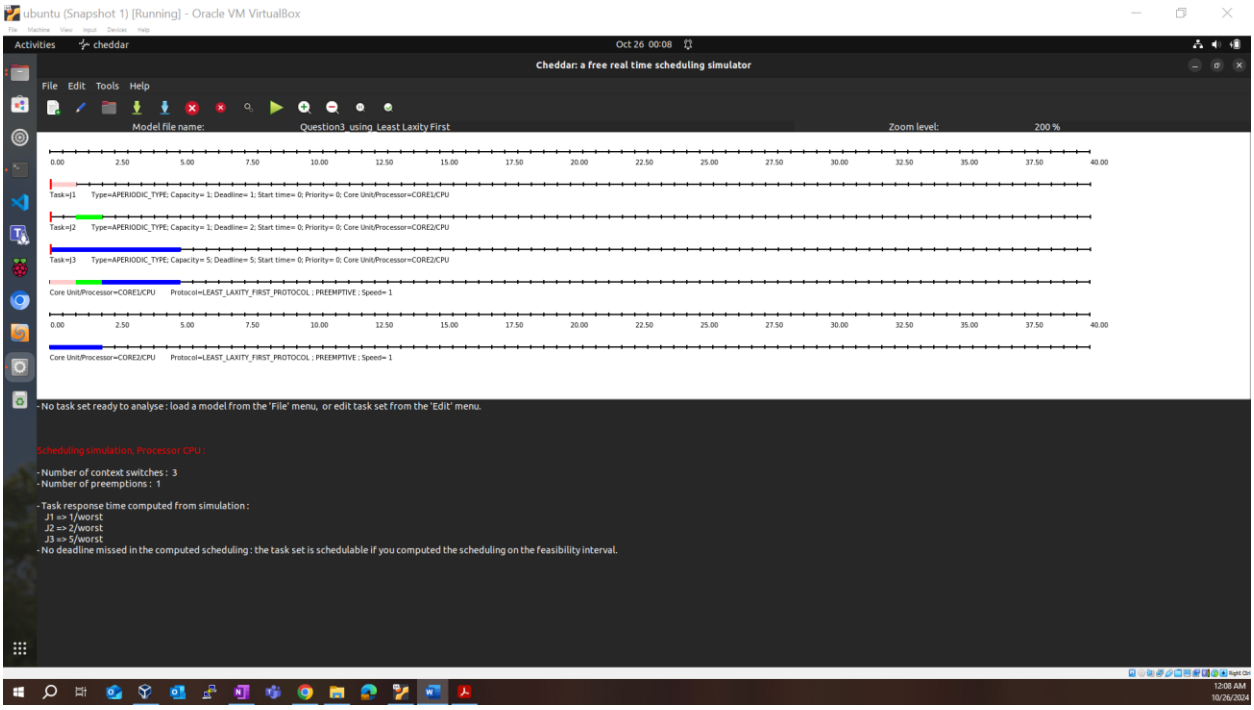

ubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
Activities
cheddar
Oct 26 00:01
Cheddar: a free real time scheduling simulator
File Edit Tools Help
Model file name:
Question3_using_Least Laxity First
Zoom level:
100 %
Address Space
Name
Scheduling_LLF
Processor Name
CPU
Text Memory Size
Stack Memory Size
Data Memory Size
Heap Memory Size
Scheduler type
No Scheduling Protocol
Quantum
Preemptive type
Preemptive
Automaton name
Capacity
Period
Priority
User defined scheduler file name
Start time
Close
Cancel
Delete
Modify
Add
Name Processor Name Text Size Stack Size Data Size Heap Size Scheduler Quant
Scheduling_LLF CPU 0 0 0 0 No Scheduling Protocol 0
No task set ready to a
12:01 AM
10/26/2024


ubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
Activities
cheddar
Oct 26 00:06
Cheddar: a free real time scheduling simulator
File Edit Tools Help
Model file name:
Task
Name
J3
Task Type
Aperiodic
Processor Name
CPU
Core Name
CORE2
Address Space Name
Scheduling_LLF
Capacity
5
Deadline
5
Start Time
0
Priority
0
Blocking Time
0
Policy
Sched Others
Text Memory Size
0
Stack Memory Size
0
Criticality
0
Jitter
0
Period
0
Activation Rule
Predictable
False
Randomized
Seed
0
Context Switch Overhead
0
Every
0
CFG Name
Offsets Table
Activation Value
Delete
Add
Activation Value
User's Parameter
Name
Type
Integer
Value
Delete
Add
Name Type Value
Close
Cancel
Delete
Modify
Add
Name Task Type Processor Address Space Core Capacity Deadline Start time Priority Blo
J1 Aperiodic CPU Scheduling_LLF CORE1 1 1 0 0 0
J2 Aperiodic CPU Scheduling_LLF CORE2 1 2 0 0 0
J3 Aperiodic CPU Scheduling_LLF CORE2 5 5 0 0 0
No task set ready to analyse : load a model from t
12:06 AM
10/26/2024



Least Laxity First (LLF) Scheduling Preemptive Results

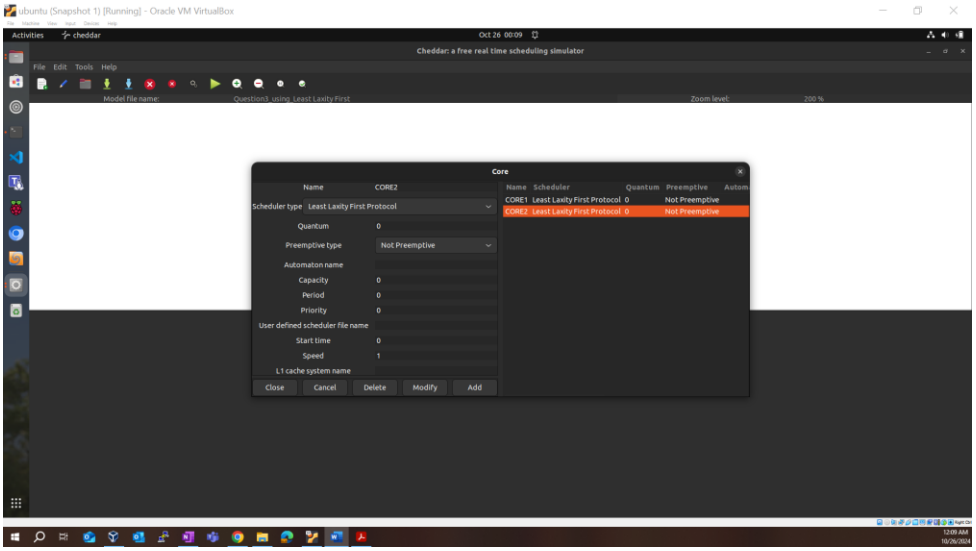
- **Screenshot:**

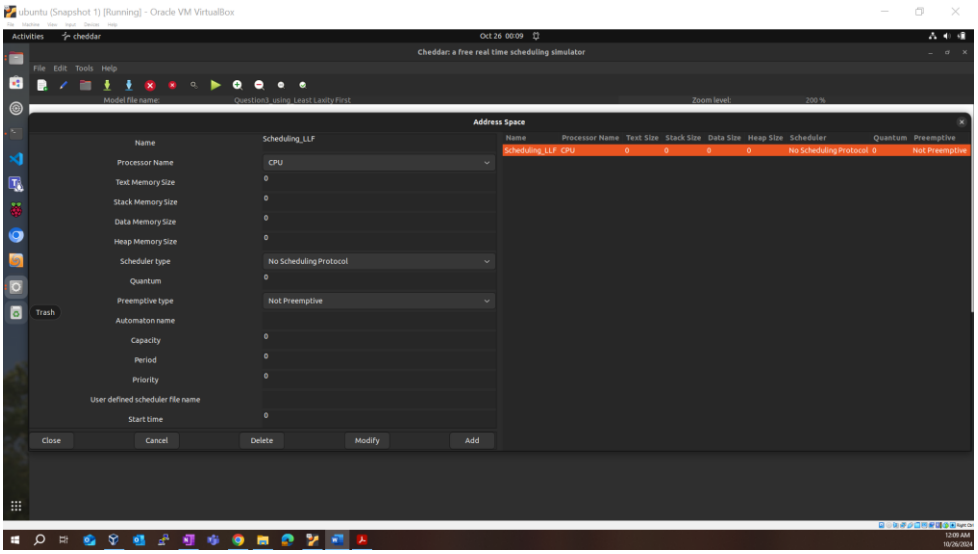


- **Summary Table:**

Task	Release Time	Completion Time	Deadline	Missed Deadline
J1	0	1	1	No
J2	0	2	2	No
J3	0	5	5	No

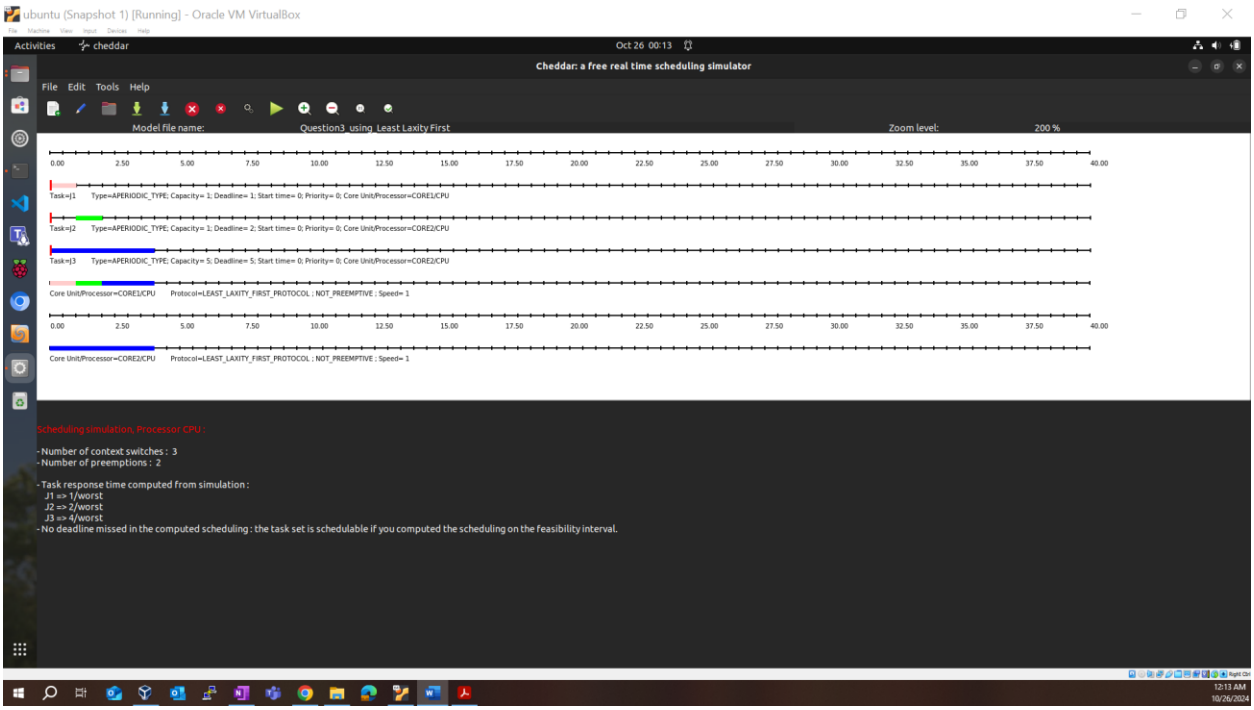
Using LLF NOT PREEMPTIVE





Least Laxity First (LLF) Scheduling Not Preemptive Results

- **Screenshot:**



- **Observations:**

LLF scheduling meets all task deadlines. This is due to LLF's ability to adjust priorities dynamically based on the tasks' laxities, prioritizing tasks with the least remaining time until their deadlines.

Analysis of Question 3 :

- **EDF Limitations**: Under EDF, tasks with closer deadlines are prioritized regardless of their laxity, which can lead to missed deadlines if tasks with significant execution times are scheduled closer to their deadlines.
- **LLF Benefits**: LLF's dynamic prioritization based on laxity results in better deadline adherence by prioritizing tasks that have the least slack time, thus improving task completion reliability.
- **Conclusion**: LLF scheduling is preferable for this task set as it ensures all tasks complete within their deadlines on the dual-core processor, whereas EDF leads to missed deadlines.

Question 4 : Show that the following tasks table causes indeterminism due to the execution time when scheduled using EDF. All tasks are periodic.

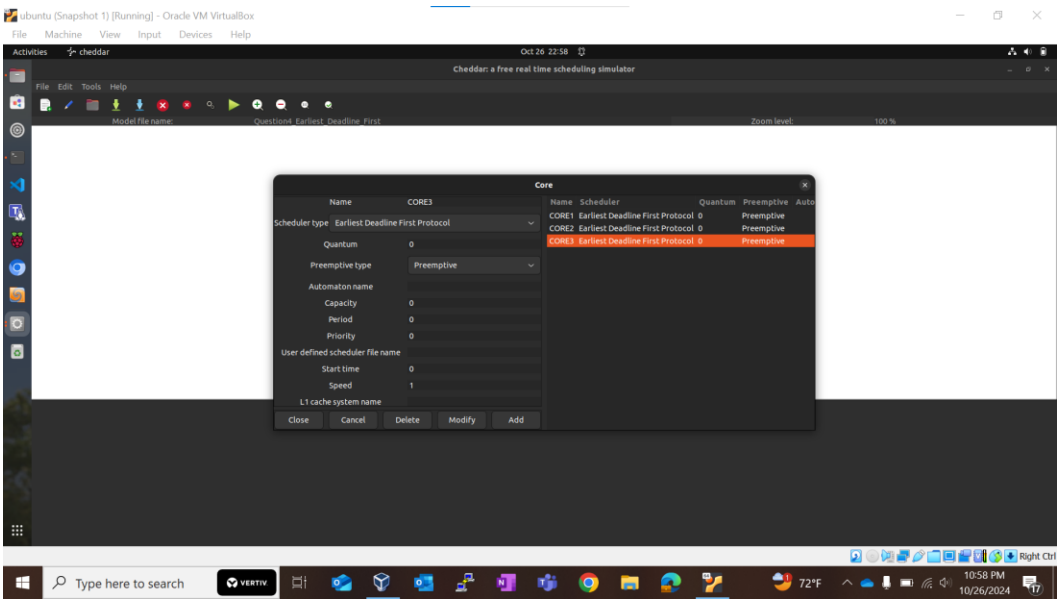
Task	R	d	e
J1	0	10	5
J2	0	10	2-6
J3	4	15	8
J4	0	20	10

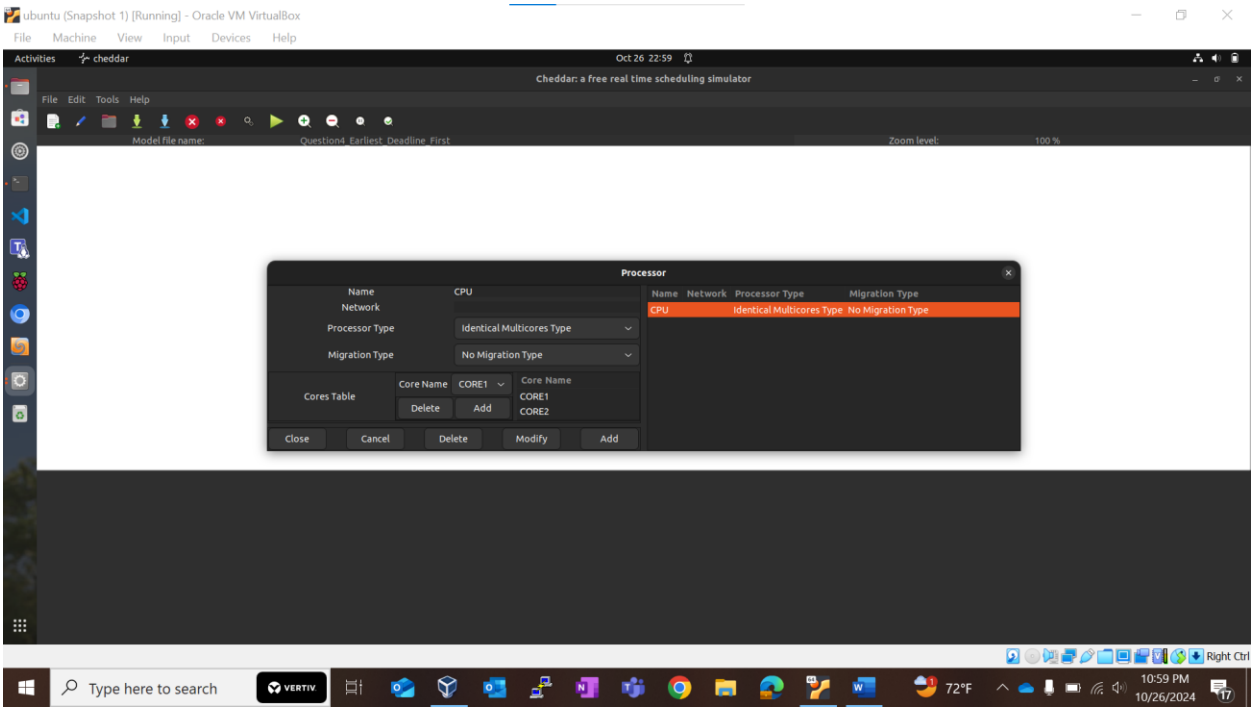
Solution : To demonstrate the impact of variable execution times on EDF scheduling and the resulting indeterminism in task scheduling.

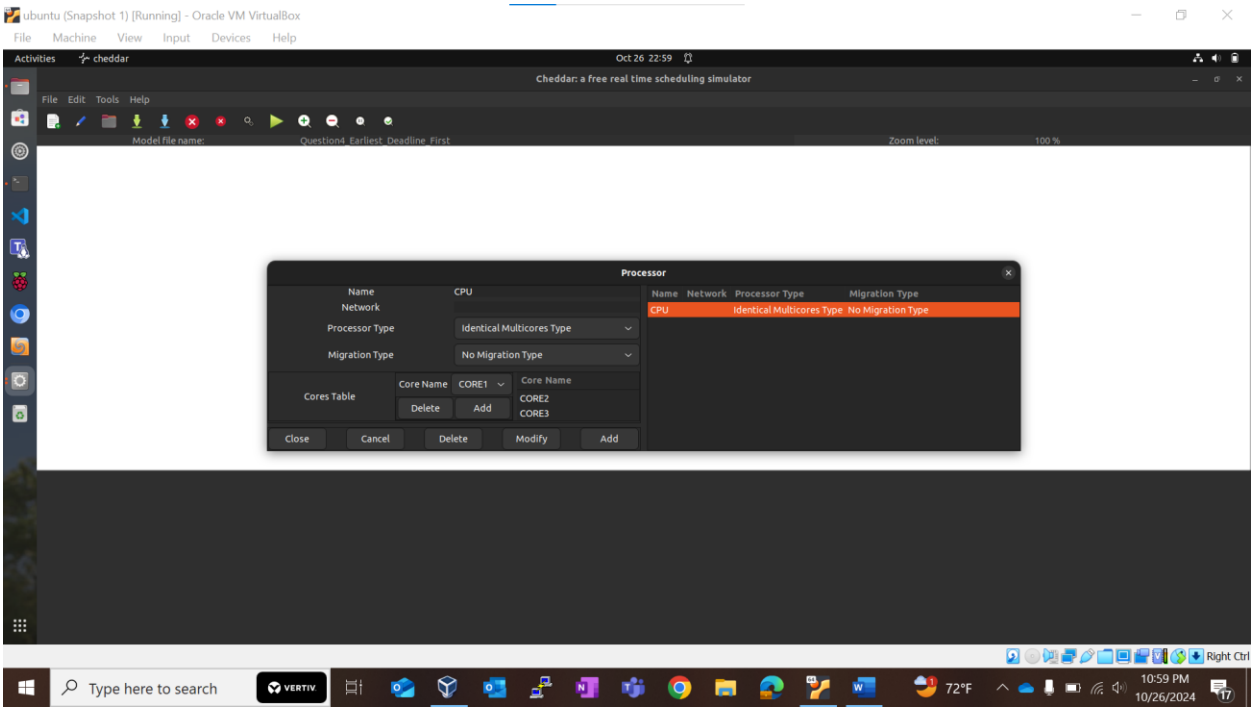
Priority Order at Task Releases

1. **At t=0:**
    - **Priority Order: J1 ≈ J2 (highest priority due to deadline 10) > J4 (deadline 20)**
2. **At t=4:**
    - **Task J3 is released with a deadline of 15.**
    - **Priority Order becomes: J1 ≈ J2 (deadline 10) > J3 (deadline 15) > J4 (deadline 20)**

Added 3 CORE's

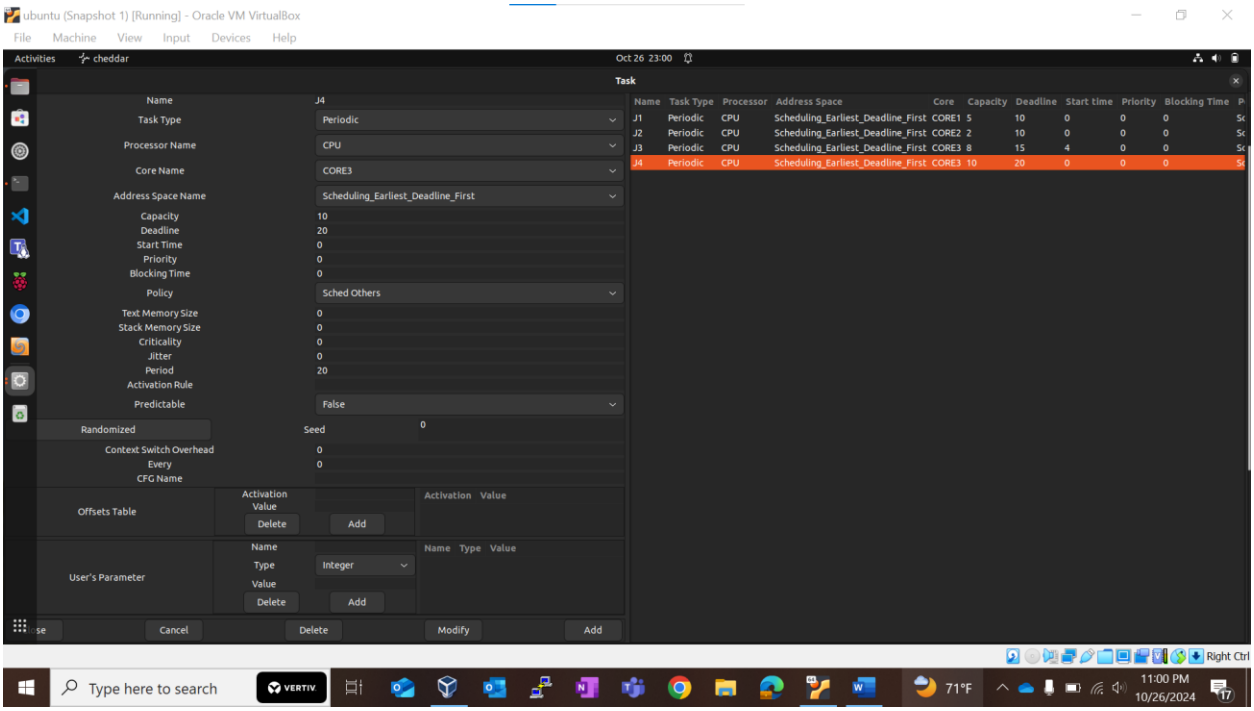


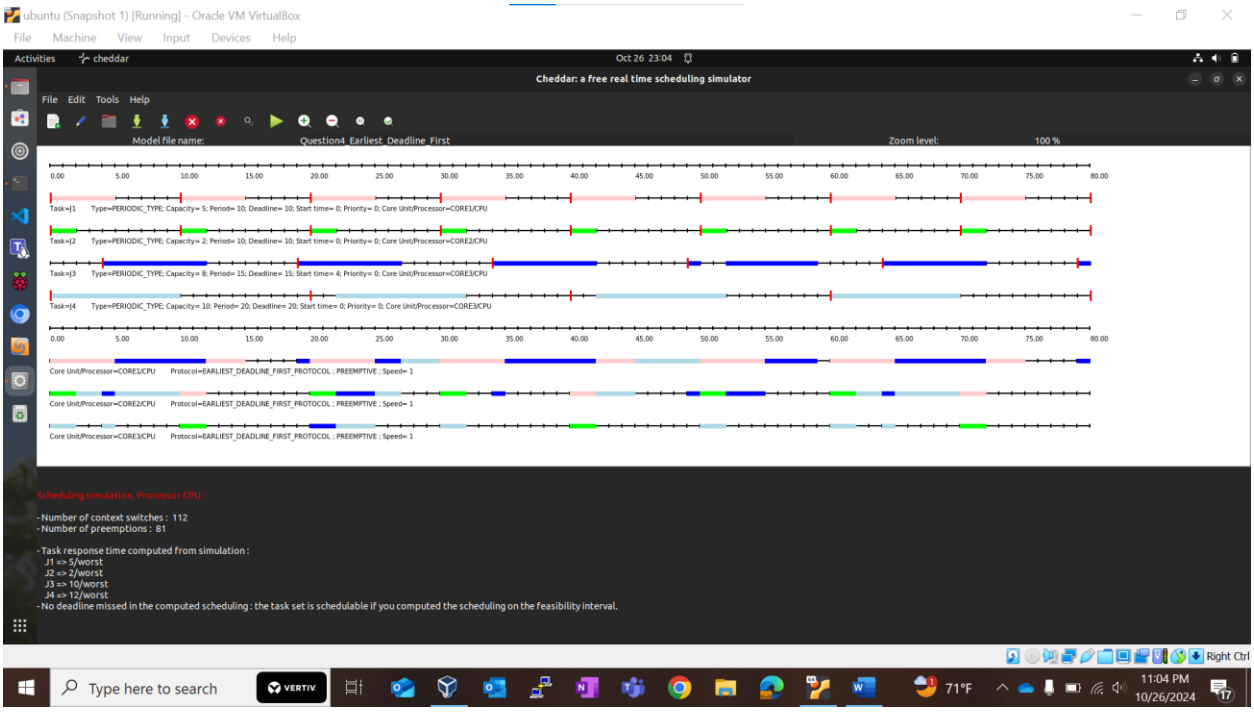

ubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities cheddar Oct 26 22:59
Cheddar: a free real time scheduling simulator
File Edit Tools Help
Model file name: Question4_Earliest_Deadline_First
Zoom level: 100 %
Processor
Name CPU
Network
Processor Type Identical Multicores Type
Migration Type No Migration Type
Cores Table
Core Name CORE1
Delete Add
Core Name
CORE1
CORE2
Close Cancel Delete Modify Add
Name Network Processor Type Migration Type
CPU Identical Multicores Type No Migration Type
Right Ctrl
Type here to search
72°F
10:59 PM
10/26/2024


ubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities cheddar Oct 26 22:59
Cheddar: a free real time scheduling simulator
File Edit Tools Help
Model file name: Question4_Earliest_Deadline_First
Zoom level: 100 %
Processor
Name CPU
Network
Processor Type Identical Multicores Type
Migration Type No Migration Type
Cores Table
Core Name CORE1
Delete Add
Core Name
CORE2
CORE3
Close Cancel Delete Modify Add
Name Network Processor Type Migration Type
CPU Identical Multicores Type No Migration Type
Right Ctrl
Type here to search
72°F
10:59 PM
10/26/2024

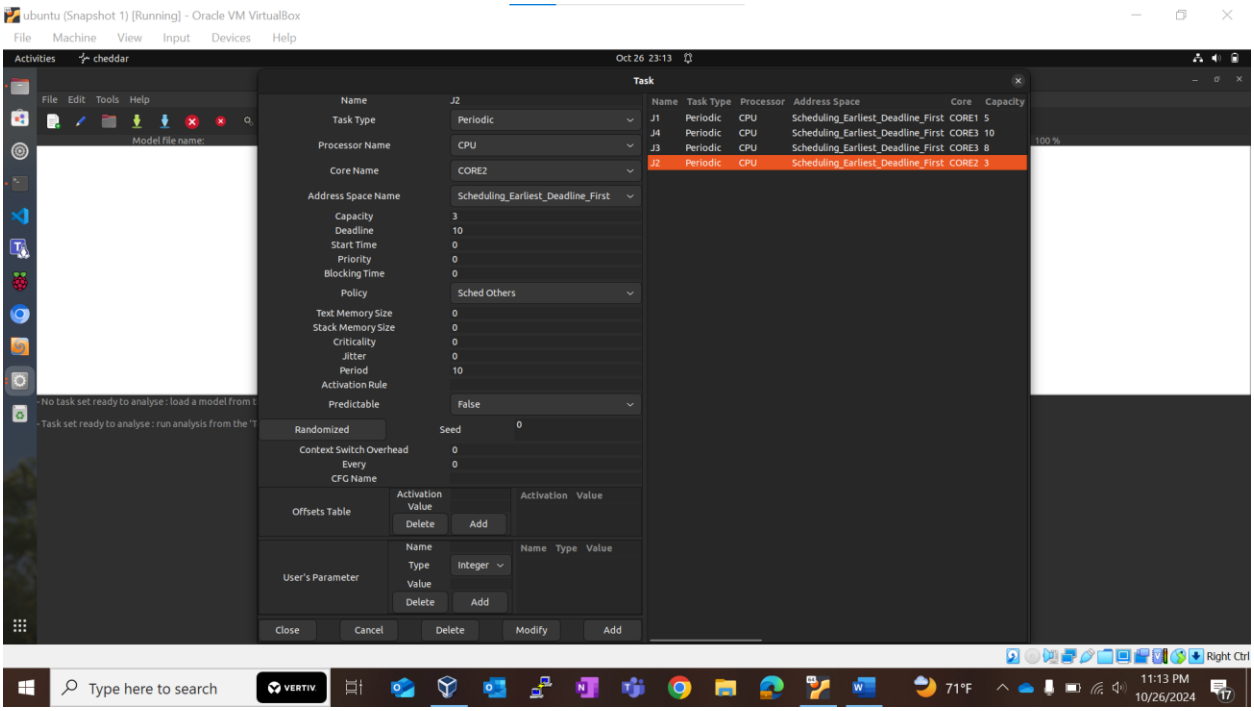
Results:

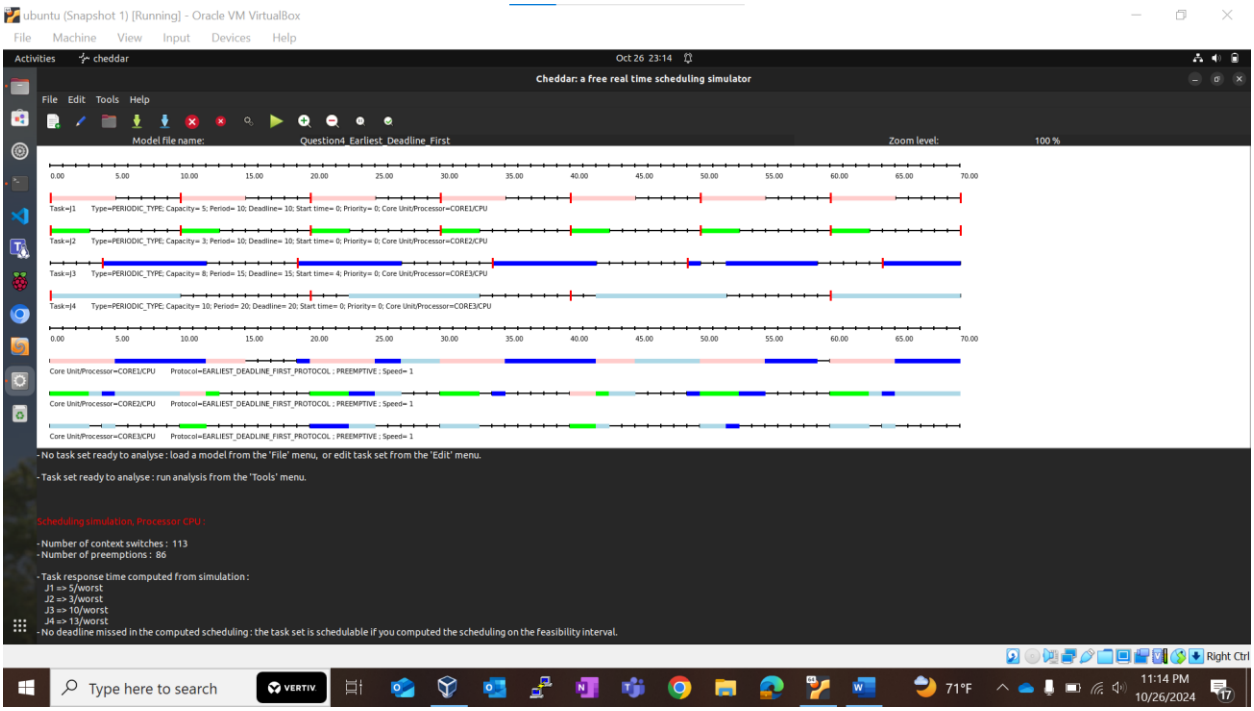
1. **J2 Execution Time = 2**



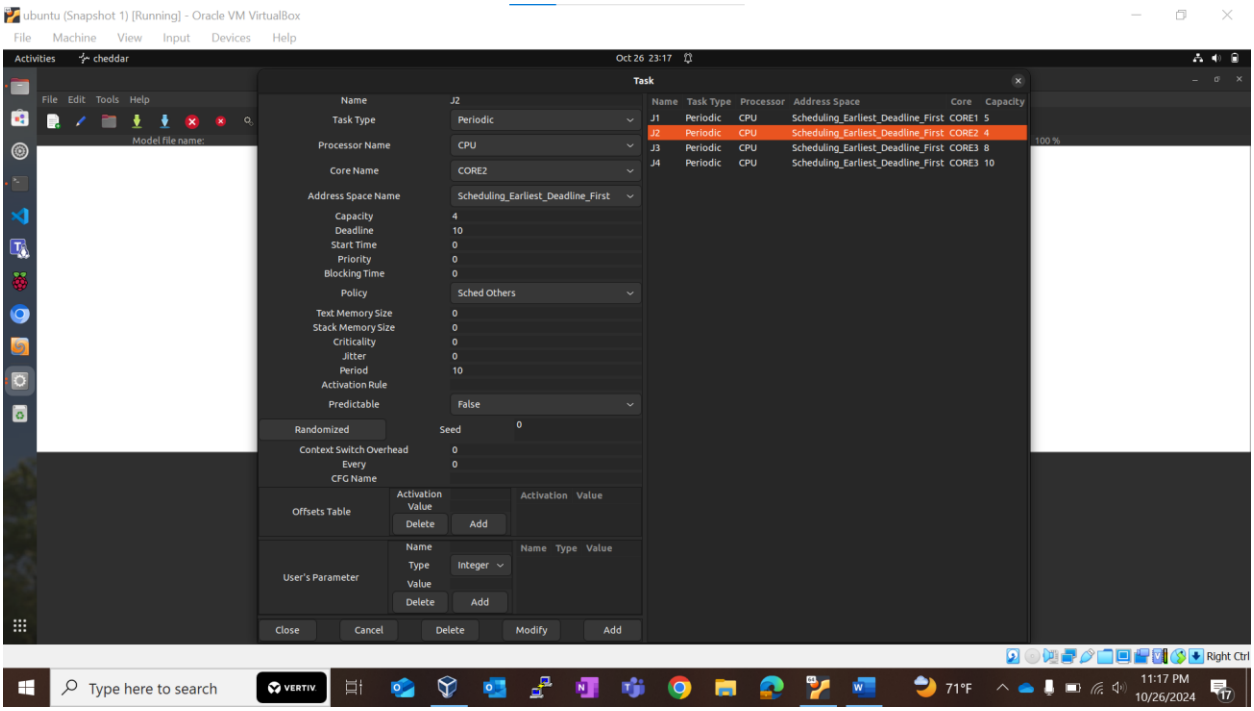


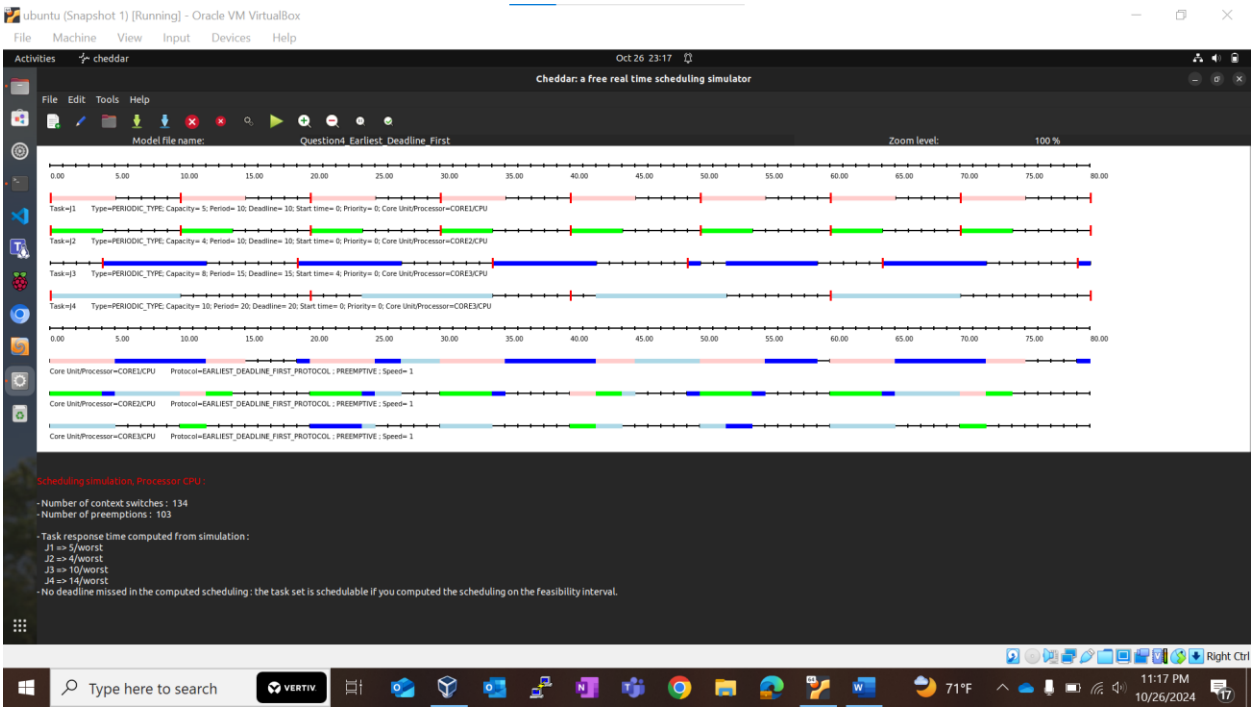
2. **J2 Execution Time = 3**



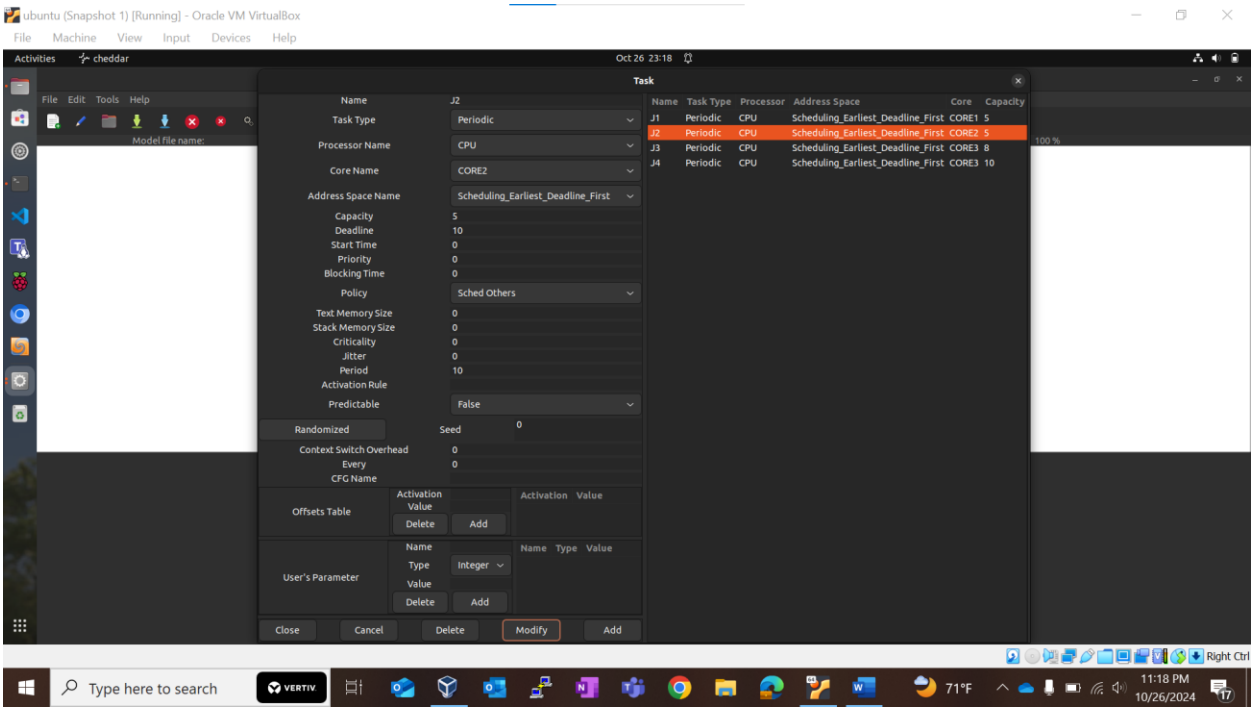


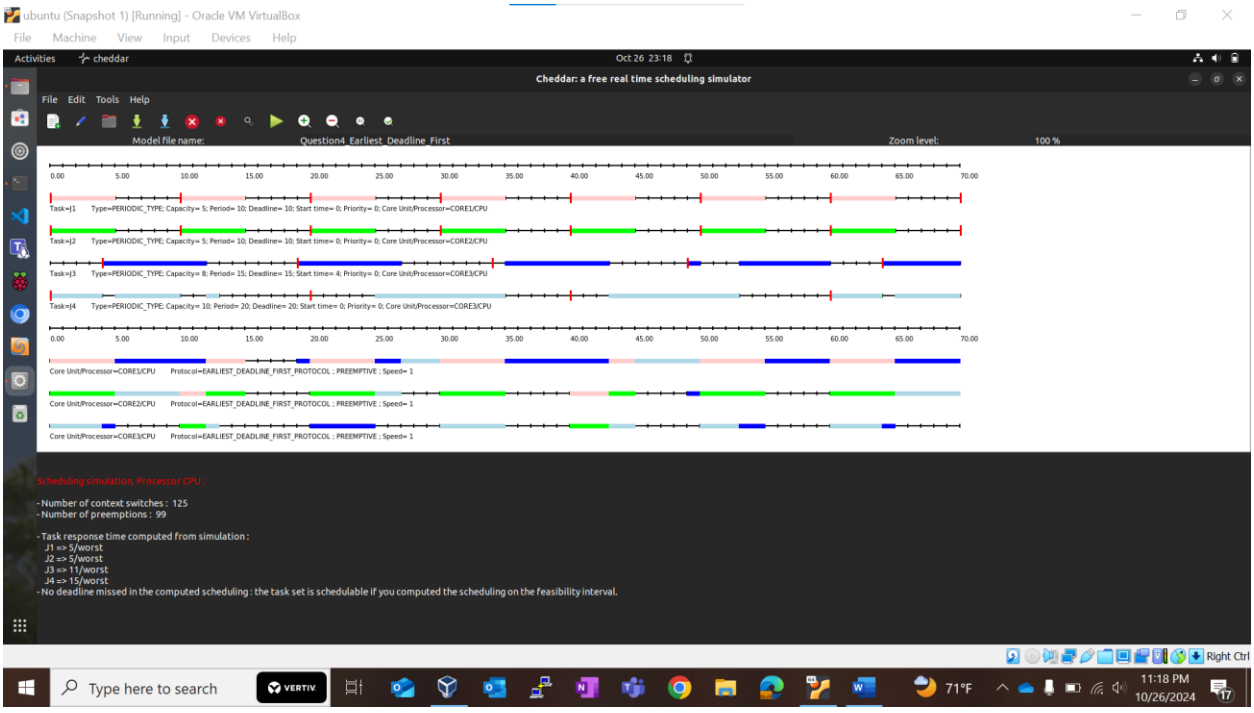
3. **J2 Execution Time = 4**



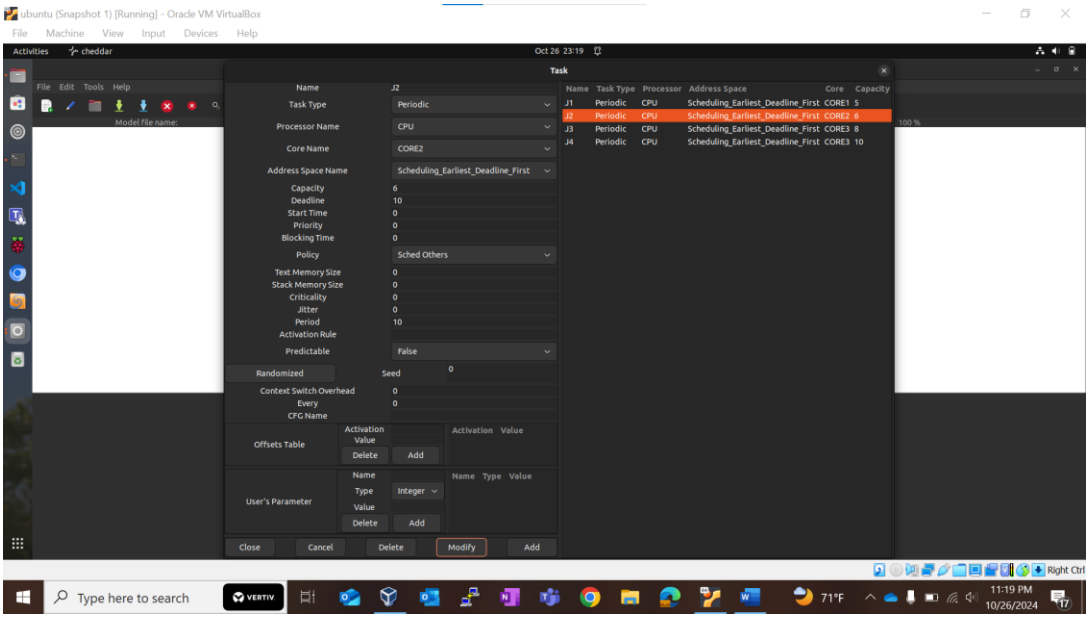


4. **J2 Execution Time = 5**





5. **J2 Execution Time = 6**



Appendix :

Click here for Github link (code).

THANKS