*E x* *p* *e* *r* *i* *m* *e* *n* *t* *3*

Implementation of algorithm for Multiprocessing

System

Objectives:

In this experiment you will learn how to implement algorithm for multiprocessing system

**TABLE OF CONTENTS**

* 1. Objective
  2. References
  3. Theoretical Background
  4. Pre Lab Assignment
  5. In-Lab Experimental Procedure
  6. Outcomes

Appendix A: In-Lab Report

* 1. Objective

This experiment aims to implement the various algorithms on multiprocessing system

* 1. References

1. Cheddae Release 3.x user’s guide

* 1. Theoretical Background

Multiprocessing is the use of two or more central processing units (CPUs) within a single computer system. The term also refers to the ability of a system to support more than one processor or the ability to allocate tasks between them.

There are two types of multiprocessors, one is called shared memory multiprocessor and another is distributed memory multiprocessor. In shared memory multiprocessors, all the CPUs shares the common memory but in a distributed memory multiprocessor, every CPU has its own private memory.

In multiple-processor scheduling multiple CPU’s are available and hence Load Sharing becomes possible. However multiple processor scheduling is more complex as compared to single processor scheduling. In multiple processor scheduling there are cases when the processors are identical i.e. homogeneous, in terms of their functionality, we can use any processor available to run any process in the queue.

Approaches to Multiple-Processor Scheduling –

One approach is when all the scheduling decisions and I/O processing are handled by a single processor which is called the Master Server and the other processors executes only the user code. This is simple and reduces the need of data sharing. This entire scenario is called Asymmetric Multiprocessing.

A second approach uses Symmetric Multiprocessing where each processor is self scheduling. All processes may be in a common ready queue or each processor may have its own private queue for ready processes. The scheduling proceeds further by having the scheduler for each processor examine the ready queue and select a process to execute.

.

* 1. In-Lab Experimental Procedure

1.4.1 Starting the simulator

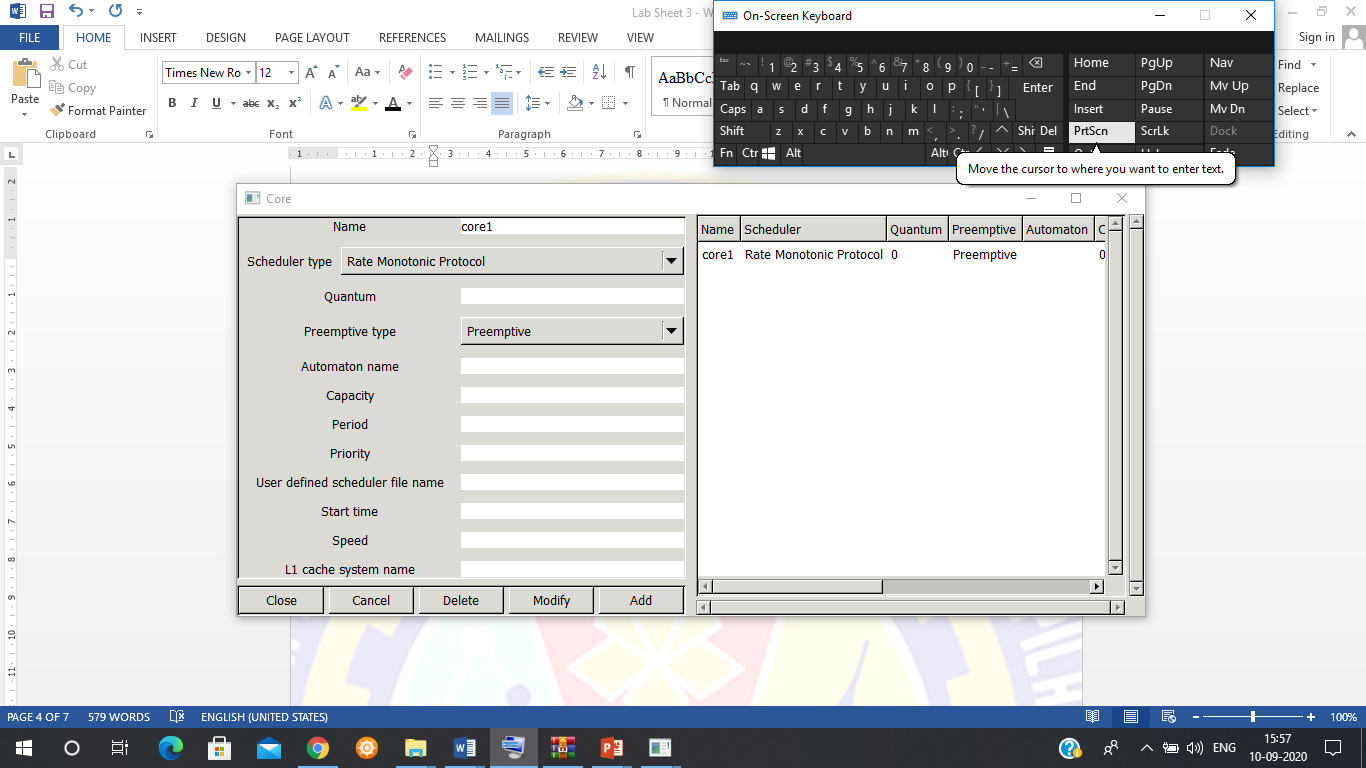
Double click on the cheddar.exe from the ‘Cheddar-3.0-win32-bin’ folder. This will

launch the application.

## 1.4.2 Building a new project:

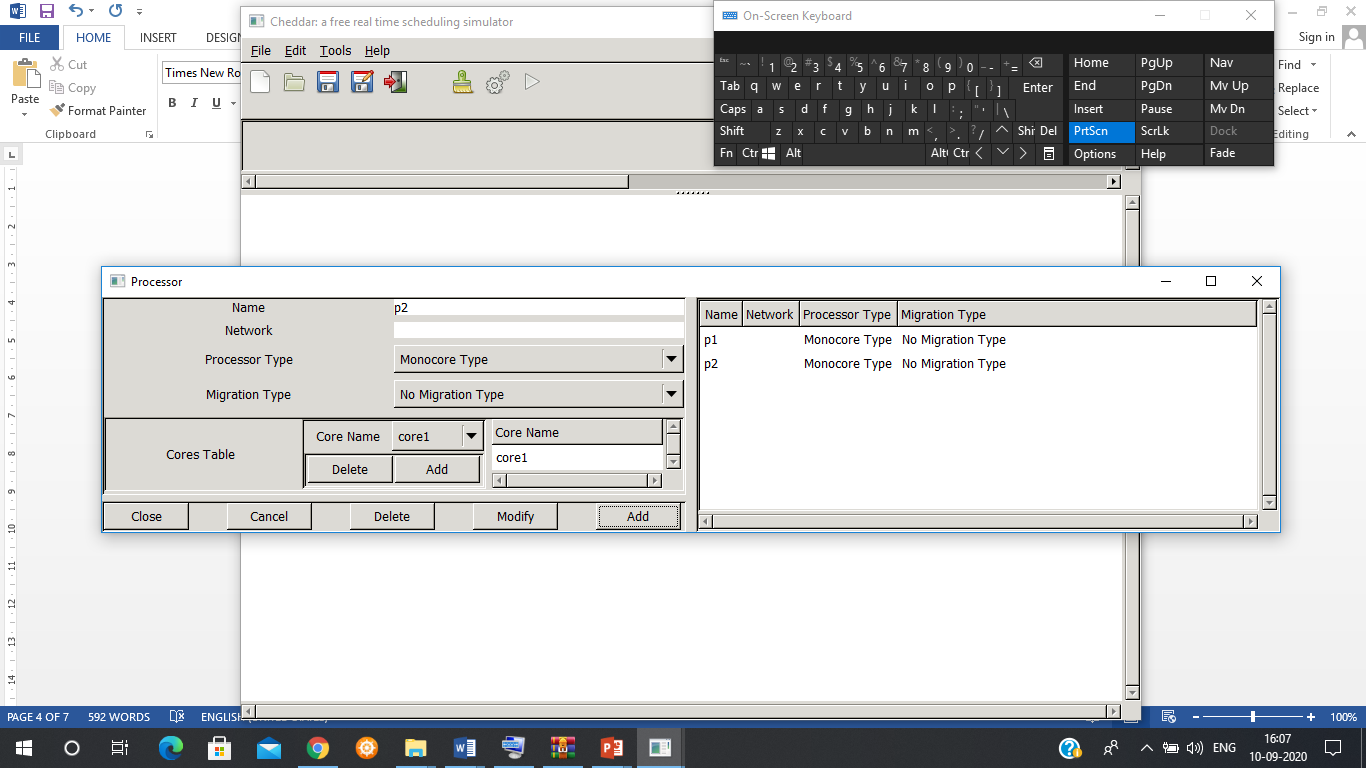
1. From the file icon in the menu bar, choose ‘new’ XML project. All newly created files have to be saved in the current path with an extension of .sc.
2. **Adding a core:**

* From the menu bar, Edit -> hardware -> Core.



**Adding the Processor:**

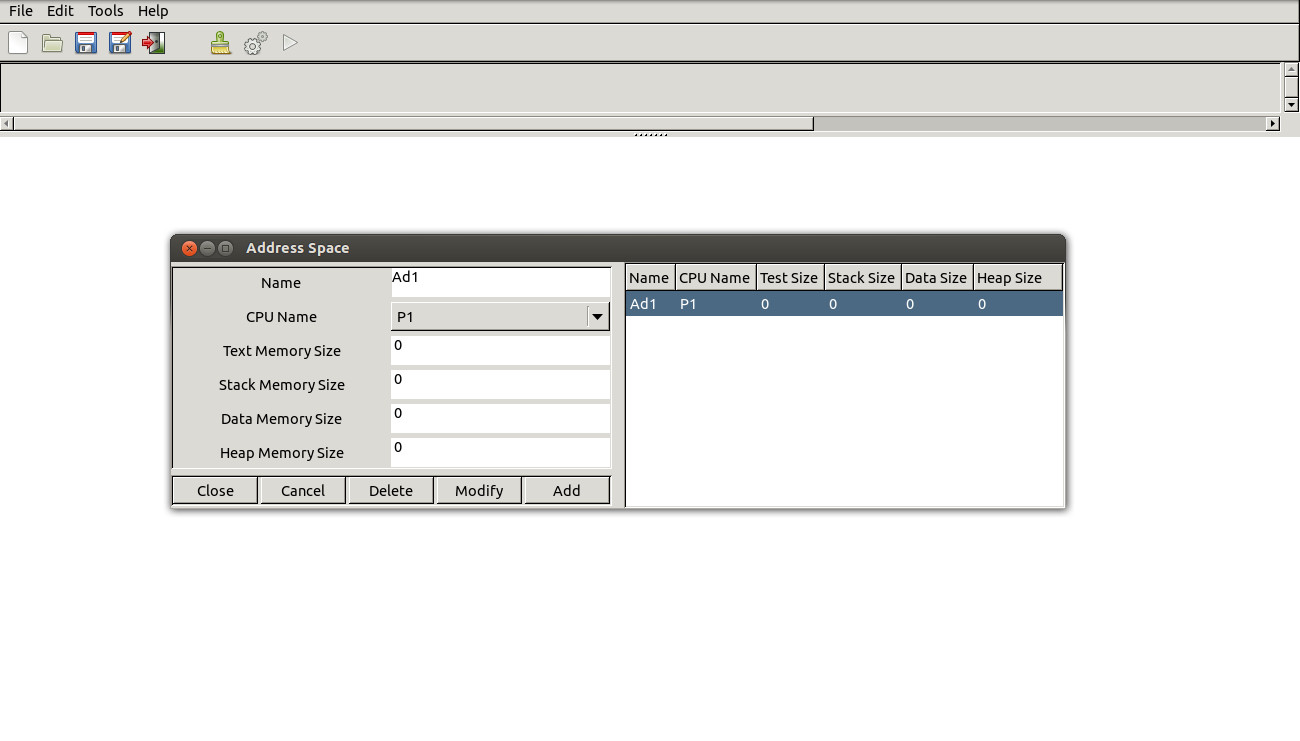
* From the menu bar, Edit -> Hardware -> Processor.



* Add the name of the processor.
* Add the core from the ‘cores table’.
* The Processor name along with the other parameters will be reflected on the right hand side.

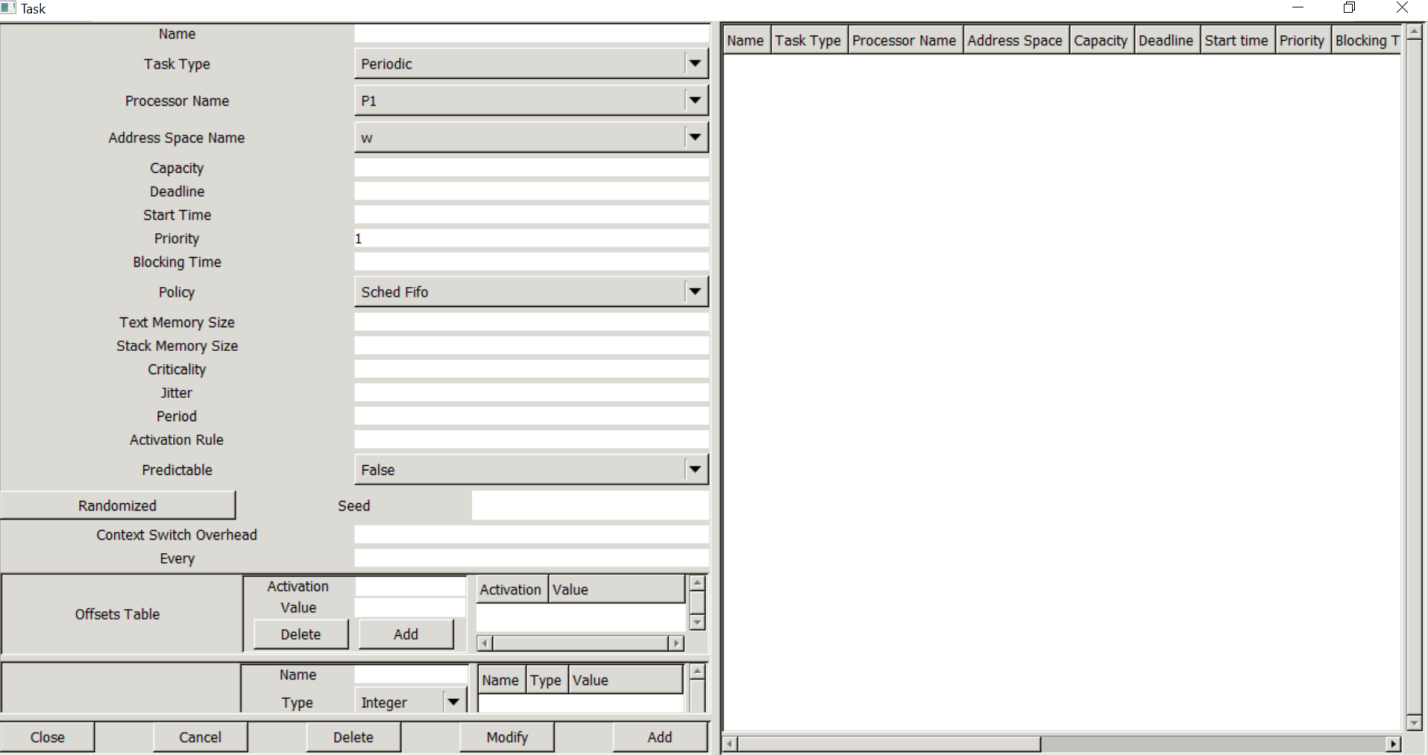
**Adding the address space:**

From the menu bar, Edit -> software -> Address space.



**Adding a task:**

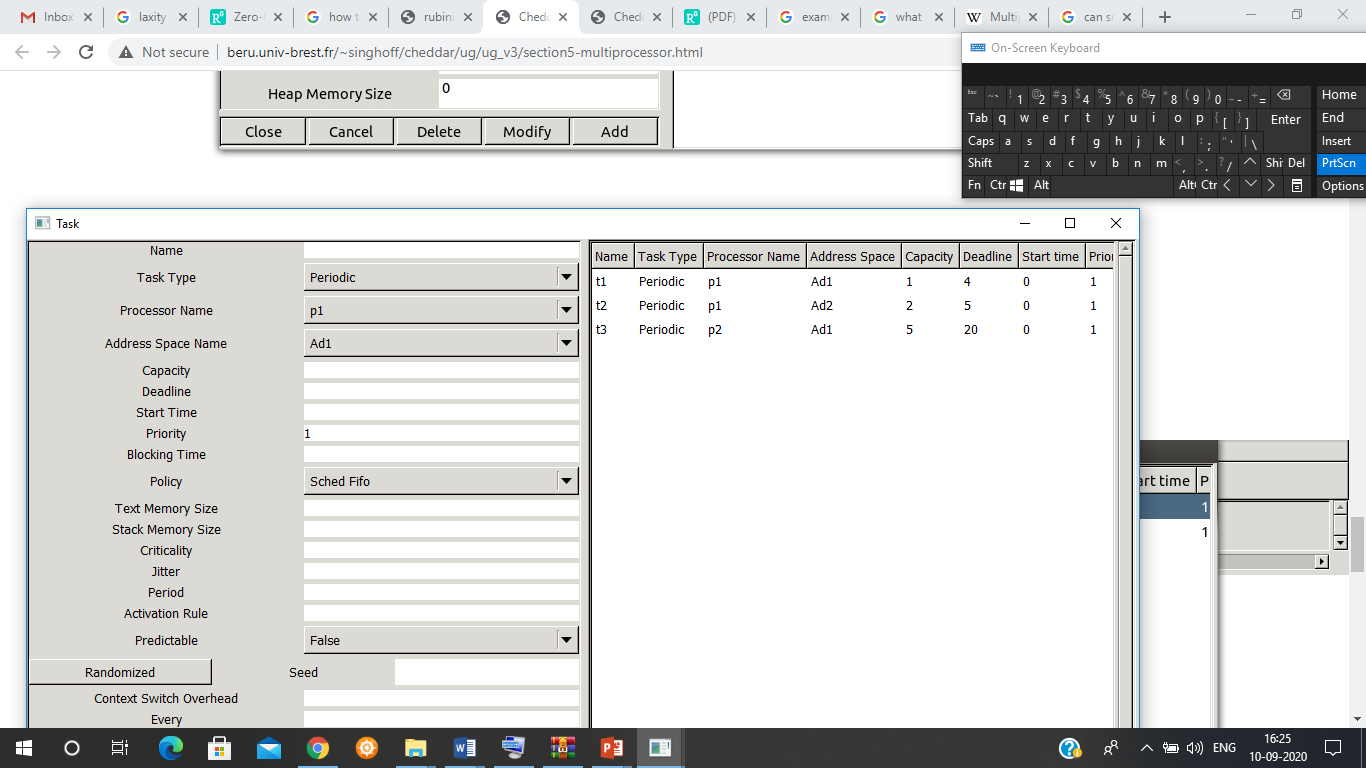
* From the menu bar, Edit -> software -> task.



* Add the name of the task.
* Task type
* Processor name: Drop down list contains all the processors defined. Choose the appropriate processor for the task.
* Capacity: Worst case Execution time of the task.
* Deadline: Absolute deadline. Refers to the completion time for the task.
* Start time: The time offset for the task, that is, the time at which the task will be ready for execution.

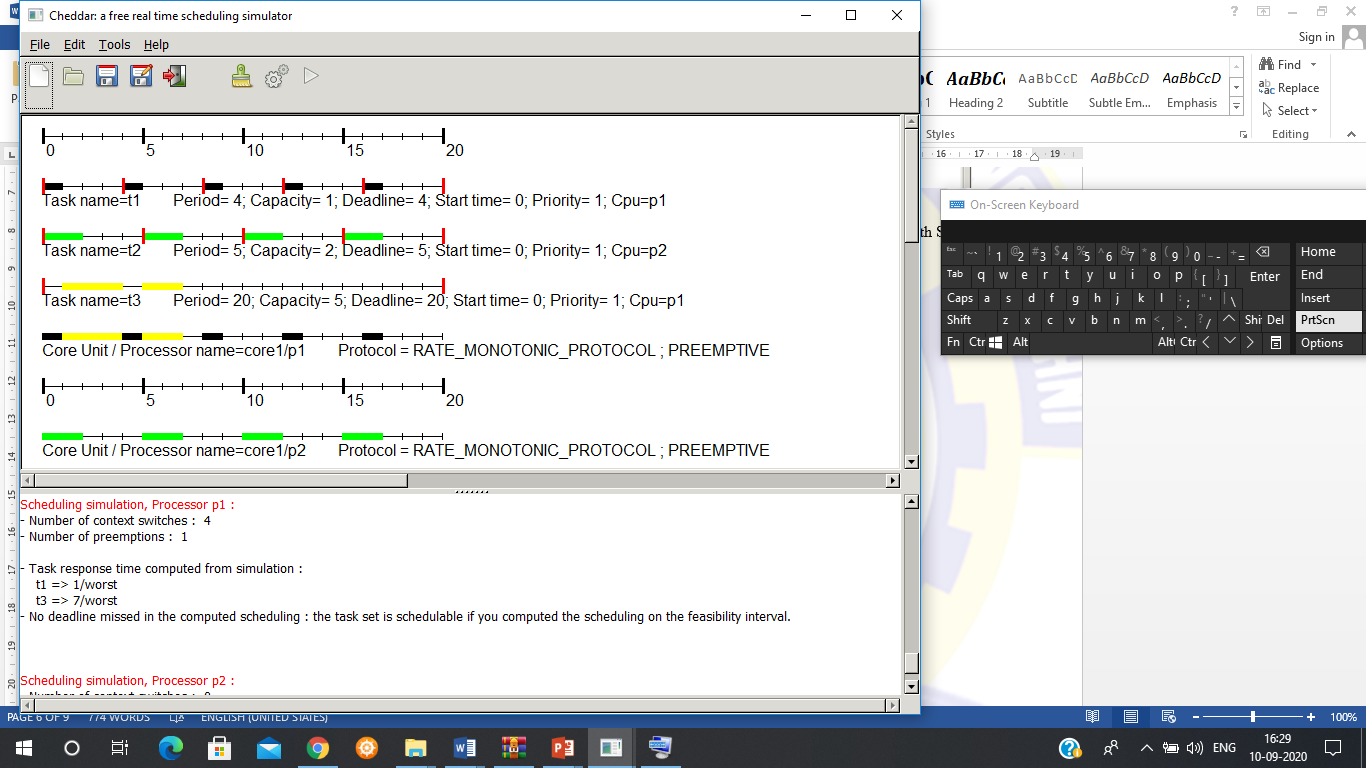
Priority: To define the priority of the task. When 2 or more tasks have the same priority, the scheduler will execute the task depending upon the scheduling policy.

* Period: The periodicity of the task, that is, the time after which the task will reoccur. (note: Deadline must be lesser than the periodicity). Do not specify periodicity for aperiodic and sporadic task.



Finally, compute partitioning, with the submenu "Tool/Scheduling/Partition/With Small Task". :

* 1. Outcomes



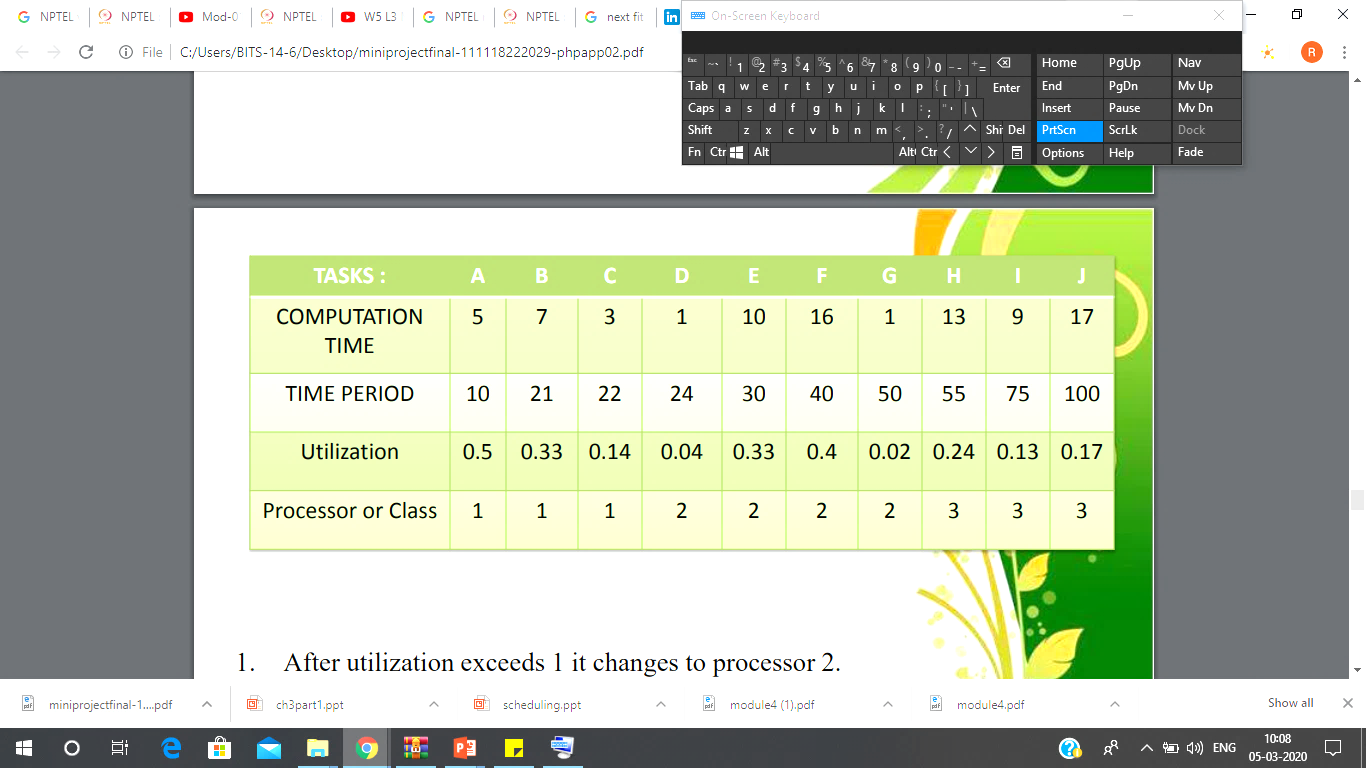
The task set can be simulated for next fit, best fit etc

Appendix A

***Experiment 3***

**In-Lab Report**

For the following Task set and scheduling algorithms – submit the screen shot of the schedule generated by Cheddar – with proper interpretations and conclusions regarding the schedule generated.

****