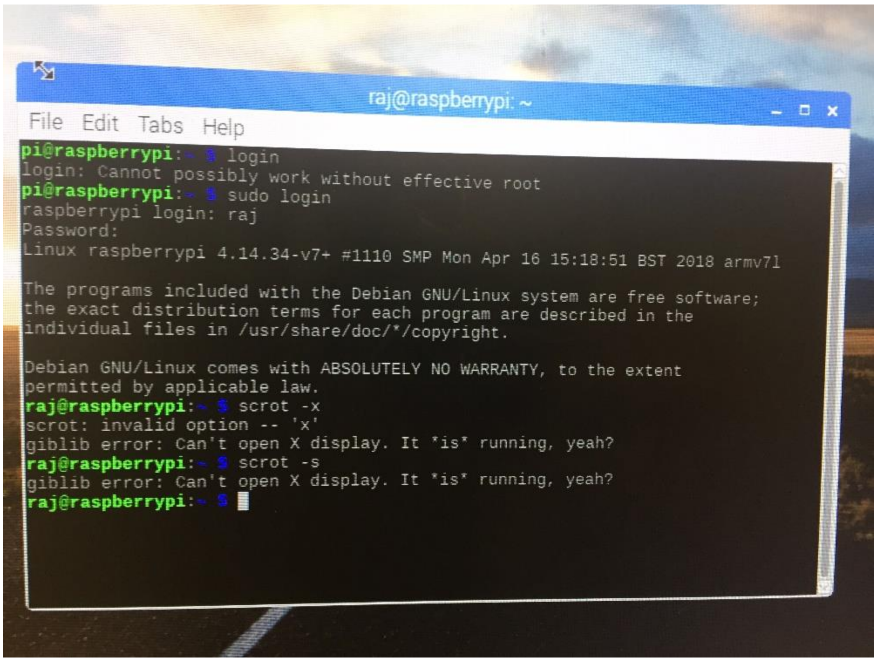
RTES Assignment - 2

Question :- 1



```
pi@raspberrypi:~ $ login
login: Cannot possibly work without effective root
pi@raspberrypi:~ $ sudo login
raspberrypi login: raj
Password:
Linux raspberrypi 4.14.34-v7+ #1110 SMP Mon Apr 16 15:18:51 BST 2018 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
raj@raspberrypi:~ $ scrot -x
scrot: invalid option -- 'x'
giblib error: Can't open X display. It *is* running, yeah?
raj@raspberrypi:~ $ scrot -s
giblib error: Can't open X display. It *is* running, yeah?
raj@raspberrypi:~ $ 
```

Figure :- 1 Username and password added

Question – 2

The Paper written by Gene D Carlow represents the architecture developed by IBM Federal Systems Division for NASA's space shuttle orbiter **PASS(Primary Avionics Software System)** . It is by far the most complex flight computer program ever developed. It employs state-of the-art multiple computer redundancy management concepts and incorporates a multitude of functions required to support Shuttle operations on the ground and in flight. It serves as the central system for all the Avionics System Functions.

Architectural Drivers :-

Factors like Data Processing System(DPS)/ IBM Ap-101, General Purpose Computer Design(GPC) and memory/CPU Constraints the multicomputer redundancy management and synchronization; the operational sequencing/mode control and man/machine interface requirements; the applications functional and performance requirements; and the requirement for design modularity and modification flexibility necessitated by the R&D nature of the Shuttle Program.

Operational Structure :-

The main memory of the PASS of AP -101 (106K – 32 Bit words) was not sufficient to contain all the software (500K),) so they used to load the software into the main memory upon the function requirement.

These constraints have been addressed with an operational structure that is based on the combination of mission phases and major application functional requirements. The segmentation of the PASS has been divided into 8 different phase/function combinations, each of which is identified with a unique operational sequence (OPS). The software required for each OPS is loaded into the GPC main memory from the mass memory at the initiation of the OPS. Except for a few specific non-mission critical situations, such as the loading of a display format during on-orbit coast periods, transfers between mass memory and a GPC main memory do not occur during the execution of an OPS.

Man/Machine Interface :-

PASS has Man/Machine Interface which is structured to accommodate a knowledgeable user, with minimum time and effort required to communicate with the computers to monitor and control a very complex Orbiter avionic system. Each OPS has one or more modes. Major modes are further sub structured into blocks that are linked to CRT display pages so as to establish an orderly sequence for the crew to communicate and maintain control of the software.

Sequencing from one mode/block to another and the processing performed within are initiated either by keyboard entry from the crew or, in some cases, automatically as the result of a specific event or condition detected by the software. At least one major mode will be entered any time an OPS is initiated. OPS also includes Specialist Function (SPEC) , Display Function (DISP). The major difference between them is that no processing function is initiated by DISP. They created a library of macros called the control segment grammar to provide standard and modularity for the design and implementation of control segments.

Systems Software:-

Flight software systems have been characterized by their small size (15K to 30K) and the limited number of functions they perform. It is a synchronous design where each application process occurs at specific point relative to the start of an overall system cycle. PASS adopted a nonsynchronous architecture where sequencing and control of the applications are accomplished through interfaces with the major elements, management and control of the GPC internal and external resources are accomplished by the FCOS, system control does the loading and initialization, user interface deals with the communication between user and systems and Inter-computer communication and redundant computer synchronization is handled by systems software. They use repeatability (However, there is a limited flexibility to accommodate changes) . Isolation of application processes from the external I/O and computer redundancy management. Sequencing and control of applications are accomplished through interfaces with the major elements of system software.

Applications Software :-

These three applications are guidance, navigation, and control (GN&C); vehicle systems management (SM); and vehicle checkout (VCO) . Vehicle checkout provides software support for the testing, integration, and/or certification of the Orbiter avionics subsystems during vehicle preparation on the ground and in-flight orbit coast period prior to entry. GN&C determines the vehicle position, velocity and attitude. SM monitors the performance and configuration of Orbiter and payload systems. VCO provides software support for the testing, integration and certification of the Orbiter avionics subsystems. Central of the GN&C design structure is the cyclic process which is initiated by the OPS control segment using FCOS process scheduling.

GN&C functions, of which there are approximately 200 called principal functions, are included in six different OPSs, three of which must execute in a redundant computer configuration. There are from one to ten major modes included in the six OPSs, and ten SPECs, each of which is available in one or more of the OPSs. The execution rates for the principal functions within an OPS vary from 25 Hz for support of the basic vehicle flight control, down to 0.25 Hz for display update.

Mid-frequency and low frequency executives are scheduled at lower priorities at rates of 6.25 Hz down to 0.25 Hz. The high-frequency executive initiates all principal function processes directly related to vehicle flight control. The mid and low frequency executive initiate principal function processes.

Implementation :-

Standardization of formats for external data input to these pre-processors has reduced the possibility of error through syntax and reasonableness checking. With a very few exceptions, the HAL/S compiler, developed for the NASA Shuttle applications, is used for all PASS software outside of the FCOS. Interface with the FCOS facilities is accomplished through a standard set of service macros (SVCs) supported in the HAL/S language. The structural aspects of the software architecture and HAL/S language together with the standardization of interfaces through the FCOS SVCs and control segment grammar have been major factors in the successful implementation of PASS software.

Advantages and Disadvantages :-

Advantages :-

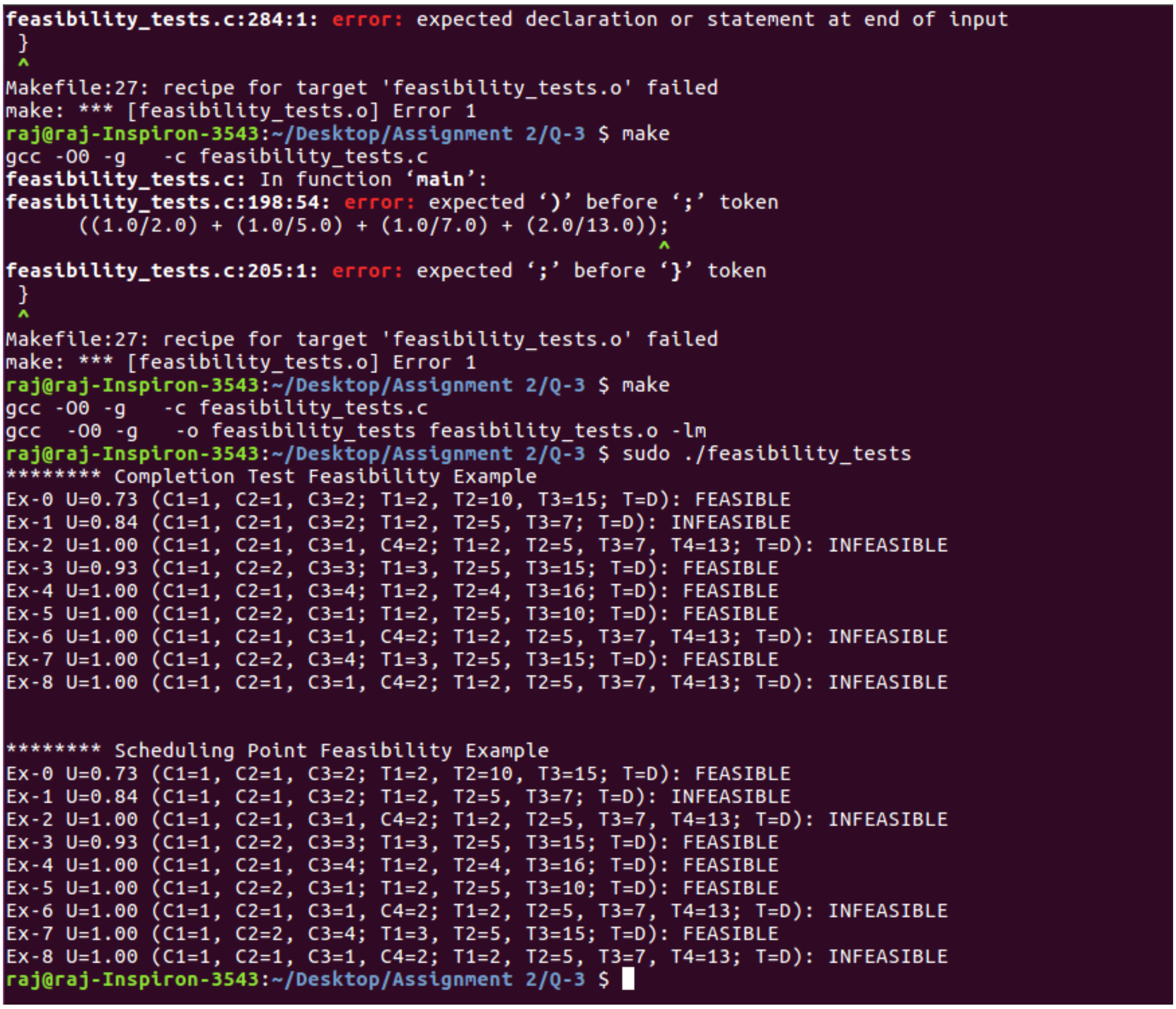
1) No OS required
2) No real time systems is necessary
3) Used in Critical Missions and safety mechanisms.
4) Almost no jitter is present
5) We can predict the behaviour and thus chances of unknown errors are less.
6) The code required can be made smaller as compared.

Disadvantages :-

1) It does not have capability to deal with any run time changes
2) Building is not flexible
3) It is costly to develop and maintain
4) Multiple polling services do not scale well
5) Tasks with dependencies cannot be acted well without problems
6) Concurrent Programming is not achievable

Question - 3

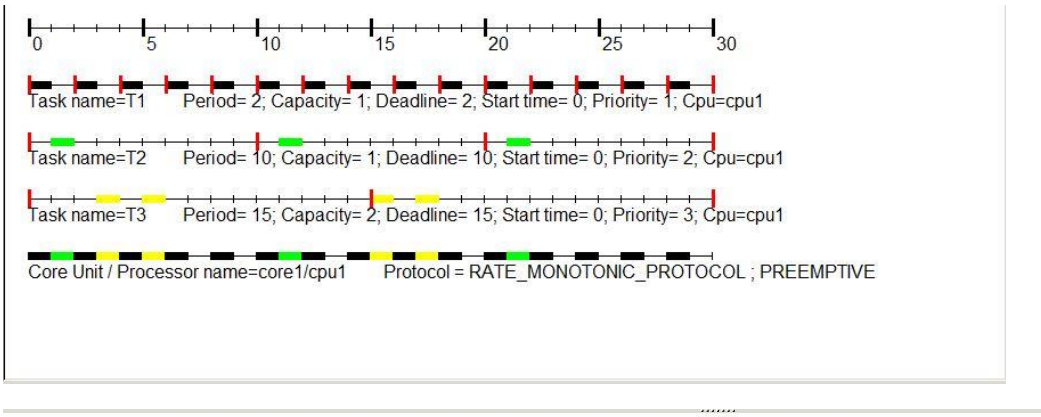
Code of Prof Sam Siewert , edited and run for feasibility check.



```
feasibility_tests.c:284:1: error: expected declaration or statement at end of input
 }
 ^
Makefile:27: recipe for target 'feasibility_tests.o' failed
make: *** [feasibility_tests.o] Error 1
raj@raj-Inspiron-3543:~/Desktop/Assignment 2/Q-3 $ make
gcc -O0 -g   -c feasibility_tests.c
feasibility_tests.c: In function ‘main’:
feasibility_tests.c:198:54: error: expected ‘)’ before ‘;’ token
      ((1.0/2.0) + (1.0/5.0) + (1.0/7.0) + (2.0/13.0));
                                                     ^
feasibility_tests.c:205:1: error: expected ‘;’ before ‘}’ token
 }
 ^
Makefile:27: recipe for target 'feasibility_tests.o' failed
make: *** [feasibility_tests.o] Error 1
raj@raj-Inspiron-3543:~/Desktop/Assignment 2/Q-3 $ make
gcc -O0 -g   -c feasibility_tests.c
gcc  -O0 -g   -o feasibility_tests feasibility_tests.o -lm
raj@raj-Inspiron-3543:~/Desktop/Assignment 2/Q-3 $ sudo ./feasibility_tests
******** Completion Test Feasibility Example
Ex-0 U=0.73 (C1=1, C2=1, C3=2; T1=2, T2=10, T3=15; T=D): FEASIBLE
Ex-1 U=0.84 (C1=1, C2=1, C3=2; T1=2, T2=5, T3=7; T=D): INFEASIBLE
Ex-2 U=1.00 (C1=1, C2=1, C3=1, C4=2; T1=2, T2=5, T3=7, T4=13; T=D): INFEASIBLE
Ex-3 U=0.93 (C1=1, C2=2, C3=3; T1=3, T2=5, T3=15; T=D): FEASIBLE
Ex-4 U=1.00 (C1=1, C2=1, C3=4; T1=2, T2=4, T3=16; T=D): FEASIBLE
Ex-5 U=1.00 (C1=1, C2=2, C3=1; T1=2, T2=5, T3=10; T=D): FEASIBLE
Ex-6 U=1.00 (C1=1, C2=1, C3=1, C4=2; T1=2, T2=5, T3=7, T4=13; T=D): INFEASIBLE
Ex-7 U=1.00 (C1=1, C2=2, C3=4; T1=3, T2=5, T3=15; T=D): FEASIBLE
Ex-8 U=1.00 (C1=1, C2=1, C3=1, C4=2; T1=2, T2=5, T3=7, T4=13; T=D): INFEASIBLE


******** Scheduling Point Feasibility Example
Ex-0 U=0.73 (C1=1, C2=1, C3=2; T1=2, T2=10, T3=15; T=D): FEASIBLE
Ex-1 U=0.84 (C1=1, C2=1, C3=2; T1=2, T2=5, T3=7; T=D): INFEASIBLE
Ex-2 U=1.00 (C1=1, C2=1, C3=1, C4=2; T1=2, T2=5, T3=7, T4=13; T=D): INFEASIBLE
Ex-3 U=0.93 (C1=1, C2=2, C3=3; T1=3, T2=5, T3=15; T=D): FEASIBLE
Ex-4 U=1.00 (C1=1, C2=1, C3=4; T1=2, T2=4, T3=16; T=D): FEASIBLE
Ex-5 U=1.00 (C1=1, C2=2, C3=1; T1=2, T2=5, T3=10; T=D): FEASIBLE
Ex-6 U=1.00 (C1=1, C2=1, C3=1, C4=2; T1=2, T2=5, T3=7, T4=13; T=D): INFEASIBLE
Ex-7 U=1.00 (C1=1, C2=2, C3=4; T1=3, T2=5, T3=15; T=D): FEASIBLE
Ex-8 U=1.00 (C1=1, C2=1, C3=1, C4=2; T1=2, T2=5, T3=7, T4=13; T=D): INFEASIBLE
raj@raj-Inspiron-3543:~/Desktop/Assignment 2/Q-3 $
```

Example:- 0

RM:-

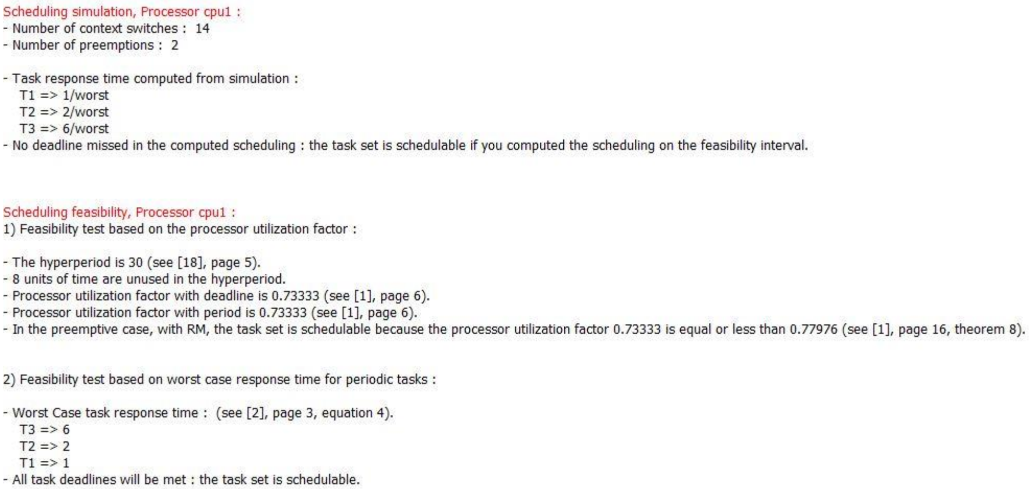


Task name=T1 Period= 2; Capacity= 1; Deadline= 2; Start time= 0; Priority= 1; Cpu=cpu1

Task name=T2 Period= 10; Capacity= 1; Deadline= 10; Start time= 0; Priority= 2; Cpu=cpu1

Task name=T3 Period= 15; Capacity= 2; Deadline= 15; Start time= 0; Priority= 3; Cpu=cpu1

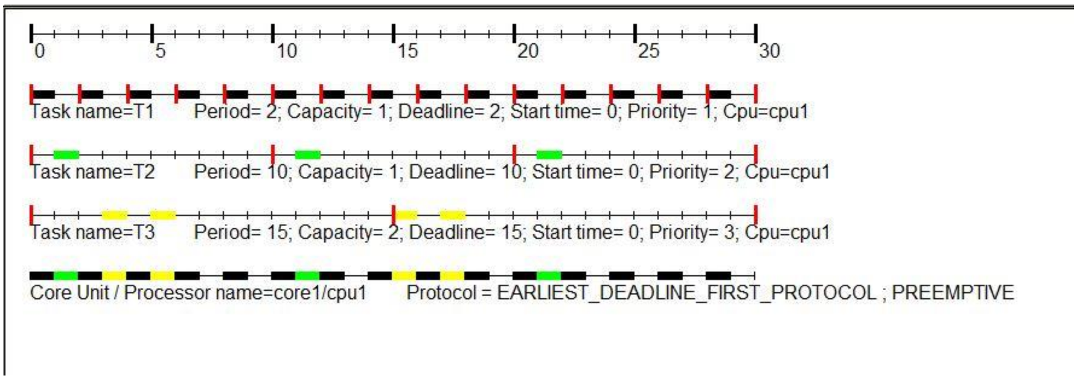
Core Unit / Processor name=core1/cpu1 Protocol = RATE_MONOTONIC_PROTOCOL ; PREEMPTIVE



```
Scheduling simulation, Processor cpu1 :
- Number of context switches :  14
- Number of preemptions :  2

- Task response time computed from simulation :
    T1 => 1/worst
    T2 => 2/worst
    T3 => 6/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 30 (see [18], page 5).
- 8 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.73333 (see [1], page 6).
- Processor utilization factor with period is 0.73333 (see [1], page 6).
- In the preemptive case, with RM, the task set is schedulable because the processor utilization factor 0.73333 is equal or less than 0.77976 (see [1], page 16, theorem 8).


2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :  (see [2], page 3, equation 4).
    T3 => 6
    T2 => 2
    T1 => 1
- All task deadlines will be met : the task set is schedulable.
```

Figure :- 2 RM Diagram & Simulation & Feasibility

EDF:-

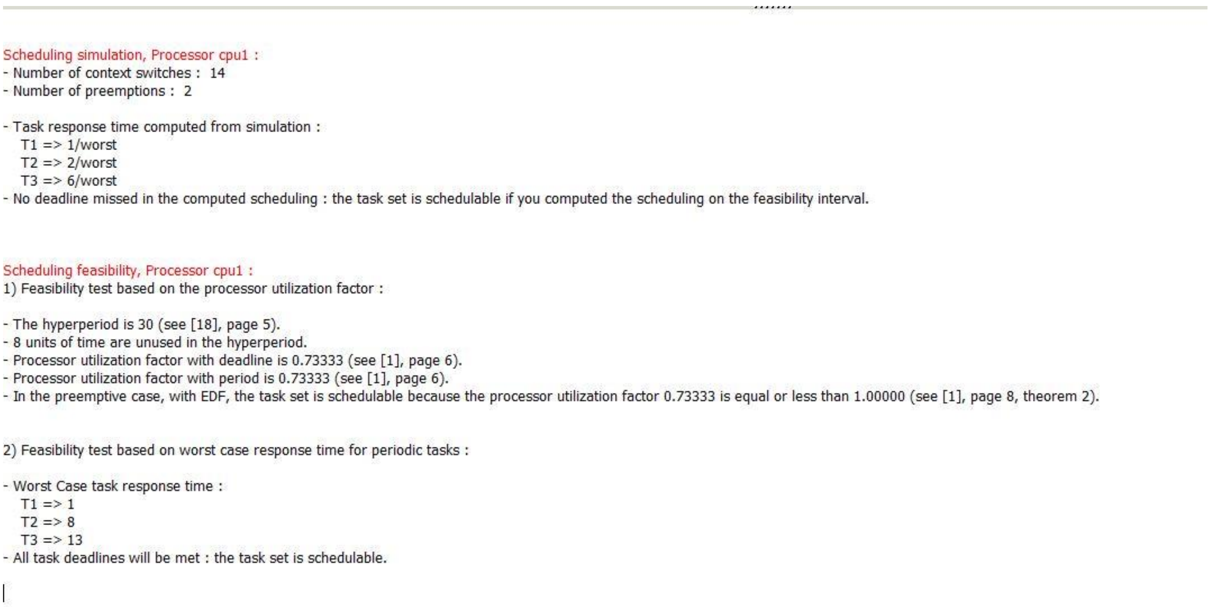


Task name=T1 Period= 2; Capacity= 1; Deadline= 2; Start time= 0; Priority= 1; Cpu=cpu1

Task name=T2 Period= 10; Capacity= 1; Deadline= 10; Start time= 0; Priority= 2; Cpu=cpu1

Task name=T3 Period= 15; Capacity= 2; Deadline= 15; Start time= 0; Priority= 3; Cpu=cpu1

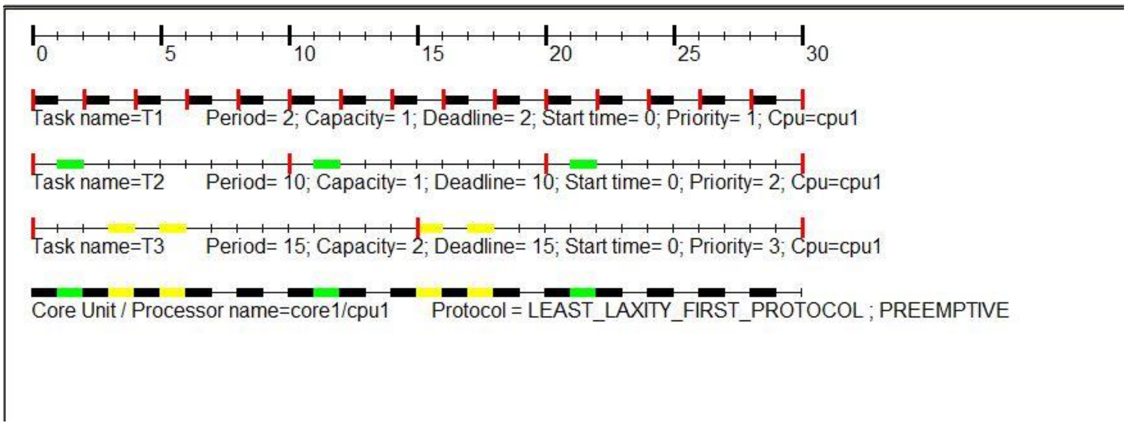
Core Unit / Processor name=core1/cpu1 Protocol = EARLIEST_DEADLINE_FIRST_PROTOCOL ; PREEMPTIVE



```
Scheduling simulation, Processor cpu1 :
- Number of context switches :  14
- Number of preemptions :  2

- Task response time computed from simulation :
    T1 => 1/worst
    T2 => 2/worst
    T3 => 6/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 30 (see [18], page 5).
- 8 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.73333 (see [1], page 6).
- Processor utilization factor with period is 0.73333 (see [1], page 6).
- In the preemptive case, with EDF, the task set is schedulable because the processor utilization factor 0.73333 is equal or less than 1.00000 (see [1], page 8, theorem 2).


2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :
    T1 => 1
    T2 => 8
    T3 => 13
- All task deadlines will be met : the task set is schedulable.
```

Figure :- 3 EDF Diagram & Simulation & Feasibility

LLF:-

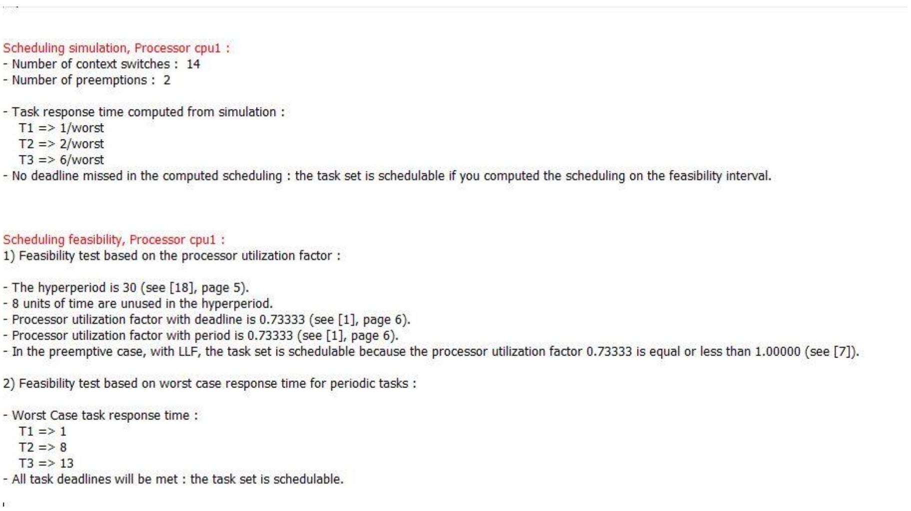


Task name=T1 Period= 2; Capacity= 1; Deadline= 2; Start time= 0; Priority= 1; Cpu=cpu1

Task name=T2 Period= 10; Capacity= 1; Deadline= 10; Start time= 0; Priority= 2; Cpu=cpu1

Task name=T3 Period= 15; Capacity= 2; Deadline= 15; Start time= 0; Priority= 3; Cpu=cpu1

Core Unit / Processor name=core1/cpu1 Protocol = LEAST_LAXITY_FIRST_PROTOCOL ; PREEMPTIVE



```
Scheduling simulation, Processor cpu1 :
- Number of context switches :  14
- Number of preemptions :  2

- Task response time computed from simulation :
    T1 => 1/worst
    T2 => 2/worst
    T3 => 6/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 30 (see [18], page 5).
- 8 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.73333 (see [1], page 6).
- Processor utilization factor with period is 0.73333 (see [1], page 6).
- In the preemptive case, with LLF, the task set is schedulable because the processor utilization factor 0.73333 is equal or less than 1.00000 (see [7]).

2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :
    T1 => 1
    T2 => 8
    T3 => 13
- All task deadlines will be met : the task set is schedulable.
```

Figure:- 4 LLF Diagram & Simulation & Feasibility

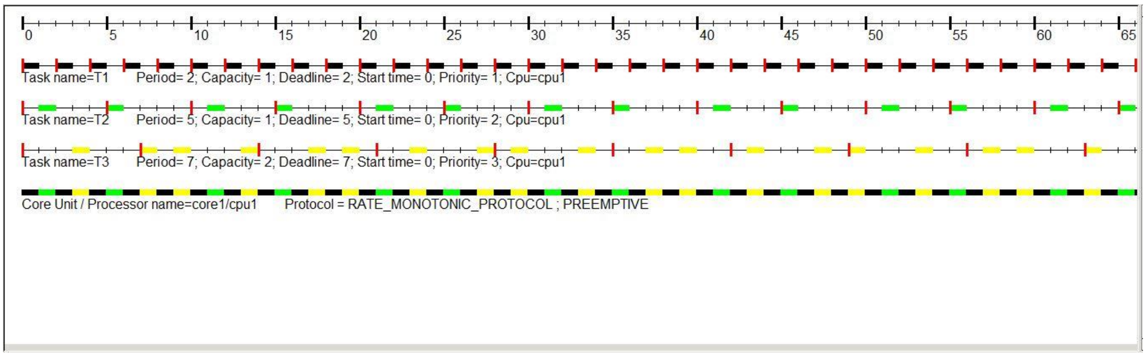
The Cheddar and the code both passes RM,EDF,LLF Feasibility tests.

The code matches/agrees with cheddar. The code algorithm is built in such a way that it checks if the scheduling of the services written is feasible or not.

It also Checks if the total utility is <=1.

Example 1:-

RM:-

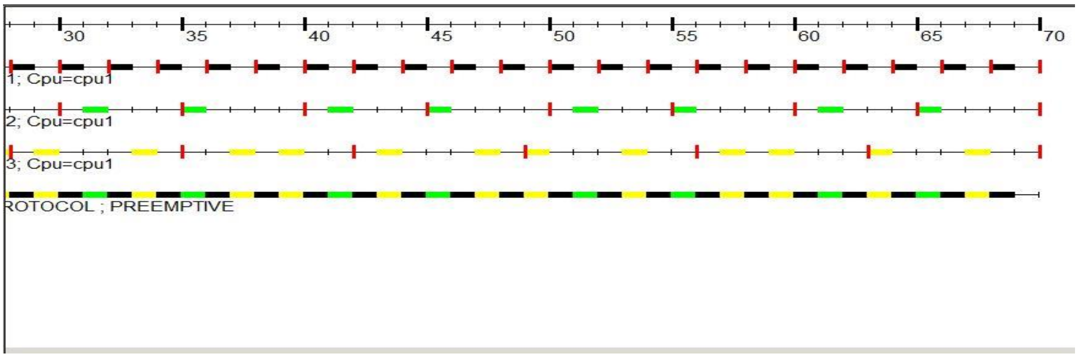


Task name=T1 Period= 2; Capacity= 1; Deadline= 2; Start time= 0; Priority= 1; Cpu=cpu1

Task name=T2 Period= 5; Capacity= 1; Deadline= 5; Start time= 0; Priority= 2; Cpu=cpu1

Task name=T3 Period= 7; Capacity= 2; Deadline= 7; Start time= 0; Priority= 3; Cpu=cpu1

Core Unit / Processor name=core1/cpu1 Protocol = RATE_MONOTONIC_PROTOCOL ; PREEMPTIVE

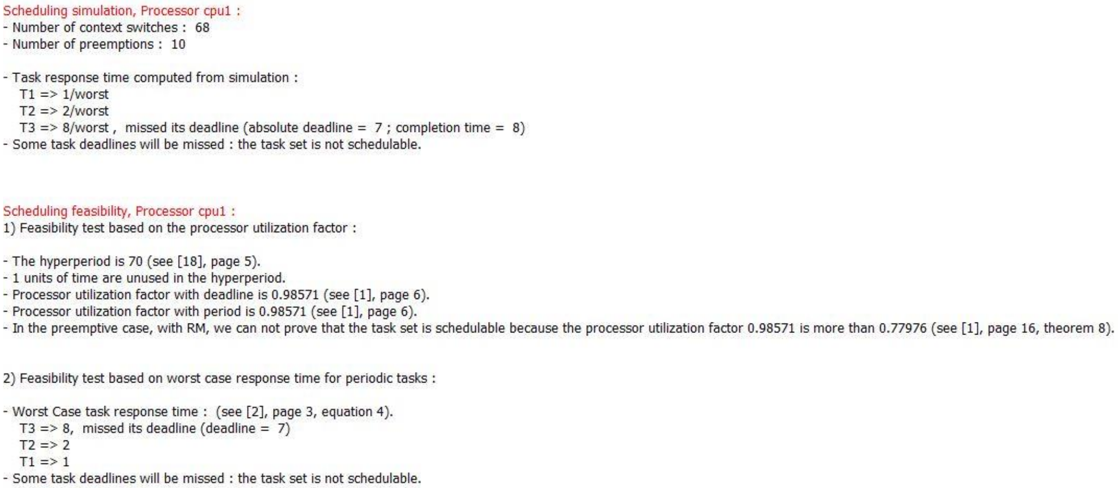


1; Cpu=cpu1

2; Cpu=cpu1

3; Cpu=cpu1

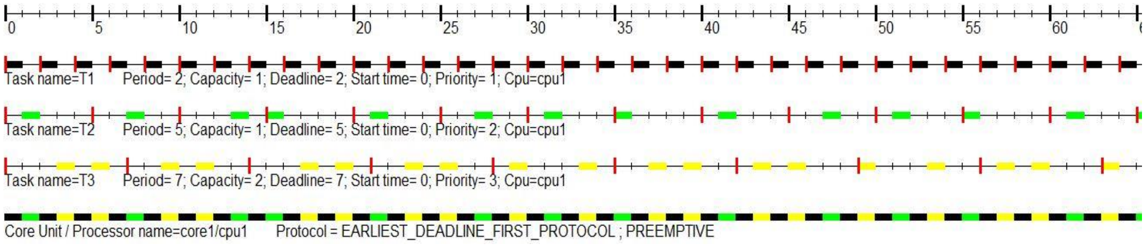
ROTOCOL ; PREEMPTIVE

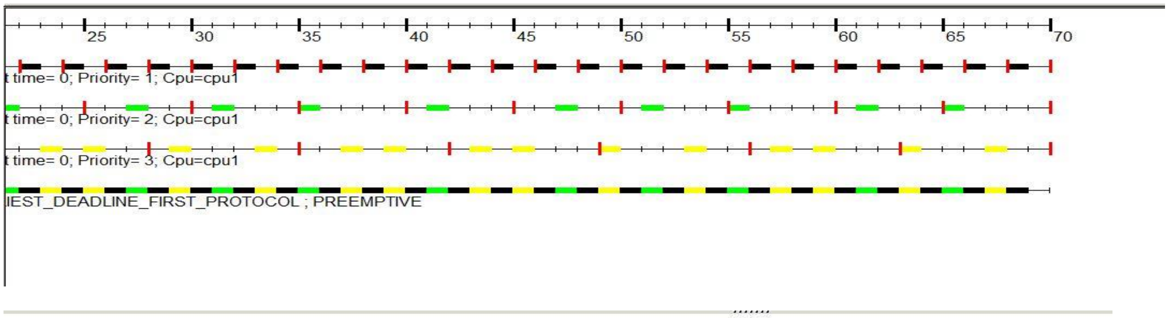


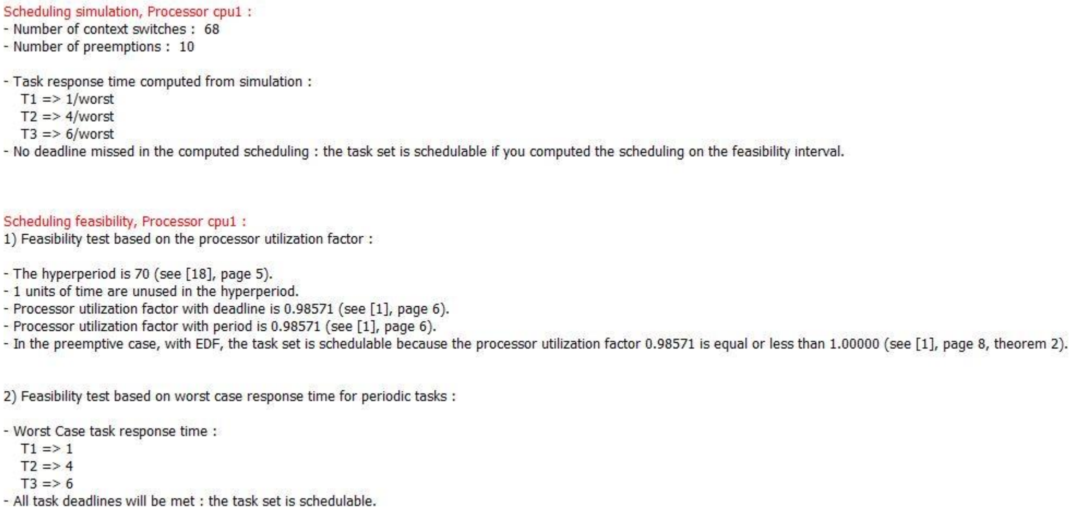
```
Scheduling simulation, Processor cpu1 :
- Number of context switches :  68
- Number of preemptions :  10

- Task response time computed from simulation :
    T1 => 1/worst
    T2 => 2/worst
    T3 => 8/worst ,  missed its deadline (absolute deadline =  7 ; completion time =  8)
- Some task deadlines will be missed : the task set is not schedulable.


Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 70 (see [18], page 5).
- 1 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.98571 (see [1], page 6).
- Processor utilization factor with period is 0.98571 (see [1], page 6).
- In the preemptive case, with RM, we can not prove that the task set is schedulable because the processor utilization factor 0.98571 is more than 0.77976 (see [1], page 16, theorem 8).


2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :  (see [2], page 3, equation 4).
    T3 => 8,  missed its deadline (deadline =  7)
    T2 => 2
    T1 => 1
- Some task deadlines will be missed : the task set is not schedulable.
```

Figure :- 5 RM Diagram & Simulation & Feasibility

EDF :-



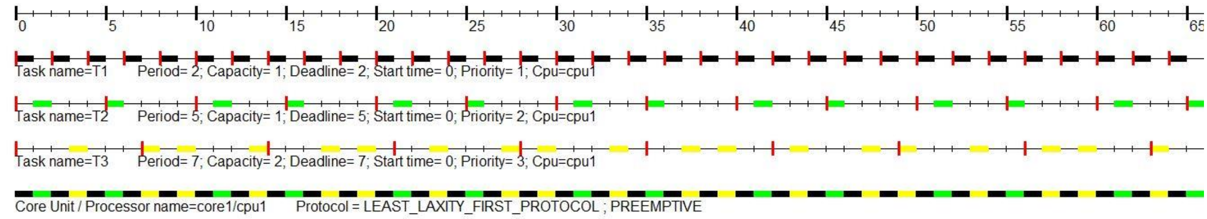


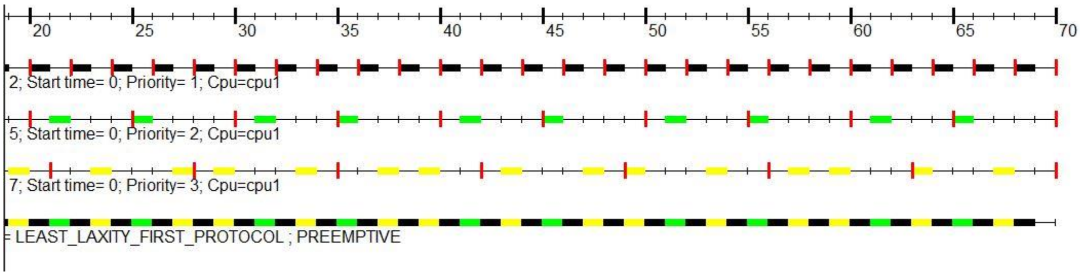


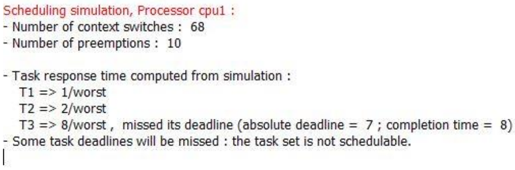
```
Scheduling simulation, Processor cpu1 :
- Number of context switches :  68
- Number of preemptions :  10

- Task response time computed from simulation :
    T1 => 1/worst
    T2 => 4/worst
    T3 => 6/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 70 (see [18], page 5).
- 1 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.98571 (see [1], page 6).
- Processor utilization factor with period is 0.98571 (see [1], page 6).
- In the preemptive case, with EDF, the task set is schedulable because the processor utilization factor 0.98571 is equal or less than 1.00000 (see [1], page 8, theorem 2).


2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :
    T1 => 1
    T2 => 4
    T3 => 6
- All task deadlines will be met : the task set is schedulable.
```

Figure :- 6 EDF Diagram & Simulation & Feasibility

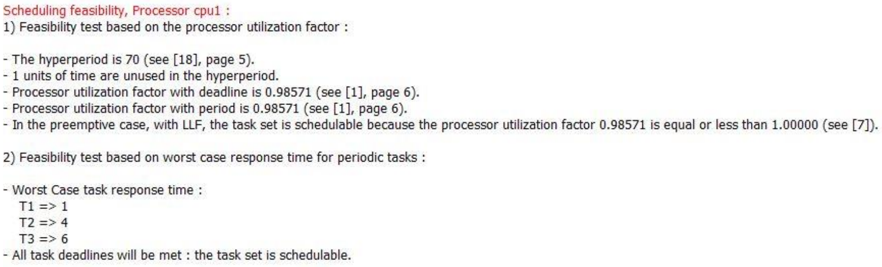
LLF:-







```
Scheduling simulation, Processor cpu1 :
- Number of context switches :  68
- Number of preemptions :  10

- Task response time computed from simulation :
    T1 => 1/worst
    T2 => 2/worst
    T3 => 8/worst ,  missed its deadline (absolute deadline =  7 ; completion time =  8)
- Some task deadlines will be missed : the task set is not schedulable.
```



```
Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 70 (see [18], page 5).
- 1 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.98571 (see [1], page 6).
- Processor utilization factor with period is 0.98571 (see [1], page 6).
- In the preemptive case, with LLF, the task set is schedulable because the processor utilization factor 0.98571 is equal or less than 1.00000 (see [7]).

2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :
    T1 => 1
    T2 => 4
    T3 => 6
- All task deadlines will be met : the task set is schedulable.
```

Figure :- 7 LLF Diagram & Simulation & Feasibility

For this case, **neither cheddar nor the code passes the RM feasibility tests**. The reason for this is service 3 misses its deadline.

There are two types of priorities. Static and Dynamic

Static is the base priority, the one given to the process by the system when the process is created. Without privilege a process can only increase (nice) above the base priority, or reset it to base. a higher number gives lower priority when processes fight for CPU time.

Dynamic is set by the kernel itself. Whenever a process blocks or it has to wait for another process, the dynamic priority is raised. A process that waits a lot therefor gets higher priority than a process that uses a lot of CPU time.

RM uses Static Priority. But in this example **EDF and LLF Passes the feasibility test. Because** EDF and LLF uses the dynamic priority which means EDF is calculated every time as Earliest Deadline and LLF is calculated every time on the basis of Least Laxity.

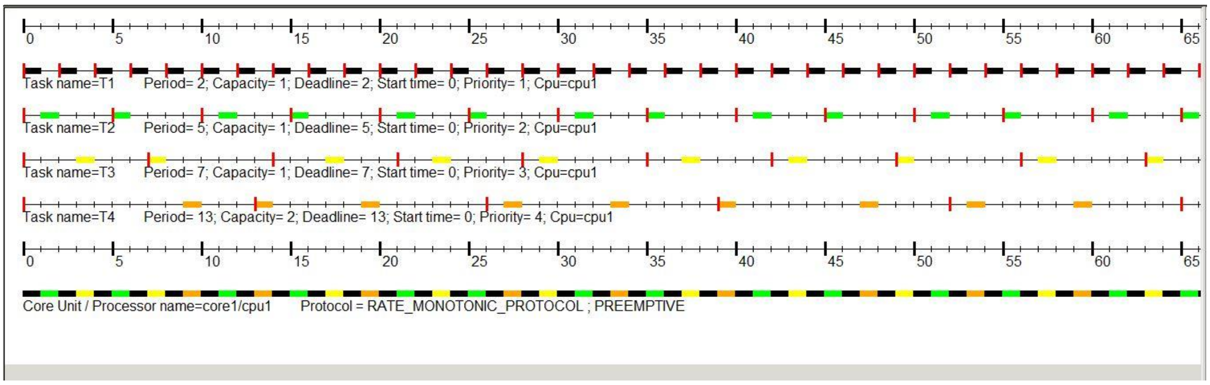
For LLF test, task set is not schedulable because T3 misses its deadline of 7 and completes at 8. It passes the feasibility test based on W_{cet}.

The code matches/agrees with cheddar. The code algorithm is built in such a way that it checks if the scheduling of the services written is feasible or not.

It also Checks if the total utility is <=1.

Example 2:-

RM:-



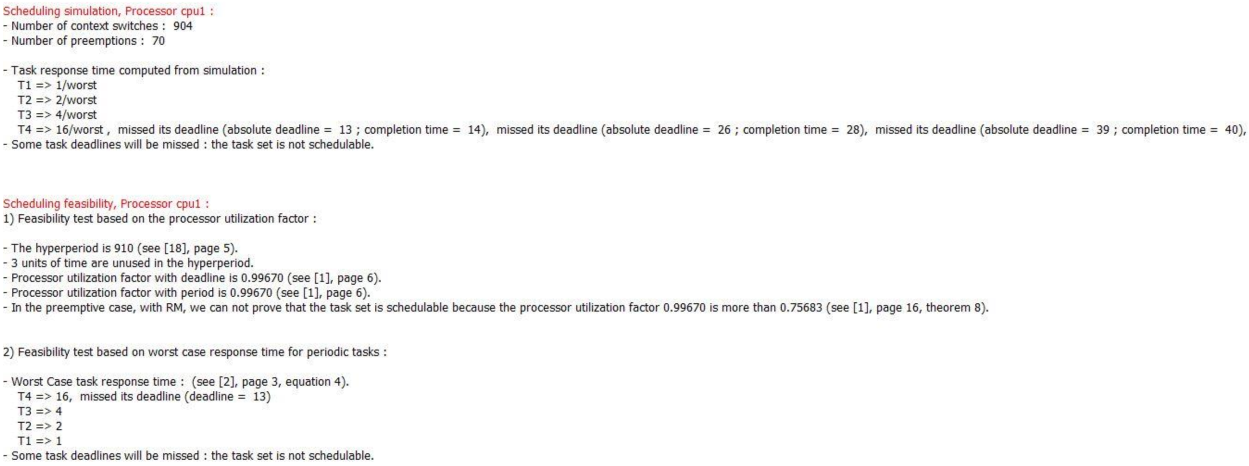
Task name=T1 Period= 2; Capacity= 1; Deadline= 2; Start time= 0; Priority= 1; Cpu=cpu1

Task name=T2 Period= 5; Capacity= 1; Deadline= 5; Start time= 0; Priority= 2; Cpu=cpu1

Task name=T3 Period= 7; Capacity= 1; Deadline= 7; Start time= 0; Priority= 3; Cpu=cpu1

Task name=T4 Period= 13; Capacity= 2; Deadline= 13; Start time= 0; Priority= 4; Cpu=cpu1

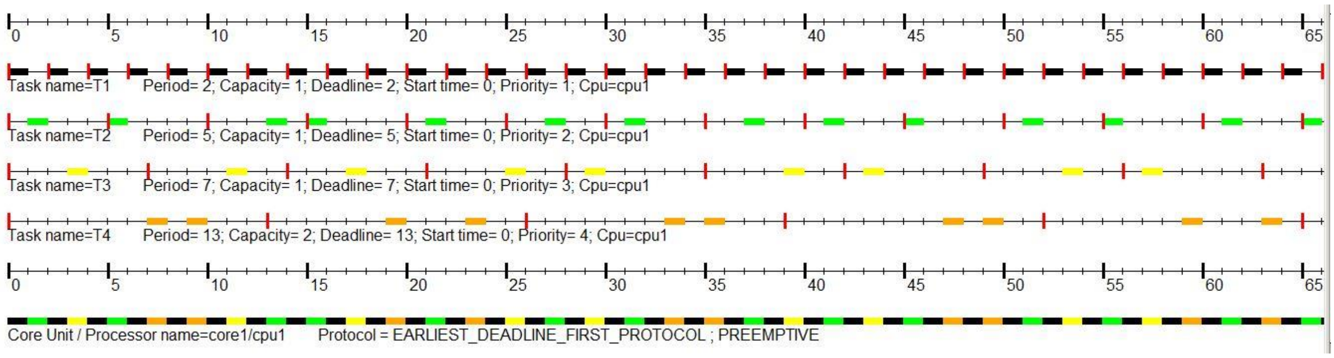
Core Unit / Processor name=core1/cpu1 Protocol = RATE_MONOTONIC_PROTOCOL ; PREEMPTIVE



```
Scheduling simulation, Processor cpu1 :
- Number of context switches :  904
- Number of preemptions :  70

- Task response time computed from simulation :
   T1 => 1/worst
   T2 => 2/worst
   T3 => 4/worst
   T4 => 16/worst ,  missed its deadline (absolute deadline =  13 ; completion time =  14),  missed its deadline (absolute deadline =  26 ; completion time =  28),  missed its deadline (absolute deadline =  39 ; completion time =  40),
- Some task deadlines will be missed : the task set is not schedulable.


Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 910 (see [18], page 5).
- 3 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.99670 (see [1], page 6).
- Processor utilization factor with period is 0.99670 (see [1], page 6).
- In the preemptive case, with RM, we can not prove that the task set is schedulable because the processor utilization factor 0.99670 is more than 0.75683 (see [1], page 16, theorem 8).


2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :  (see [2], page 3, equation 4).
   T4 => 16,  missed its deadline (deadline =  13)
   T3 => 4
   T2 => 2
   T1 => 1
- Some task deadlines will be missed : the task set is not schedulable.
```

Figure :-8 RM Diagram & Simulation & Feasibility

EDF:-



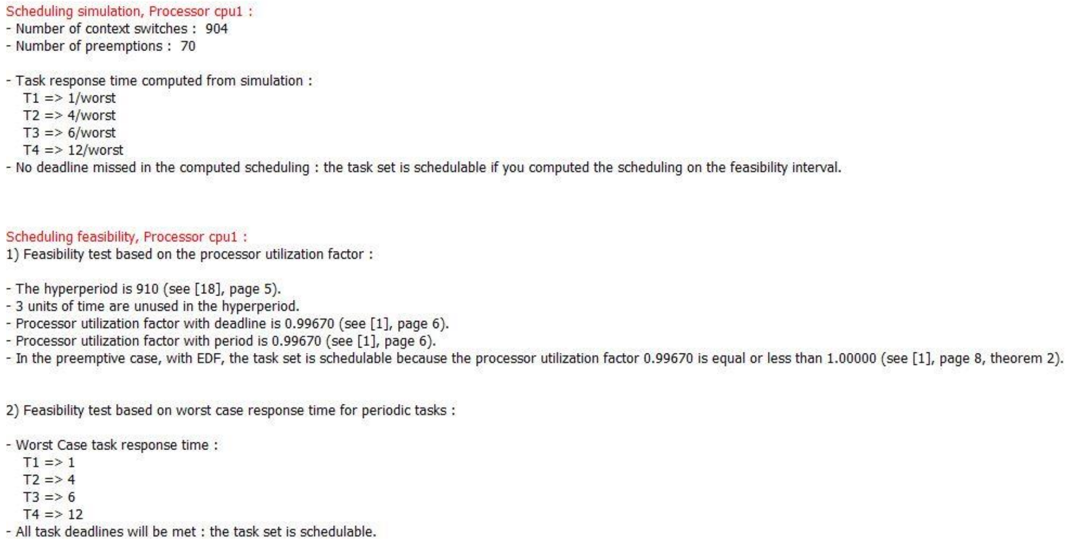
Task name=T1 Period= 2; Capacity= 1; Deadline= 2; Start time= 0; Priority= 1; Cpu=cpu1

Task name=T2 Period= 5; Capacity= 1; Deadline= 5; Start time= 0; Priority= 2; Cpu=cpu1

Task name=T3 Period= 7; Capacity= 1; Deadline= 7; Start time= 0; Priority= 3; Cpu=cpu1

Task name=T4 Period= 13; Capacity= 2; Deadline= 13; Start time= 0; Priority= 4; Cpu=cpu1

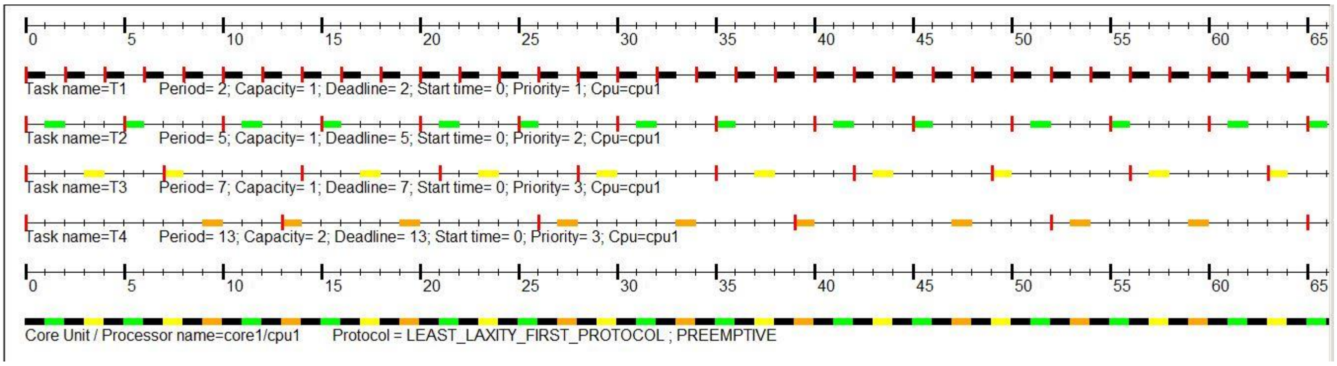
Core Unit / Processor name=core1/cpu1 Protocol = EARLIEST_DEADLINE_FIRST_PROTOCOL ; PREEMPTIVE



```
Scheduling simulation, Processor cpu1 :
- Number of context switches :  904
- Number of preemptions :  70

- Task response time computed from simulation :
   T1 => 1/worst
   T2 => 4/worst
   T3 => 6/worst
   T4 => 12/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 910 (see [18], page 5).
- 3 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.99670 (see [1], page 6).
- Processor utilization factor with period is 0.99670 (see [1], page 6).
- In the preemptive case, with EDF, the task set is schedulable because the processor utilization factor 0.99670 is equal or less than 1.00000 (see [1], page 8, theorem 2).


2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :
   T1 => 1
   T2 => 4
   T3 => 6
   T4 => 12
- All task deadlines will be met : the task set is schedulable.
```

Figure :- 9 EDF Diagram & Simulation & Feasibility

LLF:-



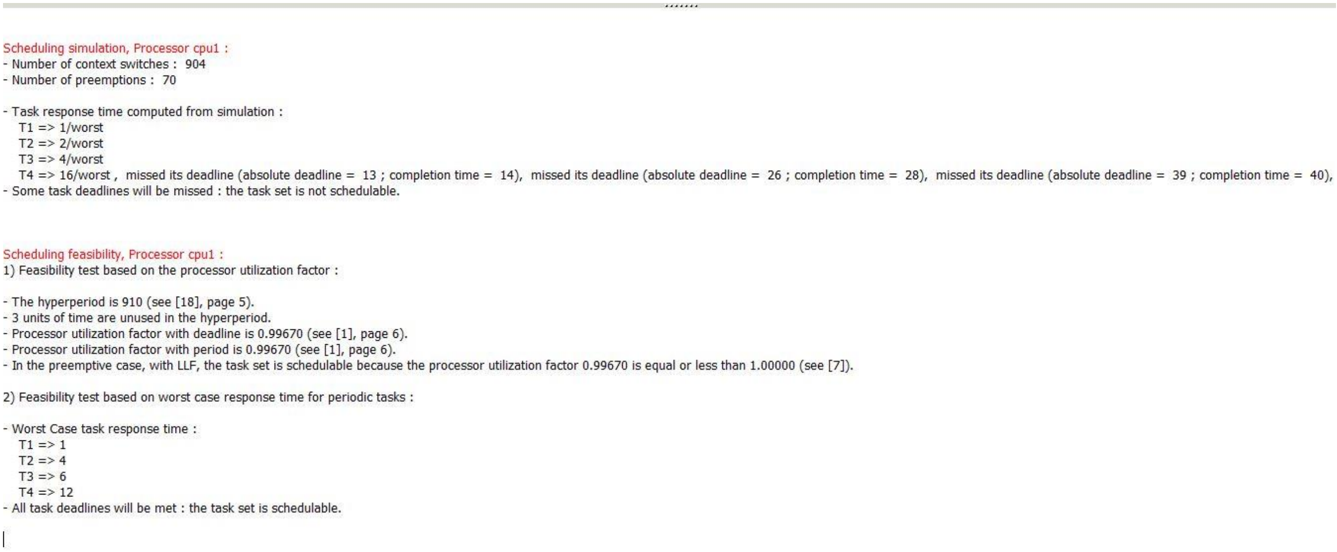
Task name=T1 Period= 2; Capacity= 1; Deadline= 2; Start time= 0; Priority= 1; Cpu=cpu1

Task name=T2 Period= 5; Capacity= 1; Deadline= 5; Start time= 0; Priority= 2; Cpu=cpu1

Task name=T3 Period= 7; Capacity= 1; Deadline= 7; Start time= 0; Priority= 3; Cpu=cpu1

Task name=T4 Period= 13; Capacity= 2; Deadline= 13; Start time= 0; Priority= 3; Cpu=cpu1

Core Unit / Processor name=core1/cpu1 Protocol = LEAST_LAXITY_FIRST_PROTOCOL ; PREEMPTIVE



```
Scheduling simulation, Processor cpu1 :
- Number of context switches :  904
- Number of preemptions :  70

- Task response time computed from simulation :
   T1 => 1/worst
   T2 => 2/worst
   T3 => 4/worst
   T4 => 16/worst ,  missed its deadline (absolute deadline =  13 ; completion time =  14),  missed its deadline (absolute deadline =  26 ; completion time =  28),  missed its deadline (absolute deadline =  39 ; completion time =  40),
- Some task deadlines will be missed : the task set is not schedulable.


Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 910 (see [18], page 5).
- 3 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.99670 (see [1], page 6).
- Processor utilization factor with period is 0.99670 (see [1], page 6).
- In the preemptive case, with LLF, the task set is schedulable because the processor utilization factor 0.99670 is equal or less than 1.00000 (see [7]).

2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :
   T1 => 1
   T2 => 4
   T3 => 6
   T4 => 12
- All task deadlines will be met : the task set is schedulable.
```

Figure :- 10 LLF Diagram & Simulation & Feasibility

For this case, **neither cheddar nor the code passes the RM feasibility tests**. The reason for this is service 4 misses its deadline.

There are two types of priorities. Static and Dynamic

Static is the base priority, the one given to the process by the system when the process is created. Without privilege a process can only increase (nice) above the base priority, or reset it to base. a higher number gives lower priority when processes fight for CPU time.

Dynamic is set by the kernel itself. Whenever a process blocks or it has to wait for another process, the dynamic priority is raised. A process that waits a lot therefor gets higher priority than a process that uses a lot of CPU time.

RM uses Static Priority. But in this example **EDF and LLF Passes the feasibility test. Because** EDF and LLF uses the dynamic priority which means EDF is calculated every time as Earliest Deadline and LLF is calculated every time on the basis of Least Laxity.

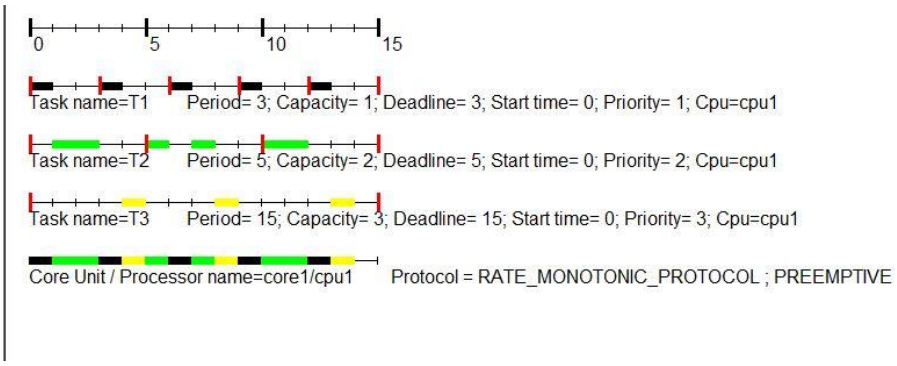
For LLF test, task set is not schedulable because T4 misses its deadline of 13 and completes at 16. It passes the feasibility test based on W_{cet}.

The code matches/agrees with cheddar. The code algorithm is built in such a way that it checks if the scheduling of the services written is feasible or not.

It also Checks if the total utility is <=1.

Example :- 3

RM:-

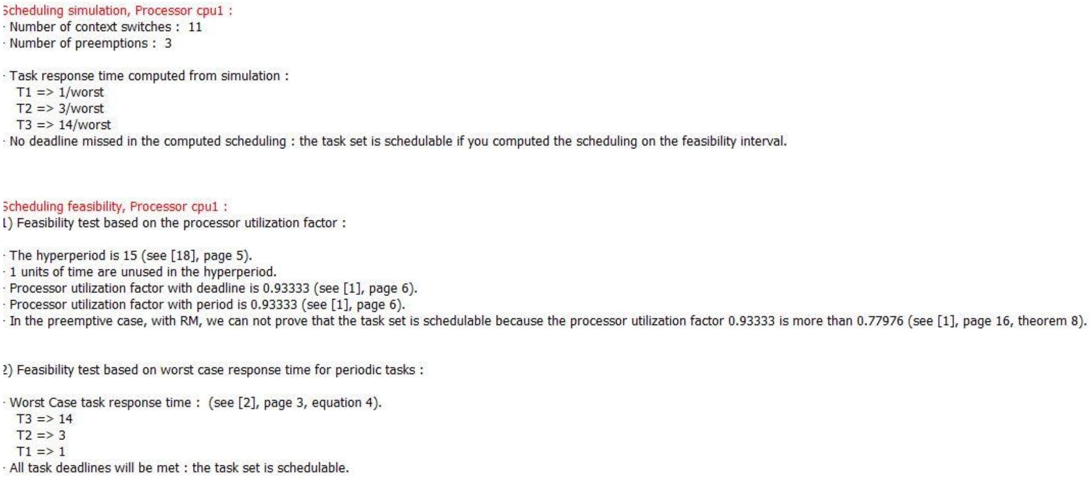


Task name=T1 Period= 3; Capacity= 1; Deadline= 3; Start time= 0; Priority= 1; Cpu=cpu1

Task name=T2 Period= 5; Capacity= 2; Deadline= 5; Start time= 0; Priority= 2; Cpu=cpu1

Task name=T3 Period= 15; Capacity= 3; Deadline= 15; Start time= 0; Priority= 3; Cpu=cpu1

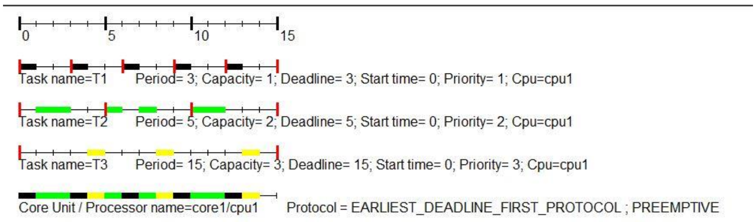
Core Unit / Processor name=core1/cpu1 Protocol = RATE_MONOTONIC_PROTOCOL ; PREEMPTIVE



```
Scheduling simulation, Processor cpu1 :
- Number of context switches :  11
- Number of preemptions :  3

- Task response time computed from simulation :
   T1 => 1/worst
   T2 => 3/worst
   T3 => 14/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 15 (see [18], page 5).
- 1 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.93333 (see [1], page 6).
- Processor utilization factor with period is 0.93333 (see [1], page 6).
- In the preemptive case, with RM, we can not prove that the task set is schedulable because the processor utilization factor 0.93333 is more than 0.77976 (see [1], page 16, theorem 8).


2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :  (see [2], page 3, equation 4).
   T3 => 14
   T2 => 3
   T1 => 1
- All task deadlines will be met : the task set is schedulable.
```

Figure :- 11 RM Diagram & Simulation & Feasibility

EDF:-

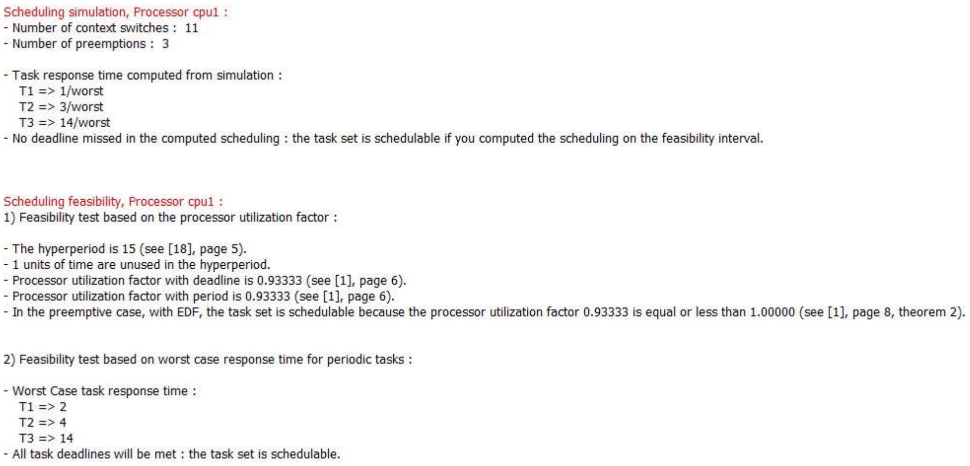


Task name=T1 Period= 3; Capacity= 1; Deadline= 3; Start time= 0; Priority= 1; Cpu=cpu1

Task name=T2 Period= 5; Capacity= 2; Deadline= 5; Start time= 0; Priority= 2; Cpu=cpu1

Task name=T3 Period= 15; Capacity= 3; Deadline= 15; Start time= 0; Priority= 3; Cpu=cpu1

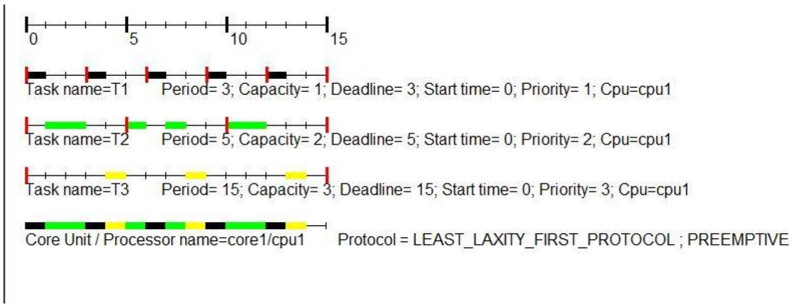
Core Unit / Processor name=core1/cpu1 Protocol = EARLIEST_DEADLINE_FIRST_PROTOCOL ; PREEMPTIVE



```
Scheduling simulation, Processor cpu1 :
- Number of context switches :  11
- Number of preemptions :  3

- Task response time computed from simulation :
   T1 => 1/worst
   T2 => 3/worst
   T3 => 14/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 15 (see [18], page 5).
- 1 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.93333 (see [1], page 6).
- Processor utilization factor with period is 0.93333 (see [1], page 6).
- In the preemptive case, with EDF, the task set is schedulable because the processor utilization factor 0.93333 is equal or less than 1.00000 (see [1], page 8, theorem 2).


2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :
   T1 => 2
   T2 => 4
   T3 => 14
- All task deadlines will be met : the task set is schedulable.
```

Figure :- 12 EDF Diagram & Simulation & Feasibility

LLF:-

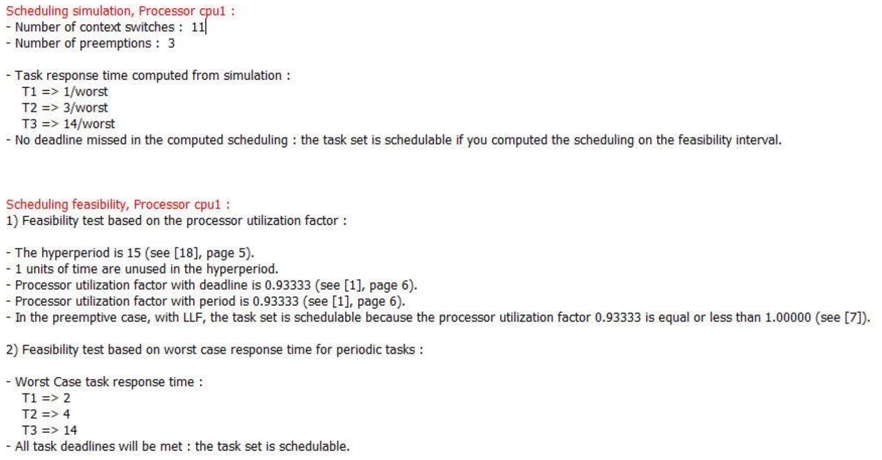


Task name=T1 Period= 3; Capacity= 1; Deadline= 3; Start time= 0; Priority= 1; Cpu=cpu1

Task name=T2 Period= 5; Capacity= 2; Deadline= 5; Start time= 0; Priority= 2; Cpu=cpu1

Task name=T3 Period= 15; Capacity= 3; Deadline= 15; Start time= 0; Priority= 3; Cpu=cpu1

Core Unit / Processor name=core1/cpu1 Protocol = LEAST_LAXITY_FIRST_PROTOCOL ; PREEMPTIVE



```
Scheduling simulation, Processor cpu1 :
- Number of context switches :  11
- Number of preemptions :  3

- Task response time computed from simulation :
    T1 => 1/worst
    T2 => 3/worst
    T3 => 14/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 15 (see [18], page 5).
- 1 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.93333 (see [1], page 6).
- Processor utilization factor with period is 0.93333 (see [1], page 6).
- In the preemptive case, with LLF, the task set is schedulable because the processor utilization factor 0.93333 is equal or less than 1.00000 (see [7]).

2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :
    T1 => 2
    T2 => 4
    T3 => 14
- All task deadlines will be met : the task set is schedulable.
```

Figure :- 13 LLF Diagram & Simulation & Feasibility

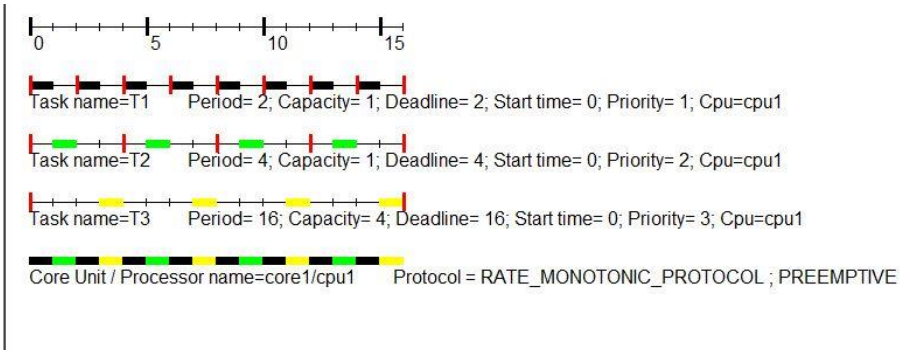
The Cheddar and the code both passes RM,EDF,LLF Feasibility tests.

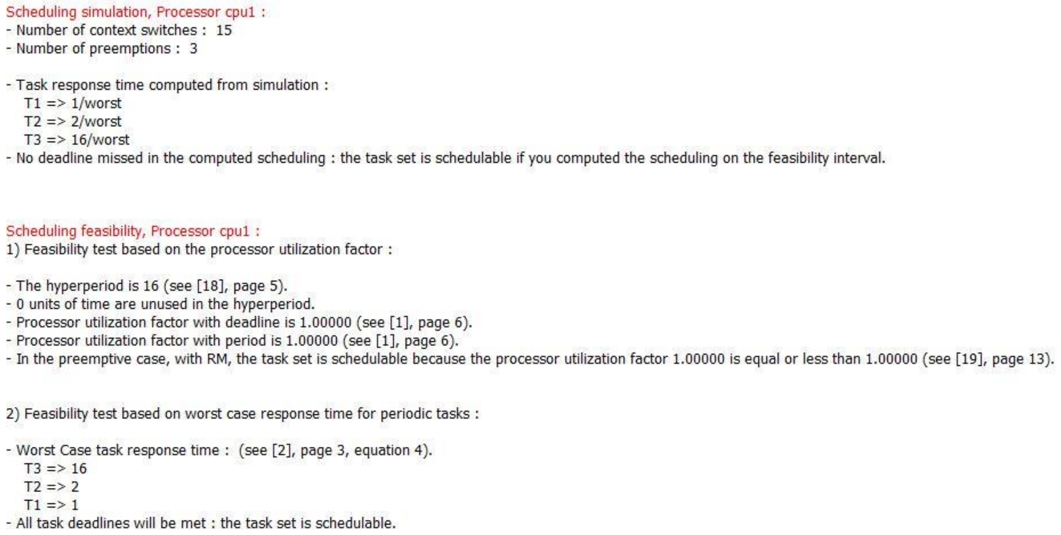
The code matches/agrees with cheddar. The code algorithm is built in such a way that it checks if the scheduling of the services written is feasible or not.

It also Checks if the total utility is <=1.

Example :- 4

RM:-

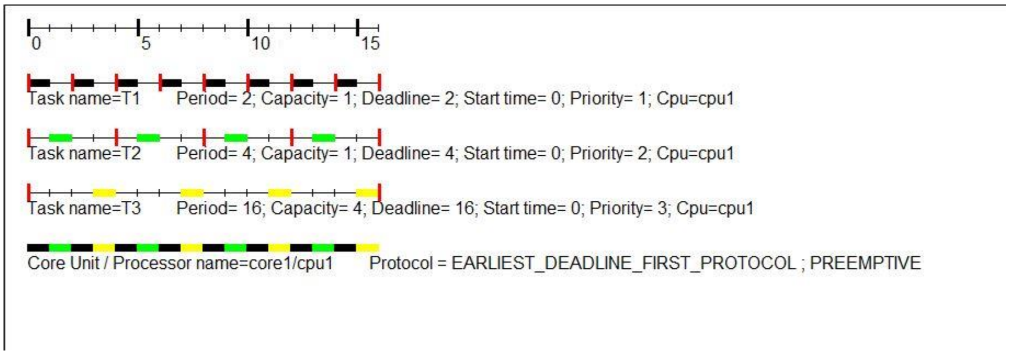




```
Scheduling simulation, Processor cpu1 :
- Number of context switches :  15
- Number of preemptions :  3

- Task response time computed from simulation :
    T1 => 1/worst
    T2 => 2/worst
    T3 => 16/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 16 (see [18], page 5).
- 0 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 1.00000 (see [1], page 6).
- Processor utilization factor with period is 1.00000 (see [1], page 6).
- In the preemptive case, with RM, the task set is schedulable because the processor utilization factor 1.00000 is equal or less than 1.00000 (see [19], page 13).


2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :  (see [2], page 3, equation 4).
    T3 => 16
    T2 => 2
    T1 => 1
- All task deadlines will be met : the task set is schedulable.
```

Figure :- 14 RM Diagram & Simulation & Feasibility

EDF:-

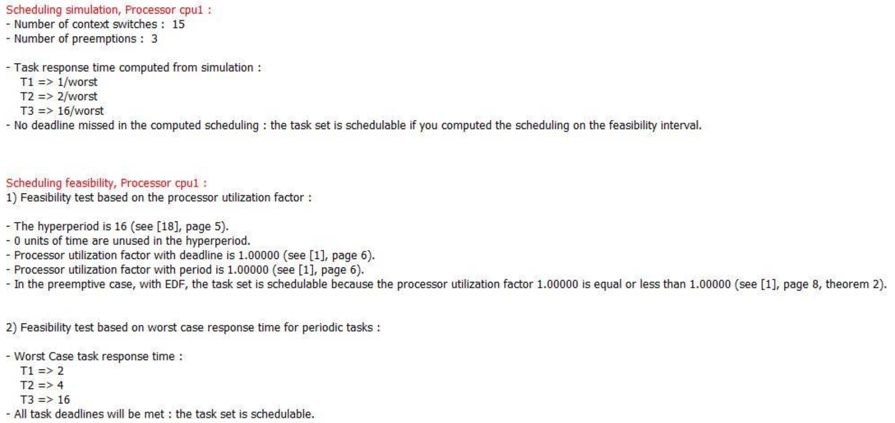


Task name=T1 Period= 2; Capacity= 1; Deadline= 2; Start time= 0; Priority= 1; Cpu=cpu1

Task name=T2 Period= 4; Capacity= 1; Deadline= 4; Start time= 0; Priority= 2; Cpu=cpu1

Task name=T3 Period= 16; Capacity= 4; Deadline= 16; Start time= 0; Priority= 3; Cpu=cpu1

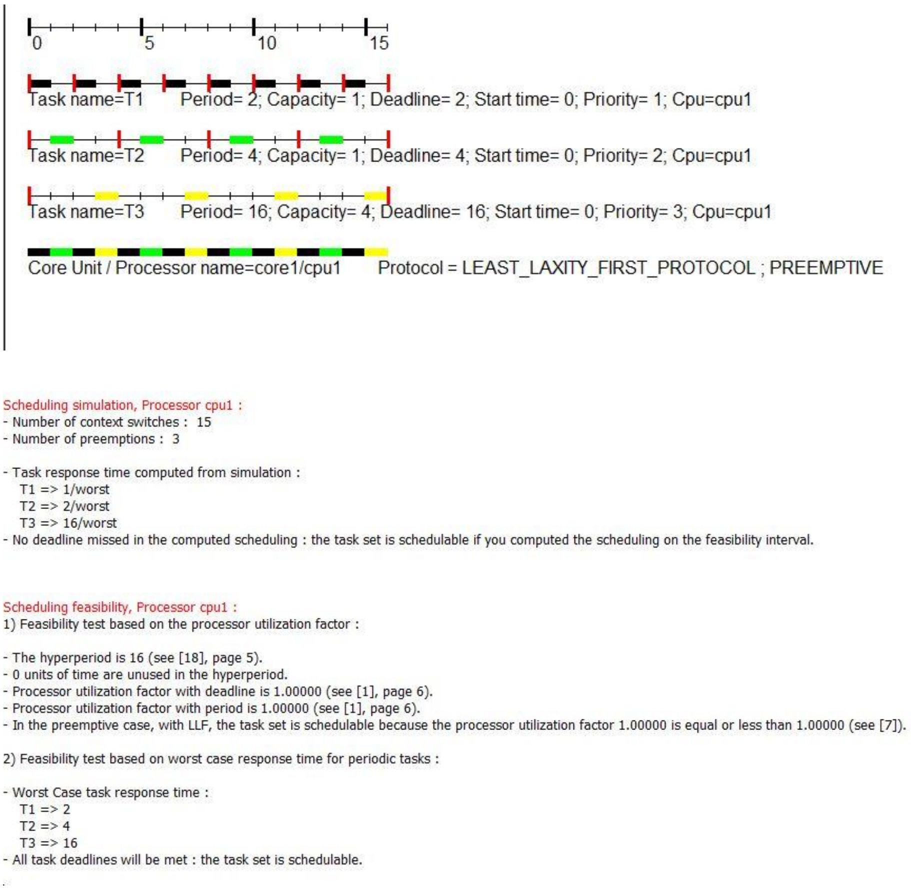
Core Unit / Processor name=core1/cpu1 Protocol = EARLIEST_DEADLINE_FIRST_PROTOCOL ; PREEMPTIVE



```
Scheduling simulation, Processor cpu1 :
- Number of context switches :  15
- Number of preemptions :  3

- Task response time computed from simulation :
    T1 => 1/worst
    T2 => 2/worst
    T3 => 16/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 16 (see [18], page 5).
- 0 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 1.00000 (see [1], page 6).
- Processor utilization factor with period is 1.00000 (see [1], page 6).
- In the preemptive case, with EDF, the task set is schedulable because the processor utilization factor 1.00000 is equal or less than 1.00000 (see [1], page 8, theorem 2).


2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :
    T1 => 2
    T2 => 4
    T3 => 16
- All task deadlines will be met : the task set is schedulable.
```

Figure :- 15 EDF Diagram & Simulation & Feasibility

LLF:-


0
5
10
15
Task name=T1 Period= 2; Capacity= 1; Deadline= 2; Start time= 0; Priority= 1; Cpu=cpu1
Task name=T2 Period= 4; Capacity= 1; Deadline= 4; Start time= 0; Priority= 2; Cpu=cpu1
Task name=T3 Period= 16; Capacity= 4; Deadline= 16; Start time= 0; Priority= 3; Cpu=cpu1
Core Unit / Processor name=core1/cpu1 Protocol = LEAST_LAXITY_FIRST_PROTOCOL ; PREEMPTIVE

```
Scheduling simulation, Processor cpu1 :
- Number of context switches :  15
- Number of preemptions :  3

- Task response time computed from simulation :
    T1 => 1/worst
    T2 => 2/worst
    T3 => 16/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 16 (see [18], page 5).
- 0 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 1.00000 (see [1], page 6).
- Processor utilization factor with period is 1.00000 (see [1], page 6).
- In the preemptive case, with LLF, the task set is schedulable because the processor utilization factor 1.00000 is equal or less than 1.00000 (see [7]).

2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :
    T1 => 2
    T2 => 4
    T3 => 16
- All task deadlines will be met : the task set is schedulable.
```

Figure :- 16 LLF Diagram & Simulation & Feasibility

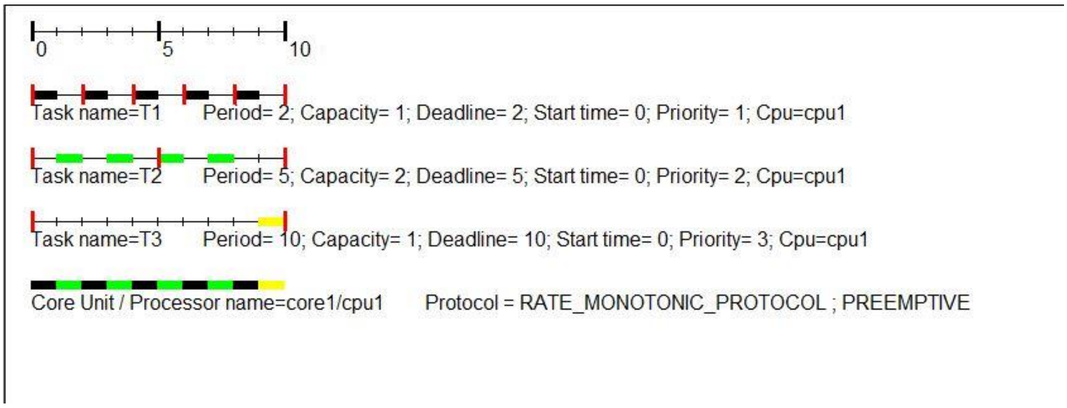
The Cheddar and the code both passes RM,EDF,LLF Feasibility tests.

The code matches/agrees with cheddar. The code algorithm is built in such a way that it checks if the scheduling of the services written is feasible or not.

It also Checks if the total utility is <=1.

Example :- 5

RM:-

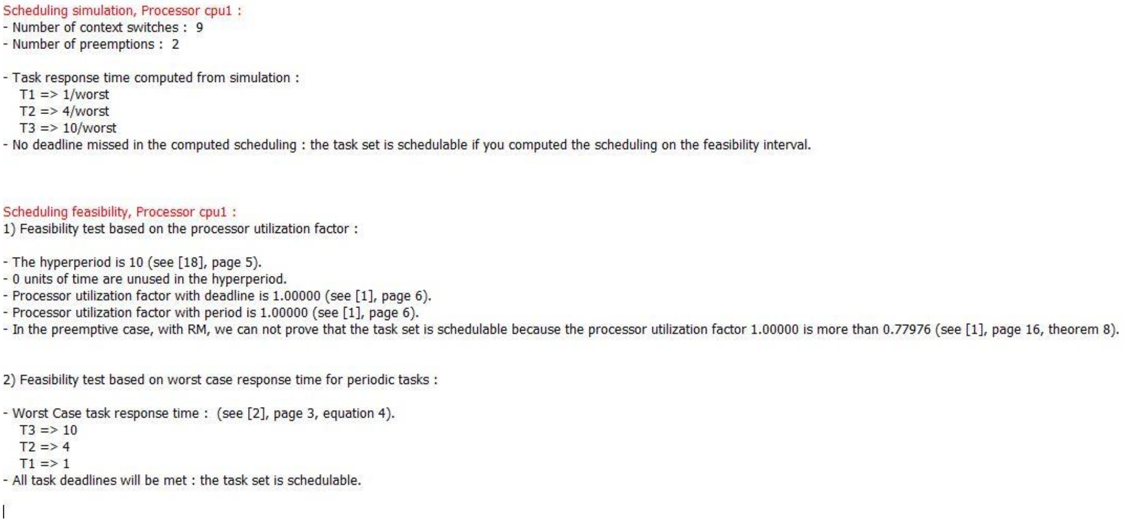


Task name=T1 Period= 2; Capacity= 1; Deadline= 2; Start time= 0; Priority= 1; Cpu=cpu1

Task name=T2 Period= 5; Capacity= 2; Deadline= 5; Start time= 0; Priority= 2; Cpu=cpu1

Task name=T3 Period= 10; Capacity= 1; Deadline= 10; Start time= 0; Priority= 3; Cpu=cpu1

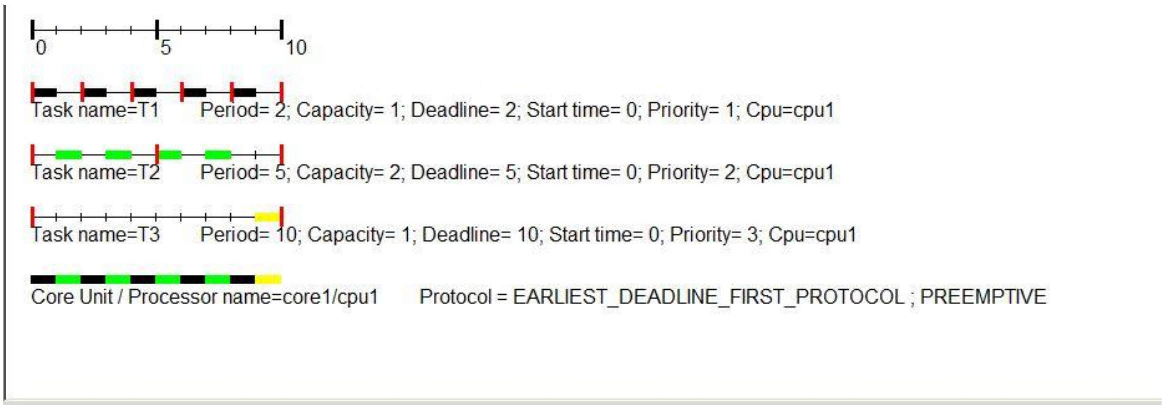
Core Unit / Processor name=core1/cpu1 Protocol = RATE_MONOTONIC_PROTOCOL ; PREEMPTIVE



```
Scheduling simulation, Processor cpu1 :
- Number of context switches :  9
- Number of preemptions :  2

- Task response time computed from simulation :
    T1 => 1/worst
    T2 => 4/worst
    T3 => 10/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 10 (see [18], page 5).
- 0 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 1.00000 (see [1], page 6).
- Processor utilization factor with period is 1.00000 (see [1], page 6).
- In the preemptive case, with RM, we can not prove that the task set is schedulable because the processor utilization factor 1.00000 is more than 0.77976 (see [1], page 16, theorem 8).


2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :  (see [2], page 3, equation 4).
    T3 => 10
    T2 => 4
    T1 => 1
- All task deadlines will be met : the task set is schedulable.
```

Figure :- 17 RM Diagram & Simulation & Feasibility

EDF:-

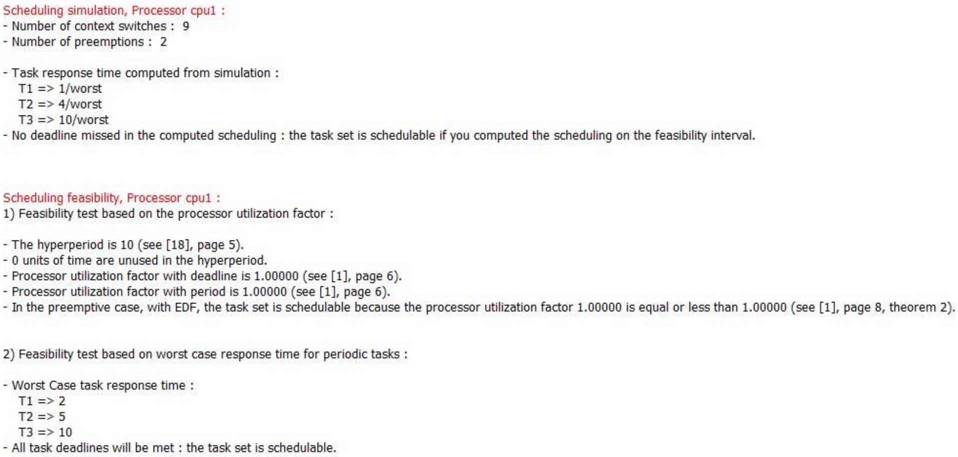


Task name=T1 Period= 2; Capacity= 1; Deadline= 2; Start time= 0; Priority= 1; Cpu=cpu1

Task name=T2 Period= 5; Capacity= 2; Deadline= 5; Start time= 0; Priority= 2; Cpu=cpu1

Task name=T3 Period= 10; Capacity= 1; Deadline= 10; Start time= 0; Priority= 3; Cpu=cpu1

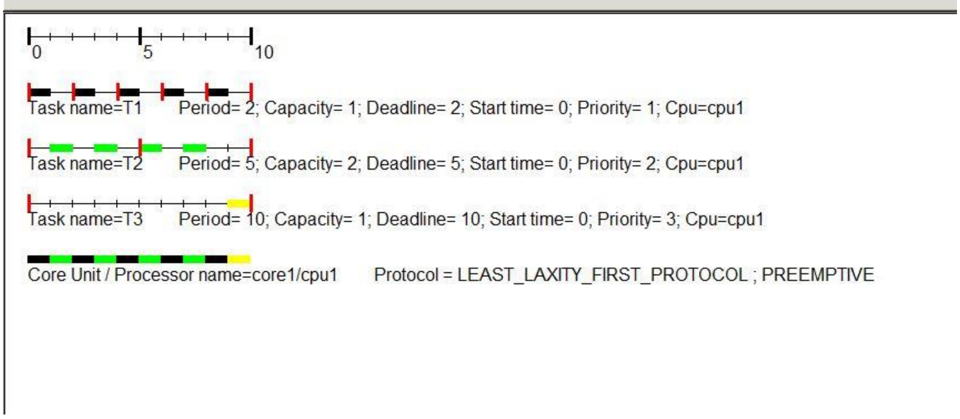
Core Unit / Processor name=core1/cpu1 Protocol = EARLIEST_DEADLINE_FIRST_PROTOCOL ; PREEMPTIVE



```
Scheduling simulation, Processor cpu1 :
- Number of context switches :  9
- Number of preemptions :  2

- Task response time computed from simulation :
    T1 => 1/worst
    T2 => 4/worst
    T3 => 10/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 10 (see [18], page 5).
- 0 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 1.00000 (see [1], page 6).
- Processor utilization factor with period is 1.00000 (see [1], page 6).
- In the preemptive case, with EDF, the task set is schedulable because the processor utilization factor 1.00000 is equal or less than 1.00000 (see [1], page 8, theorem 2).


2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :
    T1 => 2
    T2 => 5
    T3 => 10
- All task deadlines will be met : the task set is schedulable.
```

Figure :- 18 EDF Diagram & Simulation & Feasibility

LLF:-

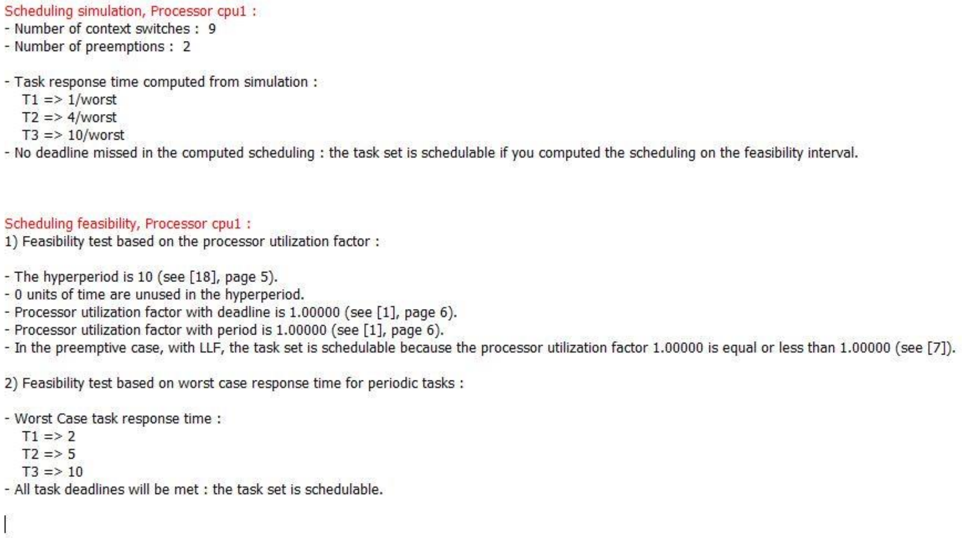


Task name=T1 Period= 2; Capacity= 1; Deadline= 2; Start time= 0; Priority= 1; Cpu=cpu1

Task name=T2 Period= 5; Capacity= 2; Deadline= 5; Start time= 0; Priority= 2; Cpu=cpu1

Task name=T3 Period= 10; Capacity= 1; Deadline= 10; Start time= 0; Priority= 3; Cpu=cpu1

Core Unit / Processor name=core1/cpu1 Protocol = LEAST_LAXITY_FIRST_PROTOCOL ; PREEMPTIVE



```
Scheduling simulation, Processor cpu1 :
- Number of context switches :  9
- Number of preemptions :  2

- Task response time computed from simulation :
   T1 => 1/worst
   T2 => 4/worst
   T3 => 10/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 10 (see [18], page 5).
- 0 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 1.00000 (see [1], page 6).
- Processor utilization factor with period is 1.00000 (see [1], page 6).
- In the preemptive case, with LLF, the task set is schedulable because the processor utilization factor 1.00000 is equal or less than 1.00000 (see [7]).

2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :
   T1 => 2
   T2 => 5
   T3 => 10
- All task deadlines will be met : the task set is schedulable.
```

Figure :- 19 LLF Diagram & Simulation & Feasibility

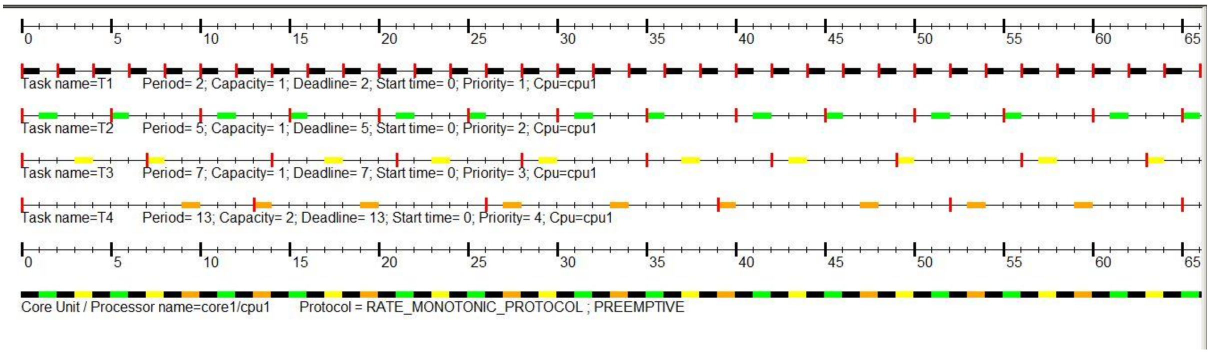
The Cheddar and the code both passes RM,EDF,LLF Feasibility tests.

The code matches/agrees with cheddar. The code algorithm is built in such a way that it checks if the scheduling of the services written is feasible or not.

It also Checks if the total utility is <=1.

Example :6

RM :-



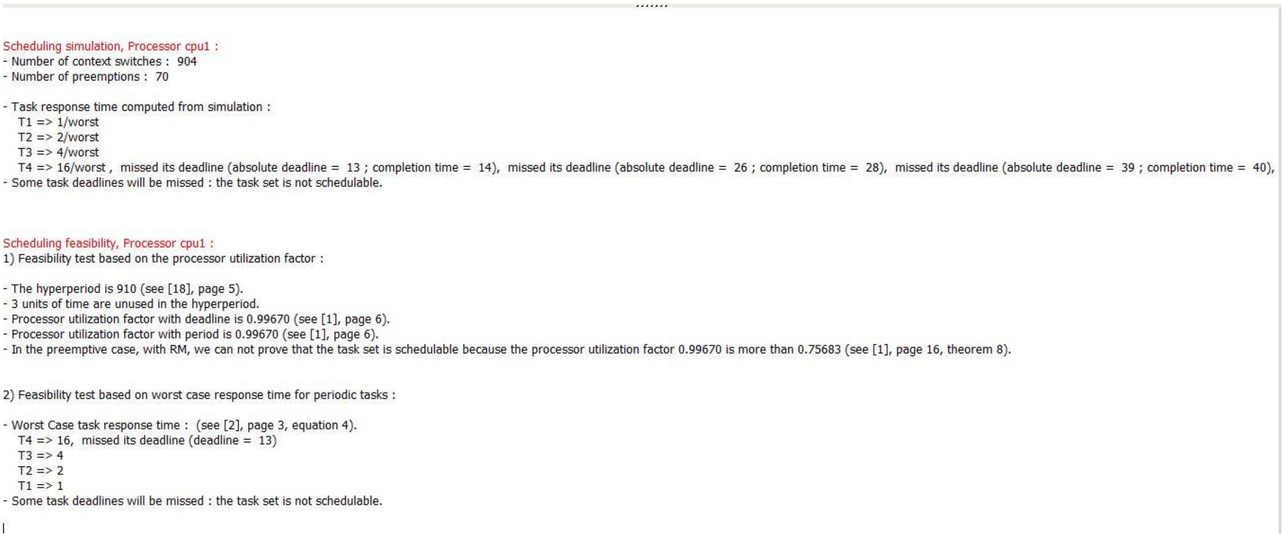
Task name=T1   Period= 2; Capacity= 1; Deadline= 2; Start time= 0; Priority= 1; Cpu=cpu1

Task name=T2   Period= 5; Capacity= 1; Deadline= 5; Start time= 0; Priority= 2; Cpu=cpu1

Task name=T3   Period= 7; Capacity= 1; Deadline= 7; Start time= 0; Priority= 3; Cpu=cpu1

Task name=T4   Period= 13; Capacity= 2; Deadline= 13; Start time= 0; Priority= 4; Cpu=cpu1

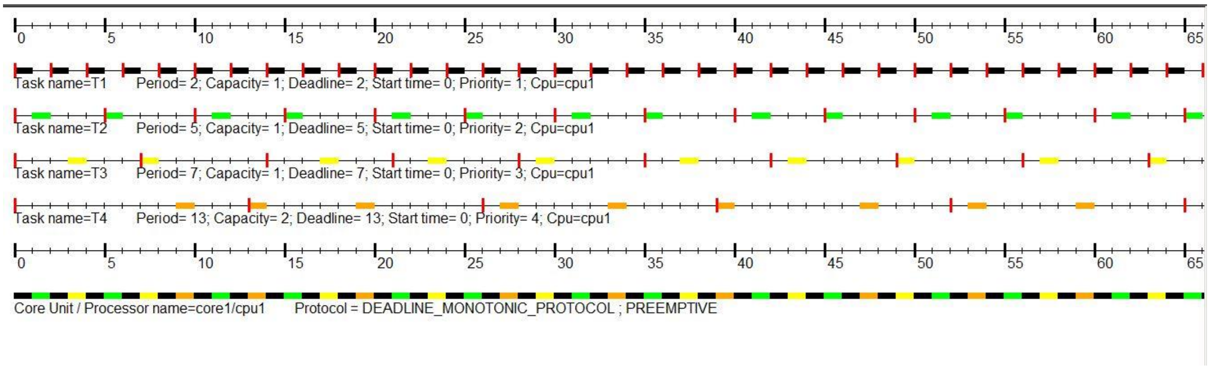
Core Unit / Processor name=core1/cpu1   Protocol = RATE_MONOTONIC_PROTOCOL ; PREEMPTIVE



```
Scheduling simulation, Processor cpu1 :
- Number of context switches :  904
- Number of preemptions :  70

- Task response time computed from simulation :
    T1 => 1/worst
    T2 => 2/worst
    T3 => 4/worst
    T4 => 16/worst ,  missed its deadline (absolute deadline =  13 ; completion time =  14),  missed its deadline (absolute deadline =  26 ; completion time =  28),  missed its deadline (absolute deadline =  39 ; completion time =  40),
- Some task deadlines will be missed : the task set is not schedulable.


Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 910 (see [18], page 5).
- 3 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.99670 (see [1], page 6).
- Processor utilization factor with period is 0.99670 (see [1], page 6).
- In the preemptive case, with RM, we can not prove that the task set is schedulable because the processor utilization factor 0.99670 is more than 0.75683 (see [1], page 16, theorem 8).


2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :  (see [2], page 3, equation 4).
    T4 => 16,  missed its deadline (deadline =  13)
    T3 => 4
    T2 => 2
    T1 => 1
- Some task deadlines will be missed : the task set is not schedulable.
```

Figure :- 20 RM Diagram & Simulation & Feasibility

DM :-



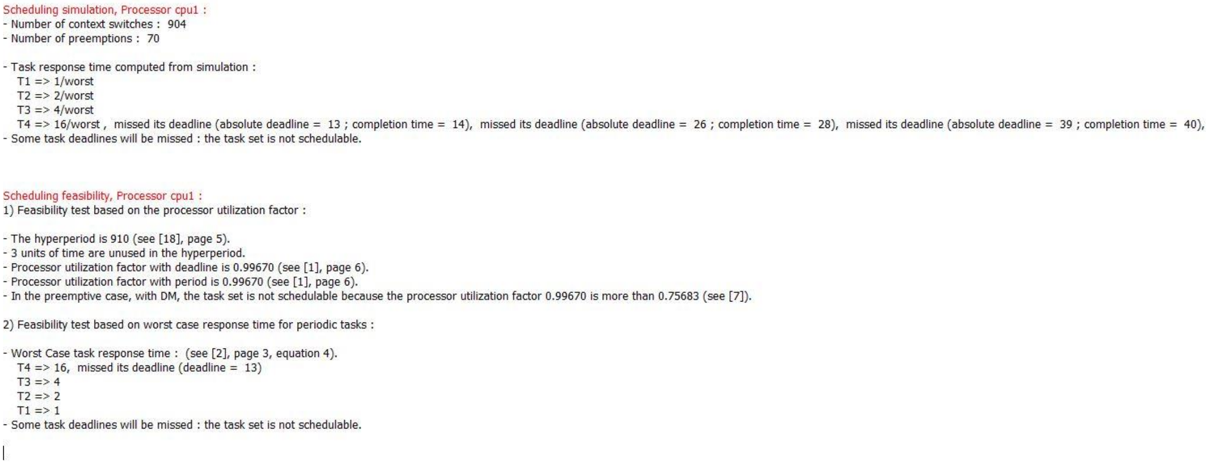
Task name=T1 Period= 2; Capacity= 1; Deadline= 2; Start time= 0; Priority= 1; Cpu=cpu1

Task name=T2 Period= 5; Capacity= 1; Deadline= 5; Start time= 0; Priority= 2; Cpu=cpu1

Task name=T3 Period= 7; Capacity= 1; Deadline= 7; Start time= 0; Priority= 3; Cpu=cpu1

Task name=T4 Period= 13; Capacity= 2; Deadline= 13; Start time= 0; Priority= 4; Cpu=cpu1

Core Unit / Processor name=core1/cpu1 Protocol = DEADLINE_MONOTONIC_PROTOCOL ; PREEMPTIVE



```
Scheduling simulation, Processor cpu1 :
- Number of context switches :  904
- Number of preemptions :  70

- Task response time computed from simulation :
    T1 => 1/worst
    T2 => 2/worst
    T3 => 4/worst
    T4 => 16/worst ,  missed its deadline (absolute deadline =  13 ; completion time =  14),  missed its deadline (absolute deadline =  26 ; completion time =  28),  missed its deadline (absolute deadline =  39 ; completion time =  40),
- Some task deadlines will be missed : the task set is not schedulable.


Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 910 (see [18], page 5).
- 3 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.99670 (see [1], page 6).
- Processor utilization factor with period is 0.99670 (see [1], page 6).
- In the preemptive case, with DM, the task set is not schedulable because the processor utilization factor 0.99670 is more than 0.75683 (see [7]).

2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :  (see [2], page 3, equation 4).
    T4 => 16,  missed its deadline (deadline =  13)
    T3 => 4
    T2 => 2
    T1 => 1
- Some task deadlines will be missed : the task set is not schedulable.
```

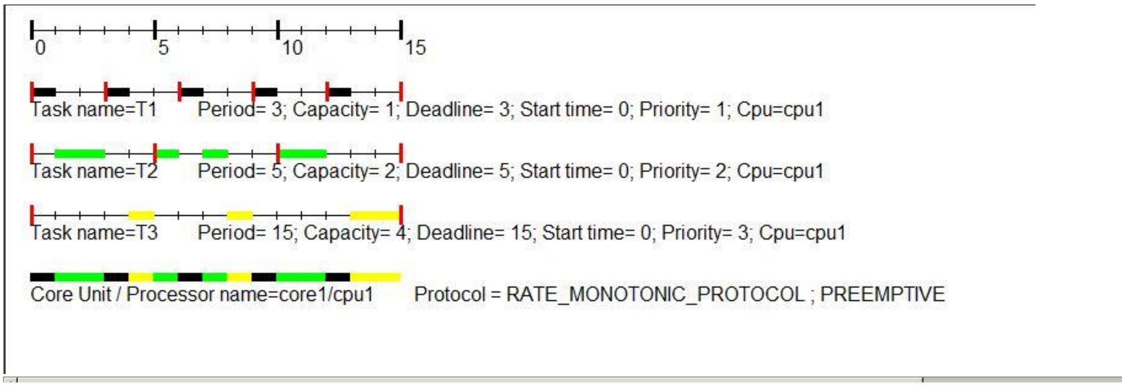
Figure :- 21 DM Diagram & Simulation & Feasibility

Here, neither cheddar nor code passes the RM and DM Feasibility tests.

Moreover, neither cheddar nor code passes the scheduling point test thus we can say that it is infeasible.

Example :7

RM:-

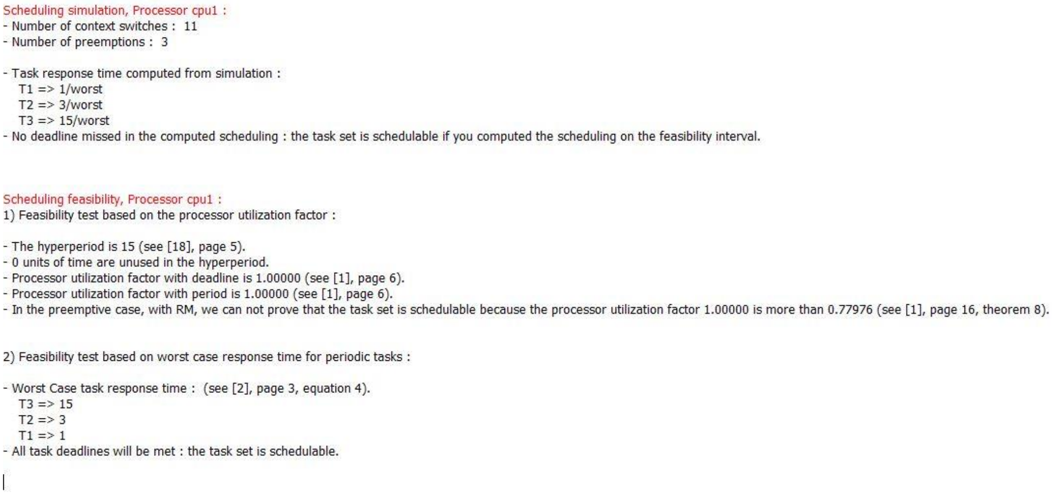


Task name=T1 Period= 3; Capacity= 1; Deadline= 3; Start time= 0; Priority= 1; Cpu=cpu1

Task name=T2 Period= 5; Capacity= 2; Deadline= 5; Start time= 0; Priority= 2; Cpu=cpu1

Task name=T3 Period= 15; Capacity= 4; Deadline= 15; Start time= 0; Priority= 3; Cpu=cpu1

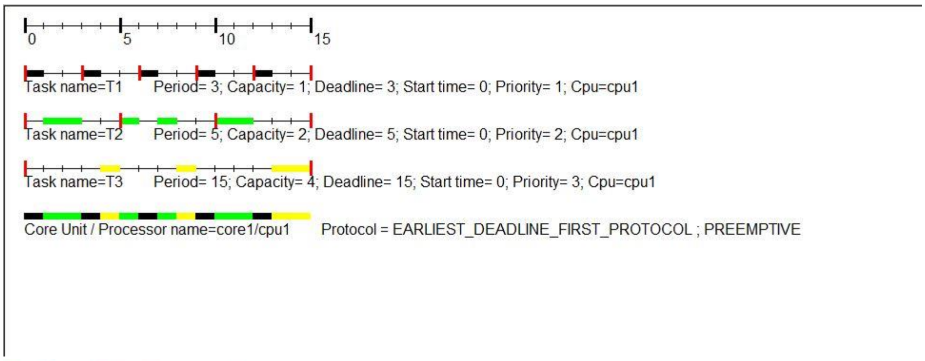
Core Unit / Processor name=core1/cpu1 Protocol = RATE_MONOTONIC_PROTOCOL ; PREEMPTIVE



```
Scheduling simulation, Processor cpu1 :
- Number of context switches :  11
- Number of preemptions :  3

- Task response time computed from simulation :
   T1 => 1/worst
   T2 => 3/worst
   T3 => 15/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 15 (see [18], page 5).
- 0 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 1.00000 (see [1], page 6).
- Processor utilization factor with period is 1.00000 (see [1], page 6).
- In the preemptive case, with RM, we can not prove that the task set is schedulable because the processor utilization factor 1.00000 is more than 0.77976 (see [1], page 16, theorem 8).


2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :  (see [2], page 3, equation 4).
   T3 => 15
   T2 => 3
   T1 => 1
- All task deadlines will be met : the task set is schedulable.
```

Figure :- 22 RM Diagram & Simulation & Feasibility

EDF:-

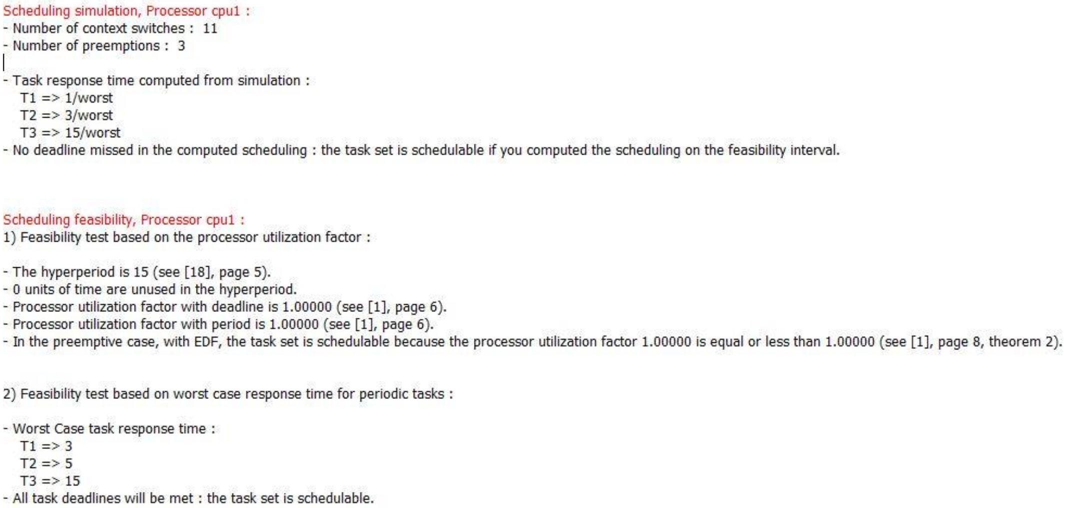


Task name=T1 Period= 3; Capacity= 1; Deadline= 3; Start time= 0; Priority= 1; Cpu=cpu1

Task name=T2 Period= 5; Capacity= 2; Deadline= 5; Start time= 0; Priority= 2; Cpu=cpu1

Task name=T3 Period= 15; Capacity= 4; Deadline= 15; Start time= 0; Priority= 3; Cpu=cpu1

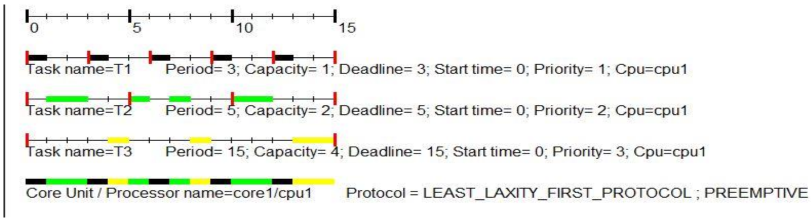
Core Unit / Processor name=core1/cpu1 Protocol = EARLIEST_DEADLINE_FIRST_PROTOCOL ; PREEMPTIVE



```
Scheduling simulation, Processor cpu1 :
- Number of context switches :  11
- Number of preemptions :  3

- Task response time computed from simulation :
    T1 => 1/worst
    T2 => 3/worst
    T3 => 15/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 15 (see [18], page 5).
- 0 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 1.00000 (see [1], page 6).
- Processor utilization factor with period is 1.00000 (see [1], page 6).
- In the preemptive case, with EDF, the task set is schedulable because the processor utilization factor 1.00000 is equal or less than 1.00000 (see [1], page 8, theorem 2).


2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :
    T1 => 3
    T2 => 5
    T3 => 15
- All task deadlines will be met : the task set is schedulable.
```

Figure :- 23 EDF Diagram & Simulation & Feasibility

LLF:-

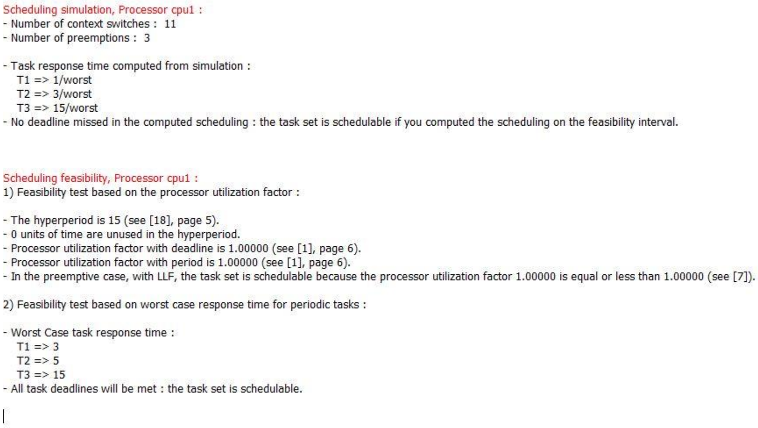


Task name=T1 Period= 3; Capacity= 1; Deadline= 3; Start time= 0; Priority= 1; Cpu=cpu1

Task name=T2 Period= 5; Capacity= 2; Deadline= 5; Start time= 0; Priority= 2; Cpu=cpu1

Task name=T3 Period= 15; Capacity= 4; Deadline= 15; Start time= 0; Priority= 3; Cpu=cpu1

Core Unit / Processor name=core1/cpu1 Protocol = LEAST_LAXITY_FIRST_PROTOCOL ; PREEMPTIVE



```
Scheduling simulation, Processor cpu1 :
- Number of context switches :  11
- Number of preemptions :  3

- Task response time computed from simulation :
    T1 => 1/worst
    T2 => 3/worst
    T3 => 15/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 15 (see [18], page 5).
- 0 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 1.00000 (see [1], page 6).
- Processor utilization factor with period is 1.00000 (see [1], page 6).
- In the preemptive case, with LLF, the task set is schedulable because the processor utilization factor 1.00000 is equal or less than 1.00000 (see [7]).

2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :
    T1 => 3
    T2 => 5
    T3 => 15
- All task deadlines will be met : the task set is schedulable.
```

Figure :- 24 LLF Diagram & Simulation & Feasibility

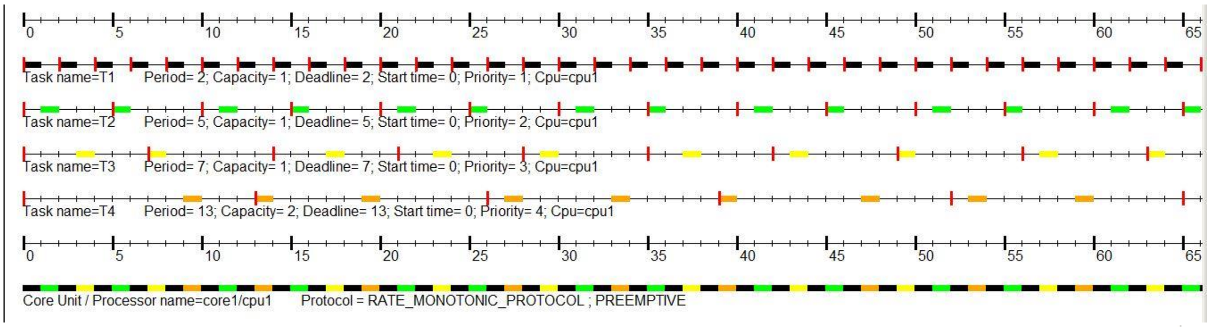
The Cheddar and the code both passes RM,EDF,LLF Feasibility tests.

The code matches/agrees with cheddar. The code algorithm is built in such a way that it checks if the scheduling of the services written is feasible or not.

It also Checks if the total utility is <=1.

Example :- 8

RM:-



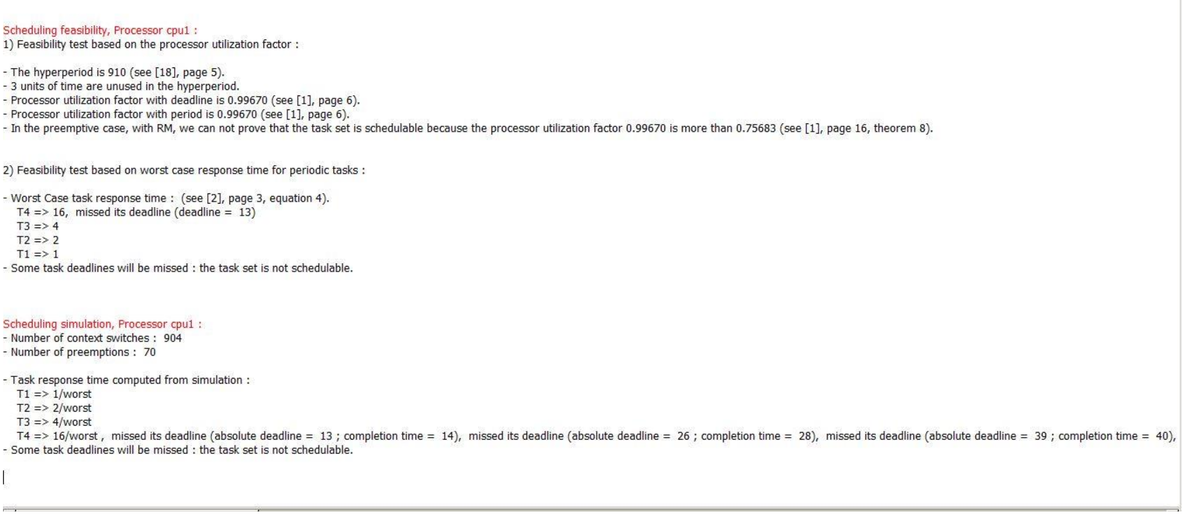
Task name=T1 Period= 2; Capacity= 1; Deadline= 2; Start time= 0; Priority= 1; Cpu=cpu1

Task name=T2 Period= 5; Capacity= 1; Deadline= 5; Start time= 0; Priority= 2; Cpu=cpu1

Task name=T3 Period= 7; Capacity= 1; Deadline= 7; Start time= 0; Priority= 3; Cpu=cpu1

Task name=T4 Period= 13; Capacity= 2; Deadline= 13; Start time= 0; Priority= 4; Cpu=cpu1

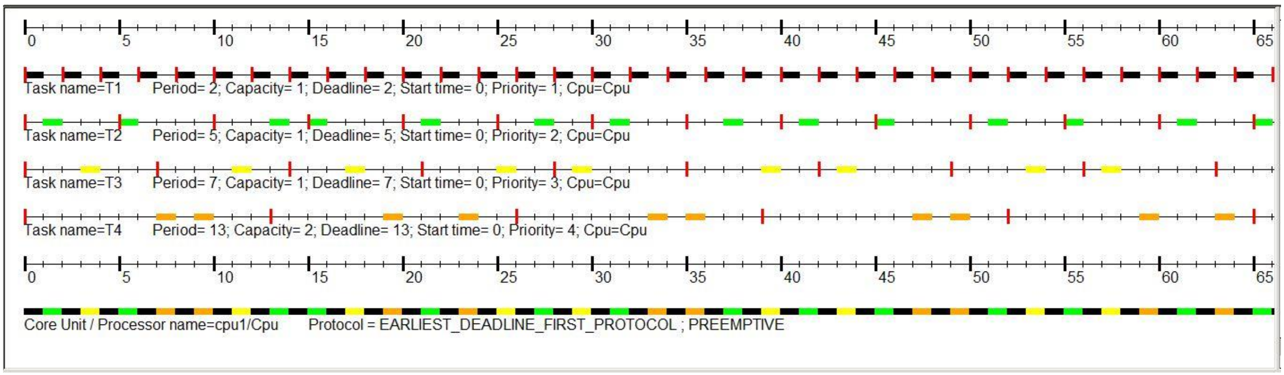
Core Unit / Processor name=core1/cpu1 Protocol = RATE_MONOTONIC_PROTOCOL ; PREEMPTIVE



```
Scheduling feasibility, Processor cpu1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 910 (see [18], page 5).
- 3 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.99670 (see [1], page 6).
- Processor utilization factor with period is 0.99670 (see [1], page 6).
- In the preemptive case, with RM, we can not prove that the task set is schedulable because the processor utilization factor 0.99670 is more than 0.75683 (see [1], page 16, theorem 8).


2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :  (see [2], page 3, equation 4).
   T4 => 16,  missed its deadline (deadline =  13)
   T3 => 4
   T2 => 2
   T1 => 1
- Some task deadlines will be missed : the task set is not schedulable.



Scheduling simulation, Processor cpu1 :
- Number of context switches :  904
- Number of preemptions :  70

- Task response time computed from simulation :
   T1 => 1/worst
   T2 => 2/worst
   T3 => 4/worst
   T4 => 16/worst ,  missed its deadline (absolute deadline =  13 ; completion time =  14),  missed its deadline (absolute deadline =  26 ; completion time =  28),  missed its deadline (absolute deadline =  39 ; completion time =  40),
- Some task deadlines will be missed : the task set is not schedulable.
```

Figure :- 25 RM Diagram & Simulation & Feasibility

EDF:-



Task name=T1 Period= 2; Capacity= 1; Deadline= 2; Start time= 0; Priority= 1; Cpu=Cpu

Task name=T2 Period= 5; Capacity= 1; Deadline= 5; Start time= 0; Priority= 2; Cpu=Cpu

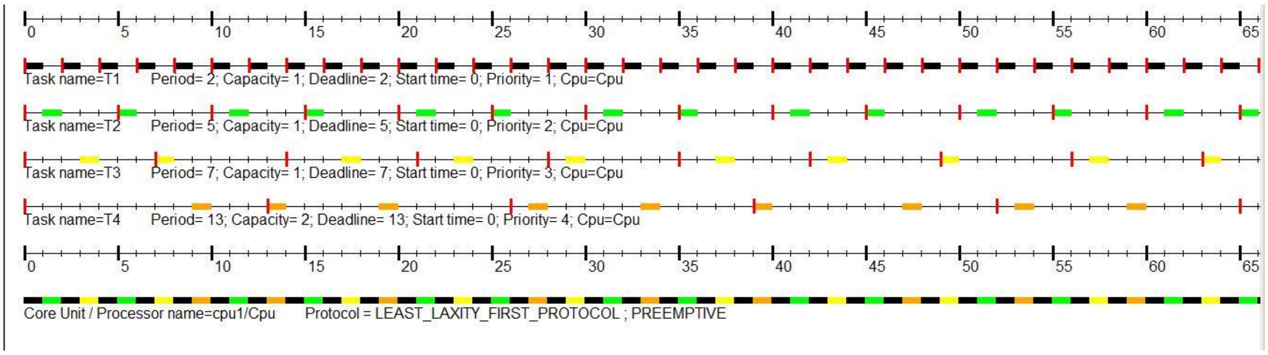
Task name=T3 Period= 7; Capacity= 1; Deadline= 7; Start time= 0; Priority= 3; Cpu=Cpu

Task name=T4 Period= 13; Capacity= 2; Deadline= 13; Start time= 0; Priority= 4; Cpu=Cpu

Core Unit / Processor name=cpu1/Cpu Protocol = EARLIEST_DEADLINE_FIRST_PROTOCOL ; PREEMPTIVE

```
Scheduling simulation, Processor Cpu :
- Number of context switches :  904
- Number of preemptions :  70

- Task response time computed from simulation :
   T1 => 1/worst
   T2 => 4/worst
   T3 => 6/worst
   T4 => 12/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling feasibility, Processor Cpu :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 910 (see [18], page 5).
- 3 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.99670 (see [1], page 6).
- Processor utilization factor with period is 0.99670 (see [1], page 6).
- In the preemptive case, with EDF, the task set is schedulable because the processor utilization factor 0.99670 is equal or less than 1.00000 (see [1], page 8, theorem 2).


2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :
   T1 => 1
   T2 => 4
   T3 => 6
   T4 => 12
- All task deadlines will be met : the task set is schedulable.
```

LLF:-



Task name=T1 Period= 2; Capacity= 1; Deadline= 2; Start time= 0; Priority= 1; Cpu=Cpu

Task name=T2 Period= 5; Capacity= 1; Deadline= 5; Start time= 0; Priority= 2; Cpu=Cpu

Task name=T3 Period= 7; Capacity= 1; Deadline= 7; Start time= 0; Priority= 3; Cpu=Cpu

Task name=T4 Period= 13; Capacity= 2; Deadline= 13; Start time= 0; Priority= 4; Cpu=Cpu

Core Unit / Processor name=cpu1/Cpu Protocol = LEAST_LAXITY_FIRST_PROTOCOL ; PREEMPTIVE

```
Scheduling simulation, Processor Cpu :
- Number of context switches :  904
- Number of preemptions :  70

- Task response time computed from simulation :
   T1 => 1/worst
   T2 => 2/worst
   T3 => 4/worst
   T4 => 16/worst ,  missed its deadline (absolute deadline =  13 ; completion time =  14),  missed its deadline (absolute deadline =  26 ; completion time =  28),  missed its deadline (absolute deadline =  39 ; completion time =  40),
- Some task deadlines will be missed : the task set is not schedulable.


Scheduling feasibility, Processor Cpu :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 910 (see [18], page 5).
- 3 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.99670 (see [1], page 6).
- Processor utilization factor with period is 0.99670 (see [1], page 6).
- In the preemptive case, with LLF, the task set is schedulable because the processor utilization factor 0.99670 is equal or less than 1.00000 (see [7]).

2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :
   T1 => 1
   T2 => 4
   T3 => 6
   T4 => 12
- All task deadlines will be met : the task set is schedulable.
```

Here, RM is not passed but EDF and LLF are passed. The code matches/agrees with cheddar. The code algorithm is built in such a way that it checks if the scheduling of the services written is feasible or not.

It also Checks if the total utility is <=1.

Question - 4

Three Constraints are as follows :-

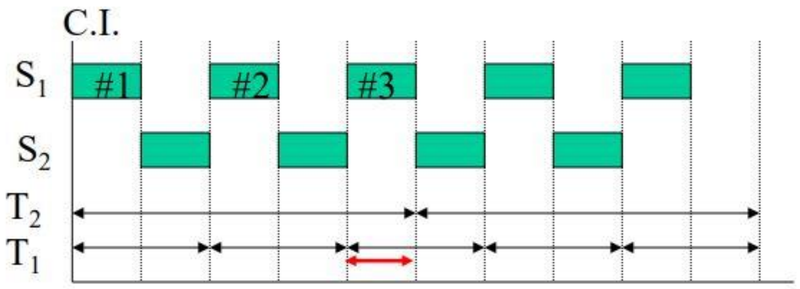
1) Each Task must have a fixed priority
2) The scheduling should be done in such a way that all the high priority tasks are able to pre-empt the lower priority tasks
3) The deadline of the task is considered to be equal to period of the task. In real world this might not always be true

Assumptions are as follows :-

1) Each task must be completed before next request occurs.
2) Requesting for all the services are not dependent on each other. Means even if one service is not completed then also it does not matter for the other service to start or stop.
3) Run time for each service is constant. It does not vary with time or the output of any other service.
4) All the services are requested on a periodic basis and period of them is also constant.

Key steps :-

Key step :- 1(Case 1 – C1 short enough to fit all three releases in T2.)



$$C_1 \le T_2 - T_1 \lfloor T_2 / T_1 \rfloor$$
$$C_2 = T_2 - C_1 \lceil T_2 / T_1 \rceil$$
$$U = \frac{C_1}{T_1} + \frac{C_2}{T_2}$$

Figure :- 26 Example of critical Time zone

C1 is small enough to fit into fractional 3rd T1 shown in the figure as red line
C2 = T2 **–** Interference from C1 Releases
Goal is to find U. To find U , first of all we have to find C1 and C2 from the diagram given above and then put the value of C2 in the equation of U = C1/T1 + C2/T2.

C1 is given by the equation :-

$$C_1 \leq T_2 - T_1 \lfloor T_2/T_1 \rfloor .$$

C2 is given by the equation :-

$$C_2 = T_2 - C_1 \lceil T_2/T_1 \rceil .$$

Thus we get the equation as below:-

$$U = \frac{C_1}{T_1} + \frac{[T_2 - C_1 \lceil T_2 / T_1 \rceil]}{T_2}$$

Now, lets consider the case when S1 (shown in figure) comes completely in T2. Then , S1 will occur exactly Ceil(T2/T1) times during T2.
Equating the equation of U more we get ,

$$U = \frac{C_1}{T_1} + \frac{T_2}{T_2} + \frac{[-C_1 \lceil T_2 / T_1 \rceil]}{T_2} \quad \text{(pull out } T_2 \text{ term)}$$

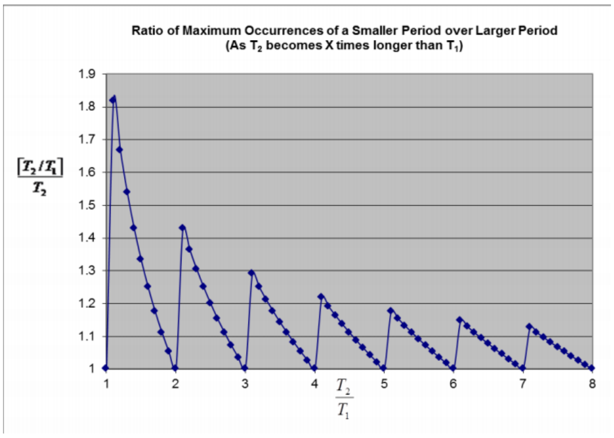
$$U = \frac{C_1}{T_1} + 1 + \frac{[-C_1 \lceil T_2 / T_1 \rceil]}{T_2} \quad \text{(note that } T_2 \text{ term is 1)}$$

$$U = 1 + C_1 \left[(1/T_1) - \frac{\lceil T_2 / T_1 \rceil}{T_2} \right] \quad \text{(combine } C_1 \text{ terms)}$$

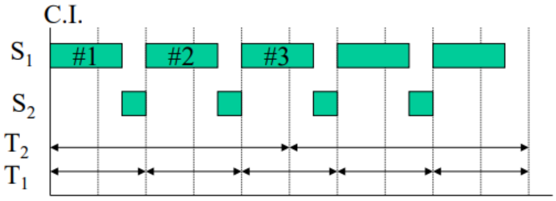
This gives you Equation 3.5:

$$U = 1 + C_1 \left[(1/T_1) - \frac{\lceil T_2 / T_1 \rceil}{T_2} \right]$$

U decreases monotonically, with increasing C1 when T2> T1



Key step :- 2 (Case 2 – C1 too large to fit last release in T2.)



$$C_1 \ge T_2 - T_1\lfloor T_2 / T_1 \rfloor$$
$$C_2 = T_1\lfloor T_2 / T_1 \rfloor - C_1\lfloor T_2 / T_1 \rfloor$$
$$U = \frac{C_1}{T_1} + \frac{C_2}{T_2}$$

Figure :- 27 Case 2 Overrun of Critical time zone by S1.

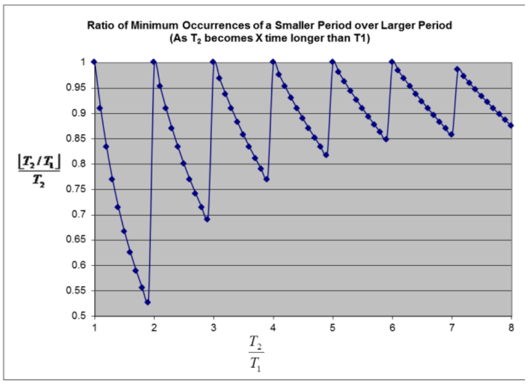
Even though S1 overruns the critical time zone, time remains for S2, and we could find a value of C2 for S2 that still allows it to meet its deadline of T2.
To find the value of C2 , S1 first release #1 plus #2 along with some fraction of #3 leave some amount of time left over for S2.

If the third release exceeds the critical time zone, sometime remains for S2. The sum of all occurrences of T1 during T2 can be expressed as T1*floor(T2/T1). The amount of time S1 takes during this T1*floor(T2/T1) duration is exactly C1*floor(T2/T1).

Putting the value of C2 in U= C1/T1 + C2/T2

We get,
U = (T1/T2) *floor(T2/T1) + C1*((1/T1) - (1/T2) * floor(T2/T1)).

U monotonically increases with increasing C1 when T2>T1



Key Step -3 :-

Determining U in terms of T1 & T2 and independent of C1

After comparing the formula of U from the 2 cases and comparing them we get the formula for RM LUB, in which C1 and C2 are given by

$$C_1 \geq T_2 - T_1\lfloor T_2 / T_1 \rfloor$$
$$C_2 = T_1\lfloor T_2 / T_1 \rfloor - C_1\lfloor T_2 / T_1 \rfloor$$
$$U = \frac{C_1}{T_1} + \frac{C_2}{T_2}$$

Substituting C1 and C2 in the formula for U, we get

U = 1- (T1/T2) * (ceil(T2/T1) - (T2/T1)) * ((T2/T1)-floor(T2/T1))

Substituting I = floor (T2/T1

$$U = 1 - \left(\frac{f(1-f)}{(T_2 / T_1)} \right)$$

References:-

1) **Assumptions and Constraints of Q-4 taken from Professor Siewert's Notes**
2) http://qa.geeksforgeeks.org/5569/qa.geeksforgeeks.org/5569/difference-between-static-dynamic-priorities-scheduling
3) **Liu Layland Paper**
4) **RTES book of Prof Sam Siewert chapter 3**