Exercise 2 – Service Scheduling Feasibility Analysis

Anuhya Kuraparthy

June 28, 2023

0.0.1 If you're using embedded Linux, make yourself an account on your R-Pi3b or 4b. For embedded Linux, we will use native development tools (rather than cross-compiling and cross-development commonly used for smaller-scale embedded systems).

1. Account setup and verification – You can use the default "pi" account and "raspberry" password, but it is more secure to make a user account. You can make a user account during install or later. To add an account later use "sudo adduser," enter the well-known password, and enter user information as you see fit. Add your new user account as a "sudoer" using "visudo" right below root with the same privileges (if you need help with "vi", here's a quick reference or reference card– use arrows to position cursor, below root hit Esc, "i" for insert, type username and privileges as above, and when done, Esc, ":", "wq"). The old unix vi editor was one of the first full-screen visual editors – it still has the advantage of being found on virtually any Unix system in existence, but is otherwise cryptic – it is still widely used in IT, by developers and systems engineers, so it's good to know the basics. If you really do not like vi or vim, your next best bet is "nano," "VS code" or "geany" for Unix systems (you can normally do "sudo apt-get install geany" or any alternative editor you like best). You are welcome to use whatever editor works best for you in this class. Finally, you'll want an SSH tool (e.g., MobaXterm for Windows) and you'll want to enable SSH via the GUI (or with a headless methods). Do a quick "sudo whoami" to demonstrate success for account creation.

   Using the sudo visudo command, the file shown in Figure 1 is edited to make the user account a root account. This is proven to be a success by checking with the command- sudo whoami in Figure 2.

```
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
anuhya  ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

@includedir /etc/sudoers.d
```

Figure 1: File for 'sudo visudo'

```
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.
Last login: Thu Jun 22 17:40:41 2023
anuhya@raspberrypi:~ $ sudo whoami
root
anuhya@raspberrypi:~ $
```

Figure 2: Output of 'sudo whoami'

2. Logout and test your login, then logout. Use Alt+Print-Screen to capture your desktop and save as proof you set up your account. Note that you can always get a terminal with Ctrl+Alt+t key combination. Show evidence that you have created a custom login with screenshots or photos with your smart phone. If you have any questions about your Linux systems setup, please get help from student assistants or the instructor.

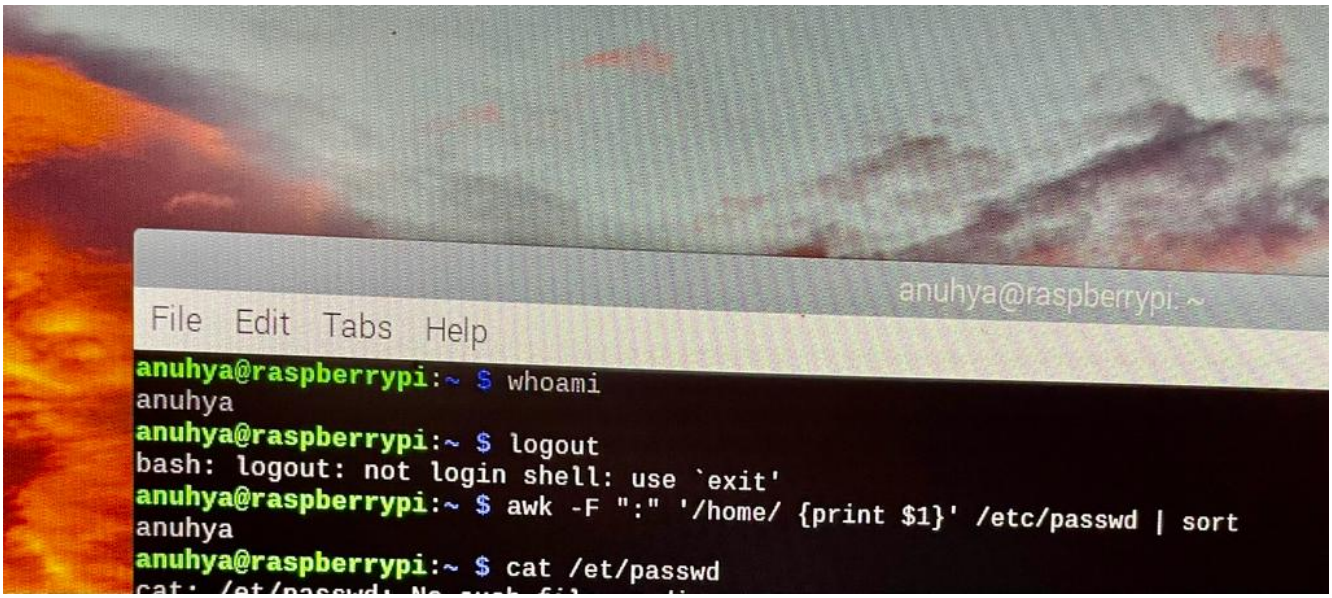

Figure 3: User accounts

The output of the command- awk -F ":" '/home/ print $1' /etc/passwd — sort is used to show the users listed in '/etc/passwd' file which have a home directories. This is shown in Figure 3. The listed users are only the newly created users with a custom login.

3. Make sure you can access graphical tools with MobaXterm or VNC and show that tools you may need in the future work such as editors like "geany" or "eom" (sudo apt-get install geany, or sudo apt-get install eom), a tool to display graphics in PPM or PGM format. For example, show that our class web page on Firefox (or default browser) and set your home page to https://sites.google.com/colorado.edu/ecen5623-summer. Overall, make sure you are comfortable with development, debug, compiler general native or cross-development tools and document and demonstrate that you know them.

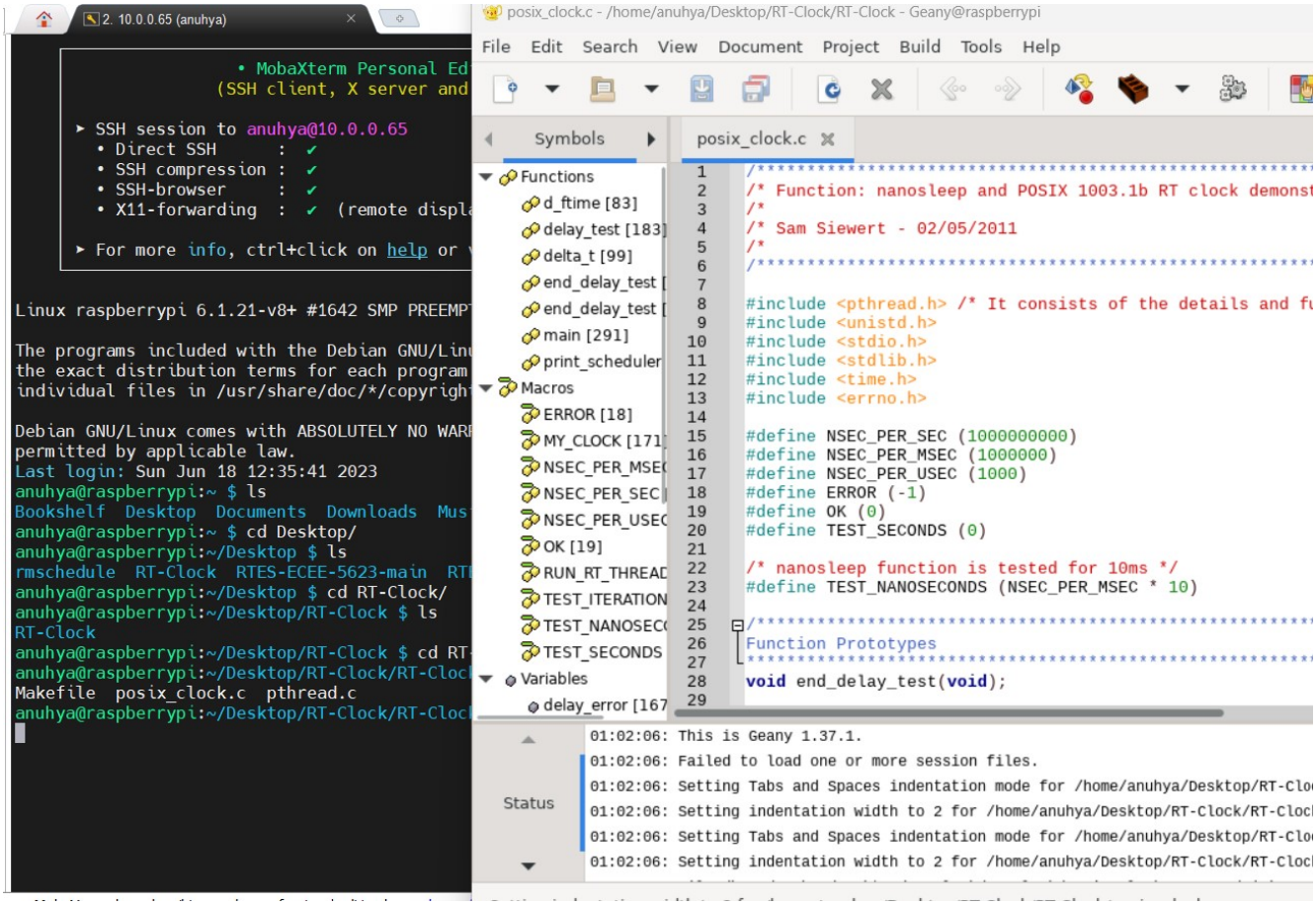

Figure 4: Working of Geany

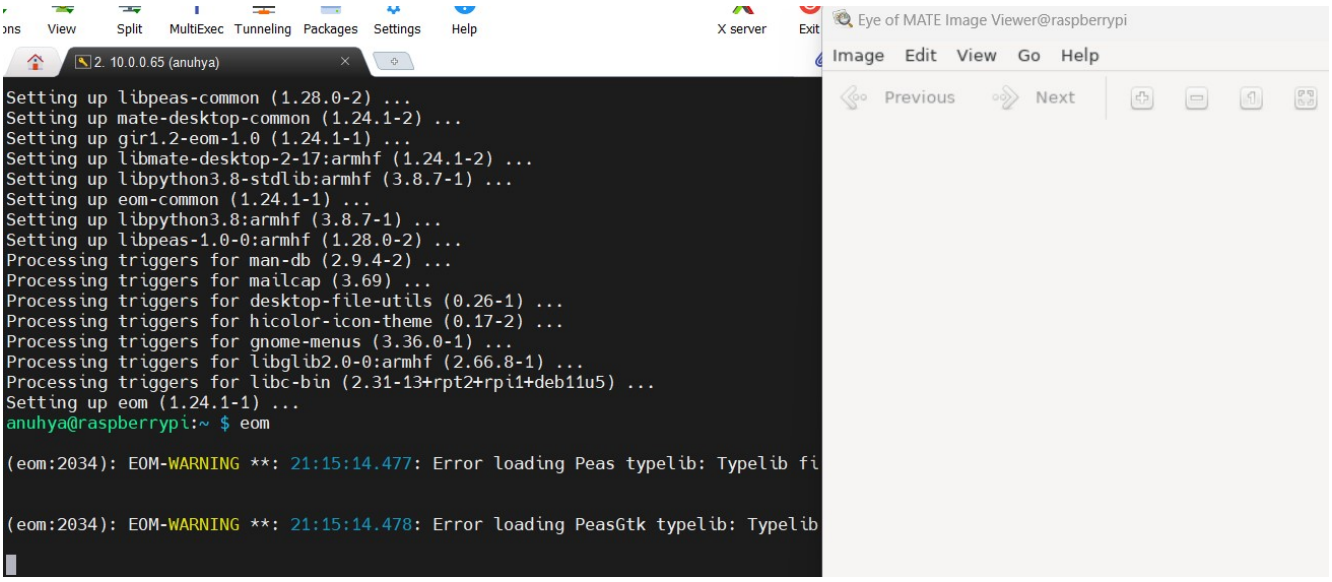

Figure 5: Working of EOM

0.0.2 Read the paper "Architecture of the Space Shuttle Primary Avionics Software System" [available on Canvas], by Gene Carlow.

1. Provide an explanation and critique of the frequency executive architecture.

The paper describes the architecture of NASA's Space Shuttle Orbiter Primary Avionics System. The article places a strong emphasis on the well-structured architecture which led to the success of PASS architecture. Factors like the avionic system design requirements, computer design (Data processing system and general-purpose computer), memory constraints, redundancy management, man/machine interface, functional and performance requirements of the applications, and the requirement for design modularity and modification flexibility were considered for the design. The successful accommodation of these factors contributed to the success of the PASS system.

Operational structure: The architecture was adjusted to meet operational, reliability and physical constraints. The main memory capacity is limited to 106K 32-bit words which is not sufficient to contain all the software code. For this reason, the main memory has been segmented into eight different phase/function combinations, each linked to a unique operation sequence. The software required to support a single critical mission phase was decided to be placed in each redundant GPC main memory throughout that phase to ensure more system reliability. This would be loaded into the GPC main memory at the initiation of the OPS. The memory load for each OPS consisted of 3 parts: resident or systems software (common data and code to all OPS loads), major function base (common to major application function), and OPS overlay (unique to an OPS load). The memory loading occurs during the transition from one OPS to another in response to a major application function.

Man/machine interface: The PASS man/machines was designed to accommodate a user with minimal effort and sufficient knowledge about the system. The structure is a part of OPS and contains: major nodes, specialist functions, and display functions. Major modes are divided into blocks linked to the CRT display pages to facilitate communication and control. The modes are triggered either by a keyboard entry or as the result of a event detected by the software. The specialist function (SPEC) is only initiated with a keyboard entry from the user, and executes independently with other processing in the OPS. They are also linked to the CRT display to enable control and monitoring for background functions. Lastly, the display function (DISP) is only used to monitor the processing results of a SPEC or major mode function.
The control segment implements the OPS and consists of logic blocks that define the structure of OPS, major mode, SPEC, or DISP. The control segment grammar is a library of macros that was developed to standardise this design and implementation of control segments.

System software: Traditionally, the flight system software had synchronous design where each application was dispatched at specific points in the system cycle. These require careful synchronisation to prevent overruns at process and system levels. However, the vast number of application functions, and expected changes in the Shuttle program led to a nonsynchronous approach for the PASS architecture. The major elements of the system software include: Flight computer operating system, system control, user interface.

FCOS: The FCOS performs the following functions

- Process management: The FCOS controls the allocation of all internal computer resources. This is done based on the requests from other systems or application processes. It ensures that the highest priority system or process in the multitasking priority queue structure receives the CPU resources.
- IO management: The input/output processor resources for each GPC contain the master sequence controller, and bus control elements. The function handles I/O requests, ensures the redundant computers receive identical data, and performs I/O fault isolation and correction.
- Data processing system (DPS) configuration management: It handles the loading of the GPC memory, sequencing and control of GPC and IOP operating states. Further, the code/data transfers between the mass memory and GPC main memory are enabled by this function. Lastly, the function establishes the redundant computer set for support of critical mission failures.

System control: The system control performs initialization and configuration control of the DPS and avionic data network. It is evoked to establish an interface and primary/secondary relationship with other GPCs, communication path with keyboard/display units, and support for avionic subsystems during OPS execution. During the GPC start-up, a control segment is implemented. The GPC is placed in idle OPS and system-level SPECS performs the following functions: initiating a memory dump, changing configuration of DPS.

Application software: Application software for the following applications were developed: Guidance, navigation, and control (GN&C), vehicle systems management (SM) and vehicle checkout (VCO). The GN&C software determines the vehicle's position, velocity and altitude. It performs sensor redundancy, provides the crew with displays and data entry capabilities to monitor and control the subsystem; and issues the engine

commands for a mission. The SM monitors and performs configuration of the Orbiter and payload subsystems. It detects any abnormal behaviour and alerts the crew via CRT displays or alarms. The VCO provides software support for testing, integration and certification of avionics subsystems during vehicle preparation and in-flight orbit coast period.

Real-time Scheduling in PASS: The GN&C is a cyclic closed loop application which performs all the functions with tight timing and phasing relationships. It includes 6 different OPSs. The execution rates in these OPS vary from 25 Hz for support of basic vehicle flight control to 0.25Hz for display update. The design structure is based on phasing of the function executions and I/O, and this is performed by a cyclic process-the executive. It uses a table-driven dispatching design to initiate and control the principles functions. This enables sequencing of the function executions to be altered through a dispatcher table update (DTU) module. There are 3 executive structures: high-frequency executive for functions related to vehicle flight control, medium-frequency executive for functions related to navigation (state estimation from multiple sensors over time) and lastly, the low-frequency executive for functions related to guidance (long term targets). Thus, it can be concluded to be a cyclic executive with high, medium and low frequency tasks with priorities.

2. What advantages and disadvantages does the frequency executive have compared to the real-time threading and tasking implementation methods for real-time software systems? Please be specific about the advantages and disadvantages and provide at least three advantages as well as three disadvantages.

   Advantages of Frequency Executive:

   (a) Determinism: The system provides deterministic behaviour as it operates based on a fixed, predictable schedule. Thus, it is possible to predict the entire future history of the state of the system, once the start time is determined. This also satisfies the basic requirement of a hard real time system.

   (b) Simplicity: As no individual task can be pre-empted by another application process or system during its execution, the system design is simpler and this can lead to more reliability and easier development. Further, the executive overhead of context switching faced in real-time threading can be kept to a minimum. The design also avoids the complexities faced with managing threads and their synchronization, and priority inversion issues.

   (c) Low jitter: Jitter refers to the variation in the time a computed resulted is output to the external environment every cycle. Controlling jitter is important in real-time systems such as feedback control systems, or for applications dependent on specialized I/O devices. Because of the deterministic nature of the scheduling of the system, it will exhibit low jitter in the execution of each periodic task.

   (d) Resource efficiency: The CPU resources and time are allocated based on the task priorities and thus, allows for efficient utilization of system resources.

   Disadvantages of Frequency Executive:

   (a) Lack of Flexibility: The system operates on a fixed schedule, which might face application fragility during any further changes to the system. At the system integration or during maintenance of the system, when additional functionality needs to be implemented, the resulting application will exceed its time frame as defined for the original execution. In real-time threading and tasking implementation, it is more convenient to add additional tasks to a fixed priority system or even a dynamic priority system could be implemented.

   (b) Limited concurrency: Frequency executive could be called single-thread application and hence, there is limited concurrency. For applications with concurrent tasks that require parallel execution, there is not much support. Real-time threading and tasking allow for usage of multi-core processors.

   (c) Portability: The system is often hardware and operating system specific. This would limit usage of the software across different platforms. Real-time threading can be adapted to various operating systems and architectures.

   (d) Input and output buffers usage: functions whose execution time is longer than the period of the most frequent task, are often broken into shorter ones which can fit into multiple frames between the high-rate sampling frames. This is essentially pre-empting the longer task. In this method, the problem with shared resources persists and it is handled by using multiple input and output buffers which is similar to the way real-time threading handles synchronization problems.

   (e) Scalability issues: As the number of tasks in a system increase, it becomes more complex to assign their frequency and schedule such tasks. Real-time threading offers more support in implementing such complex systems.

0.0.3 Download feasibility example code and build it on your Raspberry Pi or alternate system of your choice and execute the code. Note that the C code already implements the RM LUB and RM exact feasibility analysis using scheduling point and completion test methods, but we have not studied methods to automate EDF or LLF feasibility beyond hand analysis methods, so for EDF and LLF, rely upon Cheddar and your hand analysis.

1. Compare the tests provided to analysis using Cheddar for the first 4 examples tested by the code.

```
anuhya@raspberrypi:~/Desktop/RTES-ECEE-5623-main/Feasibility $ sudo ./feasibility_tests
******** Completion Test Feasibility Example
Ex-0 U=73.33% (C1=1, C2=1, C3=2; T1=2, T2=10, T3=15; T=D): CT test FEASIBLE
for 3, utility_sum = 0.000000
for 0, wcet=1.000000, period=2.000000, utility_sum = 0.500000
for 1, wcet=1.000000, period=10.000000, utility_sum = 0.600000
for 2, wcet=2.000000, period=15.000000, utility_sum = 0.733333
utility_sum = 0.733333
LUB = 0.779763
RM LUB FEASIBLE

Ex-1 U=98.57% (C1=1, C2=1, C3=2; T1=2, T2=5, T3=7; T=D): INFEASIBLE
for 3, utility_sum = 0.000000
for 0, wcet=1.000000, period=2.000000, utility_sum = 0.500000
for 1, wcet=1.000000, period=5.000000, utility_sum = 0.700000
for 2, wcet=2.000000, period=7.000000, utility_sum = 0.985714
utility_sum = 0.985714
LUB = 0.779763
RM LUB INFEASIBLE

Ex-2 U=99.67% (C1=1, C2=1, C3=1, C4=2; T1=2, T2=5, T3=7, T4=13; T=D): INFEASIBLE
for 4, utility_sum = 0.000000
for 0, wcet=1.000000, period=2.000000, utility_sum = 0.500000
for 1, wcet=1.000000, period=5.000000, utility_sum = 0.700000
for 2, wcet=1.000000, period=7.000000, utility_sum = 0.842857
for 3, wcet=2.000000, period=13.000000, utility_sum = 0.996703
utility_sum = 0.996703
LUB = 0.756828
RM LUB INFEASIBLE

Ex-3 U=93.33% (C1=1, C2=2, C3=3; T1=3, T2=5, T3=15; T=D): FEASIBLE
for 3, utility_sum = 0.000000
for 0, wcet=1.000000, period=3.000000, utility_sum = 0.333333
for 1, wcet=2.000000, period=5.000000, utility_sum = 0.733333
for 2, wcet=3.000000, period=15.000000, utility_sum = 0.933333
utility_sum = 0.933333
LUB = 0.779763
RM LUB INFEASIBLE

Ex-4 U=100.00% (C1=1, C2=1, C3=4; T1=2, T2=4, T3=16; T=D): FEASIBLE
for 3, utility_sum = 0.000000
for 0, wcet=1.000000, period=2.000000, utility_sum = 0.500000
for 1, wcet=1.000000, period=4.000000, utility_sum = 0.750000
for 2, wcet=4.000000, period=16.000000, utility_sum = 1.000000
utility_sum = 1.000000
LUB = 0.779763
RM LUB INFEASIBLE



******** Scheduling Point Feasibility Example
Ex-0 U=73.33% (C1=1, C2=1, C3=2; T1=2, T2=10, T3=15; T=D): FEASIBLE
for 3, utility_sum = 0.000000
for 0, wcet=1.000000, period=2.000000, utility_sum = 0.500000
for 1, wcet=1.000000, period=10.000000, utility_sum = 0.600000
for 2, wcet=2.000000, period=15.000000, utility_sum = 0.733333
utility_sum = 0.733333
LUB = 0.779763
RM LUB FEASIBLE

Ex-1 U=98.57% (C1=1, C2=1, C3=2; T1=2, T2=5, T3=7; T=D): INFEASIBLE
for 3, utility_sum = 0.000000
for 0, wcet=1.000000, period=2.000000, utility_sum = 0.500000
for 1, wcet=1.000000, period=5.000000, utility_sum = 0.700000
for 2, wcet=2.000000, period=7.000000, utility_sum = 0.985714
utility_sum = 0.985714
LUB = 0.779763
RM LUB INFEASIBLE

Ex-2 U=99.67% (C1=1, C2=1, C3=1, C4=2; T1=2, T2=5, T3=7, T4=13; T=D): INFEASIBLE
for 4, utility_sum = 0.000000
for 0, wcet=1.000000, period=2.000000, utility_sum = 0.500000
for 1, wcet=1.000000, period=5.000000, utility_sum = 0.700000
for 2, wcet=1.000000, period=7.000000, utility_sum = 0.842857
for 3, wcet=2.000000, period=13.000000, utility_sum = 0.996703
utility_sum = 0.996703
LUB = 0.756828
RM LUB INFEASIBLE

Ex-3 U=93.33% (C1=1, C2=2, C3=3; T1=3, T2=5, T3=15; T=D): FEASIBLE
for 3, utility_sum = 0.000000
for 0, wcet=1.000000, period=3.000000, utility_sum = 0.333333
for 1, wcet=2.000000, period=5.000000, utility_sum = 0.733333
for 2, wcet=3.000000, period=15.000000, utility_sum = 0.933333
utility_sum = 0.933333
LUB = 0.779763
RM LUB INFEASIBLE

Ex-4 U=100.00% (C1=1, C2=1, C3=4; T1=2, T2=4, T3=16; T=D): FEASIBLE
for 3, utility_sum = 0.000000
for 0, wcet=1.000000, period=2.000000, utility_sum = 0.500000
for 1, wcet=1.000000, period=4.000000, utility_sum = 0.750000
for 2, wcet=4.000000, period=16.000000, utility_sum = 1.000000
utility_sum = 1.000000
LUB = 0.779763
RM LUB INFEASIBLE
```

Figure 6: Output of feasibility_tests program

The feasibility_tests program checks the feasibility of the 5 service sets by running the completion time and scheduling points tests and also compares the CPU utilisation with the RM LUB. The RM LUB is considered to be a pessimistic sufficient test to check the feasibility with RM scheduling. This means that it will also fail a service set that is real-time-safe occasionally as well. However, necessary and sufficient tests are precise. For example, service sets with relatively harmonic periods can easily fail the sufficient RM LUB test and be shown to be safe when analyzed. The competition time and scheduling point tests will precisely pass the service sets.

Lehoczky, Sha and Ding introduced the scheduling point test based on the assumptions listed in Liu and Layland's paper. The theorem is also follows:

$$\forall i, 1 \le i \le n, \min \sum_{j=1}^{i} C_j \left\lceil \frac{(l)T_k}{T_j} \right\rceil \le (l)T_k$$

$$(k,l) \in R_i$$

$$R_i = \left\{ (k,l) | 1 \le k \le i, l = 1, \ldots, \left\lfloor \frac{T_i}{T_k} \right\rfloor \right\}$$

- Where n is the number of tasks in the set S_i to S_n, where S_1 has higher priority than S_2, and S_n has higher priority than $S_{n>1}$.
- j identifies S_j, a service in the set between S_1 and S_n.
- k identifies S_k, a service whose l periods must be analyzed.
- l represents the number of periods of S_k to be analyzed.
- $\left\lceil \frac{(l)T_k}{T_j} \right\rceil$ represents the number of times S_j executes within l period of S_k.

$C_j \left\lceil \frac{(l)T_k}{T_j} \right\rceil$ is the time required by S_j to execute within l periods of S_k—

if the sum of these times for the set of tasks is smaller than l period of S_k, then the service set is feasible.

Figure 7: Scheduling point test

The completion time test is presented as follows:

$$a_n(t) = \sum_{j=1}^{n} \left\lceil \frac{t}{T_j} \right\rceil C_j$$

- $\left\lceil \frac{t}{T_j} \right\rceil$ is the number of executions of S_j at time t.
- $\left\lceil \frac{t}{T_j} \right\rceil C_j$ is the demand of S_j in time at t.
- $a_n(t)$ is the total cumulative demand from the n tasks up to time t.

Figure 8: Completion time test

If an(t) is less than or equal to the deadline for Sn, then it can be proven that Sn is feasible.

As the program output shown in the Figure 6, according to the N&S tests, service sets 0, 3 and 4 are feasible while service sets 1 and 2 are not feasible to schedule. According to the RM LUB test, only service sets 0 is feasible while the service sets 1, 2, 3 and 4 are not feasible to schedule. This analysis is verfied using Cheddar as follows.

Cheddar analysis

(a) **Sample Set 1**
Time periods: T1=2, T2=10, T3=15
Run-Times: C1=1, C2=1, C3=2

Set 1	Services	Freq f	f0 multiple	Period		WCET		Utility			
	S1	0.5	7.5	T1	2	C1	1	U1	0.5	**LCM =**	**30**
	S2	0.1	1.5	T2	10	C2	1	U2	0.1	**LUB =**	**77.98%**
	S3	0.0667	1	T3	15	C3	2	U3	0.134	**Utot =**	**73.33%**

Figure 9: Timing table for service set 1

Rate Monotonic Analysis

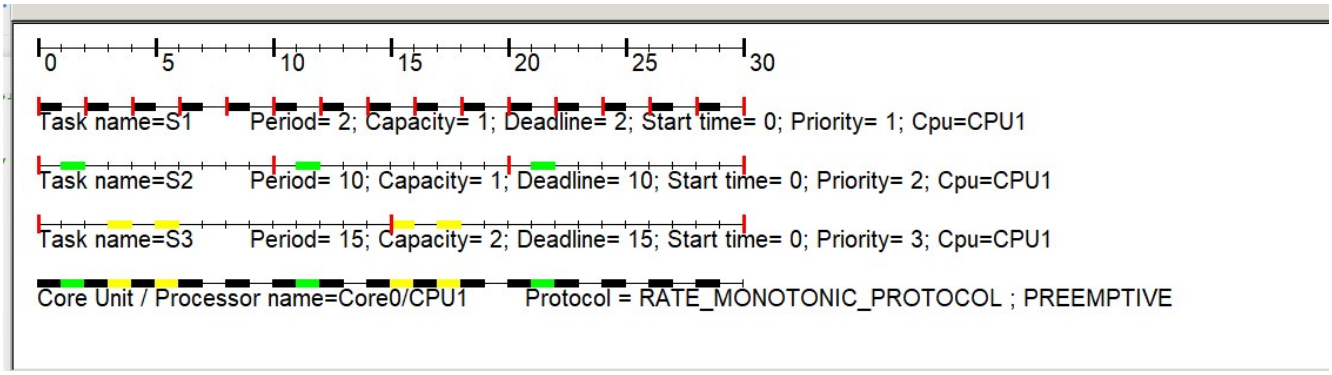

Figure 10: RM analysis for service set 1

```
Scheduling simulation, Processor CPU1 :
- Number of context switches :  14
- Number of preemptions :  2

- Task response time computed from simulation :
   S1 => 1/worst
   S2 => 2/worst
   S3 => 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).
   S3 => 6
   S2 => 2
   S1 => 1
- All task deadlines will be met : the task set is schedulable.
```

Earliest Deadline First

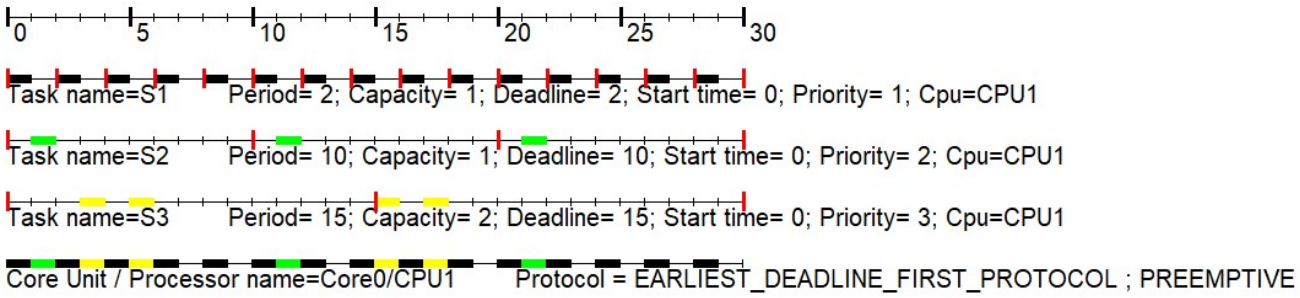

Figure 11: EDF analysis for service set 1

Scheduling simulation, Processor CPU1 :
- Number of context switches : 14
- Number of preemptions : 2

- Task response time computed from simulation :
  S1 => 1/worst
  S2 => 2/worst
  S3 => 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 :
  S1 => 1
  S2 => 8
  S3 => 13
- All task deadlines will be met : the task set is schedulable.

Least Laxity First

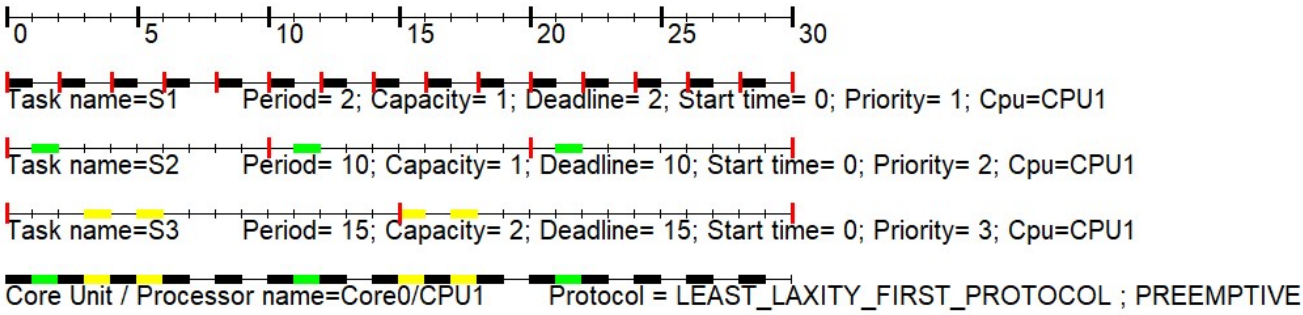

Figure 12: LLF analysis for service set 1

Scheduling simulation, Processor CPU1 :
- Number of context switches : 14
- Number of preemptions : 2

- Task response time computed from simulation :
  S1 => 1/worst
  S2 => 2/worst
  S3 => 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 :
  S1 => 1
  S2 => 8
  S3 => 13
- All task deadlines will be met : the task set is schedulable.

The service set (Figure 9) can be scheduled with all the policies: RM, EDF and LLF. The program output (Figure 6) confirms it is feasible with the analysis of the competition time and scheduling point tests. Further, the processor utilization is only 73.33%, which is lower than the RM LUB ie. 77.97%. With that we can conclude that the scheduling of the service set is both feasible and safe. The scheduling with all the policies is also confirmed with Cheddar analysis as shown in Figures 10, 11, 12.

(b) **Sample Set 2**
Time periods: T1=2, T2=5, T3=7
Run-Times: C1=1, C2=1, C3=2

Set 2	Services	Freq f	f0 multiple	Period		WCET		Utility			
	S1	0.5	3.5	T1	2	C1	1	U1	0.5	**LCM =**	**70**
	S2	0.2	1.4	T2	5	C2	1	U2	0.2	**LUB =**	**77.98%**
	S3	0.14286	1	T3	7	C3	2	U3	0.2857	**Utot =**	**98.57%**

Figure 13: Timing table for service set 2.

Rate Monotonic Analysis

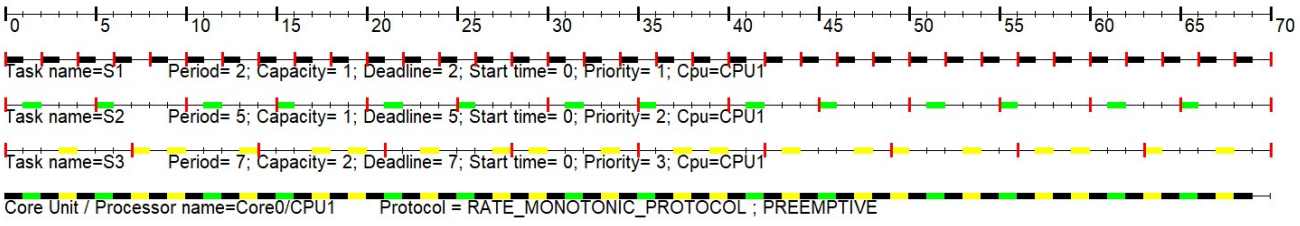

Figure 14: RM analysis for service set 2

Scheduling simulation, Processor CPU1 :
- Number of context switches : 68
- Number of preemptions : 10

- Task response time computed from simulation :
  S1 => 1/worst
  S2 => 2/worst
  S3 => 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).
  S3 => 8, missed its deadline (deadline = 7)
  S2 => 2
  S1 => 1
- Some task deadlines will be missed : the task set is not schedulable.

Earliest Deadline First

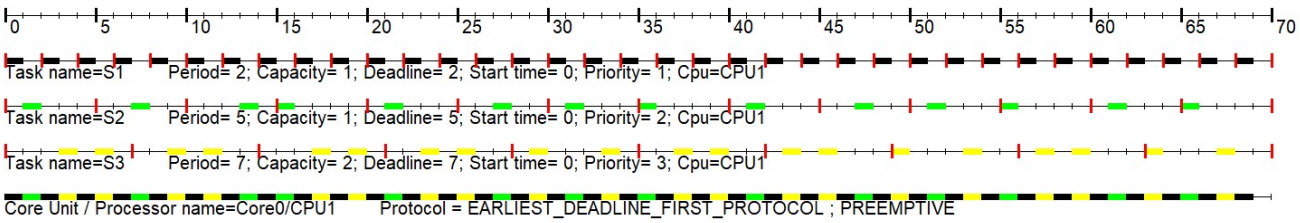

Figure 15: EDF analysis for service set 2

Scheduling simulation, Processor CPU1 :
- Number of context switches : 68
- Number of preemptions : 10

- Task response time computed from simulation :
  S1 => 1/worst
  S2 => 4/worst
  S3 => 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 :
  S1 => 1
  S2 => 4
  S3 => 6
- All task deadlines will be met : the task set is schedulable.

Least Laxity First

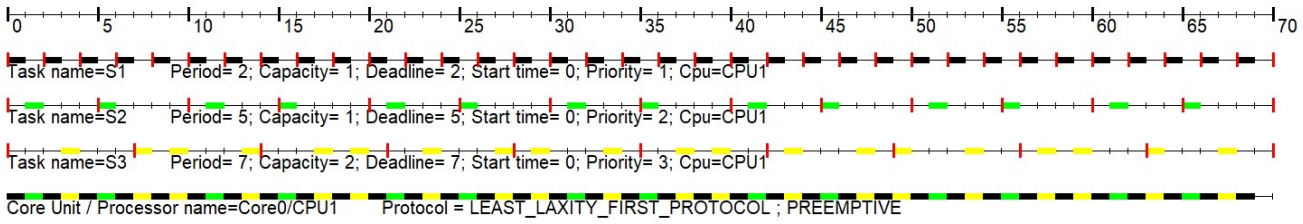

Figure 16: LLF analysis for service set 2

```
Scheduling simulation, Processor CPU1 :
- Number of context switches :  68
- Number of preemptions :  10

- Task response time computed from simulation :
   S1 => 1/worst
   S2 => 2/worst
   S3 => 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 :
   S1 => 1
   S2 => 4
   S3 => 6
- All task deadlines will be met : the task set is schedulable.
```

The service set (Figure 13) cannot be scheduled with the RM policies; however, it can be scheduled with dynamic priority policies – both EDF and LLF. The feasibility_tests program output also shows that the service set is unfeasible as it fails both the necessary and sufficient feasibility tests: scheduling point and completion time. Further, the processor utilization is only 98.57%, which is much higher than the RM LUB ie. 77.97%. The full processor efficiency can be achieved in cases where the time periods are harmonic, unlike this service set. Thus, even though the RM LUB is only a sufficiency test, since the service set cannot be scheduled for the least common period of all the time periods, RM scheduling is not feasible. In this overload case, the least priority service is missing its deadline. Cheddar analysis confirms that RM scheduling is not possible with this service set as shown in Figure 14, but EDF and LLF scheduling is possible as shown in Figure 15 & 16. In the entire hyperperiod, only one unit of time is unused and this would place constraints on IO and memory latencies.

(c) **Sample Set 3**
Time periods: T1=2, T2=5, T3=7, 74=13
Run-Times: C1=1, C2=1, C3=2, C4=2

Rate Monotonic Analysis

Set 3	Services	Freq f	f0 multiple	Period		WCET		Utility			
	S1	0.5	6.5	T1	2	C1	1	U1	0.5	**LCM =**	**910**
	S2	0.2	2.6	T2	5	C2	1	U2	0.2	**LUB =**	**75.68%**
	S3	0.14286	1.857	T3	7	C3	1	U3	0.143	**Utot =**	**99.67%**
	S4	0.0769	1	T4	13	C4	2	U4	0.1538		

Figure 17: Timing table for service set 3.

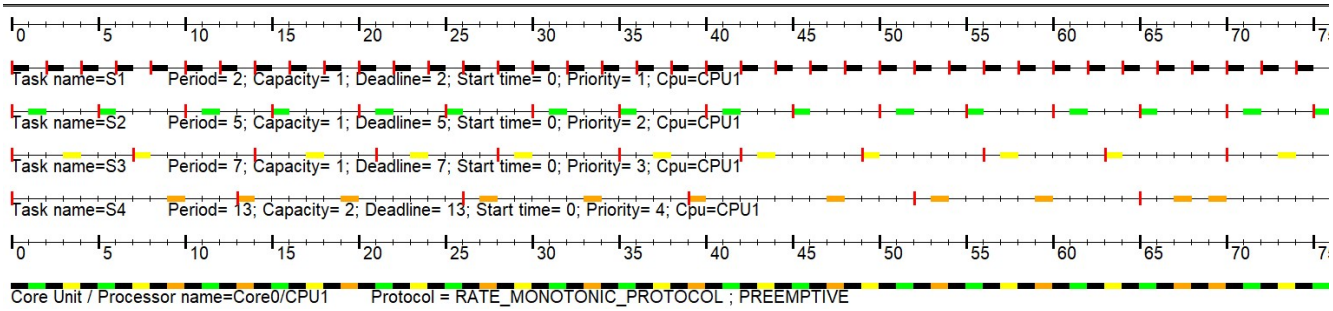

Figure 18: RM analysis for service set 3

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).
  S4 => 16, missed its deadline (deadline = 13)
  S3 => 4
  S2 => 2
  S1 => 1
- Some task deadlines will be missed : the task set is not schedulable.

Earliest Deadline First

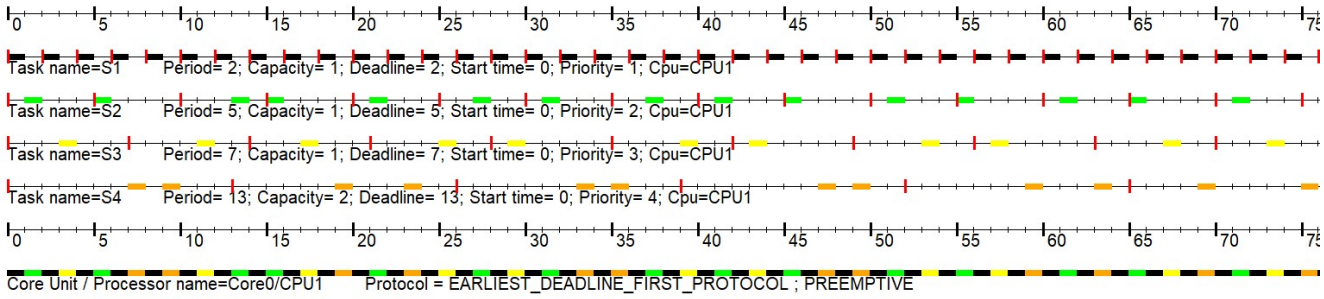

Figure 19: EDF analysis for service set 3

Scheduling simulation, Processor CPU1 :
- Number of context switches : 904
- Number of preemptions : 70

- Task response time computed from simulation :
  S1 => 1/worst
  S2 => 4/worst
  S3 => 6/worst
  S4 => 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 :
  S1 => 1
  S2 => 4
  S3 => 6
  S4 => 12
- All task deadlines will be met : the task set is schedulable.

Least Laxity First

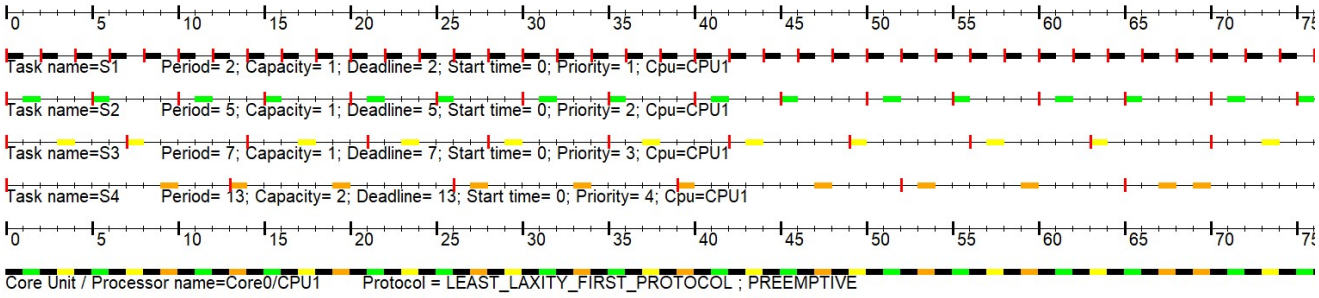

Figure 20: LLF analysis for service set 3

Scheduling simulation, Processor CPU1 :
- Number of context switches : 904
- Number of preemptions : 70

- Task response time computed from simulation :
  S1 => 1/worst
  S2 => 2/worst
  S3 => 4/worst
  S4 => 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), missed its deadline (absolut
- 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 :
  S1 => 1
  S2 => 4
  S3 => 6
  S4 => 12
- All task deadlines will be met : the task set is schedulable.

The service set (Figure 17) cannot be scheduled with the RM policy (Figure 18) as it does not pass the completion time test, scheduling point test and the processor utilisation is over the RM LUB limit. But the service set can be scheduled with the dynamic priority policy-EDF (Figure 19) but not LLF (Figure

20).

(d) **Sample Set 4**
Time periods: T1=3, T2=5, T3=15
Run-Times: C1=1, C2=2, C3=3

Set 4	Services	Freq f	f0 multiple	Period		WCET		Utility			
	S1	0.33	5	T1	3	C1	1	U1	0.33	**LCM =**	**15**
	S2	0.2	3	T2	5	C2	2	U2	0.4	**LUB =**	**77.98%**
	S3	0.0667	1	T3	15	C3	3	U3	0.2	**Utot =**	**98.33%**

Figure 21: Timing table for service set 4.

Rate Monotonic Analysis

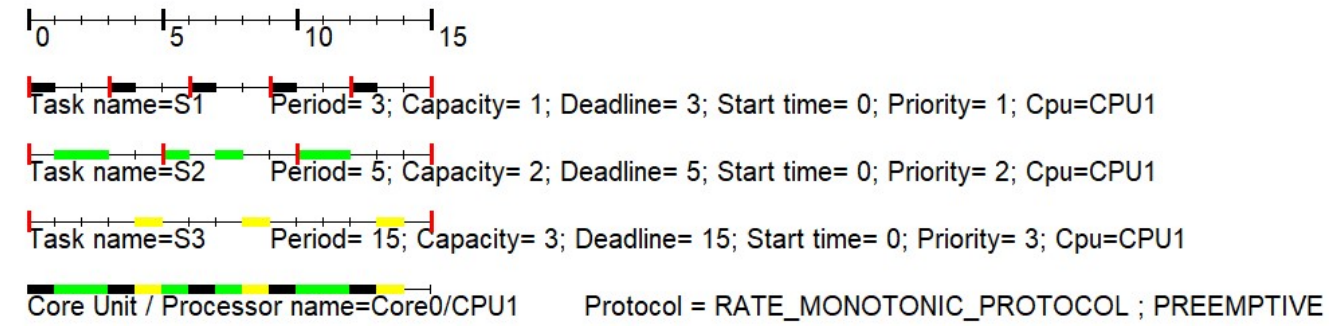

Figure 22: RM analysis for service set 4.

```
Scheduling simulation, Processor CPU1 :
- Number of context switches :  11
- Number of preemptions :  3

- Task response time computed from simulation :
   S1 => 1/worst
   S2 => 3/worst
   S3 => 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).
   S3 => 14
   S2 => 3
   S1 => 1
- All task deadlines will be met : the task set is schedulable.
```

Earliest Deadline First

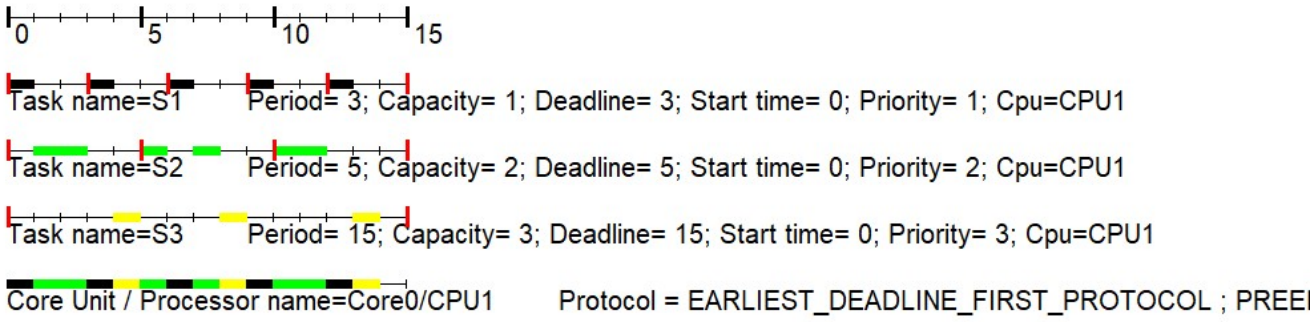

Figure 23: EDF analysis for service set 4.

Scheduling simulation, Processor CPU1 :
- Number of context switches : 11
- Number of preemptions : 3

- Task response time computed from simulation :
  S1 => 1/worst
  S2 => 3/worst
  S3 => 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 :
  S1 => 2
  S2 => 4
  S3 => 14
- All task deadlines will be met : the task set is schedulable.

Least Laxity First

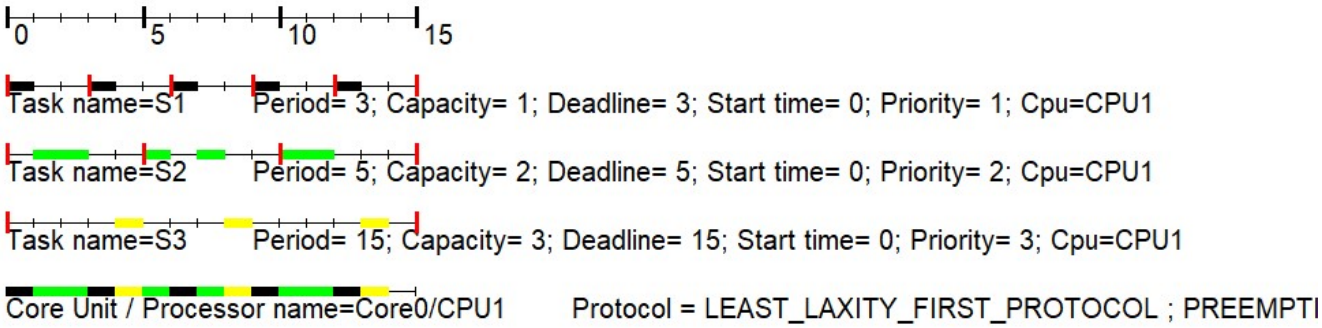

Figure 24: LLF analysis for service set 4.

```
Scheduling simulation, Processor CPU1 :
- Number of context switches :  11
- Number of preemptions :  3

- Task response time computed from simulation :
   S1 => 1/worst
   S2 => 3/worst
   S3 => 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 :
   S1 => 2
   S2 => 4
   S3 => 14
- All task deadlines will be met : the task set is schedulable.
```

The service set (Figure 21) can be scheduled with all the policies: RM, EDF and LLF. The feasibility_tests program output also shows that the service set is feasible as it passes both the necessary and sufficient feasibility tests: scheduling point and completion time. Even though the processor utilization (93.33%) is much higher than the RM LUB (77.97%), the service set is feasible. This is because the RM LUB is especially pessimistic for cases where the time periods are harmonic as higher efficiency can be achieved. Cheddar analysis confirms that RM scheduling (Figure 22) is possible with this service set, and even dynamic priorities policies- EDF (Figure 23) and LLF (Figure 24) scheduling is possible.

2. Now, implement additional examples for practice [6 more] of your interest from those that we reviewed in class (found here). Complete analysis using Cheddar RM. In cases where RM fails (fixed priority does not work), test EDF or LLF dynamic priority policies using hand analysis or Cheddar to see if either of those succeeds and if so, explain why. Cheddar uses both service simulations over the LCM of the periods as well as feasibility analysis based on the RM LUB and scheduling-point/completion-test algorithms, referred to as "Worst Case Analysis."

   (a) **Sample Set 5**
   Time periods: T1=2, T2=4, T3=16
   Run-Times: C1=1, C2=1, C3=4

Set 5	Services	Freq f	f0 multiple	Period		WCET		Utility			
	S1	0.5	8	T1	2	C1	1	U1	0.5	**LCM =**	**16**
	S2	0.25	4	T2	4	C2	1	U2	0.25	**LUB =**	**77.98%**
	S3	0.0625	1	T3	16	C3	4	U3	0.25	**Utot =**	**100.00%**

Figure 25: Timing table for service set 5.

Rate Monotonic Analysis

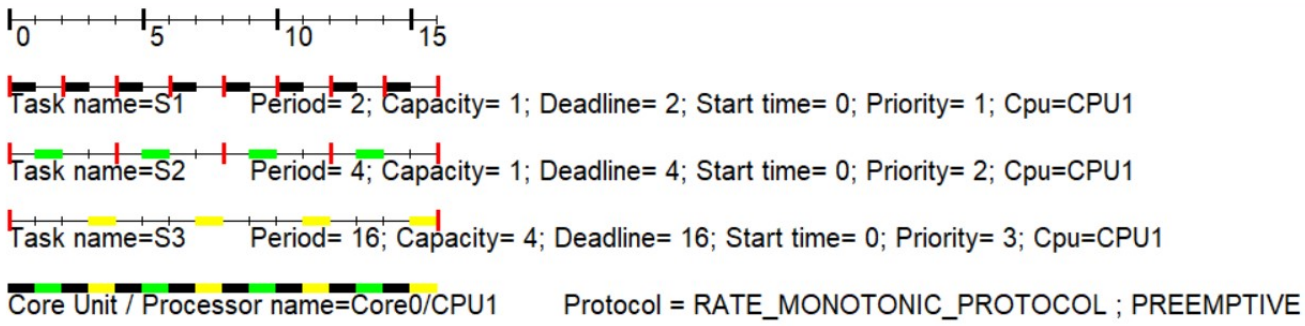

Figure 26: RM analysis for service set 5

```
Scheduling simulation, Processor CPU1 :
- Number of context switches :  15
- Number of preemptions :  3

- Task response time computed from simulation :
   S1 => 1/worst
   S2 => 2/worst
   S3 => 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).
   S3 => 16
   S2 => 2
   S1 => 1
- All task deadlines will be met : the task set is schedulable.
```

The service set (Figure 25) can be scheduled with the RM policy (Figure 26). The feasibility_tests program output (Figure 43) also shows that the service set is feasible as it passes both the necessary and sufficient feasibility tests: scheduling point and completion time. Even though the processor utilization (100%) is much higher than the RM LUB (77.97%), the service set is feasible. This is because the RM LUB is especially pessimistic for cases where the time periods are harmonic as higher efficiency can be achieved. Cheddar analysis confirms that RM scheduling (Figure 26) is possible with this service set.

(b) **Sample Set 6**
Time periods: T1=3, T2=5, T3=15
Run-Times: C1=1, C2=2, C3=5

Set 6	Service	Freq f	f0 multiple	Period		WCET		Utility			
	S1	0.333333333	5	T1	3	C1	1	U1	33.33%	**LCM =**	**15**
	S2	0.2	3	T2	5	C2	2	U2	40.00%	**LUB =**	**77.98%**
	S3	0.066666667	1	T3	15	C3	5	U3	33.33%	**Utot =**	**106.66%**

Figure 27: Timing table for service set 6.

Rate Monotonic Analysis

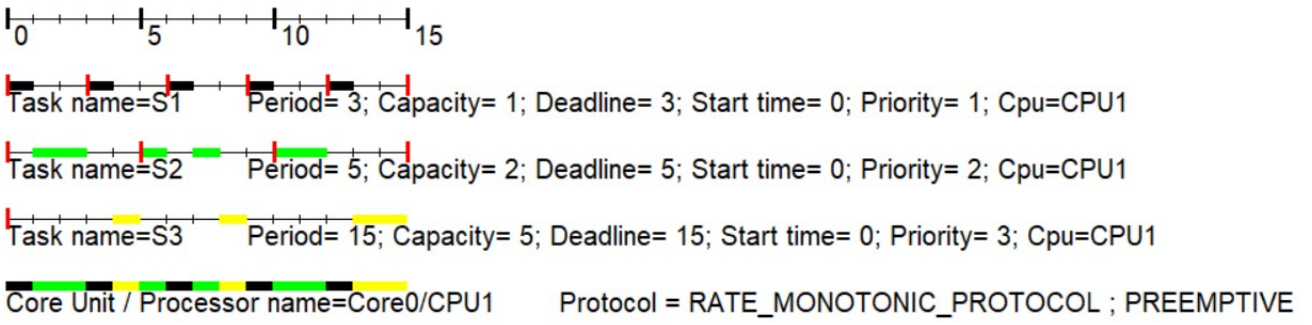

Figure 28: RM analysis for service set 6.

Scheduling simulation, Processor CPU1 :
- Number of context switches : 11
- Number of preemptions : 3

- Task response time computed from simulation :
  S1 => 1/worst
  S2 => 3/worst
  S3 => 0/worst , response time not computed since the task did not run all its capacity
- One or several tasks did not complete their execution.

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 1.06667 (see [1], page 6).
- Processor utilization factor with period is 1.06667 (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 1.06667 is more than 0.77976 (see [1], page 16, theorem 8).

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

- Processor utilization exceeded : can not compute worst case response time with this task set.

Earliest Deadline First

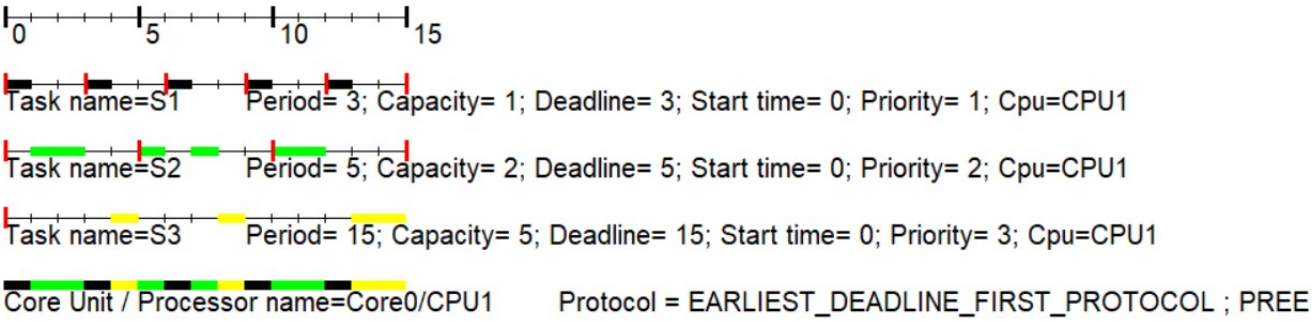

Figure 29: EDF analysis for service set 6.

```
Scheduling simulation, Processor CPU1 :
- Number of context switches :  11
- Number of preemptions :  3

- Task response time computed from simulation :
   S1 => 1/worst
   S2 => 3/worst
   S3 => 0/worst ,  response time not computed since the task did not run all its capacity
- One or several tasks did not complete their execution.


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 1.06667 (see [1], page 6).
- Processor utilization factor with period is 1.06667 (see [1], page 6).
- In the preemptive case, with EDF, the task set is not schedulable because the processor utilization factor 1.06667 is more than 1.00000 (see [1], page 8, theorem 2).


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

- Processor utilization exceeded : can not compute worst case response time with this task set.
```

Least Laxity First

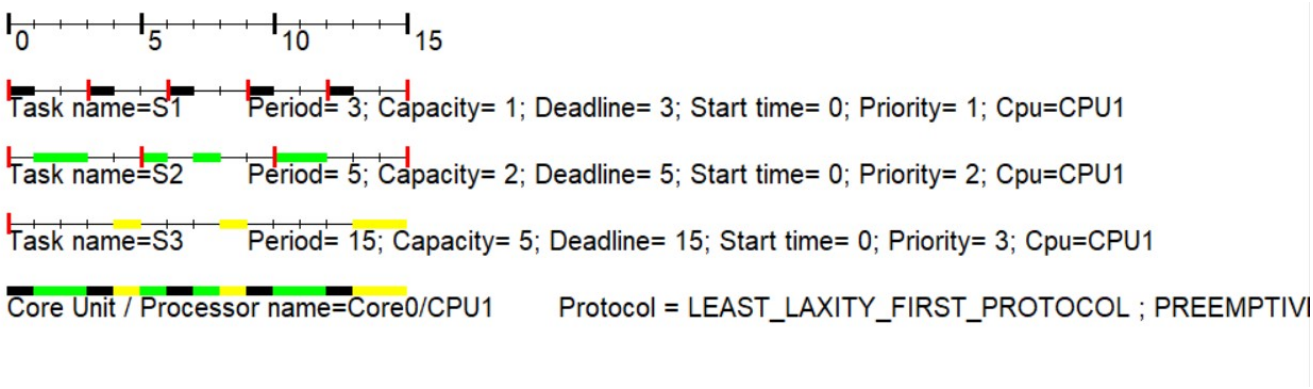

Figure 30: LLF analysis for service set 6.

```
Scheduling simulation, Processor CPU1 :
- Number of context switches :  11
- Number of preemptions :  3

- Task response time computed from simulation :
   S1 => 1/worst
   S2 => 3/worst
   S3 => 0/worst ,  response time not computed since the task did not run all its capacity
- One or several tasks did not complete their execution.


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 1.06667 (see [1], page 6).
- Processor utilization factor with period is 1.06667 (see [1], page 6).
- In the preemptive case, with LLF, the task set is not schedulable because the processor utilization factor 1.06667 is more than 1.00000 (see [7]).

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

- Processor utilization exceeded : can not compute worst case response time with this task set.
```

The service set (Figure 27) cannot be scheduled with either fixed priority policy or dynamic priorities policies. The processor utilisation factor in all cases is more than 100%. This is also proven by Cheddar

analysis (Figures- 28, 29, 30).This is also shown in the feasibility_tests program output (Figure 43).

(c) **Sample Set 7**
Time periods: T1=1, T2=1, T3=1, T4=2
Run-Times: C1=2, C2=5, C3=7, C4=14

Set 7	Service	Period		Freq f	f0 multiple	WCET		Utility			
	S1	T1	2	0.5	7	C1	1	U1	50.00%	**LCM =**	**70**
	S2	T2	5	0.2	2.8	C2	1	U2	20.00%		
	S3	T3	7	0.142857143	2	C3	1	U3	14.29%	**LUB =**	**75.68%**
	S4	T4	14	0.071428571	1	C4	2	U4	14.29%	**Utot =**	**98.57%**

Figure 31: Timing table for service set 7.

Rate Monotonic Analysis

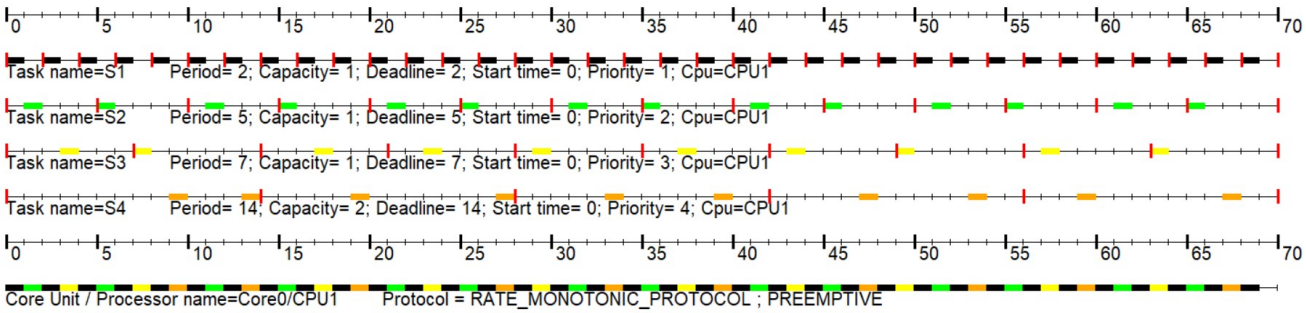

Figure 32: RM analysis for service set 7.

```
Scheduling simulation, Processor CPU1 :
- Number of context switches :  68
- Number of preemptions :  5

- Task response time computed from simulation :
   S1 => 1/worst
   S2 => 2/worst
   S3 => 4/worst
   S4 => 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 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 0.75683 (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).
   S4 => 14
   S3 => 4
   S2 => 2
   S1 => 1
- All task deadlines will be met : the task set is schedulable.
```

The service set (Figure 31) can be scheduled with RM policy. The service set has a processor utililization factor of 0.98571 which is more than the RM LUB- 0.7568. Thus, the task cannot be proven to be schedulable until computed over the hyperperiod. However, the feasibility_tests program output (Figure

43) shows that the service set is feasible as it passes both the necessary and sufficient feasibility tests: scheduling point and completion time. Hence, the service set can be scheduled with RM policy as proven with Cheddar analysis as well (Figure 32).

(d) **Sample Set 8**
Time periods: T1=2, T2=3, T3=4
Run-Times: C1=1, C2=1, C3=2

Set 8	Service	Freq f	f0 multiple	Period		WCET		Utility			
	S1	0.2	3.2	T1	5	C1	2	U1	40.00%	**LCM =**	**720**
	S2	0.111111111	1.777777778	T2	9	C2	3	U2	33.33%	**LUB =**	**77.98%**
	S3	0.0625	1	T3	16	C3	4	**U3**	**25.00%**	**Utot =**	**98.33%**

Figure 33: Timing table for service set 8.

Rate Monotonic Analysis

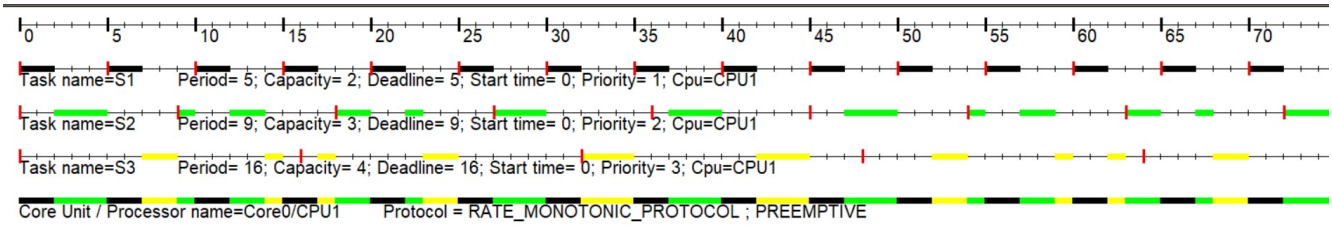

Figure 34: RM analysis for service set 8.

```
Scheduling feasibility, Processor CPU1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 720 (see [18], page 5).
- 12 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.98333 (see [1], page 6).
- Processor utilization factor with period is 0.98333 (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 0.98333 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).
   S3 => 18,  missed its deadline (deadline =  16)
   S2 => 5
   S1 => 2
- Some task deadlines will be missed : the task set is not schedulable.
```

Earliest Deadline First

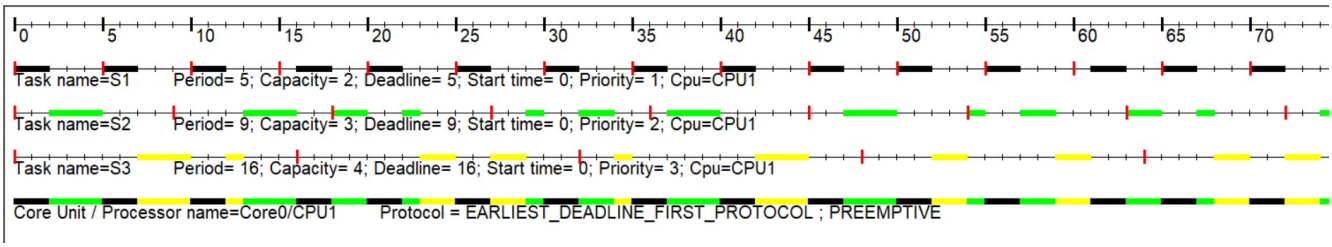

Figure 35: EDF analysis for service set 8.

```
Scheduling simulation, Processor CPU1 :
- Number of context switches :  359
- Number of preemptions :  91

- Task response time computed from simulation :
   S1 => 3/worst
   S2 => 7/worst
   S3 => 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 720 (see [18], page 5).
- 12 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.98333 (see [1], page 6).
- Processor utilization factor with period is 0.98333 (see [1], page 6).
- In the preemptive case, with EDF, the task set is schedulable because the processor utilization factor 0.98333 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 :
   S1 => 3
   S2 => 7
   S3 => 14
- All task deadlines will be met : the task set is schedulable.
```

Least Laxity First

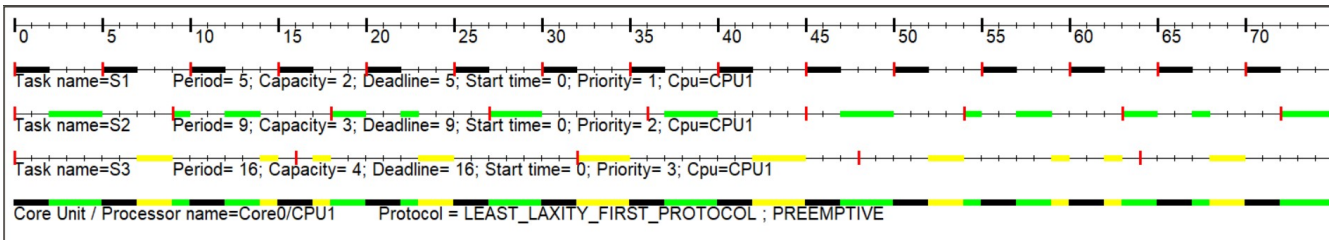

Figure 36: LLF analysis for service set 8.

```
Scheduling simulation, Processor CPU1 :
- Number of context switches :  351
- Number of preemptions :  97

- Task response time computed from simulation :
   S1 => 2/worst
   S2 => 5/worst
   S3 => 18/worst ,  missed its deadline (absolute deadline =  16 ; completion time =  18),  missed its deadline (absolute deadline =  32 ; completion time =  34),  missed its deadline (absolute deadline =  96 ; completion time =  98),  missed its deadline (absolut
- 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 720 (see [18], page 5).
- 12 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.98333 (see [1], page 6).
- Processor utilization factor with period is 0.98333 (see [1], page 6).
- In the preemptive case, with LLF, the task set is schedulable because the processor utilization factor 0.98333 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 :
   S1 => 3
   S2 => 7
   S3 => 14
- All task deadlines will be met : the task set is schedulable.
```

The service set (Figure 33) cannot be scheduled with RM policy as the service, S3 misses its deadline (Figure 34). This is shown in the feasibility_tests program output (Figure 43) also. The service set does not pass the scheduling point and completion time tests. The processor utilization is 0.9833 which is higher than the RM LUB-0.7797. It is possible to schedule it with EDF policy (Figure 35), but when scheduled with LLF (Figure 36), the service S3 misses some deadlines.

(e) **Sample Set 9**
Time periods: T1=2, T2=4, T3=7

Run-Times: C1=1, C2=1, C3=1

Set 9	Services	Freq f	f0 multiple	Period		WCET		Utility			
	S1	0.5	3.5	T1	2	C1	1	U1	0.5	**LCM =**	**28**
	S2	0.25	1.75	T2	4	C2	1	U2	0.25	**LUB =**	**77.98%**
	S3	0.142857143	1	T3	7	C3	1	U3	0.142857143	**Utot =**	**89.29%**

Figure 37: Timing table for service set 9.

Rate Monotonic Analysis

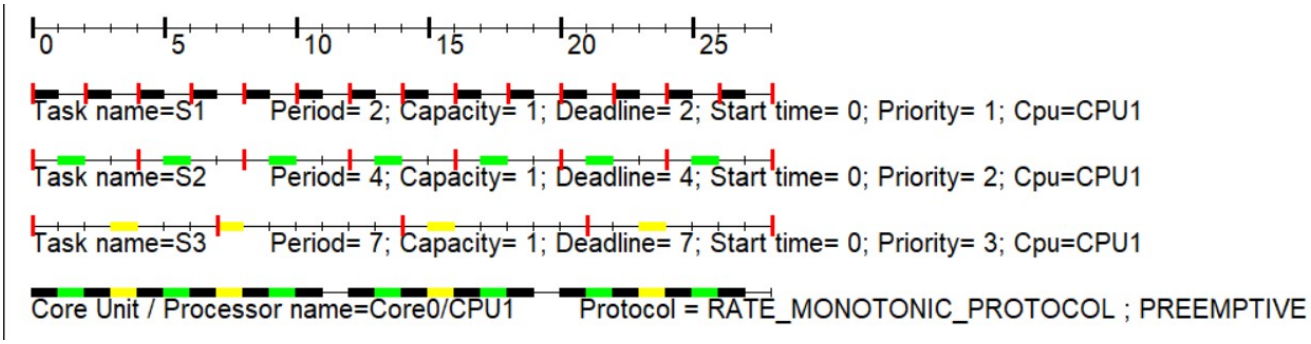

Figure 38: RM analysis for service set 9.

```
Scheduling simulation, Processor CPU1 :
- Number of context switches :  22
- Number of preemptions :  0

- Task response time computed from simulation :
   S1 => 1/worst
   S2 => 2/worst
   S3 => 4/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 28 (see [18], page 5).
- 3 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.89286 (see [1], page 6).
- Processor utilization factor with period is 0.89286 (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 0.89286 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).
   S3 => 4
   S2 => 2
   S1 => 1
- All task deadlines will be met : the task set is schedulable.
```

The service set (Figure 37) is schedulable with RM policy (Figure 38) even though the processor utilization is more than the RM LUB of the set. The feasibility_tests program output (Figure 43) shows that the service set is feasible as it passes both the necessary and sufficient feasibility tests: scheduling point and completion time.

(f) **Sample Set 10**
Time periods: T1=5, T2=9, T3=15
Run-Times: C1=3, C2=3, C3=3

Set 10	Service	Freq f	f0 multiple	Period		WCET		Utility			
	S1	0.2	3	T1	5	C1	3	U1	60.00%	**LCM =**	**45**
	S2	0.111111111	1.666666667	T2	9	C2	3	U2	33.33%	**LUB =**	**77.98%**
	S3	0.066666667	1	T3	15	C3	3	**U3**	**20.00%**	**Utot =**	**113.33%**

Figure 39: Timing table for service set 10.

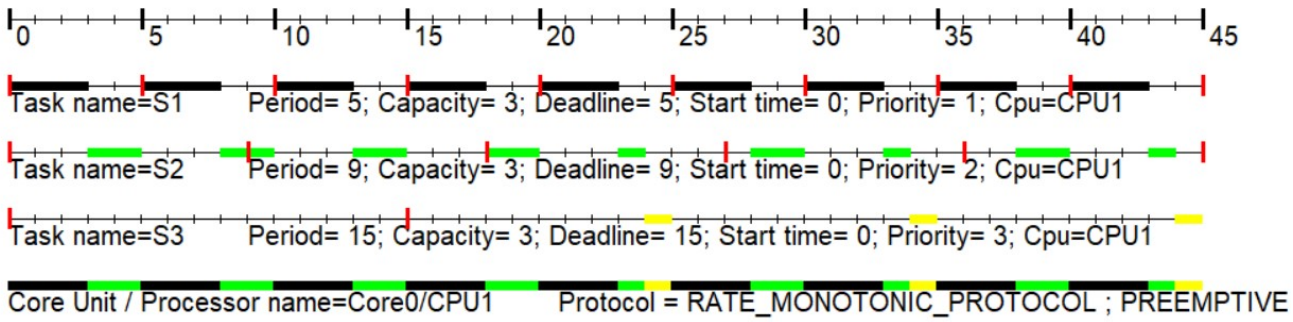

Figure 40: RM analysis for service set 10.

Rate Monotonic Analysis

Scheduling simulation, Processor CPU1 :
- Number of context switches : 20
- Number of preemptions : 7

- Task response time computed from simulation :
  S1 => 3/worst
  S2 => 9/worst
  S3 => 45/worst , missed its deadline (absolute deadline = 15 ; completion time = 45)
- 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 45 (see [18], page 5).
- 0 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 1.13333 (see [1], page 6).
- Processor utilization factor with period is 1.13333 (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 1.13333 is more than 0.77976 (see [1], page 16, theorem 8).

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

- Processor utilization exceeded : can not compute worst case response time with this task set.

Earliest Deadline First

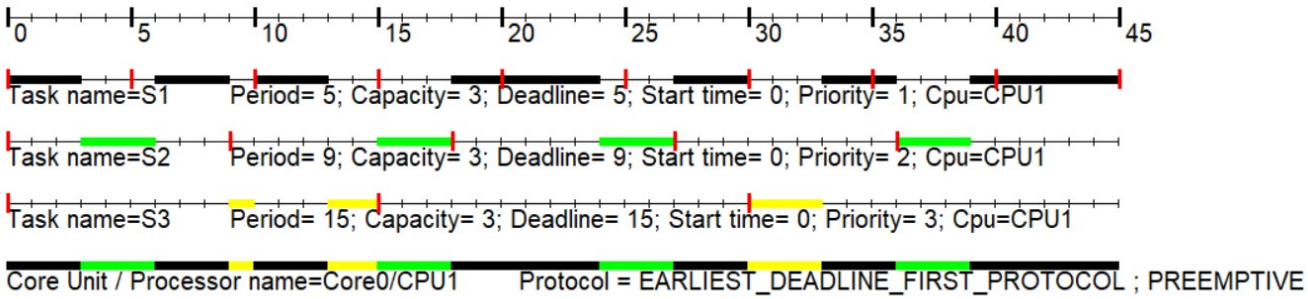

Figure 41: EDF analysis for service set 10.

Scheduling simulation, Processor CPU1 :
- Number of context switches : 13
- Number of preemptions : 1

- Task response time computed from simulation :
  S1 => 7/worst , missed its deadline (absolute deadline = 20 ; completion time = 21), missed its deadline (absolute deadline = 35 ; completion time = 36), missed its deadline (absolute deadline = 40 ; completion time = 42)
  S2 => 12/worst , missed its deadline (absolute deadline = 36 ; completion time = 39)
  S3 => 18/worst , missed its deadline (absolute deadline = 30 ; completion time = 33)
- 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 45 (see [18], page 5).
- 0 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 1.13333 (see [1], page 6).
- Processor utilization factor with period is 1.13333 (see [1], page 6).
- In the preemptive case, with EDF, the task set is not schedulable because the processor utilization factor 1.13333 is more than 1.00000 (see [1], page 8, theorem 2).

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

- Processor utilization exceeded : can not compute worst case response time with this task set.

Least Laxity First

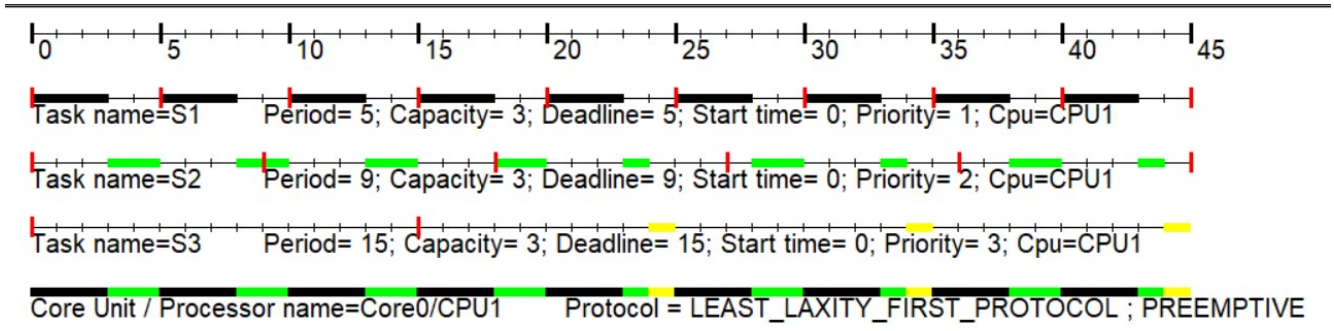

Figure 42: LLF analysis for service set 10.

Scheduling simulation, Processor CPU1 :
- Number of context switches : 20
- Number of preemptions : 7

- Task response time computed from simulation :
  S1 => 3/worst
  S2 => 9/worst
  S3 => 45/worst , missed its deadline (absolute deadline = 15 ; completion time = 45)
- 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 45 (see [18], page 5).
- 0 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 1.13333 (see [1], page 6).
- Processor utilization factor with period is 1.13333 (see [1], page 6).
- In the preemptive case, with LLF, the task set is not schedulable because the processor utilization factor 1.13333 is more than 1.00000 (see [7]).

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

- Processor utilization exceeded : can not compute worst case response time with this task set.

The service set (Figure 39) has a processor utilisation of 113.33% and thus, it is not possible to schedule it with any of the scheduling policies- RM, EDF, LLF (Figures 40-43).

3. Does your modified feasibility code agree with Cheddar analysis for additional cases? Why or why not?

   The modified feasibility code is in the folder marked Feasibility. The output of the program with the six additional sets of services in shown in the Figures below.

```
anuhya@raspberrypi:~/Desktop/Feasibility $ sudo ./feasibility_tests
******** Completion Test Feasibility Example
Ex-5 U=100.00% (C1=1, C2=1, C3=4; T1=2, T2=4, T3=16; T=D):
CT test FEASIBLE
 SP test FEASIBLE
for 3, utility_sum = 0.000000
for 0, wcet=1.000000, period=2.000000, utility_sum = 0.500000
for 1, wcet=1.000000, period=4.000000, utility_sum = 0.750000
for 2, wcet=4.000000, period=16.000000, utility_sum = 1.000000
utility_sum = 1.000000
LUB = 0.779763
RM LUB INFEASIBLE

Ex-6 U=106.67% (C1=1, C2=2, C3=5; T1=3, T2=5, T3=15; T=D):
CT test INFEASIBLE
SP test INFEASIBLE
for 3, utility_sum = 0.000000
for 0, wcet=1.000000, period=3.000000, utility_sum = 0.333333
for 1, wcet=2.000000, period=5.000000, utility_sum = 0.733333
for 2, wcet=5.000000, period=15.000000, utility_sum = 1.066667
utility_sum = 1.066667
LUB = 0.779763
RM LUB INFEASIBLE

Ex-7 U=98.57% (C1=1, C2=1, C3=1, C4=2; T1=2, T2=5, T3=7, T4=14; T=D):
CT test FEASIBLE
SP test FEASIBLE
for 4, utility_sum = 0.000000
for 0, wcet=1.000000, period=2.000000, utility_sum = 0.500000
for 1, wcet=1.000000, period=5.000000, utility_sum = 0.700000
for 2, wcet=1.000000, period=7.000000, utility_sum = 0.842857
for 3, wcet=2.000000, period=14.000000, utility_sum = 0.985714
utility_sum = 0.985714
LUB = 0.756828
RM LUB INFEASIBLE

Ex-8 U=98.33% (C1=2, C2=3, C3=4; T1=5, T2=9, T3=16; T=D):
CT test INFEASIBLE
SP test INFEASIBLE
for 3, utility_sum = 0.000000
for 0, wcet=2.000000, period=5.000000, utility_sum = 0.400000
for 1, wcet=3.000000, period=9.000000, utility_sum = 0.733333
for 2, wcet=4.000000, period=16.000000, utility_sum = 0.983333
utility_sum = 0.983333
LUB = 0.779763
RM LUB INFEASIBLE

Ex-9 U=89.29% (C1=1, C2=1, C3=1; T1=2, T2=4, T3=7; T=D):
CT test FEASIBLE
SP test FEASIBLE
for 3, utility_sum = 0.000000
for 0, wcet=1.000000, period=2.000000, utility_sum = 0.500000
for 1, wcet=1.000000, period=4.000000, utility_sum = 0.750000
for 2, wcet=1.000000, period=7.000000, utility_sum = 0.892857
utility_sum = 0.892857
LUB = 0.779763
RM LUB INFEASIBLE

Ex-10 U=113.33% (C1=3, C2=3, C3=3; T1=5, T2=9, T3=15; T=D):
CT test INFEASIBLE
SP test INFEASIBLE
for 3, utility_sum = 0.000000
for 0, wcet=3.000000, period=5.000000, utility_sum = 0.600000
for 1, wcet=3.000000, period=9.000000, utility_sum = 0.933333
for 2, wcet=3.000000, period=15.000000, utility_sum = 1.133333
utility_sum = 1.133333
LUB = 0.779763
RM LUB INFEASIBLE
```

Figure 43: Output of modified feasibility_tests program.

Modified Feasibility code: The code is modified to get the tests' results for the 6 new service sets. This is shown in Figure 43. With the RM LUB test, none of the service sets are feasible to schedule. According to the competition test and scheduling point tests, only service sets 5, 7, 9 are feasible while service sets 6, 8 and 10 are not feasible. This is proven to be true as shown in the Cheddar analysis. Thus, the modified code agrees with Cheddar analysis.

0.0.4 Read Chapter 3 of the textbook.

1. Provide 3 constraints that are made in the RM LUB derivation and 3 assumptions as documented in the Liu and Layland paper and in Chapter 3 of RTECS with Linux and RTOS. Describe whether you think each is

reasonable for actual practice or whether you think each is only applicable to an idealized model of practice.

Constraints made in the RM LUB derivation:

- For all the tasks, the deadline must be equal to the period of the task.
- The tasks are all have fixed priority which cannot be changed during run-time of the scheduler. Further, the tasks should be preemptive and must run to competition. This affirms that deterministic inputs will provide deterministic behaviour.
- Interference is neglected in the derivation. When a context switch occurs and a task is pre-empted prior to completing, this is called interference.

Assumptions made in the Liu and Layland paper- "Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment":

- The requests for all tasks for which hard deadlines exist are periodic, with constant interval between requests
- The tasks are independent in that requests for a certain task do not depend on the initiation or the completion of requests for other tasks (no known phasing).
- Run-time for each task is constant for that task and does not vary with time. Run-time here refers to the time which is taken by a processor to execute the task without interruption.

2. Finally, list three key derivation steps in the RM LUB derivation that you either do not understand or that you would consider "tricky" math. Attempt to describe the rationale for those steps as best you can do based upon reading in Chapter 3 of RTECS with Linux and RTOS.

   Key derivation steps:

   - The necessity for two cases: Case 1- C_1 short enough to fit all three releases in T_2 ie. S_2 critical time zone and case 2- C_1 too large to fit last release in T_2 ie. does not fit in S_2 critical time zone. The explanation of discrete math used, the floor and ceiling of a fraction concepts were made easier as explained in the book. The main concept is that in case 1, there is maximum number of occurrences of S_1 during T_2 and in case 2, it would be the minimum.

     In case 1, all service 1 (S_1) releases requiring execution time fit in T_2. This is shown in Eq. (8). Thus, the length of (C_2) of service 2 is long enough to use all the time not used by service 1. This is shown in Eq. (2).

     $$C_1 <= T_2 - T_1 \lfloor \frac{T_2}{T_1} \rfloor \tag{1}$$

     $$C_2 = T_2 - C_1 \lceil \frac{T_2}{T_1} \rceil \tag{2}$$

     In case 2, the last release of S_1 does not fit into T_2. This spillover condition is expressed as Eq. (3). The execution of service 2 will still complete in time even though S_1 overruns the critical time zone in this case. The execution time (C_2) is simply the sum of all full occurrences of T_1 during T_2 accounted for the amount of time S_1 takes during that duration. This is shown in Eq. (4).

     $$C_1 >= T_2 - T_1 \lfloor \frac{T_2}{T_1} \rfloor \tag{3}$$

     $$C_2 = T_1 \lfloor \frac{T_2}{T_1} \rfloor - C_1 \lfloor \frac{T_2}{T_1} \rfloor \tag{4}$$

   - In the derivation of both the cases, by substituting the appropriate execution times in formula for U (shown in Eq. (5), the following formulae Eq. (6) and Eq. (7) can be derived:

     $$U = \frac{C_1}{T_1} + \frac{C_2}{T_2} \tag{5}$$

     $$U = 1 + C_1 [\frac{1}{T_1} - \frac{\lceil \frac{T_2}{T_1} \rceil}{T_2}] \tag{6}$$

$$U = \frac{C_1}{T_1} + \frac{T_1\lfloor\frac{T_2}{T_1}\rfloor - C_1\lfloor\frac{T_2}{T_1}\rfloor}{T_2} \tag{7}$$

For the equation Eq. (6), it is mentioned that U monotonically decreases with increasing execution time, C_1.

This is clearer when the second term is plotted. By fixing T_1=1, and since T_2 is always greater than T_1, it has a range of (1, infinity). By plotting the second term, we can see that it is a periodic function that oscillates between 1 and 2 as T_2 is increased. This is shown in the graph, Figure 44.

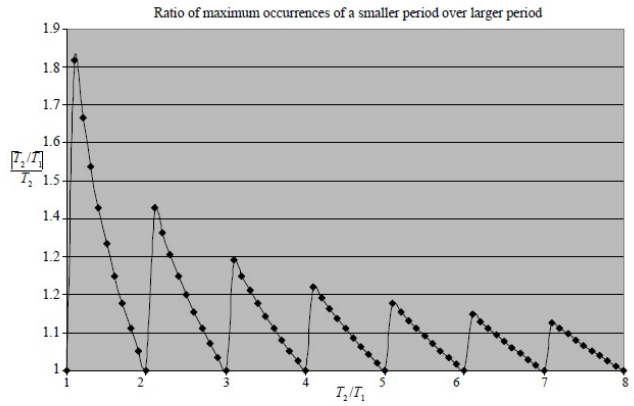

Figure 44: Case 1 Relationship of T_2 and T_1

For the equation Eq. (7), it is mentioned that U monotonically increases with increasing execution time, C_1. Similarly, when the third term is plotted, it is clearer that the periodic function is less than 1 in all cases. This is shown in the graph, Figure 45.

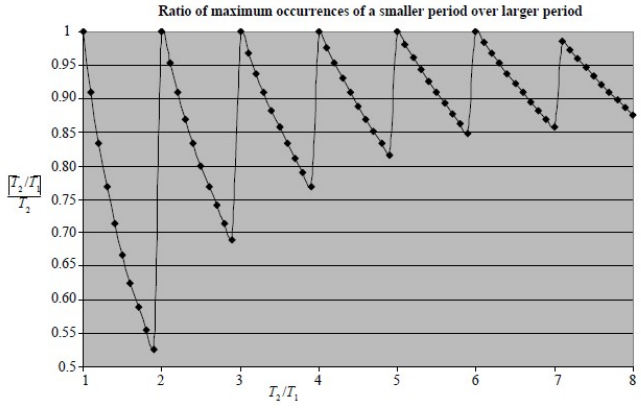

Figure 45: Case 2 Relationship of T_2 and T_1

- The values derived for C_1 are equated as both the cases will only be valid if they intersect. By plugging these C_1 and C_2 values in the formula for U, the following formula was derived.

$$U = 1 - (\frac{T_1}{T_2})[\lceil\frac{T_2}{T_1}\rceil - \frac{T_2}{T_1}][\frac{T_2}{T_1} - \lfloor\frac{T_2}{T_1}\rfloor] \tag{8}$$

However, utility is further derived into a function of whole integer number of interferences and fractional interference. This derivation is vague in the paper but the key is substitution of this formula: ceiling(N.d) = 1 + floor(N.d).