ECE 612
Real-Time Embedded Systems

PROJECT REPORT

ON

EDF AND RATE MONTONIC SCHEDULING IN

MATLAB FOR MULTIPROCESSORS

INSTRUCTOR: **PROFESSOR. A C SABZEVARI**

GEORGE MASON UNIVERSITY

Submitted BY

SAI KASYAP KAMARAJU

G00845717

Abstract:

Application of EDF and Rate monotonic algorithms to perform different tasks for 3 processors in MATLAB is the highlight of this work. Time Optimization of Resources Scheduling (TORSCHE), a special toolbox, which is available online, is made use of for the implementation of this project. Three cases (examples) are taken in to consideration separately for EDF & RM and common tasks for both EDF and RM scheduling. The worst case response time, utilization factor is found out for each task. It is also possible to determine whether the set of tasks are schedulable for a specific scheduling algorithm.

Introduction:

MATLAB is a high-level language and interactive environment for numerical computation, visualization, and programming. The language, tools, and built-in math functions in MATLAB enable us to explore multiple approaches and reach a solution faster than with spreadsheets or traditional programming languages, such as C/C++ or Java. The only disadvantage associated with MATLAB is that it does not have in built MATLAB toolbox which is necessary for Real time Scheduling. This can be overcome by using an online available toolbox.

TORSCHE (Time Optimization, Resources, Scheduling) Scheduling Toolbox for MATLAB is a freely (GNU GPL) available toolbox developed at the Czech Technical University and this is used in this project. TORSCHE is written in object oriented MATLAB Programming Language and provides functions and structures for describing tasks and scheduling problems.

Schedulability is a property indicating whether a real-time system (a set of real-time tasks) can meet their deadlines.

Processor utilization factor:
For a given a set T of n periodic tasks, processor utilization factor, U is defined as the fraction of processor time spent in the execution of the task set:

$$U = \Sigma_{i=1}^{n} \frac{C_i}{T_i}$$

The **Earliest Deadline First (EDF) algorithm** is a dynamic priority-scheduling algorithm in which a task with a shorter deadline has a higher priority. So it executes a job with the earliest deadline. Whenever a scheduling event occurs (a task finishes or a new task is released) the set of tasks ready for execution will be searched for the task closest to its deadline.0

Rate Monotonic algorithm is optimal static-priority scheduling in which a task is assigned priority according to period. In short, a task with a shorter period has a higher priority. So it executes a job with the shortest period first. For RM schedulability analysis, all tasks must be periodic, independent and have deadlines equal to their periods.

IMPELEMENTATION:

1. Firstly, the toolbox is downloaded from the site http://rtime.felk.cvut.cz/scheduling-toolbox/download.php. Then it is added to the MATLAB as per the procedure mentioned in the website.
2. A set of periodic tasks are defined for this project using ptask in MATLAB.
3. There are many properties for a periodic task in MATLAB like processing time, period of task, release date, deadline, due date, weight (priority), dedicated processor etc. All these properties are assigned prior to the implementation of EDF and RM scheduling algorithms.
4. The functions, EDF and RM are defined in my code, so that they could be reused to solve different cases as mentioned.
5. The overall utilization factor is found out by using util() in MATLAB.
6. For plotting the graphs, plot() in MATLAB is used.

EDF Algorithm pseudo code:

```
Priority is assigned to task deadline.
Base period is found as LCM of all tasks periods.
The simulation is ran up to the base periods.
The worst case response of each task is found out in the following way :
For 1 to Base period
For  1 to No number of tasks in the Task set
If offset =0 & exec = 0;
exec = process time
resp = 0;
end if
end for
T. process time left  = exec;
Then the task is selected to executed if  T.Ready = T.ProcTimeLeft;
Increment resp;
If exec (task)  = 0;
Then worst case response, Wc=  maximimum(resp(task), Wc(task))
end if and for
Scheduled parameters are added in the form of a array up to until the simulation of base period is
complete.
The utilization factor is found using util(T)
If the worst case response > deadline;
Then task set is not schedulable;
Else
Schedulable;
```

Case 1:

Implementation of EDF algorithm for multiprocessors when Periods are not equal to deadlines

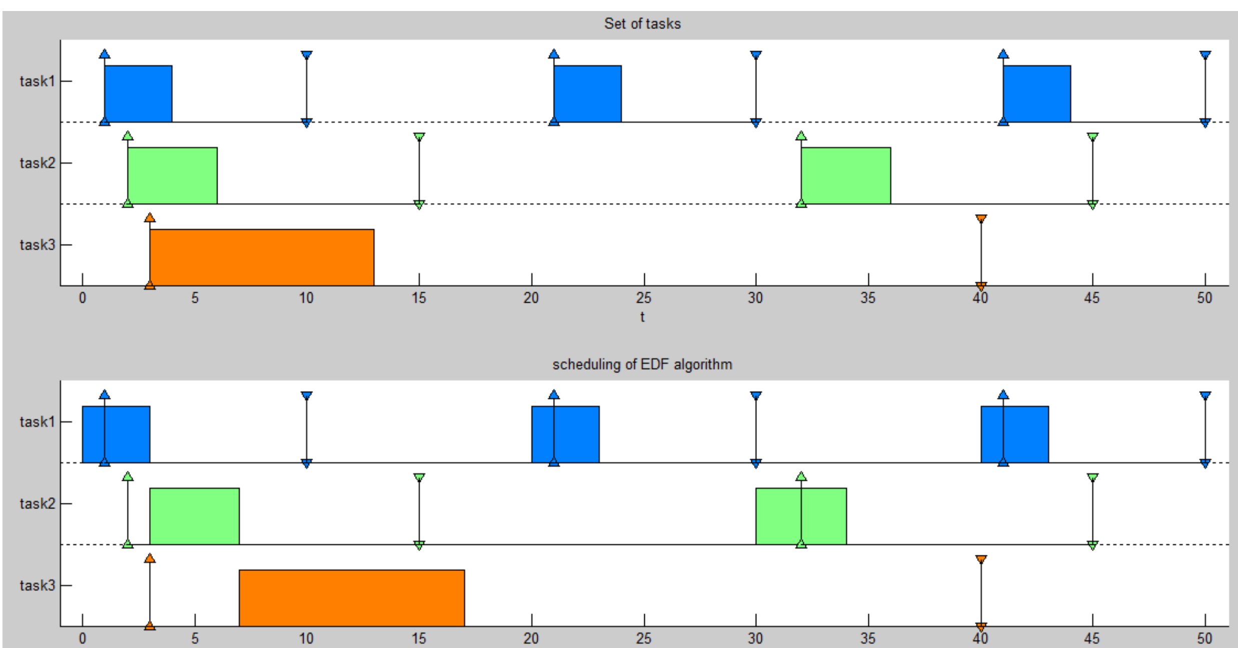
Task set details:

Name	processing time	Period	Release Time	Dead line	Due Date	Weight	dedicated processor
Task1	3	20	1	10	10	0	0
Task2	4	30	2	15	15	0	1
Task3	10	60	3	40	40	0	2

Output :

Base period is 60
Processor utilization is 0.4500
Worst-case response times for each task:
task1 : 3
task2 : 7
task3 : 17
Task set is schedulable

Plot :



Rate Monotonic Scheduling pseudo code:

The only difference between the codes of EDF and RM is priority is assigned to periods instead of deadlines in RM. Rest is the same.

Case 2 :

Implementation of RM algorithm for multiprocessors When Periods are not equal to deadlines.

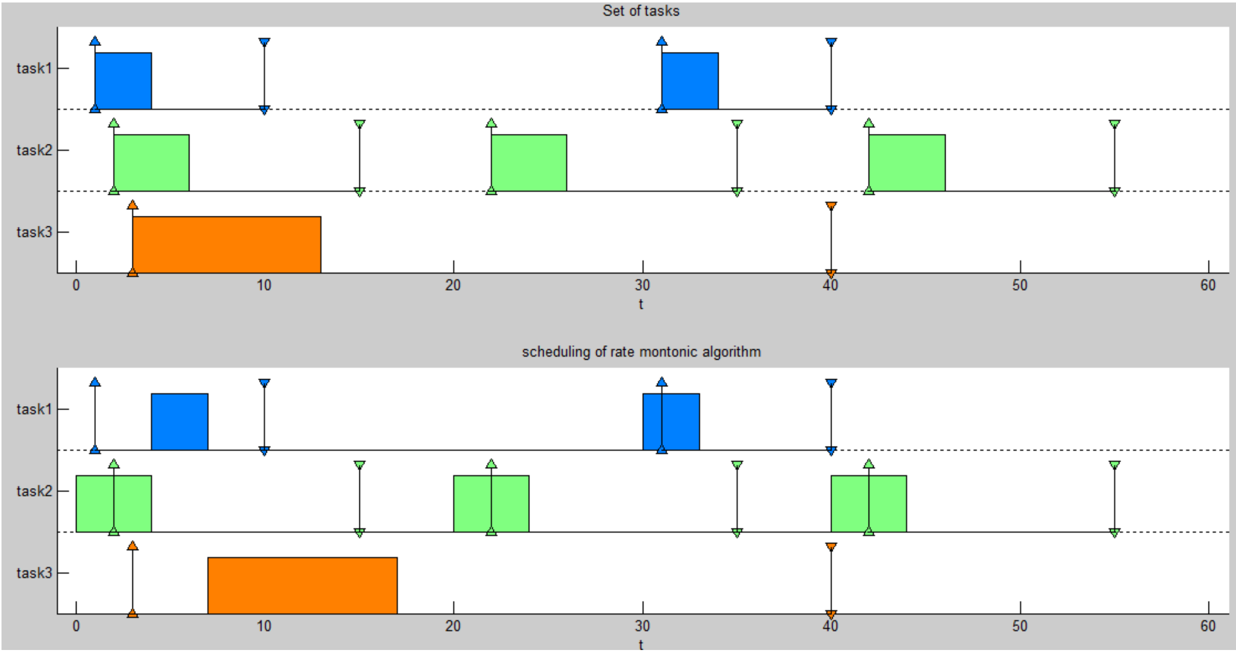
Task set details:

Name	processing time	Period	Release Time	Dead line	Due Date	Weight	dedicated processor
Task1	3	30	1	10	10	0	0
Task2	4	20	2	15	15	0	1
Task3	10	60	3	40	40	0	2

Plot :

Base period is 60
Processor utilization is 0.4667
Worst-case response times for each task:
task1 : 7
task2 : 4
task3 : 17
Task set is schedulable

Plot :



Case 3 :

Implementation of RM and EDF algorithm where periods are equal to deadlines. Then EDF and RM algorithm do the same function.

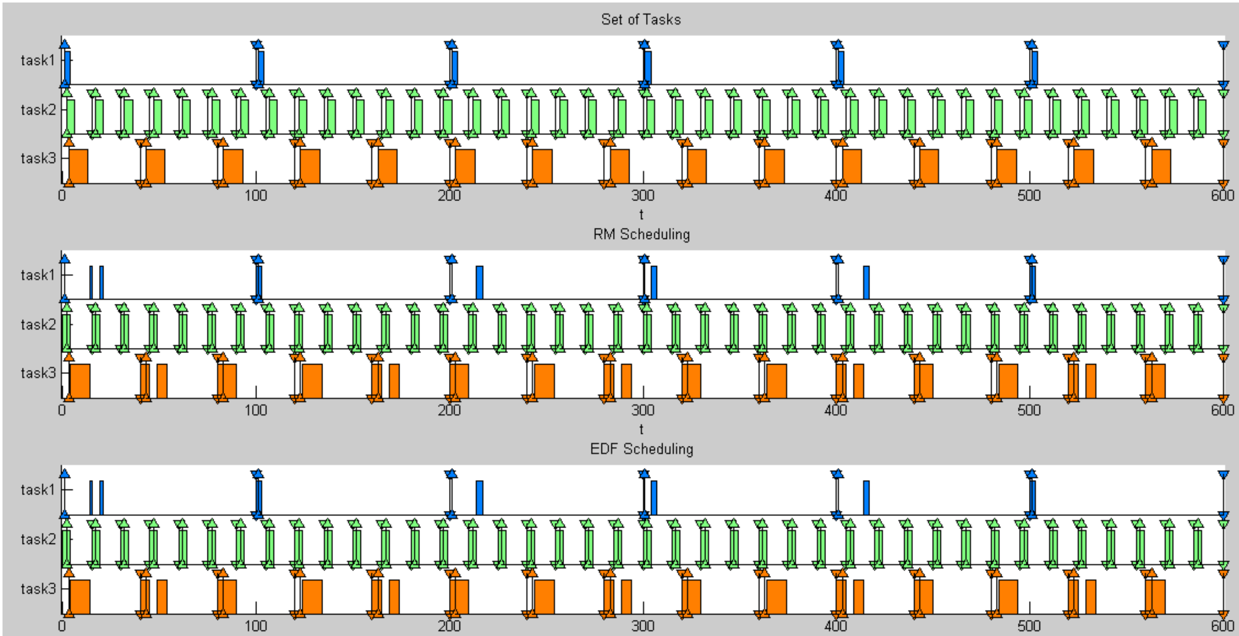
Task set Details :

Name	processing time	Period	Release Time	Dead line	Due Date	Weight	dedicated processor
Task1	3	100	1	100	100	0	0
Task2	4	15	2	15	15	0	1
Task3	10	40	3	40	40	0	2

Output :

Base period is 600
Processor utilization is 0.5467
Worst-case response times for each task:
task1 : 21
task2 : 4
task3 : 14
Task set is schedulable

Plot:



Challenge Faced:

- The toolbox which is used for real time scheduling TORSCHE is not a full-fledged toolbox for real time scheduling as it is less popular due to which problem was faced in coding the algorithms.

Comparison between Doing the same project in MATLAB and Java (if done):

In MATLAB	In Java
There is no threading required.	Multi-threading concepts are required and each thread is treated as a separate process.
There is a function to define task which is easier to implement.	Task is defined as a program which takes certain time to execute. (Need to do 3 different programs for 3 different tasks).
Each task has specific set of properties which are user defined.	This is difficult as we cannot set the deadlines specifically on our own (program dependent) and for running periods we need to use looping again.
The coding is easier and shorter in MATLAB than Java and computation of various things such as utilization factor etc. is easier as there are many Math functions defined in MATLAB.	The code is lengthy (complex) and computation of utilization factor and other functions is tough to calculate.
All tasks and scheduling of various algorithms can be easily plotted in MATLAB.	Difficult to plot outputs in Java.
MATLAB supports real time scheduling only when TORSCHE toolbox is added to it.	Java does not support real time scheduling unless it is done say using real java programming packages and then doing it in FLEX compiler.

Conclusion:

In each case, the utilization factors and worst case response times (for each task) are found.
In the each case Utilization factor U

$\sum U_i \leq n\,(2^{1/n}-1)<1$ satisfies the criteria (n=3 here).

In all the cases, the task set is made sure that it is schedulable by assigning variables to the tasks.

References:

http://pundit.pratt.duke.edu/wiki/MATLAB:User-defined_Function#Naming_and_Calling_Functions
http://rtime.felk.cvut.cz/scheduling-toolbox/manual/
www.mathworks.com
http://www.researchgate.net/publication/221224279_Worst-Case_Response_Time_Analysis_of_Real-Time_Tasks_under_Fixed-Priority_Scheduling_with_Deferred_Preemption_Revisited
http://www.win.tue.nl/san/people/rbril/publications/bvl-ecrts-wip04.pdf
http://en.wikipedia.org/
http://www.fi.muni.cz/~xpelanek/IA158/slides/periodic.pdf