

Програмирање у реалном времену

Извјештај

Задатак 1а:

Нераспоредив скупу задатака и за **RM** и за **EDF** алгоритам.

Задатак	Период	Вријеме извршавања	Приоритет	Искориштеност
A	20	4	4	0.2
B	15	12	3	0.8
C	10	4	2	0.4
D	8	6	1	0.75

Лако се показује да скуп задатака није распоредив ни по једном од ова два алгоритма. Наиме, фактор искориштења процесора за дати скуп задатака је:

$$U = 0.2 + 0.8 + 0.4 + 0.75 = 2.15$$

За **EDF** алгоритам потребан и довољан услов за распоредивост је $U \leq 1$, а како овај услов није испуњен, то ни скуп задатака није распоредив **EDF** алгоритмом.

Слично се показује и за **RM** алгоритам. Услов распоредивност је да је фактор искориштености процесора мањи од доње границе која се рачуна као:

$$U_S = n * (\sqrt[n]{2} - 1)$$

У овом случају доња граница је:

$$U_S = 4 * (\sqrt[4]{2} - 1) = 4 * (1.189 - 1) = 0.756$$

Ово је довољан услов за распоредивност односно скуп задатака може бити распоредив чак и ако овај услов није испуњен. Ако се изврши анализа времена одзива добија се следеће:

$$w_C^0 = 4$$

$$w_C^1 = 4 + \left\lceil \frac{4}{8} \right\rceil * 6 = 4 + 6 = 10$$

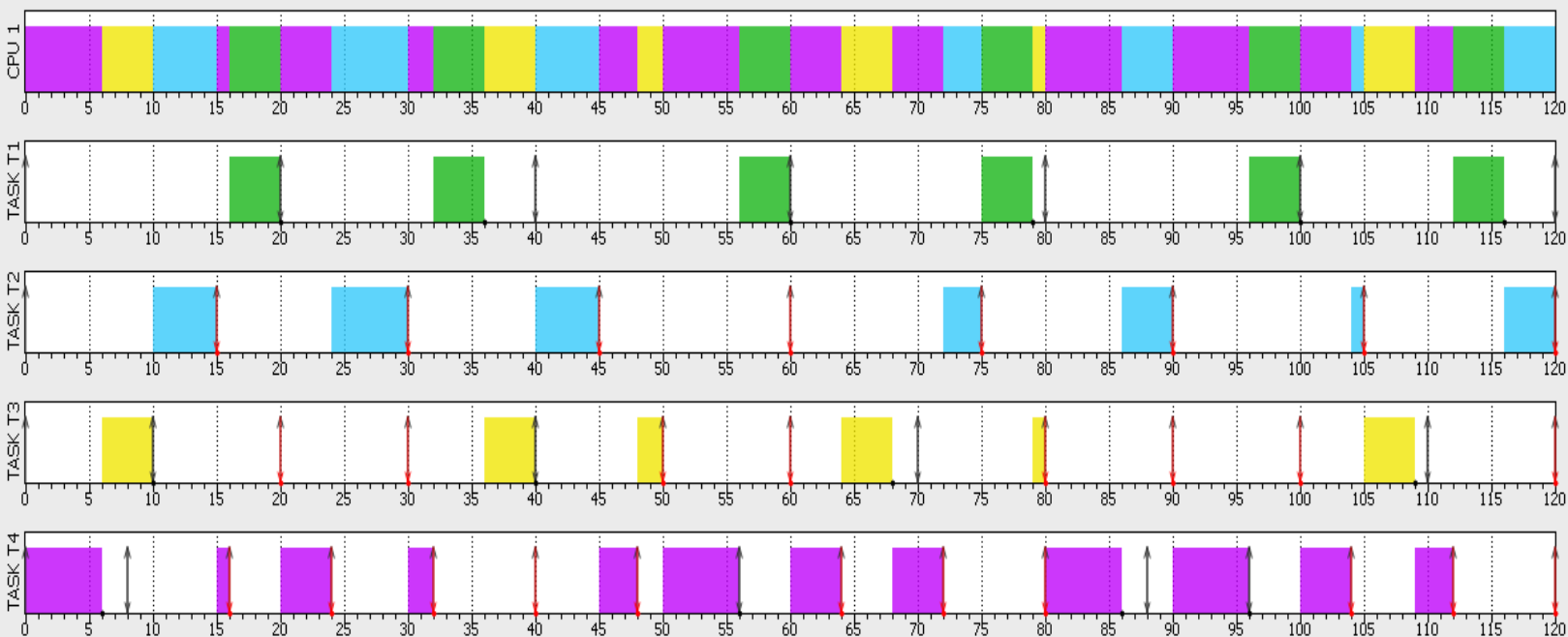
$$w_C^2 = 4 + \left\lceil \frac{10}{8} \right\rceil * 6 = 4 + 12 = 16$$

$$w_C^3 = 4 + \left\lceil \frac{16}{8} \right\rceil * 6 = 4 + 12 = 16$$

$$\Rightarrow R_C = w_C^3 = 16 > T_C$$

Овим је показано да је најгори случај времена одзив задатка C већи од његовог апсолутног рока (периода) што значи да овај задатак није распоредив, а самим тим ни овај скуп задатака (није потребно ни вршити анализу за задатке A и B).

Хиперпериод се рачуна као најмањи заједнички дјелилац периода свих задатака из скупа. У овом случају хиперпериод је 120.



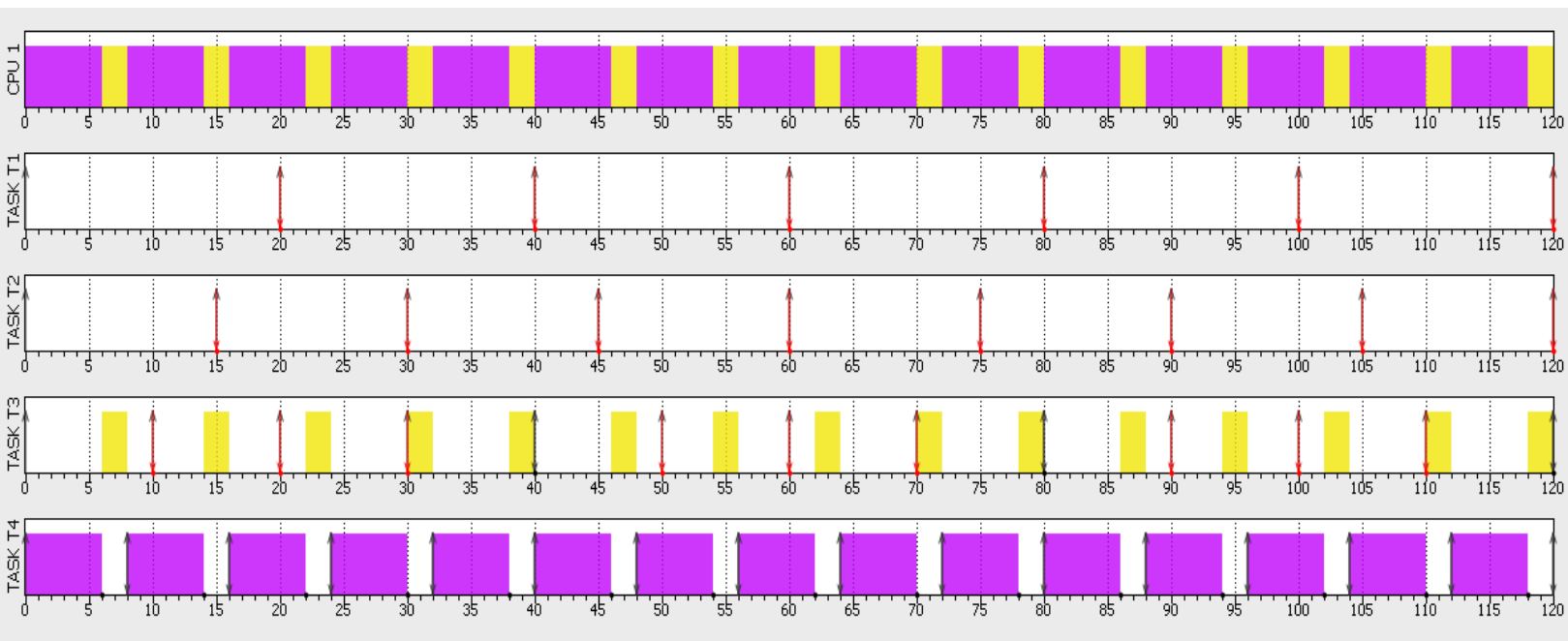
Слика 1a.1 Гантов дијаграм за **EDF** алгоритам

Са дијаграма се јасно види да задаци D (TASK T4), C (TASK T3) и B (TASK T2) по више пута пробију рокове и не изврше се до краја. Задатак A (TASK T1) никако не пробија рок што би се могло објаснити тиме што има мало вријеме извршавања у односу на период односно рок.

Observation Window:			
from 0.00 to 120.00 ms		Configure...	
	Total load	Payload	System load
CPU 1	1.0000	1.0000	0.0000
Average	1.0000	1.0000	0.0000

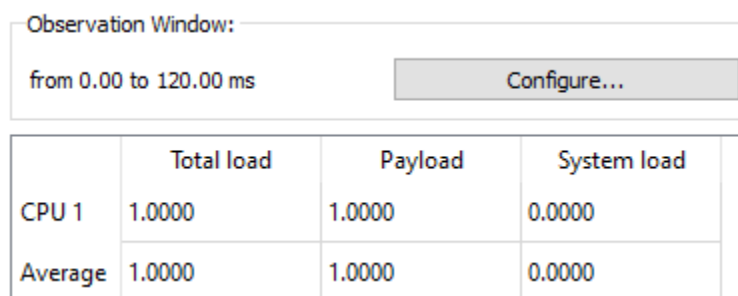
Слика 1a.2 Оптерећење процесора

Оптерећење процесора на основу извршавања у току једног хиперпериода је очекивано 100 %.



Слика 1а.3 Гантов дијаграм за **RM** алгоритам

Са дијаграма се види да се задатак *D* (TASK T4) увијек извршава у року, али остали задаци не. Задаци *A* (TASK T1) и *B* (TASK T2) чак никада и не добију процесорско вријеме. Ово се дешава због тога што задатак *D* има највиши приоритет, а и велику искориштеност процесора, па због тога „гуши“ задатке нижег приоритета.



Слика 1а.4 Оптерећење процесора

Као и у претходном случају оптерећење процесора је 100 %.

Задатак 16:

Нераспоредив скуп задатак за **RM** алгоритам, али распоредив за **EDF** алгоритам.

Задатак	Период	Вријеме извршавања	Приоритет	Искориштеност
A	40	10	4	0.25
B	24	6	3	0.25
C	20	6	2	0.3
D	10	2	1	0.2

Фактор искориштености процесора за дати скуп је:

$$U = 0.25 + 0.25 + 0.3 + 0.2 = 1$$

Закључујемо да је скуп задатака сигурно распоредив **EDF** алгоритмом.

Са друге стране за **RM** алгоритам, доња граница је као у првом случају $U_S = 0.756$ па скуп задатака не пролази тест распоредивости, па је потребно извршити анализу времена одзива.

За задатак *D* нема нити један задатак вишег приоритета, па је најгоре вријеме одзива једнако његовом времену извршавања, а то је 2, мање од периода, па је овај задатак сигурно распоредив.

За задатак *C*:

$$w_C^0 = 6$$

$$w_C^1 = 6 + \left\lceil \frac{6}{10} \right\rceil * 2 = 6 + 2 = 8$$

$$w_C^2 = 6 + \left\lceil \frac{8}{10} \right\rceil * 2 = 6 + 2 = 8$$

$$\Rightarrow R_C = w_C^2 = 8 < T_C = 20$$

Најгоре вријеме одзива задатка *C* је мање од периода (рока) па је овај задатак распоредив.

За задатак *B*:

$$w_B^0 = 6$$

$$w_B^1 = 6 + \left\lceil \frac{6}{20} \right\rceil * 6 + \left\lceil \frac{6}{10} \right\rceil * 2 = 6 + 6 + 2 = 14$$

$$w_B^2 = 6 + \left\lceil \frac{14}{20} \right\rceil * 6 + \left\lceil \frac{14}{10} \right\rceil * 2 = 6 + 6 + 4 = 16$$

$$w_B^3 = 6 + \left\lceil \frac{16}{20} \right\rceil * 6 + \left\lceil \frac{16}{10} \right\rceil * 2 = 6 + 6 + 4 = 16$$

$$\Rightarrow R_B = w_B^3 = 16 < T_B = 24$$

И задатак B пролази тест распоредивости.

За задатак A :

$$w_A^0 = 10$$

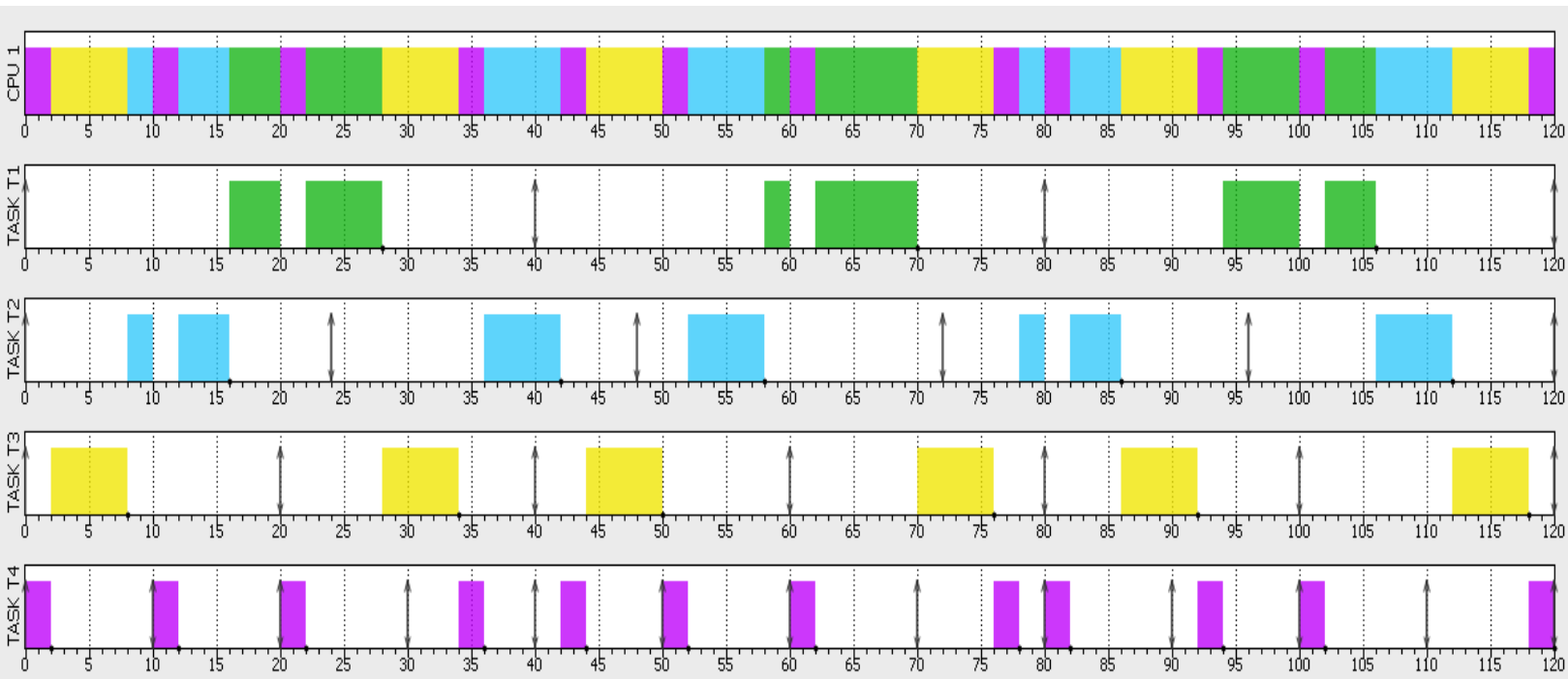
$$w_A^1 = 10 + \left\lceil \frac{10}{24} \right\rceil * 6 + \left\lceil \frac{10}{20} \right\rceil * 6 + \left\lceil \frac{10}{10} \right\rceil * 2 = 10 + 6 + 6 + 2 = 24$$

$$w_A^2 = 10 + \left\lceil \frac{24}{24} \right\rceil * 6 + \left\lceil \frac{24}{20} \right\rceil * 6 + \left\lceil \frac{24}{10} \right\rceil * 2 = 10 + 6 + 12 + 6 = 34$$

$$w_A^3 = 10 + \left\lceil \frac{34}{24} \right\rceil * 6 + \left\lceil \frac{34}{20} \right\rceil * 6 + \left\lceil \frac{34}{10} \right\rceil * 2 = 10 + 12 + 12 + 8 = 42$$

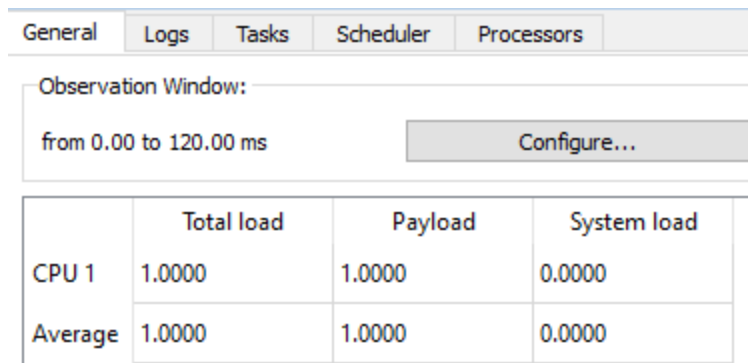
Пошто знамо да је скуп вриједности $w_i^0, w_i^1, \dots, w_i^n$ монотono неопадајући, а како је већ $w_A^3 = 42$ веће од периода задатка A , то значи да је и најгоре вријеме одзива сигурно веће од периода овог задатка, што опет даље значи да овај задатак није распоредив, па ни сам скуп задатака није распоредив **RM** алгоритмом.

Хиперпериод за дати скуп задатака је 120.

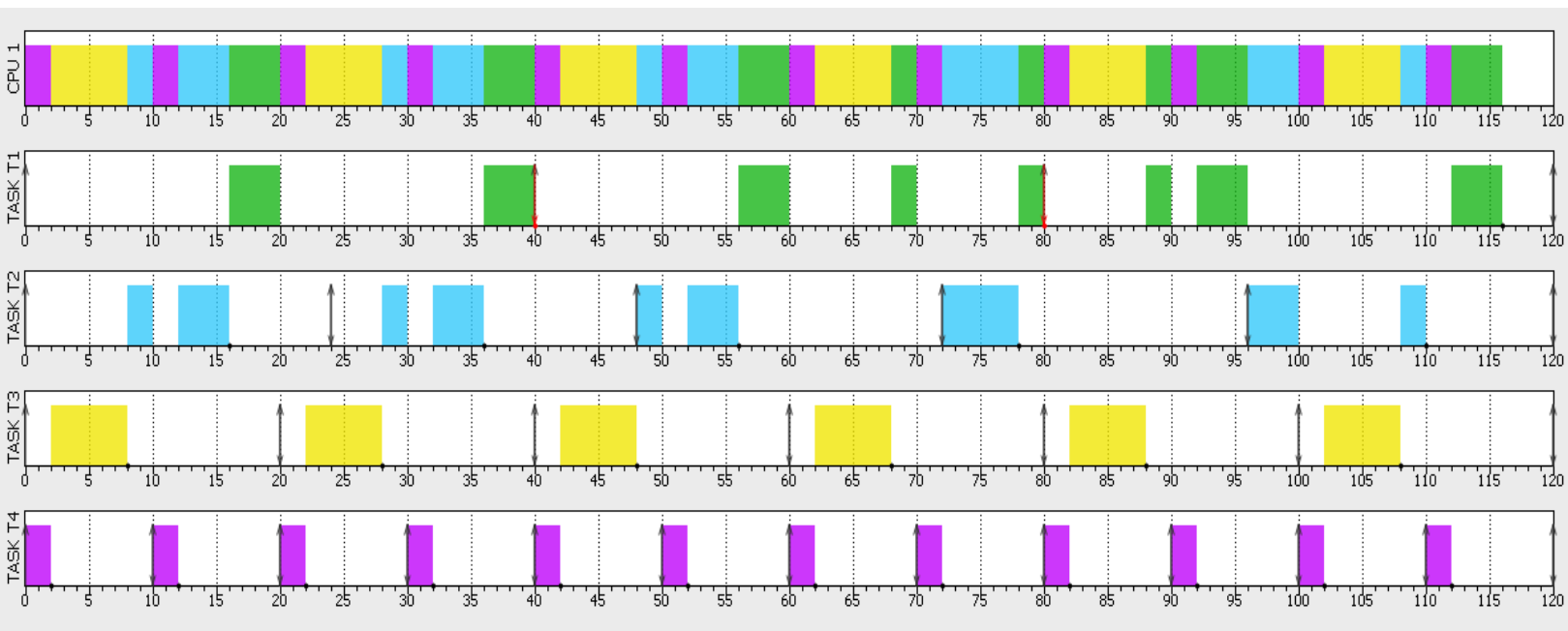


Слика 16.1 Гантов дијаграм за **EDF** алгоритам

Са Гантовог дијаграма се јасно види да нити један задатак не пробија рок, чиме се потврђује распоредивост скупа задатака. Такође, види се да је искориштеност процесора једнака 100 % тј. процесор није никако у *idle* стању.



Слика 16.2 Оптерећење процесора



Слика 16.3 Гантов дијаграм за **RM** алгоритам

Са овог дијаграма се види да задатак *A* (*TASK T1*) пробија рок у два наврата, чиме се потврђује претходно спроведена анализа. Другим ријечима, задаци *B* (*TASK T2*) и *C* (*TASK T3*) су распоредиви, али дати скуп није распоредив јер задатак *A* није распоредив.

General	Logs	Tasks	Scheduler	Processors				
General	TASK T1	TASK T2	TASK T3	TASK T4				
Activation	Start	End	Deadline	Comp. time	Resp. time	CPI	Preemptions	Migrations
0.0000	0.0000	40.0000	40.0000	8.0000	40.0000		1	0
40.0000	40.0000	80.0000	80.0000	8.0000	40.0000		2	0
80.0000	80.0000	116.0000	120.0000	10.0000	36.0000		2	0
120.0000	120.0000		160.0000				0	0

Слика 16.4 Задатак A

Observation Window:			
from 0.00 to 120.00 ms		Configure...	
	Total load	Payload	System load
CPU 1	0.9667	0.9667	0.0000
Average	0.9667	0.9667	0.0000

Слика 16.5 Оптерећење процесора

Иако се за оптерећење процесора добија вриједност мања од 1, скуп задатака није распоредив **RM** алгоритмом.

Задатак 1ц:

Распоредив скуп задатака по оба алгоритма.

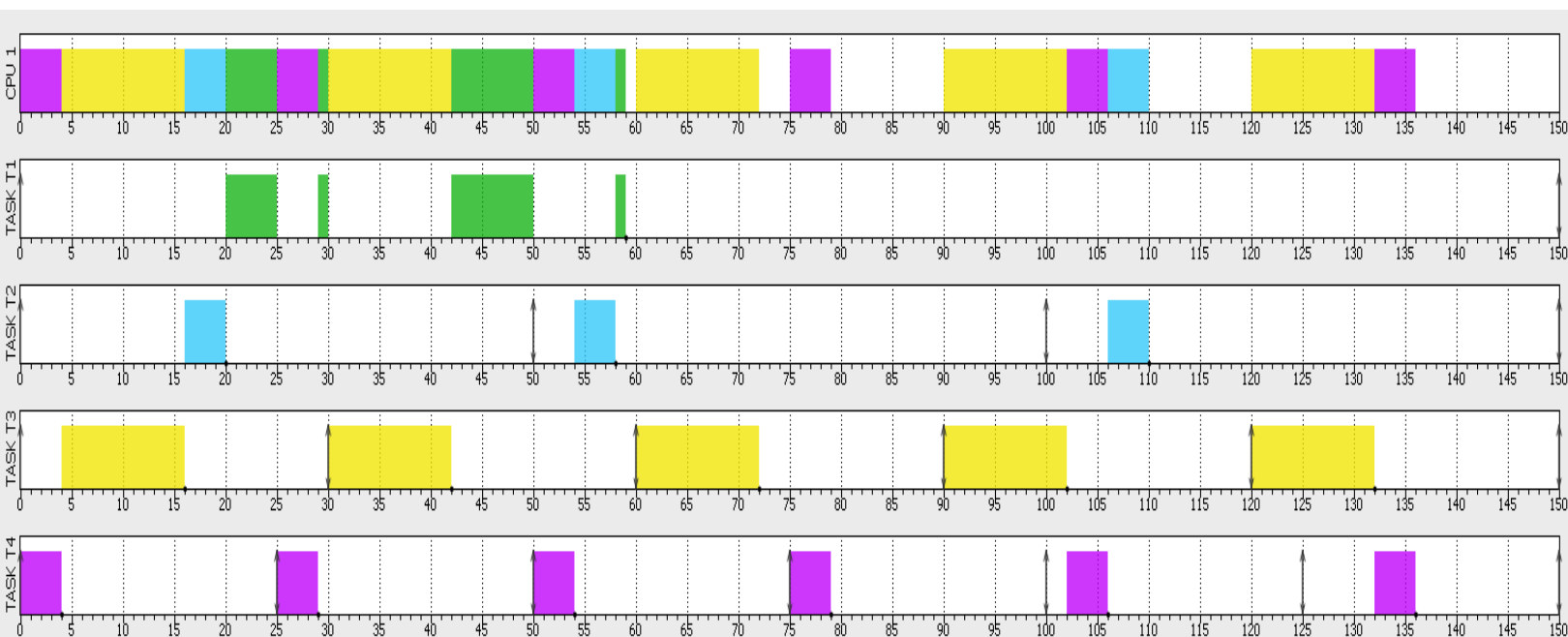
Задатак	Период	Вријеме извршавања	Приоритет	Искориштеност
A	150	15	4	0.1
B	50	4	3	0.08
C	30	12	2	0.4
D	25	4	1	0.16

Фактор искориштености процесора у овом случају је:

$$U = 0.1 + 0.08 + 0.4 + 0.16 = 0.74$$

Фактор искориштености је дакле мањи од 1 (скуп задатака је сигурно распоредив **EDF** алгоритмом), али мањи је и од доње границе за **RM** алгоритам која износи $U_S = 0.756$, што даље значи да је скуп сигурно распоредив и овим алгоритмом.

Хиперпериод за дати скуп задатака износи 150.



Слика 1ц.а Гантов дијаграм за **EDF** алгоритам

Јасно се са дијаграма види да сви задаци испуњавају рокове.

На слици 1ц.2 приказано је оптерећење процесора на хиперпериоду и показано да оно стварно износи 74 %, као што је и очекивано.

zadatak_1_c_EDF.xml			
General	Logs	Tasks	Scheduler
Processors			
Observation Window:			
from 0.00 to 150.00 ms			
	Total load	Payload	System load
CPU 1	0.7400	0.7400	0.0000
Average	0.7400	0.7400	0.0000

Слика 1ц.2 Оптерећење процесора

Response time:				
Task	min	avg	max	std dev
TASK T1	59.000	59.000	59.000	0.000
TASK T2	8.000	12.667	20.000	5.249
TASK T3	12.000	12.800	16.000	1.600
TASK T4	4.000	5.500	11.000	2.566

Слика 1ц.3 Времена одзива (*EDF*)

На слици 1ц.3 приказану су времена одзива (минимално, просјечно и максимално) за дати скуп задатака. Очекивано је да задатак *A* (*TASK T1*) има највеће вријеме одзива, због његовог великог периода у односу на друге задатке. Просјечно вријеме одзива задатка *C* (*TASK T3*) је веће од просјечног времена одзива задатка *B* (*TASK T2*) што се дешава због његовог дужег времена извршавања. Задатак *A* (*TASK T1*) очекивано има најкраће просјечно вријеме одзива, што због најкраћег периода, што због кратког времена извршавања.

Вријеме одзива се рачуна као:

$$R_{i,k} = f_{i,k} - a_{i,k}$$

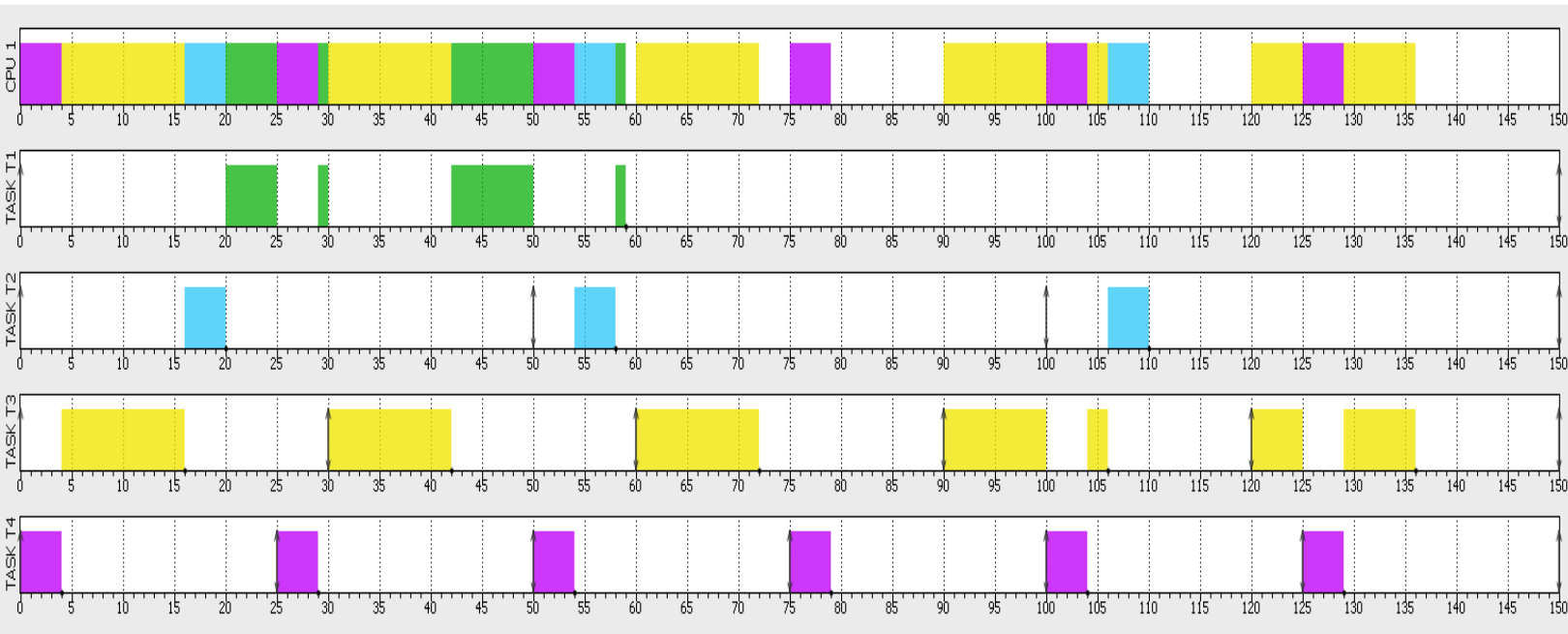
Времена одзива по инстанцама:

$$R_{A1} = 59$$

$$R_{B1} = 20, R_{B2} = 8, R_{B3} = 10$$

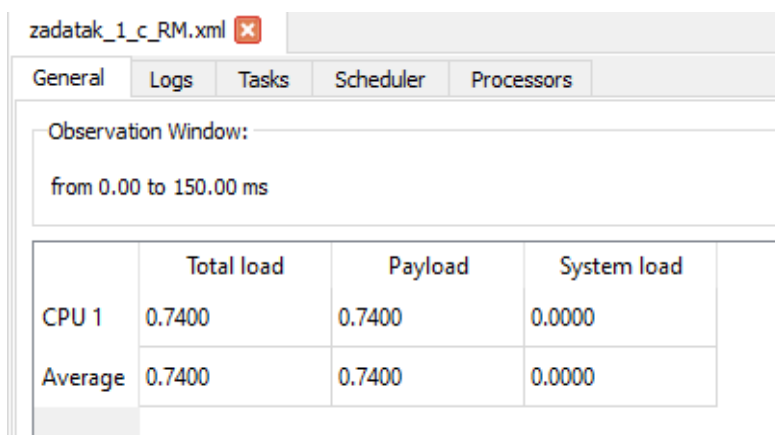
$$R_{C1} = 16, R_{C2} = 12, R_{C3} = 12, R_{C4} = 12, R_{C5} = 12$$

$$R_{D1} = 4, R_{D2} = 4, R_{D3} = 4, R_{D4} = 4, R_{D5} = 6, R_{D6} = 11$$



Слика 1ц.4 Гантов дијаграм за **RM** алгоритам

И у овом случају се потврђује распоредивост скупа задатака.



Слика 1ц.5 Оптерећење процесора

Оптерећење процесора је исто као и у случају **EDF** алгоритма, износи 74 %, како је и очекивано.

Response time:				
Task	min	avg	max	std dev
TASK T1	59.000	59.000	59.000	0.000
TASK T2	8.000	12.667	20.000	5.249
TASK T3	12.000	14.400	16.000	1.960
TASK T4	4.000	4.000	4.000	0.000

Слика 1ц.6 Времена одзива

Јасно се види да задатак највишег приоритета има најкраће могуће вријеме одзива. Задатак *C* има веће просјечно вријеме одзива од задатка *B* чак и ако има већи приоритет, али то се дешава због дужег времена извршавања.

Времена одзива по инстанцама:

$$R_{A1} = 59$$

$$R_{B1} = 20, R_{B2} = 8, R_{B3} = 10$$

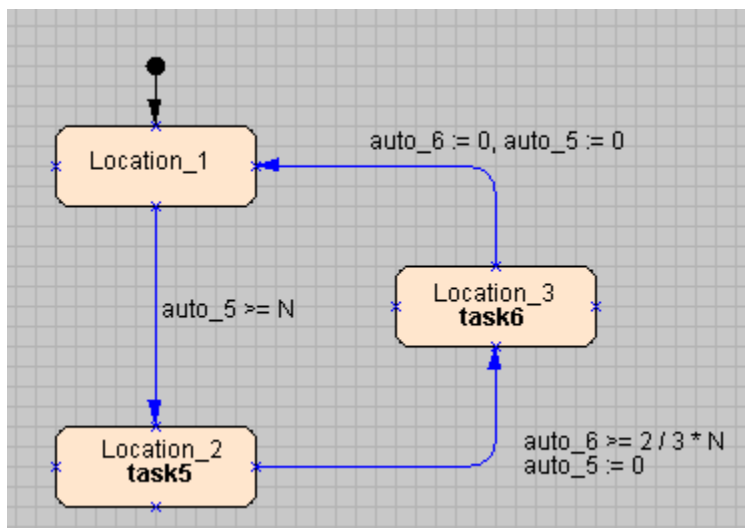
$$R_{C1} = 16, R_{C2} = 12, R_{C3} = 12, R_{C4} = 16, R_{C5} = 16$$

$$R_{D1} = 4, R_{D2} = 4, R_{D3} = 4, R_{D4} = 4, R_{D5} = 4, R_{D6} = 4$$

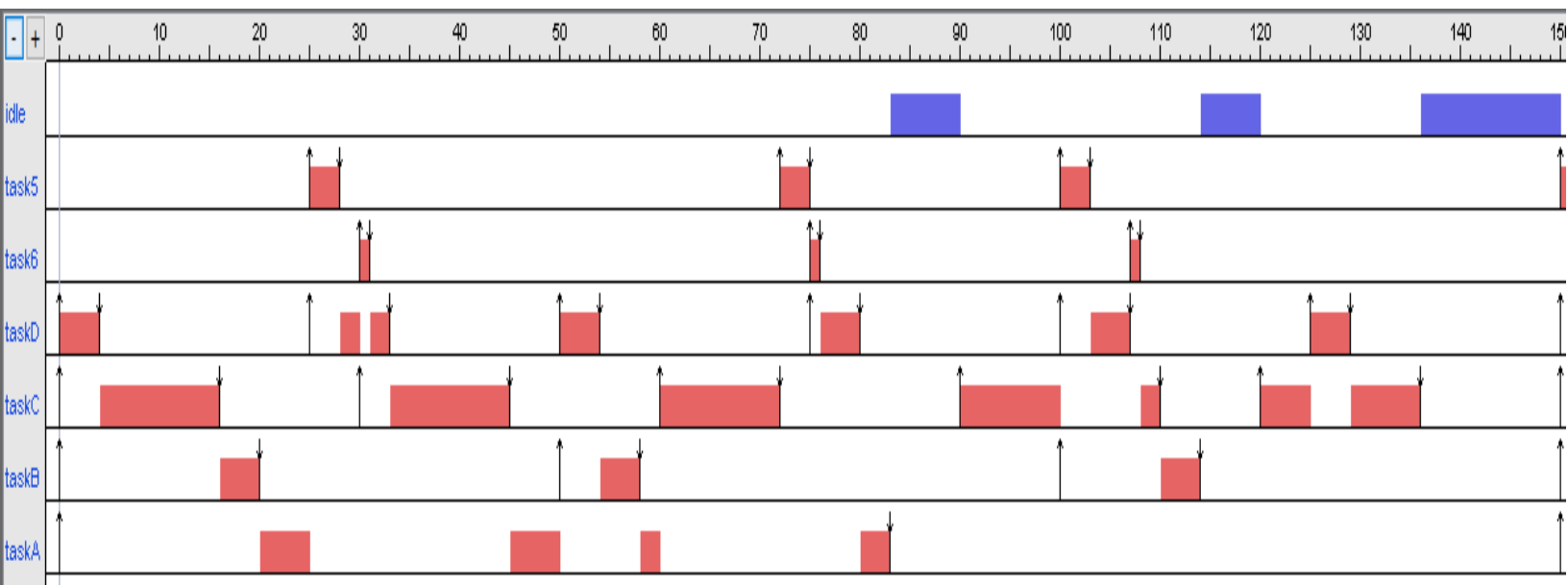
Задатак 2:

Поред периодичних задатака из претходног примјера, овде се појављују и два аperiодична задатка:

Задатак	Вријеме извршавања	Рок
task5	3	15
task6	1	4



Слика 2.1 Машина стања



Слика 2.2 Резултат симулације

За распоређивање периодичних задатака се користи **RM** алгоритам. У овом случају не постоји зависност између задатака и дати скуп је распоредив. Времена одзива су нешто дужа управо због аperiодичних задатака, који сада имају виши приоритет.

Времена одзива по инстанцама:

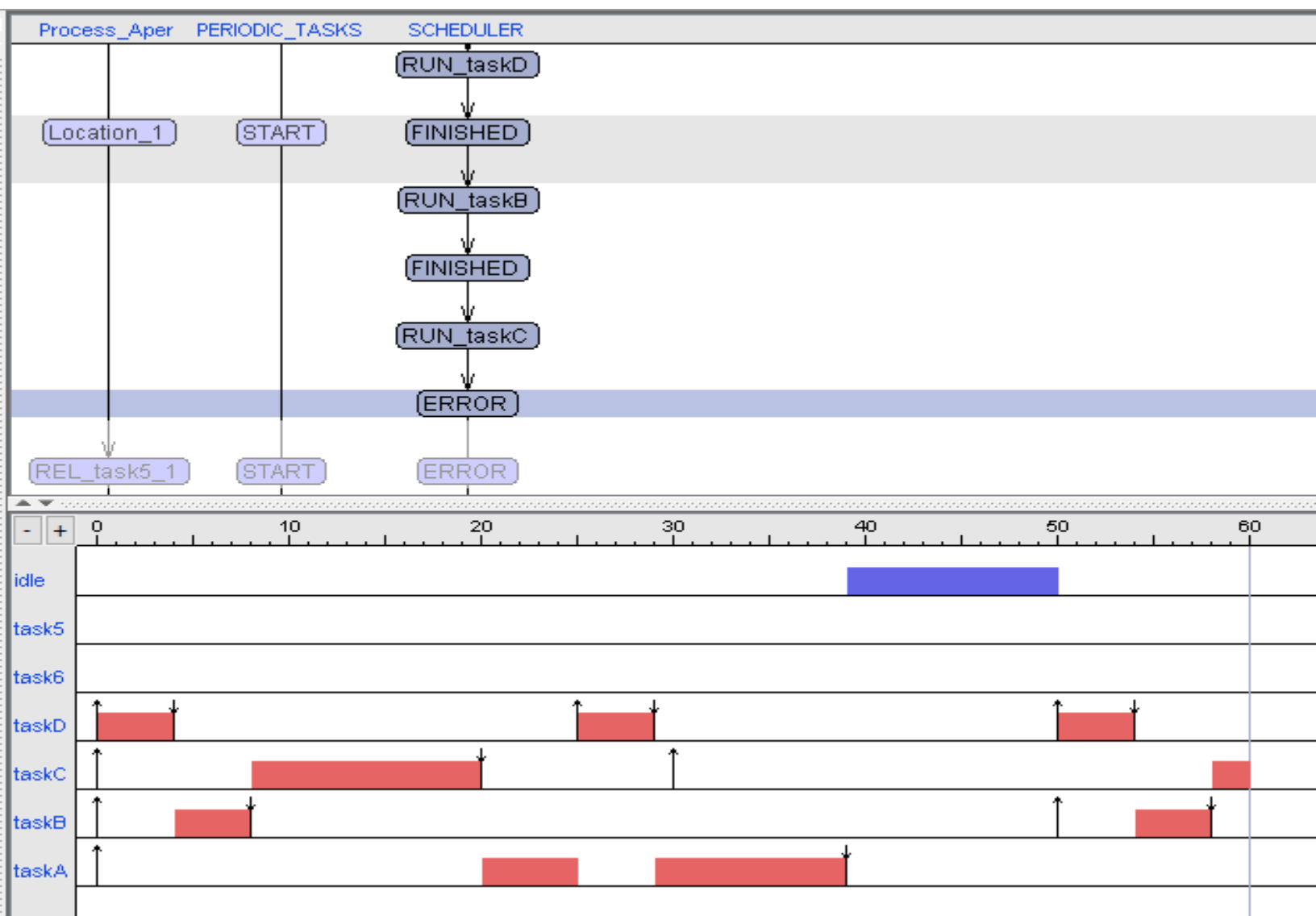
$$R_{A1} = 83$$

$$R_{B1} = 20, R_{B2} = 8, R_{B3} = 14; R_{AVG} = 14$$

$$R_{C1} = 16, R_{C2} = 15, R_{C3} = 12, R_{C4} = 20, R_{C5} = 16; R_{AVG} = 15.8$$

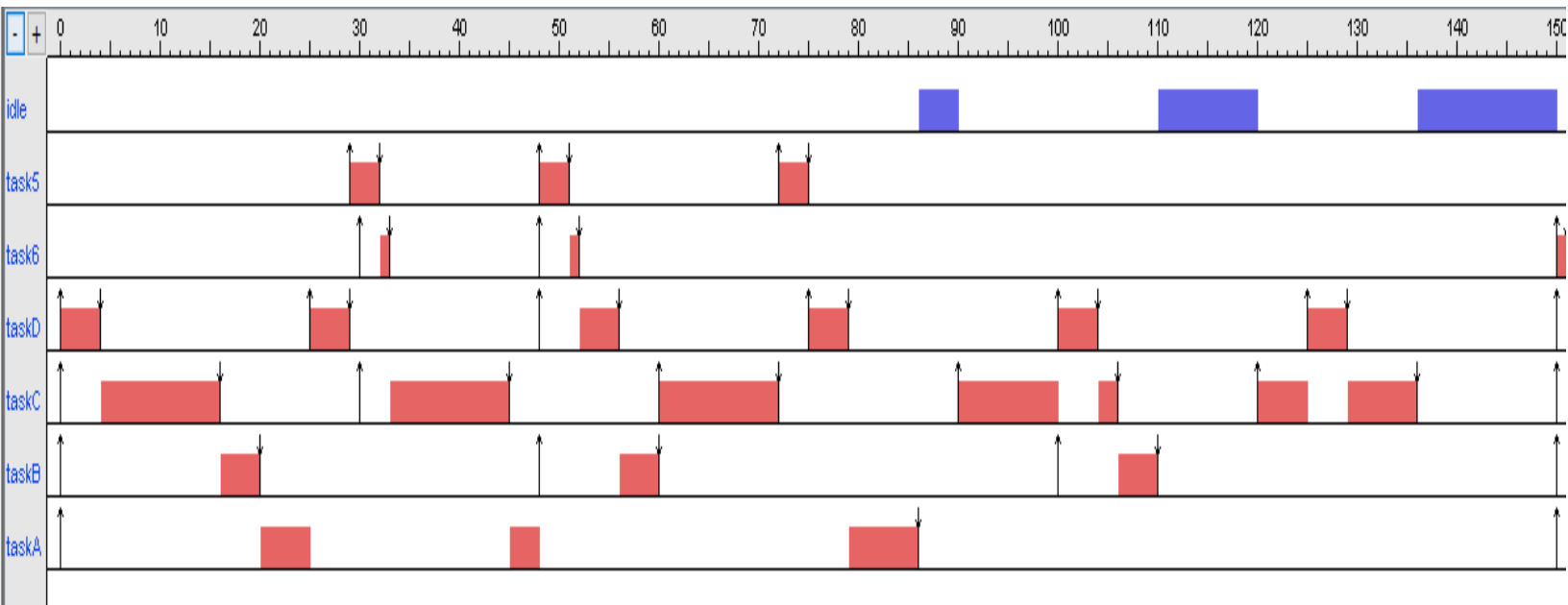
$$R_{D1} = 4, R_{D2} = 8, R_{D3} = 4, R_{D4} = 5, R_{D5} = 7, R_{D6} = 4; R_{AVG} = 5.33$$

Што се тиче времена одзива аperiодичних задатака, у конкретном примјеру увијек је једнако времену извршавања тих задатака.



Слика 2.3 Резултат симулације (задатак C зависи од задатка B)

Сада овај скуп задатака није распоредив. Задатак C има виши приоритет од задатка B , али и зависи од њега, односно не може да почне да се извршава док се задатак B не заврши. Како је период задатка B мањи, очигледно је да ће у неком тренутку доћи до тога да задатак C пробије рок.



Слика 2.4 Резултат симулације (задатак B зависи од задатка C)

Сада задатак нижег приоритета зависи од задатка вишег приоритета, што у конкретном случају није проблем, јер по **RM** алгоритму, задатак вишег приоритета би се свакако извршавао прије задатка нижег приоритета.