Немања Мичић 1199/18

ПРОГРАМИРАЊЕ У РЕАЛНОМ ВРЕМЕНУ

~Лабораторијска вјежба 3~

Задатак 1.

Распоређивање периодичних, независних задатка може да се изврши на три начина, то су: статичко распоређивање, распоређивање са динамичким приоритетима (ЕДФ) и распоређивање са статичким приоритетима (РМ).

Распоређивање са динамичким приоритетима (Earliest Deadline First), сваком задатку динамички додјељује приоритет на основу његовог апсолутног рока извршавања. Задатак који има најкраћи апсолутни рок извршавања добија највећи приоритет и процесорско вријеме. Овај алгоритам је алгоритам са преузимањем, тј један задатак вишег приоритета може да прекине други задатак нижег приоритета.

Услов распоредивости ЕДФ алгоритма је да је искоришћеност процесора U <=1. Ово је потребан и довољан услов распоредивости.

Искоришћеност процесора U се дефинише као U = $\sum_{i=1}^{n} Ci/Ti$.

- n - је број периодичних задатака у скупу задатака
- Ci – вријеме извршавања и-тог задатка
- Ti – период и-тог задатка

RM (Rate Monotonic) алгоритам је алгоритам базиран на фиксним приоритетима, који се периодичним задацима додјељују на основу њиховог периода понављања. Задатак који се чешће понавља тј има мањи период, ће добити већи приоритет и процесорско вријеме. Алгоритам је такође preemptive, тј задатак вишег приоритета може да прекине извршавање задатка нижег приоритета.

Услов распоредивости РМ алгоритма је да искоришћеност процесора мора бити мања од неке границе која је дефинисана формулом Us = n($\sqrt[n]{2}$ – 1). Гдје број n, представља број задатака у скупу задатака који се распоређују. Ово је потребан али не и довољан услов.

а)

У овом примјеру је било потребно одабрати скуп независних периодичних задатка који није распоредив ни ЕДФ ни РМ алгоритмом.

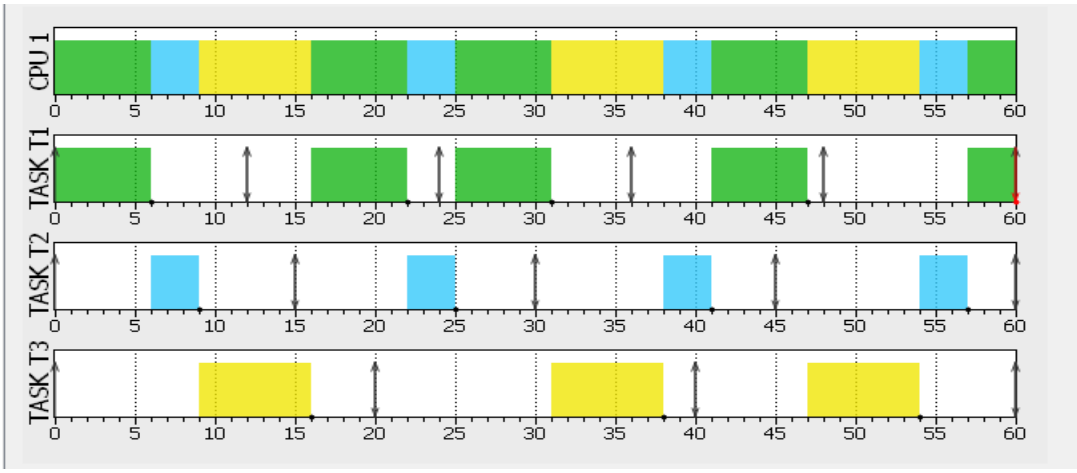
За све примјере је кориштено распоређивање на једном процесору, и претпоставили смо да се сви задаци активирају у истом временском тренутку тј да је ai = 0.

Хиперпериод представља најмањи заједнички садржалац периода задатака из скупа задатака који се распоређује. У овом примјеру вриједност хиперпериода је 60 ms.

id	Name	Task type	Abort on miss	Act. Date (ms)	Period (ms)	List of Act. dates (ms)	Deadline (ms)	WCET (ms)
1	TASK T1	Periodic	☑ Yes	0.0	12.0	-	12.0	6.0
2	TASK T2	Periodic	☑ Yes	0.0	15.0	-	15.0	3.0
3	TASK T3	Periodic	☑ Yes	0.0	20.0	-	20.0	7.0

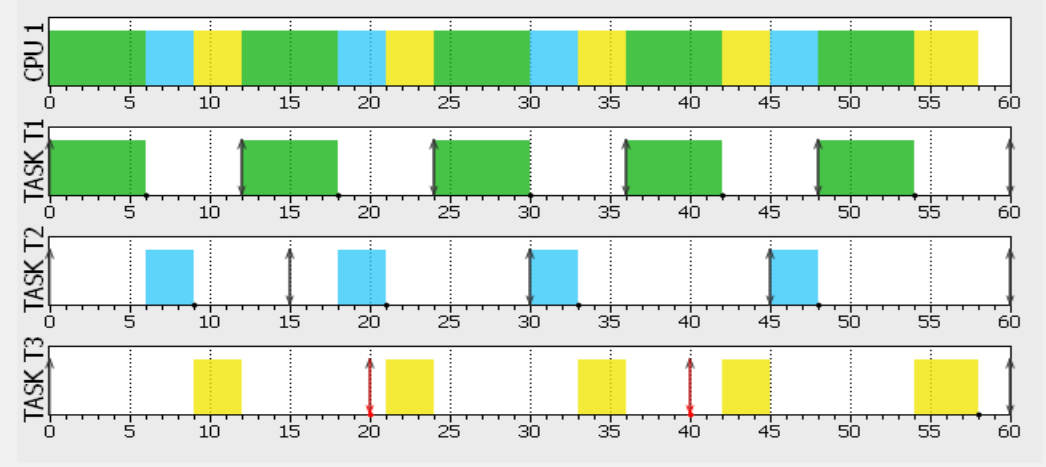
Слика 1. Скуп задатака који није распоредив ни РМ ни ЕДФ алгоритмом.

У случају ЕДФ алгоритма искоришћеност процесора у трајању хиперпериода је 100% или 1.0 , док за РМ алгоритам искоришћеност процесора је 97% или 0.97. Ако се искоришћеност процесора израчуна преко формуле добије се да је искоришћеност процесора 1.05, што нам показује да скуп задатака сигурно не може бити распоређен ни ЕДФ ни РМ алгоритмом.



Слика 2. Гантове карта за ЕДФ алгоритам.

Са слике 2, видимо да у тренутку 60, задатак 1 није комплетирао своје ивршавање, када се тај задатак поново активира, чиме је нарушен deadline, што значи да задатак 1 није распоредив и да скуп задатака није распоредив.



Слика 3. Гантове карте за PM алгоритам.

На основу Гантових карти, видимо да задаци 1 и 2, који имају краћи период имају и већи приоритет па задатак 3 не добија довољно времена како би комплетирао своје ивршавање унутар периода. Како је нарушен deadline задатка 3, овај скуп задатака није распоредив ни PM алгоритмом.

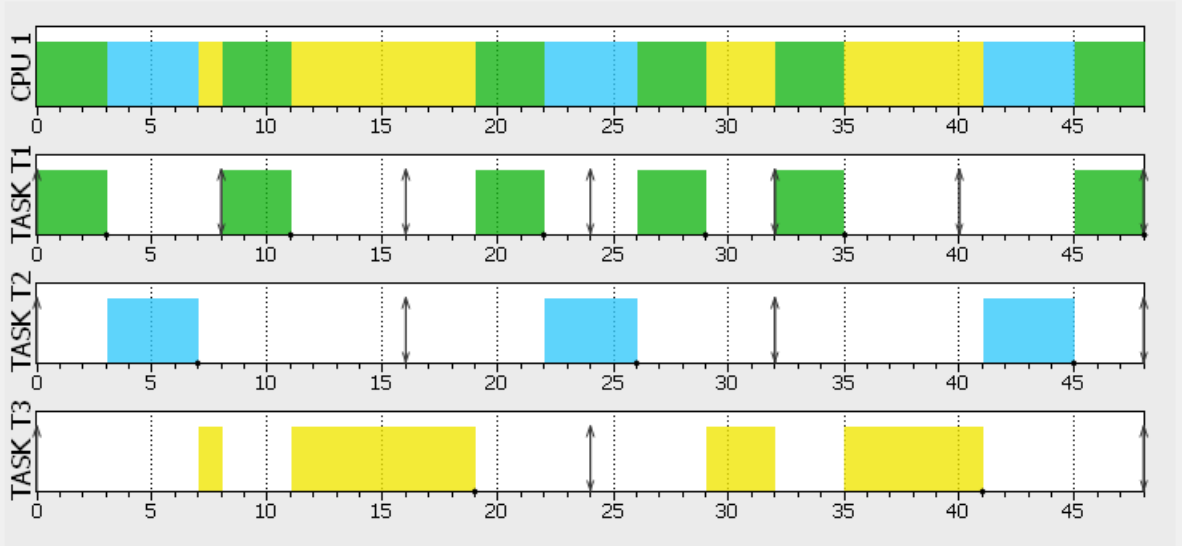
б)

У овом примјеру је било потребно одабрати скуп независних, периодичних задатака који ће бити распоредив ЕДФ алгоритмом али неће бити распоредив РМ алгоритмом. Неки скуп периодичних задатака ће бити распоредив ЕДФ алгоритмом уколико је искоришћеност процесора мања или једнака 1. За РМ алгоритам је услов распоредивости дефинисан тако да је искоришћеност процесора мања од неке доње границе искоришћености, Us = n($\sqrt[n]{2}$ – 1). Гдје број n, представља број задатака у скупу задатака који се распоређују. Међутим ово је потребан али не и довољан услов. У мом случају број задатака је 3, па је доња граница је 78% или 0.78.

id	Name	Task type	Abort on miss	Act. Date (ms)	Period (ms)	List of Act. dates (ms)	Deadline (ms)	WCET (ms)
1	TASK T1	Periodic	☑ Yes	0.0	8	-	8	3
2	TASK T2	Periodic	☑ Yes	0.0	16.0	-	16.0	4
3	TASK T3	Periodic	☑ Yes	0.0	24	-	24	9

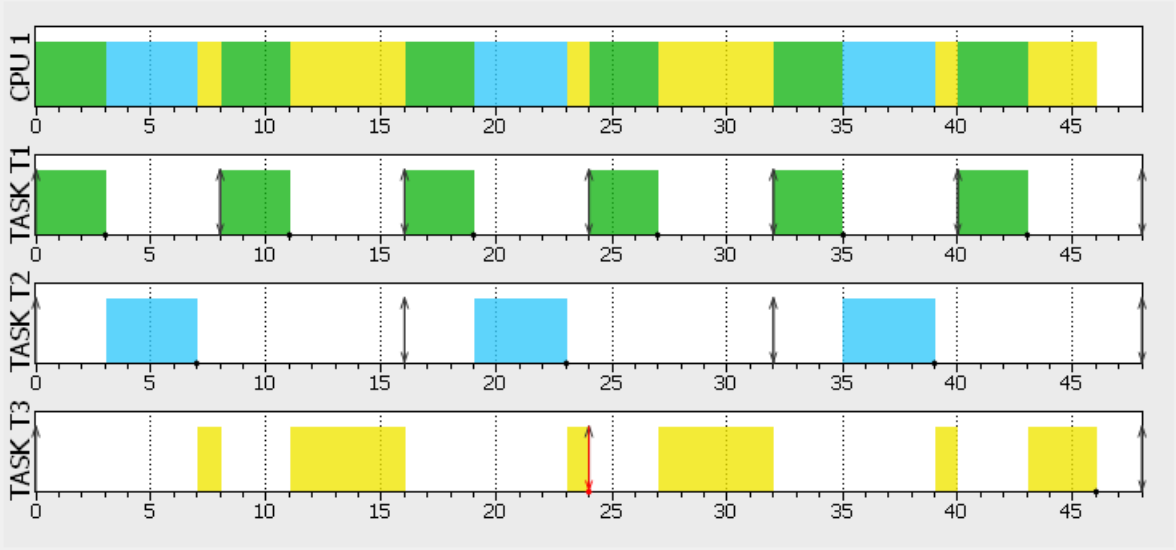
Слика 4. Скуп задатака распоредив само ЕДФ алгоритмом.

За овако дефинисан скуп задатака, искоришћеност процесора је 1 или 100%, а вриједност хиперпериода је 48 ms. Како је искоришћеност процесора за овако дефинисан скуп задатака 1, она је већа од доње границе распоредивости за РМ алгоритам па овај скуп задатака неће бити распоредив РМ алгоритмом, у тексту испод је описано како неки скуп задатака може бити распоредив РМ алгоритмом и ако је искоришћеност процесора већа од границе која је дефинисана формулом. Овако дефинисан скуп задатака је распоредив само помоћу ЕДФ алгоритма, што је приказано испод на сликама.



Слика 5. Гантове карте за ЕДФ алгоритам.

На основу Гантових карти са слике 5. видимо да је горе дефинисан скуп задатака распоредив помоћу ЕДФ алгоритма.



Слика 6. Гантове карте за РМ алгоритам.

На основу Гантових карти, видимо да задатак 1 који има најкраћи период добија процесорско вријеме чим се активира јер има највећи приоритет. У тренутку 24, долази нова инстанца задатка 3 али он није завршио своју активност па је нарушен његов deadline, па овако дефинисан скуп задатака није распоредив РМ алгоритмом. У овом случају искоришћеност процесора је 0.96 за РМ алгоритам док је 1.0 за ЕДФ алгоритам.

Неки скуп задатака може бити распоредив РМ алгоритмом иако је искоришћење процесора веће од доње границе која је дефинисана горе наведеном формулом. Наиме уколико се за сваки задатак израчуна најгоре вријеме одзива, и ако је најгоре вријеме одзива за сваки задатак мање од његовог периода онда је тај скуп задатака распоредив.

Најгоре вријеме одзива неког задатка зависи од његовог трајања, и од трајања и периода задатака који имају виши приоритет од специфікованог задатка.

Интерференција представља напредну метрику помоћу које се може детаљније испитати распоредивост задатка и самог скупа задатака. На основу формуле изведене на предавању знамо да се алгоритам извршава док не добијемо две идентичне вриједности у два корака алгоритма. Уколико се примјени алгоритам за овако дефинисан скуп задатака, задаци 1 и 2 су распоредиви јер је њихово најгоре вријеме извршавања мање од њиховог периода. Док задатак 3 није распоредив јер његово најгоре вријеме извршавања 29 ms, док је његов период 24 ms, што значи да задатак 3 није распоредив и да скуп задатака дефинисан на овај начин није распоредив РМ алгоритмом.

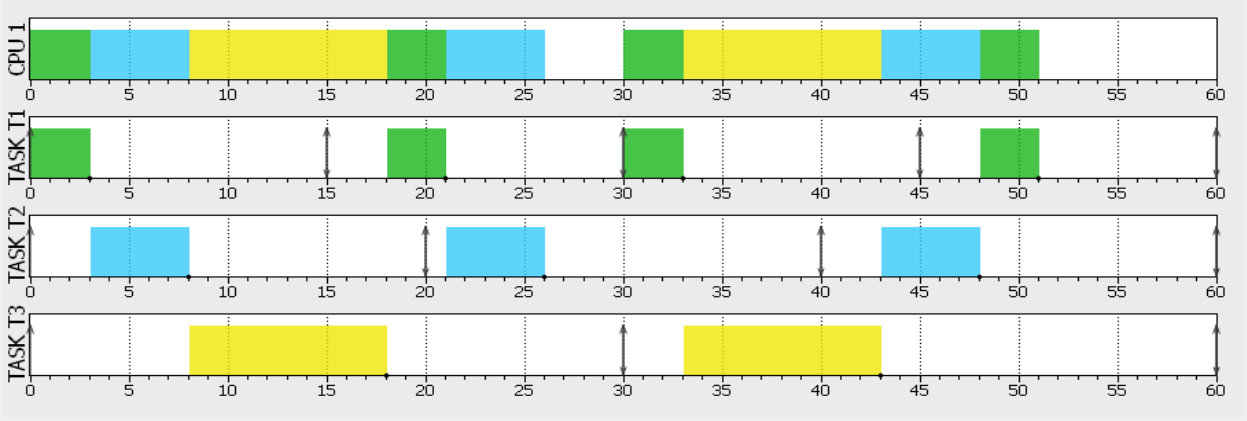
в)

У овом примјеру је било потребно одабрати скуп периодичних задатака који ће бити распоредив и ЕДФ и РМ алгоритмом.

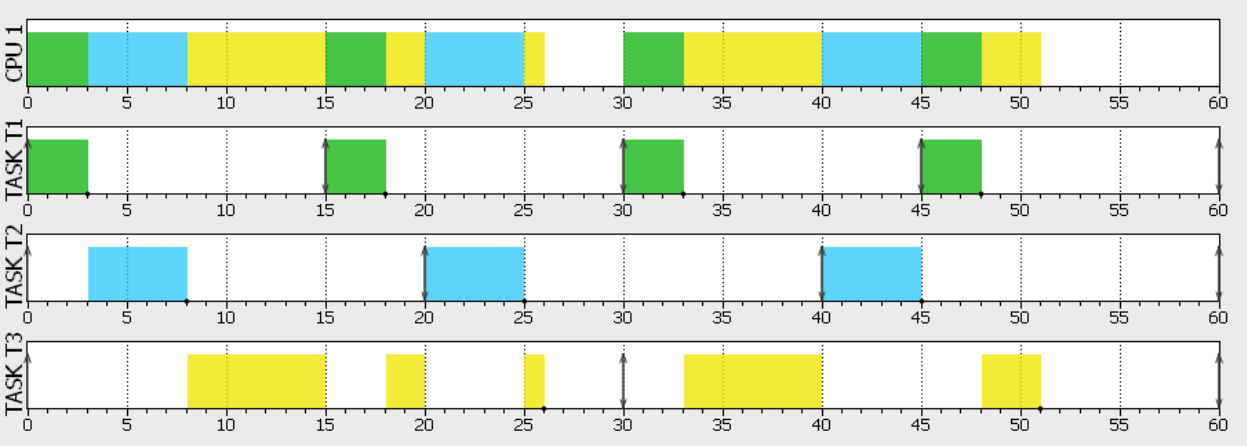
id	Name	Task type	Abort on miss	Act. Date (ms)	Period (ms)	List of Act. dates (ms)	Deadline (ms)	WCET (ms)
1	TASK T1	Periodic	☑ Yes	0	15	-	15	3
2	TASK T2	Periodic	☑ Yes	0	20	-	20	5
3	TASK T3	Periodic	☑ Yes	0	30	-	30	10

Слика 7. Скуп задатака који је распоредив са оба алгоритма.

Хиперпериод за овако дефинисан скуп задатака је 60 ms, док оптерећење процесора износи 0.78 или 78% за оба алгоритма. Испуњени су услови распоредивости за оба алгоритма што је приказано на сликама испод.



Слика 8. Гантове карте за ЕДФ алгоритам.



Слика 9. Гантове карте за PM алгоритам.

Са слика 8. и 9. видимо да је горе дефинисани скуп задатака распоредив са оба алгоритма.

Времена одзива (Response Time):

1. **ЕДФ алгоритам**

General | TASK T1 | TASK T2 | TASK T3

Activation	Start	End	Deadline	Comp. time	Resp. time	CPI	Preemptions	Migrations
0.0000	0.0000	3.0000	15.0000	3.0000	3.0000		0	0
15.0000	15.0000	21.0000	30.0000	3.0000	6.0000		1	0
30.0000	30.0000	33.0000	45.0000	3.0000	3.0000		0	0
45.0000	45.0000	51.0000	60.0000	3.0000	6.0000		0	0
60.0000	60.0000		75.0000				0	0

Слика 10. Вријеме одзива за задатак 1, коришћењем ЕДФ алгоритма.

General | TASK T1 | TASK T2 | TASK T3

Activation	Start	End	Deadline	Comp. time	Resp. time	CPI	Preemptions	Migrations
0.0000	0.0000	8.0000	20.0000	5.0000	8.0000		0	0
20.0000	20.0000	26.0000	40.0000	5.0000	6.0000		0	0
40.0000	40.0000	48.0000	60.0000	5.0000	8.0000		1	0
60.0000	60.0000		80.0000				0	0

Слика 11. Вријеме одзива за задатак 2, коришћењем ЕДФ алгоритма.

General | TASK T1 | TASK T2 | TASK T3

Activation	Start	End	Deadline	Comp. time	Resp. time	CPI	Preemptions	Migrations
0.0000	0.0000	18.0000	30.0000	10.0000	18.0000		1	0
30.0000	30.0000	43.0000	60.0000	10.0000	13.0000		1	0
60.0000	60.0000		90.0000				0	0

Слика 12. Вријеме одзива за задатак 3, коришћењем ЕДФ алгоритма.

2. PM алгоритам

General | TASK T1 | TASK T2 | TASK T3

Activation	Start	End	Deadline	Comp. time	Resp. time	CPI	Preemptions	Migrations
0.0000	0.0000	3.0000	15.0000	3.0000	3.0000		0	0
15.0000	15.0000	18.0000	30.0000	3.0000	3.0000		0	0
30.0000	30.0000	33.0000	45.0000	3.0000	3.0000		0	0
45.0000	45.0000	48.0000	60.0000	3.0000	3.0000		0	0
60.0000	60.0000		75.0000				0	0

Слика 13. Вријеме одзива за задатак 1, коришћењем PM алгоритма.

General | TASK T1 | TASK T2 | TASK T3

Activation	Start	End	Deadline	Comp. time	Resp. time	CPI	Preemptions	Migrations
0.0000	0.0000	8.0000	20.0000	5.0000	8.0000		0	0
20.0000	20.0000	25.0000	40.0000	5.0000	5.0000		0	0
40.0000	40.0000	45.0000	60.0000	5.0000	5.0000		0	0
60.0000	60.0000		80.0000				0	0

Слика 14. Вријеме одзива за задатак 2, коришћењем PM алгоритма.

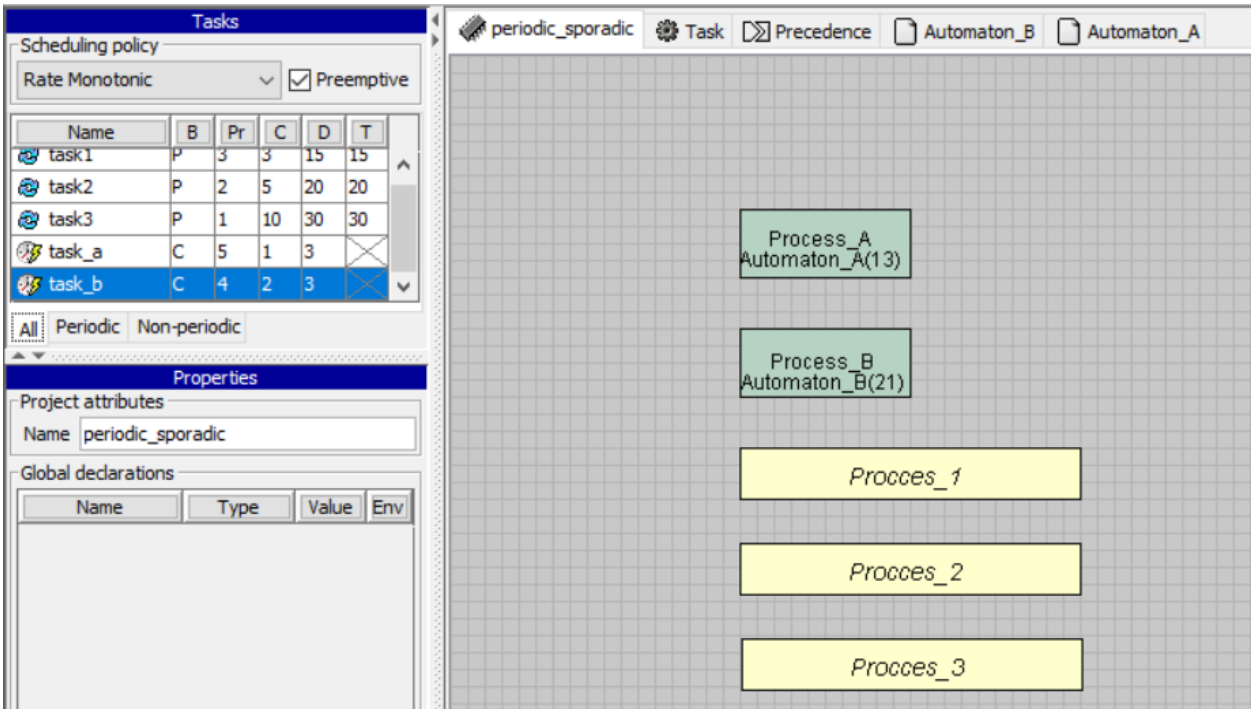
General | TASK T1 | TASK T2 | TASK T3

Activation	Start	End	Deadline	Comp. time	Resp. time	CPI	Preemptions	Migrations
0.0000	0.0000	26.0000	30.0000	10.0000	26.0000		2	0
30.0000	30.0000	51.0000	60.0000	10.0000	21.0000		1	0
60.0000	60.0000		90.0000				0	0

Слика 15. Вријеме одзива за задатак 3, коришћењем PM алгоритма.

Задатак 2.

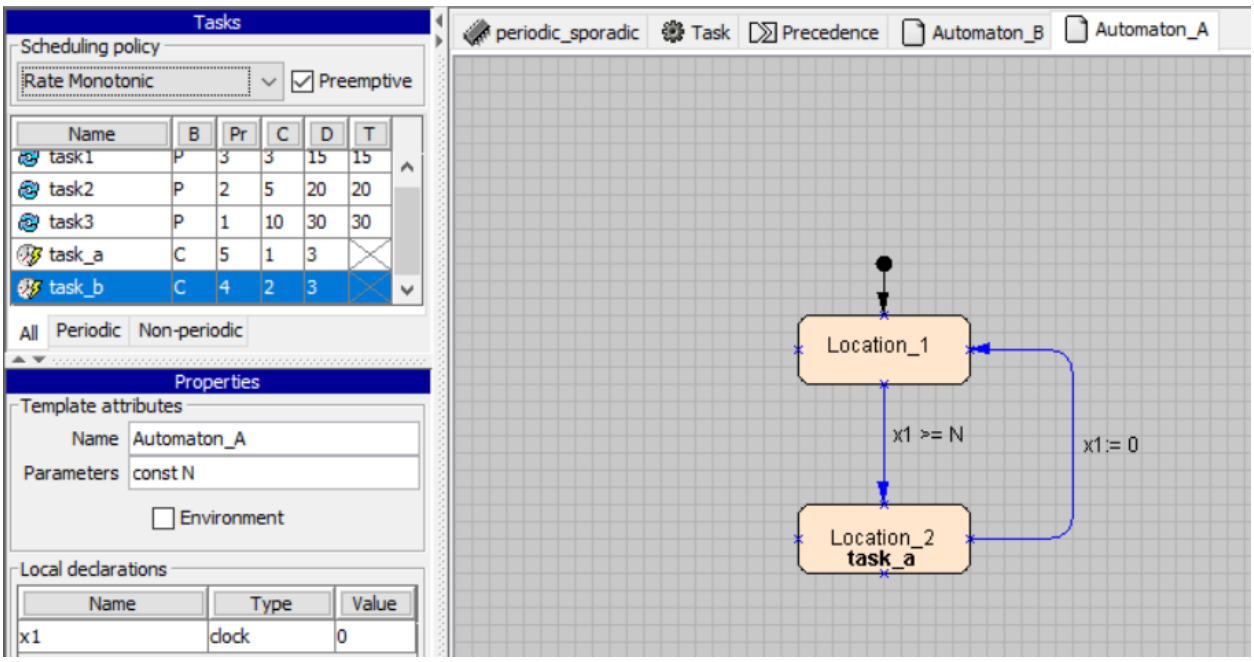
У овом задатаку смо користили алат TimesTool за симулацију распоређивања задатака. На скуп задатака дефинисан у претходном примјеру 1.в додали смо два апериодична задатка, названи task_a и task_b.



Слика 16. Приказ пројекта у алату TimesTool.

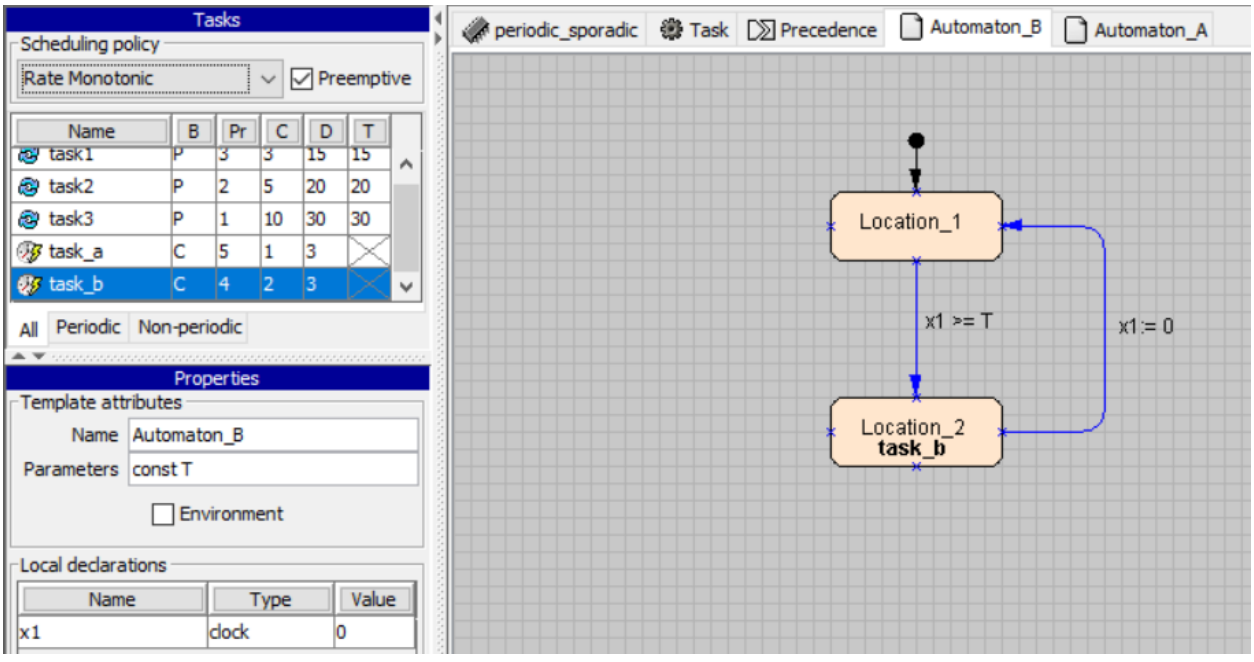
У оквиру пројекта дефинисали смо два процеса којима смо додјелили одговарајуће машине стања које служе да активирају апериодичне тј спорадичне задатке. Први процес којем одговара машина стања Аутоматон_А активира апериодични задатак А, док други процес активира апериодични задатак Б. Задатак А траје 1 ms и активира се у 13. ms, док задатак Б траје 2 ms а активира се у 21. ms. Задаци 1,2,3 представљају периодичне задатке дефинисане у оквиру задатка 1.в.

У овом случају нисмо дефинисали зависност између задатака и за распоређивање смо користили алгоритам PM. Вриједи напоменути да када се користи скуп који има и периодичне и апериодичне задатке, за распоређивање периодичних задатака се користи PM алгоритам распоређивања.

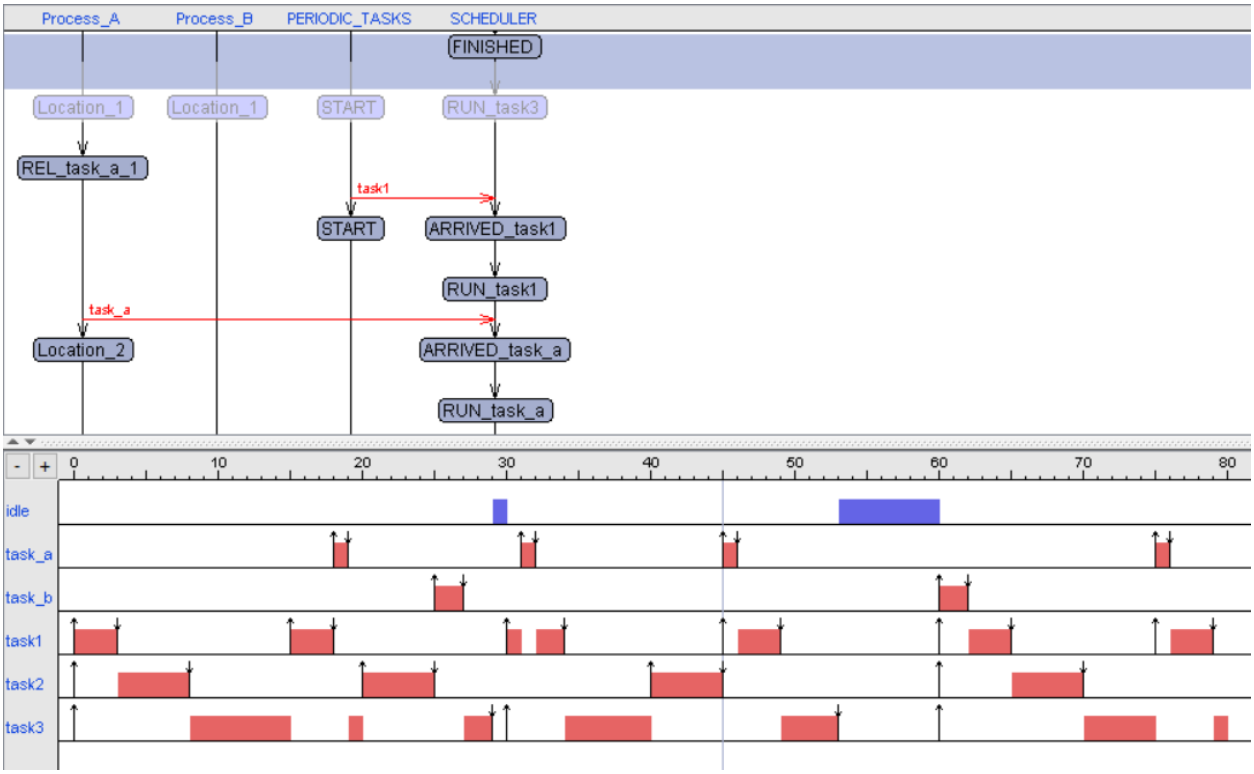


Слика 17. Машина стања која активира апериодични задатак А.

Када вриједност варијабле x1 буде једнака дефинисаној константи, активираће се спорадични задатак и када се аперидични задатак изврши машина стања се враћа у претходно стање.



Слика 18. Машина стања која активира апериодични задатак Б.

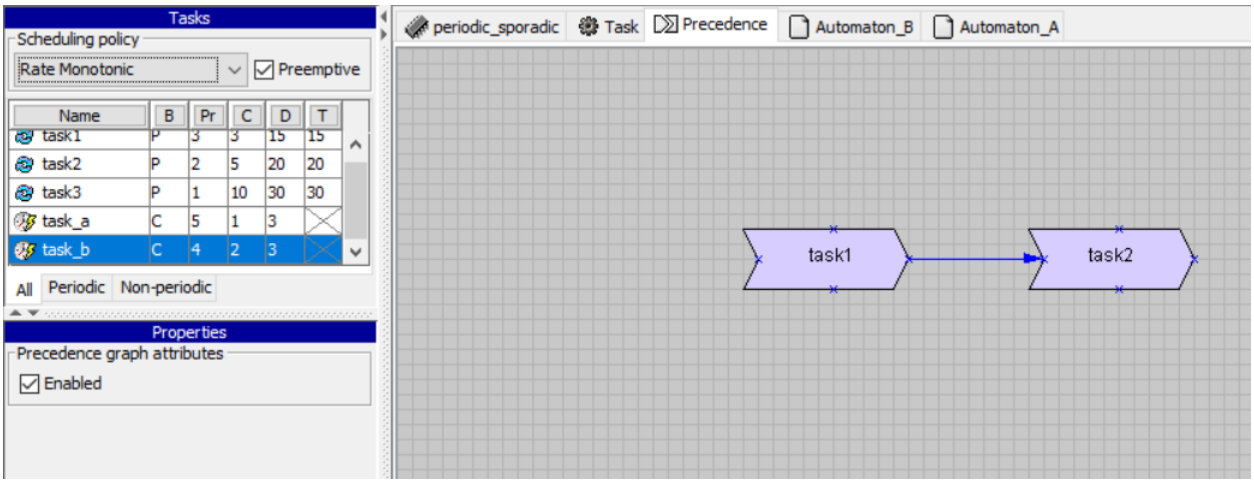


Слика 19. Гантове карте за скуп периодичних и апериодичних задатака.

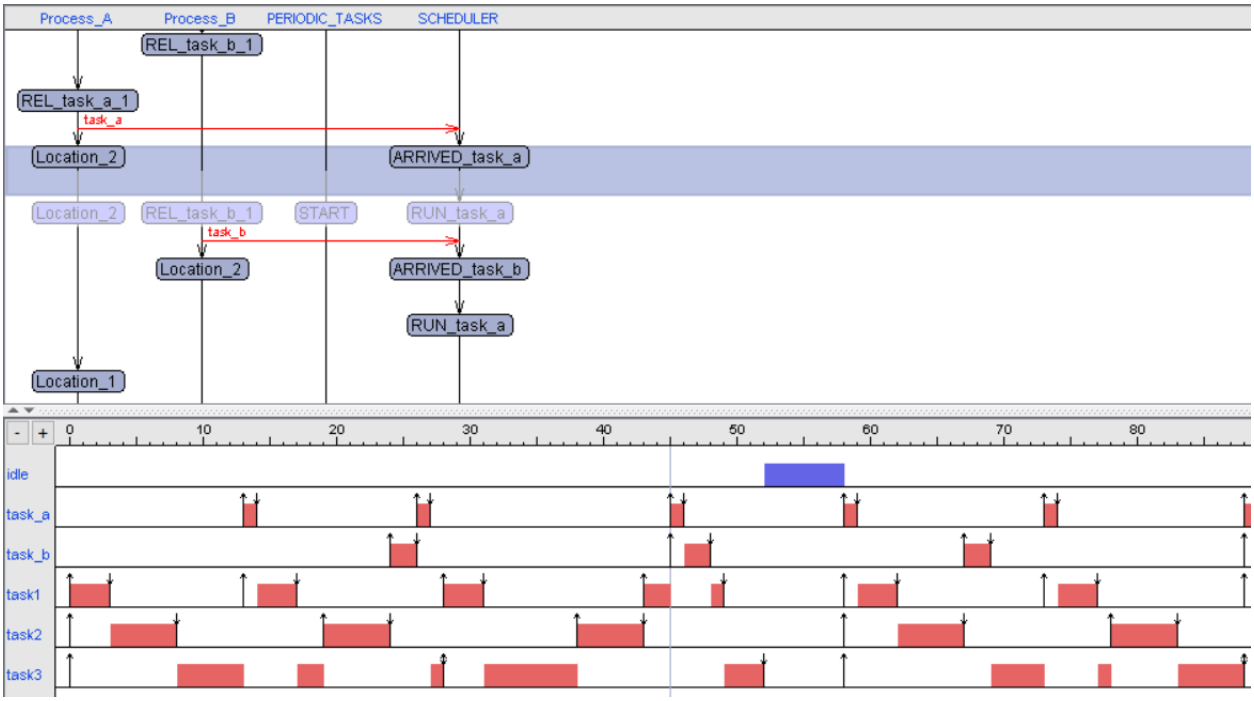
У односу на Гантове карте из примјера 1.в видимо да у овом случају имамо веће искоришћење процесора због апериодичних задатака. Видимо да када апериодични задатак пристигне да он прекида извршавање ниже приоритетних периодичних задатака.

Апериодични задаци имају већи приоритет од периодичних задатака, и када се користи овај алат, он аутоматски задацима додјељује приоритете. Правило за приоритет је да што је већи број у пољу за приоритет задатак има већи приоритет.

У оквиру овог алата се може спецификовати зависност између задатака, тј да када се заврши један задатак тек онда се други задатак који зависи од тога задатка може активирају тј започети своје извршавање. Треба водити рачуна да уколико се спецификује да неки задатак вишег приоритета зависи од завршетка неког задатка нижег приоритета скуп задатака неће бити распоредив, јер то представља супротност РМ алгоритму.



Слика 20. Спецификација зависности задатка 2 од задатка 1.



Слика 21. Гантове карте за случај када имамо зависност између задатака 1 и 2.