

Programiranje u realnom vremenu

Laboratorijska vježba 3
Domaći zadatak

Luka Knežić, 11128/18

Zadatak 1

Zadatak 1 zahtijeva da se odaberu skupovi nezavisnih periodičnih zadataka koji mogu biti rasporedivi ili nerasporedivi određenim algoritmom, te da se analizira njihova rasporedivost *SimSo* alatom.

Dio datog zadatka pod a) traži da se odabere skup nezavisnih periodičnih zadataka koji je nerasporediv RM i EDF algoritmom. U Tabeli 1.1 prikazan je proizvoljno odabran skup koji ispunjava tražene zahtjeve. Odabran je skup od četiri zadatka čiji periodi i vremena izvršavanja (T_i i C_i , respektivno) zadovoljavaju postavljene uslove. Kada se sumiraju pojedinačna iskorišćenja zadataka, uočljivo je da ukupan faktor iskorišćenja procesora (koristi se jedan procesor) prelazi 1, odnosno iznosi 1.15 ($U = 1.15$), što je jasan pokazatelj da ovako definisan skup zadataka sigurno nije rasporediv ni sa jednim algoritmom. Dobijena vrijednost faktora iskorišćenja procesora praktično govori da bi procesor trebao uložiti 115% vremena od trenutka aktivacije svih zadataka (svi zadaci se aktiviraju u nultom trenutku) da rasporedi zadatke što je fizički nemoguće, pa se i iz ovog zaključka može jasno vidjeti da priloženi skup nije rasporediv.

Zadatak	Period (T_i)	Vrijeme izvršavanja (C_i)	Prioritet	Iskorišćenje ($\frac{C_i}{T_i}$)
T1	50	10	4	0.2
T2	40	12	3	0.3
T3	16	4	2	0.25
T4	10	4	1	0.4

Tabela 1.1 Skup zadataka za Zadatak 1 pod a)

Na slikama 1.1, 1.2 i 1.3 prikazani su skup zadataka iz Tabele 1.1, faktor iskorišćenja procesora nakon simulacije i Gantov dijagram za dati skup, respektivno, korišćenjem *SimSo* alata. Potrebno je naglasiti da je simulacija prikazana na hiperperiodu skupa da bi faktor U bio identičan (ili približan) onom iz proračuna. Vrijednost hiperperioda iznosi 400, jer je 400 najmanji zajednički sadržalac za sve periode skupa. Ukoliko bi ovaj skup bio rasporediv, nakon vremena hiperperioda raspoređivanje zadataka bi se ponavljalo. Na Slici 1.1 se vidi da je parametar D_i (relativni *deadline*) postavljen na vrijednost perioda određenog zadatka, što odgovara teoriji. Jednostavno, da bi zadatak bio rasporediv mora da se jedna njegova instanca izvrši u okviru njegovog perioda. Slika 1.2 govori da je procesor zauzet u svakom trenutku vremena od početka aktivacije svih zadataka datog skupa. Proračunata vrijednost faktora U je 1.15, a na Slici 1.2 iznosi 1. Razlog je taj što je prosto nemoguće da procesor bude aktivan u 115% vremena od trenutka aktivacije. Slike 1.3 i 1.4 daju najbolji uvid u nerasporedivost skupa. Slika 1.3 prikazuje Gantov dijagram skupa za EDF algoritam. Uočljivo je da se peta instanca zadatka $T4$ neće izvršiti u okviru njegovog perioda. Naime, kako EDF algoritam dinamički dodjeljuje prioritete na osnovu apsolutnog roka zadataka (apsolutni rok i -te instance određenog zadatka iznosi iT , gdje je T period zadatka), u trenutku 40 prednost dobija zadatak $T1$ koji u tom trenutku imati apsolutni rok jednak 50 kao i zadatak $T4$ u istom trenutku. *SimSo* alat prednost daje zadatku $T1$ jer on nije izvršio još nijednu instancu i on se praktično izvršava u intervalu jednog perioda zadatka $T4$ i na taj način zadatak $T4$ neće imati svoju petu instancu u okviru definisanog perioda. Samim tim, dati skup nije rasporediv EDF algoritmom. Sa druge strane, vidi se sa Slike 1.4 da dati skup nije rasporediv ni sa RM algoritmom. Jasno se vidi da se zadatak $T1$ neće izvršiti u okviru njegovog perioda. Kako RM algoritam prednost daje zadatku višeg prioriteta, zadatak $T1$ prosto neće doći na red za izvršavanje u svom periodu, jer ima najniži

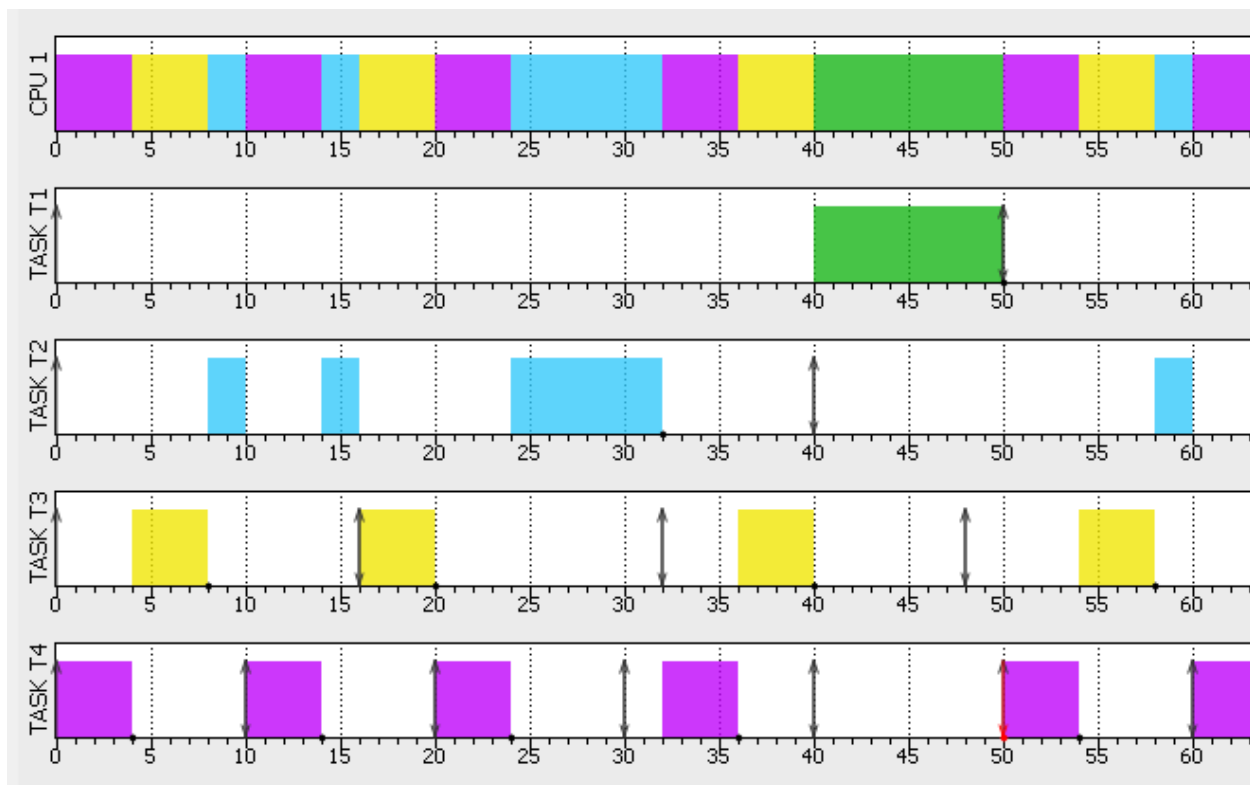
prioritet, a ostali zadaci moraju ispuniti uslov izvršenja u okviru svog perioda. Zbog definisanog perioda ostalih zadataka i njihovog vremena izvršavanja jasno je da se prva instanca zadatka *T1* ne može izvršiti u definisanom periodu za dati zadatak. Ovaj dio se nalazi u *xml* fajlu *zadatak1-a*.

id	Name	Task type	Abort on miss	Act. Date (ms)	Period (ms)	List of Act. dates (ms)	Deadline (ms)	WCET (ms)	Followed by
1	TASK T1	Periodic	<input checked="" type="checkbox"/> Yes	0.0	50.0	-	50.0	10.0	
2	TASK T2	Periodic	<input checked="" type="checkbox"/> Yes	0.0	40.0	-	40.0	12.0	
3	TASK T3	Periodic	<input checked="" type="checkbox"/> Yes	0.0	16.0	-	16.0	4.0	
4	TASK T4	Periodic	<input checked="" type="checkbox"/> Yes	0.0	10.0	-	10.0	4.0	

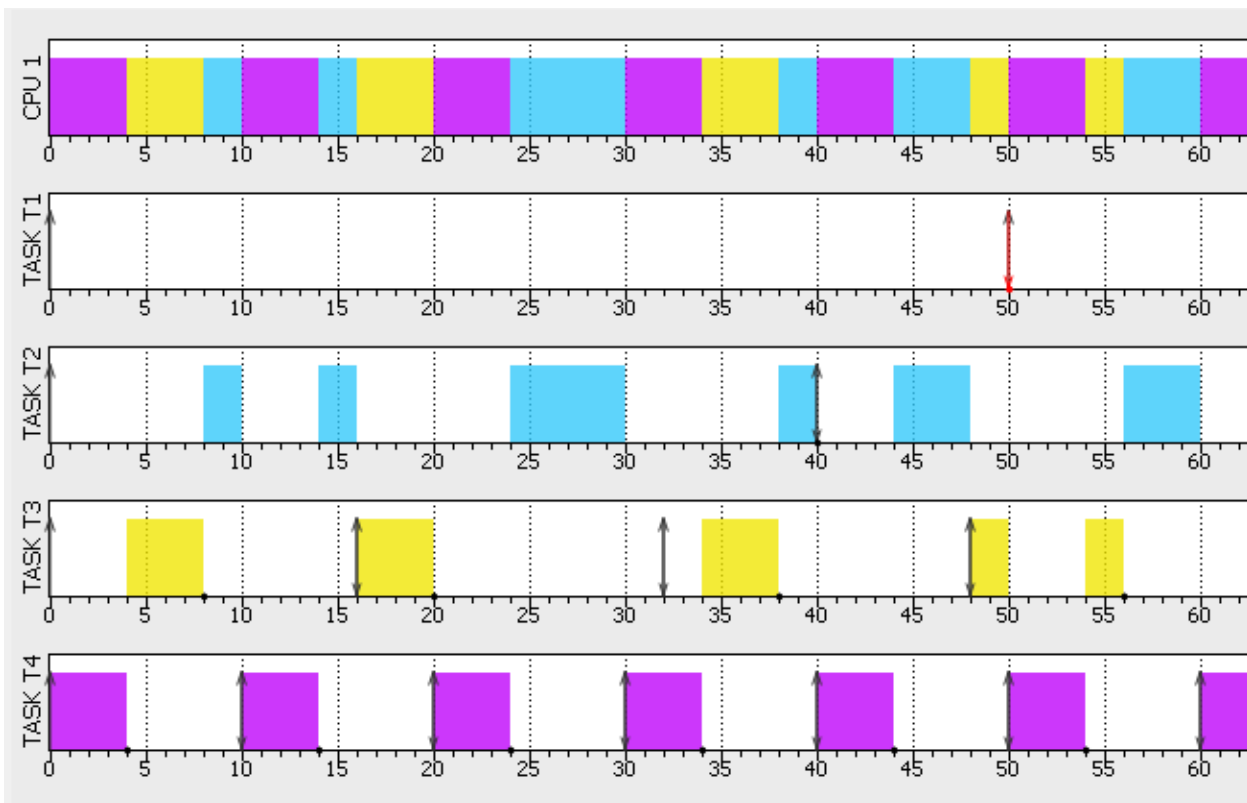
Slika 1.1 Skup zadataka za Zadatak 1 pod a) prikazan u SimSo alatu

	Total load	Payload	System load
CPU 1	1.0000	1.0000	0.0000
Average	1.0000	1.0000	0.0000

Slika 1.2 Prikaz iskorišćenja procesora za Zadatak 1 pod a) u SimSo alatu (EDF i RM algoritam)



Slika 1.3 Gantov dijagram za skup zadataka iz Zadatka 1 pod a) u SimSo alatu (EDF algoritam)



Slika 1.4 Gantov dijagram za skup zadataka iz Zadataka 1 pod a) u SimSo alatu (RM algoritam)

Zadatak 1 pod b) traži skup nezavisnih periodičnih zadataka koji je rasporediv EDF algoritmom i nerasporediv RM algoritmom. U skladu s tim proizvoljno je odabran skup koji ispunjava ove zahtjeve i prikazan je u Tabeli 1.2. Skup posjeduje četiri zadatka i periodi i vremena izvršavanja zadataka su izabrani da ispunjavaju uslove zadatka. Faktor U prema proračunu (sumirajući pojedinačna opterećenja) iznosi 0.98. Na osnovu ovog podatka se zna da je ovakav skup rasporediv EDF algoritmom, jer je za ovaj algoritam potreban i dovoljan uslov rasporedivosti $U \leq 1$. Za analizu rasporedivosti RM algoritmom potrebno je najprije izračunati faktor U_s prema formuli $U_s = n(\sqrt[n]{2} - 1)$, gdje je n broj zadataka određenog skupa. U konkretnom slučaju je $n = 4$, pa je $U_s = 0.76$. Uočljivo je da je $U > U_s$ i po teoriji ovaj skup ne prolazi test rasporedivosti. Međutim, on i dalje može biti rasporediv, jer prethodno navedeni uslov nije potreban i dovoljan da bi se moglo reći da je skup zadataka sigurno nerasporediv. U konkretnom slučaju skup zadataka prikazan u Tabeli 1.2 nije rasporediv RM algoritmom, što potvrđuje i simulacija u *SimSo* alatu.

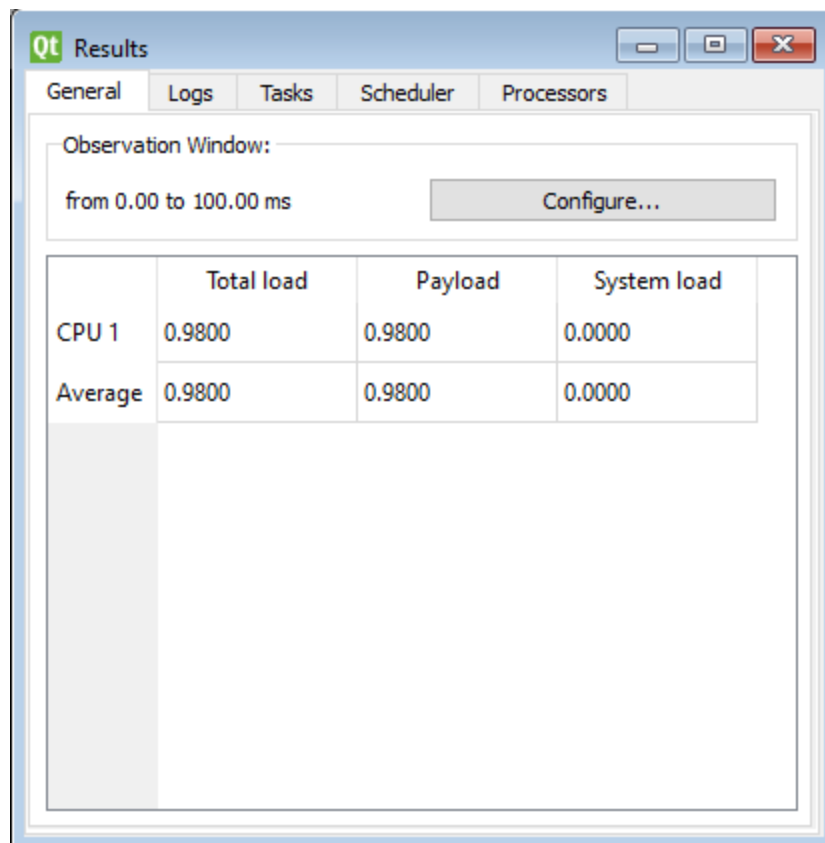
Zadatak	Period (T_i)	Vrijeme izvršavanja (C_i)	Prioritet	Iskorištenje ($\frac{C_i}{T_i}$)
T1	50	20	4	0.4
T2	25	2	3	0.08
T3	20	4	2	0.2
T4	10	3	1	0.3

Tabela 1.2 Skup zadataka za Zadatak 1 pod b)

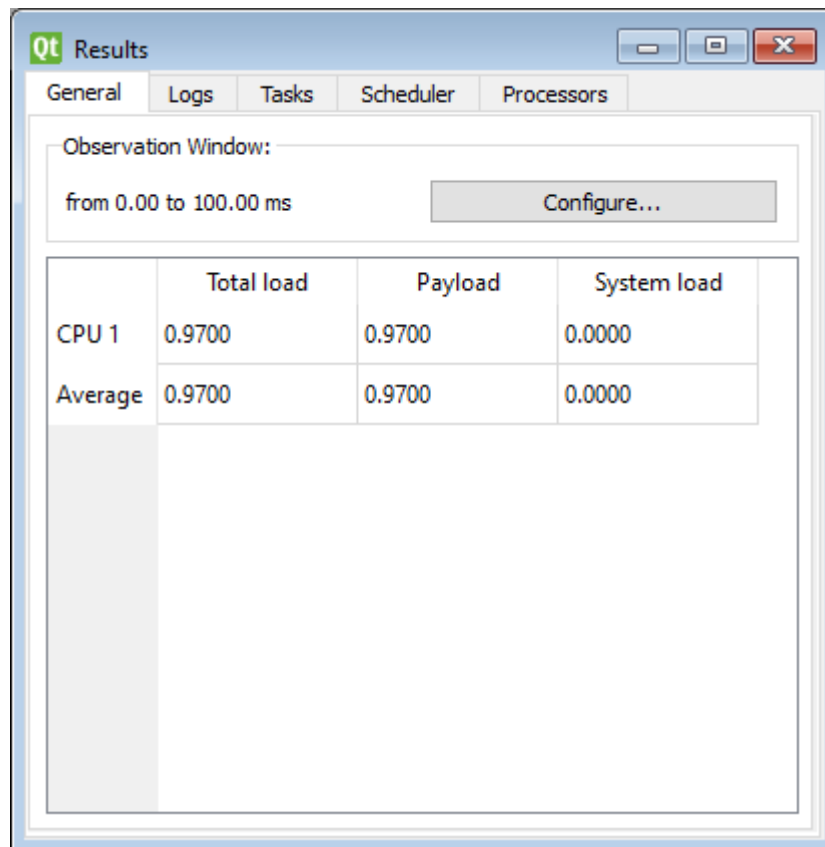
Na Slici 1.5 prikazan je skup iz Tabele 1.2 u *SimSo* alatu. Slike 1.6 i 1.7 prikazuju faktor iskorišćenja procesora za EDF i RM algoritam, respektivno. *SimSo* alat u slučaju RM algoritma pokazuje da je vrijednost faktora iskorišćenja procesora jednaka 0.97, što u ovom slučaju ne igra veliku ulogu. Hiperperiod datog skupa iznosi 100 i poslije njega se raspoređivanje ponavlja za EDF algoritam. Simulacija je prikazana na navedenom hiperperiodu za oba algoritma. Slika 1.8 prikazuje Gantov dijagram za EDF algoritam skupa iz dijela zadatka pod b). Pošto je uslov rasporedivosti EDF algoritmom ispunjen dati skup zadataka je rasporediv što i simulacija potvrđuje. Na Slici 1.8 se vidi da je skup rasporediv na hiperperiodu, samim tim i na bilo kojem vremenskom intervalu u skladu sa prethodno izloženim načinom raspoređivanja EDF algoritmom za primjer pod a). Na Slici 1.9 prikazan je Gantov dijagram za RM algoritam iz dijela zadatka pod b). Sa priložene slike se vidi da se zadatak *T1* neće izvršiti u okviru svog definisanog perioda. S obzirom da on ima najniži prioritet, tj. posljednji dolazi na red za izvršavanje vidljivo je da on uspio da izvrši 19 vremenskih jedinica u okviru perioda, a vrijeme izvršavanja mu je 20 vremenskih jedinica. Samim tim, ovaj skup nije rasporediv RM algoritmom. Dio Zadatka 1 pod b) nalazi se u *xml* fajlu *zadatak1-b*.

id	Name	Task type	Abort on miss	Act. Date (ms)	Period (ms)	List of Act. dates (ms)	Deadline (ms)	WCET (ms)	Followed by
1	TASK T1	Periodic	<input checked="" type="checkbox"/> Yes	0.0	50.0	-	50.0	20.0	
2	TASK T2	Periodic	<input checked="" type="checkbox"/> Yes	0.0	25.0	-	25.0	2.0	
3	TASK T3	Periodic	<input checked="" type="checkbox"/> Yes	0.0	20.0	-	20.0	4.0	
4	TASK T4	Periodic	<input checked="" type="checkbox"/> Yes	0.0	10.0	-	10.0	3.0	

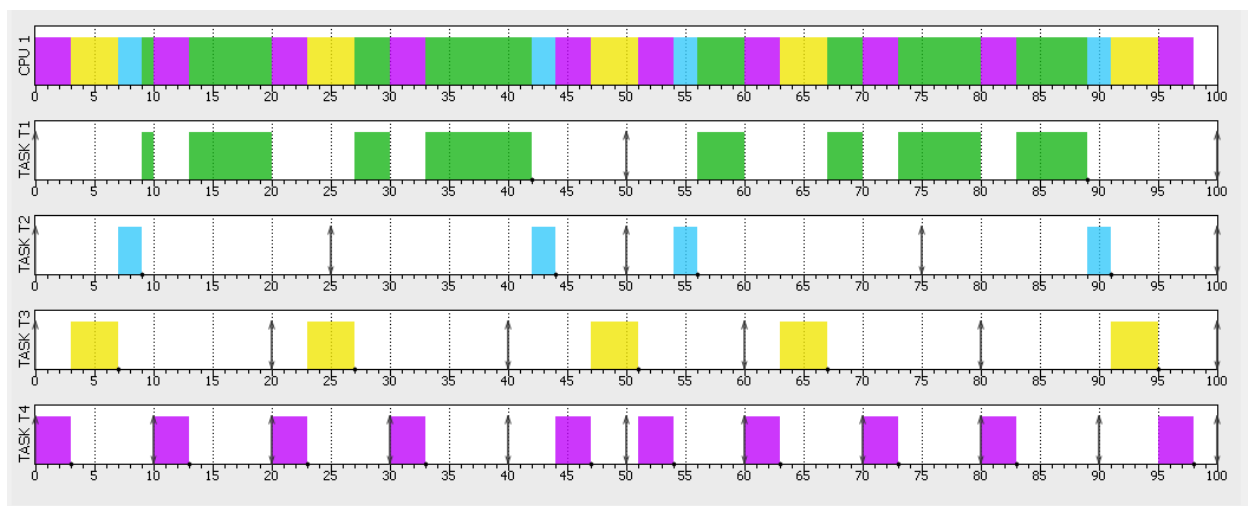
Slika 1.5 Skup zadataka iz Zadatka 1 pod b) prikazan u *SimSo* alatu



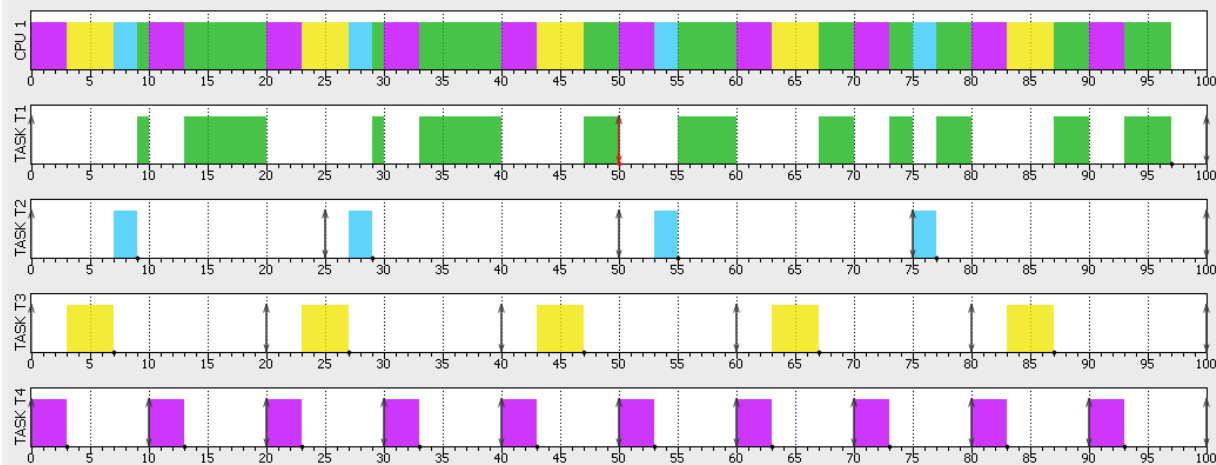
Slika 1.6 Prikaz iskorišćenja procesora za Zadatak 1 pod b) u SimSo alatu (EDF algoritam)



Slika 1.7 Prikaz iskorišćenja procesora za Zadatak 1 pod b) u SimSo alatu (RM algoritam)



Slika 1.8 Gantov dijagram za skup zadataka iz Zadatak 1 pod b) u SimSo alatu (EDF algoritam)



Slika 1.9 Gantov dijagram za skup zadataka iz Zadanka 1 pod b) u SimSo alatu (RM algoritam)

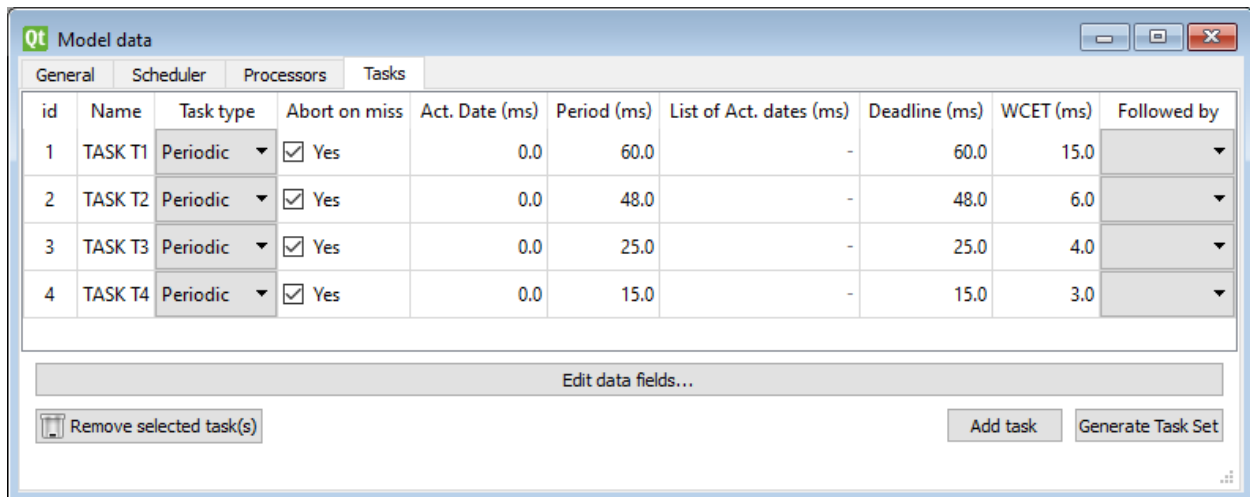
Dio zadatka 1 pod c) zahtijeva jedan skup zadataka koji u sebi sadrži nezavisne periodične zadatke koji su rasporedivi i EDF i RM algoritmom. Proizvoljno je odabran skup od četiri zadatka čiji parametri ispunjavaju tražene zahtjeve i prikazan je u Tabeli 1.3. Faktor U iznosi 0.735, a faktor U_s iznosi 0.76 kao u prethodnom slučaju jer je isti broj zadataka u skupu. Pošto je $U \leq 1$ skup je rasporediv EDF algoritmom. Dodatno, kako je $U \leq U_s$, skup je rasporediv i RM algoritmom jer je prethodna nejednakost dovoljan uslov da određeni skup zadataka postane rasporediv. Dakle, skup iz Tabele 1.3 je rasporediv sa oba algoritma.

Zadatak	Period (T_i)	Vrijeme izvršavanja (C_i)	Prioritet	Iskorištenje ($\frac{C_i}{T_i}$)
T1	60	15	4	0.25
T2	48	6	3	0.125
T3	25	4	2	0.16
T4	15	3	1	0.2

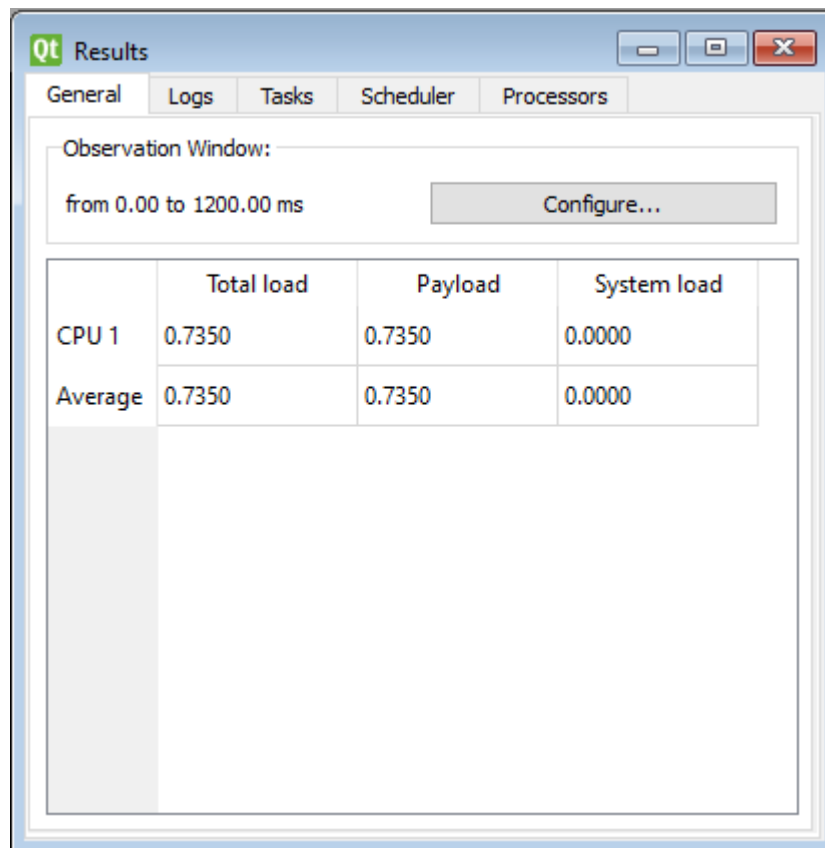
Tabela 1.3 Skup zadataka za Zadatak 1 pod c)

Slika 1.10 predstavlja skup zadataka iz Tabele 1.3 prikazan u SimSo alatu. Sa Slike 1.11 se vidi da faktor U odgovara proračunu. Hiperperiod za ovaj skup iznosi 1200 i simulacija je odrađena na ovom periodu jer se raspoređivanje zadataka nakon njega ponavlja kod oba algoritma. Na Slici 1.12 prikazan je Gantov dijagram za EDF algoritam. Iako nije prikazan na kompletnom hiperperiodu jasno je da je skup iz Tabele 1.3 rasporediv EDF algoritmom jer je ispunjen uslov rasporedivosti, što i simulacija potvrđuje. Na Slici 1.13 prikazan je Gantov dijagram za RM algoritam. Kako je i uslov za rasporedivost RM algoritmom ispunjen, dati skup je rasporediv ovim algoritmom što i simulacija potvrđuje. Vrijeme odziva (najgore vrijeme odziva) svih zadataka iz datog skupa za EDF i RM algoritam prikazani su u Tabeli 1.4. Vremena odziva se mogu vidjeti u okviru prozora *Results* u SimSo alatu izborom opcije *Tasks*, zatim izborom zadatka od interesa. Dio *Resp. time* pokazuje vrijeme odziva svake instance posmatranog zadatka. Na osnovu ovoga pregleda uzima se najveće vrijeme kao vrijeme odziva zadatka. Ovo je vidljivo i sa Gantovih dijagrama. Naravno, ovo je moguće izračunati i primjenom teorijskog koncepta za izračunavanje vremena odziva. Primjetno je da je vrijeme odziva za zadatak T2 bolje u slučaju RM algoritma, dok su ostala vremena odziva jednaka u

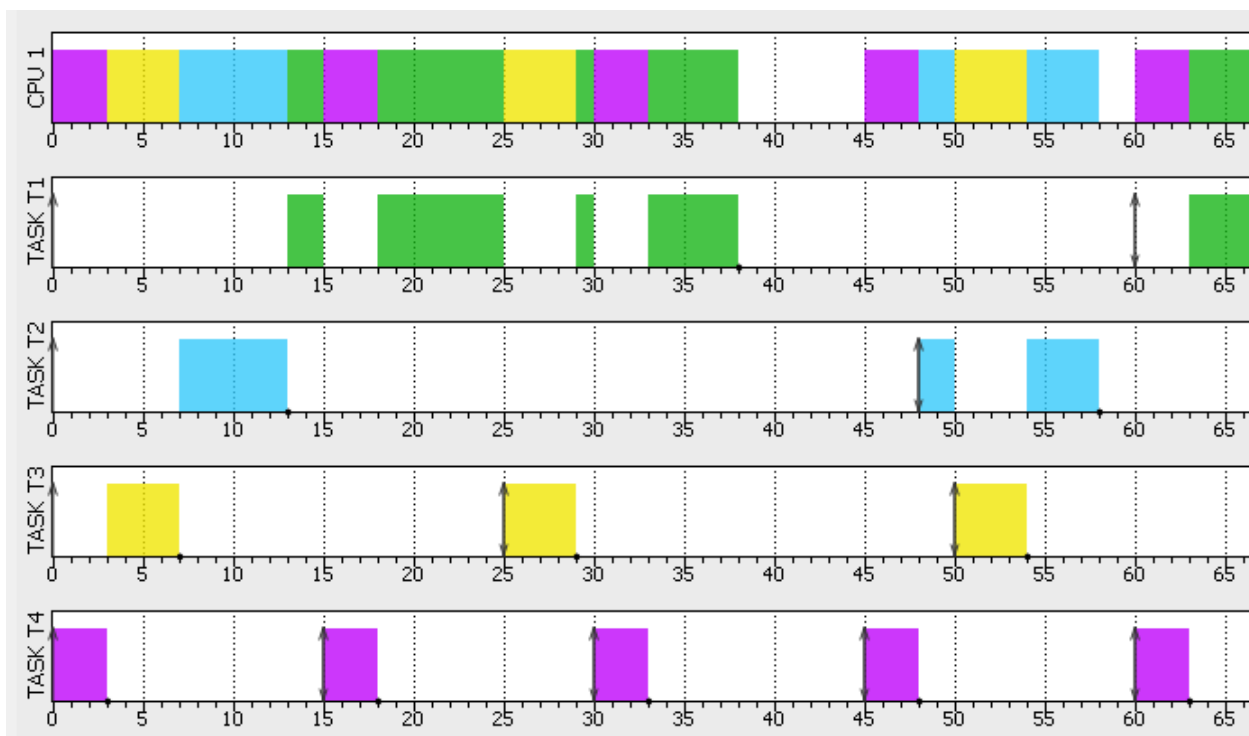
oba algoritma. Dakle, za nijansu je optimalniji RM algoritam za ovaj slučaj u pogledu vremena odziva. Dio Zadataka 1 pod c) nalazi se u *xml* fajlu *zadatak1-c*.



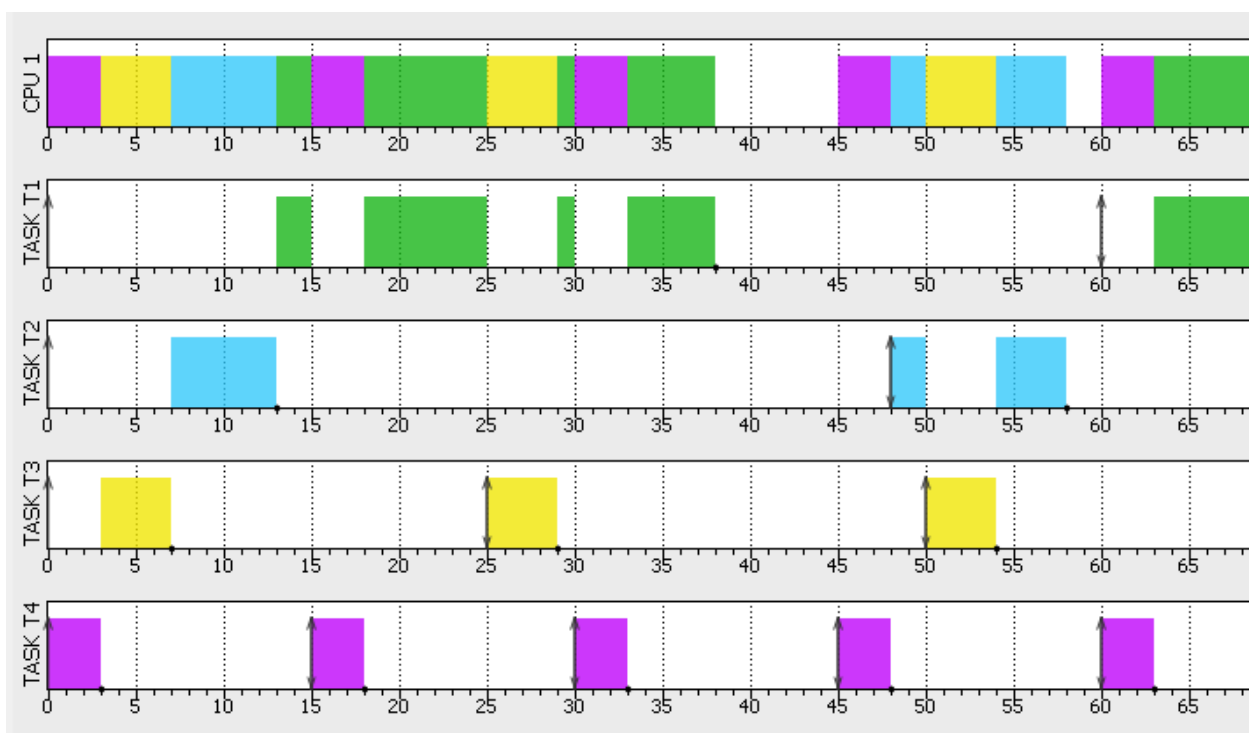
Slika 1.10 Skup zadataka za Zadatak 1 pod c) prikazan u SimSo alatu



Slika 1.11 Prikaz iskorišćenja procesora za Zadatak 1 pod c) u SimSo alatu (EDF i RM algoritam)



Slika 1.12 Gantov dijagram za skup zadataka iz Zadatka 1 pod c) u SimSo alatu (EDF algoritam)



Slika 1.13 Gantov dijagram za skup zadataka iz Zadatka 1 pod c) u SimSo alatu (RM algoritam)

Zadatak	Period (T_i)	Vrijeme izvršavanja (C_i)	Vrijeme odziva (EDF algoritam)	Vrijeme odziva (RM algoritam)
T1	60	15	38	38
T2	48	6	26	13
T3	25	4	7	7
T4	15	3	3	3

Tabela 1.4 Vremena odziva zadataka za Zadatak 1 pod c) (EDF i RM algoritam)

Zadatak 2

Zadatak 2 ima zahtjev da se proširi skup iz zadatka 1 pod c) dodavanjem dva aperiodična zadatka koja se aktiviraju u proizvoljno odabranim vremenskim trenucima i koji imaju proizvoljno odabrano vrijeme izvršavanja. Zatim, potrebno je analizirati slučajeve sa i bez zavisnosti barem dva periodična zadatka iz novoformiranog skupa korišćenjem *Timestool* alata.

Novonastali skup zadataka prikazan je u Tabeli 2.1. On prolazi test rasporedivosti pri korišćenju RM algoritma za raspoređivanje periodičnih zadataka. Naime, priloženi skup svakako prolazi test rasporedivosti ako se uzmu njegovi periodični zadaci, tj. skup iz dijela zadatka 1 pod c) RM algoritmom. Faktor U ima vrijednost 0.735 što znači da ostaje dovoljno procesorskog vremena za izvršavanje dva aperiodična zadatka. Za aperiodične zadatke ne važi RM algoritam, oni se mogu rasporediti tzv. pozadinskim raspoređivanjem (*background scheduling*), *Polling Server* ili *Deferred Server* algoritmom. Pošto u datom skupu nema servera (periodičnog zadatka koji uslužuje aperiodične zadatke), *Polling Server* i *Deferred Server* se ne mogu koristiti. Sa Slike 2.5 se vidi da je iskorištena neka vrsta pozadinskog raspoređivanja, jer aperiodični zadaci koriste vrijeme kada nisu aktivni periodični zadaci.

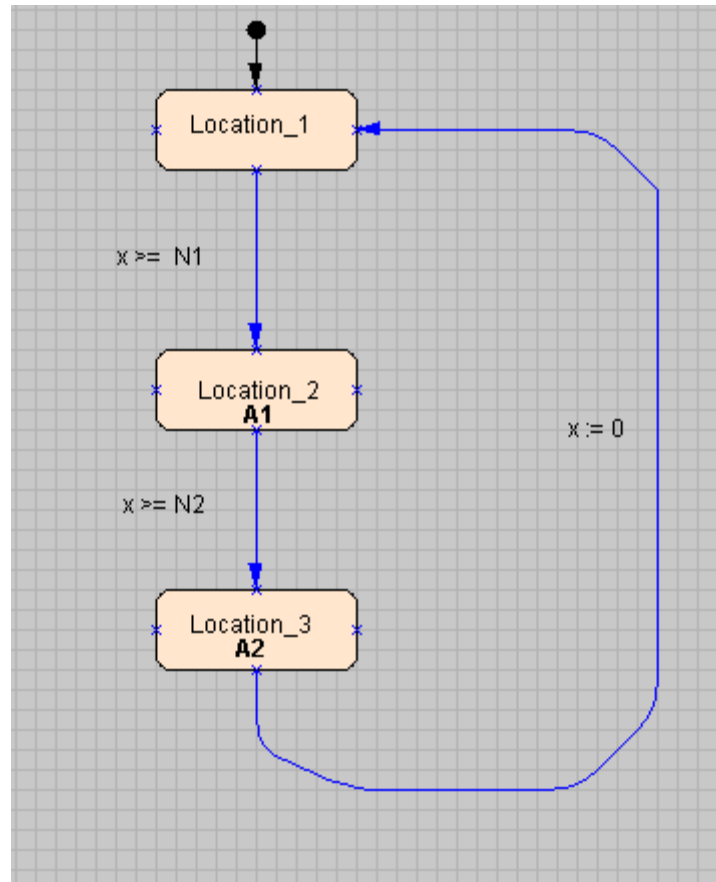
Zadatak	Tip zadatka	Period (T_i)	Vrijeme izvršavanja (C_i)	Relativni <i>deadline</i> (D)	Prioritet	Iskorištenje ($\frac{C_i}{T_i}$)
A1	aperiodični	-	1	10	5	-
A2	aperiodični	-	3	12	6	-
T1	periodični	60	15	60	4	0.25
T2	periodični	48	6	48	3	0.125
T3	periodični	25	4	25	2	0.16
T4	periodični	15	3	15	1	0.2

Tabela 2.1 Skup zadataka za Zadatak 2

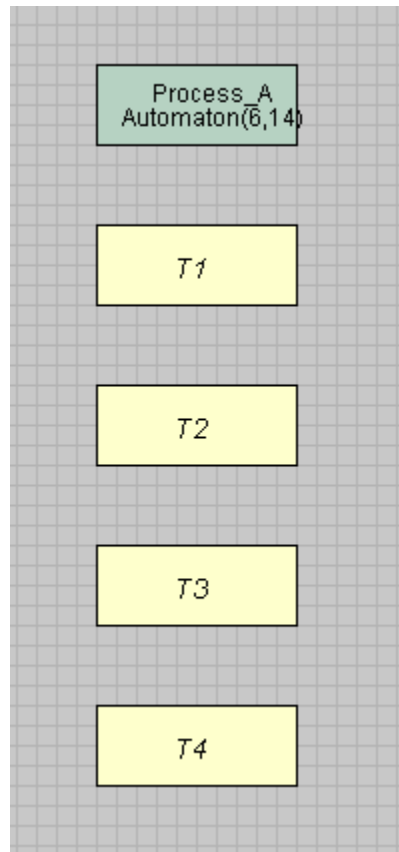
Na slikama 2.1, 2.2, 2.3 i 2.4 prikazani su skup zadataka, automat stanja koji je projektovan za definisanje aktivacije aperiodičnih zadataka, proces za aktiviranje aperiodičnih zadataka, te Schedulability analysis, u *Timetool* alatu, respektivno. Automat stanja je realizovan na način da ima tri stanja. Prvo stanje je početno, dok druga dva predstavljaju stanja aktivacije određenog aperiodičnog zadatka. Definisane su dvije konstante $N1$ i $N2$ koje predstavljaju vremena aktivacije aperiodičnih zadataka $A1$ i $A2$, respektivno. Dodatno, definisana je promjenljiva x tipa *clock* koja kada zadovolji uslov da je veća ili jednaka $N1$ postavlja automat u stanje aktivacije zadatka aperiodičnog $A1$, a kada zadovolji uslov da je veća ili jednaka $N2$ automat dolazi u stanje aktivacije aperiodičnog zadatka $A2$. Nakon toga se vraća u početno stanje i dobija vrijednost nula. Proces prikazan na Slici 2.3 koristi automat stanja definisan na prethodno izložen način i aktivira zadatak $A1$ u vremenskoj jedinici 6, dok zadatak $A2$ aktivira u vremenskoj jedinici 14. Schedulability analysis nam govori da je skup rasporediv što se i vidi na Slici 2.5. Naime, na Slici 2.5 je uočljivo da aperiodični zadaci koriste vrijeme kada periodični zadaci nisu u fazi izvršavanja i na taj način se oni sami izvršavaju. Dodatno, uočljivo je da su vremena izvršavanja periodičnih zadataka takođe zadovoljena, što i simulacija potvrđuje. Ovaj dio Zadataka 2 nalazi se u *xml* fajlu *zadatak2-1*.

Scheduling policy					
Rate Monotonic					<input checked="" type="checkbox"/> Preemptive
Name	B	Pr	C	D	T
A1	C	6	1	10	
A2	C	5	3	12	
T1	P	1	15	60	60
T2	P	2	6	48	48
T3	P	3	4	25	25
T4	P	4	3	15	15

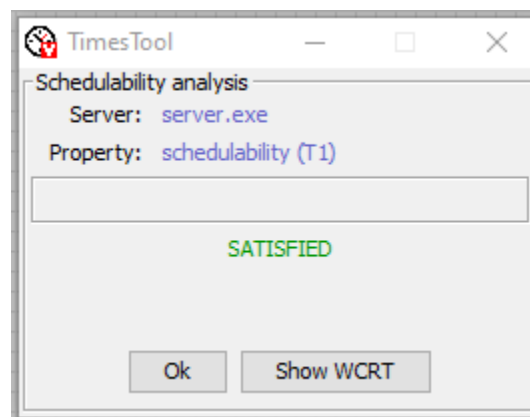
Slika 2.1 Skup zadataka za Zadatak 2 prikazan u *Timetool* alatu



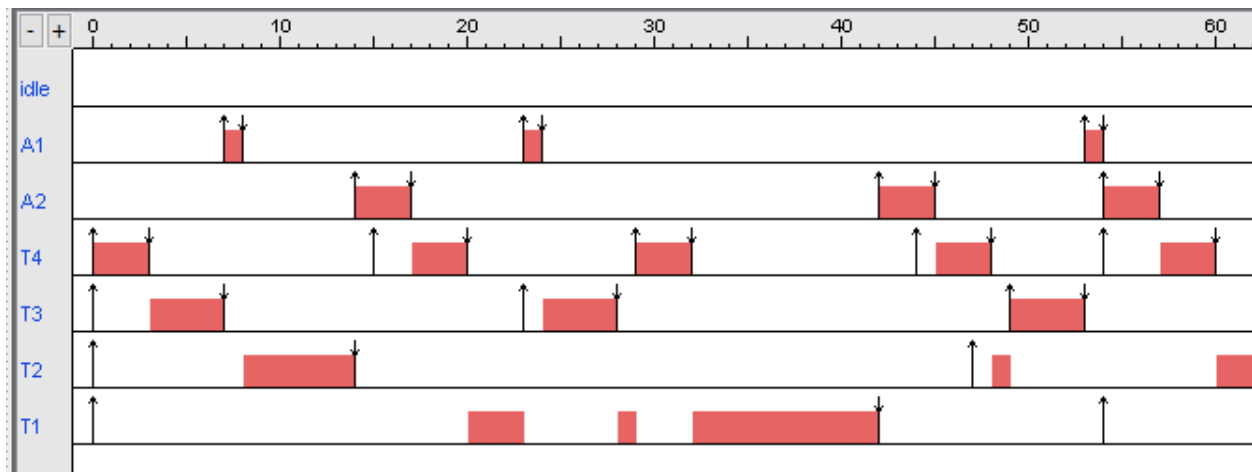
Slika 2.2 Automat stanja za Zadatak 2 implementiran u Timestool alatu



Slika 2.3 Proces za Zadatak 2 implementiran u Timestool alatu



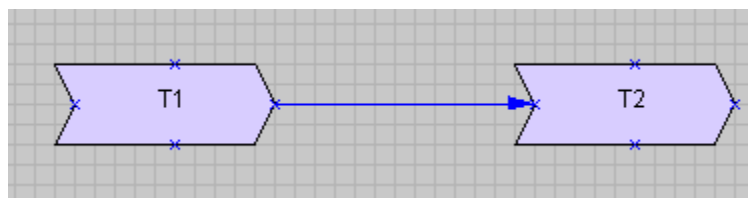
Slika 2.4 Schedulability analysis za skup iz Zadatka 2 bez zavisnosti zadataka



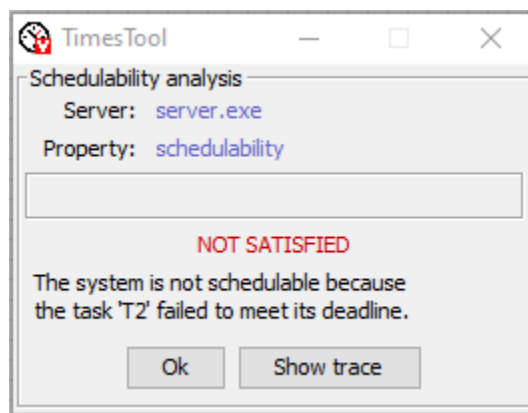
Slika 2.5 Gantov dijagram za skup zadataka iz Zadatka 2 bez zavisnosti zadataka

Drugi dio Zadatka 2 traži da se uvede određena zavisnost između proizvoljno odabaranih zadataka iz skupa prikazanog u Tabeli 2.1, te da se nakon toga izvrši simulacija u *Timestool* alatu.

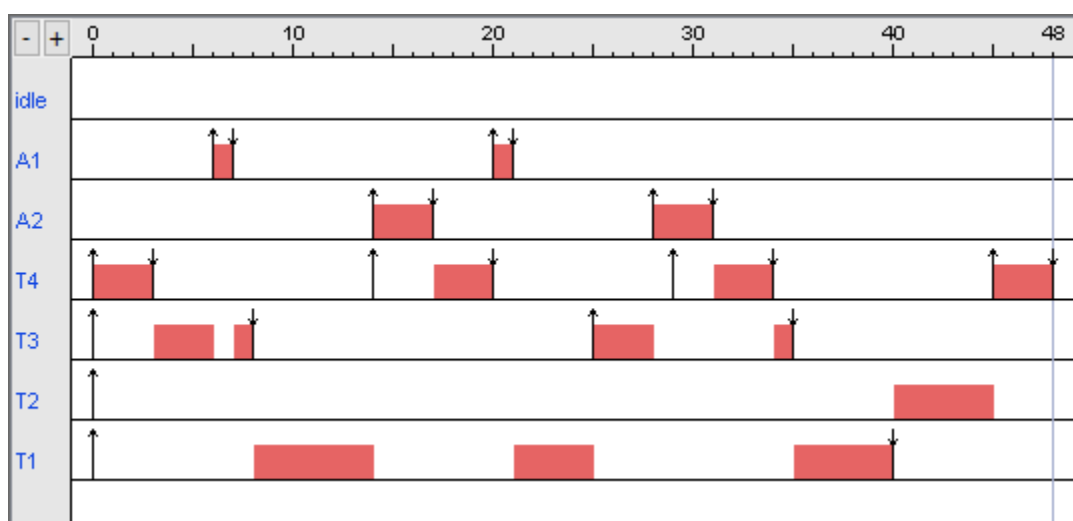
Za implementaciju ovog dijela Zadatka 2 zadržane su sve postavke iz prvog dijela ovog zadatka (skup, proces, automat stanja), te nije potrebno ponovo objašnjavati pojedine dijelove realizacije. Ono što se izdvaja u odnosu na prvi dio zadatka jeste zavisnost zadataka. Proizvoljno je odabrana zavisnost zadataka *T1* i *T2* na način da zadatak *T2* zavisi od zadatka *T1*, odnosno, zadatak *T2* se može izvršiti tek kad se izvrši zadatak *T1*. Kako je ova zavisnost realizovana prikazano je na Slici 2.6. Podešena su dva zadatka *T1* i *T2* i odabrana strelica koja ide od zadatka *T2* ka zadatku *T1* što pokazuje da zadatak *T1* zavisi od zadatka *T2*, tj. da će se on postati aktivan tek poslije završetka zadatka *T1*. Slika 2.7 jasno pokazuje da ovako definisan skup nije rasporediv. Jednostavno, zavisnost zadataka narušava koncept RM algoritma koji uvijek daje prednost zadatku višeg prioriteta. Samim tim, ako se uvede zavisnost zadataka (određeni zadatak nižeg prioriteta dobija prednost u odnosu na neki zadatak višeg prioriteta), više ne važi RM algoritam i skup postaje nerasporediv. Na Slici 2.8 prikazana je simulacija ovako realizovanog skupa zadataka. Vidljivo je da na prikazanom vremenskom intervalu zadatak *T2* nije izvršen u okviru definisanog perioda. Izloženi dio Zadatka 2 nalazi se u *xml* fajlu *zadatak2-2*.



Slika 2.6 Zavisnost zadatka *T2* od zadatka *T1* realizovana u *Timestool* alatu



Slika 2.7 Schedulability analysis za skup iz Zadataka 2 sa zavisnošću zadataka

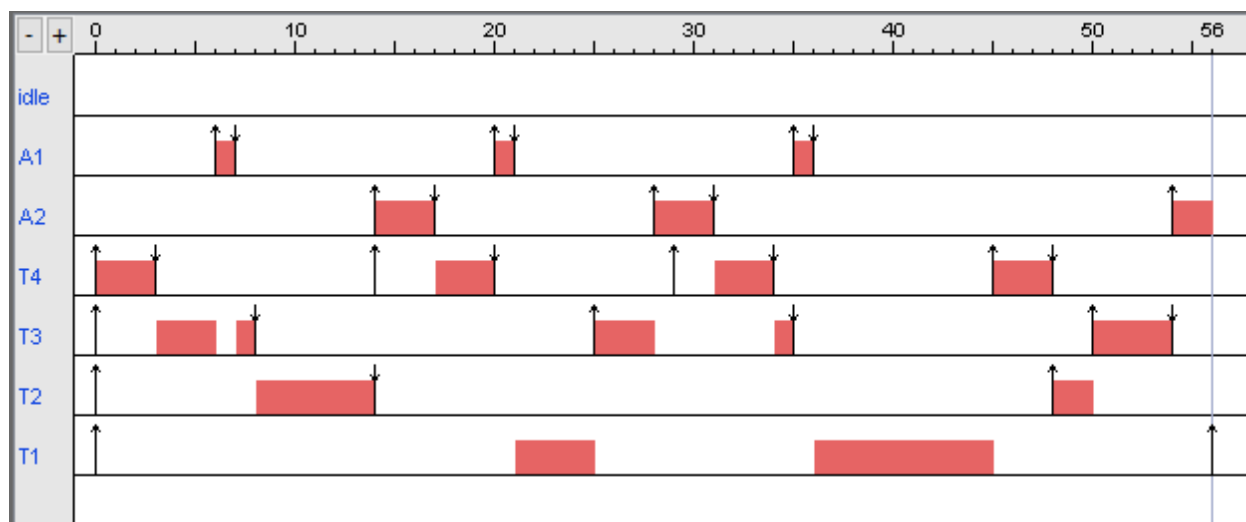


Slika 2.8 Gantov dijagram za skup zadataka iz Zadataka 2 sa zavisnošću zadataka

Analiziran je i slučaj zavisnosti zadataka *T1* i *T2* na način da zadatak *T1* zavisi od zadataka *T2*, tj. da će zadatak *T1* početi da se izvršava kada je izvršen zadatak *T2*. Na Slici 2.9 prikazana je jedina izmjena u odnosu na prethodno implementiran slučaj. Slika pokazuje da izvršavanje zadataka *T1* zavisi od izvršavanja zadataka *T2*. Ovako definisan skup nije rasporediv što se vidi na Gantovom dijagramu prikazanom na Slici 2.10. U trenutku 56 aktivira se nova instanca zadataka *T1*, a prethodna instanca nije izvršena do kraja. Samim tim skup nije rasporediv.



Slika 2.9 Zavisnost zadataka *T1* od zadataka *T2* realizovana u Timestool alatu



Slika 2.10 Gantov dijagram za skup zadataka iz Zadatka 2 sa zavisnošću zadataka (drugi slučaj)