ANALIZA I PRIKAZ VREMENSKOG SLEDA DOGAĐAJA U RTS-U SA RM ALGORITMOM PLANIRANJA

Sandra Đošić, Milun Jevtić
Elektronski fakultet Niš (sandra.djosic, milun.jevtic)@elfak.ni.ac.rs

Sadržaj – *U radu je predstavljena jedna realizacija softvera za analizu i grafički prikaz redosleda izvršenja zadataka sistema koji radi u realnom vremenu. Razmatrani su sistemi kod kojih je aktivan rate monotonic algoritam za raspoređivanje zadataka. Softver je realizovan iz dva dela. Prvi deo predstavlja model rate monotonic algoritma kojim se analizira sled događaja u sistemu. Drugi deo se odnosi na realizaciju grafičkog prikaza rasporeda izvršenja zadataka. Cilj softvera je da omogući da se na jednostavan i pregledan način sagleda kako parametri zadataka utiču na ispunjenje vremenskih zahteva za njihovo izvršenje u RT sistemu.*

1. UVOD

Osnovna karakteristika po kojoj se razlikuju sistemi za rad u realnom vremenu (RTS-*real-time systems*) od ostalih sistema jeste ponašanje u vremenu. Da bi RTS ispravno funkcionisao potrebno je ne samo da daje korektan rezultat na izlazu već i da ga da u tačno definisanom vremenskom intervalu, [1]. Danas, sistemi koji rade u realnom vremenu imaju veoma bitnu ulogu u mnogim oblastima svakodnevnog života, počevši od kontrole jednostavnih električnih uređaja pa do kontrole svemirskih stanica. Neke od tipičnih primena RTS-a su: u robotici, kontrola leta u avionima, kontrola saobraćaja, kontrola procesa u industriji, laboratorijama kao i u nuklearnim centralama, daljinska istraživanja podvodnih prostranstava, svemira, oblasti visokog rizika...

Da bi se svi zadaci jednog RTS-a izvršili striktno u predviđenim vremenskim rokovima treba napraviti algoritam rada planera koji će definisati koji zadatak i u koje vreme će se izvršiti u skladu sa vremenskim rokovima, prioritetima i dostupnim resursima, [2]. Postoji nekoliko kriterijuma za podelu algoritama po kome planeri rade, neki od njih su: priroda problema, karakteristike zadataka, konfiguracija sistema...

U zavisnosti od prirode problema algoritmi planera se mogu podeliti na: statičke, dinamičke i statičko-dinamičke. Za statičke algoritme planera je karakteristično da unapred poseduju određene informacije o zadacima koje se mogu iskoristiti prilikom definisanja algoritma. Ovi algoritmi su pogodni kod sistema kod kojih se unapred znaju vremena izvršenja zadataka, frekvencija pojave zahteva za njihovo izvršenje i slično, a prioriteti koji se dodeljuju zadacima su fiksni i ne mogu se menjati tokom izvršenja. Dinamički algoritmi planera su, za razliku od statičkih, definisani da rade sa zadacima kod kojih se unapred ne zna kada će se javiti zahtev za njihovo izvršenje i kod kojih nije poznato vreme izvršenja. Kombinacijom karakteristika statičkih i dinamičkih algoritama planera nastali su statičko-dinamički algoritmi.

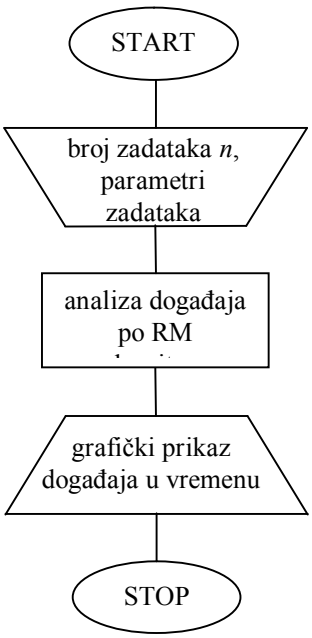
U zavisnosti od karakteristika zadataka algoritmi planera se mogu podeliti na: *preemptive* i *nonpreemptive*. U prvu grupu spadaju algoritmi planera kod kojih je karakteristično da se zadatak koji se trenutno izvršava može prekinuti dolaskom zadatka višeg prioriteta. Kasnije taj zadatak nastavlja sa izvršenjem bez uticaja na mogućnost prekoračenja rokova, odnosno ne ugrožavajući bilo koje vremenske karakteristike planera. Za drugu grupu planera karakteristično je da trenutno izvršenje zadatka ne može biti prekinuto bez obzira na prioritet dolazećeg zadatka.

Jedan od bazičnih algoritama planera kada se govori o planiranju izvršenja zadataka sa fiksnim prioritetima jeste RM (*rate monotonic*) algoritam. RM algoritam spada u grupu vremenski baziranih algoritama sa periodom zadatka kao osnovnim kriterijumom prilikom određivanja rasporeda izvršenja zadataka, [3]. Ovaj algoritam zadatku sa kraćim periodom pojavljivanja dodeljuje veći prioritet prilikom izvršenja. Kako je period zadatka konstantna veličina RM algoritam spada u grupu statičkih algoritama.

U okviru naših istraživanja RM algoritam je dosta zastupljen i često ga koristimo prilikom analize rada RTS-a, [4], [5], [6], kao jedan od mogućih algoritama po kome se zadaci u sistemu izvršavaju. RM algoritam planera je i baza ovog rada. U radu je predstavljen realizovani softver za analizu rada i prikaza vremenskog sleda događaja jednog RTS-a kod koga je aktivan RM algoritam planiranja izvršenja zadataka. Softver je realizovan uz pretpostavku da je RM algoritam planera *preemptive*. Na osnovu realizovanog softvera moguće je sagledati kako vremenski parametri RT (*real-time*) zadataka utiču na ispunjenje vremenskih zahteva u toku rada RTS-a.

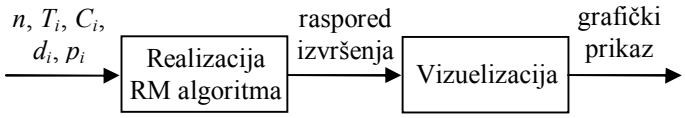
2. ALGORITAM ZA GRAFIČKI PRIKAZ DOGAĐAJA RTS-a

Na Sl. 1 prikazan je algoritam kojim se realizuje softver za analizu i prikaz vremenskog sleda događaja u RTS-u za aktivan RM algoritam planiranja.



Sl.1. *Algoritam za grafički prikaz događaja u vremenu jednog RTS-a za aktivan RM algoritam.*

Algoritam prikazan na Sl. 1 realizovan je iz dva dela, kao što je prikazano na Sl. 2.



Sl. 2. *Realizacija algoritma za grafički prikaz događaja.*

Prvi deo se odnosi na realizaciju samog RM algoritma gde su kao ulazni podaci neophodni parametri koji karakterišu date zadatke. Pre svega neophodno je navesti broj zadataka n, a zatim za svaki zadatak njegove osnovne parametre kao što su period zadatka T_i, najgore vreme izvršenja zadatka C_i, rok za izvršenje zadatka d_i kao i prioritet zadatka p_i. Izlazni podaci prvog dela predstavljaju ujedno i ulazne podatke drugog dela pri čemu ti podaci predstavljaju parametre rasporeda izvršenja zadataka u skladu sa RM algoritmom. Zadatak drugog dela je grafički prikaz rasporeda izvršenja zadataka.

2.1 REALIZACIJA RM ALGORITMA

Algoritam simulacije rada RM planera prikazan je na Sl. 3. Algoritam će biti objašnjen kroz jedan jednostavan primer tri RT zadatka, τ_1, τ_2 i τ_3. Osnovni parametri zadataka prikazani su u Tabeli 1.

Tabela 1. *Parametri zadataka τ_1, τ_2 i τ_3*

Zadatak	d_i	C_i
τ_1	10	2
τ_2	20	3
τ_3	30	5

Pretpostavljeno je da su periodi sva tri zadatka jednaki njihovim rokovima za izvršenje, tj. da je $T_i=d_i$ kao i da u isto vreme t=0 stižu zahtevi za izvršenje sva tri zadatka. U skladu sa RM planerom zadatak τ_1 ima najveći prioritet a zadatak τ_3 najmanji i važi da je $p_1>p_2>p_3$.

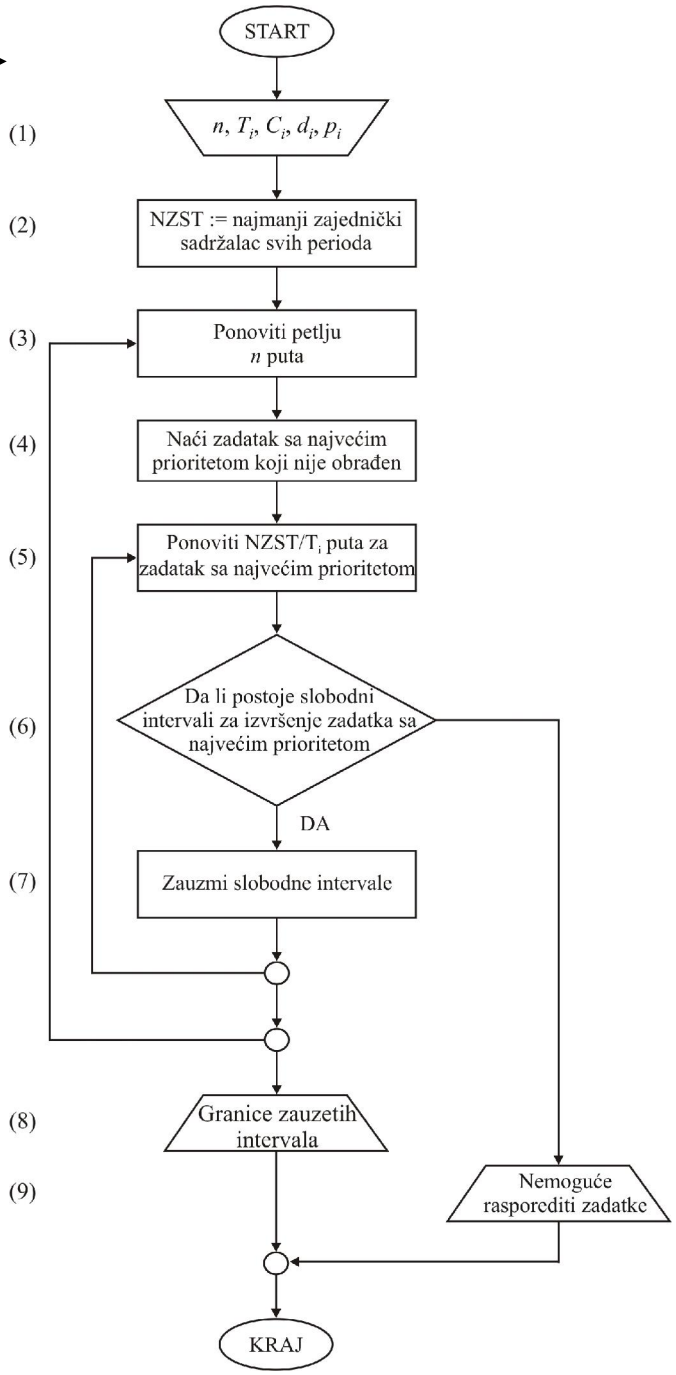
Ulazni podaci za algoritam sa Sl. 3. su broj RT zadataka n, period zadatka T_i, najgore vreme izvršenja zadatka C_i, rok za izvršenje zadatka d_i kao i prioritet zadatka p_i, korak (1).

U koraku (2) računa se najmanji zajednički sadržalac perioda svih zadataka. Za dati primer NZST je 60. Nadalje se algoritam ograničava samo na period od 60 vremenskih jedinica jer se nakon tog vremena raspored izvršenja sva tri zadatka ponavlja.

Korak (3) predstavlja prvu petlju koja se izvršava n puta tj. dok se ne obrade svi RT zadaci. Za dati primer ova petlja će se ponoviti tri puta.

U narednom koraku (4) traži se zadatak sa najvećim prioritetom koji nije obrađen i za konkretni primer to je zadatak τ_1.

Sada sledi nova petlja korak (5) koja se izvršava onoliko puta koliko ima zahteva za izvršenje zadatka definisanog u koraku (4) u toku perioda NZST. Za dati primer prvi zahtev za izvršenje zadatka τ_1 je u trenutku t=0. Kako je period ovog zadatka 10 vremenskih jedinica to znači da naredni zahtev za izvršenje stiže u trenutku t=10, pa zatim u trenutku t=20, t=30, t=40 i t=50. Sledi da će se kroz petlju u koraku (5) proći ukupno 6 puta.



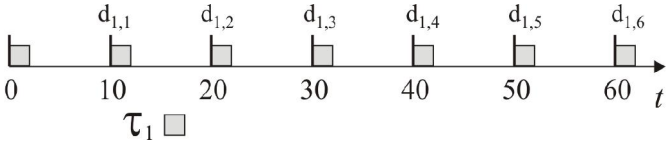
Sl. 3. *Realizacija RM algoritma.*

U okviru svakog prolaska kroz petlju u koraku (5) postavlja se pitanje korak (6) da li postoje slobodni intervali, dovoljno veliki, za izvršenje zadatka definisanog u koraku (4).

Ako je uslov ispunjen onda se zauzimaju ti slobodni intervali korak (7) i koriste za izvršenje zadatka definisanog u koraku (4).

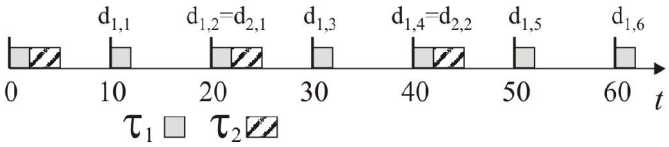
Ukoliko uslov, korak (6), nije ispunjen algoritam se završava porukom korak (9) da je nemoguće zadovoljiti sva vremenska ograničenja zadataka tj. rasporediti RT zadatke u skladu sa RM planerom.

U datom slučaju je uslov korak (6) ispunjen tako da zadatak τ_1 zauzima intervale na način prikazan na Sl. 4.



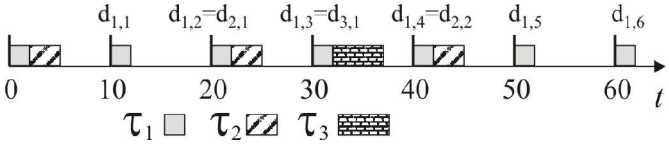
Sl. 4. *Raspored izvršenja zadatka* τ_1.

Nakon toga neophodno je ponoviti petlju korak (3) i izvršiti neophodne korake za zadatak sledeći po prioritetu, a to je τ_2. Prvi zahtev za izvršenje ovog zadatka dolazi u trenutku t=0. Kako je interval od t=0 do t=2 već zauzet, prvi naredni slobodni interval je od trenutka t=2 do t=10. Kako je u tom intervalu moguće izvršiti zadatak τ_2 (njegovo vreme izvršenja je 3 vremenske jedinice), zauzima se interval od t=2 do t=5. Kako se u intervalu od 60 vremenskih jedinica zadatak τ_2 javlja ukupno tri puta, to znači da kroz petlju korak (5) treba proći upravo toliko puta. Sledeći zahtev za izvršenje stiže u trenutku t=20 pri čemu je interval od t=20 do t=22 takođe već zauzet i prvi naredni slobodni interval je od trenutka t=22 do t=30. Sada zadatak τ_2 zauzima interval od t=22 do t=25. Poslednji zahtev za izvršenje, u posmatranom periodu od 60 vremenskih jedinica, stiže u trenutku t=40. Slično kao i u prethodna dva puta interval od t=40 do t=42 je zauzet tako da zadatak τ_2 zauzima interval od t=42 do t=45. Na Sl. 5. prikazan je raspored izvršenja zadatka τ_1 i τ_2.



Sl. 5. *Raspored izvršenja zadatka* τ_1 *i* τ_2.

Ostalo je rasporediti još zadatak τ_3. Ovaj zadatak se javlja dva puta u intervalu od 60 vremenskih jedinica pa će se i kroz petlju korak (5) proći dva puta. Prvo se ispituje da li je moguće naći slobodan interval koji bi zauzeo zadatak τ_3. Kako su zadaci τ_1 i τ_2 već zauzeli interval od t=0 do t=5 ispituje se prvi slobodni vremenski interval, a to je interval od t=5 do t=10. Kako je vreme izvršenja zadatka τ_3 jednako 5 vremenskih jedinica, a upravo toliko iznosi ovaj slobodni interval, zadatak τ_3 je moguće rasporediti. Sledeći zahtev za izvršenje stiže u trenutku t=30 i kako je interval od t=30 do t=32 zauzet zadatak τ_3 zauzima interval od t=32 do t=37. Raspored izvršenja sva tri zadatka τ_1, τ_2 i τ_3 prikazan je na Sl. 6.



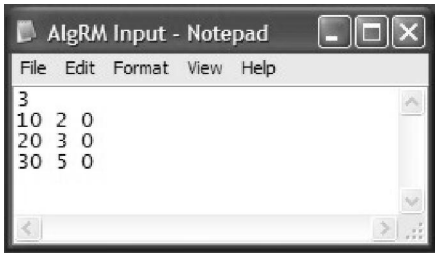
Sl. 6. *Raspored izvršenja zadatka* τ_1, τ_2 *i* τ_3.

Koristeći programski jezik C izvršena je softverska realizacija algoritma sa Sl. 3, napisan je programski kod i generisan izvršni fajl „AlgRM.exe“. Izvršni fajl se poziva iz komandne linije naredbom

ALGRM [<ulazni_fajl>] [<izlazni_fajl>]

pri čemu se imena ulaznog i izlaznog fajla specificiraju kao prvi i drugi pozivni parametar. Ukoliko se ovi parametri ne navedu, koriste se standardna imena ulaznog i izlaznog fajla „AlgRM Input.txt“ i „ProcVis.txt“.

Ulazni fajl programa AlgRM je u tekst formatu, sa parametrima odvojenim razmacima. Na Sl. 7 prikazan je ulazni fajl za prethodno korišćeni primer. Prilikom zadavanja ulaznih parametara u prvoj liniji fajla treba navesti celobrojni parametar n, koji predstavlja broj RT zadataka, a zatim u narednih n linija navesti parametre zadatka d_i, C_i i r_i.



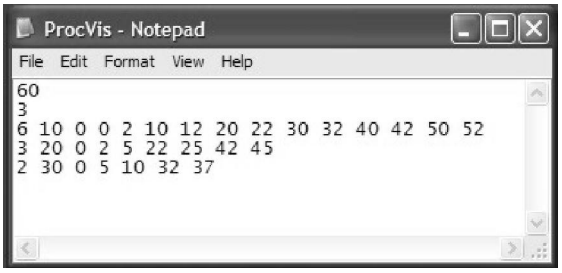
Sl. 7. *Ulazni fajl programa AlgRM.*

Izlazni fajl programa AlgRM je takođe u tekst formatu.

Ukoliko je nemoguće rasporediti RT zadatke u skladu sa RM planerom, izlazni fajl sadrži samo jedan red sa tekstom „Nemoguće rasporediti zadatke“.

Ako se zadaci mogu rasporediti u skladu sa RM planerom onda izlazni fajl sadrži parametre odvojene razmacima. U prvom redu fajla je parametar koji označava najmanji zajednički sadržalac perioda svih zadataka NZST, a u drugom parametar koji označava broj zadataka n. U svakoj od sledećih n linija, prvi parametar je broj intervala koje taj zadatak zauzima u periodu od NZST vremenskih jedinica. Ovo se odnosi i na slučajeve kada je zadatak nemoguće izvršiti odjednom već ga je potrebno podeliti. Odnosno kada dolazi do prekida izvršenja trenutno aktivnog zadatka zbog izvršenja zadatka većeg prioriteta. Drugi parametar u liniji je period zadatka T_i, a treći trenutak kada stiže prvi zahtev za izvršenje zadatka r_i. Preostali parametri predstavljaju početke i krajeve intervala u kojima se zadatak izvršava, bez obzira da li se zadatak izvršava odjednom, ili ga je potrebno podeliti.

Na Sl. 8 prikazan je izlazni fajl programa AlgRM za posmatrani primer. Iz izlaznog fajla može se videti da će zadatak τ_1 zauzeti ukupno 6 intervala od t=0 do t=2, od t=10 do t=12, od t=20 do t=22, od t=20 do t=32, od t=40 do t=42 i od t=50 do t=52. Slično se može zaključiti i za zadatke τ_2 i τ_3. Dobijeni intervali u kojima se izvršavaju zadaci identični su onima prikazanim na Sl. 6.

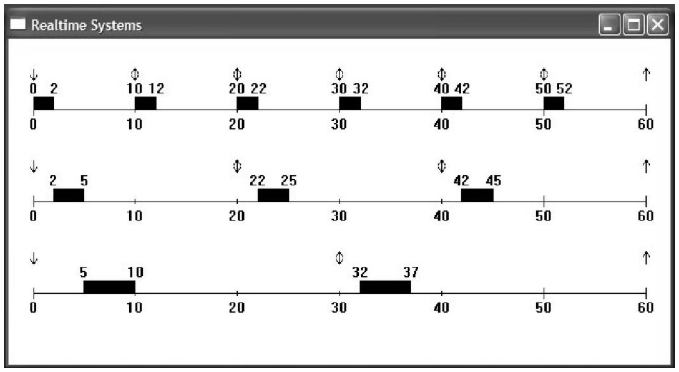


Sl. 8. *Ilazni fajl programa AlgRM.*

2.2 VIZUELIZACIJA

Za vizuelizaciju simulacije rada RM algoritma realizovan je program ProcVis koji kao ulaz koristi izlazni fajl programa AlgRM. Zadatak programa ProcVis je da na osnovu podataka iz ulaznog fajla programa AlgRM otvara prozor u kome iscrtava kako se i kada RT zadaci izvršavaju.

Na Sl. 9 prikazan je izlazni fajl programa ProcVis za posmatrani primer. Može se zaključiti da je raspored izvršenja zadataka dobijen programom ProcVis isti kao onaj prikazan na Sl. 6, čime je dokazana korektnost predstavljenog algoritma.



Sl. 9. *Izlazni fajl programa ProcVis.*

3. ZAKLJUČAK

U radu je predstavljen realizovani softver za analizu i prikaz vremenskog sleda događaja u jednom RTS kod koga je aktivan RM planer. Softver je realizovan iz dva dela.

Prvi deo se odnosi na realizaciju modela rada RM planera. Treba istaći da je RM planer jedan od bazičnih i češće korišćenih planera u sistemima koji rade u realnom vremenu. Kao takav često se koristi i prilikom različitih analiza RTS-a i veoma je zastupljen i u okviru naših istraživanja. To su upravo i razlozi zašto je realizovan softver baš za ovaj planer.

Drugi deo realizovanog softvera odnosi se na grafički prikaz rasporeda izvršenja zadataka. Ovaj vizulelni način prezentacije je mnogo pregledniji i jednostavniji za korisnika pogotovo ako se radi o velikom broju RT zadataka.

Cilj realizacije ovog softvera bio je da se na osnovu vremenskih parametara RT zadataka izvrši analiza RTS-a. Kao rezultat analize dobija se informacija da li je moguće zadatke RTS-a izvršiti u skladu sa RM planerom ili ne. Ako je izvršenje moguće dobija se vizuelna predstava o redosledu po kome se RT zadaci izvršavaju. Na osnovu realizovanog softvera moguće je sagledati kako vremenski parametri zadataka utiču na ispunjenje vremenskih zahteva za njihovo izvršenje u RT sistemu. Za realizovani softver urađen je niz testova koji su dokazali njegovu korektnost. Jedan od tih testova prikazan je i u ovom radu.

LITERATURA

[1] N. Nissanke, *Raltime Systems,* Prentice Hall, 1997.

[2] K. Juvva, *Real-Time Systems,* Carnegie Mellon University 18-849b Dependable Embedded Systems ili www-2.cs.cmu.edu/~koopman/des_s99/real_time/

[3] F. Cottet, J. Delacroix, Z. Mammeri, *Scheduling in Real-Time Systems*, John Wiley & Sons, 2002.

[4] S. Đošić, M. Jevtić, "Analysis of Real-Time Systems Timing Constrains", in *Proc. of Small Systems Simulation Symposium 2010*, Niš, pp. 56-60, February 2010.

[5] M. Jevtić, M. Cvetković, S. Brankov, "Task Execution in Real-Time Systems for Industrial Control and Monitoring", *Electronics*, Faculty of Electrical Engineering University of Banjaluka, vol. 6, no. 2, pp. 56-61, December 2002.

[6] S. Brankov, M. Jevtić, "Algoritmi planera HRTS-a sa tolerisanjem greške", Zbornik XLVI konferencije za elektroniku, telekomunikacije, računarstvo, automatiku i nuklearnu tehniku ETRAN 2002, Banja Vrućica, pp. I.70-I.73, Juni 2002.

Abstract – This paper presents realized software for analysis and graphical illustration of real-time tasks execution sequences in one real-time system (RTS). Focus of the paper is RTS with active rate monotonic scheduling algorithm. The software is realized as two part project. The first part presents model of rate monotonic algorithm which can be used for analysis of tasks execution in the system. The second part is visual presentation of tasks scheduling. With realized software we can find out how timing constraints of real-time tasks affect on tasks scheduling in one RTS.

ANALYSIS AND GRAPHICAL ILLUSTRATION OF TASKS SCHEDULING IN ONE RTS WITH ACTIVE RM SCHEDULING ALGORITHM

Sandra Đošić, Milun Jevtić