Praktikum iz operativnih sistema

Lekcija 5: Raspoređivanje periodičnih poslova

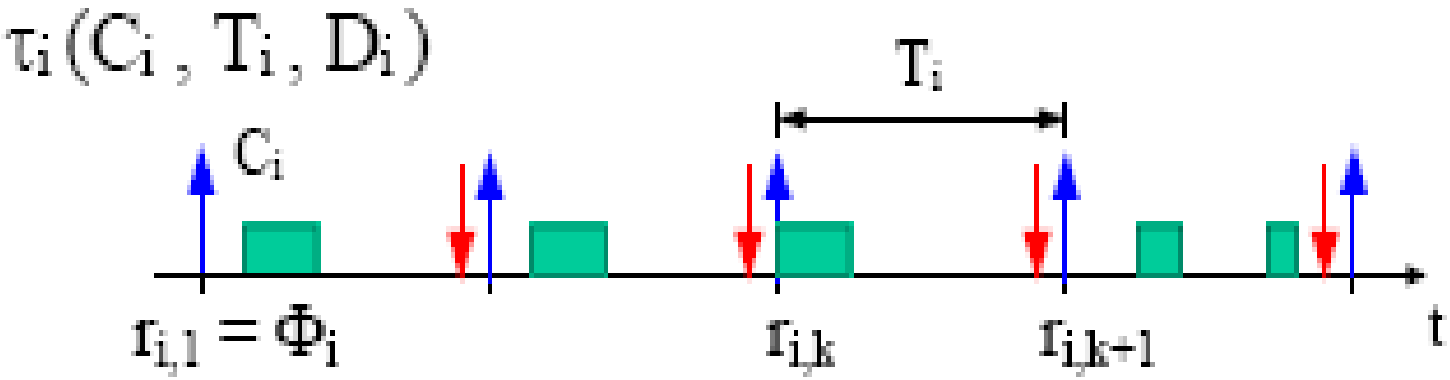
zima 2019/2020

Prof. dr Branimir Trenkić

Uvod

- *Real-time* aplikacija:
- Više *istovremenih* ***periodičnih poslova sa individualnim vremenskim zahtevima***
- Operativni sistem:
- Mora garantovati za svaku instancu ovih poslova
  a) da će biti ***regularno aktivirana*** (odgovarajućim intezitetom)
  b) da će izvršenje biti ***kompletirano*** unutar njenog deadline-a

Definicije



Γ skup periodičnih poslova

τ_i proizvoljni periodični posao

$\tau_{i,j}$ *j*-ta instanca (job) posla τ_i

$r_{i,j}$ trenutak aktiviranje *j*-te instance posla τ_i

Φ_i faza posla τ_i: ($\Phi_i = r_{i,1}$);

D_i relativni *deadline* posla τ_i

Definicije

T_i perioda posla τ_i

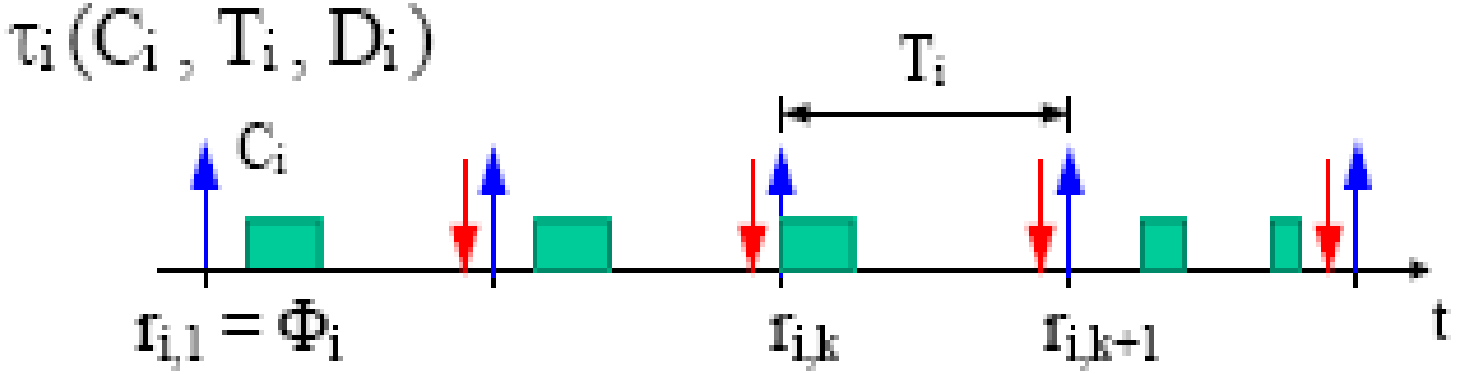
$d_{i,j}$ apsolutni *deadline* *j*-te instance posla τ_i;

$$d_{i,j} = \Phi_i + (j - 1)T_i + D_i$$

$s_{i,j}$ početak izvršavanja *j*-te instance posla τ_i;

trenutak kada instanca $\tau_{i,j}$ počinje izvršavanje

$f_{i,j}$ vreme završetka izvršavanja *j*-te instance posla τ_i



Hipoteze

A1. Instance periodičnog posla τ_i se ***regularno aktiviraju*** **konstantnim intezitetom**. Interval T_i između dve susedne aktivacije se naziva **periodom** posla

A2. Sve instance periodičnog posla τ_i imaju ***isto vreme izvršavanja*** u najgorem slučaju C_i

A3. Sve instance periodičnog posla τ_i imaju ***isti relativni deadline*** D_i, koji je ***jednak*** periodi T_i

A4. Svi poslovi u Γ su **nezavisni**

Definicije

- Ako predpostavimo da hipoteze A1, A2, A3, i A4 važe

- Periodični posao τ_i je potpuno određen sa tri parametra:

$$\Gamma = \{\tau_i(\Phi_i, T_i, C_i), i = 1,\ldots,n\}$$

vreme aktiviranja i ***apsolutni deadline k***-te instance:

$$r_{i,k} = \Phi_i + (k-1)T_i$$

$$d_{i,k} = r_{i,k} + T_i = \Phi_i + kT_i$$

Definicije

- Drugi parametri:
- **Vreme odziva instance**. To je vreme (mereno od vremena aktiviranja) ***za koje je instanca kompletirana***: $\mathbf{R_{i,k} = f_{i,k} - r_{i,k}}$

- **Kritična tačka**. To je ***trenutak*** u kome će aktiviranje instance proizvesti ***najduže vreme odziva***
- **Kritični vremenski interval**. To je ***interval*** između ***kritične tačke*** i ***trenutka kompletiranja izvršavanja*** odgovarajuće instance

Definicije

- **Relativni *Jitter* aktivacije**. Maksimalna varijacija ***startnih vremena u odnosu na aktivaciju*** dve ***susedne*** instance:

$$RRJ_i = \max_k |(s_{i,k} - r_{i,k}) - (s_{i,k-1} - r_{i,k-1})|$$

- **Apsolutni *Jitter* aktivacije**. Maksimalna varijacija ***startnih vremena u odnosu na aktivaciju bilo koje dve*** instance:

$$ARJ_i = \max_k (s_{i,k} - r_{i,k}) - \min_k (s_{i,k} - r_{i,k})$$

Definicije

- **Relativni *Jitter* završetka**. Maksimalna varijacija ***vremena završetka u odnosu na aktivaciju*** dve ***susedne*** instance:

$$RFJ_i = \max_k |(f_{i,k} - r_{i,k}) - (f_{i,k-1} - r_{i,k-1})|$$

- **Apsolutni *Jitter* završetka**. Maksimalna varijacija ***vremena završetka u odnosu na aktivaciju*** ***bilo koje dve*** instance:

$$AFJ_i = \max_k (f_{i,k} - r_{i,k}) - \min_k (f_{i,k} - r_{i,k})$$

Definicije

- **Relativni *Jitter* izvršenja**. Maksimalna varijacija ***intervala izvršenja*** ***dve susedne*** instance:

$$\mathbf{REJ_i = max_k \left|(f_{i,k} - s_{i,k}) - (f_{i,k-1} - s_{i,k-1})\right|}$$

- **Apsolutni *Jitter* izvršenja**. Maksimalna varijacija ***intervala izvršenja*** ***bilo koje dve*** instance:

$$\mathbf{AEJ_i = max_k\ (f_{i,k} - s_{i,k}) - min_k\ (f_{i,k} - s_{i,k})}$$

Izvodljivost

- Periodični posao τ_j je ***izvodljiv***

- ***sve*** njegove ***instance*** se izvrše pre njihovih *deadline*-ova

- Za skup poslova Γ se kaže da je **izvodljiv** ili **rasporedljiv**

- su svi poslovi iz Γ izvodljivi

Iskorišćenost procesora

- Za dati skup Γ, *n* periodičnih poslova, **faktor iskorišćenosti procesora, *U*** je ***deo*** procesorskog vremena **potrošenog na** izvršenje poslova iz skupa Γ
- C_i/T_i je ***deo procesorskog vremena*** potrošenog na izvršenje posla τ_i

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

- ***U*** – ***mera opterećenja CPU*** izazvanog izvršavanjem skupa periodičnih poslova

Iskorišćenost procesora

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

- ***Poboljšanje iskorišćenosti (ili povećanje opterećenja)*** se može postići na dva načina:
  - ***a) povećanjem*** vremena ***izvršenja*** poslova (***C_i***)
  - ***b) smanjenjem*** njihovih ***perioda*** (***T_i***)

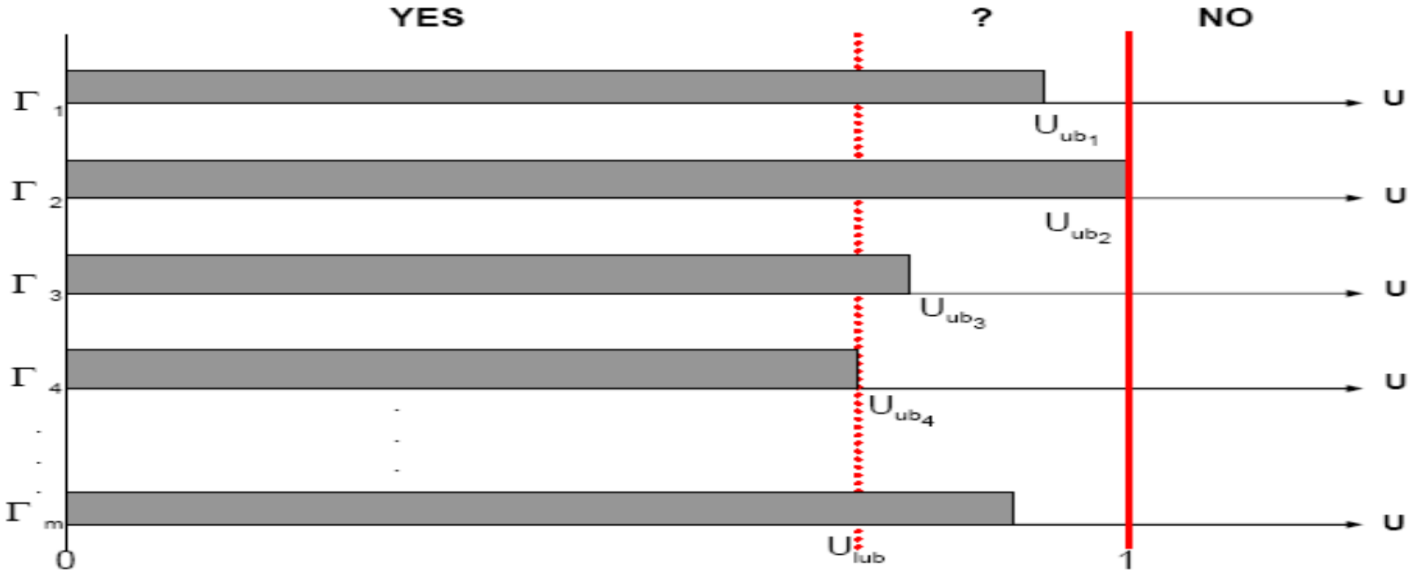
- Postoji ***maksimalna vrednost U (U_{ub})***
  - ***ispod koje*** je Γ **rasporedljiv**
  - ***iznad koje*** Γ **nije rasporedljiv**

Iskorišćenost procesora

- Ta ***maksimalna vrednost U zavisi od***:
  - **skupa poslova** (međusobne relacije između perioda poslova u skupu)
  - **algoritma** koji se koristi za raspoređivanje
- $U_{ub}(\Gamma, A)$ – ***gornja granica*** vrednosti procesorske iskorišćenosti za skup poslova Γ i algoritmam ***A***
- Kada je $U = U_{ub}(\Gamma, A)$ – skup Γ
  - ***potpuno koristi*** procesor
  - je ***rasporedljiv sa A***
- Povećanje vremena izvršavanja bilo kog posla će usloviti neizvodljivost

Iskorišćenost procesora

- U daljoj analizi ***fiksiramo A***
- Za dati algoritam A, $U_{lub}(A)$ – ***najniža gornja granica*** minimalni faktor iskorišćenosti u odnosu na sve skupove poslova koji potpuno koriste procesor $U_{lub}(A) = \min_{\Gamma} U_{ub}(\Gamma, A)$



Test rasporedljivosti

- **$U_{lub}(A)$** definiše važnu karakteristiku algoritma raspoređivanja, jer omogućuje jednostavnu ***proveru rasporedljivosti*** nekog skupa poslova

- Za bilo koji skup poslova čiji je faktor iskorišćenosti procesora **ispod $U_{lub}(A)$** možemo reći da je **rasporedljiv** sa algoritmom

- Ako je faktor iskorišćenosti nekog skupa poslova ***veći od jedan*** skup poslova **ne može biti rasporedljiv** za bilo koji algoritam

Ciklično raspoređivanje

- ***Timeline Scheduling*** **(TS)**
- ***Najčešće korišćeni pristup*** u raspoređivanju periodičnih poslova ***u vojnim sistemima***, ***navigacionim sistemima***, ***sistemima za monitorisanje***
- Primeri:
- Sistemi za kontrolu avio-saobraćaja
- Space Shuttle
- Boeing 777

Ciklično raspoređivanje

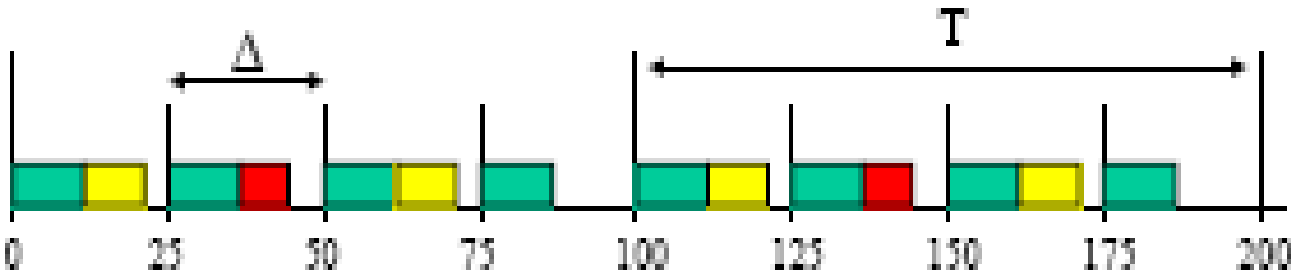
- *Metod*:
- ***Vremenska osa*** je ***podeljena*** na intervale jednakih dužina (***time slice***).
- Po ***off-line principu*** ***svakom intervalu*** se dodeljuje ***jedan ili više poslova*** na izvršenje
- ***Dodeljivanje*** mora biti ***saglasno aplikacionim zahtevima***
- ***Tajmer sinhronizuje aktivaciju poslova***
  - ***Aktivacija*** na početku time slice-a

Primer

task	f	T
A	40 Hz	25 ms
B	20 Hz	50 ms
C	10 Hz	100 ms

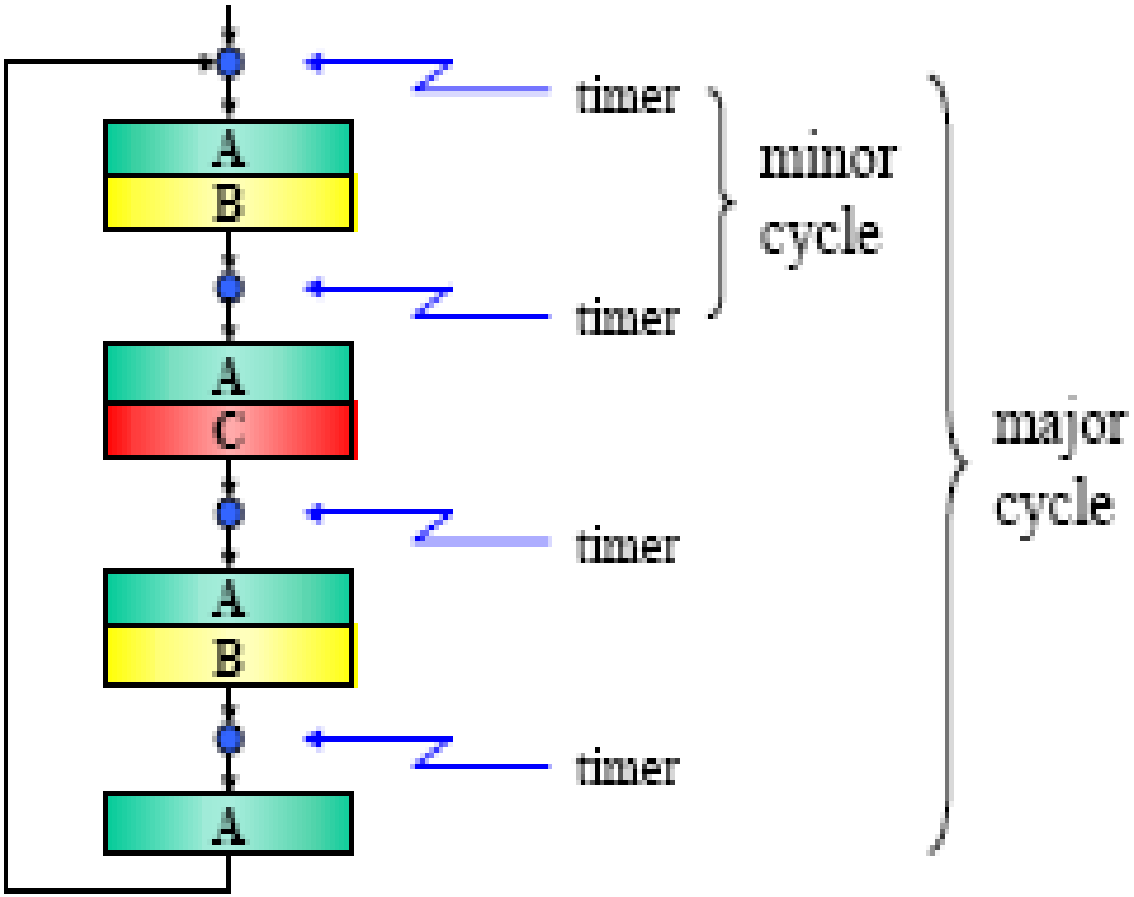
Δ = NZD (**minor cycle**) ***Kratki ciklus***

T = NZS (**major cycle**) ***Dugi ciklus***



Garancija izvodljivosti: $\begin{cases} C_A + C_B \leq 25ms \\ C_A + C_C \leq 25ms \end{cases}$

Realizacija



Ciklično raspoređivanje

- Prednosti:
- ***Jednostavna realizacija*** (ne zahteva se real-time operativni sistem)
- ***Dodatno procesorsko opterećenje*** prouzrokovano izvršavanjem algoritma je ***veoma malo***
- Omogućuje ***kontrolu jitter-a***

Ciklično raspoređivanje

- Nedostaci:

- Tipični za ***off-line*** tehnike ***raspoređivanja***
- ***Nije robustan*** u toku preopterećenja
- ***Komplikovan*** postupak ***proširenja rasporeda***
- Nije jednostavno upravljanje aperiodičnim poslovima

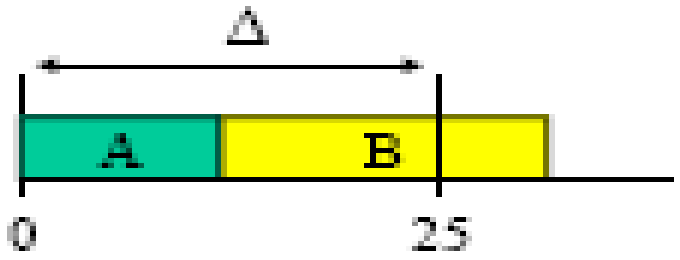
Problem preopterećenja

Šta raditi u slučaju da posao nije kompletiran na granici kratkog (minor) ciklusa (**overrun**)?

1. Pustiti da se ***izvršavanje nastavi***
   - može se dogoditi ***domino – efekat*** na sve ostale poslove (timeline break)
2. ***Prekid*** (abort) izvršenja posla
   - sistem može ostati u nekonsistentnom stanju, izlažući opasnosti njegovo korektno ponašanje

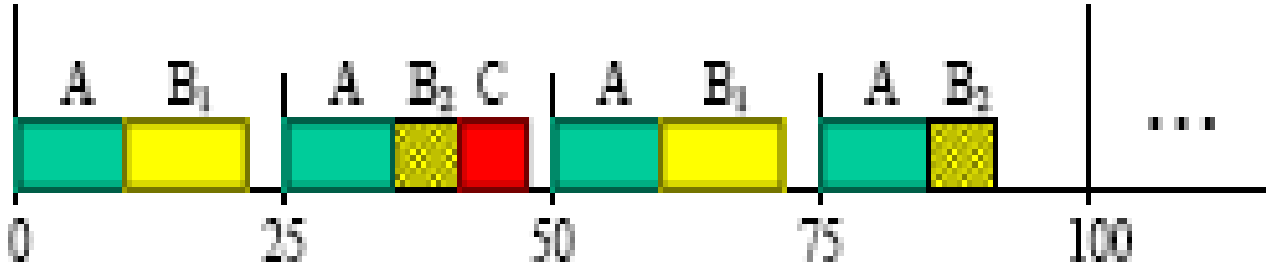
Proširljivost

- Ako je potreban ***upgrade*** jednog ili više ***poslova***, potrebno je ponovo uraditi ***re-dizajn čitavog rasporeda***
- Primer: izvršen je *upgrade* posla B $(C_A + C_B > \Delta)$



Proširljivost

- Mora se ***podeliti*** posao B na dva podposla (B_1, B_2) i ponovo napraviti raspored:



Garancija izvodljivosti:

$$\begin{cases} C_A + C_{B_1} \leq \Delta \\ C_A + C_{B_2} + C_C \leq \Delta \end{cases}$$

Proširljivost

- ***Promena frekvencije*** nekog posla je veoma značajna za raspoređivanje

task	T	T
A	25 ms	25 ms
B	50 ms	40 ms
C	100 ms	100 ms

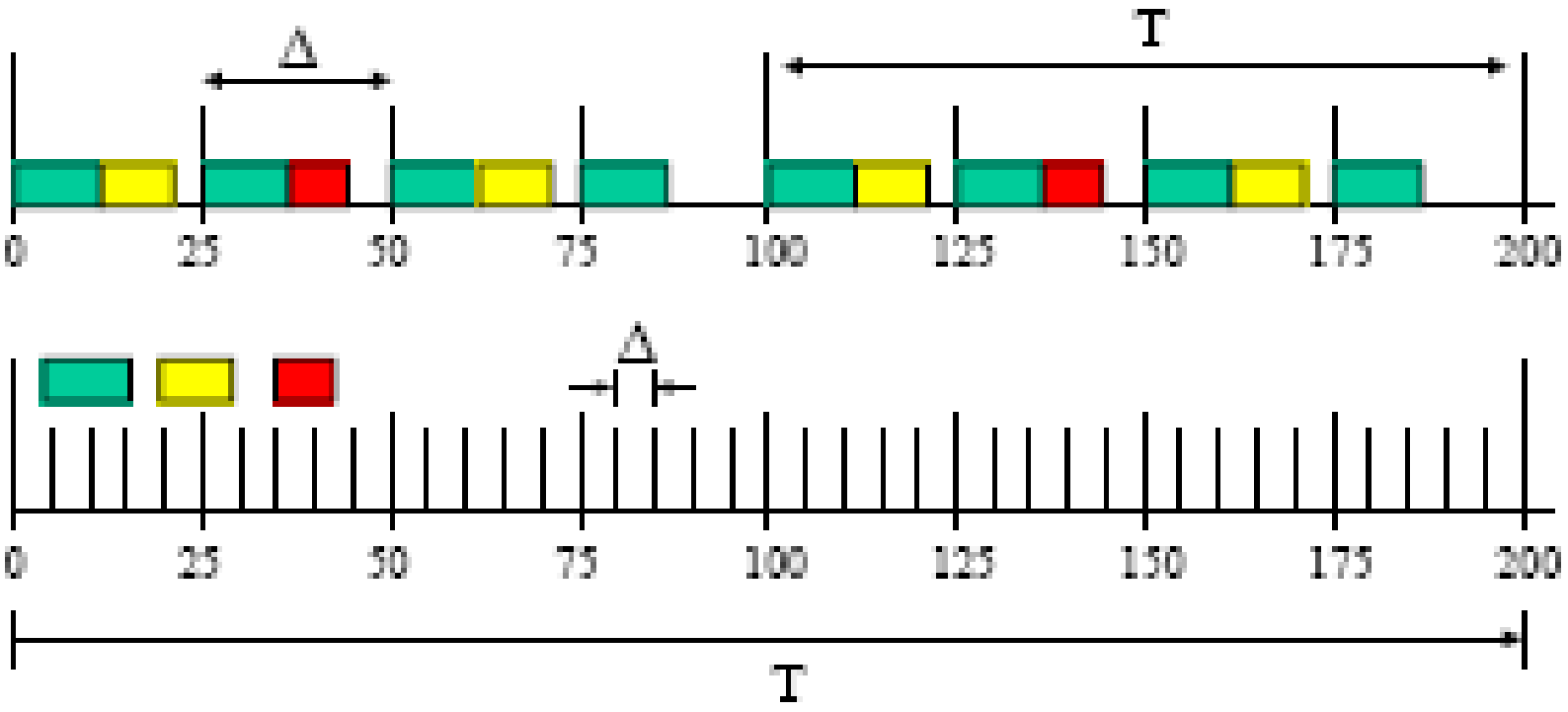
pre **posle**

minor cycle: $\Delta = 25$ $\Delta = 5$

major cycle: $T = 100$ $T = 200$

(40 sync. per cycle!)

Primer



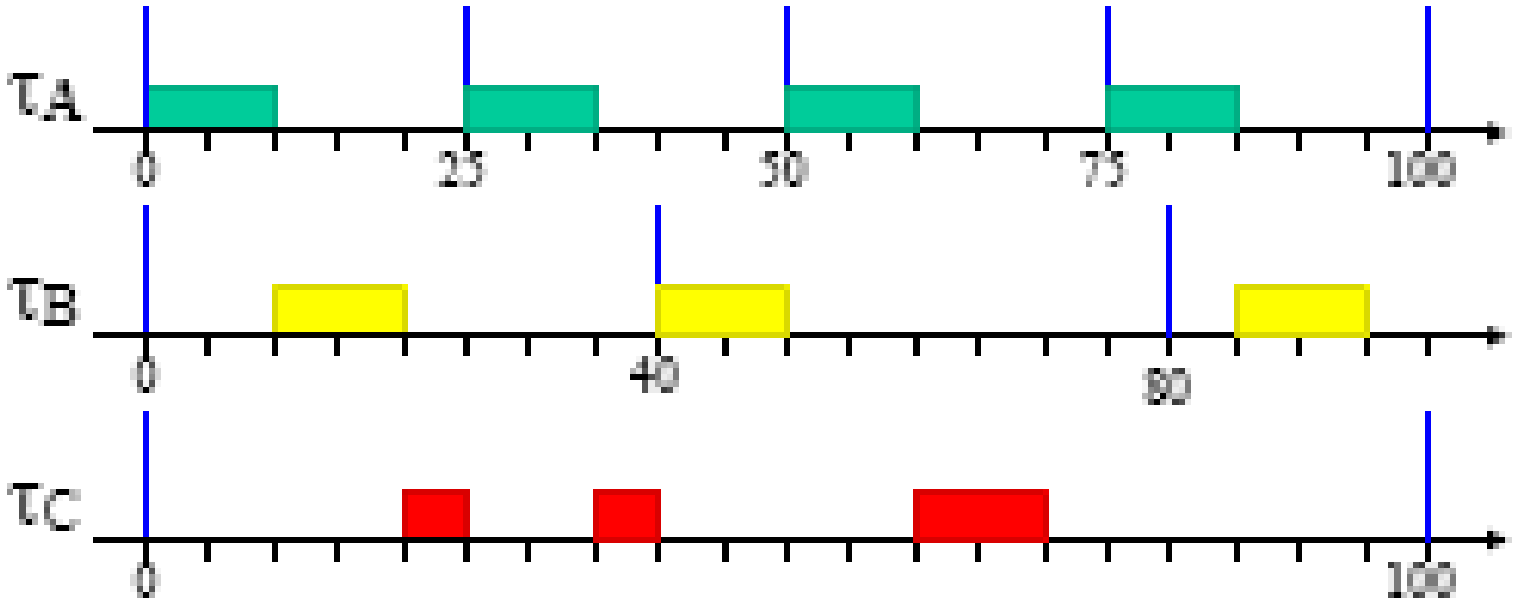
Deliti procedure A, B i C saglasno novom time-slice!

Prioritetno raspoređivanje

- ***Metod***:
- ***Svakom poslu*** se dodeljuje ***prioritet saglasno vremenskim zahtevima*** (timing-u posla)
- ***Izvodljivost rasporeda*** se verifikuje ***analitičkim tehnikama***
- Poslovi se izvršavaju na kernelu koji se bazira na prioritetima
- U biti ***prekidajući*** (preemptive) ***algoritmi***

Algoritam monotonog inteziteta

- ***Rate Monotonic, RM***
- Svakom poslu se dodeljuje ***fiksni prioritet***, ***saglasno*** njegovom ***intezitetu aktivacije*** [Lui & Layland ’73].



Verifikacija izvodljivosti

- Svaki posao koristi procesor deo ukupnog vremena:

$$U_i = C_i/T_i$$

- Odatle je ukupna ***iskorišćenost procesora***:

$$U_p = \sum_{i=1}^{n} \frac{C_i}{T_i}$$

U_p je mera **opterećanja procesora**

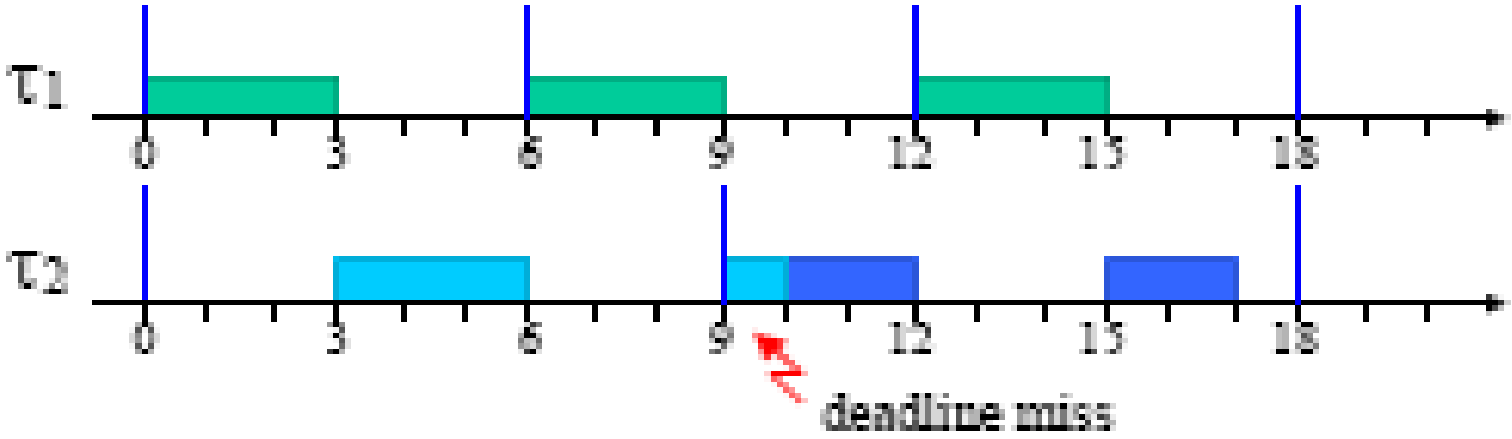
Potreban uslov

- **Ako je $U_p > 1$ procesor je preopterećen – skup poslova ne može biti rasporedljiv**

- Međutim, postoje slučajevi u kojima je $U_p < 1$, ali skup poslova ***nije rasporedljiv*** sa RM-om

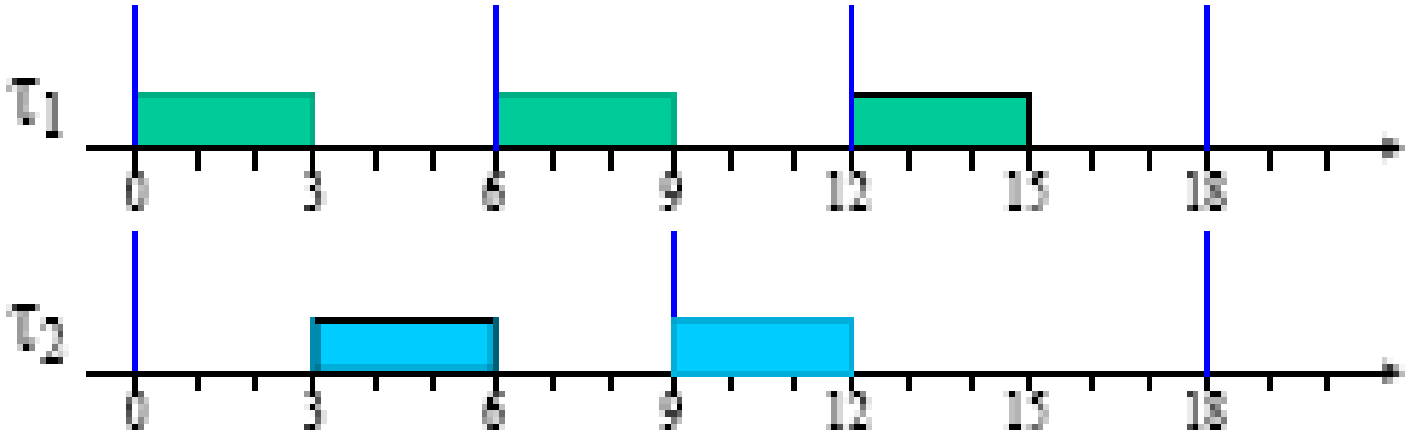
Neizvodljivi RM

$$U_p = \frac{3}{6} + \frac{4}{9} = 0.944$$



Gornja granica iskorišćenosti

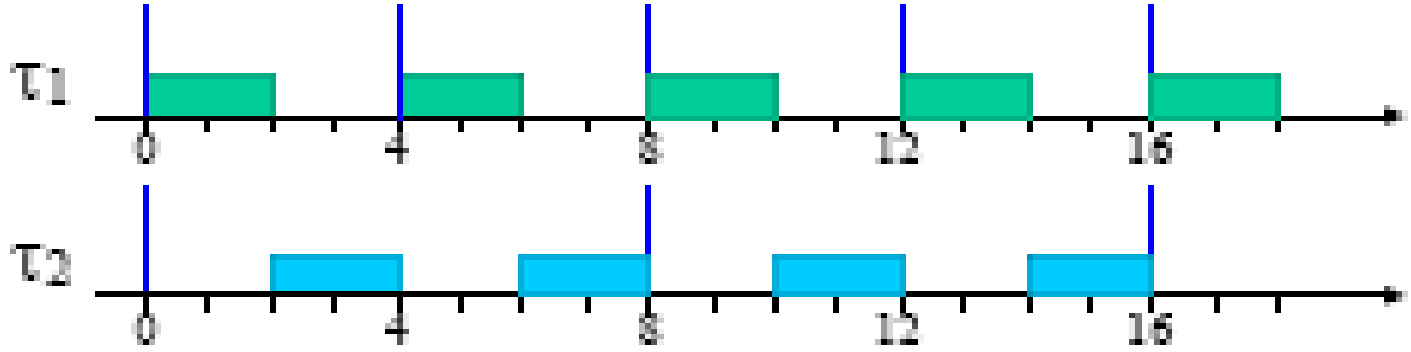
$$U_p = \frac{3}{6} + \frac{3}{9} = 0.833$$



Napomena: Ako se C_1 ili C_2 poveća, τ_2 će ***prekoračiti*** svoj deadline!

Različite gornje granice

$$U_p = \frac{2}{4} + \frac{4}{8} = 1$$

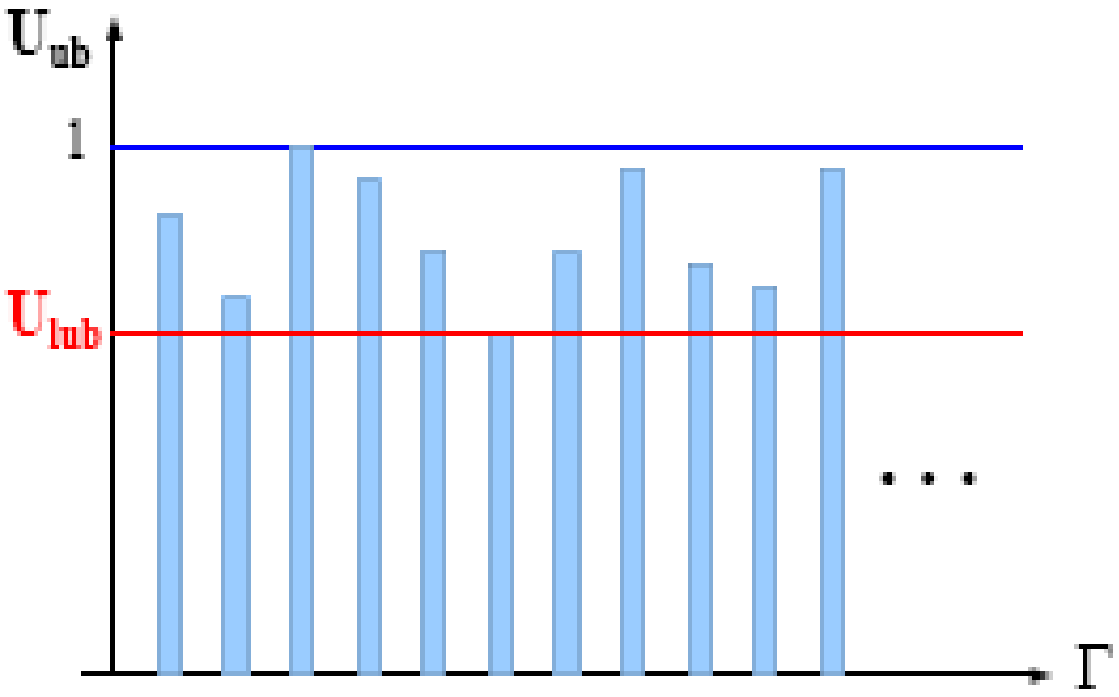


Zaključak:

Gornja granica U_{ub} zavisi od ***prirode skupa poslova***

Najniža gornja granica

- U cilju ***provere rasporedljivosti*** - važno je naći ***minimalnu takvu vrednost***
- U_{lub} - minimalna gornja granica



Dovoljan uslov

- **Ako je $U_p \leq U_{lub}$, skup poslova je <u>sigurno rasporedljiv</u> sa RM algoritmom**.

- Ako je **$U_{lub} < U_p \leq 1$**, ne može se ***ništa reći o izvodljivosti*** rasporeda na datom skupu poslova

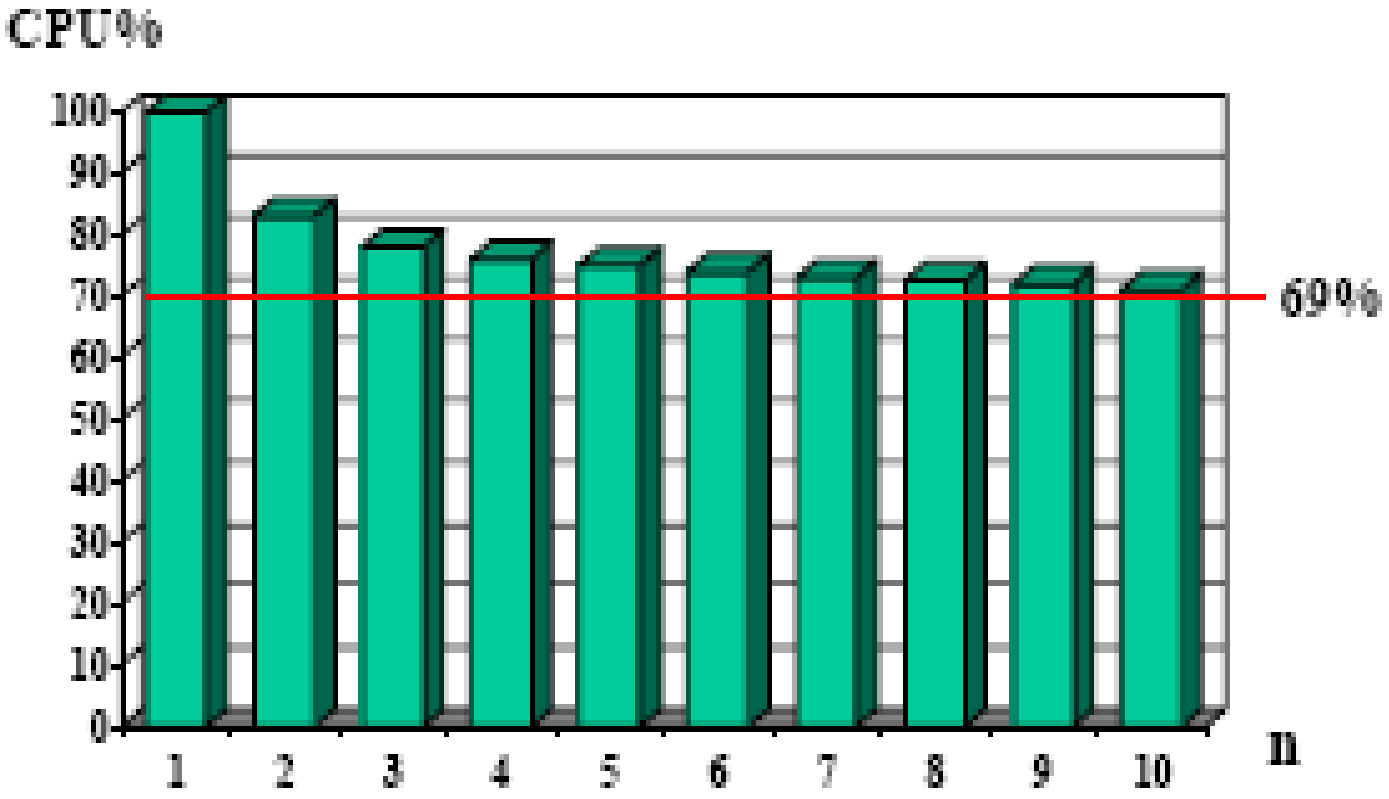
U_{lub} za RM

- 1973, Lui & Layland su dokazali da ***za skup od n periodičnih poslova***:

$$U_{\text{lub}}^{RM} = n\left(2^{1/n} - 1\right)$$

for $n \to \infty$ $U_{lub} \to \ln 2$

RM rasporedljivost



RT test garancije

- ***Izračunavamo*** iskorišćenost procesora kao:

$$U_p = \sum_{i=1}^{n} \frac{C_i}{T_i}$$

- ***Test garancije*** (*samo dovoljan*):

$$U_p \leq n\left(2^{1/n} - 1\right)$$

Osnovne predpostavke

A1. C_i je konstanta za svaku instancu τ_i ;

A2. T_i je konstanta za svaku instancu τ_i ;

A3. Za svaki posao važi: $D_i = T_i$;

A4. Poslovi su nezavisni;

- nema uslovljavanja sa prvenstvom izvršavanja
- nema uslovljavanja resursima

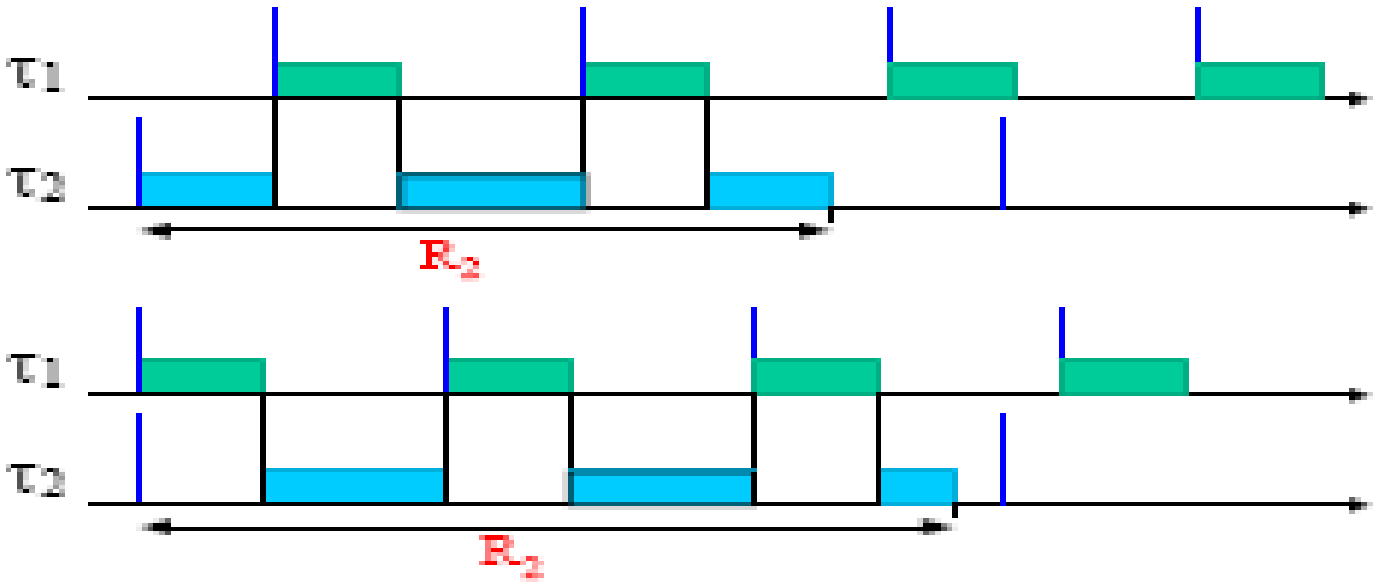
RM Optimalnost

- ***RM je optimalan u smislu izvodljivosti*** u odnosu na sve ***algoritme sa fiksnim (statičkim) prioritetima***:
- ***Ako postoji*** fiksno dodeljivanje prioriteta koje ***dovodi do izvodljivog*** rasporeda za Γ , ***tada je RM*** dodeljivanje ***izvodljivo*** za Γ

- ***Ako*** Γ ***nije rasporedljiv sa RM***, tada se on ne može rasporediti ***ni sa jednim algoritmom*** sa fiksnim prioritetima

Kritična vremenska tačka

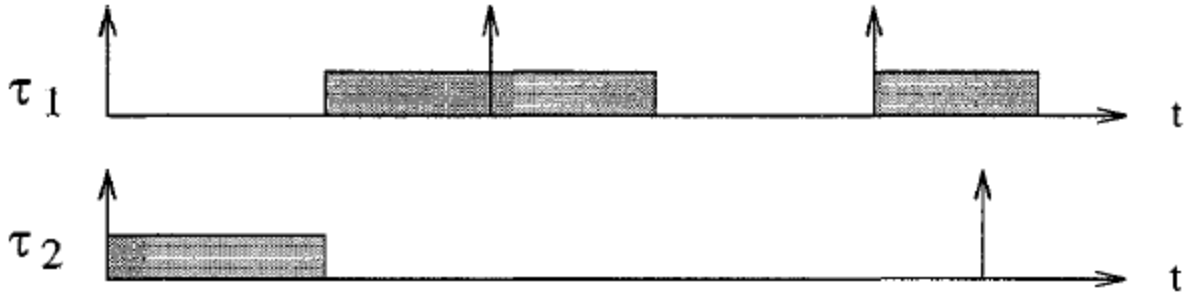
- Za bilo koji posao τ_i, ***najduže vreme odziva*** će se desiti kada se on ***aktivira*** ***<u>u isto vreme</u>*** kada i svi poslovimi koji imaju ***viši prioritet***



<u>Zaključak</u>: - ***Dovoljno je testirati*** rasporedljivost skupa ***u kritičnim tačkama***!

RM Optimalnost

- Posmatramo skup od ***dva periodična posla*** τ_1 i τ_2 sa $T_1 < T_2$
- Ako prioriteti ***nisu dodeljeni saglasno RM*** – tada posao τ_2 ima veći prioritet

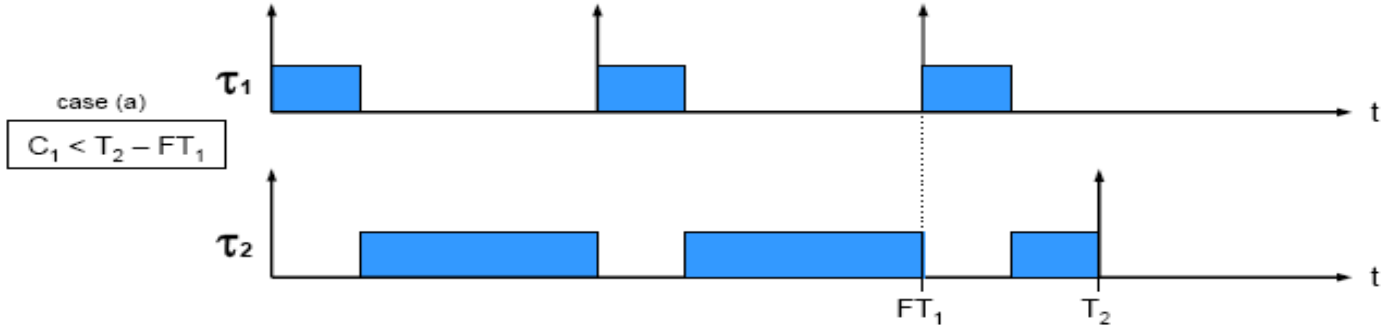


- U kritičnim tačkama – ***raspored je izvodljiv ako je zadovoljeno***:

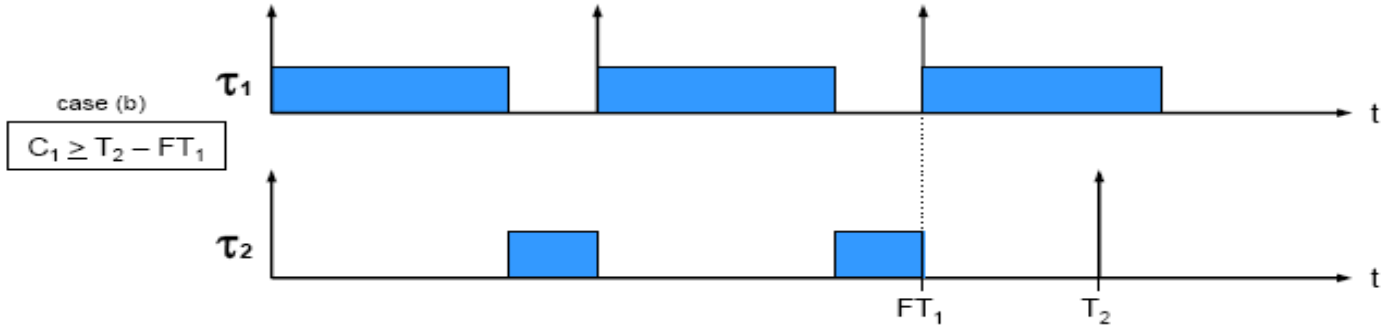
$$C_1 + C_2 \leq T_1$$

RM Optimalnost

Ako su ***prioriteti*** dodeljeni ***saglasno RM***, tada posao τ_1 ima ***viši prioritet***. Možemo razlikovati ***dva slučaja***:



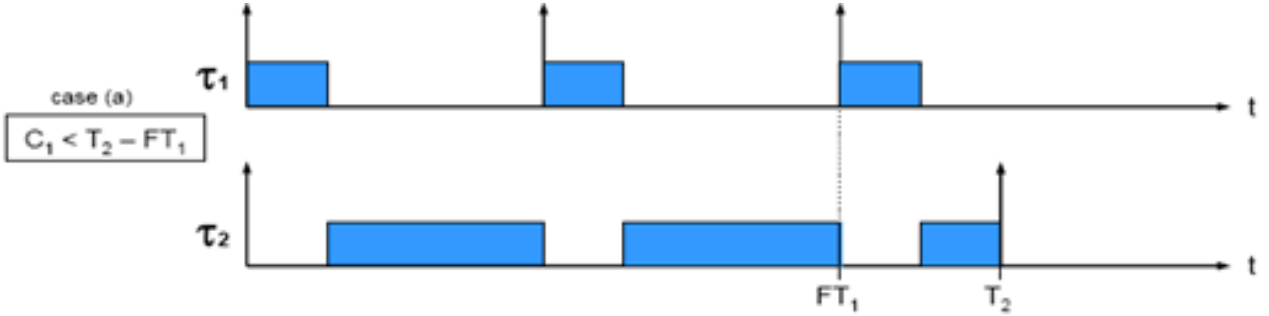
$F = \lfloor T_2/T_1 \rfloor$ **- broj perioda posla τ_1 koje su potpuno sadržane u T_2**



RM Optimalnost

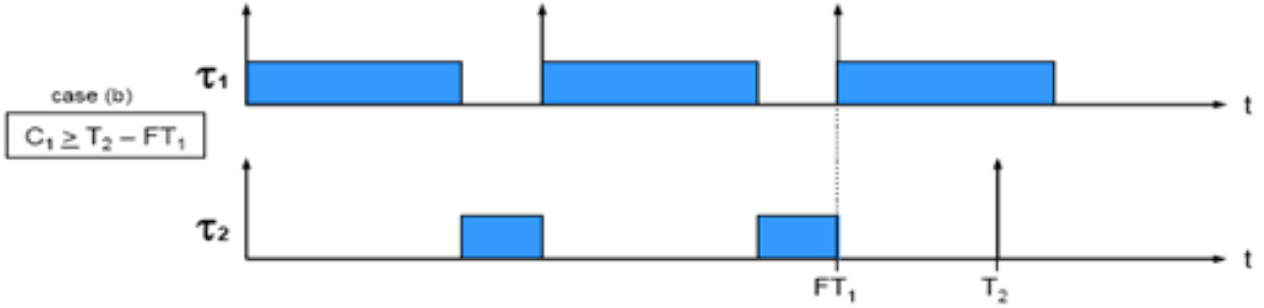
slučaj (a): ***uslov rasporedljivosti*** skupa

$$(F + 1)C_1 + C_2 <= T_2$$



slučaj (b): ***uslov rasporedljivosti*** skupa

$$FC_1 + C_2 <= FT_1$$



RM Optimalnost

slučaj (a): $C_1 < T_2 - FT_1$

uslov rasporedljivosti skupa

$$(F + 1)C_1 + C_2 <= T_2$$

Polazimo od uslova

$$C_1 + C_2 \leq T_1$$

pomnožimo nejednakost sa F

$$FC_1 + FC_2 \leq FT_1$$

kako je $F \geq 1$

$$FC_1 + C_2 \leq FC_1 + FC_2 \leq FT_1$$

na obe strane nejednakosti dodamo C_1

$$(F+1)C_1 + C_2 \leq FT_1 + C_1 \leq T_2$$

RM Optimalnost

slučaj (b): $C_1 \geq T_2 - FT_1$

uslov rasporedljivosti skupa

$FC_1 + C_2 <= FT_1$

Polazimo od uslova

$$C_1 + C_2 \leq T_1$$

pomnožimo obe strane nejednakosti sa F

$$FC_1 + FC_2 \leq FT_1$$

kako je $F \geq 1$

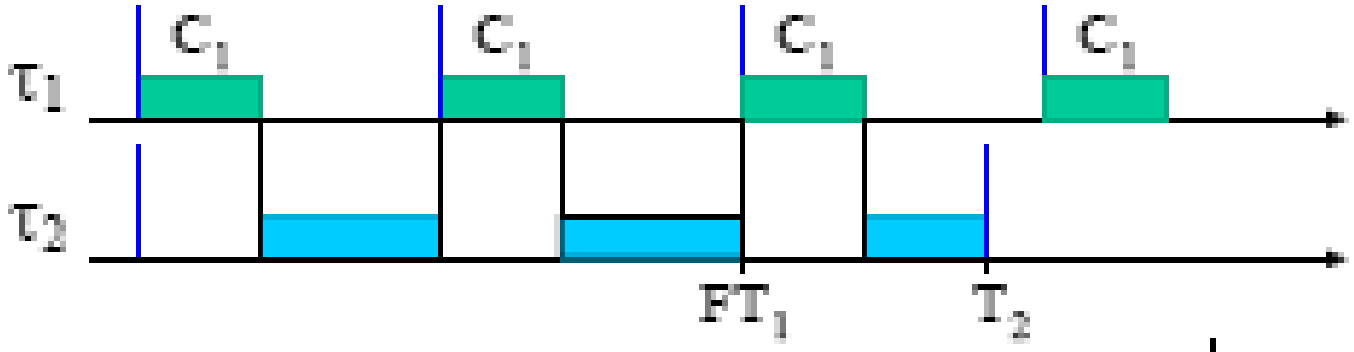
$$FC_1 + C_2 \leq FC_1 + FC_2 \leq FT_1$$

Izračunavanje U_{lub}

- Predpostavimo ***najgori mogući scenario*** za skup poslova (***istovremeno aktiviranje***)
- ***Dodela prioriteta*** saglasno ***RM algoritmu***
- ***Potpuna iskorišćenost procesora***
- Postupak:
  a) Izračunavamo ***gornju granicu U_{ub}***
     - postaviti ***vremena izvršavanja poslova*** tako da se dostigne ***puna iskorišćenost procesora*** (fully utilize)
  b) Minimiziramo U_{ub} u odnosu na sve ostale parametre posla

Izračunavanje U_{lub} za 2 posla

slučaj (a): $C_1 \le T_2 - FT_1$



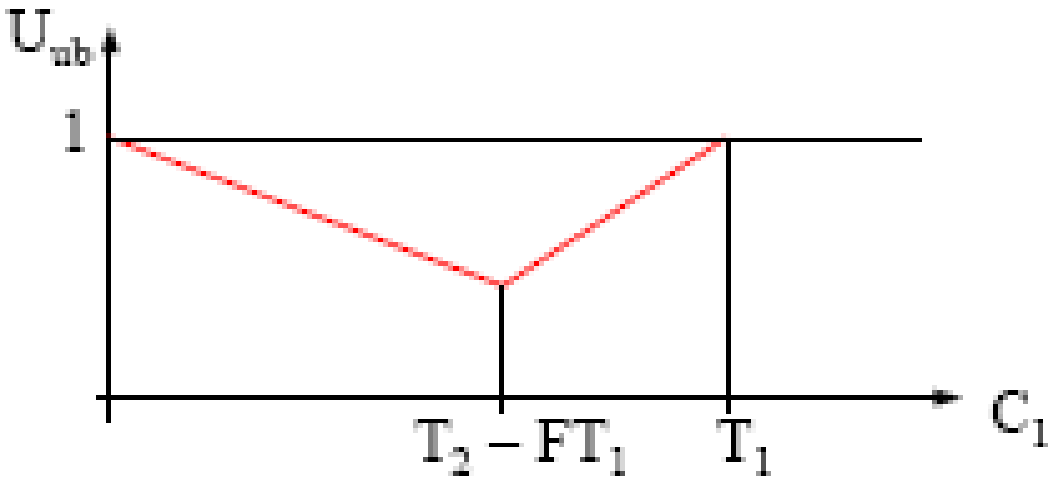
$$C_{2\,max} = T_2 - (F+1)\,C_1 \qquad F = \left\lfloor \frac{T_2}{T_1} \right\rfloor$$

$$U_{ub} = \frac{C_1}{T_1} + \frac{T_2 - (F+1)C_1}{T_2} = 1 + \frac{C_1}{T_2}\left[\frac{T_2}{T_1} - (F+1)\right]$$

Izračunavanje U_{lub} za 2 posla

slučaj (a): $C_1 \le T_2 - FT_1$

$$U_{ub} = 1 + \frac{C_1}{T_2}\left[\frac{T_2}{T_1} - (F+1)\right]$$



(A) $C_1 = T_2 - FT_1$ minimizira U_{ub}

Izračunavanje U_{lub} za 2 posla

slučaj (a): $C_1 \le T_2 - FT_1$

$$U_{lub} = U_{ub}\Big|_{C_1 = T_2 - FT_1} = \frac{T_1}{T_2}\left[F + \left(\frac{T_2}{T_1} - F\right)^2\right]$$

Funkcija raste sa ***F***, zato ćemo postaviti da je ***F = 1***

Minimiziranje u odnosu na $k = T_2/T_1$, daje:

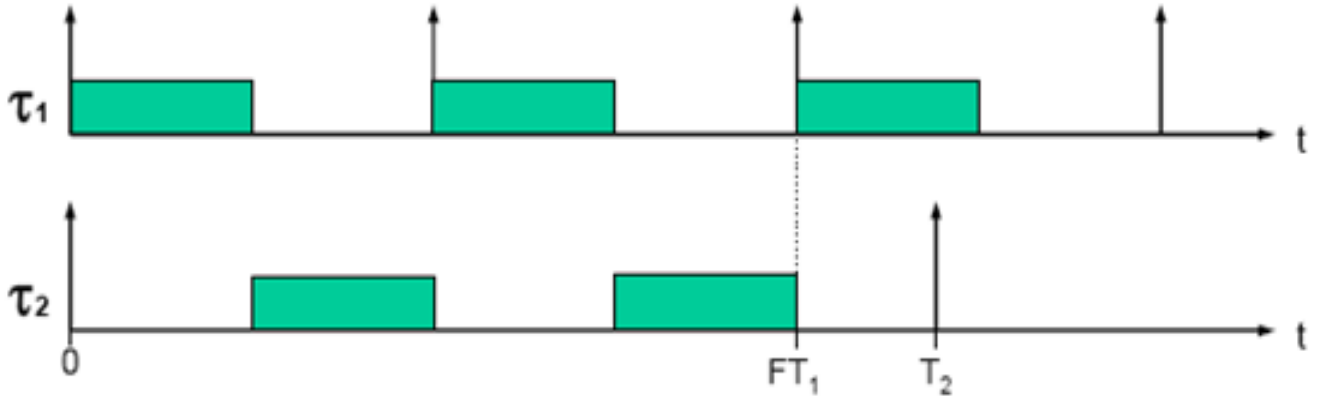
$$U_{lub} = \frac{1 + (k-1)^2}{k} \qquad \frac{dU_{lub}}{dk} = \frac{k^2 - 2}{k^2}$$

$dU/dk = 0$ for $k = \sqrt{2}$

$$U_{lub} = 2\left(\sqrt{2} - 1\right) \cong 0.83$$

Izračunavanje U_{lub} za 2 posla

slučaj (b): $C_1 \geq T_2 - FT_1$



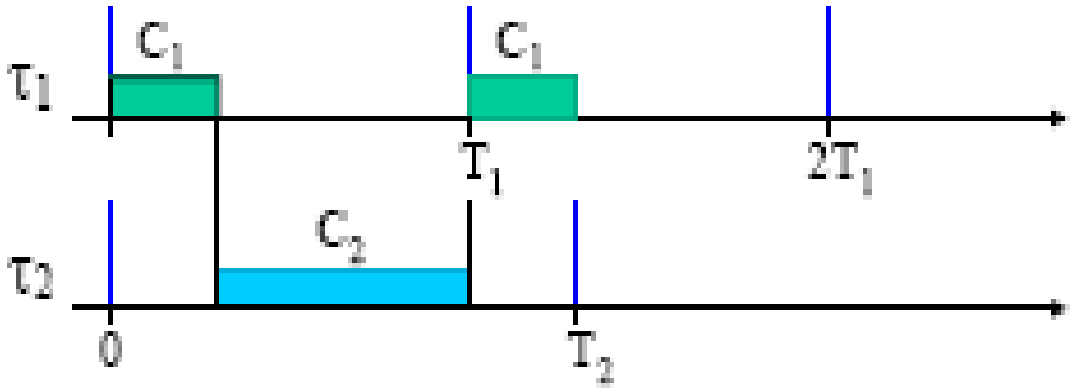
Najveća moguća vrednost za $C_{2\,max}$:

$$C_2 = (T_1 - C_1)\ F$$

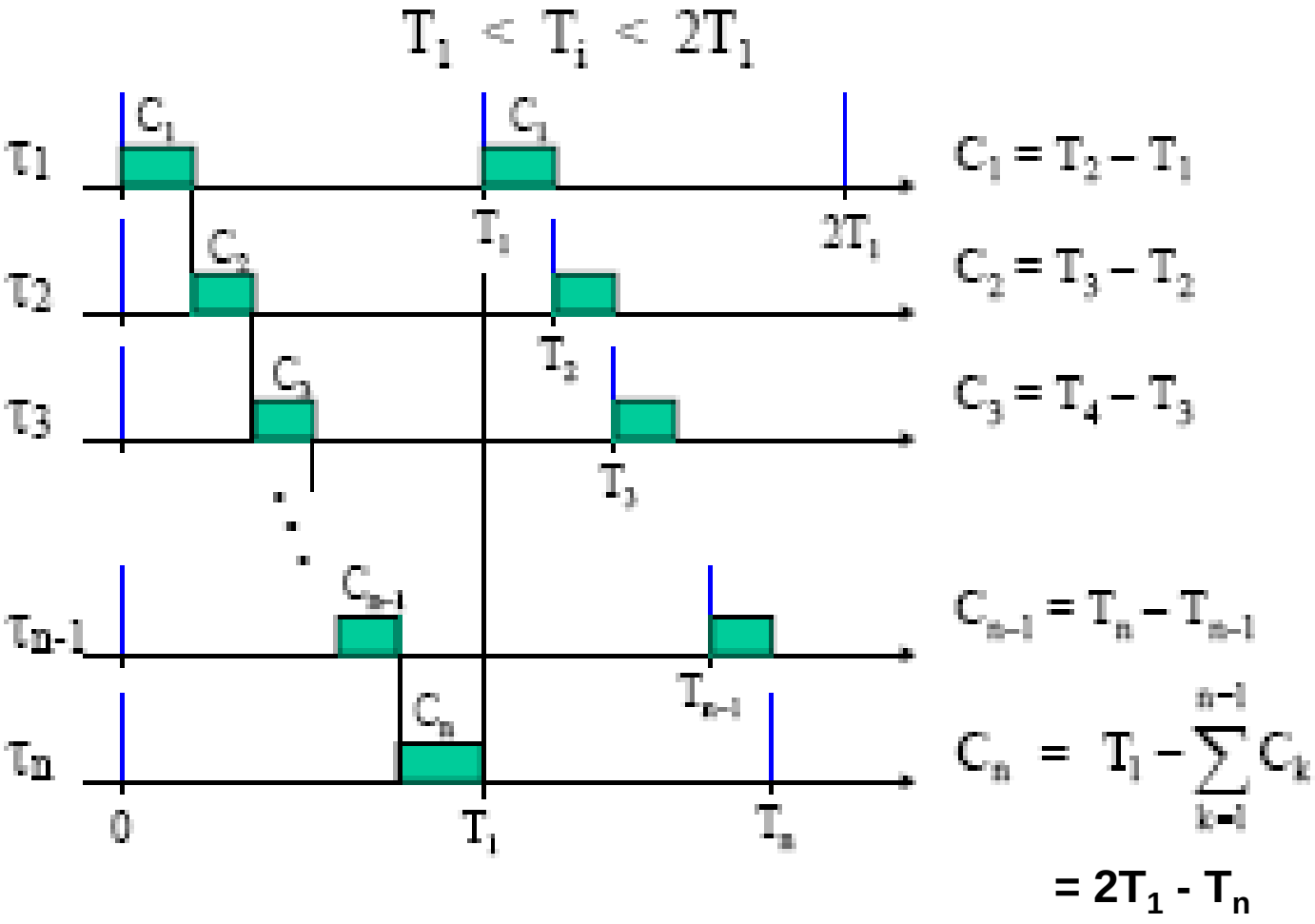
Sledimo isti postupak kao u slučaju (a).....

Najgori slučaj za 2 posla

- $F = 1$ ⇒ $T_1 < T_2 < 2T_1$
- $C_1 = T_2 - FT_1$ ⇒ $C_1 = T_2 - T_1$



Najgori slučaj za n poslova



Izračunavanje U_{lub} za n poslova

$$U_{ub} = \frac{T_2 - T_1}{T_1} + \frac{T_3 - T_2}{T_2} + \cdots + \frac{T_n - T_{n-1}}{T_{n-1}} + \frac{2T_1 - T_n}{T_n}$$

$$U_{ub} = \frac{T_2}{T_1} + \frac{T_3}{T_2} + \cdots + \frac{T_n}{T_{n-1}} + \frac{2T_1}{T_n} - n$$

Ako definišemo $R_i = T_{i+1}/T_i$ i kako je: $P = \prod_{i=1}^{n-1} R_i = \frac{T_n}{T_1}$

Možemo pisati:

$$U_{ub} = \sum_{i=1}^{n-1} R_i + \frac{2}{P} - n$$

Izračunavanje U_{lub} za n poslova

$$U_{ub} = \sum_{i=1}^{n-1} R_i + \frac{2}{P} - n$$

Minimiziranjem u odnosu na R_i, imamo:

$$\frac{\partial U_{ub}}{\partial R_i} = 1 - \frac{2}{R_i P} = 0 \qquad za \qquad R_i = 2^{1/n}$$

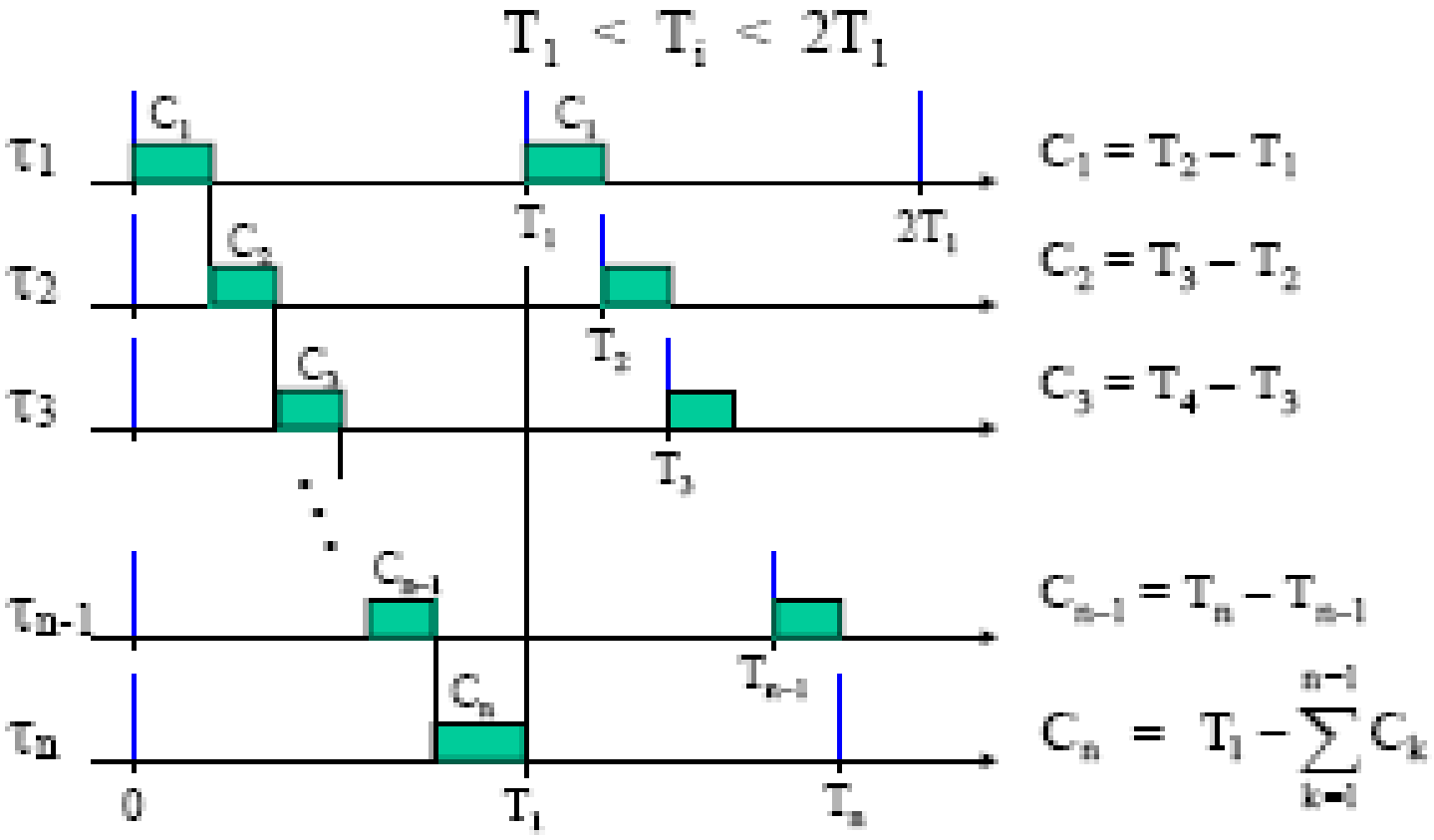
$$U_{\text{lub}} = n\left(2^{1/n} - 1\right)$$

Hiperbolička granica

- 2000, Bini & al. su dokazali da je skup od *n* periodičnih poslova rasporedljiv sa RM ako:

$$\prod_{i=1}^{n} (U_i + 1) \leq 2$$

Skica dokaza



Skica dokaza

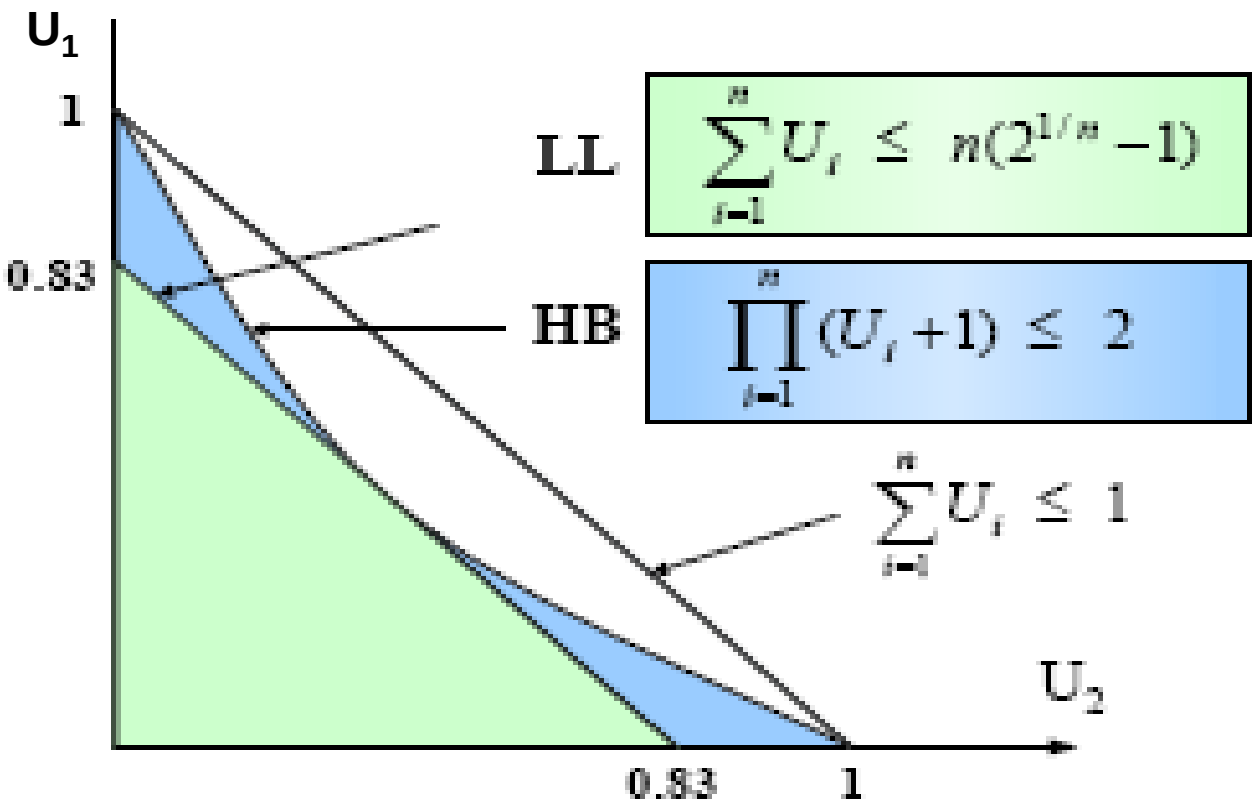
$$C_i = T_{i+1} - T_i \quad \Rightarrow \quad U_i = R_i - 1 \quad \Rightarrow \quad R_i = U_i + 1$$

$$C_n \le T_1 - \sum_{k=1}^{n-1} C_k \quad \Rightarrow \quad C_n \le 2T_1 - T_n$$

$$U_n \le \frac{2}{T_n/T_1} - 1 \quad \Rightarrow \quad \frac{T_n}{T_1}(U_n + 1) \le 2$$

$$\frac{T_n}{T_1} = \prod_{i=1}^{n-1} R_i = \prod_{i=1}^{n-1} (U_i + 1) \quad \Rightarrow \quad \boxed{\prod_{i=1}^{n} (U_i + 1) \le 2}$$

HB vs. LL


U1
1
0.83
LL
$\sum_{i=1}^{n} U_i \leq n(2^{1/n} - 1)$
HB
$\prod_{i=1}^{n} (U_i + 1) \leq 2$
$\sum_{i=1}^{n} U_i \leq 1$
U2
0.83
1

Algoritam najranijeg deadline-a

- **E**arliest **D**eadline **F**irst, **EDF**
- Svakoj instanci (job) se ***dodeljuje apsolutni deadline***

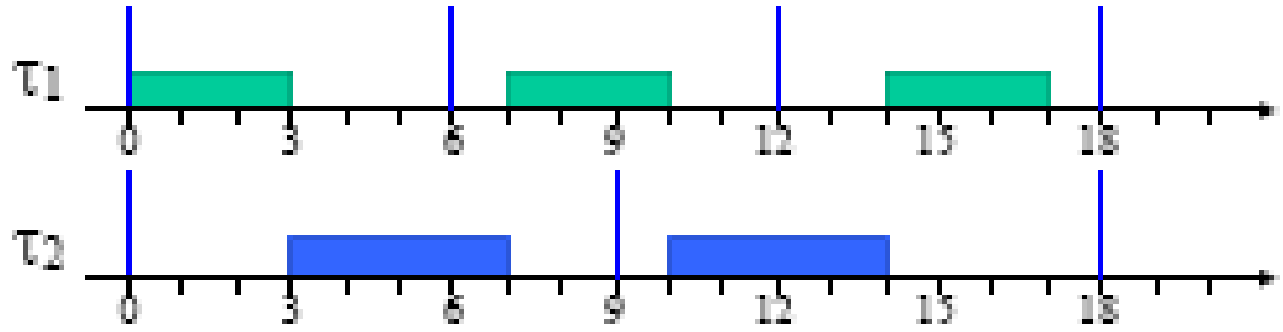
$$d_{i,k} = r_{i,k} + D_i$$

- ***U bilo kom trenutku*** procesor se dodeljuje instanci sa ***najranijim deadline***-om
- Sa EDF-om, proizvoljni skup poslova može koristiti procesor i do ***100%***

EDF - Primer

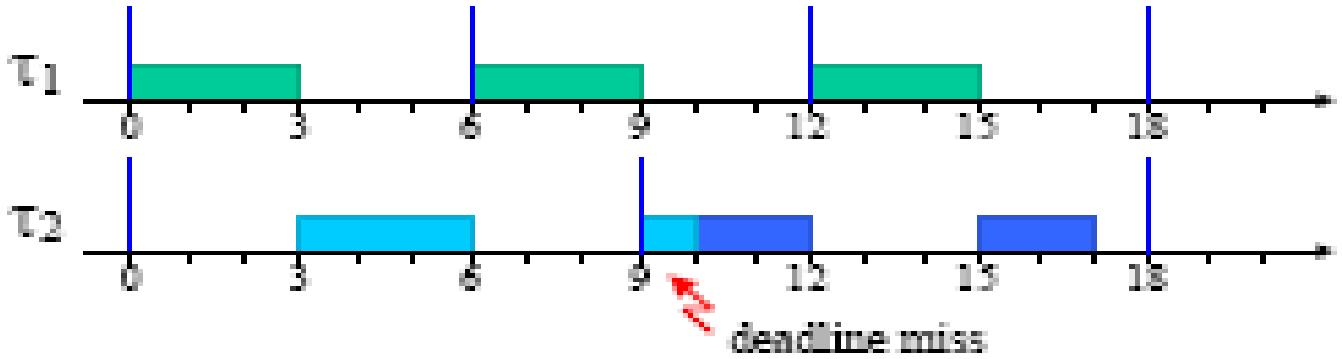
$D_i = T_i$

$$U_p = \frac{3}{6} + \frac{4}{9} = 0.94$$



RM – neizvodljiv raspored

$$U_p = \frac{3}{6} + \frac{4}{9} = 0.944$$



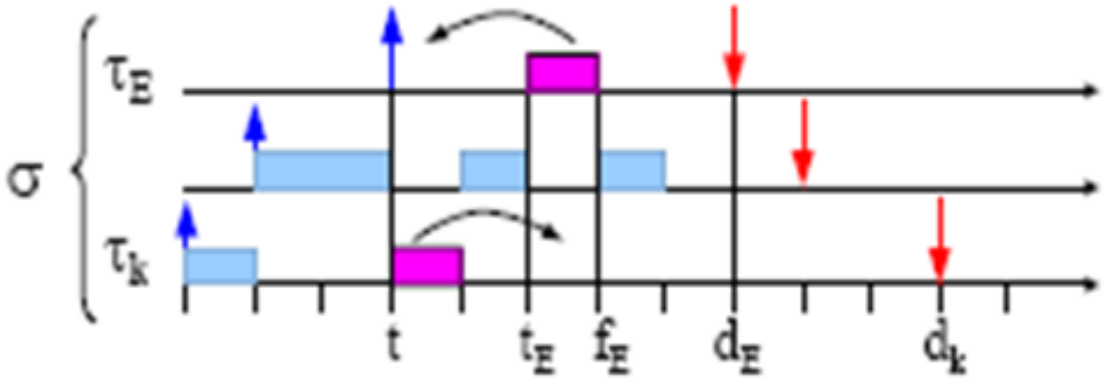
EDF - Optimalnost

- Među svim algoritmima, ***EDF je optimalan***:

- Ako ***postoji izvodljiv*** raspored za Γ, tada će ***EDF garantovati izvodljiv*** raspored

- Ako Γ ***nije rasporedljiv*** sa EDF, tada on ne može biti raspoređen ***<u>ni sa jednim</u>*** algoritmom

EDF - Optimalnost



Transformacija σ u σ'

$$\begin{cases} \sigma'(t) = \sigma(t_E) \\ \sigma'(t_E) = \sigma(t) \end{cases}$$

Izvodljivost je očuvana

$$f_k' = f_E \leq d_E \leq d_k$$

EDF - Rasporedljivost

- 1973, Lui & Layland su dokazali da ***za skup od n periodičnih poslova važi***:

$$U_{\text{lub}}^{EDF} = 1$$

- To znači da je skup Γ rasporedljiv sa algoritmom EDF ***ako i samo ako***

$$U_p \leq 1$$