Praktikum iz operativnih sistema

Lekcija 6: Raspoređivanje periodičnih poslova (II)

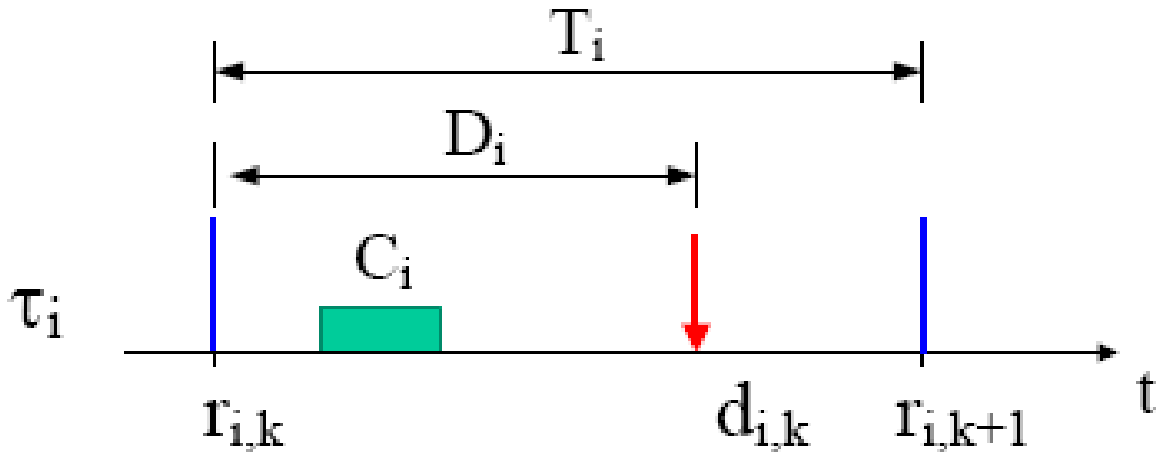
zima 2019/2020

Prof. dr Branimir Trenkić

Proširenje na poslove sa D < T

- Dosadašnja analiza pasporedljivosti skupa periodičnih poslova – hipoteze ***A1***, ***A2***, ***A3*** i ***A4***
- ***Hipoteza A3*** - Sve instance periodičnog posla τ_i imaju ***isti*** relativni deadline $\mathbf{D_i}$, koji je jednak periodi $\mathbf{T_i}$
- Ova hipoteza dozvoljava izvršavanje instance bilo gde unutar periode – to u realnim aplikacijama vrlo ***često nije slučaj***
- ***Relaksiranje uslova A3*** će omogućiti ***fleksibilniji model*** procesa

Proširenje na poslove sa D < T



- ***Algoritmi raspoređivanja***:
- **DM** (***Deadline Monotonic***): $p_i \leftrightarrow 1/D_i$ (***statički***)
- **EDF** (***Earliest Deadline First***): $p_i \leftrightarrow 1/d_i$ (***dinamički***)

DM (*Deadline Monotonic*)

- Proširenje RM- algoritma u slučajevima kada poslovi ***mogu imati relativne deadline-ove manje od njihovih perioda***
- DM način dodele prioriteta ***relaksira*** uslov $D_i = T_i$ unutar šeme raspoređivanja sa statičkim prioritetima
- Svaki periodični posao τ_i je dat sa:
  - **fazom** Φ_i
  - ***vremenom izvršavanja*** u najgorem slučaju C_i
  - ***relativnim deadline***-om, D_i
  - ***periodom*** T_i

DM - Analiza rasporedljivosti

- **Verifikacija izvodljivosti** skupa poslova sa deadline-ovima koji nisu jednaki periodama, može biti izvršena ***RM testom rasporedljivosti*** skupa
- ***Modifikacija testa*** – redukovanje perioda posla sa relativnim deadline-ovima:

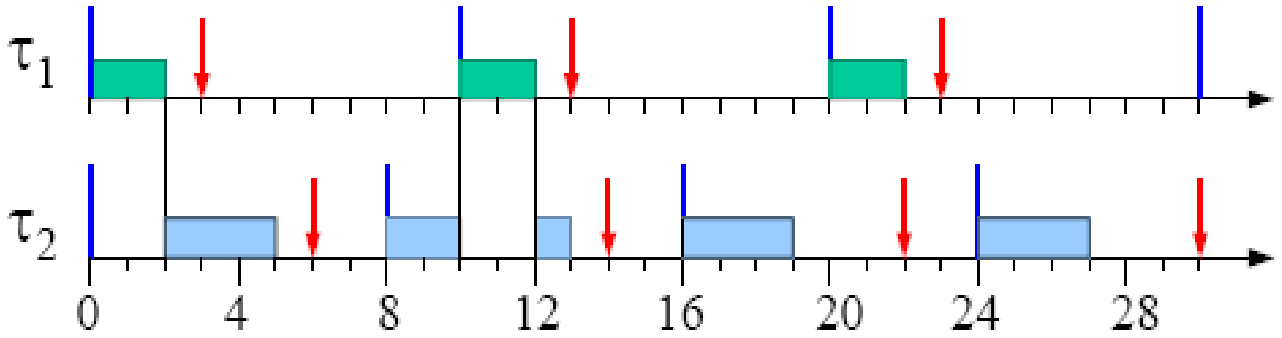
$$U_p = \sum_{i=1}^{n} \frac{C_i}{D_i} \leq n\left(2^{\frac{1}{n}} - 1\right)$$

- ***Ovaj test nije optimalan*** – ***vrlo je pesimističan*** (***precenjuje se opterećenje procesora!)***

DM - Analiza rasporedljivosti

Primer:

	C_i	T_i	D_i
τ_1	2	10	3
τ_2	3	8	6



- Problem sa **ograničenjem** faktora iskorišćenosti procesora:

$$U_p = \sum_{i=1}^{n} \frac{C_i}{D_i} = \frac{2}{3} + \frac{3}{6} = 1.16 > 1$$

ali, skup poslova je još uvek **rasporedljiv**!

DM - Analiza rasporedljivosti

- Manje pesimistički **test rasporedljivosti** se može dobiti na sledeći način:
- Ukupno zahtevano procesorsko vreme ***u najgorem slučaju***, desiće se kada se ***svi poslovi aktiviraju simultano*** (**kritična tačka**)

Test:

Za svaki posao τ_i , **zbir**

(i) njegovog ***vremena izvršavanja*** i

(ii) ***interferencije*** (prekidanja) nametnute poslovima većeg prioriteta,

mora biti manji ili jednak od D_i

DM - Analiza rasporedljivosti

- Predpostavimo da su ***poslovi uređeni po rastućim relativnim deadline-ovima***:

$$\mathbf{i < j \leftrightarrow D_i < D_j}$$

test je dat sa:

$$\forall i : 1 \le i \le n \quad C_i + I_i \le D_i$$

gde je I_i ***mera interferencije*** na posao τ_i :

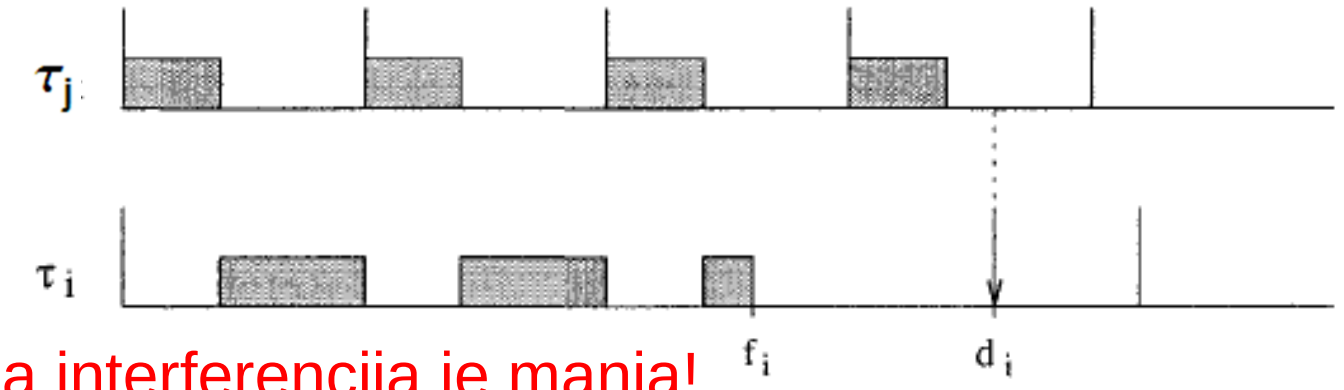
$$I_i = \sum_{j=1}^{i-1} \left\lceil \frac{D_i}{T_j} \right\rceil C_j$$

DM - Analiza rasporedljivosti

- ***Ovaj test <u>je dovoljan</u> mada <u>nije potreban</u>*** za garanciju rasporedljivosti skupa poslova
- Razlog:

I_i se računa sa predpostavkom da posao višeg prioriteta τ_j utiče tačno $\lceil D_i/T_j \rceil$ puta na posao τ_i

- Što ne mora biti tačno:



Aktuelna interferencija je manja!

Potreban i dovoljan uslov testa

- Analiza bazirana na ***vremenu odziva*** (***response time analysis***)
- Za svaki posao τ_i, izračunava se ***interferencija*** izazvana poslovima višeg prioriteta:

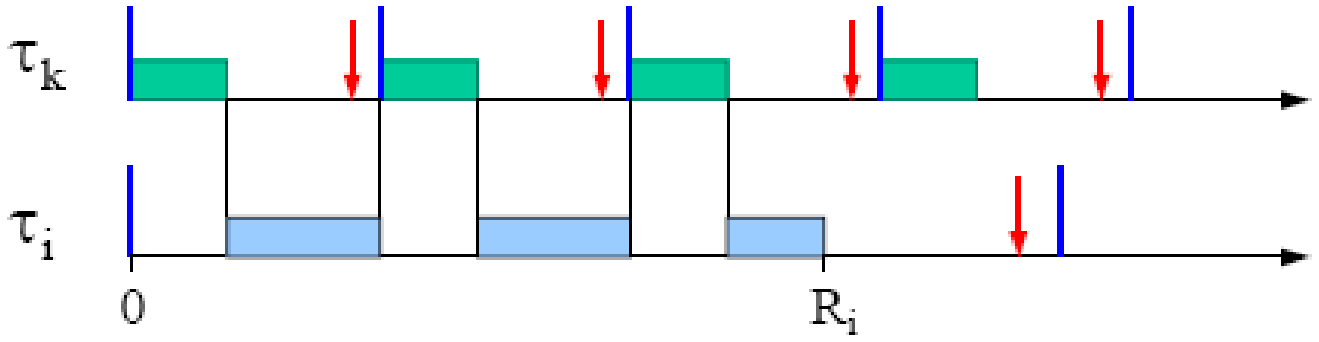
$$I_i = \sum_{D_k < D_i} C_k$$

- Izračunava se zatim ***vreme odziva*** kao

$$R_i = C_i + I_i$$

- Zatim ***verifikacija*** ako je $R_i \leq D_i$

Proračun interferencije



- Interferencija ***posla*** τ_k na izvršenje posla τ_i u intervalu ***[0, R_i]***:

$$I_{ik} = \left\lceil \frac{R_i}{T_k} \right\rceil C_k$$

  - Interferencija ***svih posla*** sa višim prioritetom na posao τ_i :

$$I_i = \sum_{k=1}^{i-1} \left\lceil \frac{R_i}{T_k} \right\rceil C_k$$

Proračun vremena odziva

$$R_i = C_i + \sum_{k=1}^{i-1} \left\lceil \frac{R_i}{T_k} \right\rceil C_k$$

- ***Iterativno rešenje*** proračuna:

$$R_i^{(0)} = \sum_{k=1}^{i} C_k$$

$$R_i^{(s)} = C_i + \sum_{k=1}^{i-1} \left\lceil \frac{R_i^{(s-1)}}{T_k} \right\rceil C_k$$

sve dok je $R_i^{(s)} > R_i^{(s-1)}$

Proračun vremena odziva

- Kada ***uslov iteracija*** prestane da važi, dobija se ***konačna vrednost*** za R_i:

$$R_i = R_i^{(s)} = R_i^{(s-1)}$$

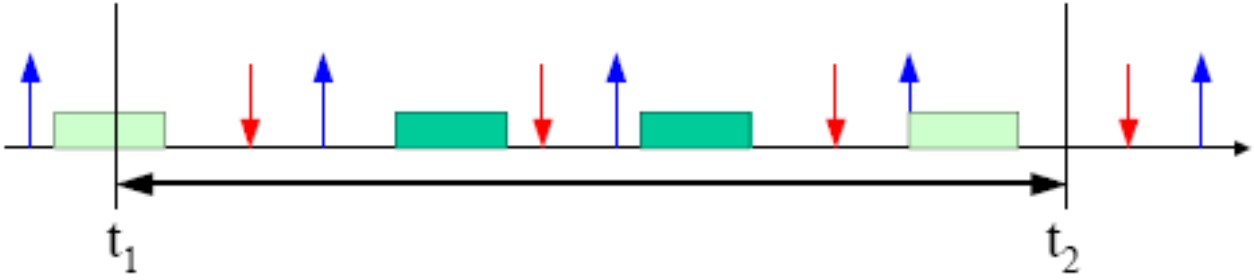
- Ostalo je samo još proveriti uslov rasporedljivosti - $R_i \leq D_i$

Dinamičko dodeljivanje prioriteta

- **EDF**:
- Raspoređivanje bazirano na ***apsolutnim deadline-ovima***
- ***Analiza rasporedljivosti***:
- Bazira se na kriterijumu **zahtevanog procesorskog vremena** (***processor demand***):

U bilo kom intervalu, ***zbir zahteva za procesorskim vremenom*** svih poslova iz skupa, ***ne sme biti veći*** od respoloživog vremena

Proračun potrebnog procesorskog vremena

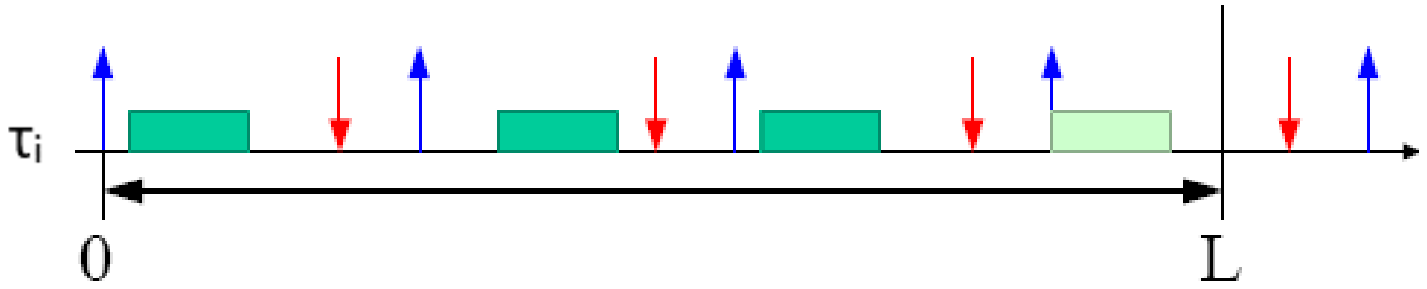


- Potrebno da se obrade **sve instance** posla τ_i za **koje važi**:
  - *release time* (aktiviranje)– ***posle ili jednak*** t_1
  - *deadline* – ***pre ili jednek*** t_2

Uslov rasporedljivosti!

$$g(t_1, t_2) = \sum_{\substack{r_i \geq t_1 \\ d_i \leq t_2}} C_i \leq (t_2 - t_1)$$

Proračun potrebnog procesorskog vremena



- Potrebno ***procesorsko vreme*** na intervalu [0,L] za posao τ_i :

$$g_i(0,L) = \left(\left\lfloor \frac{L - D_i}{T_i} \right\rfloor + 1 \right) C_i$$

- Za ceo ***skup od n*** periodičnih ***poslova***:

$$g(0,L) = \sum_{i=1}^{n} \left\lfloor \frac{L - D_i + T_i}{T_i} \right\rfloor C_i$$

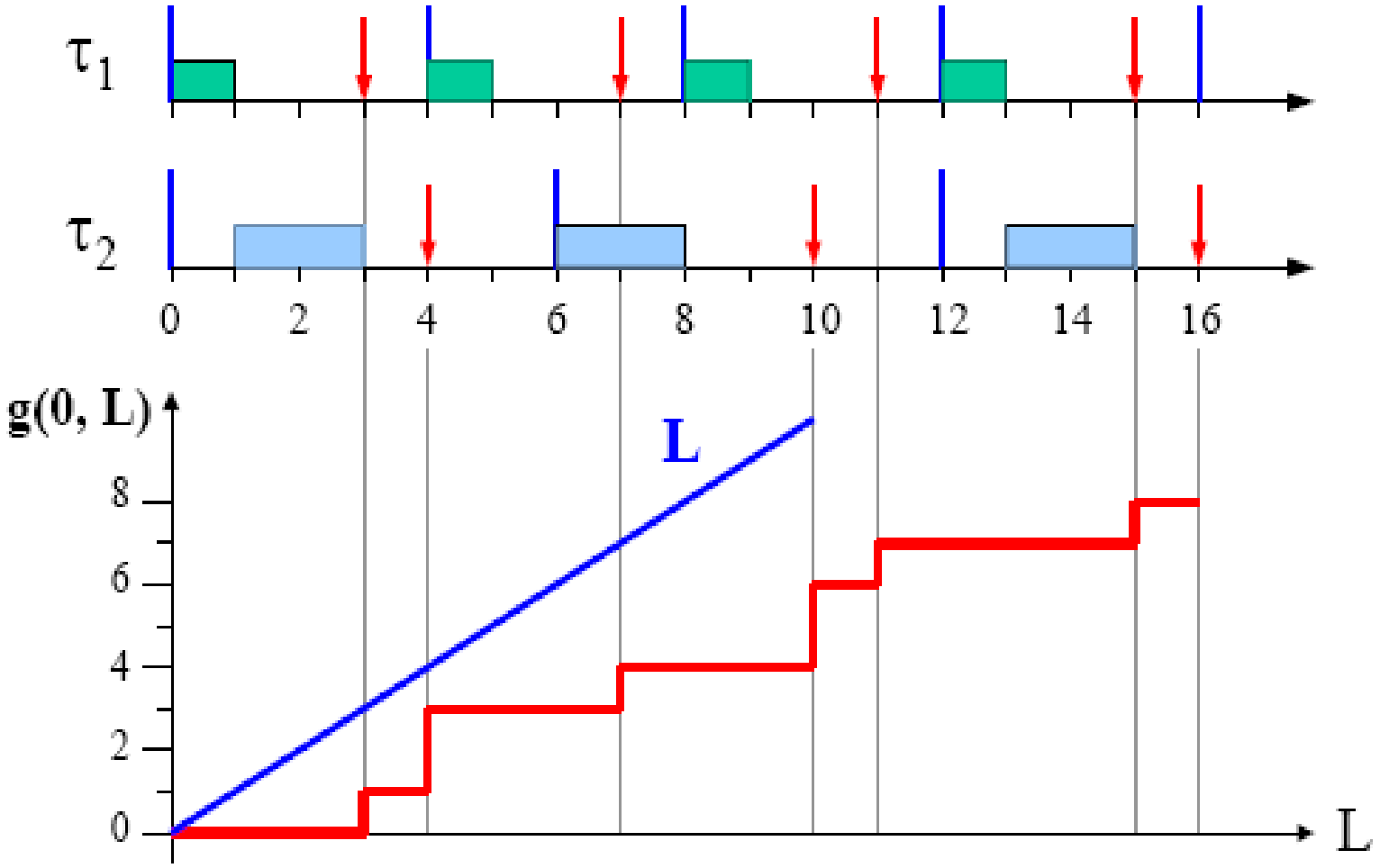
Testiranje zahteva

$$\forall L > 0, \quad g(0, L) \leq L$$

- ***Pitanje***:

Kako ograničiti broj intervala u kojima ćemo Izvršiti testiranje?

Primer



Ograničenje složenosti proračuna

- Kako je **g(0,L)** **step- funkcija**

provera izvodljivosti se može vršiti ***samo u tačkama deadline-a*** (gde ova funkcija menja vrednost)

- Ako su poslovi **sinhroni** i **$U_p < 1$**

provera izvodljivosti se vrši najviše u ***hiper-periodu H*** (***najmanji zajednički sadržalac***):

$$H = NZS(T_1, \ldots, T_n)$$

Ograničenje složenosti proračuna

- Pored toga, važi: $g(0,L) \leq G(0,L)$

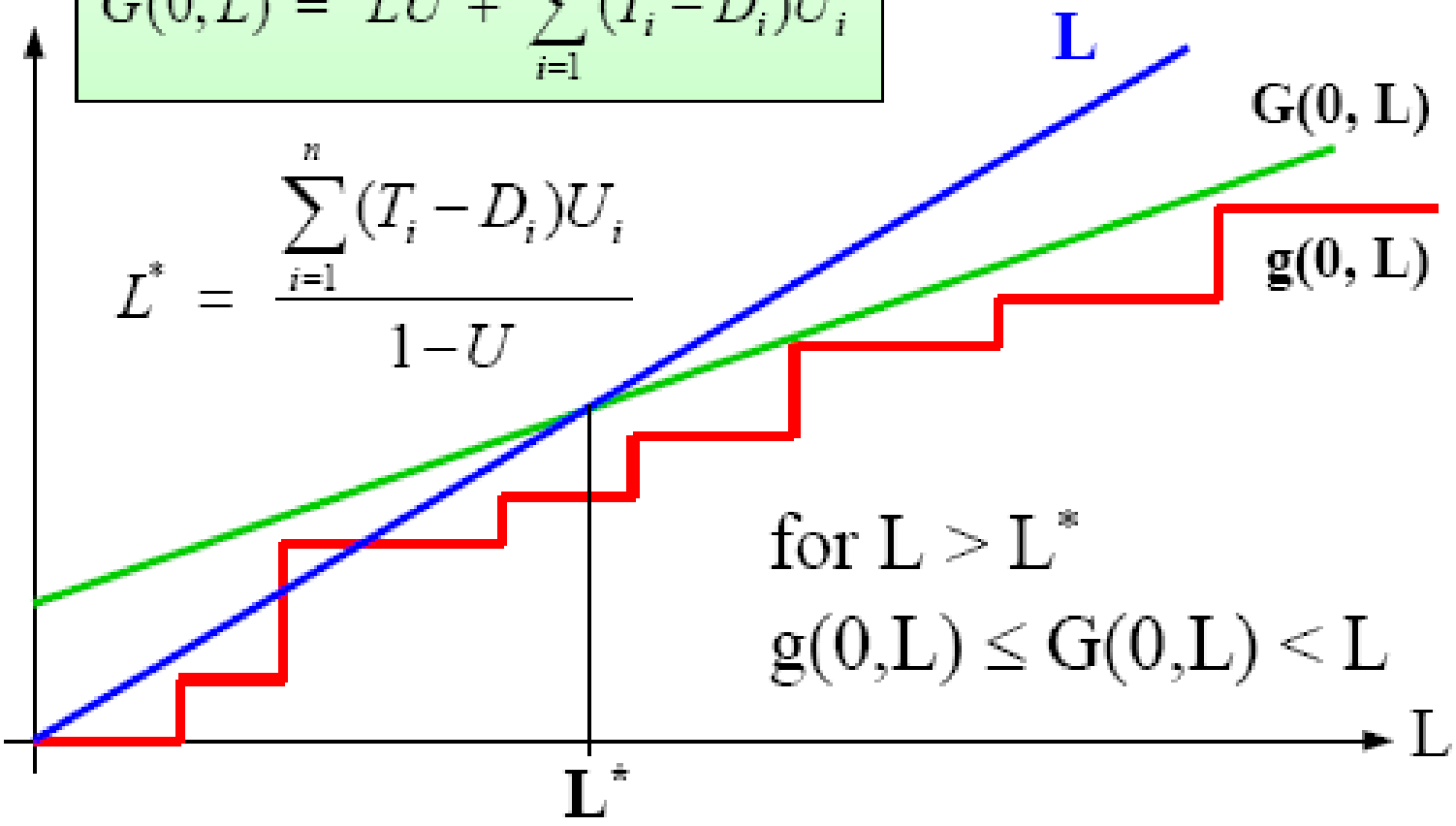
$$G(0,L) = \sum_{i=1}^{n} \left(\frac{L + T_i - D_i}{T_i} \right) C_i$$

$$= \sum_{i=1}^{n} L \frac{C_i}{T_i} + \sum_{i=1}^{n} (T_i - D_i) \frac{C_i}{T_i}$$

$$= LU + \sum_{i=1}^{n} (T_i - D_i) U_i$$

Ograničenje složenosti proračuna

$$G(0,L) = LU + \sum_{i=1}^{n}(T_i - D_i)U_i$$



Ograničenje složenosti proračuna

$$\forall L \in D, \quad g(0, L) \leq L$$

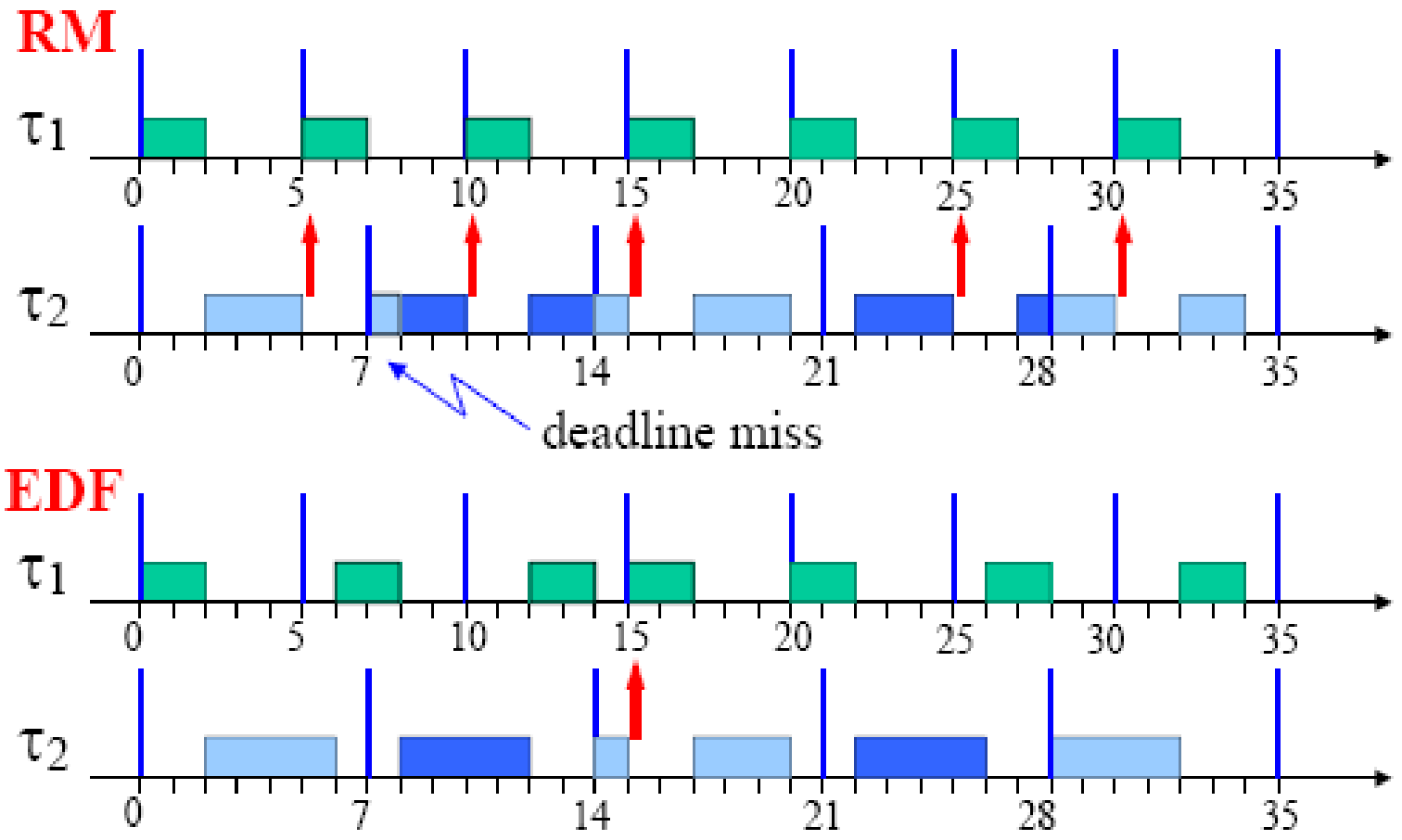
$$D = \{ \mathrm{d_k} \mid \mathrm{d_k} \leq \min(\mathrm{H}, \mathrm{L}^*) \}$$

$$\begin{cases} H = lcm(T_1, \ldots, T_n) \\ L^* = \dfrac{\sum_{i=1}^{n} (T_i - D_i) U_i}{1 - U} \end{cases}$$

Zaključak

- ***Tri pristipa raspoređivanja***:
- ***Off-line*** konstrukcija (Ciklično raspoređivanje)
- ***Statička*** dodela prioriteta (RM, DM)
- ***Dinamička*** dodela prioriteta (EDF)
- ***Tri analitičke tehnike***:
- Ograničavanje iskorišćenosti procesora $U \leq U_{lub}$
- Analiza vremena odziva $\forall i \; R_i \leq D_i$
- Kriterijum zahteva za procesorskim vremenom $\forall L \; g(0,L) \leq L$

Prebacivanje konteksta



RM vs. EDF : Zaključak

- ***EDF***:
- Mnogo ***efikasniji***
- ***Redukuje*** prebacivanje konteksta
- ***RM***:
- ***Jednostavan za primenu*** u komercijalnim sistemima realnog vremena