4. Raspoređivanje poslova

Prvi dio – uvod + raspoređivanje na jednoprocesorskim sustavima

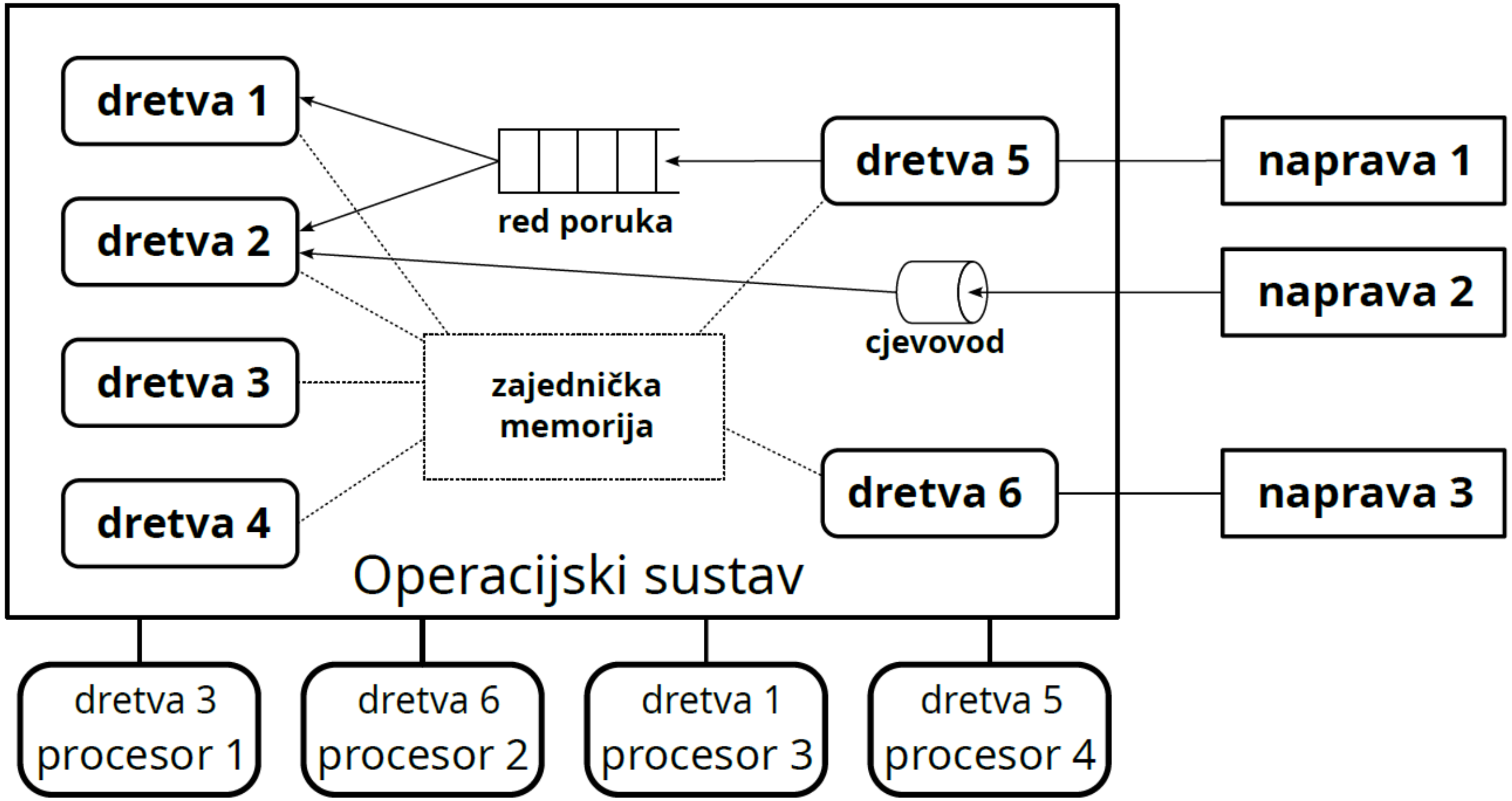
Pojmovi: posao, zadatak, dretva

- Iako interpretacija nije uvijek identična, uglavnom će se koristiti slijedeća:
  - **posao** (eng. *job*) – nešto što treba napraviti;
    - npr. upravljati vozilom da dođe od A do B
    - složeni posao se rastavlja na manje dijelove = zadatke
  - **zadatak** (eng. *task*) – dio posla, (relativno) nezavisan od drugih zadataka
    - jedan posao obično sadržava više zadataka, ali može i samo jedan
- U idućim razmatranjima će se koristiti pojmovi:
  - zadatak
  - **skup/sustav** zadataka
  - u izvođenju se zadatak obavlja zasebnom **dretvom** (unutar OS-a);
    ipak, u 4. poglavlju će se koristiti samo pojam *zadatka* – od 5. *dretva*
  - **raspoređivanje zadataka** (eng. *task scheduling*)

Zašto više zadataka

- … umjesto jednog koji upravlja svima?
- Računalo koje upravlja složenijim sustavom ima više (nezavisnih) elemenata
  - objedinjavanje upravljanja bilo bi presloženo (u kodu) ili čak neizvedivo uz zadovoljavanje svih ograničenja
  - podjela upravljanja pojedinim elementima u zasebne dijelove – zadatke smanjuje složenost, svaki je zadatak pojedinačno znatno jednostavniji
    - ali zahtijeva OS da upravlja tim zadacima:
      - daje im procesor u pravim trenucima (raspoređuje ih)
      - kontrolira pristup zajedničkim sredstvima i UI napravama
      - daje mehanizme za sinkronizaciju i komunikaciju
- Drugi razlog jest u bržem radu na višeprocesorskom sustavu

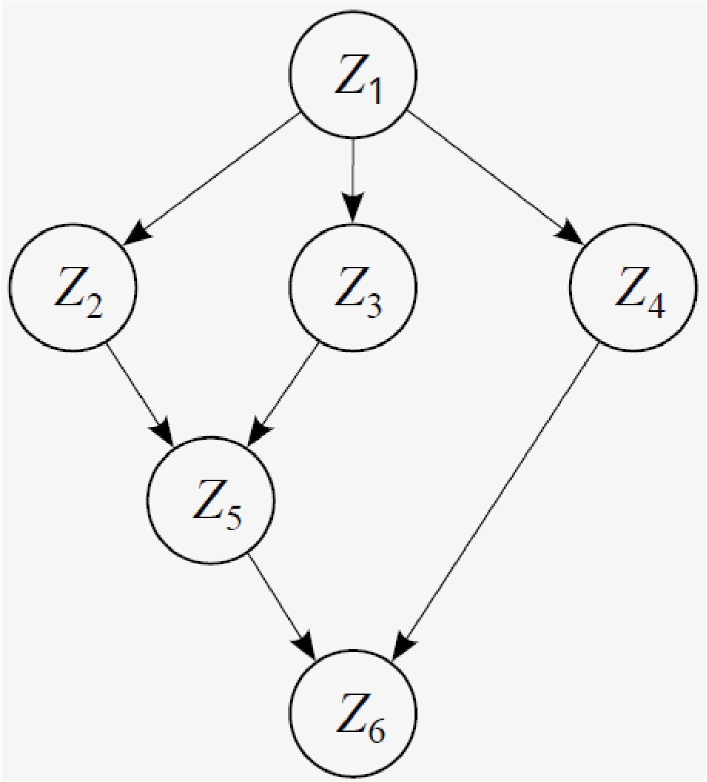
OS

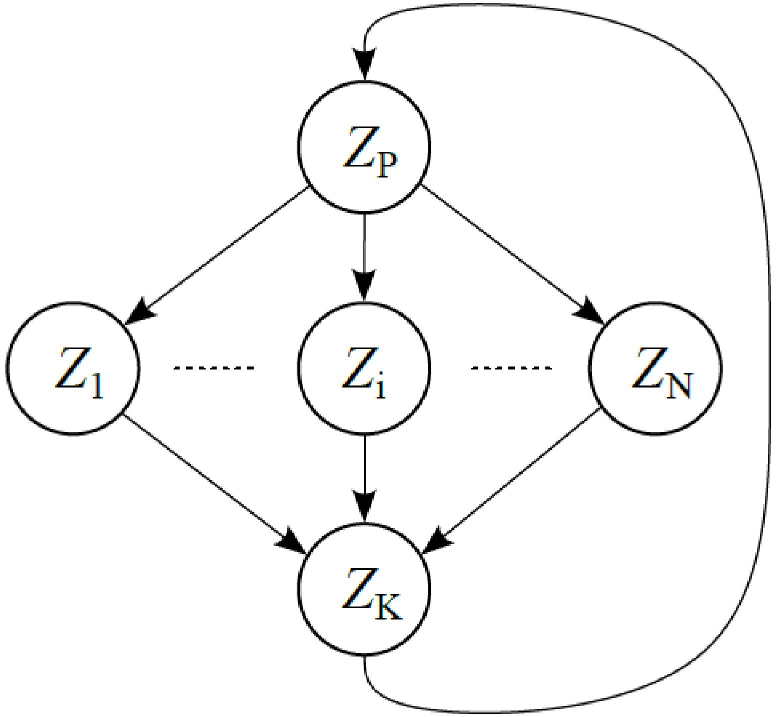


Slika 4.1. Primjer višedretvenog upravljačkog sustava

Podjela posla u zadatke – kako?

- ovisi o poslu
- potreba za
  - sinkronizacijom
  - komunikacijom

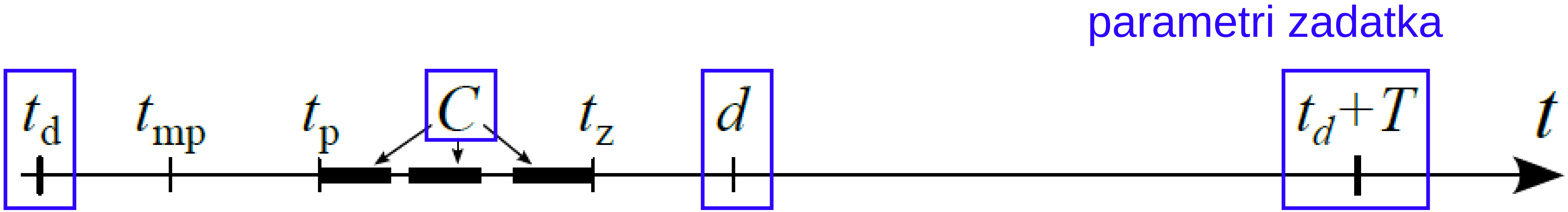




- u većem dijelu 4. poglavlja se međuovisnost neće razmatrati, razmatraju se samo nezavisni zadaci

4.3. Vremenska svojstva zadataka

- razmatraju se periodički zadaci

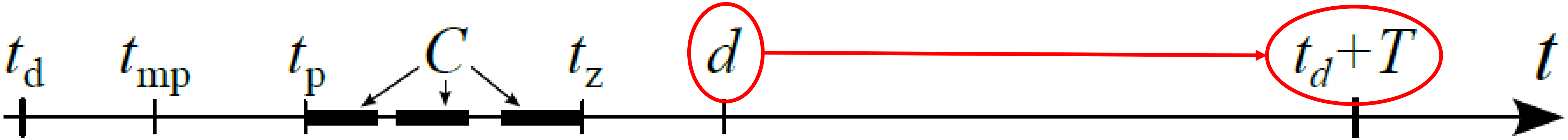


Legenda:

t_d	–	trenutak dolaska	C	–	trajanje obrade
t_mp	–	trenutak mogućeg početka	d	–	rok završetka
t_p	–	trenutak početka	T	–	period ponavljanja
t_z	–	trenutak završetka			

Slika 4.4. Svojstveni trenutci u životnom ciklusu jednog zadatka

4.3. Vremenska svojstva zadataka



- **osnovni problem: osigurati $t_z <= d$ za svaki zadatak!**

- u nastavku se analiziraju sustavi kod kojih vrijedi: $d = t_z + T$

4.4. Postupci raspoređivanja

- podjela prema vremenu i načinu donošenja odluka raspoređivanja
  - prije pokretanja (statički)
  - za vrijeme rada (dinamički)
- podjela prema postupanju sa zadatcima u izvođenju
  - pušta ih se do kraja (neprekidljivi, *non-preemptive*)
  - mogu se prekidati tijekom izvođenja (prekidljivi, *preemptive*)

4.4.1.1. Statički postupci raspoređivanja

- odluke prije pokretanja sustava
- odluke mogu biti „hardkodirane” u nekim koracima

  ne radi ništa do t1
  pokreni zadatak Z1
  kad Z1 završi ne radi ništa do t2
  pokreni Z2
  ako je Z2 završio prije t3 tada
      pričekaj do t3
  u t = t3: ...

- ili dodjela **prioriteta** zadacima, prema onome što rade, čime upravljaju
  - samo raspoređivanje ide u radu, ali s tim statički dodijeljenim prioritetima

4.4.1.2. Dinamički postupci raspoređivanja

- svojstva zadataka kojima se koristimo pri raspoređivanju mijenjaju se tijekom vremena
- neki događaj, protok vremena ili slično mijenja ta svojstva
  - npr. ako se primjenjuje raspoređivanje prema prioritetu, prioritet se dinamički mijenja
  - primjeri (koje ćemo obrađivati):
    - raspoređivanje prema roku – zadatak s najbližim rokom treba prvi raditi
    - raspoređivanje prema labavosti
    - raspoređivanje nekritičnih zadataka pravednom podjelom vremena (5.pogl.)
- problem dinamičkih postupaka – složeniji od statičkih, treba više procesorskog vremena (overhead), unose nedeterminizam (zbog složenosti koja nije O(1))

4.4.2. Postupci raspoređivanja prilagođeni računalu

- broj procesora
  - jednoprocesorska računala
  - višeprocesorska računala
- višeprocesorska nude više snage, u teoriji proporcionalno broju procesora
  - diskretni višeprocesorski, multicore (manycore), NUMA (non-uniform memory access)
  - koji algoritmi raspoređivanja, koje strukture podataka
    - organizacija i korištenje priručnog spremnika uvelike određuje učinkovitost različitih algoritama raspoređivanja
    - složenost algoritama …
  - učinkovitost jako ovisi o svojstvima zadataka
    - npr. memorijski intenzivni zadaci zagušuju zajedničku sabirnicu
      - zadaci manjeg prioriteta mogu smetati onima s većim prioritetom

4.4.3. Prekidljivost zadataka

- Pod prekidljivosti smatramo
  - prekid izvođenja jednog zadatka zbog nastavka rada drugog zadatka
- Prekid izvođenja zadatka zbog sklopovskih *prekida* i njihove obrade te po završetku povratak u isti zadatak ne smatramo prekidljivost

- U praksi se i kod prekidljivih sustava pojedini zadatak u nekim trenucima može zaštititi od prekida jednostavnom zabranom *prekida* neposredno prije nekog kritičnog posla koji ne smije biti prekinut, te dozvolom *prekida* po završetku
- Prednost neprekidljivih – jednostavnije raspoređivanje i manji overhead
- Prednost prekidljivih – bitniji mogu prekidati obrade manje bitnih

4.5.1. Jednoprocesorsko statičko raspoređivanje

- Razmatramo samo raspoređivanje prema prioritetima, statički dodijeljenima
- Kako pridijeliti prioritete zadacima?
- Negdje je to jasno iz onog što zadaci rade (čime upravljaju)
  - bitnijim dati veće prioritete (pogotovo ako trebaju brzo odgovoriti – ali tada zbog toga su „bitniji")
- U nastavku pretpostavljamo
  - da su nam svi zadaci jednako bitni,
  - želimo da svi stignu obaviti svoje u svojim rokovima

Definicija 4.1. Početne pretpostavke za raspoređivanje zadataka

1. Razmatra se konačan skup (sustav) nezavisnih zadataka:

$$\mathcal{S} = \{Z_i = \{C_i, T_i\} \mid i \in \{1..N\}\} \tag{4.2.}$$

2. Svi zadatci su periodički – pojavljaju se u sustavu u trenutcima $k \cdot T_i$, gdje je $k \in \mathbb{N}_0$ i T_i označava periodu zadatka Z_i.
3. Pojedina pojava zadatka Z_i u trenutku $k \cdot T_i$ se označava sa z_i^k.
4. Zadatak z_i^k može započeti sa svojim izvođenjem odmah po pojavi.
5. U promatranom sustavu (s dostupnom brzinom procesora) izvođenje zadatka z_i^k u svakoj periodi ($\forall k$) traje jednako i iznosi C_i.
6. Rok za završetak rada zadatka z_i^k koji se pojavio u periodi k, u trenutku $k \cdot T_i$, jest početak iduće periode: $d_i^k = (k+1) \cdot T_i$.
7. Zadatak z_i^k može se prekidati u svom izvođenju.
8. Sustav $\mathcal{S}$ uređen je tako da vrijedi: $T_1 < T_2 < \cdots < T_N$, tj. manje indekse imaju zadatci s kraćim periodama.

Definicija 4.2. Procesorska iskoristivost

Procesorska iskoristivost za sustav zadataka $\mathcal{S}$ (prema definiciji 4.1.) računa se prema:

$$U_{\mathcal{S}} = \sum_{i=1}^{N} \frac{C_i}{T_i} \tag{4.3.}$$

Definicija 4.3. Nužan uvjet rasporedivosti

Sustav zadataka $\mathcal{S}$ (prema definiciji 4.1.) zadovoljava nužan uvjet rasporedivosti ako vrijedi:

$$U_{\mathcal{S}} \leq 1 \tag{4.4.}$$

Definicija 4.4. Kritični slučaj

Za sustav zadataka $\mathcal{S}$ (prema definiciji 4.1.) definira se "kritični slučaj" kao trenutak kada se poklope početci perioda svih zadataka.

4.5.1.2. Određivanje prioriteta zadataka

- Koristi se prioritetno raspoređivanje – u svakom se trenutku od svih zadataka pripravnih za izvođenje izabire onaj zadatak koji ima najveći prioritet
- Kako postaviti prioritete zadacima
  - a da svi stignu do roka obaviti svoj posao u svakoj svojoj periodi?
- Tražimo najjednostavnije algoritme
  - parametri zadataka: T, C (d=T)
- Usporedbom T-ova:
  - oni s većim T-om **veći** prioritet?
  - oni s većim T-om **manji** prioritet?

Primjer 4.3. Dodjela prioriteta prema učestalosti pojave

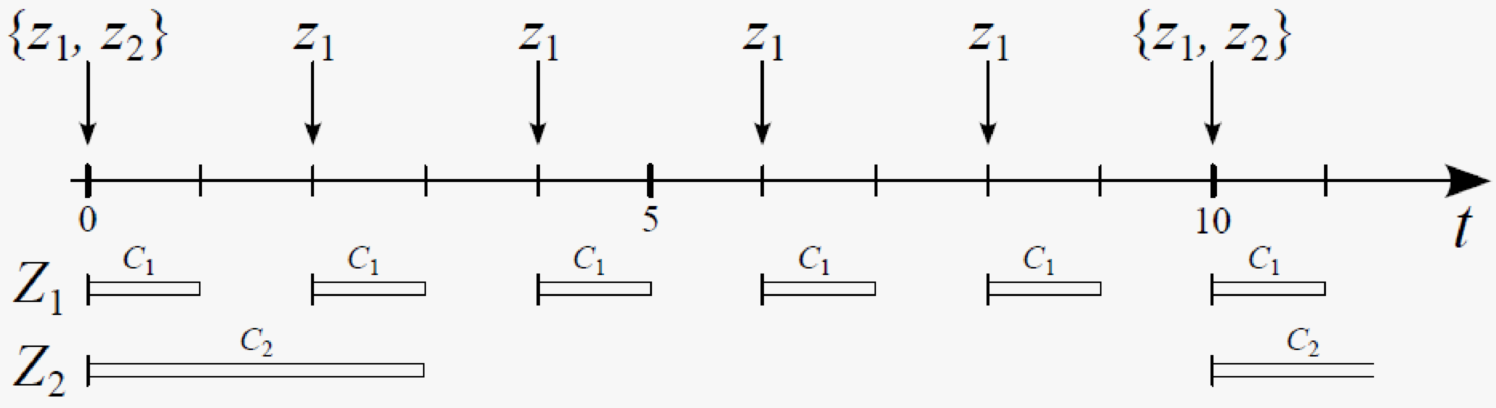
Zadan je sustav s dva zadatka $\mathcal{S} = \{Z_1, Z_2\}$. Neka se prvi zadatak javlja s $T_1 = 2$ s te neka njegovo računanje traje $C_1 = 1$ s. Drugi zadatak neka se javlja rjeđe, s $T_2 = 10$ s, ali traje dulje, $C_2 = 3$ s.

Provjerom nužnog uvjeta izvodljivosti raspoređivanja može se ustanoviti da je on zadovoljen:

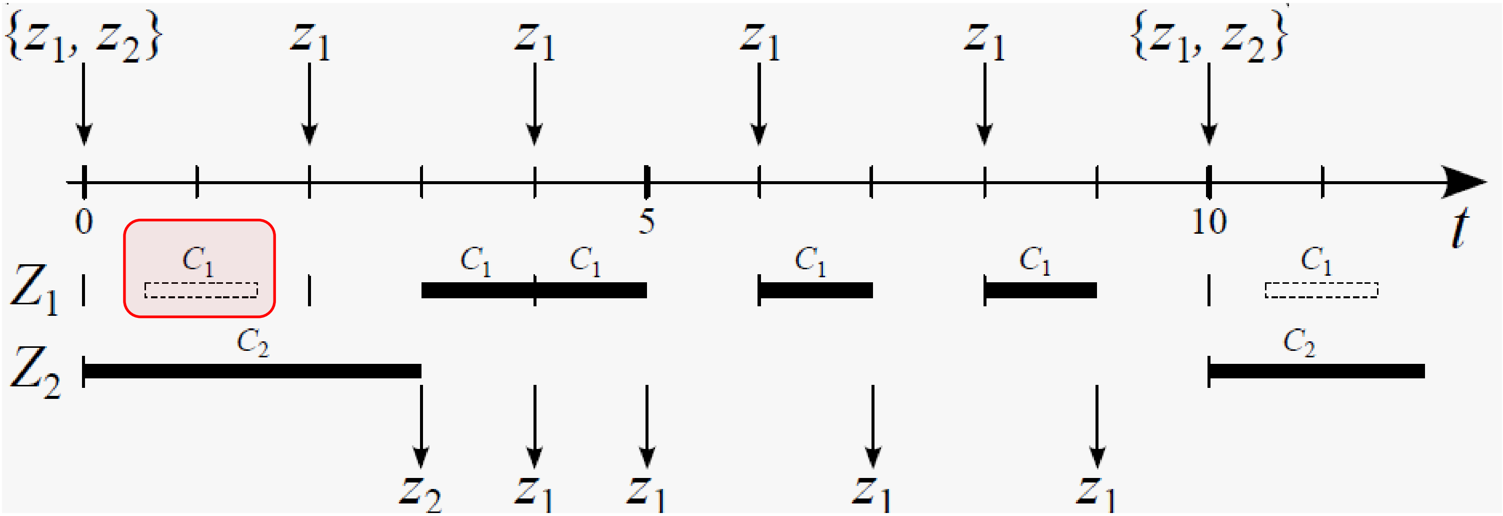
$$U_{\mathcal{S}} = \sum_{i=1}^{N} \frac{C_i}{T_i} = \frac{1}{2} + \frac{3}{10} = 0,5 + 0,3 = 0,8 \leq 1$$

Nužan uvjet ne osigurava da je sustav rasporediv korištenjem prioritetnog raspoređivača.

Slika 4.5. prikazuje zadani sustav, periode ponavljanja i potrebna vremena u svakoj periodi.

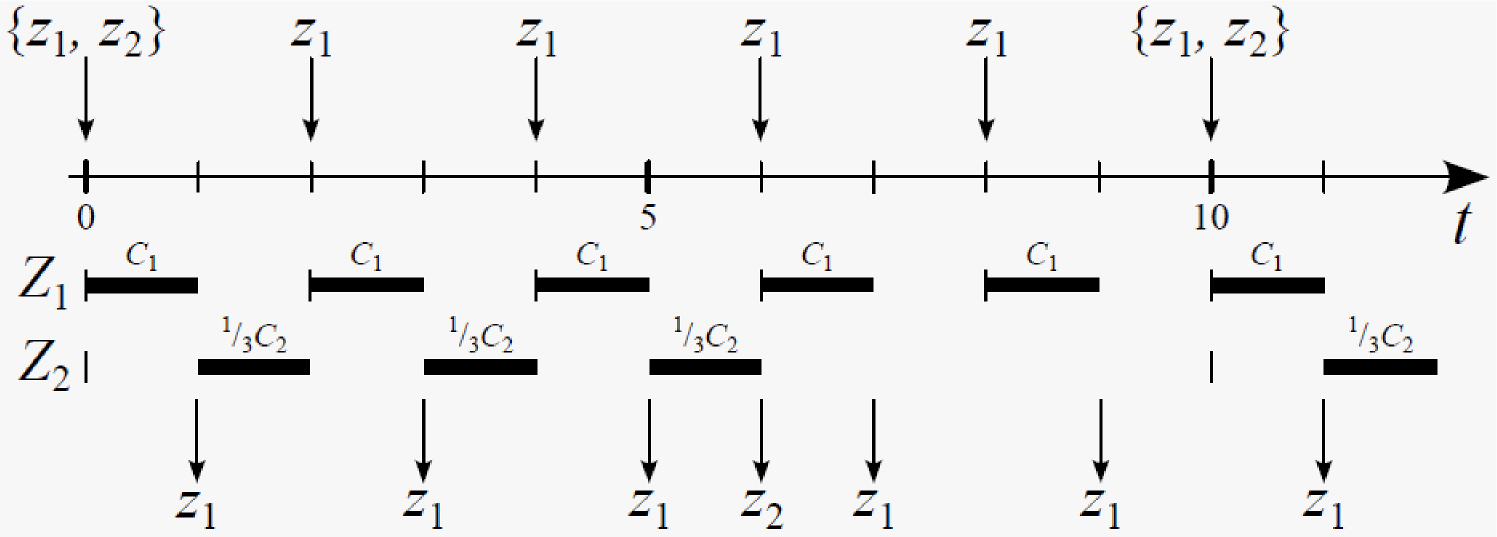


Slika 4.5. Sustav s dva zadatka



Slika 4.6. Veći prioritet zadatcima koji se rjeđe pojavljuju

C1 nije obavio svoje u prvoj periodi !!!



Slika 4.7. Veći prioritet zadatcima koji se češće javljaju

svi su stigli obaviti svoje u svakoj periodi !

Dodjela prioriteta (raspoređivanje) mjerom ponavljanja

- zadacima s kraćim periodama veći prioritet ([Liu, 1973])
  - engl. rate monotonic scheduling – RMS, rate monotonic priority assignment – RMPA

Definicija 4.5. Dodjela prioriteta mjerom ponavljanja

Postupak pridjele prioriteta zadatcima iz skupa $\mathcal{S}$ (prema definiciji 4.1.) korištenjem mjere ponavljanja, zadatcima $\{Z_1, Z_2, \ldots, Z_N\}$ će dodijeliti prioritete $\{p_1, p_2, \ldots, p_N\}$ (zadatku Z_1 prioritet p_1, zadatku Z_2 prioritet p_2, itd.) tako da vrijedi:

$$p_1 > p_2 > \ldots > p_N \tag{4.5.}$$

Zadatcima s kraćom periodom postupak mjere ponavljanja dodjeljuje veći prioritet.

- iz Definicije 4.1.: $T_1 < T_2 < \ldots < T_N$

4.5.1.3. Ograničenja pri raspoređivanju zadataka sa stalnim prioritetima

- što se sve da rasporediti samo prema prioritetu preko mjere ponavljanja?
- ajmo prvo par primjer grafički riješiti

Definicija 4.6. Grafička provjera rasporedivosti

Sustav zadataka $\mathcal{S}$ (prema definiciji 4.1.) bit će rasporediv odabranim postupkom raspoređivanja ako se grafičkom provjerom (simulacijom rada raspoređivača) u kritičnom slučaju potvrdi da su se svi zadatci koji su se pojavili u kritičnom slučaju $s = \{z_1^0, z_2^0, \ldots, z_N^0\}$ stigli obaviti do svojih trenutaka krajnjih završetaka (d_i) primjenom pravila postupka raspoređivanja, uzimajući u obzir i sve iduće pojave zadataka z_i^j koje se zbivaju za to vrijeme (dok se svi iz s ne obave).

Primjer 4.4. Grafička provjera rasporedivosti (1)

Zadan je sustav s tri zadatka s periodama i vremenima računanja prema:

$$
\begin{aligned}
Z_1 : &\quad T_1 = 5 \text{ ms}, &\quad C_1 = 2 \text{ ms} \\
Z_2 : &\quad T_2 = 15 \text{ ms}, &\quad C_2 = 5 \text{ ms} \\
Z_3 : &\quad T_3 = 25 \text{ ms}, &\quad C_3 = 5 \text{ ms}
\end{aligned}
$$

Provjera rasporedivosti:

a) Nužan uvjet

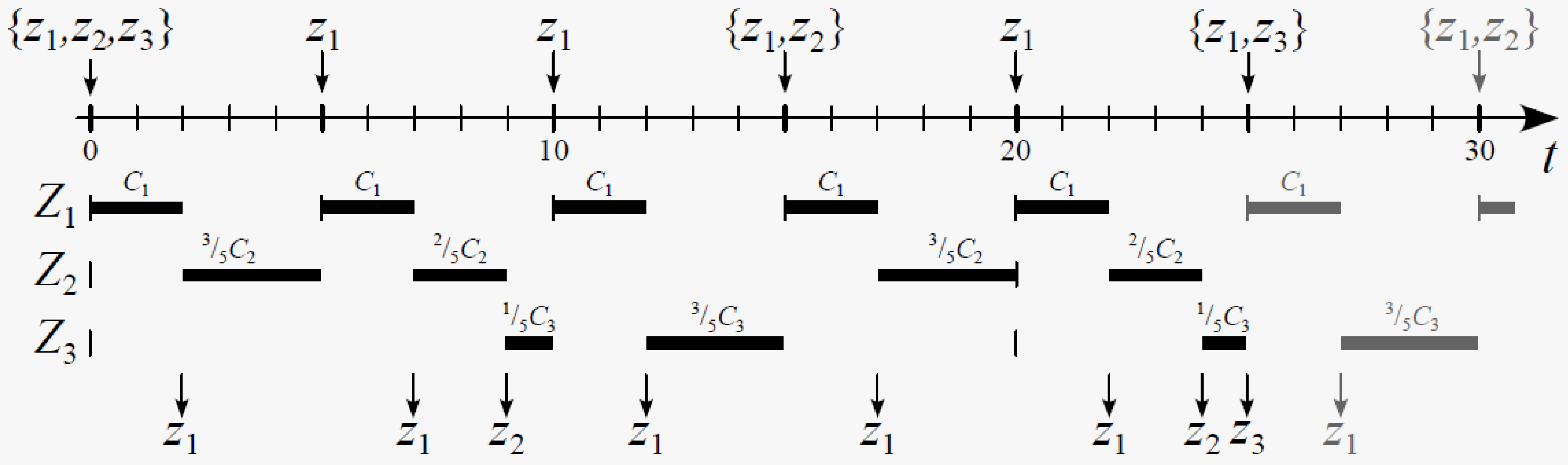
$$U = \sum_{i=1}^{3} \frac{C_i}{T_i} = \frac{2}{5} + \frac{5}{15} + \frac{5}{25} = 0,933 < 1$$

Nužan uvjet je zadovoljen.

b) Grafička provjera u kritičnom slučaju

Slika 4.8. prikazuje grafički postupak provjere rasporedivosti.

Prioritetni raspoređivač uvijek odabire zadatak najvećeg prioriteta. Prema postupku mjere ponavljanja zadatak Z_1 ima najveći prioritet s obzirom na to da ima najkraću periodu.



Slika 4.8.

rasporedivo

Primjer 4.5. Grafička provjera rasporedivosti (2)

Zadan je sustav triju zadataka s periodama i vremenima računanja prema:

$$
\begin{aligned}
Z_1 :\quad & T_1 = 10 \text{ ms}, \quad C_1 = 5 \text{ ms}\\
Z_2 :\quad & T_2 = 15 \text{ ms}, \quad C_2 = 5 \text{ ms}\\
Z_3 :\quad & T_3 = 20 \text{ ms}, \quad C_3 = 1 \text{ ms}
\end{aligned}
$$

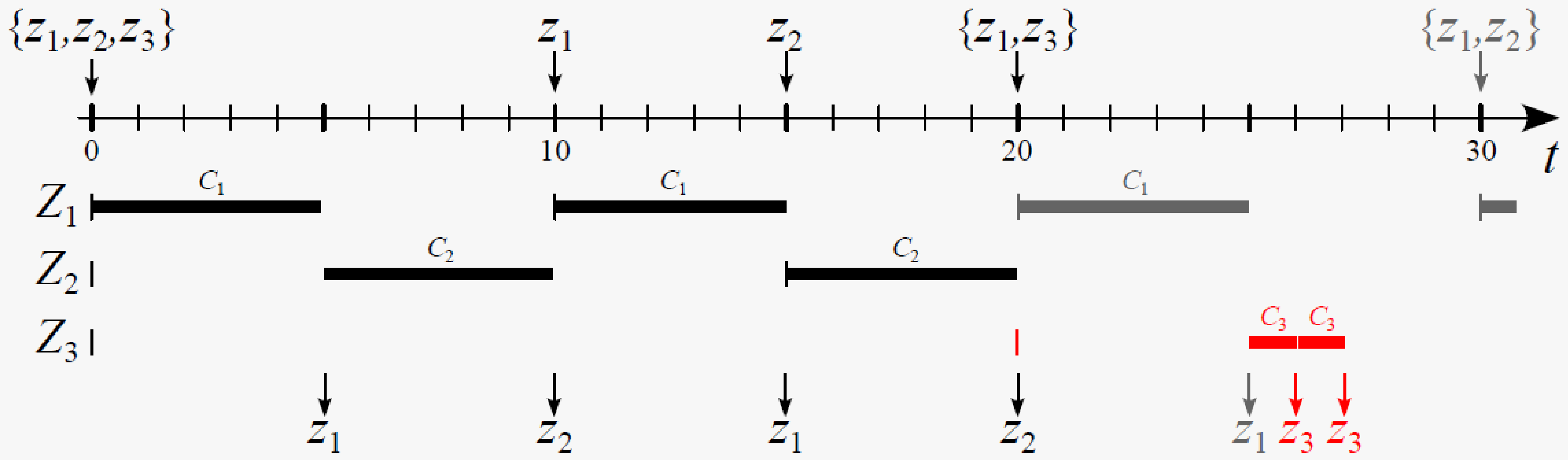
Provjera rasporedivosti:

a) Nužan uvjet

$$U = \sum_{i=1}^{3} \frac{C_i}{T_i} = \frac{5}{10} + \frac{5}{15} + \frac{1}{20} = 0,883 < 1$$

Nužan uvjet je zadovoljen.

b) Grafička provjera u kritičnom slučaju



Slika 4.9.

nerasporedivo

Definicija 4.7. Implicitni rok – ID_i

Za zadatak Z_i iz sustava zadataka $\mathcal{S}$ (prema definiciji 4.1.) definira se *implicitni rok* ID_i (engl. *implicit deadline*) kao trenutak u kojem zadatak mora završiti, računajući od pojave tog zadatka u kritičnom slučaju (za $t = 0$, za pojavu z_i^0).

ID_i može biti jednak $d_i = T_i$, ali može biti i manji ako je vrijeme od ID_i do T_i popunjeno obradama prioritetnijih zadataka.

ID_i se traži među točkama raspoređivanja zadatka Z_i.

Definicija 4.8. Točke raspoređivanja D_i

U kontekstu razmatranja rasporedivosti sustava zadataka zadanog prema definiciji 4.1. promatranog u kritičnom slučaju, *točke raspoređivanja* D_i za zadatak Z_i su trenutci idućih pojava zadataka iz skupa $\{Z_1, Z_2, \ldots, Z_i\}$ (prioritetniji zadatci od Z_i uz sam Z_i) u intervalu $\langle 0, T_i \rangle$ (pojave z_j^p uz $j \le i$, $p > 0$ te $0 < p \cdot T_j \le T_i$).

Slika 4.10. prikazuje odnos D_i u odnosu na početak i kraj periode zadatka Z_i za $D_i < T_i$.

Slika 4.10. D_i – implicitni d_i

Definicija 4.9. Opći kriterij rasporedivosti

Sustav zadataka $\mathcal{S}$ (prema definiciji 4.1.) bit će rasporediv postupkom mjere ponavljanja ako za svaki zadatak Z_i iz skupa $\mathcal{S}$ postoji $D_i \in [0, T_i]$ koji u kritičnom slučaju zadovoljava uvjet:

$$(a): \quad \sum_{j=1}^{i} \left\lceil \frac{D_i}{T_j} \right\rceil \cdot C_j \leq D_i \tag{4.6.}$$

D_i odabire se među točkama raspoređivanja zadatka Z_i (prema definiciji 4.8.). Operator $\lceil x \rceil$ vraća prvi cijeli broj jednak ili veći od x.

Definicija 4.10. Određivanje implicitna roka

Za zadatak Z_i iz sustava zadataka $\mathcal{S}$ (prema definiciji 4.1.), vrijednost D_i jest implicitni rok ID_i ako je to **najmanja vrijednost** koja zadovoljava uvjet (a) iz definicije 4.9. te uvjet (b) prema formuli (4.8.).

$$(b): \quad \sum_{j=1}^{i-1} \left(\left\lceil \frac{T_i}{T_j} \right\rceil - \left\lceil \frac{D_i}{T_j} \right\rceil \right) \cdot C_j \geq T_i - D_i \tag{4.8.}$$

Primjer 4.6.

Razmotrimo jednostavni primjer s dva zadatka $\{Z_1, Z_2\}$ koji se javljaju svakih $T_1 = 7$ te svakih $T_2 = 10$ jedinica vremena. Neka su trajanja računanja $C_1 = 3$ te $C_2 = 1$.

Pri razmatranju rasporedivosti zadatka Z_1 jedina točka raspoređivanja koja je kandidat za D_1 je $D_1 = T_1$. U njoj su zadovoljena oba uvjeta iz definicija 4.9. i 4.10.

$$
\begin{aligned}
(a):&\quad C_1 \le T_1\\
(b):&\quad 0 \ge T_1 - T_1
\end{aligned}
$$

Pri razmatranju rasporedivosti zadatka Z_2 točke raspoređivanja su: $D_2 \in \{T_1, T_2\}$ te se uvjeti provjeravaju prema:

$$
\begin{aligned}
(a):&\quad \left\lceil \frac{D_2}{T_1} \right\rceil \cdot C_1 + \left\lceil \frac{D_2}{T_2} \right\rceil \cdot C_2 \overset{?}{\le} D_2\\
(b):&\quad \left(\left\lceil \frac{T_2}{T_1} \right\rceil - \left\lceil \frac{D_2}{T_1} \right\rceil \right) \cdot C_1 \overset{?}{\ge} T_2 - D_2
\end{aligned}
$$

Za $D_2 = T_1 = 7$ uvrštavanjem se dobiva:

$$(a): \quad 1 \cdot 3 + 1 \cdot 1 \overset{?}{\leq} 7 \quad \checkmark$$

$$(b): \quad (2-1) \cdot 3 \overset{?}{\geq} 3 \quad \checkmark$$

Oba su uvjeta zadovoljena, iz čega slijedi da je zadatak rasporediv, pa tako i sustav, s obzirom na to da je i prvi zadatak rasporediv.

Što se dobiva kad bi se uzelo drugu vrijednost za D_2, tj. $D_2 = T_2 = 10$?

$$(a): \quad 2 \cdot 3 + 1 \cdot 1 \overset{?}{\leq} 10 \quad \checkmark$$

$$(b): \quad (2-2) \cdot 3 \overset{?}{\geq} 0 \quad \checkmark$$

Kada se radi samo provjera rasporedivosti sustava zadataka, dovoljno je koristiti se definicijom 4.9. (nije potrebno dodatno pronalaziti i ID_i).

Opet su oba uvjeta zadovoljena. U ovom se primjeru potvrda rasporedivosti dobiva odabirom bilo koje vrijednosti za D_2.

Prema definiciji 4.7. implicitni d_i je trenutak kada zadatak *može* završiti. S obzirom na to da za ovaj primjer vrijeme od 7. do 10. jedinice vremena troši zadatak Z_1, očito je da implicitni rok nije $T_2 = 10$, zadatak Z_2 ne može se ni izvoditi ni završiti u tom intervalu (provjera uvjeta (b) za $D_2 = 7$ to je potvrdila). Preostaje prva točka raspoređivanja $D_2 = T_1 = 7$ kada doista zadatak Z_2 može završiti s obzirom na to da je prije toga procesor slobodan (Z_1 koristi se intervalom $[0, 3]$ te je interval $[3, 7]$ slobodan za Z_2).

Primjer 4.7.

Prikažimo korištenje općeg kriterija rasporedivosti na istom sustavu kao i u primjeru 4.4.:

$$
\begin{aligned}
Z_1: \quad & T_1 = 5 \text{ ms}, \quad & C_1 = 2 \text{ ms} \\
Z_2: \quad & T_2 = 15 \text{ ms}, \quad & C_2 = 5 \text{ ms} \\
Z_3: \quad & T_3 = 25 \text{ ms}, \quad & C_3 = 5 \text{ ms}
\end{aligned}
$$

a) Rasporedivost zadatka Z_1

Za prvi, najprioritetniji zadatak Z_1 jedina točka raspoređivanja je iduće pojavljivanje tog istog zadatka, tj. provjera se obavlja za $D_1 = 5$ ms.

$$(a): \quad \left\lceil \frac{D_1}{T_1} \right\rceil \cdot C_1 \overset{?}{\leq} D_1 \quad \Rightarrow \quad \left\lceil \frac{5}{5} \right\rceil \cdot 2 \overset{?}{\leq} 5 \quad \checkmark$$

Zaključak: prvi je zadatak rasporediv.

b) Rasporedivost zadatka Z_2

Za drugi zadatak Z_2 točke raspoređivanja su: $D_2 \in \{5, 10, 15\}$ ms.

Provjera za vrijednost $D_2 = 5$ ms:

$$(a): \quad \left(\left\lceil \frac{D_2}{T_1} \right\rceil \cdot C_1 + \left\lceil \frac{D_2}{T_2} \right\rceil \cdot C_2 \right) \overset{?}{\leq} D_2 \quad \Rightarrow \quad \left(\left\lceil \frac{5}{5} \right\rceil \cdot 2 + \left\lceil \frac{5}{10} \right\rceil \cdot 5 \right) \overset{?}{\leq} 5 \quad ✗$$

Za prvu vrijednost $D_2 = 5$ prvi uvjet nije zadovoljen, tj. do 5. ms nema vremena za obavljanje prvih dvaju zadataka.

Provjera za vrijednost $D_2 = 10$ ms:

$$(a): \left(\left\lceil \frac{10}{5} \right\rceil \cdot 2 + \left\lceil \frac{10}{10} \right\rceil \cdot 5 \right) \overset{?}{\leq} 10 \quad ✓$$

Uvjet (a) je zadovoljen, zadatak Z_2 je rasporediv.

c) Rasporedivost zadatka Z_3

Za treći zadatak Z_3 točke raspoređivanja su: $D_3 \in \{5, 10, 15, 20, 25\}$ ms. Istim postupkom mogli bi provjeriti za sve točke raspoređivanja. U nastavku je dano rješenje samo za zadnju točku raspoređivanje, tj. za $D_3 = 25$ ms (u ostalim točkama zadatak nije rasporediv).

$$(a): \quad \left\lceil \frac{D_3}{T_1} \right\rceil \cdot C_1 + \left\lceil \frac{D_3}{T_2} \right\rceil \cdot C_2 + \left\lceil \frac{D_3}{T_3} \right\rceil \cdot C_3 \overset{?}{\leq} D_3 \quad \Rightarrow \quad \left\lceil \frac{25}{5} \right\rceil \cdot 2 + \left\lceil \frac{25}{15} \right\rceil \cdot 5 + \left\lceil \frac{25}{25} \right\rceil \cdot 5 \overset{?}{\leq} 25 \quad \checkmark$$

Zaključak: treći je zadatak rasporediv.

Sva tri zadatka su rasporediva te je prema tome i sustav zadataka rasporediv.

Primjer 4.8.

Zadan je sustav četiriju zadataka s periodama i vremenima računanja prema:

$$\begin{aligned}
Z_1 :\quad & T_1 = 5 \text{ ms}, && C_1 = 1 \text{ ms} \\
Z_2 :\quad & T_2 = 8 \text{ ms}, && C_2 = 1 \text{ ms} \\
Z_3 :\quad & T_3 = 9 \text{ ms}, && C_3 = 2 \text{ ms} \\
Z_4 :\quad & T_4 = 10 \text{ ms}, && C_3 = 3 \text{ ms}
\end{aligned}$$

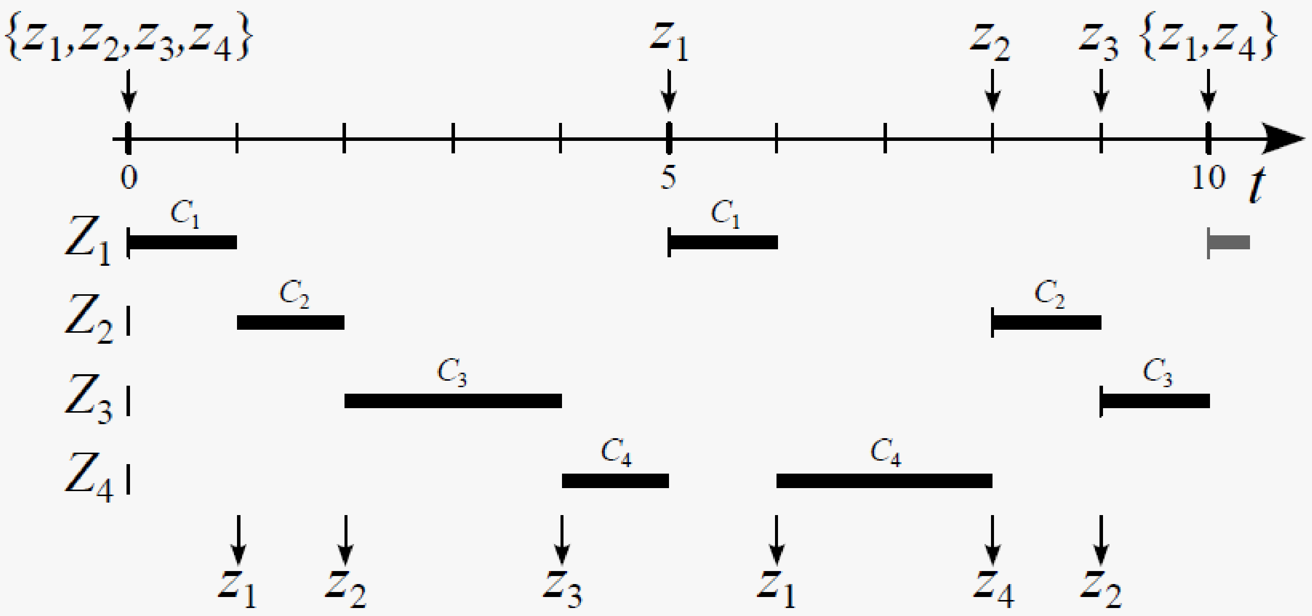
Provjera rasporedivosti:

a) Nužan uvjet

$$U = \sum_{i=1}^{4} \frac{C_i}{T_i} = \frac{1}{5} + \frac{1}{8} + \frac{2}{9} + \frac{3}{10} = 0,847 < 1 \qquad \checkmark$$

Nužan uvjet je zadovoljen.

b) Grafička provjera u kritičnom slučaju



Slika 4.11.

Za četvrti zadatak Z_4 točke raspoređivanja su: $D_4 \in \{5, 8, 9, 10\}$ ms.

Iz slike 4.11. vidljivo je da bi provjere za $D_4 \in \{5, 10\}$ bile neuspješne, odnosno za $D_4 \in \{8, 9\}$ uspješne. Međutim, $D_4 = 8$ je prava vrijednost implicitnog d_4 jer se Z_4 ne može izvoditi u intervalu $< 8, 9]$ obzirom da ga u potpunosti koristi Z_2. Prikažimo rasporedivost za Z_4 općim kriterijem i definicijom 4.10. za točku raspoređivanja $D_4 = 8$.

Uvjet (a) (rasporedivost):

$$(a): \left\lceil \frac{D_4}{T_1} \right\rceil \cdot C_1 + \left\lceil \frac{D_4}{T_2} \right\rceil \cdot C_2 + \left\lceil \frac{D_4}{T_3} \right\rceil \cdot C_3 + \left\lceil \frac{D_4}{T_4} \right\rceil \cdot C_4 \overset{?}{\leq} D_4$$

Uvrštavanjem:

$$(a): \left\lceil \frac{8}{5} \right\rceil \cdot 1 + \left\lceil \frac{8}{8} \right\rceil \cdot 1 + \left\lceil \frac{8}{9} \right\rceil \cdot 2 + \left\lceil \frac{8}{10} \right\rceil \cdot 3 \overset{?}{\leq} 8 \quad \checkmark$$

Uvjet (b) (implicitni rok):

$$(b): \left(\left\lceil \frac{T_4}{T_1} \right\rceil - \left\lceil \frac{D_4}{T_1} \right\rceil \right) \cdot C_1 + \left(\left\lceil \frac{T_4}{T_2} \right\rceil - \left\lceil \frac{D_4}{T_2} \right\rceil \right) \cdot C_2 + \left(\left\lceil \frac{T_4}{T_3} \right\rceil - \left\lceil \frac{D_4}{T_3} \right\rceil \right) \cdot C_3 \overset{?}{\geq} T_4 - D_4$$

Uvrštavanjem:

$$(b): \left(\left\lceil \frac{10}{5} \right\rceil - \left\lceil \frac{8}{5} \right\rceil \right) \cdot 1 + \left(\left\lceil \frac{10}{8} \right\rceil - \left\lceil \frac{8}{8} \right\rceil \right) \cdot 1 + \left(\left\lceil \frac{10}{9} \right\rceil - \left\lceil \frac{8}{9} \right\rceil \right) \cdot 2 \overset{?}{\geq} 10 - 8 \quad \checkmark$$

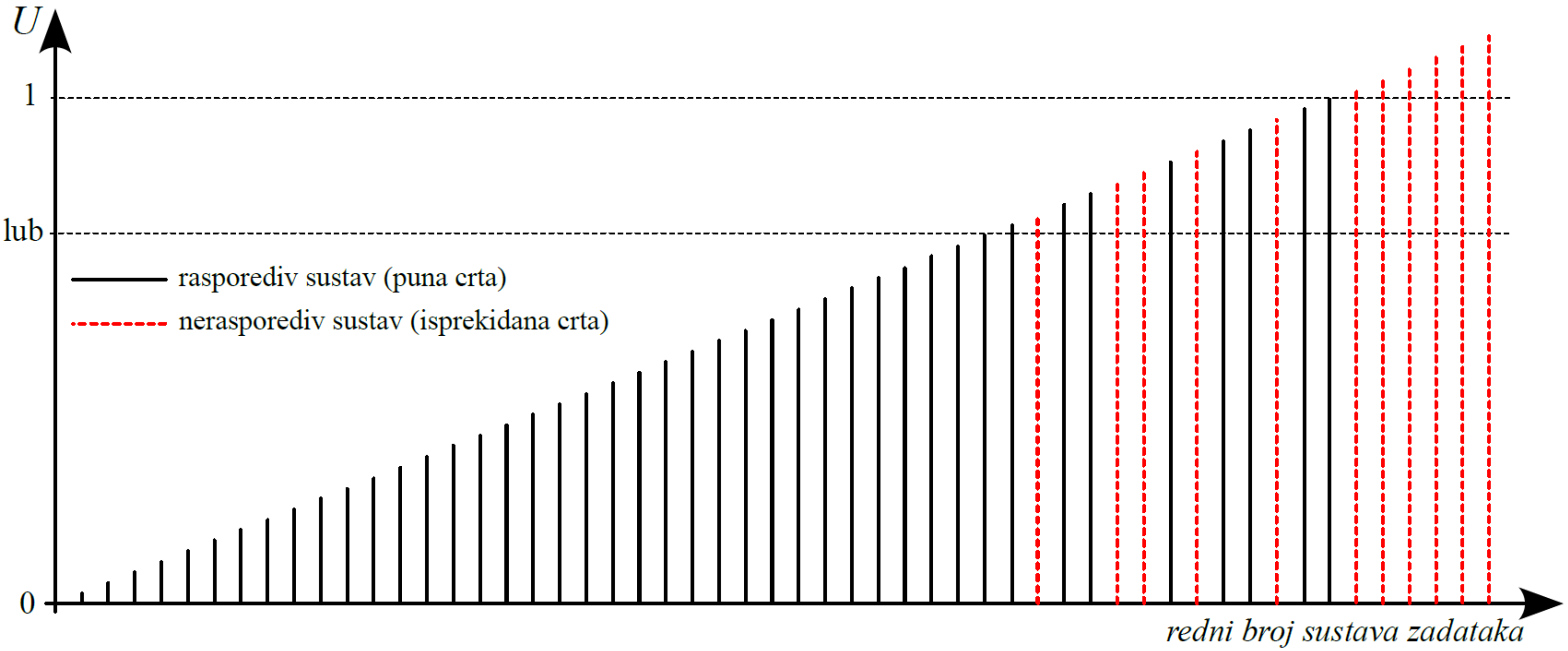
nije potrebno raditi ako se provjerava samo rasporedivost (ne traži se određivanje implicitnog roka)

Očekivani zaključak: zadatak Z_4 je rasporediv.

4.5.1.4. Granice procesorske iskoristivosti

Definicija 4.11. Potpuno iskorištenje procesora
U kontekstu raspoređivanja zadataka skup zadataka *potpuno iskorištava procesor* ako bilo koje povećanje računalnih vremena izvođenja bilo kojeg zadatka uzrokuje da sustav postaje nerasporediv zadanim postupkom.

Definicija 4.12. Najmanja gornja granica faktora procesorskog iskorištenja – $\mathrm{lub}(U)$
Najmanja gornja granica faktora procesorskog iskorištenja (engl. *least upper bound – lub*) minimalna je vrijednost faktora procesorskog iskorištenja svih mogućih skupova zadataka koji su rasporedivi i koji potpuno iskorištavaju procesor.



Slika 4.12. Primjeri sustava zadataka različitih faktora procesorskog iskorištenja

Definicija 4.13. $\mathrm{lub}(U)$ **za sustav zadataka koji se koristi mjerom ponavljanja**

Najniža gornja granica faktora iskorištenja procesora za sustav $\mathcal{S}$ s N zadataka (prema definiciji 4.1.) koji se raspoređuju prema prioritetu koji su dodijeljeni zadatcima mjerom ponavljanja (prema definiciji 4.5.) je:

$$\mathrm{lub}(U) = N\left(\sqrt[N]{2} - 1\right)$$

Definicija 4.14. Dovoljan uvjet rasporedivosti kada se koristimo mjerom ponavljanja

Sustav $\mathcal{S}$ s N zadataka (prema definiciji 4.1.) sigurno se može rasporediti postupkom mjere ponavljanja (prema definiciji 4.5.) ako za njegov faktor iskorištenja procesora $U_{\mathcal{S}}$ vrijedi:

$$U_{\mathcal{S}} \leq \mathrm{lub}(U) \qquad (4.9.)$$

Definicija 4.14. definira rasporedivost sustava samo ako je $U_{\mathcal{S}}$ **u granicama** $[0, \mathrm{lub}(U)]$. Kada $U_{\mathcal{S}}$ pada u granice $\langle \mathrm{lub}(U), 1]$, rasporedivost treba provjeriti drugim kriterijima, primjerice grafičkim postupkom ili općim kriterijem rasporedivosti.

Tablica 4.1. $\mathrm{lub}(U)$ **za različiti broj zadataka**

N	1	2	3	4	5	10	100	1000	10^{10}	∞
$\mathrm{lub}(U)$	1	0,83	0,78	0,76	0,74	0,72	0,696	0,6934	0,6931	$\ln 2$

4.5.2. Jednoprocesorsko dinamičko raspoređivanje

- 4.5.2.1. Raspoređivanje prema rokovima završetaka zadataka
  - engl. *earliest deadline first* – EDF;*deadline driven scheduling* – DDS

Definicija 4.15. Raspoređivanje prema rokovima

Raspoređivanje prema rokovima je postupak raspoređivanja koji u svakom trenutku među zadatcima spremnim za izvođenje odabire onaj koji ima najbliži rok.

Da bi sustav zadataka $\mathcal{S}$ (prema definiciji 4.1.) bio rasporediv ovom metodom na jednoprocesorskom računalu, dovoljno je da vrijedi nužan uvjet rasporedivosti (definicija 4.3.).

Ako dva ili više zadataka u nekom trenutku imaju **isti rok**, odabir jednog od njih je ili **proizvoljan** ili se definiraju **dodatni (sekundarni) kriteriji** raspoređivanja (npr. red prispijeća, prioriteti).

Primjer 4.9.

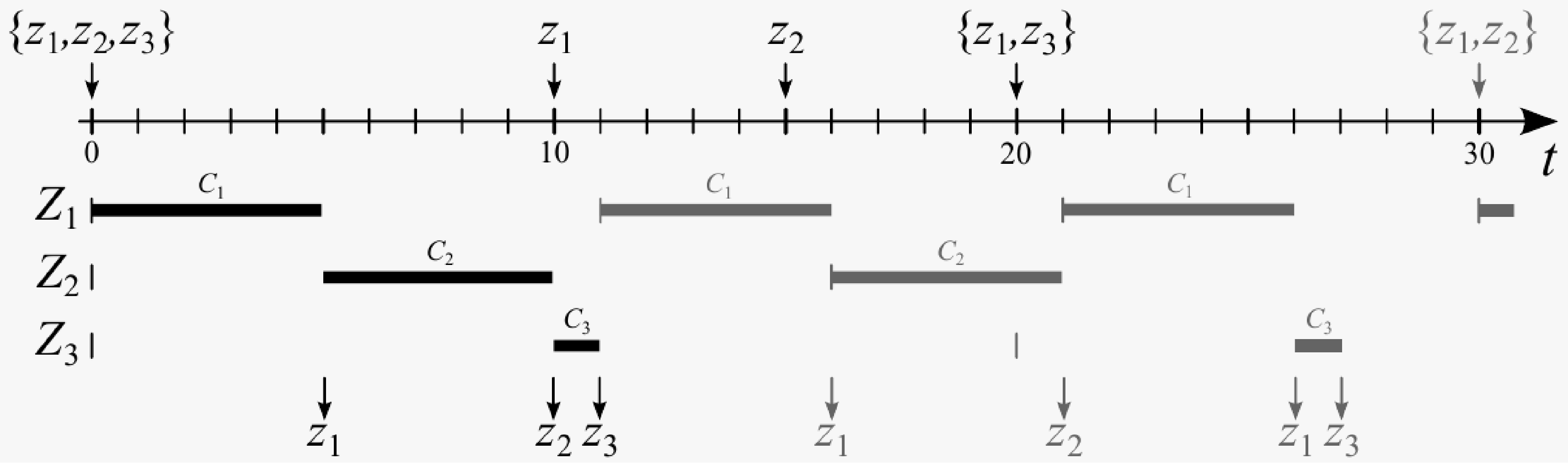
U ovom primjeru prikazano je korištenje raspoređivanja prema rokovima nad sustavom zadataka iz primjera 4.5. Zadatci u sustavu su:

$$Z_1: \quad T_1 = 10 \text{ ms}, \quad C_1 = 5 \text{ ms}$$
$$Z_2: \quad T_2 = 15 \text{ ms}, \quad C_2 = 5 \text{ ms}$$
$$Z_3: \quad T_3 = 20 \text{ ms}, \quad C_3 = 1 \text{ ms}$$

Nužan uvjet rasporedivosti:

$$U = \frac{C_1}{T_1} + \frac{C_2}{T_2} + \frac{C_3}{T_3} = \frac{5}{10} + \frac{5}{15} + \frac{1}{20} = 0,883 \leq 1 \quad \checkmark$$

Raspoređivanje prema rokovima (grafički), uz **sekundarni kriterij prema redu prispijeća** prikazano je slikom 4.13.



Primjer 4.9.

U ovom primjeru prikazano je korištenje raspoređivanja prema rokovima nad sustavom zadataka iz primjera 4.5. Zadatci u sustavu su:

$$
\begin{aligned}
Z_1 &: \quad T_1 = 10 \text{ ms}, \quad C_1 = 5 \text{ ms} \\
Z_2 &: \quad T_2 = 15 \text{ ms}, \quad C_2 = 5 \text{ ms} \\
Z_3 &: \quad T_3 = 20 \text{ ms}, \quad C_3 = 1 \text{ ms}
\end{aligned}
$$

Nužan uvjet rasporedivosti:

$$U = \frac{C_1}{T_1} + \frac{C_2}{T_2} + \frac{C_3}{T_3} = \frac{5}{10} + \frac{5}{15} + \frac{1}{20} = 0,883 \leq 1 \quad \checkmark$$

Ako bi sekundarni kriterij bio prema mjeri ponavljanja (zadatci kraće periode imaju veći prioritet) tada bi raspoređivanje od 10. ms nadalje bilo ponešto drukčije, prema slici 4.14.