Praktikum iz operativnih sistema

Lekcija 2: Procesi – osnovni pojmovi

zima 2019/2020

Prof. dr Branimir Trenkić

Zaključak prve lekcije

- “if something can go wrong, it will go wrong”
- ***Testiranja***, mada neophodna, omogućuju samo delimičnu verifikaciju ponašanja sistema
- ***Predvidljivost*** mora biti ostvarena **na nivou jezgra operativnog sistema**
- Rukovanje vršnim opterećenjima, i otpornost na greške (*fault – tolerance*)
- Kritični sitemi moraju biti projektovani na osnovu ***pesimističkih*** pretpostavki

Procesi

- ***Računarski sistem*** sa jednim procesorom – ***jedna instrukcija u jednom trenutku***
- ***Operativni sistem*** mora ***svim*** korisnicima (svim programima - procesima) da ***obezbedi pristup procesoru***
  - Da bi to uspešno ostvario – mora imati određene ***informacije o svakom korisniku (programu - procesu) u svakom trenutku***
- ***Proces*** – najvažniji koncept u teoriji operativnih sistema
- Šta predstavlja proces?

Proces - program

- Proces je ***program ili deo programa u stanju izvršavanja***, ***zajedno sa svim resursima*** koji su potrebni za rad programa
- ***Program*** – ***pasivan element*** u računarskom sistemu
- Kada se program ***učita u radnu memoriju*** – on postaje ***aktivan element*** , t.j. ***postaje proces*** – aktivan entitet koji obavlja neku aktivnost u sistemu (***definicija***)
- **Osnovni softverski entitet** tretiran od svakog operativnog sistema

Procesi

- ***Terminološka usaglašavanja***
- ***Aktivnost procesora*** se u literaturi naziva različitim imenima:
- ***Procesi*** (engl. *processes*)
- ***Poslovi*** (engl. *jobs*)
- ***Zadaci*** (engl. *tasks*)

Procesi

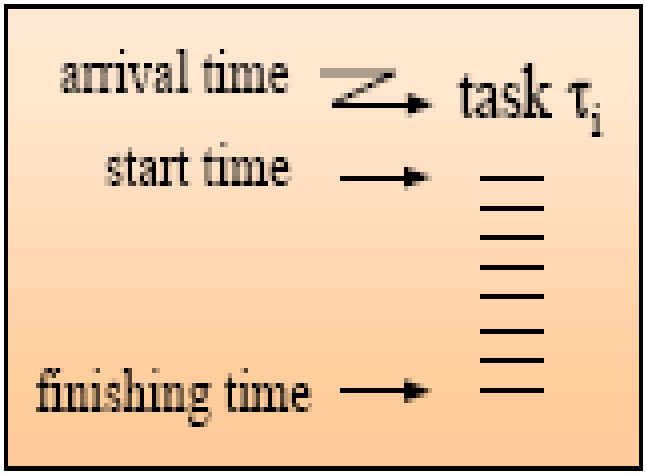
- ***Model procesa*** možemo predstaviti na ***dva načina***
- ***Prvi način*** (***u prostoru***, memoriji)
- Svaki proces ima ***tri memorijska dela*** (sekcije)

1. Programska sekcija
   - Ne menja se (*read-only*)
   - Sadrži ***programski kod***
2. Stek sekcija
   - Sadrži ***privremene podatke*** (povratne adrese, lokalne promenljive)
3. Sekcija podataka
   - Sadrži ***globalne promenljive***

Osnovni pojmovi - definicije

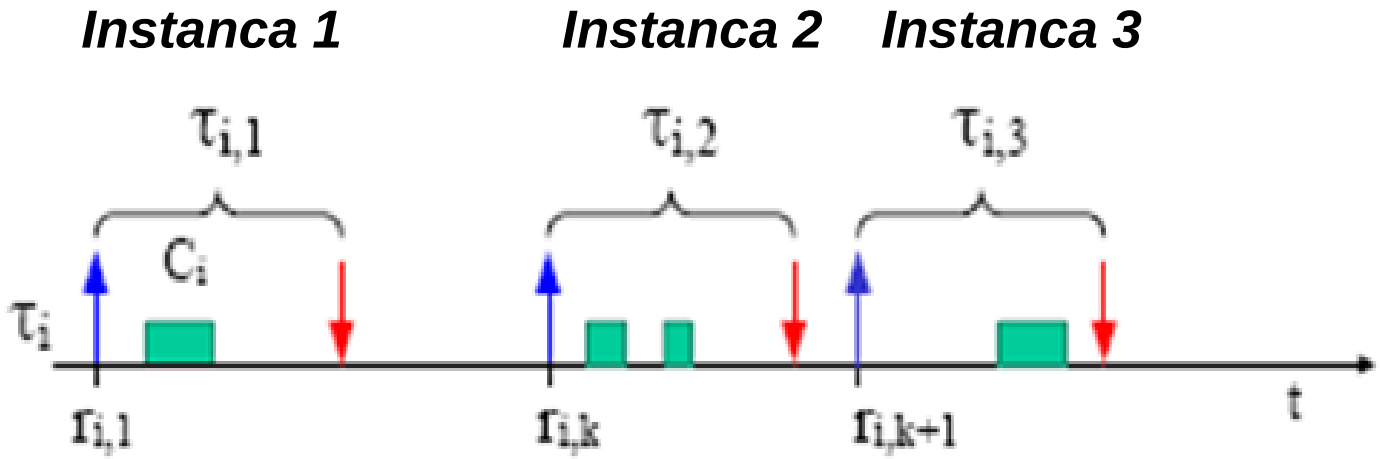
- Model procesa možemo predstaviti na ***dva načina***
- ***Drugi način*** (***u vremenu***)
- Ovaj način je ***prikladan za naš pristup*** analizi
  - niz instrukcija koje se u nedostatku drugih aktivnosti kontinualno izvršavaju sve do svog kraja

- ***Gantt***-ov dijagram:



Posao (task) i instanca (job)

- Program se najčešće izvršava više puta u sistemu
- ***Posao*** (***job***, ***task***) se može posmatrati kao beskonačni ***niz njegovih instanci*** (realizacija):



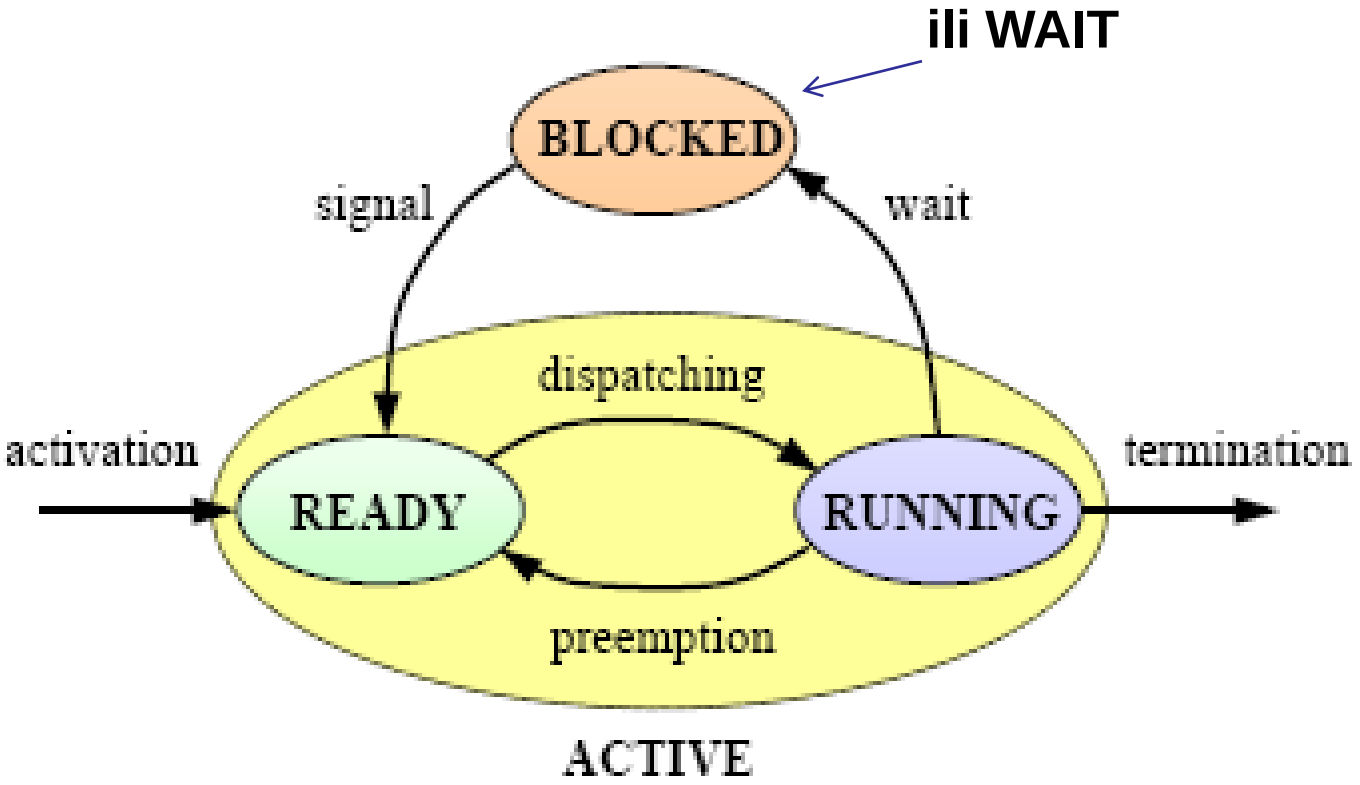
Osnovni pojmovi - definicije

- **A**) ***jedan*** procesor (CPU)
- **B**) ***skup*** konkurentnih ***poslova***
  - Međusobno se mogu preklapati u vremenu

- CPU se mora ***dodeliti*** različitim poslovima
- Predhodno definisani ***kriterijumi dodeljivanja*** – ***politika raspoređivanja***
- ***Skup pravila*** kojima se ***u svakom trenutku*** određuje redosled izvršavanja – ***algoritam raspoređivanja***
- Sama operacija dodele CPU - ***dispatching***

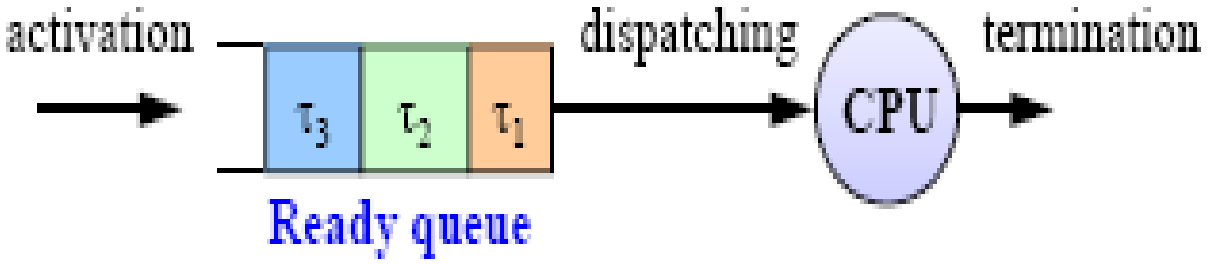
Prelazna stanja posla

Dijagram stanja procesa:



Niz spremnih poslova

- Poslovi koji su spremni za izvršavanje, se smeštaju u niz za čekanje koji nazivamo ***niz spremnih***
- ***Strategija izbora*** spremnog procesa za izvršenje na CPU naziva se ***algoritam raspoređivanja***



Prekid izvršenja - *preemption*

- U operativnim sistemima koji dozvoljavaju ***dinamičko* aktiviranje poslova**
- Posao koji se trenutno izvršava, može biti ***privremeno suspendovan*** u izvršavanju (vraćen u red spremnih), zbog izvršavanja drugog posla koji je značajniji.
- ***Operacija prekidanja*** (*preemption*) – suspenzija izvršavanja posla i njegovo ponovno ubacivanje u red spremnih poslova

Raspored (schedule)

- ***Raspored*** **je određena dodela poslova procesorima koji su na raspolaganju.**

Neka je dat ***skup poslova*** $\Gamma = \{\tau_1, \tau_2, \ldots, \tau_n\}$,

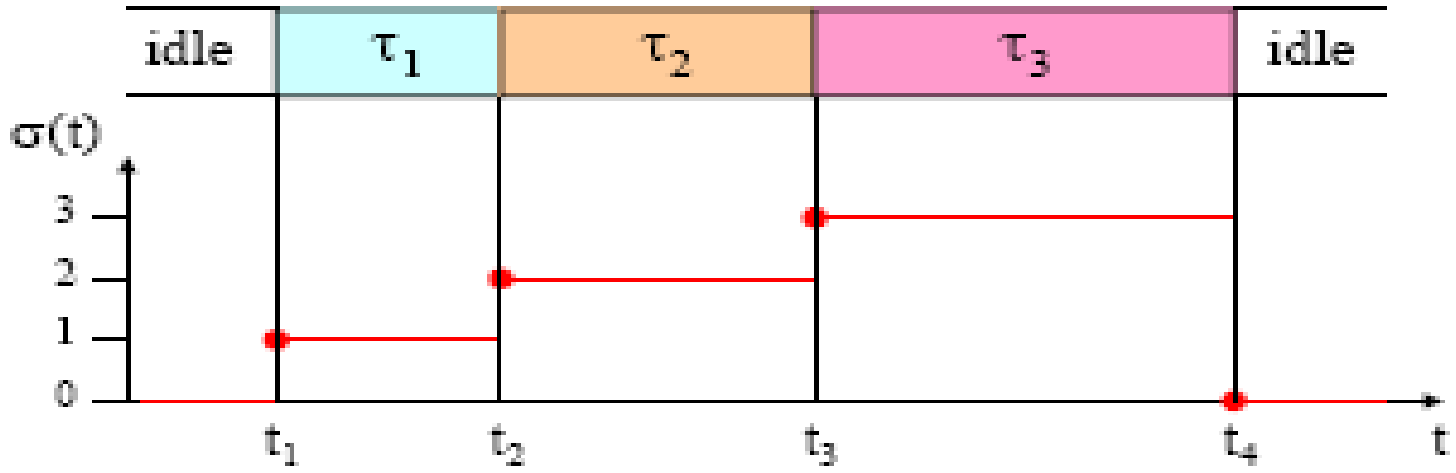
raspored je funkcija tipa $\sigma : R^+ \rightarrow N$ tako da

za $\forall t \in R^+, \exists t_1, t_2$:

$$t \in [t_1, t_2) \rightarrow \forall t' \in [t_1, t_2) : \sigma(t) = \sigma(t')$$

$$\sigma(t) = \begin{cases} k > 0 & \tau_k \text{ se izvrsava} \\ 0 & \text{procesor slobodan} \end{cases}$$

Raspored - Primer

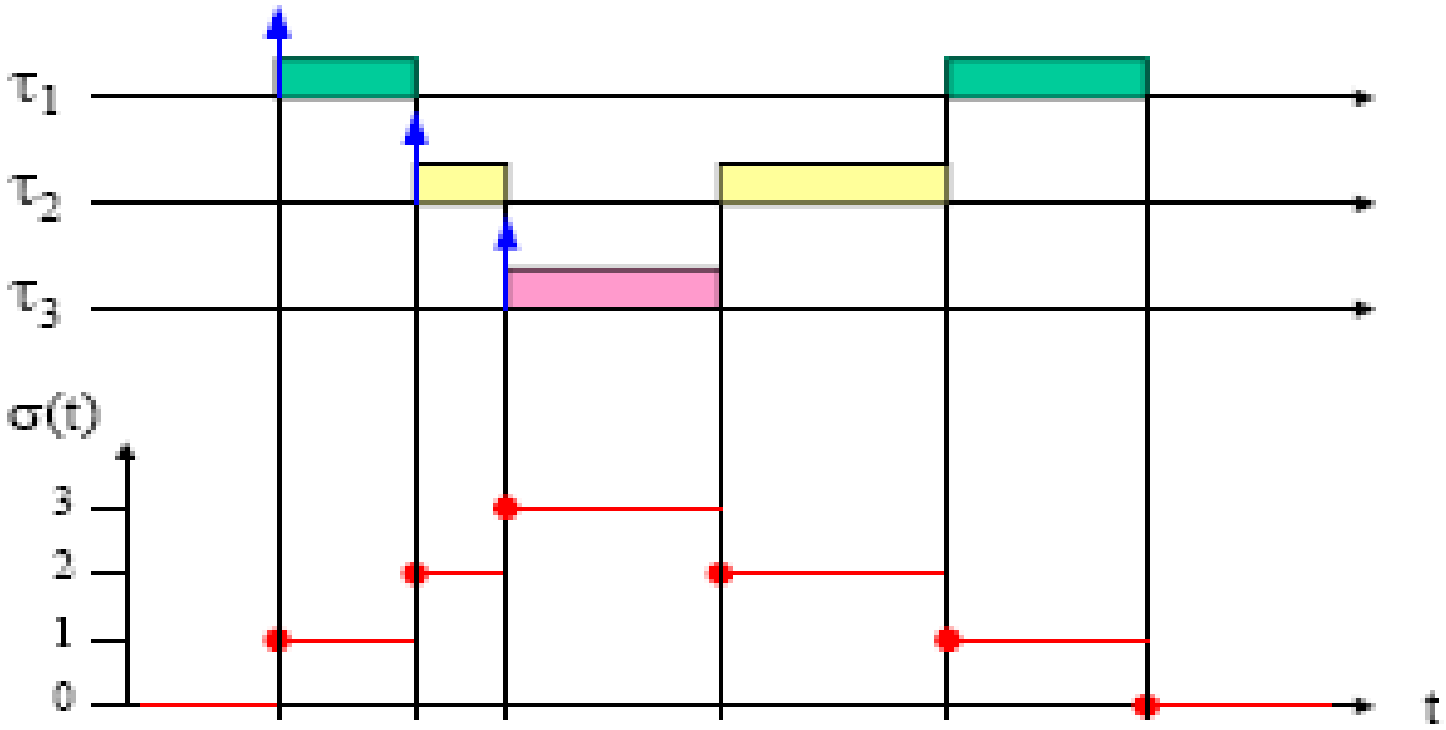


- U trenucima t_1, t_2, t_3, t_4 - ***prebacivanje konteksta***
- Svaki interval $[t_i, t_{i+1})$ u kome je $\sigma(t)$ ***konstanta*** naziva se vremenski prozor – ***time slice***

Raspored sa prekidanjem

- ***Raspored sa prekidanjem*** je raspored u kome izvršavanje nekog posla ***može biti suspendovano u bilo kom trenutku***, CPU se dodeljuje drugom poslu saglasno definisanoj politici raspoređivanja
- Na ovaj način, poslovi se mogu izvršiti u nepreklapajućim intervalima vremena

Raspored sa prekidanjem - Primer



Raspored

- Neke važne ***definicije***:
- Za **raspored** se kaže da je **izvodljiv**, ako se izvršavanje svih poslova kompletira ***saglasno skupu specificiranih zahteva***
- Za **skup poslova** kažemo da je **rasporedljiv** ako postoji ***najmanje jedan*** ***algoritam raspoređivanja*** koji može da ***proizvede izvodljiv raspored***

Tipovi zahteva na poslovima

- ***Vremenski*** zahtevi
  - Aktiviranje, kompletiranje, odstupanje
- Zahtevi po pitanju ***prvenstva izvršavanja***
  - Obezbeđuju uređenost (redosled) izvršavanja
- Zahtevi po pitanju ***deljenih resursa***
  - Uslovljavaju ***sinhronizaciju u pristupu resursima*** koja zahtevaju ***međusobnu isključivost u pristupu***

Vremenski zahtevi

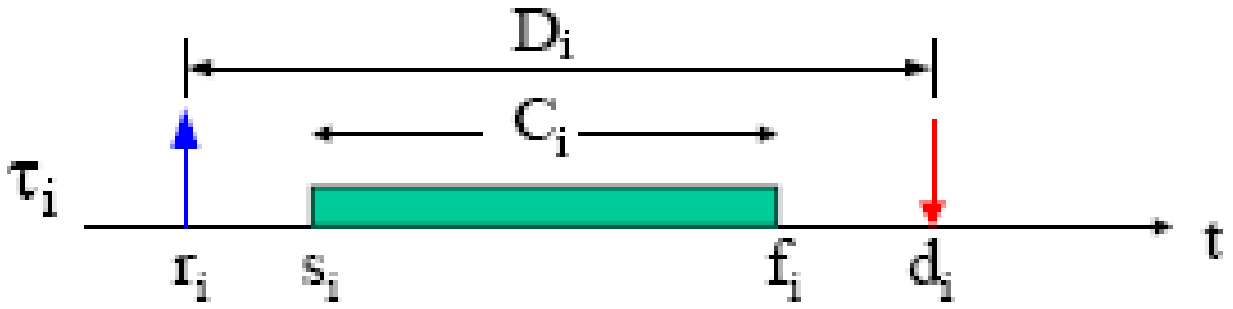
- ***Vremenski zahtevne sisteme*** karakterišu ***pre svega vremenski zahtevi***
- Tipičan vremenski zahtev: **deadline**

deadline =

vreme pre kog se posao treba kompletirati (izvršiti)

- ***Odnos prema prekoračenju*** deadline-a definiše ***dve klase sistema***:
- ***Hard***: prekoračenje deadline-a može dovesti do ***katastrofalnih posledica***
- ***Soft***: prekoračenje deadline-a ***smanjuje performanse*** sistema

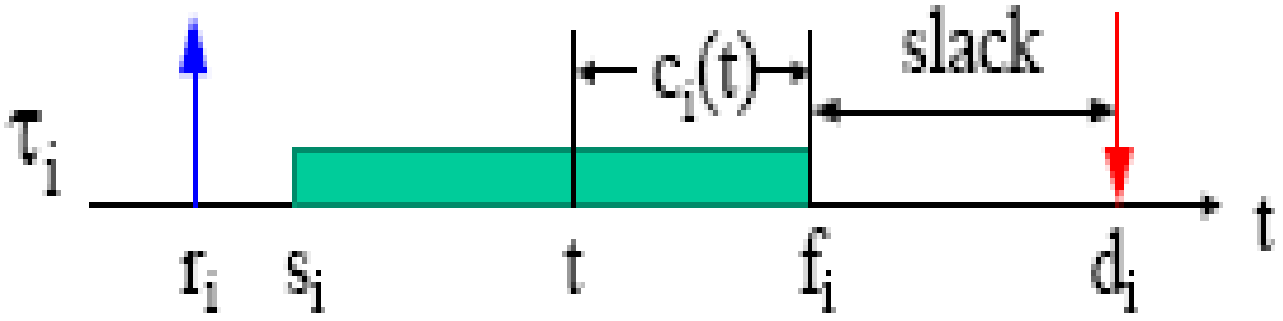
Vremenski zahtevi



Vremenski parametri:

- ***r_i*** – vreme aktiviranja (trenutak nailaska ***a_i***)
- ***s_i*** – vreme početka izvršavanja
- ***C_i*** – vreme izvršavanja ***u najgorem slučaju*** (wcet)
- ***d_i*** – apsolutni ***deadline***
- ***D_i*** - relativni deadline
- ***f_i*** – vreme završetka izvršavanja

Još neki parametri

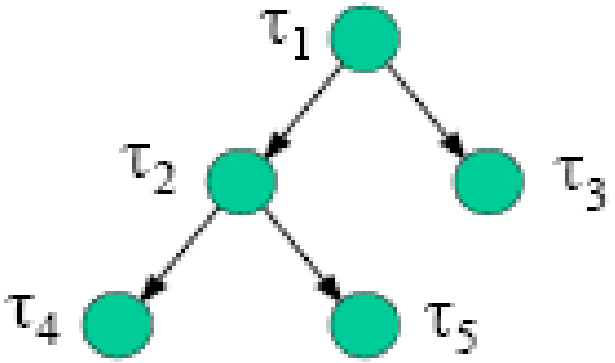


- Prekoračenje - $L_i = f_i - d_i$
- Maksimalno prekoračenje – $L_{max} = max(0, L_i)$
- Rezidualno vreme izvršavanja (wcet)– $c_i(t)$ $c_i(r_i)=C_i$
- Besposlenost ("***prazan hod***") (*slack*) –

$$X_i(t) = d_i - t - c_i(t)$$

Uslovljenost odnosima prvenstva

- Ponekad se poslovi moraju izvršavati u određenom redosledu, specificiranom sa ***direktnim acikličnim grafom***:



predhodnik

$\tau_1 < \tau_4$

neposredni predhodnik

$\tau_1 \rightarrow \tau_2$

Početni poslovi / krajnji poslovi

Uslovljenost po pitanju resursa

- **Resurs** je bilo koja softverska struktura, koju proces može iskoristiti ***u cilju unapređenja izvršenja***

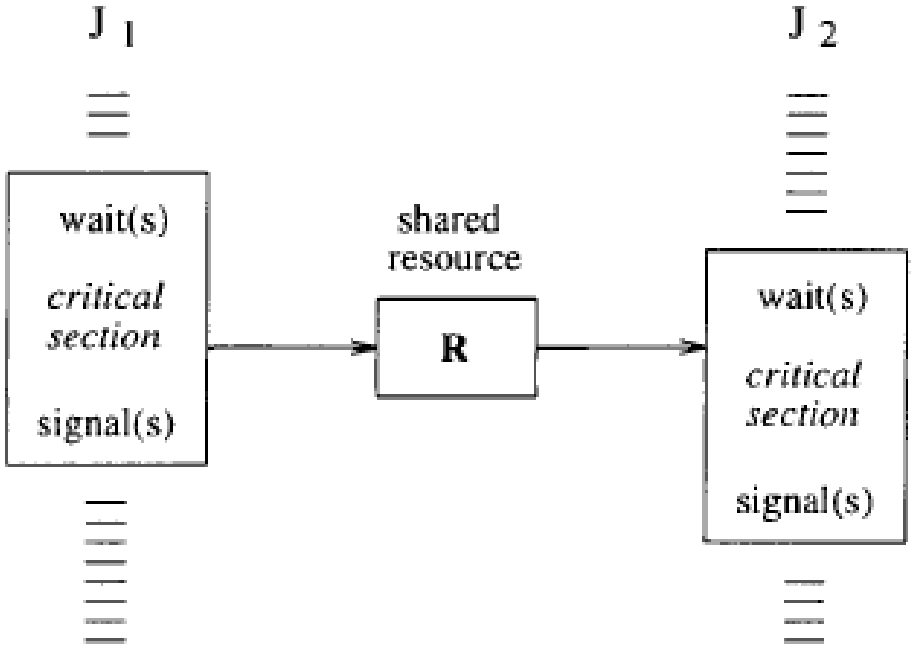
A. Resurs dodeljen ***samo jednom*** (određenom) procesu – ***privatni resurs***

B. Resurs koji može koristiti ***više*** procesa – ***deljeni resurs***

- ***Kod deljenog resursa*** se ne sme dozvoliti istovremeno korišćenje od strane više procesa, već ***međusobna isključivost u pristupu*** – ***konzistentnost podataka***
- Kritična sekcija

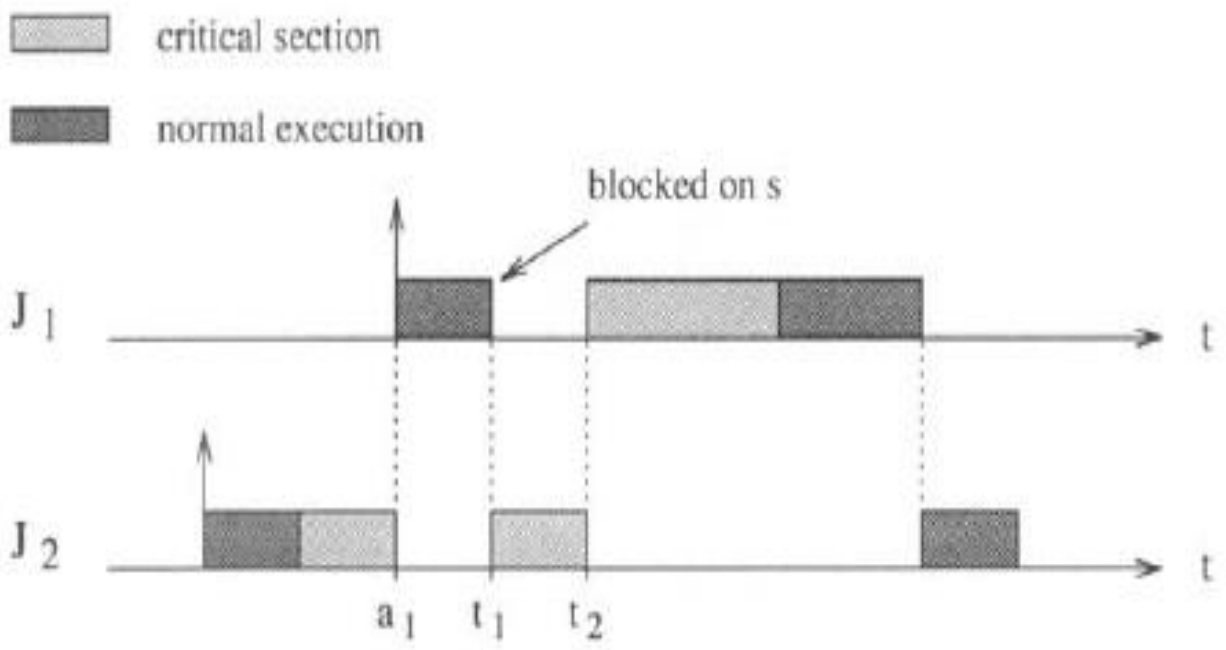
Uslovljenost po pitanju resursa

- ***Operativni sistem*** često obezbeđuje ***mehanizam sinhronizacije*** (kao što je npr. ***semafor***)
- Dva posla J_1 i J_2 dele (sa isključivošću) resurs R:



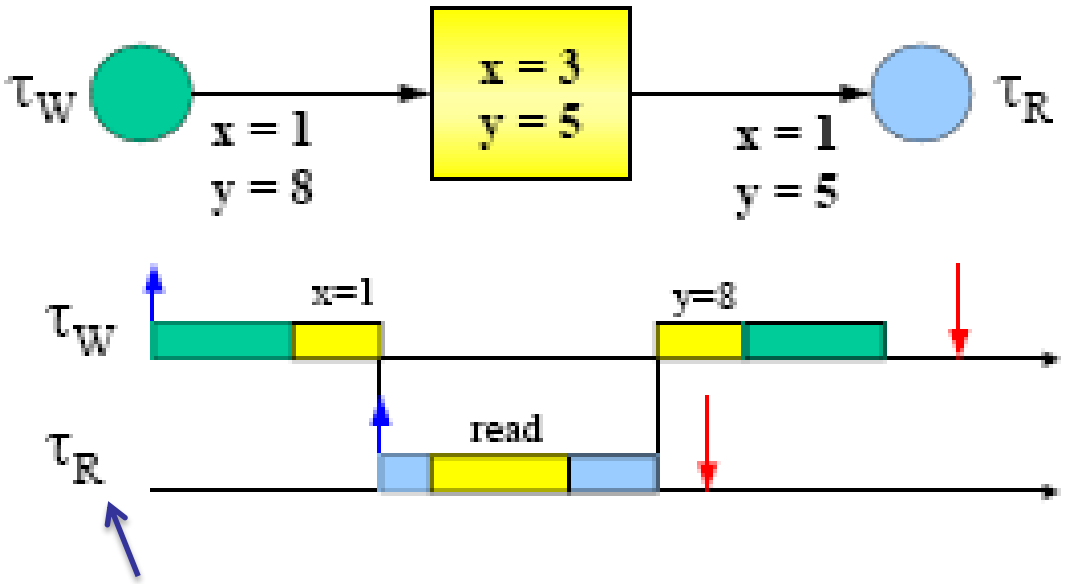
Uslovljenost po pitanju resursa

- Šta je ***blokiranje***?
- Primer blokiranja na ekskluzivnom resursu:



Uslovljenost po pitanju resursa

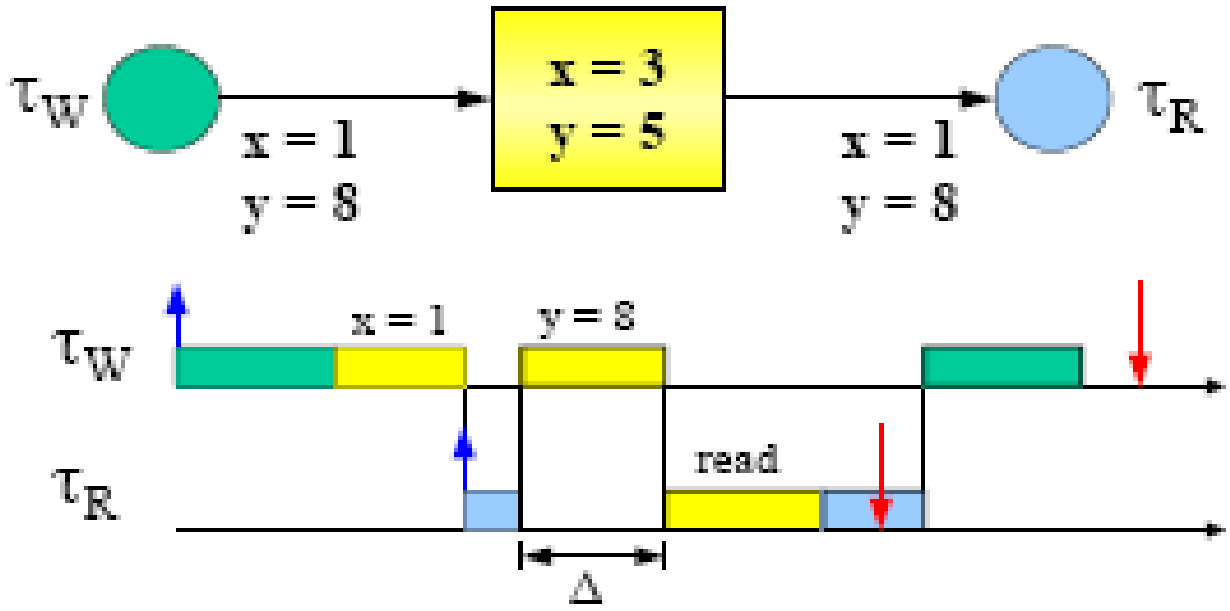
- Da bi se ***zaštitila konsistentnost podataka***, deljenim resursima se mora pristupati sa ***međusobnom isključivošću***:



Prioritetniji posao jer ima bliži deadline!

Uslovljenost po pitanju resursa

- Međutim, međusobna ***isključivost*** prouzrokuje ***dodatno kašnjenje***:



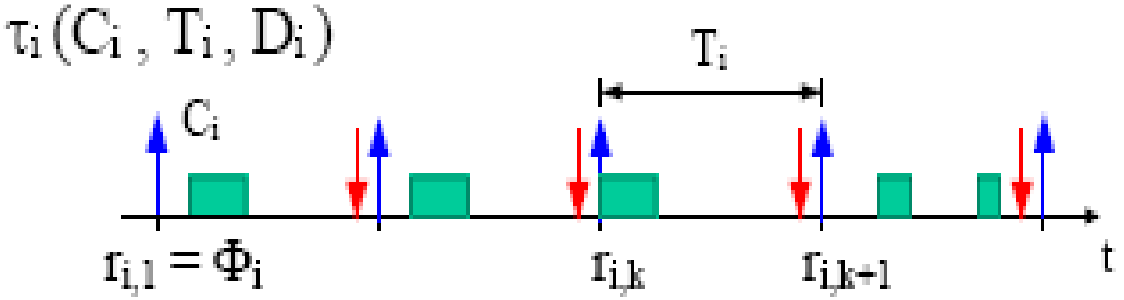
Ostali vremenski zahtevi

- Odnose se na ***regularnost aktiviranja poslova***
- ***Periodični poslovi***
  - ***vremenski pokretani***
  - pokreću se od strane ***jezgra operativnog sistema*** u ***regularnim vremenskim intervalima***
- ***Aperiodični poslovi***
  - ***pokrenuti od strane nekog događaja***
  - Pokretanje ovih poslova je uslovljeno dešavanjem nekog događaja, ili kroz eksplicitno pozivanje aktivacione primitive
- ***Sporadični (ne periodični) poslovi***

Model periodičnog posla

$$\begin{cases} r_{i,1} = \Phi_i \\ r_{i,k+1} = r_{i,k} + T_i \end{cases}$$

Φ_i – faza
T_i - perioda



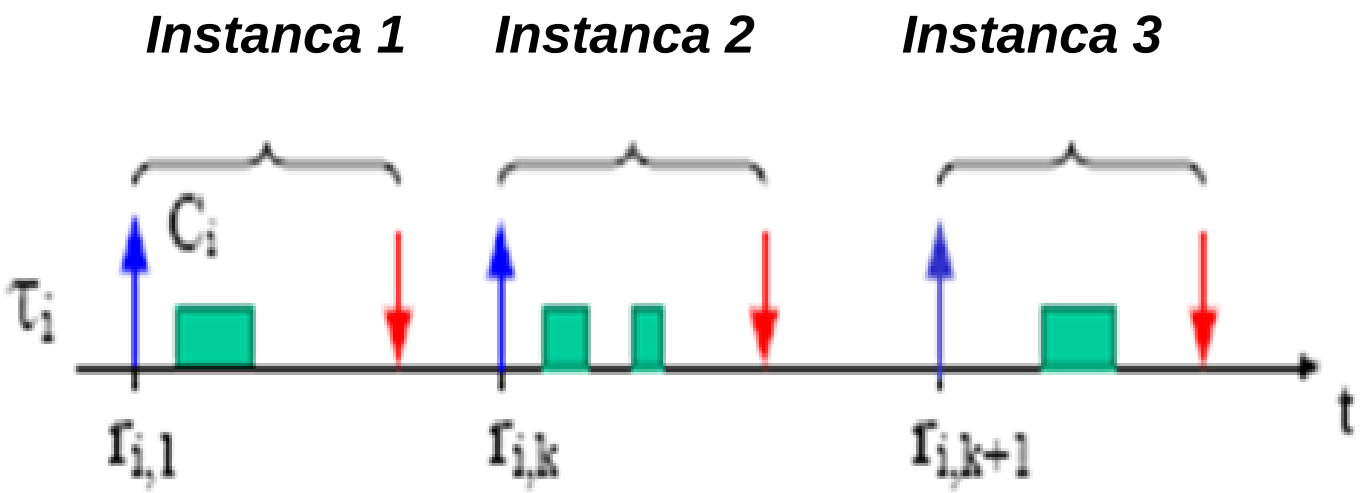
$$r_{i,k} = \Phi_i + (k-1)\,T_i$$
$$d_{i,k} = r_{i,k} + D_i$$

Često $D_i = T_i$

Model aperiodičnog posla

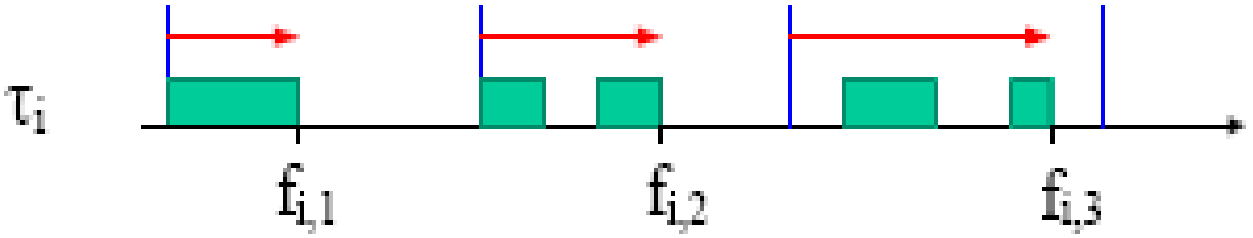
- ***Aperiodičan***: $r_{i,k+1} > r_{i,k}$

- ***Sporadičan***: $r_{i,k+1} \geq r_{i,k} + T_i$



Odstupanje (Jitter)

- Parametar ***vremenske varijacije*** periodičnog događaja:
- Odstupanje vremena ***završetka izvršavanja***

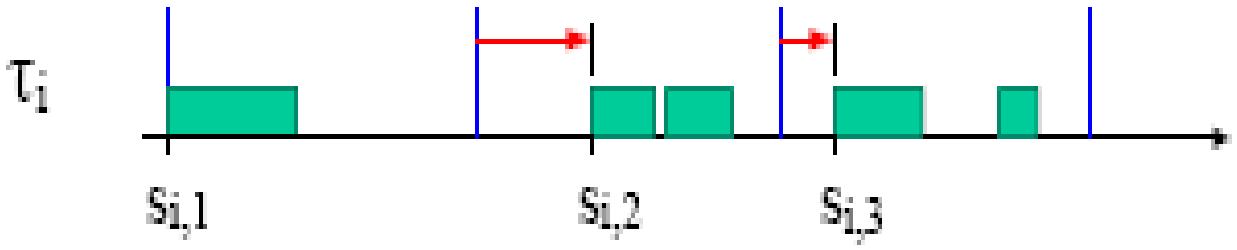


Apsolutni: $max(f_{i,k} - r_{i,k}) - min(f_{i,k} - r_{i,k})$ po k

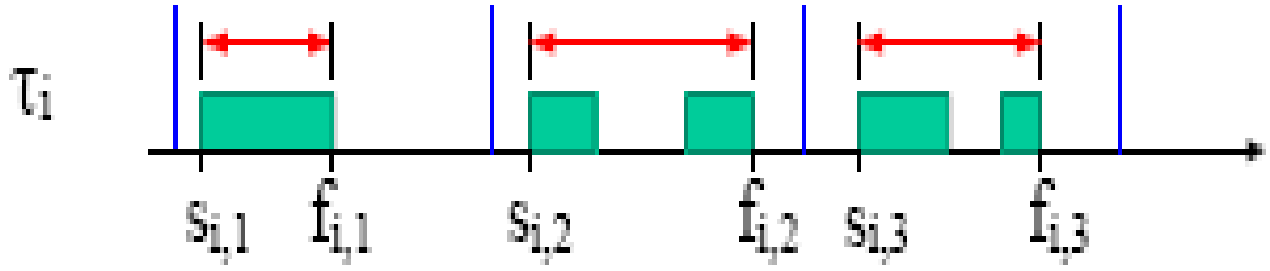
Relativni: $max\ |(f_{i,k} - r_{i,k}) - (f_{i,k-1} - r_{i,k-1})|$ po k

Drugi tipovi odstupanja

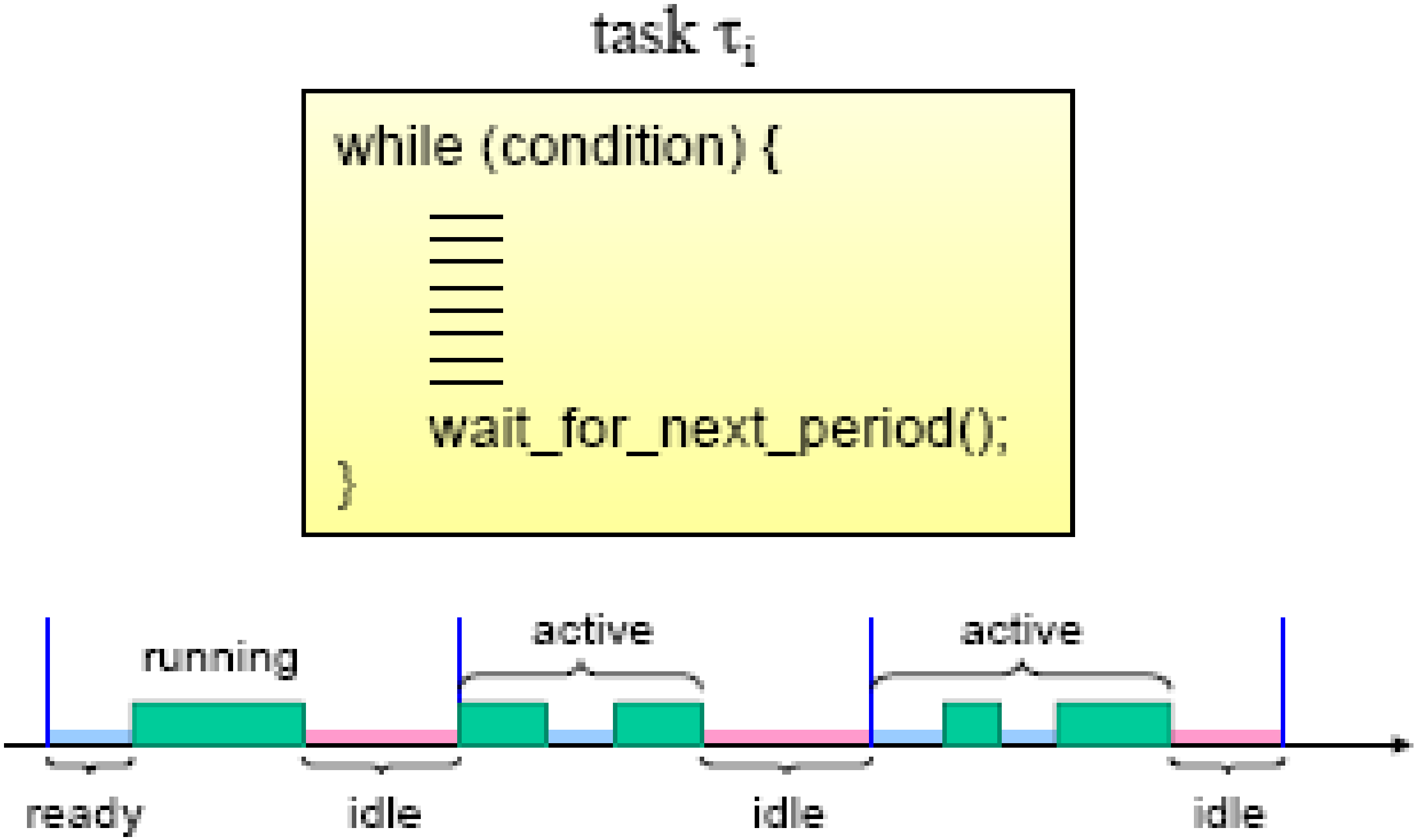
- Odstupanje vremena ***početka izvršavanja***



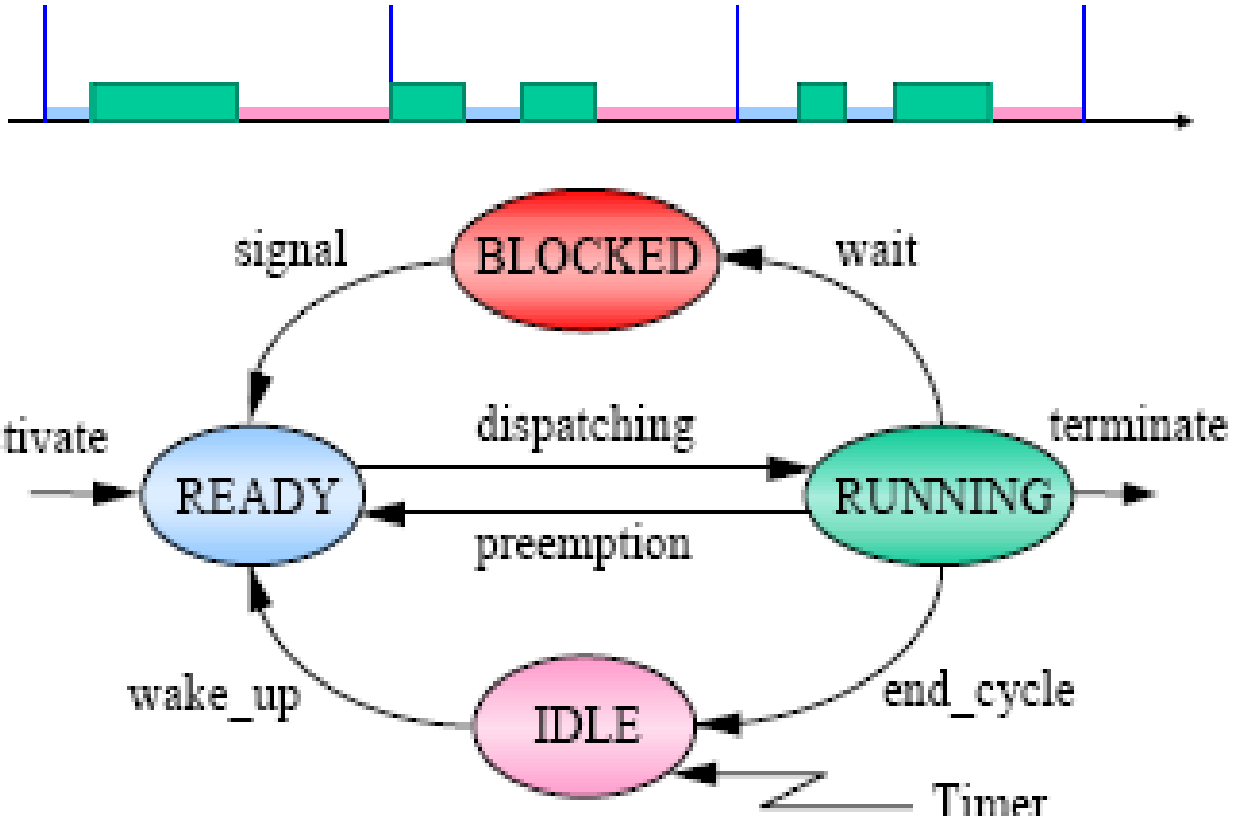
- Odstupanje u odnosu na ***vreme kompletiranja***



OS podrška periodičnim poslovima

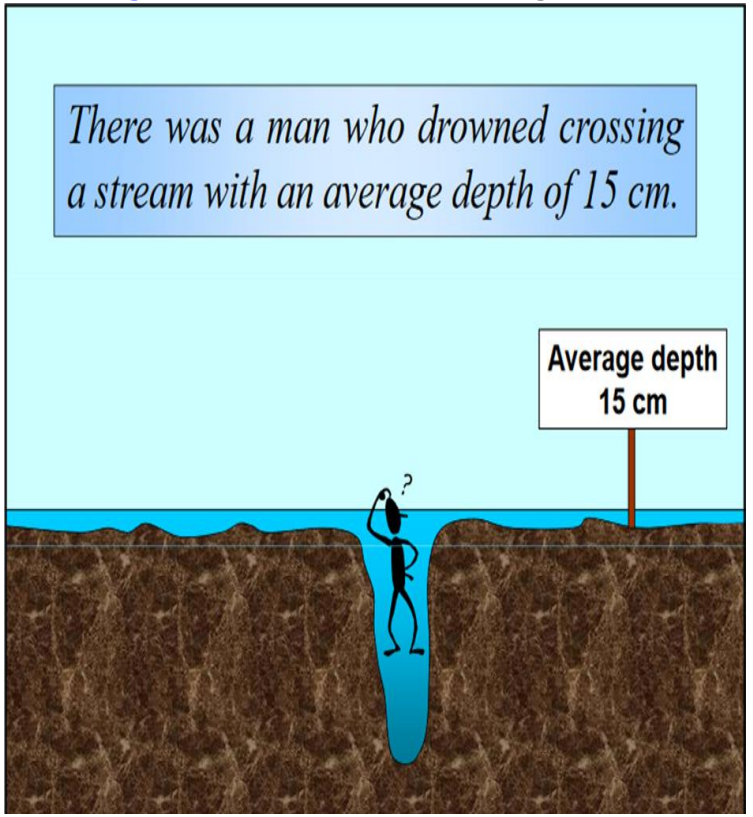


IDLE stanje posla



Mere za procenu performanse

- Procena kroz ***cost- funkciju*** definisanu ***na skupu poslova***
- ***Klasični algoritmi raspoređivanja*** pokušavaju da minimiziraju:
  - Prosečno vreme odziva
  - Ukupno vreme kompletiranja
  - Težinsku sumu vremena kompletiranja
  - Maksimalno vremensko prekoračenje (*lateness*)



Mere za procenu performanse

- ***Deadline*** kao uslov
  - Kompletiranje svih poslova u skupu ***pre njihovih deadline-ova***
- Ako se neki *deadline*-ovi ne mogu zadovoljiti sa određenim algoritmom A => ***raspored je neizvodljiv sa A***

Mere za procenu performanse

- Neke od čestih ***cost- funkcija***:
- ***Vreme odziva (turnaround time, response time)***
  - Vreme potrebno da se nakon slanja zahteva (trenutak aktiviranja, a_i) pojave prvi rezultati izvršenja procesa (vreme završetka izvršavanja, f_i) – $f_i - a_i$ (za i-ti posao)
  - ***Prosečno vreme odziva*** (n poslova):

$$\bar{R} = \frac{1}{n} \sum_{i=1}^{n} (f_i - a_i)$$

  - Zavisno od namene sistema algoritmi se optimizuju za određene kriterijume, najmanje, najveće ili srednje vrednosti

Mere za procenu performanse

- Neke od čestih ***cost- funkcija***:
- ***Vreme potrebno za kompletiranje posla*** (*burst time,* C_i)
  - Ukupno vreme potrebno da se izvrši pojedinačni posao (proces)
  - ***Ukupno vreme kompletiranja***:

$$t_r = \max_i(f_i) - \min_i(a_i)$$

Mere za procenu performanse

- Neke od čestih ***cost- funkcija***:
- ***Vreme čekanja*** (*waiting time*)
  - Razlika između turnaround (response) time i burst time (vreme potrebno za kompletiranje, C_i)
  - ***Srednje vreme čekanja (n poslova)***:

$$\bar{W} = \frac{1}{n}\left(\sum_{i=1}^{n}\left(f_i - (a_i + C_i)\right)\right)$$

Mere za procenu performanse

- ***Težinska suma vremena kompletiranja***:

$$t_w = \sum_{i=1}^{n} w_i f_i$$

- ***Maksimalno prekoračenje***:

$$L_{\max} = \max_i \left(f_i - d_i \right)$$

- ***Maksimalan broj poslova koji su prekoračili svoj deadline***:

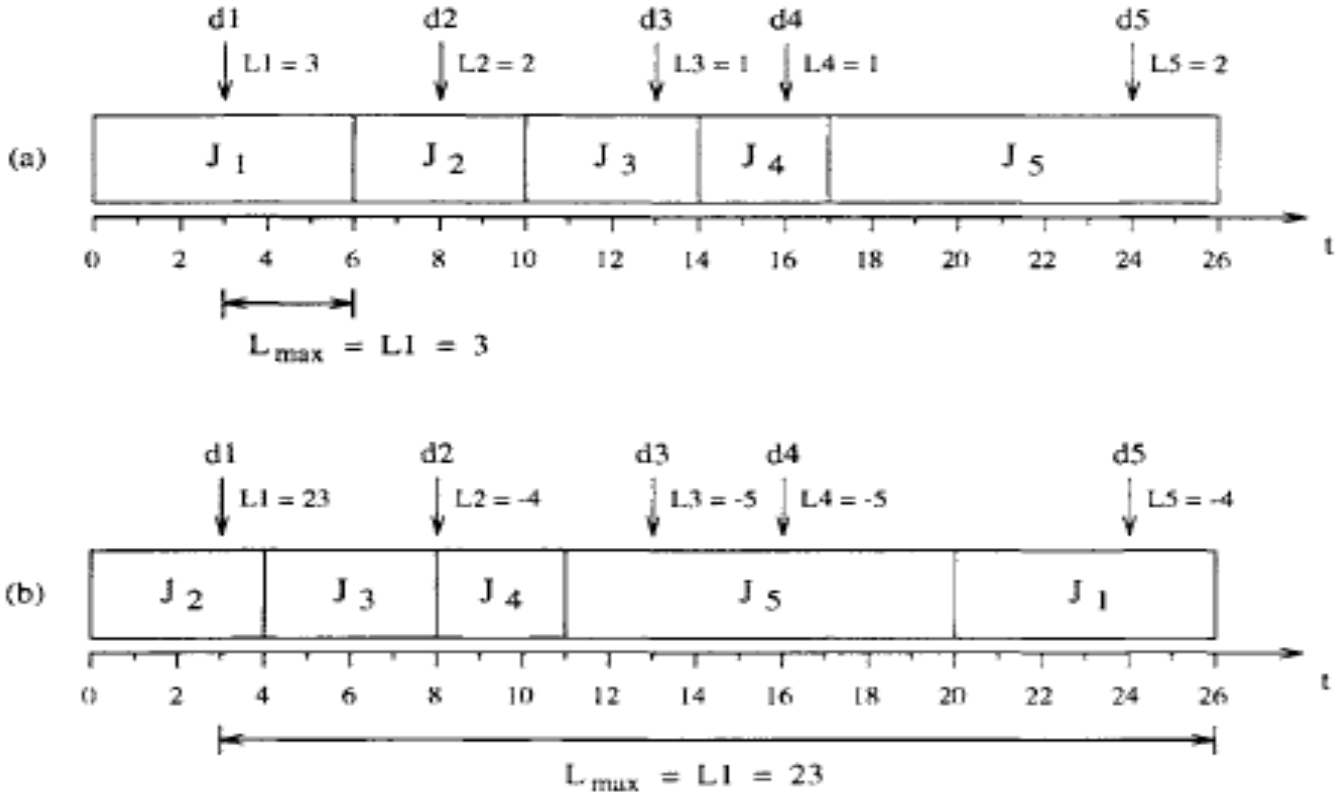
$$N_{late} = \sum_{i=1}^{n} miss\left(f_i \right) \qquad miss\left(f_i \right) = \begin{cases} 0 & f_i \le d_i \\ 1 & inace \end{cases}$$

Prikladnost datih mera

- Usvajanje neke od navedenih mera ***za ciljnu***, u određenom algoritmu raspoređivanja, može imati ***velike implikacije na performanse*** vremenski zahtevnih ***sistema***
  - ***Prosečno vreme odziva*** je ***neprikladna mera*** za (***hard***) ***sisteme*** – nema direktne ocene ***individualnih vremenskih zahteva*** kao što su period ili deadline
  - ***Ukupno vreme kompletiranja*** – isto
  - ***Težinska suma vremena kompletiranja*** je ***relevantna samo*** ako su ***poslovi različitog značaja*** na kompletiranje celog sistema
  - ***Maksimalno prekoračenje može biti relevantno u vreme projektovanja***, ako se resursi mogu dodavati do lateness = 0

Prikladnost datih mera

Maksimalno prekoračenje - $L_{\max} = \max_i \left(f_i - d_i \right)$



Minimiziranje L_{max} ne znači i minimalan broj poslova koji prekoračuju svoj *deadline*!

Funkcija vreme - upotrebljivost

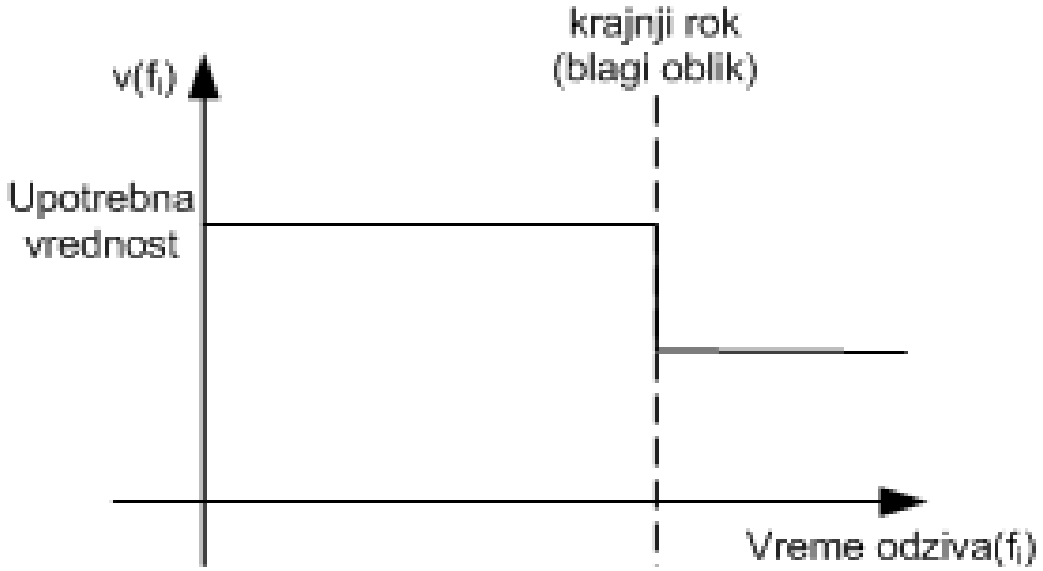
- ***Ukupna korist*** koja se dobija izvršavanjem nekog posla ne zavisi samo od ***samog značaja tog posla*** već i od vremena ***kada je on kompletiran***
- Neka je $v(f_i)$ funkcija odnosa ***vreme – upotrebljivost rezultata*** (koristi za sistem) pri obradi informacija u realnom vremenu – dakle, to je ***funkcija*** koja ***prikazuje upotrebljivost dobijenog rezultata*** ako je dostupan u trenutku f_i (završetka izvršavanja)

Funkcija vreme - upotrebljivost

- ***Prekidne tačke*** funkcije ***v*** – odnosno ***njeni izvodi*** prvog i drugog reda – označavaju vremenski zahtev (*deadline*)
- Jedan od naziva klase sistema koju analiziramo: ***sistemi sa vremenskim ograničenjem odziva*** (*deadline*)
- ***Klasifikacija sistema*** u toj klasi može biti izvršena ***saglasno funkciji v***

Funkcija vreme - upotrebljivost

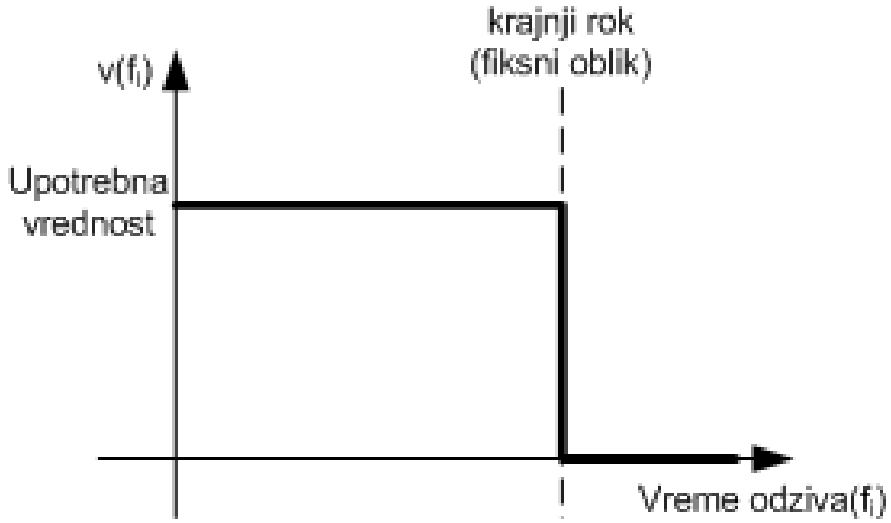
- Sistemi sa vremenskim ograničenjem odziva (meki, blaža varijanta, engl. ***soft***)



Primer: ***dekompresija filma***

Funkcija vreme - upotrebljivost

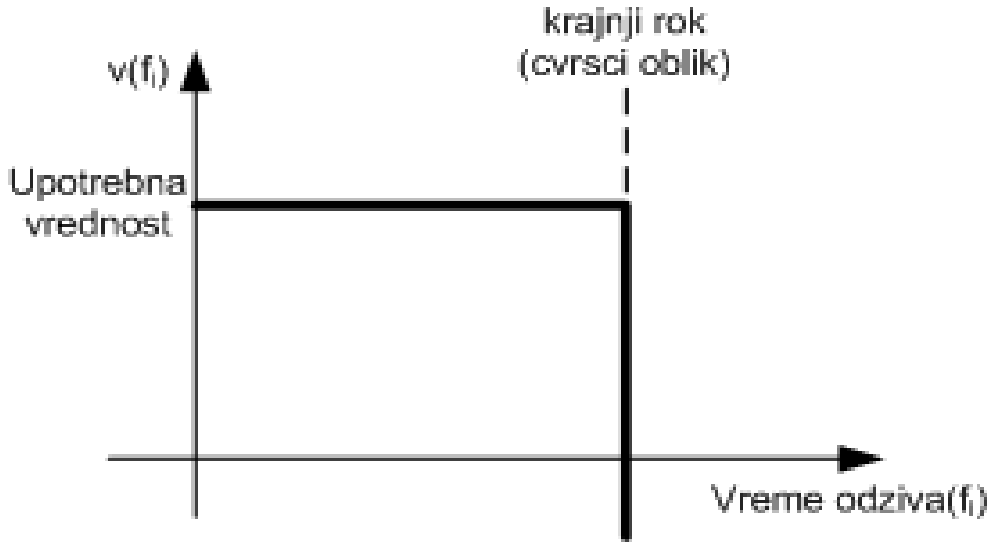
- Otporni (engl. ***firm***) sistemi
- ***Upotrebna vrednost*** rezultata dostupnog posle krajnjeg roka (*deadline*), ***jednaka je nuli***



Primer: procesor ***obrade govora*** u mobilnom telefonu .

Funkcija vreme - upotrebljivost

- Sistemi sa vremenskim ograničenjem odziva (čvrsta varijanta, engl. ***hard***)
- Odziv posle krajnjeg roka može biti ***sa katastrofalnim posledicama***



Funkcija vreme - upotrebljivost

- Ako definišemo ovakvu funkciju za svaki posao iz skupa
- ***Performansa algoritma raspoređivanja*** može biti merena i tzv. ***kumulativnom vrednošću***:

$$\text{Kumulativna\_vrednost} = \sum_{i=1}^{n} v(f_i)$$

Anomalije raspoređivanja

- ***„Povećanje snage procesiranja bezuslovno prouzrokuje poboljšanje performansi*** skupa poslova!“
- Slede pojedinačni primeri koji jasno ilustruju da ***ova tvrdnja nije tačna!***
- Načini povećanja snage procesiranja u sistemu:
  - povećanje broja procesora
  - smanjenjem vremena izvršavanja poslova
  - relaksiranje odnosa prvenstva
- Takve situacije – ***Richard-ove anomalije***

Anomalije raspoređivanja

- ***Teorema (Richard Graham, 1976)***

Ako je ***skup poslova*** ***optimalno raspoređen*** na više-procesorskom sistemu, sa fiksnim

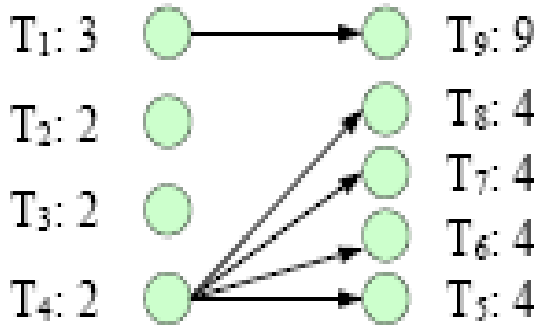
1. ***brojem*** procesora
2. ***vremenom*** izvršavanja
3. ***zahtevima prvenstva*** izvršavanja

tada,

povećanjem broja procesora, ***redukovanjem*** vremena izvršavanja ili odnosa prvenstva

može se povećati ukupno vreme kompletiranja (t_r)

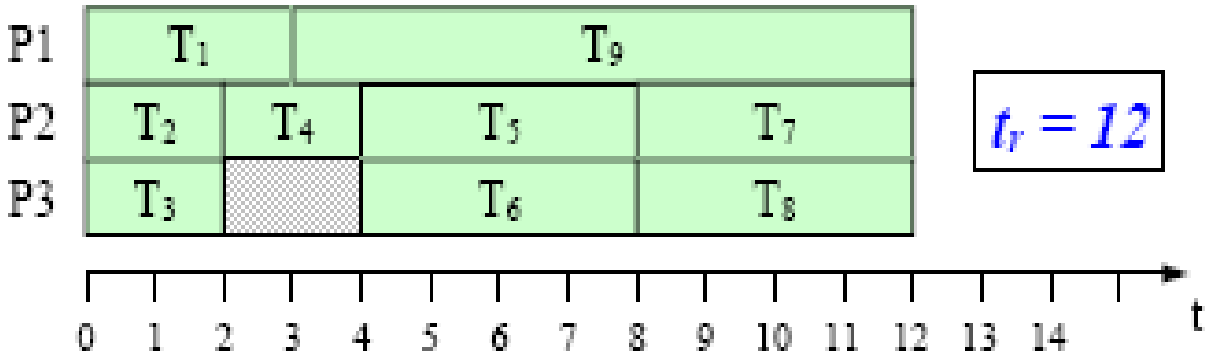
Anomalije raspoređivanja



priority

$P_i > P_j \quad \forall\, i < j$

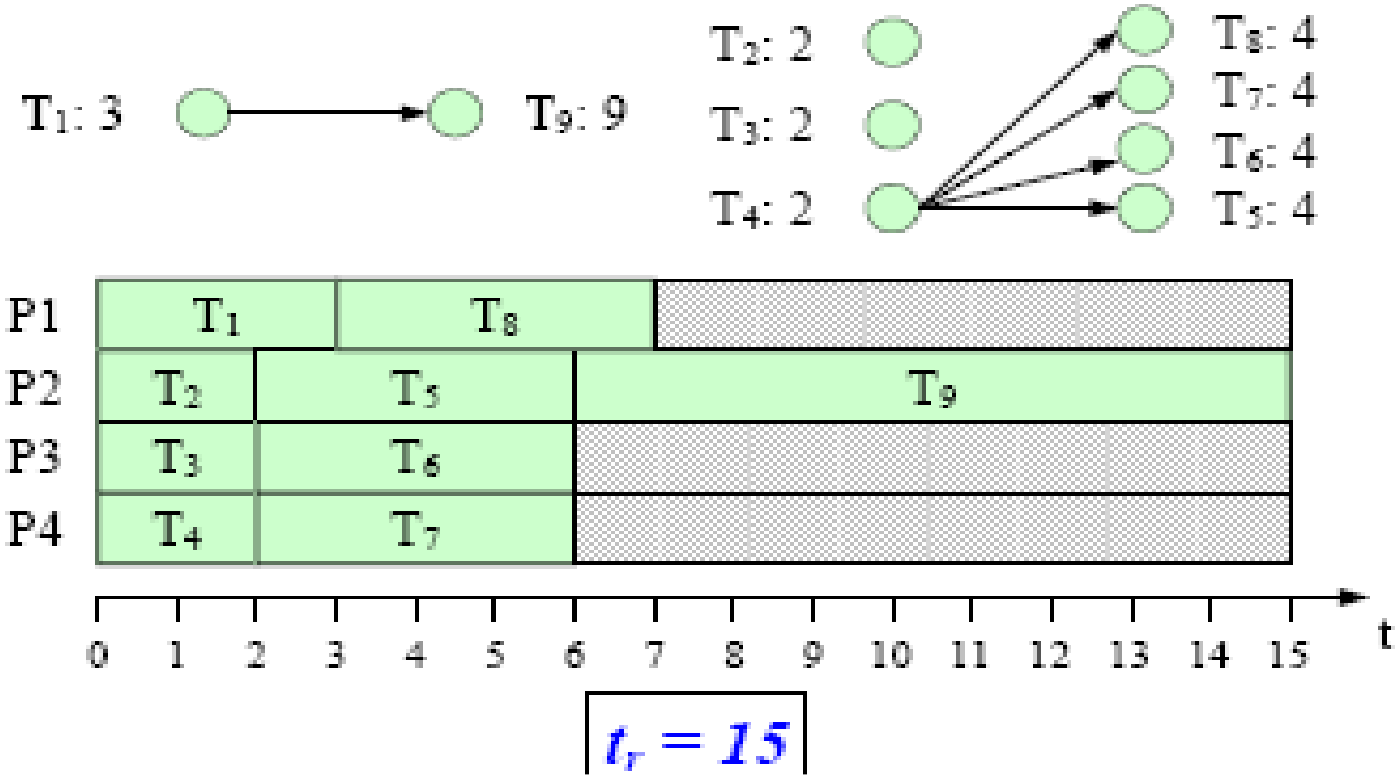
Optimalni raspored:



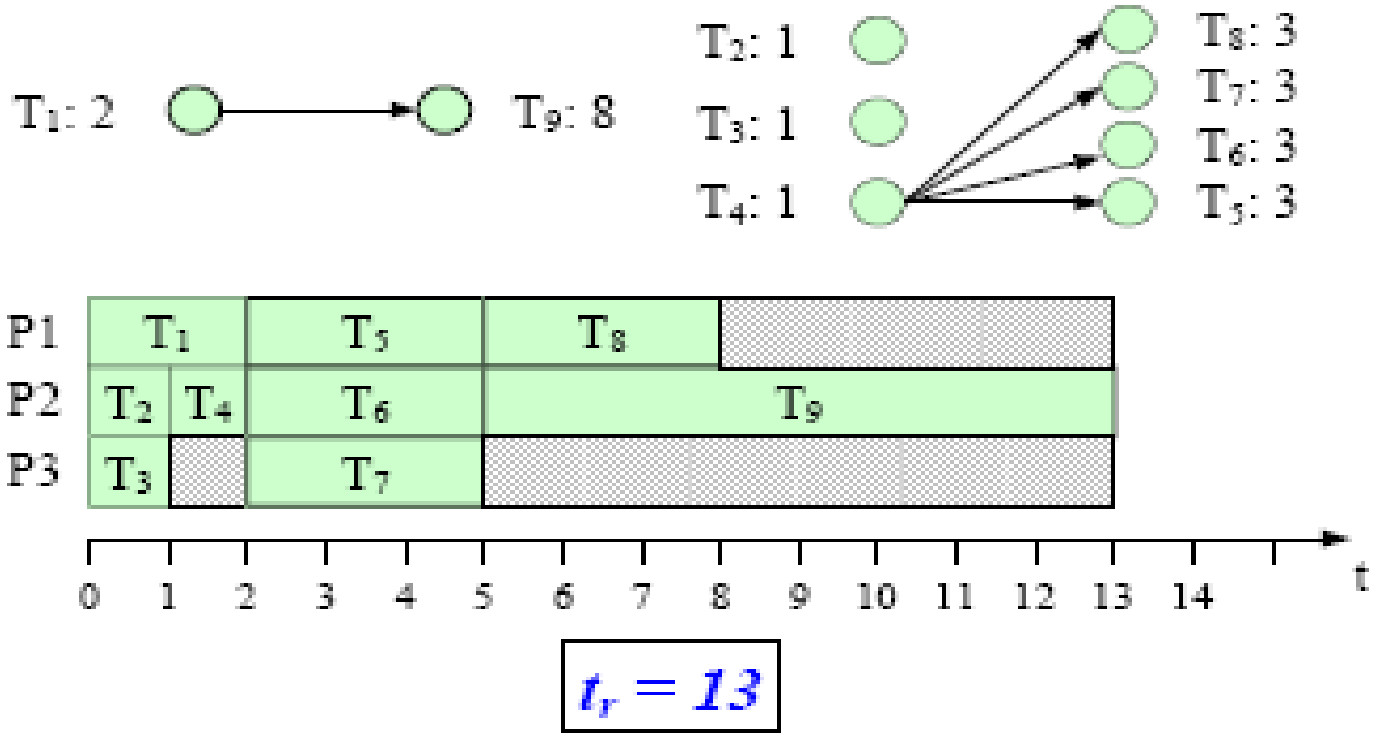
$t_r = 12$

Ukupno vreme kompletiranja - $t_r = \max_i(f_i) - \min_i(a_i)$

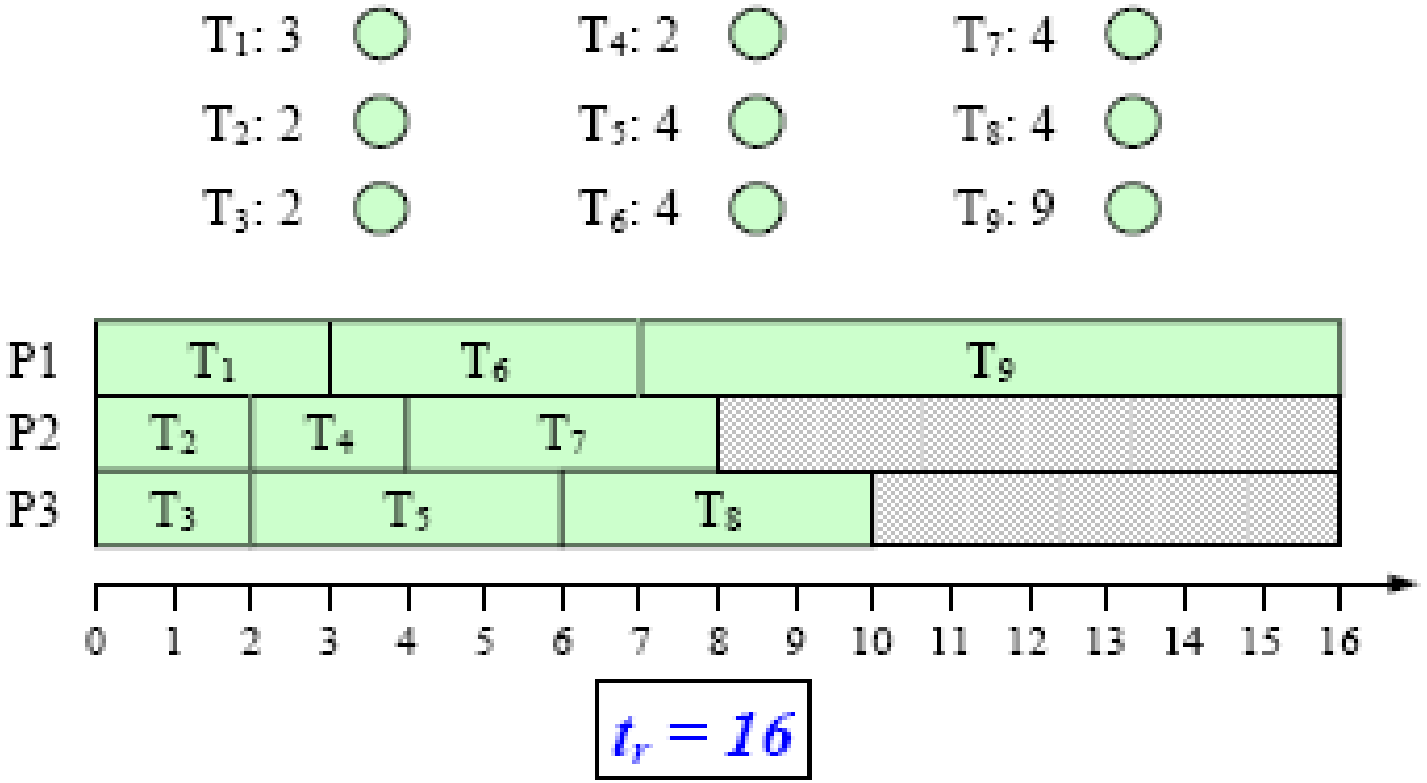
Povećanje broja procesora



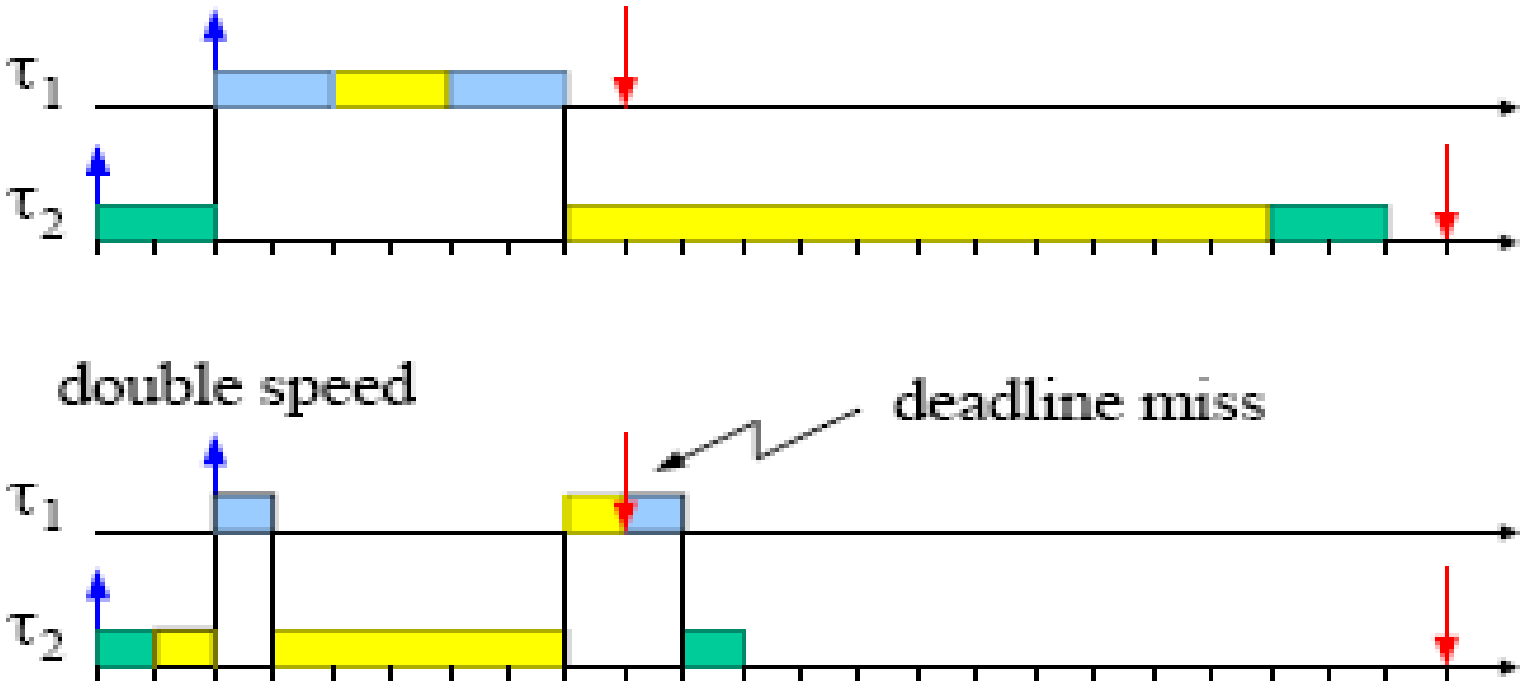
Manji poslovi



Odustajanje od ograničenja

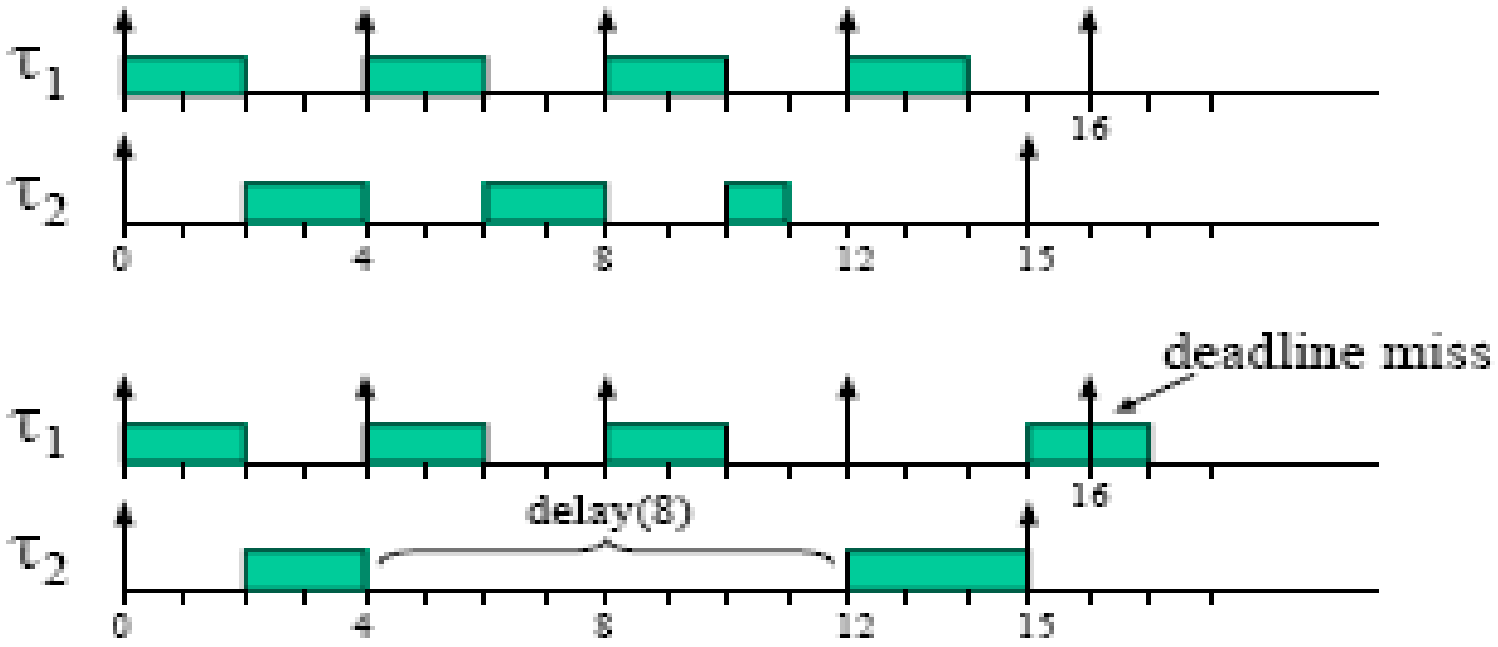


Brži procesor



Opasna operacija: DELAY

- ***delay(D)*** – komanda koja privremeno suspenduje izvršavanje programa na određeno vreme (D)
- Može prouzrokovati povećanje vremena odziva drugih poslova (prekoračenje *deadline*-a)



Šta smo naučili?

- ***Testiranje nije dovoljno*** za sisteme u realnom vremenu
- ***Intuitivna rešenja*** neće uvek raditi
- Operacija kašnjenja (***delay***) se ***ne sme koristiti*** u poslovima sa ograničenim vremenom odziva (poslovi realnog vremena)
- ***Bezbedan pristup***:
  - Koristi ***predvidljive*** mehanizme operativnog sistema
  - ***Analiziraj sistem da bi predvideo njegovo ponašanje***.

Ostvarenje predvidljivosti

- ***Za predvidljivo ponašanje sistema najodgovorniji je operativni sistem***
- ***Konkurentnost*** upravljanja se može sprovesti sa:
  - odgovarajućim ***algoritmima raspoređivanja***
  - odgovarajućim ***protokolima sinhronizacije***
  - efikasnim ***komunikacionim mehanizmima***
  - predvidljivim ***upravljanjem prekida***