

# Rozvrhovanie – 2.časť

- Operatívny manažment výrobného procesu
  - Plánovanie vs. rozvrhovanie
- Rozvrhovanie – hlavné a doplnkové charakteristiky, typy úloh
  - Rozvrh, optimálny rozvrh, používané kritériálne funkcie
  - Príklad úlohy rozvrhovania a tvorby rozvrhu (Ganttov diagram)
- **Typy rozvrhovacích úloh a ich riešenie**
  - 1. Rozvrhovanie na paralelných strojoch/procesoroch**
    - A. Rozvrhovanie na jednom stroji/procesore
      - i. S povoleným prerušením úloh
      - ii. Bez prerušenia úloh
    - B. Rozvrhovanie na viacerých strojoch/procesoroch
  - 2. Rozvrhovanie na špecializovaných (dedikovaných) strojoch**
    - A. Open shop
    - B. Flow shop
    - C. Job shop

# Typy rozvrhovacích úloh

## 1. Paralelné procesory (stroje)

- A. Rozvrhovanie **na jednom procesore** (jednostupňová výroba) – s prerušením, alebo bez prerušenia
- B. Rozvrhovanie **na viacerých procesoroch** (viacstupňová výroba)

## 2. Špecializované (dedikované) procesory (stroje)

- úlohy sa rozdeľujú do skupín, tzv. zákaziek:

$$J_k = [T_{1,k}, \dots, T_{n_k,k}]$$

- každá úloha v rámci zákazky  $J_k$  beží na inom stroji

- Rozlišujeme **3 základné typy týchto úloh** –

**A) open shop, B) flow shop a C) job shop**

(podrobnosti budú vysvetlené na budúcej prednáške)

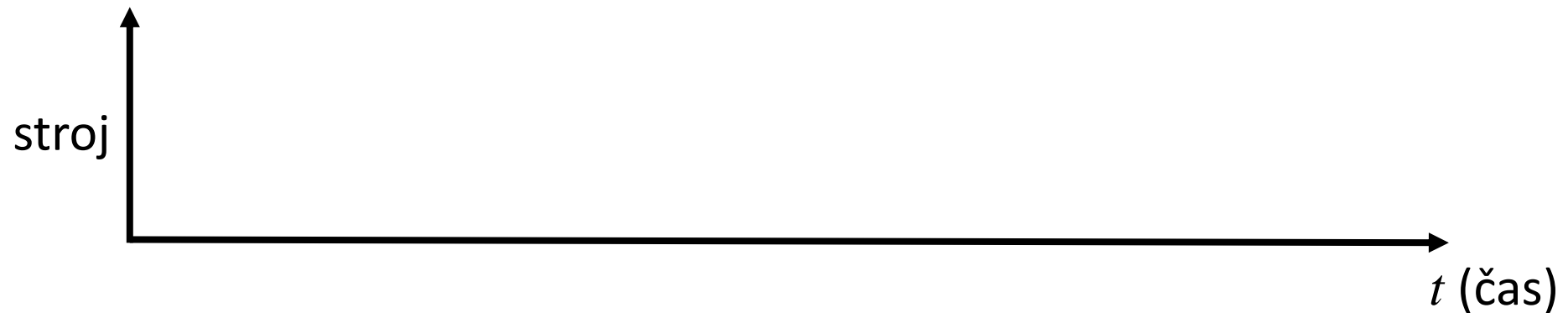
# 1. Rozvrhovanie na paralelných procesoroch (strojoch)

# A. Rozvrhovanie na jednom stroji/procesore – s prerušením

- JACKSONOV ALGORITMUS:
  - Máme  $n$  úloh, rôzne  $r_i$  a  $d_i$ . Potom algoritmus pre nájdenie optimálneho rozvrhu v zmysle kritéria  $L_{max}$  funguje takto:
    1. Vždy aktivuj úlohu s najskoršou dobou ukončenia ( $d_i$ ) spomedzi všetkých, ktoré sú aktuálne pripravené.
    2. Akonáhle začne byť úloha  $T_i$  pripravená a procesor je obsadený úlohou  $T_j$ , pozastav úlohu  $T_j$  v prospech úlohy  $T_i$  práve vtedy, ak čas ukončenia  $i$ -tej úlohy je skorší ako čas ukončenia  $j$ -tej úlohy, inak ponechaj bežať úlohu  $T_j$ .

# Príklad – jednostupňová výroba

Úloha	$t_i$	$r_i$	$d_i$
A	6	4	32
B	8	0	27
C	4	9	22
D	5	15	43
E	8	20	38
F	8	21	36



# Príklad – jednostupňová výroba

Úloha	$t_i$	$r_i$	$d_i$
A	6	4	32
B	8	0	27
C	4	9	22
D	5	15	43
E	8	20	38
F	8	21	36

- Vypočítajte rôzne typy kriteriálnych funkcií pre výsledný rozvrh (typu C, F, L, T)

$$c_i(R) = [ \quad , \quad , \quad , \quad , \quad , \quad ]$$

$$l_i(R) = [ \quad , \quad , \quad , \quad , \quad , \quad ]$$

$$t_i(R) = [ \quad , \quad , \quad , \quad , \quad , \quad ]$$

$$f_i(R) = [ \quad , \quad , \quad , \quad , \quad , \quad ]$$

$$C_{max}(R) =$$

$$C(R) =$$

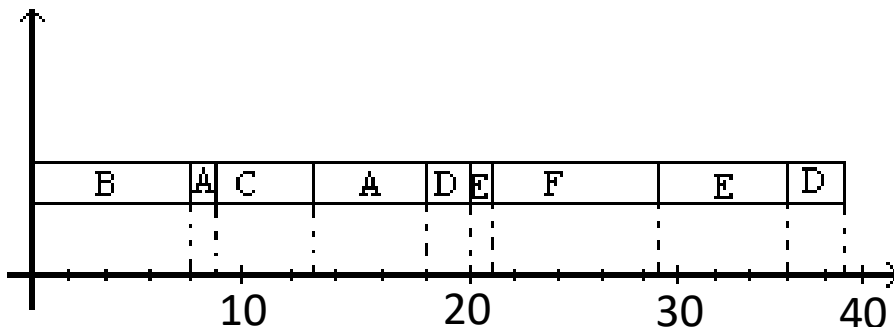
$$L_{max}(R) =$$

$$L(R) =$$

$$T_{max}(R) =$$

$$T(R) =$$

$$n_T(R) =$$



# A. Rozvrhovanie na jednom stroji/procesore – bez prerušenia (1)

- Zložitejšia úloha ako v prípade s prerušením, nakoľko ide o permutačnú úlohu ( $n!$  možných rozvrhov), ktorú až na špeciálne prípady nemožno riešiť v polynomiálnom čase.
- Niektoré špeciálne prípady:
  1. **Úlohy  $T_i$  ( $i = 1, \dots, n$ ),  $r_i = 0$  (pre všetky  $i = 1, \dots, n$ ), bez zadaných  $d_i$ , bez precedencií, bez priorít**
    - Z hľadiska kritériálnej funkcie  $C_{max}$  sú všetky rozvrhy rovnako dobré, takže ľubovoľné usporiadanie úloh je optimálne.
    - Z hľadiska kritériálnej funkcie  $C$  je optimálne usporiadanie úloh podľa neklesajúcej postupnosti ich dĺžok trvania, tj.:  $t_{(1)} \leq t_{(2)} \leq \dots \leq t_{(n)}$  čiže od najkratšej úlohy po najdlhšiu.

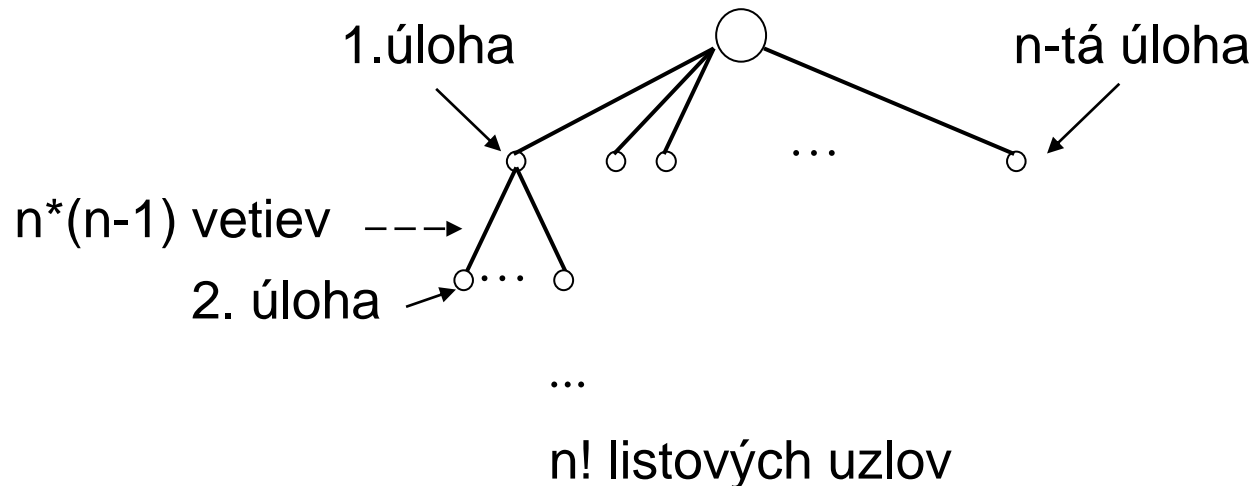
# A. Rozvrhovanie na jednom stroji/procesore – bez prerušenia (2)

2. Presne ako v predchádzajúcom prípade, ale s prioritami  $w_i$ 
  - Z hľadiska kritériálnej funkcie  $C_w$  je optimálne usporiadanie úloh podľa nerastúcej postupnosti ich priorít, tj.:  $w_{(1)} \geq w_{(2)} \geq \dots \geq w_{(n)}$  čiže od najvyššej priority po najnižšiu
3. Úlohy  $T_i$  ( $i = 1, \dots, n$ ),  $r_i = 0$  (pre všetky  $i = 1, \dots, n$ ), ale rôzne  $d_i$ , bez precedencií, bez priorít
  - Z hľadiska kritériálnej funkcie  $L_{max}$  existuje viacero heuristík, napr. Moorov algoritmus vychádza z neklesajúcej postupnosti požadovaných časov ukončenia úloh, t.j.  $d_{(1)} \leq d_{(2)} \leq \dots \leq d_{(n)}$



# A. Rozvrhovanie na jednom stroji/procesore – bez prerušenia (3)

- Všetky ostatné úlohy vedú na permutačné rozvrhy a je možné ich riešiť napríklad metódou vetvenia a medzí, ktorá sa snaží efektívne prehľadať nasledujúci priestor prehľadávania:



## B. Rozvrhovanie na viacerých paralelných procesoroch

- Úlohy sa najprv **usporiadajú podľa zvolenej heuristiky** a potom sa priradujú zaradom vždy na ten procesor, ktorý sa najskôr uvoľní. Pritom sa používajú rôzne heuristiky, napr.:
  - **LPT** (Longest Processing Time) - vyber úlohu s najdlhším trvaním ( $t_i$ )
  - **SPT** (Shortest Processing Time) - vyber úlohu s najkratším trvaním ( $t_i$ )
  - **EST** (Earliest Starting Time) - vyber úlohu s najskorším časom začiatku ( $r_i$ )
  - **LST** (Latest Starting Time) - vyber úlohu s najneskorším časom začiatku ( $r_i$ )
  - **EFT** (Earliest Finishing Time) - ... s najskorším časom ukončenia ( $d_i$ )
  - **LFT** (Latest Finishing Time) - ... s najneskorším časom ukončenia ( $d_i$ )
  - **MWR** (Most Work Remaining) - vyber úlohu s najdlhšou zvyškovou prácou (súčet trvaní úloh, ktoré ešte musia byť vykonané za vybranou úlohou)  
možno uplatniť ak sú zadané precedencie, resp. usporiadanie v zákazkách

# Príklad algoritmu s použitím heuristiky LPT

begin

vytvor usporiadaný zoznam úloh

Pri inej heuristike stačí zmeniť  
tento riadok (spôsob usporiadania)

pri LPT od najdlhšej po najkratšiu, t.j.:  $t_1 \geq t_2 \geq \dots \geq t_n$

for  $j = 1$  to  $m$   $S_j = 0$ ; % pre každý procesor inicializuj čas jeho dostupnosti na 0

$j := 1$  % počítadlo úloh  $j$  nastav na 1 (poradové číslo úlohy v zozname)

repeat % opakuj kým nespracuješ všetky úlohy v zozname

urči také  $k$ , že  $S_k = \min_{1 \leq i \leq m} \{S_i\}$  % vyber prvý voľný procesor

$S_k := S_k + t_j$ ; % prirad' úlohu  $T_j$  (v poradí  $j$ -ta v zozname) na procesor  $k$

$j := j + 1$ ; % zvýš počítadlo úloh  $j$  o 1 (posun na nasledujúcu úlohu)

until  $j = n$ ; % ak už boli spracované všetky úlohy v zozname, ukonči cyklus

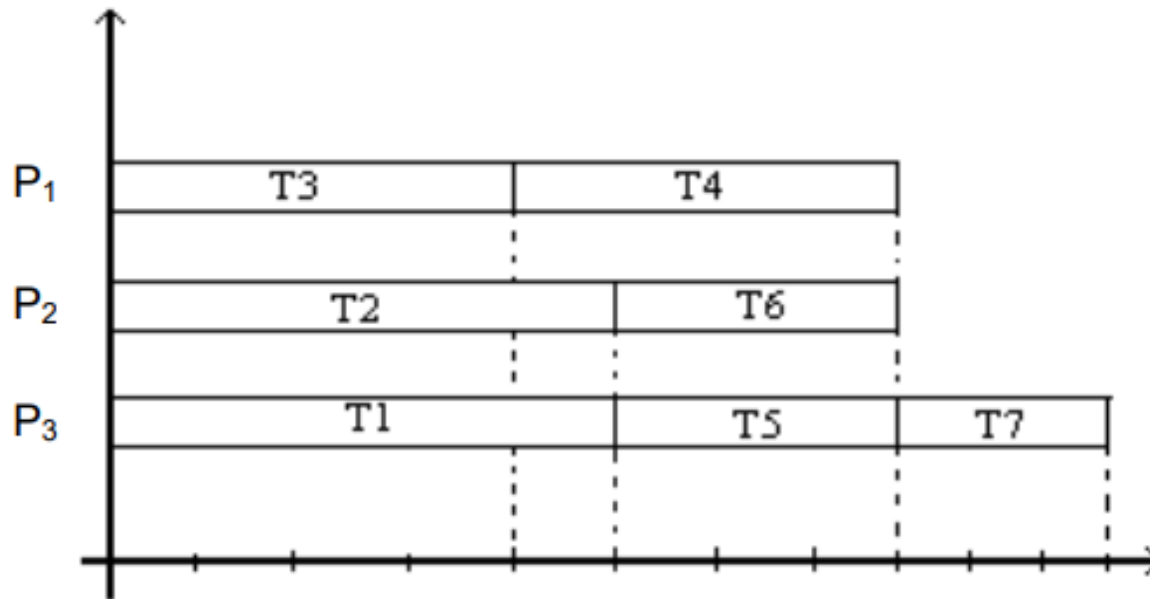
end;

# Príklad (1)

Majme 3 paralelné procesory, na ktorých je potrebné rozvrhnúť 7 úloh s takýmito dĺžkami trvania:

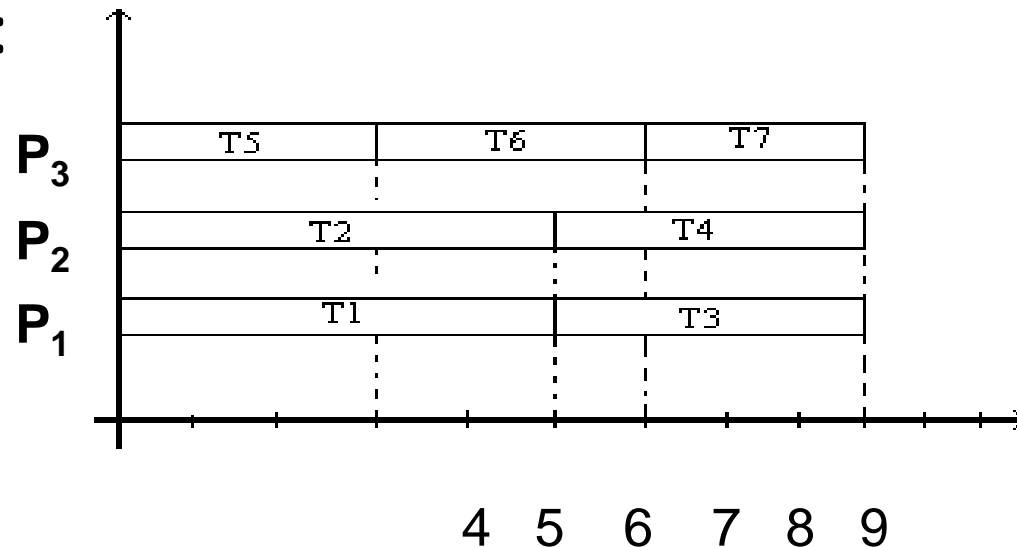
$t_1=5, t_2=5, t_3=4, t_4=4, t_5=3, t_6=3, t_7=3$ , pričom  $r_i=0 \forall i=1..7$

Pri tvorbe rozvrhu **použite heuristiku LPT**.



# Vzdialenosť LPT rozvrhov od optima

- Ale optimálny rozvrh v zmysle kritéria  $C_{\max}$  je nasledovný:

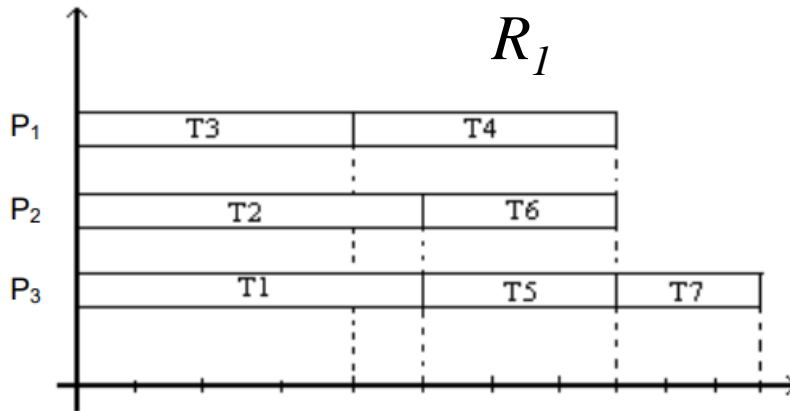


- Dá sa dokázať, že rozvrh vygenerovaný podľa LPT nie je voči optimálnemu rozvrhu podľa  $C_{\max}$  horší ako:

$$Q_{LPT} = \frac{4}{3} - \frac{1}{3m} \quad (m \text{ je počet strojov})$$

$$\text{t.j. pre } m = 3: \quad Q_{LPT} = \frac{4}{3} - \frac{1}{9} = \frac{12-1}{9} = \frac{11}{9}$$

# Príklad (2)



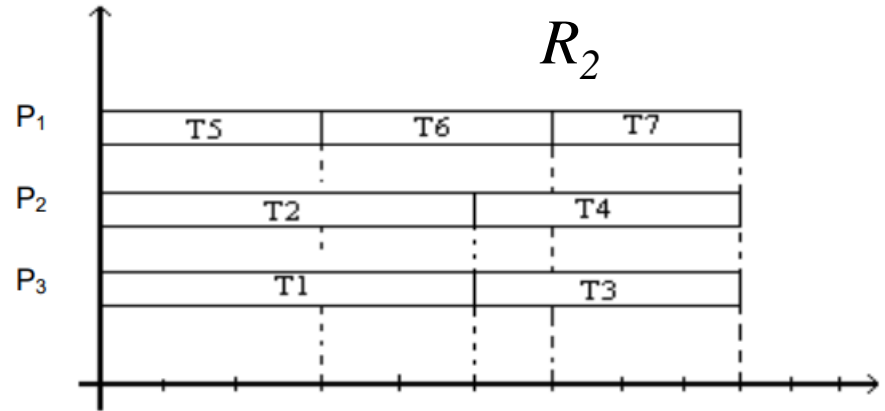
Hodnoty kritériálnych funkcií  
pre rozvrh podľa LPT ( $R_1$ ):

$$c_i(R) = [ \quad , \quad , \quad , \quad , \quad , \quad , \quad ]$$

$$C_{max}(R) =$$

$$C(R) =$$

$$\bar{C} =$$



Hodnoty kritériálnych funkcií  
pre optimálny rozvrh ( $R_2$ ):

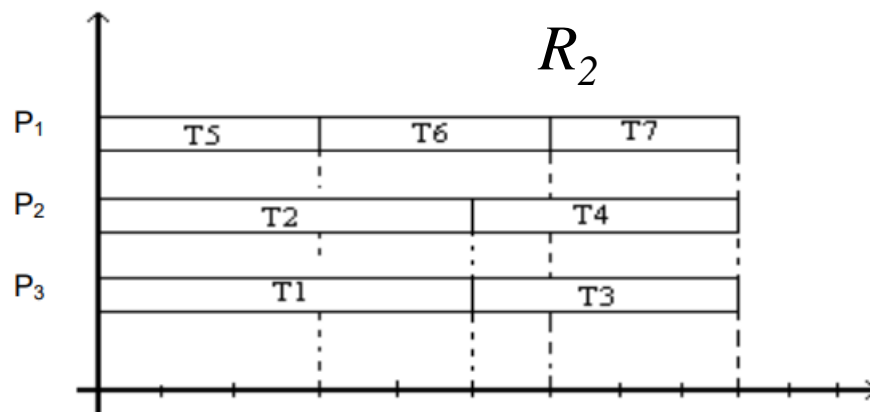
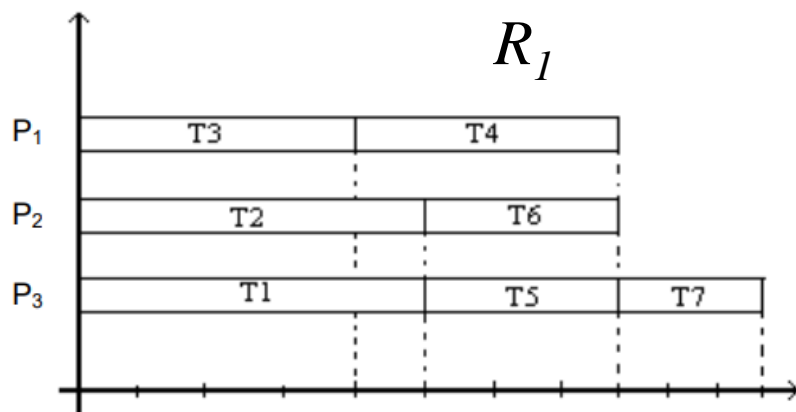
$$c_i(R) = [ \quad , \quad , \quad , \quad , \quad , \quad , \quad ]$$

$$C_{max}(R) =$$

$$C(R) =$$

$$\bar{C} =$$

# Príklad (2) – riešenie



Hodnoty kritériálnych funkcií  
pre rozvrh podľa LPT ( $R_1$ ):

$$c_i(R_1) = [5, 5, 4, 8, 8, 8, 11]$$

$$C_{max}(R_1) = 11$$

$$C(R_1) = 49$$

$$\bar{C}(R_1) = 7$$

Hodnoty kritériálnych funkcií  
pre optimálny rozvrh ( $R_2$ ):

$$c_i(R_2) = [5, 5, 9, 9, 3, 6, 9]$$

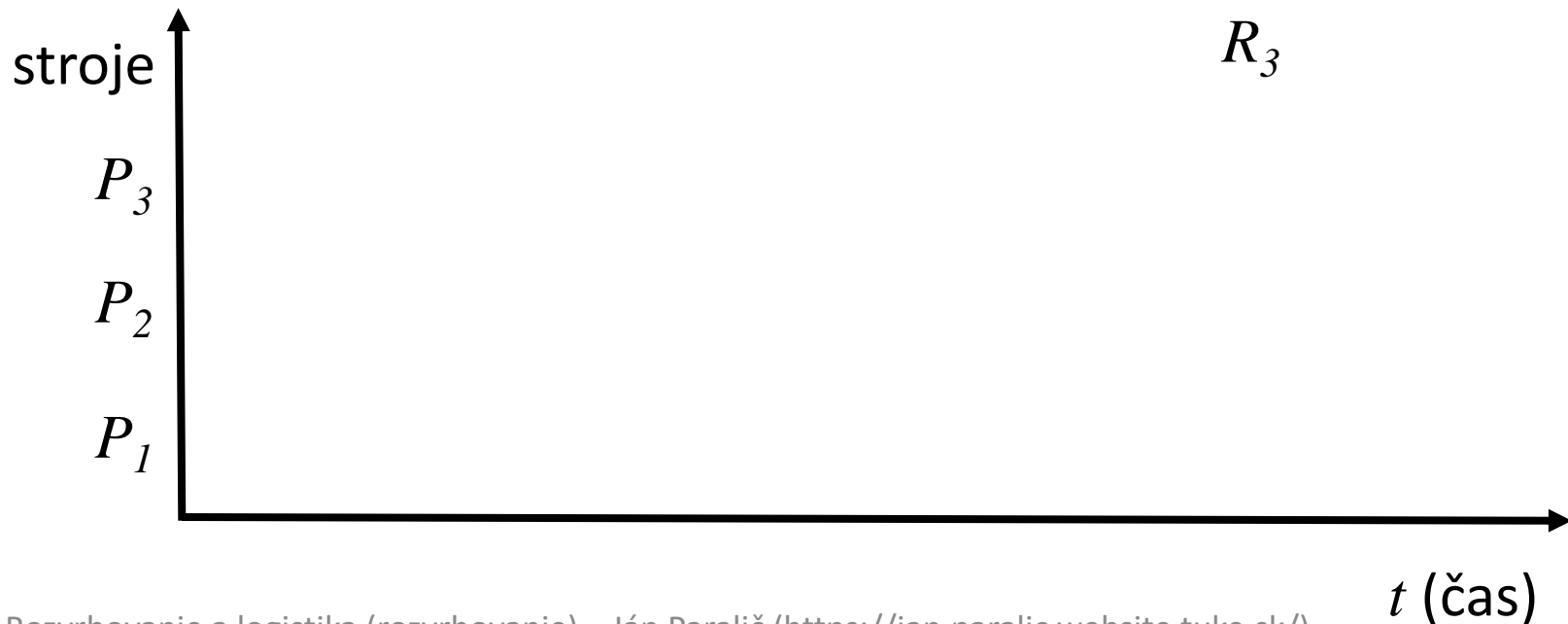
$$C_{max}(R_2) = 9$$

$$C(R_2) = 46$$

$$\bar{C}(R_2) = 6.8$$

## Príklad (3)

Pre rovnako zadané úlohy:  $t_1=5, t_2=5, t_3=4, t_4=4, t_5=3, t_6=3, t_7=3$ , pričom  $r_i=0 \forall i=1..7$  zostrojte **rozvrh podľa heuristiky SPT** a porovnajte ho podľa zvolených kritériálnych funkcií s LPT rozvrhom.





# Úloha zo 6. prednášky

1. Definujte si vlastnú úlohu rozvrhovania na jednom procesore, pričom je prípustné prerušenie úloh. Zadaťte si 7 úloh s nasledovnými údajmi pre všetky úlohy:
  - rôzne časy trvania úloh  $t_i$  (dĺžka úlohy),
  - rôzne (aj nenulové) časy pripravenosti do výroby  $r_i$  (release time),
  - rôzne časy kedy by mali byť úlohy hotové  $d_i$  (due date),
2. Zostrojte pre Vami zadanú úlohu rozvrh vo forme Ganttovho diagramu podľa Jacksonovho algoritmu.
3. Pre zostavený rozvrh vypočítajte hodnoty nasledovných kritériálnych funkcií:  $C, C_{max}, L, L_{max}, F, F_{max}, T, T_{max}, n_T$

*Termín vypracovania: 9.4.2024*