

PRECEDENČNÉ OHRANIČENIA

Príklad

Máme 3 identické paralelné procesory a daných 8 úloh s týmito parametrami:

$$P = \{P_1, P_2, P_3\}$$

$$T = \{T_1, T_2, \dots, T_8\}$$

$$\{T_1 < T_3, T_2 < T_3, T_2 < T_4, T_3 < T_6, T_3 < T_7, T_4 < T_6, T_4 < T_8, T_5 < T_8\} \text{ precedencie}$$

$$m = 3 \quad \text{paralelné procesory}$$

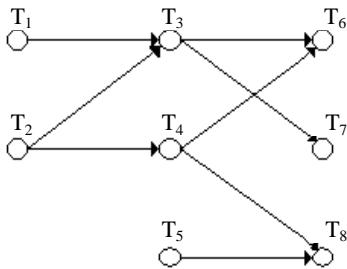
$$t_i = [3, 4, 1, 2, 1, 2, 3, 2] \quad \text{doba trvania úloh}$$

$$r_j = 0 \quad (j = 1, \dots, 8) \quad \text{časy pripravenosti}$$

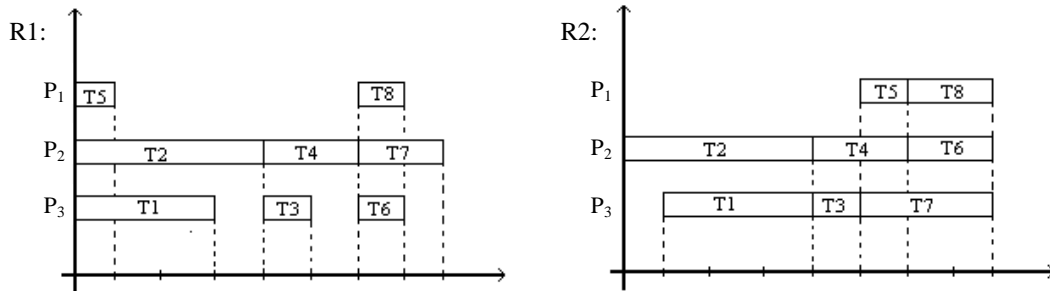
$$d_i = [5, 4, 5, 3, 7, 6, 9, 12] \quad \text{požadované časy ukončenia}$$

$$w_i = [1, 2, 1, 3, 1, 2, 2, 2] \quad \text{priority}$$

PRECEDENČNÝ GRAF:



GANTTOV DIAGRAM:



POROVNANIE:

	R_1	R_2
$C(R_i)$	[3, 4, 5, 6, 1, 8, 9, 8]	[4, 4, 5, 6, 6, 8, 8, 8]
l_i	[-2, 0, 0, 3, -6, 2, 0, -4]	[-1, 0, 0, 3, -1, 2, -1, -4]
C_{max}	9	8 *
\bar{C}	5,5 *	6,1
C	44 *	49
L_{max}	3 *	3 *
\bar{L}	-1 *	0,25
L	-7 *	-2
F_{max}	9	8 *
F	5,5 *	6,1
F	44 *	49
E	1,5 *	1
\bar{w}_T	2 *	2 *

I. PARALELNÉ PROCESORY

1) Rozvrhovanie na jednom procesore (Jednostupňová výroba)

a) 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.
- 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 j -tej úlohy, inak ponechaj bežať úlohu T_j .

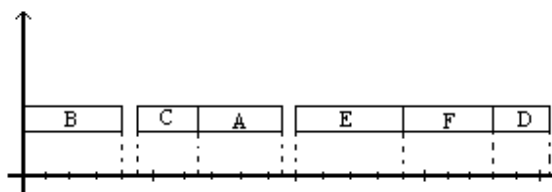
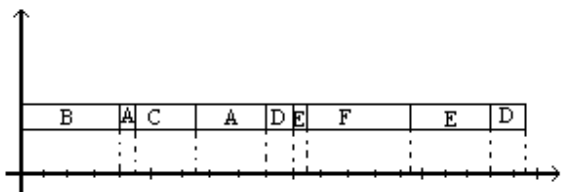
Príklad

R1:

Úloha	t_i	v_i	d_i	l_i	f_i
A	6	4	32	-14	14
B	8	0	27	-19	8
C	4	9	22	-9	4
D	5	15	43	-4	24
E	8	20	38	-2	16
F	8	21	36	-7	8
				$l_{\max} = -2$	$F = 12,3$

R2:

Úloha	f_i	
A	15	+1
B	8	-
C	4	-
D	26	+2
E	8	-8
F	15	+7



b) Bez prerušenia - zložitejšia úloha, nakoľko ide o permutačnú úlohu ($n!$ možných rozvrhov), ktorú až na špeciálne prípady nemožno riešiť v polynomiálnom čase. Niektoré špeciálne prípady nasledujú:

1. - T_i - úloh ($i = 1, \dots, n$)
 - $r_i = 0$ ($i = 1, \dots, n$)
 - bez precedencií
 - bez priorít
 - bez d_i

Z hľadiska kritériálnej funkcie C_{\max} sú všetky rozvrhy sú rovnako dobré

2. - T_i - úloh ($i = 1, \dots, n$)
 - $r_i = 0$ ($i = 1, \dots, n$)
 - bez precedencií
 - s prioritami w_i
 - bez d_i

Z hľadiska kritériálnej funkcie C_{\max} je optimálne usporiadanie úloh podľa nerastúcej postupnosti ich priorit, tj.: $w_{(1)} \geq w_{(2)} \geq \dots \geq w_{(n)}$

3. - T_i – úloh ($i = 1, \dots, n$)
 - $r_i = 0$ ($i = 1, \dots, n$)
 - bez precedencií
 - bez priorit
 - $d_i =$ rôzne

Z hľadiska kritériálnej funkcie L_{\max} (ale aj kritériálnych funkcií T a n_T) je optimálne usporiadanie úloh podľa neklesajúcej postupnosti ich požadovaných časov ukončenia, tj.:

$$d_{(1)} \leq d_{(2)} \leq \dots \leq d_{(n)}$$

4. - T_i – úloh ($i = 1, \dots, n$)
 - $r_i = 0$ ($i = 1, \dots, n$)
 - bez precedencií
 - bez priorit
 - bez d_i

Z hľadiska kritériálnej funkcie F_{\max} je optimálne usporiadanie úloh podľa neklesajúcej postupnosti ich dĺžok trvania, tj.: $t_{(1)} \leq t_{(2)} \leq \dots \leq t_{(n)}$

5. - T_i – úloh ($i = 1, \dots, n$)
 - $r_i = 0$ ($i = 1, \dots, n$)
 - bez precedencií
 - s prioritami w_i
 - bez d_i

Z hľadiska kritériálnej funkcie F_w je optimálne usporiadanie úloh podľa neklesajúcej postupnosti $w_{(1)}/t_{(1)} \geq w_{(2)}/t_{(2)} \geq \dots \geq w_{(n)}/t_{(n)}$

6. 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:

