

Systemy wbudowane - wykład do samodzielnego
opracowania
Systemy czasu rzeczywistego (3)

Przemek Błaśkiewicz

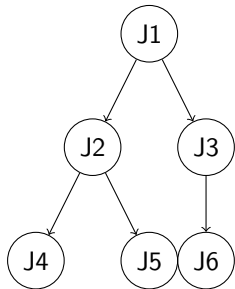
13 maja 2020

Planowanie zadań z wymaganą kolejnością wykonania

W systemach pojawiają się sekwencje zadań, które należy wykonać w określonej kolejności (np. przygotowanie danych → raportowanie).

- Planowanie bez wywłaszczenia, z asynchronicznymi pojawieniami się zadań, czasami wykonania i wymaganą kolejnością jest **NP-trudne**.
- Dla pewnych rozluźnień problemu można znaleźć optymalne rozwiązania...
- Np. gdy wszystkie zadania są dostępne od razu ($a_i = 0$):

	J_1	J_2	J_3	J_4	J_5	J_6
a_i	0	0	0	0	0	0
C_i	1	1	1	1	1	1
d_i	2	5	4	3	5	6



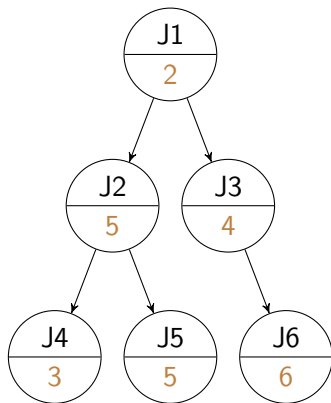
planowanie realizowane jest za pomocą LDF.

Założenie jest podobne jak przy EDF, ale organizacja zadań w drzewie pozwala na wzięcie pod uwagę wymogu ich następstw:

- ① utrzymuj zadania do planowania w grafie G , do wykonania na liście P ;
- ② spośród wszystkich liści G , wybierz zadanie J o najpóźniejszym terminie wykonania (deadline);
- ③ wstaw J na początek P i usuń je z G
- ④ wykonuj ponownie od 2, dopóki są zadania w drzewie

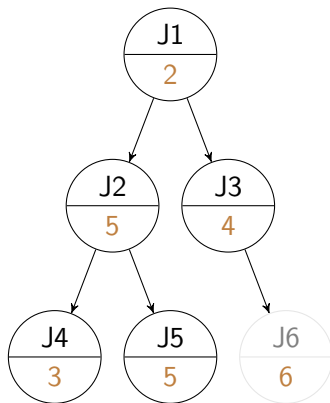
	J_1	J_2	J_3	J_4	J_5	J_6
a_i	0	0	0	0	0	0
C_i	1	1	1	1	1	1
d_i	2	5	4	3	5	6

$$P = ()$$



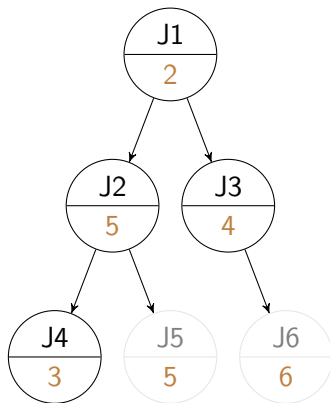
	J_1	J_2	J_3	J_4	J_5	J_6
a_i	0	0	0	0	0	0
C_i	1	1	1	1	1	1
d_i	2	5	4	3	5	6

$$P = (J_6)$$



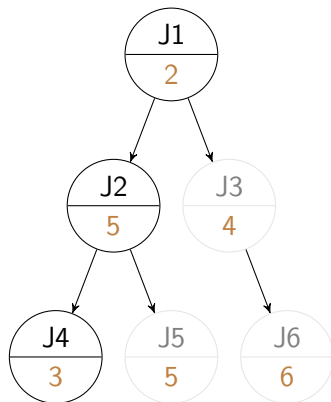
	J_1	J_2	J_3	J_4	J_5	J_6
a_i	0	0	0	0	0	0
C_i	1	1	1	1	1	1
d_i	2	5	4	3	5	6

$$P = (J_5, J_6)$$



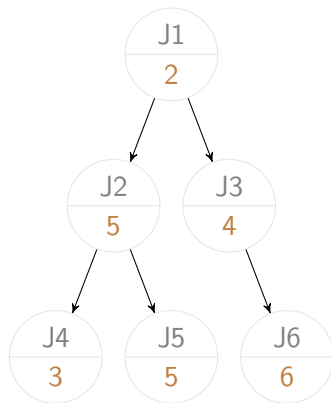
	J_1	J_2	J_3	J_4	J_5	J_6
a_i	0	0	0	0	0	0
C_i	1	1	1	1	1	1
d_i	2	5	4	3	5	6

... ♠



	J_1	J_2	J_3	J_4	J_5	J_6
a_i	0	0	0	0	0	0
C_i	1	1	1	1	1	1
d_i	2	5	4	3	5	6

$$P = (J_1, J_2, J_4, J_3, J_5, J_6)$$

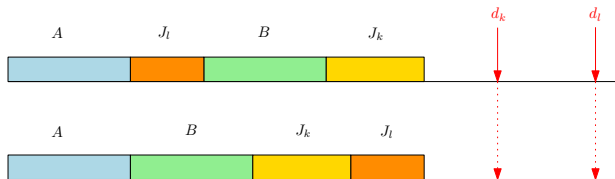


Lista P jest zachowywana jako statyczny plan dla zadań $J_{1...6}$.

Lawler '73

LDF jest optymalny ze względu na minimalizację maksymalnego opóźnienia.

- dane są zadania $(J_1, \dots, J_n)$
- niech J_l nie ma zadań “poniżej” siebie w drzewie zależności
- niech J_l ma najpóźniejszy termin wykonania



Dowód (szkic): ♠

- 1 przesun J_l na koniec kolejki wykonania (bo $d_l > d_k$);
- 2 pokaż, że kolejność wykonania jest zachowana
- 3 pokaż, że maksymalne opóźnienie nie zwiększa się

- działa jedynie jako algorytm off-line
- złożoność:
 - szukanie liści w grafie G – $\mathcal{O}(|E|)$
 - utrzymywanie posortowanej (wg. d_i) listy liści – $\mathcal{O}(\log n)$
 - ogółem: $\mathcal{O}(n * \max(|E|, \log n))$

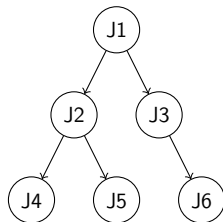
- Planowanie bez wyłączenia, z asynchronicznymi pojawieniami się zadań, czasami wykonania i wymaganą kolejnością jest **NP-trudne**.
- Dla pewnych rozluźnień problemu można znaleźć optymalne rozwiązania...
 - wszystkie zadania są dostępne od razu – LDF
 - dla schematów wyłączeniowych – modyfikowany EDF

Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie ($\exists i, j : a_i \neq a_j$), sprowadzamy najpierw ten problem to przypadku bez wyłączeń (przypomnij sobie lemat o wyłączeniu):

- ① zaktualizuj czasy nadejścia a_i :
 - każde zadanie w “korzeniu”: $a_i^* = a_i$
 - dla każdego zadania, którego czas nadejścia “rodzica” J_r został zaktualizowany: $a_i^* = \max\{a_i, a_r^* + C_r\}$
- ② zaktualizuj terminy wykonania:
 - dla każdego zadania w “liściu”: $d_i^* = d_i$
 - dla każdego zadania J_i , którego “potomkowie” J_p zostali zaktualizowani: $d_i^* = \min\{d_i, d_p^* - C_p\}$

	J_1	J_2	J_3	J_4	J_5	J_6
a_i	1	2	3	3	3	4
C_i	1	2	1	1	1	3
d_i	2	5	4	6	5	7

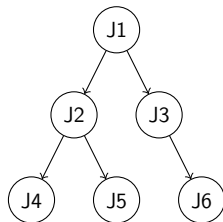
Dane początkowe, przed modyfikacjami



	J_1	J_2	J_3	J_4	J_5	J_6
a_i	1	2	3	3	3	4
a_i^*	1	2	3	4	4	4
C_i	1	2	1	1	1	3
d_i	2	5	4	6	5	7

Przelicz czasy nadejścia, zacznij od korzenia:

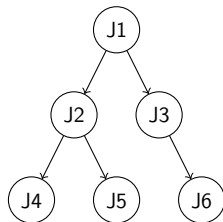
- $a_1^* = a_1 = 1$
- $a_2^* = \max\{a_2, a_1^* + C_1\} = 2$
- $a_3^* = \max\{a_3, a_1^* + C_1\} = 3$
- $a_4^* = \max\{a_4, a_2^* + C_2\} = 4$
- $a_5^* = \max\{a_5, a_2^* + C_2\} = 4$
- $a_6^* = \max\{a_6, a_3^* + C_3\} = 4$



	J_1	J_2	J_3	J_4	J_5	J_6
a_i	1	2	3	3	3	4
a_i^*	1	2	3	4	4	4
C_i	1	2	1	1	1	3
d_i	2	5	4	6	5	7
d_i^*	0	4	4	6	5	7

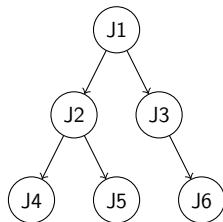
Przelicz terminy, zacznij od liści:

- $d_6^* = d_6 = 7$
- $d_5^* = d_5 = 5$
- $d_4^* = d_4 = 6$
- $d_3^* = \min\{d_3, d_6^* - C_6\} = 4$
- $d_2^* = \min\{d_2, d_5^* - C_5, d_4^* - C_4\} = 4$
- $d_1^* = \min\{d_1, d_2^* - C_2, d_3^* - C_3\} = 0$



Zmodyfikowany EDF – przykład ♠

	J_1	J_2	J_3	J_4	J_5	J_6
a_i	1	2	3	3	3	4
a_i^*	1	2	3	4	4	4
C_i	1	2	1	1	1	3
d_i	2	5	4	6	5	7
d_i^*	0	4	4	6	5	7



Termin wykonania zadania 1. wypada zanim cokolwiek się zacznie...

♠ Jak zmodyfikować zadania, by plan był wykonalny?

Przechodniość planowalności

Zbiór zadań jest planowalny z zachowaniem wymogów następstwa wykonania zadań wtedy i tylko wtedy, gdy zbiór zadań o zmodyfikowanych parametrach (a_i^*, d_i^*) jest planowalny przez EDF.

① Dla każdego planu σ zachowującego zależności musi być:

- $s_i \geq \max\{a_i, a_r^* + C_r\}$
- $f_i \leq \min\{d_i, d_p^* - C_p\}$

② Jeśli zadania $J_i \prec J_j$ planowalne są przez EDF, to:

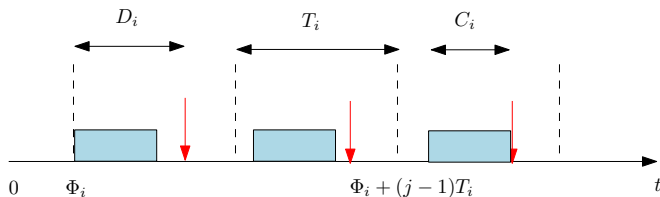
- $a_i^* < a_j^*$: J_i pojawiło się wcześniej niż J_j ;
- $d_i^* < d_j^*$: J_i ma wcześniejszy termin wykonania;

zatem:

- J_j nie może się rozpocząć przed J_i , oraz J_j nie wywłączy J_i ;
- $a_i^* \geq a_i$ oraz $d_i^* \leq d_i$



Planowanie zadań okresowych



Każde zadanie okresowe t_i określone jest przez:

- fazę Φ_i – pierwszy moment wystąpienia;
- okres T_i – czas jaki upływa między dwoma kolejnymi wystąpieniami;
- względny termin wykonania (deadline) D_i – okres czasu przeznaczony na wykonanie zadania od momentu jego wystąpienia;
- czas wykonania C_i – czas trwania obliczeń zadania.

Plan realizowalny składa się z czasów startu $s_{i,j}$ i ukończenia $f_{i,j}$ dla każdego zadania i w każdej jego instancji (pojawieniu się) j .

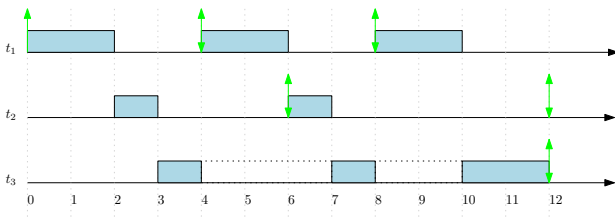
(Liu, Layland, 1973):

- każde zadanie otrzymuje priorytet $1/T_i$ (proporcjonalny do częstotliwości pojawiania się);
- wykonywane jest gotowe zadanie z aktualnie najwyższym priorytetem;
- system z wywłaszczeniami;

RM – przykład (1)

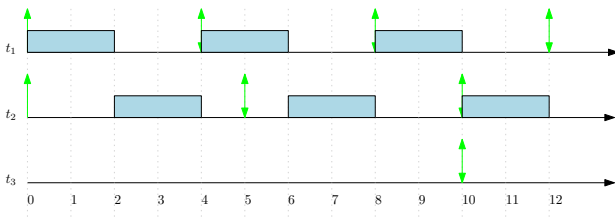
	t_1	t_2	t_3
Φ_i	0	0	0
T_i	4	6	12
C_i	2	1	4
D_i	4	6	12

Policzmy: $C_1/T_1 + C_2/T_2 + C_3/T_3 = 2/4 + 1/6 + 4/12 = 1$.



	t_1	t_2	t_3
Φ_i	0	0	0
T_i	4	5	10
C_i	2	2	1
D_i	4	5	10

Policzmy: $C_1/T_1 + C_2/T_2 + C_3/T_3 = 2/4 + 2/5 + 1/10 = 1$.



Stopień wykorzystania procesora

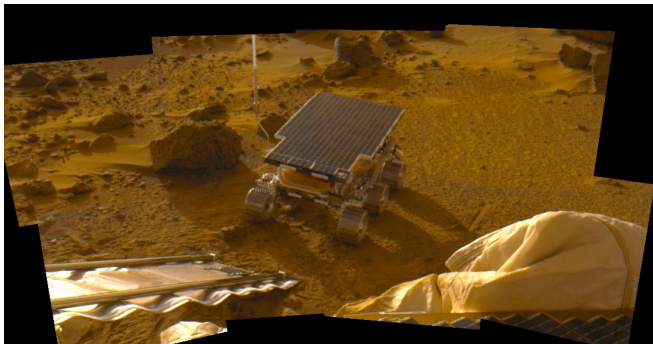
Dla danego zbioru n zadań okresowych, stopień wykorzystania procesora to:

$$U = \sum_{i=1}^n \frac{C_i}{T_i}.$$

- warunek konieczny dla stworzenia realizowalnego planu: $U \leq 1$
- dla algorytmu RM w pesymistycznym wypadku, dla $U \geq 2 \cdot (2^{1/n} - 1)$ brak gwarancji poprawnego rozwiązania.

To nie koniec...

Jak widzieliśmy, większość z tych algorytmów planowania powstała w okolicach połowy XX w. Związane jest to m.in. z dynamicznym rozwojem komputerów i potrzebą efektywnego wykorzystywania ich zasobów: minimalizacji przestojów podczas przeładowywania programów (czyli przekładania kart perforowanych w czytniku...). Oczywiście od tego czasu sporo poszło naprzód, ale ...



Na następnym wykładzie zobaczymy, że nawet w misji na Marsa planowanie zadań zmroziło ludziom krew.