Systemy wbudowane - wykład 11

Przemek Błaśkiewicz

17 maja 2018

1/94

Notes

Systemy czasu rzeczywistego - powtórka

System czasu rzeczywistego

Real-time system to system, którego poprawność działania zależy nie tylko od poprawności rezultatów (obliczeń, decyzji, akcji), ale również od czasu, kiedy zostaną one wypracowane (czas reakcji).

2/94

RTOS - planowanie zadań - powtórka

Plan

Plan (schedule) dla zbioru zadań (J_1,J_2,\ldots,J_n) to pewna funkcja: $\sigma:R^+\to\{0,\ldots,n\}$, taka, że:

 $\forall t \in R^+, \exists t_1, t_2 \in R^+ : t \in [t_1, t_2), \forall t' \in [t_1, t_2) : \sigma(t) = \sigma(t').$

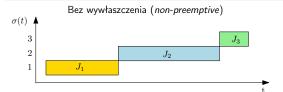
3/94

RTOS - planowanie zadań - powtórka

Plan

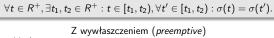
Plan (schedule) dla zbioru zadań (J_1,J_2,\ldots,J_n) to pewna funkcja: $\sigma:R^+\to\{0,\ldots,n\}$, taka, że:

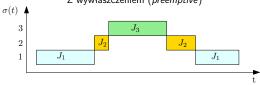
 $\forall t \in R^+, \exists t_1, t_2 \in R^+ : t \in [t_1, t_2), \forall t' \in [t_1, t_2) : \sigma(t) = \sigma(t').$



	_
lotes	
lotes	
	_
	_
lotes	
iotes	

Plan Plan (schedule) dla zbioru zadań (J_1,J_2,\ldots,J_n) to pewna funkcja: $\sigma:R^+\to\{0,\ldots,n\}$, taka, że:

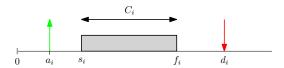




5/94

Notes

Cechy charakterystyczne zadań - powtórka



6 / 94

Planiści - powtórka

dla zadań aperiodycznych, synchronicznych – EDD

7 / 94

Planiści - powtórka

- o dla zadań aperiodycznych, synchronicznych EDD
- dla zadań aperiodycznych asynchronicznych EDF

otes	
	_
otes	
	—
otes	
	_

- dla zadań aperiodycznych, synchronicznych EDD
- dla zadań aperiodycznych asynchronicznych EDF
- j.w. + bez wywłaszczeń EDF + "idle CPU"

9 / 94

EDF bez wywłaszczeń - powtórka

10/94

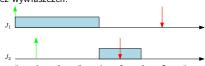
EDF bez wywłaszczeń - powtórka

	J_1	J_2
aį	0	1
Ci	4	2
di	7	5

EDF bez wywłaszczeń - powtórka

	J_1	J_2
aį	0	1
Ci	4	2
di	7	5

EDF bez wywłaszczeń:



N	otes

Notes

Notes			

EDF bez wywłaszczeń - powtórka Notes 0 1 4 2 7 EDF bez wywłaszczeń: optymalnie: EDF bez wywłaszczeń - powtórka Notes Jeffay i in. 1991 Przy braku zezwolenia na okresy bezczynności przy gotowym do wykonania zadaniu, EDF jest optymalny również dla modelu bez 14/94 Planiści - powtórka Notes o dla zadań aperiodycznych, synchronicznych – EDD • dla zadań aperiodycznych asynchronicznych – EDF o j.w. + bez wywłaszczeń – EDF + "idle CPU" 15 / 94 Planiści - powtórka Notes • dla zadań aperiodycznych, synchronicznych – EDD • dla zadań aperiodycznych asynchronicznych – EDF

j.w. + bez wywłaszczeń - EDF + "idle CPU"
 dla zadań aperdiodycznych, asynchronicznych, bez wywłaszczenia, "idle time" - algorytm Bratley'a

Alg

Algorytm Bartley'a – powtórka	
	Notes
• Bazuje na <i>odpowiedniej</i> permutacji zestawu zadań J_1, \ldots, J_n .	
Branch: wybierz do listy kolejne zadanie (z pozostałych)	
Bound:	
f a znaleziono realizowalny plan $ o$ bieżąca lista jest realizowalnym planem;	
 jeśli istnieje zadanie, którego termin minął, a nie zostało 	
jeszcze zaplanowane $ ightarrow$ bieżąca ścieżka to nierealizowalny plan (cofnij do ostatniego dokonanego wyboru).	
17/94	
Algorytm Bartley'a – powtórka	
	Notes
• Bazuje na <i>odpowiedniej</i> permutacji zestawu zadań J_1, \ldots, J_n .	
 Branch: wybierz do listy kolejne zadanie (z pozostałych) Bound: 	
lacksquare znaleziono realizowalny plan $ ightarrow$ bieżąca lista jest realizowalnym	
planem; ② jeśli istnieje zadanie, którego termin minął, a nie zostało	
jeszcze zaplanowane $ ightarrow$ bieżąca ścieżka to nierealizowalny plan (cofnij do ostatniego dokonanego wyboru).	
Złożoność wykładnicza – tylko jako algorytm offline;	
18 / 94	
Planowanie zadań z wymaganą kolejnością wykonania	
Trianowanie zadan z wymaganą kolejnością wykonania	Notes
 Planowanie bez wywłaszczenia, z asynchronicznymi pojawieniami się zadań, czasami wykonania i wymaganą 	
kolejnością jest NP-trudne.	
19/94	
7,	
Planowanie zadań z wymaganą kolejnością wykonania	Notes
	NOTES

Pla

- Planowanie bez wywłaszczenia, z asynchronicznymi pojawieniami się zadań, czasami wykonania i wymaganą kolejnością jest NP-trudne.
- Dla pewnych rozluźnień problemu można znaleźć optymalne rozwiązania...

	Notes
Planowanie bez wywłaszczenia, z asynchronicznymi	
pojawieniami się zadań, czasami wykonania i wymaganą kolejnością jest NP-trudne .	
Dla pewnych rozluźnień problemu można znaleźć optymalne	
rozwiązania ● gdy wszystkie zadania są dostępne od razu – LDF	
21/94	
LDF – latest deadline first	Notes
 utrzymuj zadania do planowania w grafie G, do wykonania na 	
liście P;	
22/94	
LDF – latest deadline first	N
	Notes
$oxed{a}$ utrzymuj zadania do planowania w grafie G , do wykonania na	
liście P;	
 spośród wszystkich liści G, wybierz zadanie J o najpóźniejszym terminie wykonania (deadline); 	
termine Hyrorama (deadine),	
23/94	
23/94	
LDF – latest deadline first	
	Notes
	Notes
LDF – latest deadline first	Notes
LDF – latest deadline first utrzymuj zadania do planowania w grafie G, do wykonania na	Notes
 LDF – latest deadline first a utrzymuj zadania do planowania w grafie G, do wykonania na liście P; a spośród wszystkich liści G, wybierz zadanie J o najpóźniejszym 	Notes
 LDF – latest deadline first 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); 	Notes
 LDF – latest deadline first a utrzymuj zadania do planowania w grafie G, do wykonania na liście P; a spośród wszystkich liści G, wybierz zadanie J o najpóźniejszym 	Notes

Planowanie zadań z wymaganą kolejnością wykonania

- 3 utrzymuj zadania do planowania w grafie G, do wykonania na liście P;
- @ spośród wszystkich liści G, wybierz zadanie J o najpóźniejszym terminie wykonania (deadline);
- ${\bf 3}$ wstaw J na początek P i usuń je z G
- 4 wykonuj ponownie od 2.

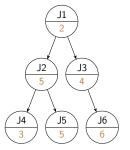
25 / 94

Notes

LDF – przykład

	J_1	J_2	J_3	J_4	J_5	J_6
aį	0	0	0	0	0	0
Ci	1	1	1	1	1	1
di	2	5	4	3	5	6



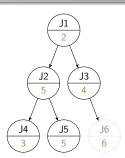


26 / 94

LDF – przykład

	J_1	J_2	J_3	J_4	J_5	<i>J</i> ₆
aį	0	0	0	0	0	0
Ci	1	1	1	1	1	1
di	2	5	4	3	5	6

 $P=(J_6)$

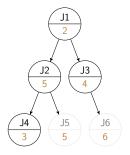


27 / 94

$\mathsf{LDF}-\mathsf{przyk}\mathsf{fad}$

	J_1	J_2	J_3	J_4	J_5	J_6
aį	0	0	0	0	0	0
Ci	1	1	1	1	1	1
di	2	5	4	3	5	6

 $P=(J_5,J_6)$



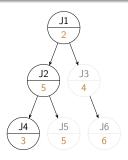
Notes

Notes

LDF – przykład

	J_1	J_2	<i>J</i> ₃	J ₄	J_5	<i>J</i> ₆
aį	0	0	0	0	0	0
Ci	1	1	1	1	1	1
di	2	5	4	3	5	6

$$P=\left(J_3,J_5,J_6\right)$$



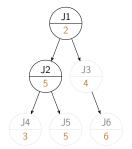
29 / 94

Notes

LDF – przykład

	J_1	J_2	J_3	J_4	J_5	J_6
aį	0	0	0	0	0	0
Ci	1	1	1	1	1	1
di	2	5	4	3	5	6

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

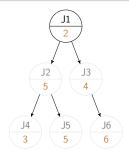


30 / 94

LDF – przykład

	J_1	J_2	<i>J</i> ₃	J_4	J_5	J ₆
aį	0	0	0	0	0	0
Ci	1	1	1	1	1	1
di	2	5	4	3	5	6

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

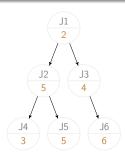


31 / 94

LDF – przykład

	J_1	J_2	J_3	J_4	J_5	J_6
aį	0	0	0	0	0	0
C_i	1	1	1	1	1	1
di	2	5	4	3	5	6

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



Notes

Notes

LDF – optymalność

Lawler '73

LDF jest optymalny ze względu na minimalizację maksymalnego opóźnienia.

LDF – optymalność

Lawler '73

LDF jest optymalny ze względu na minimalizację maksymalnego opóźnienia.

- dane są zadania (J_1,\ldots,J_n)
- ${\color{blue} \bullet}$ niech J_l nie ma zadań "poniżej" siebie w drzewie zależności
- ${\color{black} \bullet}$ niech J_l ma najpóźniejszy termin wykonania

LDF – optymalność

Lawler '73

LDF jest optymalny ze względu na minimalizację maksymalnego opóźnienia.

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



Notes	
M	
Notes	
Notes	
Notes	

LDF – optymalność

Lawler '73

LDF jest optymalny ze względu na minimalizację maksymalnego opóźnienia.

- dane są zadania (J_1, \ldots, J_n)
- ullet niech J_I nie ma zadań "poniżej" siebie w drzewie zależności
- ullet niech J_l ma najpóźniejszy termin wykonania



① przesuń J_l na koniec kolejki wykonania (bo $d_l > d_k$);

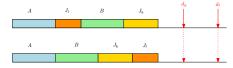
37 / 94

LDF – optymalność

Lawler '73

LDF jest optymalny ze względu na minimalizację maksymalnego opóźnienia.

- ullet dane są zadania (J_1,\ldots,J_n)
- \bullet niech J_I nie ma zadań "poniżej" siebie w drzewie zależności
- ullet niech J_I ma najpóźniejszy termin wykonania



- ① przesuń J_l na koniec kolejki wykonania (bo $d_l > d_k$);
- 2 pokaż, że kolejność wykonania jest zachowana

38 / 94

LDF – optymalność

Lawler '73

LDF jest optymalny ze względu na minimalizację maksymalnego opóźnienia.

- ullet dane są zadania (J_1,\ldots,J_n)
- ullet niech J_I ma najpóźniejszy termin wykonania



- ① przesuń 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

39 / 94

LDF – właściwości

• działa jedynie jako algorytm off-line

Notes	
Notes	
Notes	
Notes	

LDF – właściwości		N .
		Notes
 działa jedynie jako algorytm off-line 		
złożoność:		
	41 / 94	
LDF – właściwości		
		Notes
 działa jedynie jako algorytm off-line 		
 o złożoność: o szukanie liści w grafie G − O(E) 		
	42 / 94	
LDF – właściwości	_	
EDI WIASCIWOSCI		Notes
1.		
 działa jedynie jako algorytm off-line złożoność: 		
• szukanie liści w grafie $G - \mathcal{O}(E)$ • utrzymywanie posortowanej (wg. d_i) listy liści – $\mathcal{O}(\log n)$		
	43 / 94	
LDF – właściwości		Notes
 działa jedynie jako algorytm off-line złożoność: 		
 szukanie liści w grafie G − O(E) utrzymywanie posortowanej (wg. d₁) listy liści − O(log n) 		
• ogółem: $\mathcal{O}(n * \max(E , \log n))$		

Planowanie zadań z wymaganą kolejnością wykonania	_	
		Notes
 Planowanie bez wywłaszczenia, z asynchronicznymi pojawieniami się zadań, czasami wykonania i wymaganą 		
kolejnością jest NP-trudne.		
	45 / 94	
	45 / 94	
Planowanie zadań z wymaganą kolejnością wykonania	_	
r ianowanie zadan z wymaganą kolejnością wykonama		Notes
Planowanie bez wywłaszczenia, z asynchronicznymi		
pojawieniami się zadań, czasami wykonania i wymaganą kolejnością jest NP-trudne .		
 Dla pewnych rozluźnień problemu można znaleźć optymalne rozwiązania 		
i Ozwiązania		
	46 / 94	
	46 / 94	
Planowanie zadań z wymaganą kolejnością wykonania	46 / 94	Notes
Planowanie zadań z wymaganą kolejnością wykonania	46 / 94	Notes
Planowanie zadań z wymaganą kolejnością wykonania	46 / 94	Notes
Planowanie zadań z wymaganą kolejnością wykonania • Planowanie bez wywłaszczenia, z asynchronicznymi	46 / 94	Notes
 Planowanie bez wywłaszczenia, z asynchronicznymi pojawieniami się zadań, czasami wykonania i wymaganą 	46 / 94	Notes
 Planowanie bez wywłaszczenia, z asynchronicznymi pojawieniami się zadań, czasami wykonania i wymaganą kolejnością jest NP-trudne. Dla pewnych rozluźnień problemu można znaleźć optymalne 	46 / 94	Notes
 Planowanie bez wywłaszczenia, z asynchronicznymi pojawieniami się zadań, czasami wykonania i wymaganą kolejnością jest NP-trudne. 	46 / 94	Notes
 Planowanie bez wywłaszczenia, z asynchronicznymi pojawieniami się zadań, czasami wykonania i wymaganą kolejnością jest NP-trudne. Dla pewnych rozluźnień problemu można znaleźć optymalne rozwiązania 	46 / 94	Notes
 Planowanie bez wywłaszczenia, z asynchronicznymi pojawieniami się zadań, czasami wykonania i wymaganą kolejnością jest NP-trudne. Dla pewnych rozluźnień problemu można znaleźć optymalne rozwiązania 	46 / 94	Notes
 Planowanie bez wywłaszczenia, z asynchronicznymi pojawieniami się zadań, czasami wykonania i wymaganą kolejnością jest NP-trudne. Dla pewnych rozluźnień problemu można znaleźć optymalne rozwiązania 	46 / 94	Notes
 Planowanie bez wywłaszczenia, z asynchronicznymi pojawieniami się zadań, czasami wykonania i wymaganą kolejnością jest NP-trudne. Dla pewnych rozluźnień problemu można znaleźć optymalne rozwiązania 	46/94	Notes
 Planowanie bez wywłaszczenia, z asynchronicznymi pojawieniami się zadań, czasami wykonania i wymaganą kolejnością jest NP-trudne. Dla pewnych rozluźnień problemu można znaleźć optymalne rozwiązania 	46/94	Notes
 Planowanie bez wywłaszczenia, z asynchronicznymi pojawieniami się zadań, czasami wykonania i wymaganą kolejnością jest NP-trudne. Dla pewnych rozluźnień problemu można znaleźć optymalne rozwiązania 		Notes
 Planowanie bez wywłaszczenia, z asynchronicznymi pojawieniami się zadań, czasami wykonania i wymaganą kolejnością jest NP-trudne. Dla pewnych rozluźnień problemu można znaleźć optymalne rozwiązania 		
 Planowanie bez wywłaszczenia, 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 		Notes
 Planowanie bez wywłaszczenia, 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 		
 Planowanie bez wywłaszczenia, 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 Planowanie zadań z wymaganą kolejnością wykonania		
 Planowanie bez wywłaszczenia, 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 Planowanie zadań z wymaganą kolejnością wykonania Planowanie bez wywłaszczenia, z asynchronicznymi pojawieniami się zadań, czasami wykonania i wymaganą 		
 Planowanie bez wywłaszczenia, 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 Planowanie zadań z wymaganą kolejnością wykonania Planowanie bez wywłaszczenia, z asynchronicznymi pojawieniami się zadań, czasami wykonania i wymaganą kolejnością jest NP-trudne. 		
 Planowanie bez wywłaszczenia, 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 Planowanie zadań z wymaganą kolejnością wykonania Planowanie bez wywłaszczenia, z asynchronicznymi pojawieniami się zadań, czasami wykonania i wymaganą kolejnością jest NP-trudne. Dla pewnych rozluźnień problemu można znaleźć optymalne rozwiązania 		
 Planowanie bez wywłaszczenia, 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 Planowanie zadań z wymaganą kolejnością wykonania Planowanie bez wywłaszczenia, z asynchronicznymi pojawieniami się zadań, czasami wykonania i wymaganą kolejnością jest NP-trudne. Dla pewnych rozluźnień problemu można znaleźć optymalne 		

Zmodyfikowany EDF – przykład	
	Notes
Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie $(\exists i,j:a_i \neq a_j)$:	
49/94	
3,3	
Zmodyfikowany EDF – przykład	
	Notes
Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie $(\exists i,j:a_i\neq a_j)$:	
① zaktualizuj czasy nadejścia a_i :	
50/94	
5V / 94	
Zmodyfikowany EDF – przykład	
Zmodyfikowany EDF – przykład	Notes
	Notes
Zmodyfikowany EDF – przykład Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie $(\exists i,j:a_i\neq a_j)$:	Notes
Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie $(\exists i, j: a_i \neq a_j)$: ② zaktualizuj czasy nadejścia a_i :	Notes
Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie $(\exists i,j:a_i\neq a_j)$:	Notes
Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie $(\exists i, j: a_i \neq a_j)$: ② zaktualizuj czasy nadejścia a_i :	Notes
Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie $(\exists i, j: a_i \neq a_j)$: ② zaktualizuj czasy nadejścia a_i :	Notes
Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie $(\exists i, j: a_i \neq a_j)$: ② zaktualizuj czasy nadejścia a_i :	Notes
Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie $(\exists i, j: a_i \neq a_j)$: ② zaktualizuj czasy nadejścia a_i :	Notes
Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie $(\exists i, j: a_i \neq a_j)$: ② zaktualizuj czasy nadejścia a_i :	Notes
Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie $(\exists i,j:a_i\neq a_j)$: ① zaktualizuj czasy nadejścia a_i : o każde zadanie w "korzeniu": $a_i^*=a_i$	Notes
Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie $(\exists i,j:a_i\neq a_j)$: ① zaktualizuj czasy nadejścia a_i : o każde zadanie w "korzeniu": $a_i^*=a_i$	
Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie $(\exists i,j:a_i\neq a_j)$: ① zaktualizuj czasy nadejścia a_i : o każde zadanie w "korzeniu": $a_i^*=a_i$	Notes
Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie $(\exists i,j:a_i\neq a_j)$: ① zaktualizuj czasy nadejścia a_i : o każde zadanie w "korzeniu": $a_i^*=a_i$ 51/94 Zmodyfikowany EDF — przykład	
Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie $(\exists i,j:a_i\neq a_j)$: ① zaktualizuj czasy nadejścia a_i : ○ każde zadanie w "korzeniu": $a_i^*=a_i$ $a_i^*=a_i$ Zmodyfikowany EDF — przykład Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie $(\exists i,j:a_i\neq a_j)$:	
Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie (∃i, j : a _i ≠ a _j): ② zaktualizuj czasy nadejścia a _i : ○ każde zadanie w "korzeniu": a _i * = a _i 51/94 Zmodyfikowany EDF − przykład Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie (∃i, j : a _i ≠ a _j): ③ zaktualizuj czasy nadejścia a _i : ○ każde zadanie w "korzeniu": a _i * = a _i	
Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie $(\exists i,j:a_i\neq a_j)$: ① zaktualizuj czasy nadejścia a_i : ○ każde zadanie w "korzeniu": $a_i^*=a_i$ Zmodyfikowany EDF — przykład Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie $(\exists i,j:a_i\neq a_j)$: ② zaktualizuj czasy nadejścia a_i :	
Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie (∃i, j: a _i ≠ a _j): ② zaktualizuj czasy nadejścia a _i : ○ każde zadanie w "korzeniu": a _i * = a _i Zmodyfikowany EDF − przykład Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie (∃i, j: a _i ≠ a _j): ② zaktualizuj czasy nadejścia a _i : ○ każde zadanie w "korzeniu": a _i * = a _i ○ dla każdego zadania, którego czas nadejścia "rodzica" J _r został	
Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie (∃i, j: a _i ≠ a _j): ② zaktualizuj czasy nadejścia a _i : ○ każde zadanie w "korzeniu": a _i * = a _i Zmodyfikowany EDF − przykład Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie (∃i, j: a _i ≠ a _j): ② zaktualizuj czasy nadejścia a _i : ○ każde zadanie w "korzeniu": a _i * = a _i ○ dla każdego zadania, którego czas nadejścia "rodzica" J _r został	

Zmodyfikowany EDF – przykład

Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie $(\exists i,j:a_i \neq a_j)$:

- 1 zaktualizuj czasy nadejścia a_i:
 - $_{ullet}$ każde zadanie w "korzeniu": $a_{i}^{st}=a_{i}$
 - o dla każdego zadania, którego czas nadejścia "rodzica" J_r został zaktualizowany: $a_i^*=\max\{a_i,a_r^*+C_r\}$
- 2 zaktualizuj terminy wykonania:

53 / 94

Zmodyfikowany EDF – przykład

Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie $(\exists i,j:a_i\neq a_j)$:

- 1 zaktualizuj czasy nadejścia a_i:
 - $_{\odot}$ każde zadanie w "korzeniu": $a_{i}^{*}=a_{i}$
 - o dla każdego zadania, którego czas nadejścia "rodzica" J_r został zaktualizowany: $a_i^*=\max\{a_i,a_r^*+C_r\}$
- 2 zaktualizuj terminy wykonania:
 - $_{ ext{0}}$ dla każdego zadania w "liściu": $d_{i}^{st}=d_{i}$

54 / 94

Zmodyfikowany EDF – przykład

Aby zaadaptować przypadek, gdy zadania pojawiają się w systemie asynchronicznie $(\exists i,j:a_i\neq a_j)$:

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

55 / 94

Zmodyfikowany EDF – przykład

Notes	
Notes	
N	
Notes	
N	
Notes	

Zmodyfikowany EDF – przykład

	J_1	J_2	J_3	J_4	J_5	J_6
ai	1	2	3	3	3	4
a _i a _i *						
Ci	1	2	1	1	1	3
d _i d*	2	5	4	6	5	7
d*						



57 / 94

Zmodyfikowany EDF – przykład

	J_1	J_2	J_3	J_4	J_5	J_6
ai	1	2	3	3	3	4
a _i *	1	2	3	4	4	4
C_i	1	2	1	1	1	3
di	2	5	4	6	5	7
d*						

$$\bullet \ a_1^*=a_1=1$$

$$\quad \bullet \ \ a_2^* = \max\{a_2, a_1^* + C_1\} = 2$$

$$a_3^* = \max\{a_3, a_1^* + C_1\} = 3$$

$$\quad \bullet \ \ a_5^* = \max\{a_5, a_2^* + C_2\} = 4$$

$$a_6^* = \max\{a_6, a_3^* + C_3\} = 4$$



58 / 94

Zmodyfikowany EDF – przykład

ı		J_1	J_2	J_3	J_4	J_5	J_6
ĺ	aį	1	2	3	3	3	4
İ	a_i^*	1	2	3	4	4	4
Ī	Ci	1	2	1	1	1	3
ſ	di	2	5	4	6	5	7
l	d _i *						



59 / 94

Zmodyfikowany EDF – przykład

	J_1	J_2	J_3	J_4	J_5	J_6
ai	1	2	3	3	3	4
a _i *	1	2	3	4	4	4
C_i	1	2	1	1	1	3
di	2	5	4	6	5	7
d*	0	4	4	6	5	7

$$d_6^* = d_6 = 7$$

•
$$d_5^* = d_5 = 5$$

•
$$d_4^* = d_4 = 6$$

$$d_3^* = \min\{d_3, d_6^* - C_6\} = 4$$

$$\quad \bullet \ \, d_1^* = \min\{d_1, d_2^* - C_2, d_3^* - C_3\} = 0$$



Notes			

Notes			

Notes

Zmodyfikowany EDF – przykład

	J_1	J_2	J_3	J_4	J_5	J_6
aį	1	2	3	3	3	4
a _i *	1	2	3	4	4	4
Ci	1	2	1	1	1	3
di	2	5	4	6	5	7
d_i^*	0	4	4	6	5	7



Jak zmodyfikować zadania, by plan był wykonalny?

61/94

Notes

Zmodyfikowany EDF – przechodniość planowalności

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.

62 / 94

Zmodyfikowany EDF – przechodniość planowalności

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.

 $\ \, \textbf{\textcircled{1}} \,\,$ Dla każdego planu σ zachowującego zależności musi być:

63 / 94

Zmodyfikowany EDF – przechodniość planowalności

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: o $s_i \geq \max\{a_i, a_r^* + C_r\}$

	_
	_
	_
	_
	_
	_
otes	
	_
	_
	_
	_
	_
otes	
	_
	_
	_
	_
	_
	_
	_
otes	
	_
	_
	_
	_
	_
	_

Zmodyfikowany EDF – przechodniość planowalności

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:

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

Notes

Zmodyfikowany EDF – przechodniość planowalności

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:

- $s_i \ge \max\{a_i, a_r^* + C_r\}$ $f_i \le \min\{d_i, d_p^* C_p\}$
- ② Jeśli zadania $J_i \prec J_j$ planowane są przez EDF, to:

66 / 94

Zmodyfikowany EDF – przechodniość planowalności

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.

 $\ \, \textbf{\textcircled{a}} \,$ Dla każdego planu σ zachowującego zależności musi być:

- $s_i \geq \max\{a_i, a_r^* + C_r\}$
- $\bullet \ f_i \leq \min\{d_i, d_p^* C_p\}$
- ② Jeśli zadania $J_i \prec J_j$ planowane są przez EDF, to:
 - $a_i^* < a_i^*$: J_i pojawiło się wcześniej niż J_j ;

Zmodyfikowany EDF – przechodniość planowalności

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:

- $s_i \ge \max\{a_i, a_r^* + C_r\}$ $f_i \le \min\{d_i, d_p^* C_p\}$
- ② Jeśli zadania $J_i \prec J_j$ planowane 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;

Notes		
Notes		
Notes		

Zmodyfikowany EDF – przechodniość planowalności

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 $J_i \prec J_j$ planowane 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;

Notes

Zmodyfikowany EDF – przechodniość planowalności

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:

- $s_i \geq \max\{a_i, a_r^* + C_r\}$
- $\bullet \ f_i \leq \min\{d_i, d_p^* C_p\}$
- ② Jeśli zadania $J_i \prec J_j$ planowane są przez EDF, to:
 - $a_i^* < a_i^*$: J_i pojawiło się wcześniej niż J_i ;
 - $d_i^* < d_j^*$: J_i ma wcześniejszy termin wykonania;

zatem:

Zmodyfikowany EDF – przechodniość planowalności

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.

 $\textcircled{\scriptsize 1}$ 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 $J_i \prec J_j$ planowane są przez EDF, to:
 - $a_i^* < a_i^*$: J_i pojawiło się wcześniej niż J_i ;
 - \circ $d_i^* < d_j^*$: J_i ma wcześniejszy termin wykonania;

zatem:

• J_i nie może się rozpocząć przed J_i , oraz J_i nie wywłaszczy J_i ;

Zmodyfikowany EDF – przechodniość planowalności

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:

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

② Jeśli zadania $J_i \prec J_j$ planowane 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:

- \circ J_i nie może się rozpocząć przed J_i , oraz J_j nie wywłaszczy J_i ;
- $a_i^* \ge a_i$ oraz $d_i^* \le d_i$

Notes	
	_
Notes	
Notes	

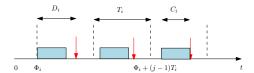
73 / 94

Notes

Notes

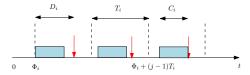
Notes

Planowanie zadań okresowych



74 / 94

Planowanie zadań okresowych

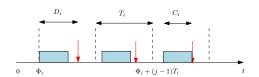


Każde zadanie okresowe t_i określone jest przez:

ullet fazę Φ_i – pierwszy moment wystąpienia;

75 / 94

Planowanie zadań okresowych

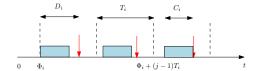


Każde zadanie okresowe t_i określone jest przez:

- fazę Φ_i pierwszy moment wystąpienia;
- okres \mathcal{T}_i czas jaki upływa między dwoma kolejnymi wystąpieniami;

Notes			

Planowanie zadań okresowych



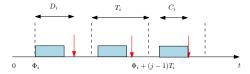
Każde zadanie okresowe t_i określone jest przez:

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

77 / 94

Notes

Planowanie zadań okresowych

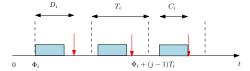


Każde zadanie okresowe t_i określone jest przez:

- 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;
- \circ czas wykonania C_i czas trwania obliczeń zadania.

78 / 94

Planowanie zadań okresowych

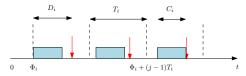


Każde zadanie okresowe t_i określone jest przez:

- ullet fazę Φ_i pierwszy moment wystąpienia;
- ullet 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.

79 / 94

Planowanie zadań okresowych



Każde zadanie okresowe t_i określone jest przez:

- ullet fazę Φ_i pierwszy moment wystąpienia;
- ullet 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.

Notes	
Notes	
Notes	
140103	

Planista RM - rate-monotonic		
		Notes
(Liu, Layland, 1973):		
 każde zadanie otrzymuje priorytet 1/T_i (proporcjonalny do częstotliwości pojawiania się); 		
	81/94	
Planista RM - rate-monotonic		
		Notes
(Liu, Layland, 1973):		
 każde zadanie otrzymuje priorytet 1/T_i (proporcjonalny do częstotliwości pojawiania się); 		
 wykonywane jest gotowe zadanie z aktualnie najwyższym 		
priorytetem;		
	82 / 94	
Planista RM - rate-monotonic		N .
		Notes
(L'or La Land 1072)		
(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;		
	83 / 94	
RM – przykład (1)		
Trivi — przykłau (1)		Notes
$\begin{array}{c cccc} & t_1 & t_2 & t_3 \\ \hline \Phi_i & 0 & 0 & 0 \end{array}$		
$egin{array}{ c c c c c c c c c c c c c c c c c c c$		
$\begin{array}{c cccc} C_1 & 2 & 1 & 4 \\ \hline D_i & 4 & 6 & 12 \\ \end{array}$		

RM – przykład (1)

	t_1	t ₂	t ₃
Фі	0	0	0
T_i	4	6	12
C_i	2	1	4
D_i	4	6	12

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

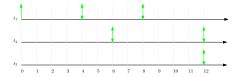
85 / 94

Notes

RM - przykład (1)

	t_1	t ₂	t ₃
Фі	0	0	0
Ti	4	6	12
C_i	2	1	4
Di	4	6	12

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

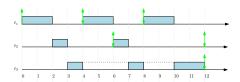


86 / 94

RM – przykład (1)

	t_1	t ₂	t ₃
Фі	0	0	0
T_i	4	6	12
C_i	2	1	4
D_i	4	6	12

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



87 / 94

RM – przykład (2)

		t_1	t ₂	t ₃
	Фі	0	0	0
	T_i	4	5	10
	C_i	2	2	1
Ì	Di	4	5	10

Votes			
Notes			
Notes			

RM – przykład (2)

	t_1	t ₂	t ₃
Фі	0	0	0
T_i	4	5	10
Ci	2	2	1
Di	4	5	10

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

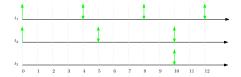
89 / 94

Notes

RM - przykład (2)

	t_1	t ₂	t ₃
Фі	0	0	0
Ti	4	5	10
C_i	2	2	1
Di	4	5	10

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

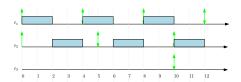


90 / 94

RM – przykład (2)

	t_1	t ₂	t ₃
Фі	0	0	0
T_i	4	5	10
C_i	2	2	1
Di	4	5	10

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



91/94

Stopień wykorzystania procesora - warunek planowalności

Stopień wykorzystania procesora

Dla danego zbioru \boldsymbol{n} zadań okresowych, stopień wykorzystania procesora to:

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

 ${\color{blue} \bullet}$ warunek konieczny dla stworzenia realizowalnego planu: ${\it U} \leq 1$

Notes			
Notes			
Notes			
Notes			
Notes			

Stopień wykorzystania procesora - warunek planowalności

Stopień wykorzystania procesora

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

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

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

93 / 94

Rzeczy do zapamiętania

- różnice między RTOS a "PC-OS";
- właściwości czasowe zadań w systemie (w tym okresowych);
- rodzaje planistów, jakie warunki biorą pod uwagę;
- algorytm EDD, EDF, LDF, Bratley'a, RM

94 / 94

Notes	
Notes	
Notes	
Notes	