Systemy wbudowane - wykład 10 Przemek Błaśkiewicz 10 maja 2018 1/91 Definicja Notes 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*). 2/91 Gotowość Notes • Program (system) jest <u>zawsze</u> gotowy na napływające dane Gotowość Notes Program (system) jest <u>zawsze</u> gotowy na napływające dane synchronicznie;

Notes

	_	
Gotowość		Notes
		TVOCCS
 Program (system) jest <u>zawsze</u> gotowy na napływające dane 		
synchronicznie;		
asynchronicznie.		
	5/91	
	_	
Gotowość		N.A
		Notes
December (control) in the control of		
 Program (system) jest <u>zawsze</u> gotowy na napływające dane synchronicznie; 		
 asynchronicznie. Proces/system przetwarza dane w sposób przewidywalny, w 		
rozumieniu:		
	6/91	
	_	
Gotowość		
		Notes
 Program (system) jest <u>zawsze</u> gotowy na napływające dane synchronicznie; 		
 asynchronicznie. 		
 Proces/system przetwarza dane w sposób <u>przewidywalny</u>, w rozumieniu: 		
 długości czasu przetwarzania; 		
	7/91	
Gotowość		
		Notes
 Program (system) jest <u>zawsze</u> gotowy na napływające dane synchronicznie; 		
synchronicznie;asynchronicznie.		

Proces/system przetwarza dane w sposób <u>przewidywalny</u>, w rozumieniu:

 długości czasu przetwarzania;
 na bieżąco (długości czasu oczekiwania na rozpoczęcie przetwarzania);

 Program (system) jest <u>zawsze</u> gotowy na napływające dane 		
synchronicznie;asynchronicznie.		
 Proces/system przetwarza dane w sposób przewidywalny, w rozumieniu: 		
 długości czasu przetwarzania; na bieżąco (długości czasu oczekiwania na rozpoczęcie 		
przetwarzania); spełnienia dodatkowych założeń dot. np. kolejności.		
spermenta dodarkowych zarożen dot. np. kolejności.		
	9/91	
Gotowość		
		Notes
 Program (system) jest <u>zawsze</u> gotowy na napływające dane synchronicznie; 		
 asynchronicznie. Proces/system przetwarza dane w sposób przewidywalny, w 		
rozumieniu: o długości czasu przetwarzania;		
 na bieżąco (długości czasu oczekiwania na rozpoczęcie przetwarzania); 		
 spełnienia dodatkowych założeń dot. np. kolejności. Współdziała ze środowiskiem, reagując na jego niezależne 		
zmiany.		
	10/91	
	_	
Przypomnienie o RTOS		
Przypomnienie o RTOS		Notes
Przypomnienie o RTOS	_	Notes
Przypomnienie o RTOS		Notes
		Notes
Przypomnienie o RTOS • Pojęcie "real-time" nie oznacza "szybki".		Notes
	11/91	Notes
Pojęcie "real-time" nie oznacza "szybki".	11/91	Notes
	11/91	Notes
Pojęcie "real-time" nie oznacza "szybki".	11/91	
Pojęcie "real-time" nie oznacza "szybki".	11/91	
Pojęcie "real-time" nie oznacza "szybki". Przypomnienie o RTOS	11/91	
Pojęcie "real-time" nie oznacza "szybki".	11/91	
Przypomnienie o RTOS Pojęcie "real-time" nie oznacza "szybki".	11/91	
Przypomnienie o RTOS Pojęcie "real-time" nie oznacza "szybki".	11/91	
Przypomnienie o RTOS Pojęcie "real-time" nie oznacza "szybki".	11/91	

Notes

- Pojęcie "real-time" nie oznacza "szybki".
- PC vs. SW vs. RTOS
- hard/soft/firm OS

13/91

Notes

Planowanie zadań

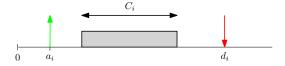
Plan

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

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

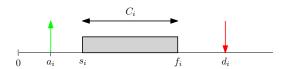
14/91

Cechy charakterystyczne zadań



15 / 91

Cechy charakterystyczne zadań



Notes	
Notes	
votes	
Notes	

Algorytmy planowania (1)	_	
		Notes
Realizowalność planu	1	
(feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie		
wykonania.		
,		
	17/91	
Algorytmy planowania (1)		
7 ligorythiy planowama (1)		Notes
		Notes
Realizowalność planu	h	
(feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje		
że wszystkie zadania kończą się nie później niż w swoim terminie		
wykonania.	J	
wywłaszczające i niewywłaszczające;		
wywiaszczające i niewywiaszczające,		
	18/91	
	18/91	
	18/91	
Algorytmy planowania (1)		
Algorytmy planowania (1)		Notes
Algorytmy planowania (1)		Notes
		Notes
Realizowalność planu		Notes
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje		Notes
Realizowalność planu		Notes
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania.		Notes
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. • wywłaszczające i niewywłaszczające;		Notes
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania.		Notes
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. • wywłaszczające i niewywłaszczające;		Notes
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. • wywłaszczające i niewywłaszczające;		Notes
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. • wywłaszczające i niewywłaszczające;		Notes
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. • wywłaszczające i niewywłaszczające;		Notes
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. • wywłaszczające i niewywłaszczające;		Notes
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. • wywłaszczające i niewywłaszczające;		Notes
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. • wywłaszczające i niewywłaszczające;		Notes
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. • wywłaszczające i niewywłaszczające; • statyczne i dynamiczne;		Notes
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. • wywłaszczające i niewywłaszczające;		Notes
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. • wywłaszczające i niewywłaszczające; • statyczne i dynamiczne;	19/91	Notes
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. • wywłaszczające i niewywłaszczające; • statyczne i dynamiczne;	19/91	
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. • wywłaszczające i niewywłaszczające; • statyczne i dynamiczne; Algorytmy planowania (1)	19/91	
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. • wywłaszczające i niewywłaszczające; • statyczne i dynamiczne; Algorytmy planowania (1)	19/91	
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. • wywłaszczające i niewywłaszczające; • statyczne i dynamiczne; Algorytmy planowania (1) Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje	19/91	
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. • wywłaszczające i niewywłaszczające; • statyczne i dynamiczne; Algorytmy planowania (1) Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie	19/91	
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. • wywłaszczające i niewywłaszczające; • statyczne i dynamiczne; Algorytmy planowania (1) Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje	19/91	
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. • wywłaszczające i niewywłaszczające; • statyczne i dynamiczne; Algorytmy planowania (1) Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie	19/91	
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. • wywłaszczające i niewywłaszczające; • statyczne i dynamiczne; Algorytmy planowania (1) Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania.	19/91	
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. • wywłaszczające i niewywłaszczające; • statyczne i dynamiczne; Algorytmy planowania (1) Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. • wywłaszczające i niewywłaszczające;	19/91	
Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. • wywłaszczające i niewywłaszczające; • statyczne i dynamiczne; Algorytmy planowania (1) Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. • wywłaszczające i niewywłaszczające; • statyczne i dynamiczne;	19/91	

Algorytmy planowania (1)Notes Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wvkonania. wywłaszczające i niewywłaszczające; statyczne i dynamiczne; jednoprocesorowe, wieloprocesorowe; optymalne, heurystyczne; 21/91 Algorytmy planowania (1) Notes Realizowalność planu (feasibility) – plan jest realizowalny, jeśli jego wykonanie powoduje że wszystkie zadania kończą się nie później niż w swoim terminie wykonania. wywłaszczające i niewywłaszczające; statyczne i dynamiczne; jednoprocesorowe, wieloprocesorowe; optymalne, heurystyczne; • dla zadań okresowych (periodycznych) i asynchronicznych; 22 / 91 Lemat o wywłaszczaniu Notes Jeśli czasy nadejścia zadań są synchroniczne, to mechanizm wywłaszczenia nie polepsza maksymalnego opóźnienia. Czyli jeśli istnieje algorytm wywłaszczeniowy o opóźnieniu L_{max} to istnieje również algorytm niewywłaszczeniowy o tym samym opóźnieniu dla tego samego zestawu zadań. Zadania aperiodyczne, synchroniczne Notes • Dany jest zestaw n zadań: J_1, \ldots, J_n , gdzie:

	Notes
$ullet$ Dany jest zestaw n zadań: J_1,\ldots,J_n , gdzie:	
\circ dla wszystkich zadań $a_i=0, i=1,\ldots n;$	
25/91	
7 1	
Zadania aperiodyczne, synchroniczne	N
	Notes
 Dany jest zestaw n zadań: J₁,, J_n, gdzie: dla wszystkich zadań a_i = 0, i = 1,n; 	
• każde zadanie ma określony deadline d_i i czas obliczenia C_i ;	
26/91	
26/91	
Zadania aperiodyczne, synchroniczne	Notes
	Notes
	Notes
Zadania aperiodyczne, synchroniczne	Notes
Zadania aperiodyczne, synchroniczne • Dany jest zestaw n zadań: J_1, \ldots, J_n , gdzie: • dla wszystkich zadań $a_i = 0, i = 1, \ldots n$;	Notes
Zadania aperiodyczne, synchroniczne • Dany jest zestaw n zadań: J_1, \ldots, J_n , gdzie: • dla wszystkich zadań $a_i = 0, i = 1, \ldots n$; • każde zadanie ma określony deadline d_i i czas obliczenia C_i ;	Notes
Zadania aperiodyczne, synchroniczne • Dany jest zestaw n zadań: J_1, \ldots, J_n , gdzie: • dla wszystkich zadań $a_i = 0, i = 1, \ldots n$;	Notes
Zadania aperiodyczne, synchroniczne • Dany jest zestaw n zadań: J_1, \ldots, J_n , gdzie: • dla wszystkich zadań $a_i = 0, i = 1, \ldots n$; • każde zadanie ma określony deadline d_i i czas obliczenia C_i ;	Notes
Zadania aperiodyczne, synchroniczne • Dany jest zestaw n zadań: J_1, \ldots, J_n , gdzie: • dla wszystkich zadań $a_i = 0, i = 1, \ldots n$; • każde zadanie ma określony deadline d_i i czas obliczenia C_i ;	Notes
Zadania aperiodyczne, synchroniczne • Dany jest zestaw n zadań: J_1, \ldots, J_n , gdzie: • dla wszystkich zadań $a_i = 0, i = 1, \ldots n$; • każde zadanie ma określony deadline d_i i czas obliczenia C_i ;	Notes
Zadania aperiodyczne, synchroniczne • Dany jest zestaw n zadań: J_1, \ldots, J_n , gdzie: • dla wszystkich zadań $a_i = 0, i = 1, \ldots n$; • każde zadanie ma określony deadline d_i i czas obliczenia C_i ;	Notes
Zadania aperiodyczne, synchroniczne • Dany jest zestaw n zadań: J_1, \ldots, J_n , gdzie: • dla wszystkich zadań $a_i = 0, i = 1, \ldots n$; • każde zadanie ma określony deadline d_i i czas obliczenia C_i ;	Notes
Zadania aperiodyczne, synchroniczne • Dany jest zestaw n zadań: J_1, \ldots, J_n , gdzie: • dla wszystkich zadań $a_i = 0, i = 1, \ldots n$; • każde zadanie ma określony deadline d_i i czas obliczenia C_i ; • zadania są niezależne (nie ma zależności wykonania).	Notes
Zadania aperiodyczne, synchroniczne • Dany jest zestaw n zadań: J_1, \ldots, J_n , gdzie: • dla wszystkich zadań $a_i = 0, i = 1, \ldots n$; • każde zadanie ma określony deadline d_i i czas obliczenia C_i ;	Notes
Zadania aperiodyczne, synchroniczne • Dany jest zestaw n zadań: J_1, \ldots, J_n , gdzie: • dla wszystkich zadań $a_i = 0, i = 1, \ldots n$; • każde zadanie ma określony deadline d_i i czas obliczenia C_i ; • zadania są niezależne (nie ma zależności wykonania).	Notes
Zadania aperiodyczne, synchroniczne • Dany jest zestaw n zadań: J_1, \ldots, J_n , gdzie: • dla wszystkich zadań $a_i = 0, i = 1, \ldots n$; • każde zadanie ma określony deadline d_i i czas obliczenia C_i ; • zadania są niezależne (nie ma zależności wykonania).	Notes
Zadania aperiodyczne, synchroniczne • Dany jest zestaw n zadań: J_1, \ldots, J_n , gdzie: • dla wszystkich zadań $a_i = 0, i = 1, \ldots n$; • każde zadanie ma określony deadline d_i i czas obliczenia C_i ; • zadania są niezależne (nie ma zależności wykonania).	Notes
Zadania aperiodyczne, synchroniczne • Dany jest zestaw n zadań: J_1, \ldots, J_n , gdzie: • dla wszystkich zadań $a_i = 0, i = 1, \ldots n$; • każde zadanie ma określony deadline d_i i czas obliczenia C_i ; • zadania są niezależne (nie ma zależności wykonania).	
Zadania aperiodyczne, synchroniczne • Dany jest zestaw n zadań: J_1, \ldots, J_n , gdzie: • dla wszystkich zadań $a_i = 0, i = 1, \ldots n$; • każde zadanie ma określony deadline d_i i czas obliczenia C_i ; • zadania są niezależne (nie ma zależności wykonania).	
Zadania aperiodyczne, synchroniczne • Dany jest zestaw n zadań: J_1, \ldots, J_n , gdzie: • dla wszystkich zadań $a_i = 0, i = 1, \ldots n$; • każde zadanie ma określony deadline d_i i czas obliczenia C_i ; • zadania są niezależne (nie ma zależności wykonania). Zadania aperiodyczne, synchroniczne • Dany jest zestaw n zadań: J_1, \ldots, J_n , gdzie:	
 Zadania aperiodyczne, synchroniczne Dany jest zestaw n zadań: J₁,, J_n, gdzie: dla wszystkich zadań a_i = 0, i = 1, n; każde zadanie ma określony deadline d_i i czas obliczenia C_i; zadania są niezależne (nie ma zależności wykonania). Zadania aperiodyczne, synchroniczne Dany jest zestaw n zadań: J₁,, J_n, gdzie: dla wszystkich zadań a_i = 0, i = 1, n; 	
 Zadania aperiodyczne, synchroniczne Dany jest zestaw n zadań: J₁,, J_n, gdzie: dla wszystkich zadań a_i = 0, i = 1, n; każde zadanie ma określony deadline d_i i czas obliczenia C_i; zadania są niezależne (nie ma zależności wykonania). Zadania aperiodyczne, synchroniczne Dany jest zestaw n zadań: J₁,, J_n, gdzie: dla wszystkich zadań a_i = 0, i = 1, n; każde zadanie ma określony deadline d_i i czas obliczenia C_i; zadania są niezależne (nie ma zależności wykonania). 	
 Zadania aperiodyczne, synchroniczne Dany jest zestaw n zadań: J₁,, J_n, gdzie: dla wszystkich zadań a_i = 0, i = 1, n; każde zadanie ma określony deadline d_i i czas obliczenia C_i; zadania są niezależne (nie ma zależności wykonania). Zadania aperiodyczne, synchroniczne Dany jest zestaw n zadań: J₁,, J_n, gdzie: dla wszystkich zadań a_i = 0, i = 1, n; każde zadanie ma określony deadline d_i i czas obliczenia C_i; 	
 Zadania aperiodyczne, synchroniczne Dany jest zestaw n zadań: J₁,, J_n, gdzie: dla wszystkich zadań a_i = 0, i = 1, n; każde zadanie ma określony deadline d_i i czas obliczenia C_i; zadania są niezależne (nie ma zależności wykonania). Zadania aperiodyczne, synchroniczne Dany jest zestaw n zadań: J₁,, J_n, gdzie: dla wszystkich zadań a_i = 0, i = 1, n; każde zadanie ma określony deadline d_i i czas obliczenia C_i; zadania są niezależne (nie ma zależności wykonania). 	

Zadania aperiodyczne, synchroniczne

Zadania aperiodyczne, synchroniczne Notes ullet Dany jest zestaw n zadań: J_1,\ldots,J_n , gdzie: \bullet dla wszystkich zadań $a_i=0, i=1,\ldots n;$ każde zadanie ma określony deadline d_i i czas obliczenia C_i; o zadania są niezależne (nie ma zależności wykonania). system bez wywłaszczenia jednoprocesorowy 29 / 91 Zadania aperiodyczne, synchroniczne Notes ullet Dany jest zestaw n zadań: J_1,\ldots,J_n , gdzie: • dla wszystkich zadań $a_i = 0, i = 1, \dots n;$ każde zadanie ma określony deadline d_i i czas obliczenia C_i; zadania są niezależne (nie ma zależności wykonania). system bez wywłaszczenia jednoprocesorowy • zadanie: znajdź optymalny plan 30/91 Planista EDD (earliest due date) Notes 31 / 91 Planista EDD (earliest due date) Notes Wykonuje zadania w kolejności niemalejącego czasu zakończenia.

Planista EDD (earliest due date)

Wykonuje zadania w kolejności niemalejącego czasu zakończenia.

Przykład 1

		_			
	J_1	J_2	<i>J</i> ₃	J_4	J_5
C_i	1	1	1	3	2
di	3	10	7	8	5

33/91

34 / 91

Planista EDD (earliest due date)

Wykonuje zadania w kolejności niemalejącego czasu zakończenia.

Przykład 1

		J_1	- J ₂	J_3	J ₄	<i>J</i> ₅
	C_i	1	1	1	3	2
ĺ	di	3	10	7	8	5

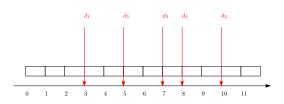


Planista EDD (earliest due date)

Wykonuje zadania w kolejności niemalejącego czasu zakończenia.

Przykład 1

	J_1	J_2	<i>J</i> ₃	J_4	J_5
Ci	1	1	1	3	2
di	3	10	7	8	5



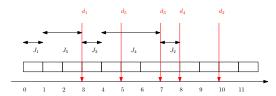
35 / 91

Planista EDD (earliest due date)

Wykonuje zadania w kolejności niemalejącego czasu zakończenia.

Przykład 1

	J_1	J_2	<i>J</i> ₃	J_4	J_5
C_i	1	1	1	3	2
d:	3	10	7	8	5



ΝI	-+	
1.71	OIES	

Notes

Notes

Notes

Planista EDD (earliest due date)

Wykonuje zadania w kolejności niemalejącego czasu zakończenia.

Przykład 2

		_			
	J_1	J_2	J_3	J_4	J_5
C_i	1	2	1	4	2
di	2	5	4	8	6

37 / 91

Planista EDD (earliest due date)

Wykonuje zadania w kolejności niemalejącego czasu zakończenia.

Przykład 2

		_			
	J_1	J_2	J_3	J_4	J_5
C_i	1	2	1	4	2
di	2	5	4	8	6



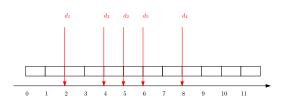
38 / 91

Planista EDD (earliest due date)

Wykonuje zadania w kolejności niemalejącego czasu zakończenia.

Przykład 2

	J_1	J_2	<i>J</i> ₃	J_4	<i>J</i> ₅
C_i	1	2	1	4	2
di	2	5	4	8	6



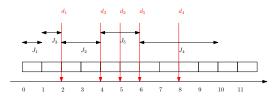
9/91

Planista EDD (earliest due date)

Wykonuje zadania w kolejności niemalejącego czasu zakończenia.

Przykład 2

		J_1	J_2	J_3	J ₄	J_5
	C_i	1	2	1	4	2
ĺ	d:	2	5	4	8	6



N	otes

Notes

Notes

Notes

-	
_	
-	
-	
_	

41 / 91

Notes

Notes

Notes

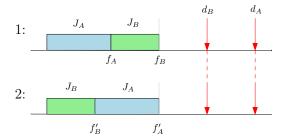
Twierdzenie o optymalności EDD

Jackson, 1955

Jeśli dany jest dowolny zestaw niezależnych zadań, pojawiających się w sytemie synchronicznie, to każdy algorytm wykonujący je w kolejności niemalejących terminów wykonania (deadlines) jest algorytmem optymalnym ze względu na minimalizację maksymalnego opóźnienia.

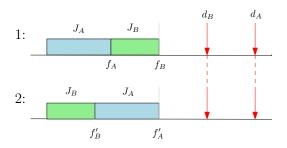
42 / 91

Twierdzenie o optymalności EDD



43/91

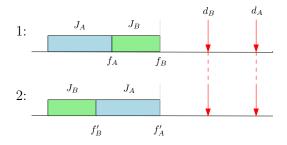
Twierdzenie o optymalności EDD



 $A f'_A - d_A = f_B - d_A \le f_B - d_B$

Notes			
ivotes			
-			

Twierdzenie o optymalności EDD

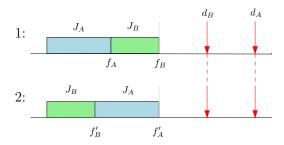


- (a) $f'_A d_A = f_B d_A \le f_B d_B$ (b) $f'_B d_B \le f_B d_B$

45 / 91

Notes

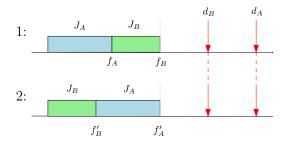
Twierdzenie o optymalności EDD



- (a) $f'_A d_A = f_B d_A \le f_B d_B$ (b) $f'_B d_B \le f_B d_B$

46 / 91

Twierdzenie o optymalności EDD



- (a) $f'_A d_A = f_B d_A \le f_B d_B$ (b) $f'_B d_B \le f_B d_B$

 $L'_{\max\limits_{a,b}} \leq L_{\max\limits_{a,b}}$

47 / 91

EDD - właściwości

• złożoność EDD: sortowanie, $\mathcal{O}(n \log n)$

	_
	—
	—
	_
Notes	
	_
	_
	_
	_
	_
	_
Notes	
	_
	_
	_
	_
	—
Notes	
	—
	—
	—

0	złożoność EDD: sortowanie, $\mathcal{O}(n \log n)$)
0	test na <i>planowalność</i> zadań $\{J_1,\ldots,J_n\}$	n}:

49 / 91

EDD - właściwości

- \circ złożoność EDD: sortowanie, $\mathcal{O}(n \log n)$
- test na planowalność zadań $\{J_1,\ldots,J_n\}$:
 - 1 posortuj listę zadań niemalejąco względem ich deadlines; (niech to będzie powyższy porządek);

50/91

EDD - właściwości

- \circ złożoność EDD: sortowanie, $\mathcal{O}(n \log n)$
- test na *planowalność* zadań $\{J_1,\ldots,J_n\}$:
 - 1 posortuj listę zadań niemalejąco względem ich deadlines; (niech to będzie powyższy porządek);
 ② sprawdź, czy ∀i ∈ 1,...,n: f_i ≤ d_i;

51/91

EDD - właściwości

- \circ złożoność EDD: sortowanie, $\mathcal{O}(n \log n)$
- test na planowalność zadań $\{J_1, \ldots, J_n\}$:
 - posortuj listę zadań niemalejąco względem ich deadlines; (niech to będzie powyższy porządek);
 sprawdź, czy ∀i ∈ 1,..., n: f_i ≤ d_i;
 ponieważ f_i = ∑_{k=1}ⁱ C_k warunek wystarczający:

$$\forall i \in 1, \dots, n: \ \sum_{k=1}^i C_k \leq d_i$$

Notes	
Notes	
Notes	
Notes	
Notes	

- złożoność EDD: sortowanie, $\mathcal{O}(n \log n)$
- test na planowalność zadań $\{J_1,\ldots,J_n\}$:
 - 1 posortuj listę zadań niemalejąco względem ich deadlines; (niech to będzie powyższy porządek); ② sprawdź, czy $\forall i \in 1, \ldots, n: f_i \leq d_i;$ ○ ponieważ $f_i = \sum_{k=1}^{i} C_k$ warunek wystarczający:

$$\forall i \in 1, \ldots, n: \sum_{k=1}^{i} C_k \leq d_i$$

• EDD jest optymalny, zatem jeśli zestawu zadań nie można rozplanować przez EDD, to nie można go rozplanować w ogólności.

53/91

Zadania aperiodyczne, asynchroniczne

54/91

Zadania aperiodyczne, asynchroniczne

- ullet Dany jest zestaw n zadań: J_1,\ldots,J_n , gdzie:
 - ai dla wszystkich zadań jest różne i nieznane;
 - każde zadanie ma określony deadline d_i i czas obliczenia C_i;
 - zadania są niezależne (nie ma zależności wykonania).
- system z wywłaszczeniem;
- jednoprocesorowy
- zadanie: znajdź plan minimalizujący maksymalne opóźnienie.

55 / 91

Planista EDF - Earliest Deadline First

EDF

W każdym momencie wykonuj zadanie z najwcześniejszym czasem wykonania spośród dostępnych zadań w systemie.

Notes	
	_
Notes	
	—
	_
Notes	
	_
Notes	
	_

Planista EDF – Earliest Deadline First EDF W każdym momencie wykonuj zadanie z najwcześniejszym czasem

57/91

58/91

wykonania spośród dostępnych zadań w systemie.

przyszłego zadania.

przyszłego zadania.

EDF - przykład

EDF - przykład

EDF

Planista EDF - Earliest Deadline First

wykonania spośród dostępnych zadań w systemie.

o Jeśli w systemie pojawia się zadanie z czasem wykonania wcześniejszym niż aktualnie wykonywane zadanie wykonywane zadanie jest wywłaszczone na rzecz nowo

W każdym momencie wykonuj zadanie z najwcześniejszym czasem

• Jeśli w systemie pojawia się zadanie z czasem wykonania wcześniejszym niż aktualnie wykonywane zadanie wykonywane zadanie jest wywłaszczone na rzecz nowo

• Założenie o znikomości czasu na przełączanie zadań!

 J_1 J_2 J_3 J_4 J_5 0 2

2 2

3 6

2 2

4 10

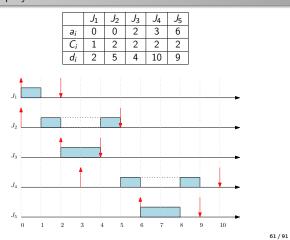
0

1 2 5

 a_i

otes
otes
otes
otes
otes

EDF - przykład



Optymalność EDF

62/91

Optymalność EDF

Horn 1974

Jeśli dany jest dowolny zestaw niezależnych zadań, pojawiających się w sytemie asynchronicznie, to każdy algorytm wykonujący je tak, że w każdym momencie wykonywane jest zadanie o najbliższym czasie wykonania (deadline) jest algorytmem optymalnym ze względu na minimalizację maksymalnego opóźnienia.

63/91

Optymalność EDF

Horn 1974

Jeśli dany jest dowolny zestaw niezależnych zadań, pojawiających się w sytemie asynchronicznie, to każdy algorytm wykonujący je tak, że w każdym momencie wykonywane jest zadanie o najbliższym czasie wykonania (deadline) jest algorytmem optymalnym ze względu na minimalizację maksymalnego opóźnienia.

– bez dowodu (mechanizm zbliżony do dowodu tw. Jacksona) –

Notes	
Notes	
Notes	
Notes	
Notes	

Zadania aperiodyczne, asynchroniczne, wersja bez wywłaszczenia		Notes
,	_	
6	55 / 91	
Zadania aperiodyczne, asynchroniczne, wersja bez	-	
wywłaszczenia		Notes
• Dany jest zestaw n zadań: J_1, \ldots, J_n , gdzie:		
 a_i dla wszystkich zadań jest różne i nieznane; każde zadanie ma określony deadline d_i i czas obliczenia C_i; zadania są niezależne (nie ma zależności wykonania). 		
• system bez wywłaszczeń;		
 jednoprocesorowy <u>zadanie:</u> znajdź plan minimalizujący maksymalne opóźnienie. 		
6	56 / 91	
EDF bez wywłaszczeń - przykład		
		Notes
6	57/91	
0	57 / 91	
EDF bez wywłaszczeń - przykład		Notes
$\begin{array}{c cc} & J_1 & J_2 \\ \hline a_i & 0 & 1 \\ \hline \end{array}$		Notes
$\begin{array}{c cc} C_i & 4 & 2 \\ \hline d_i & 7 & 5 \end{array}$		

EDF bez wywłaszczeń - przykład

	J_1	J_2
aį	0	1
Ci	4	2
di	7	5

EDF bez wywłaszczeń:



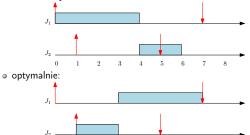
69 / 91

Notes

EDF bez wywłaszczeń - przykład

	J_1	J_2
aį	0	1
Ci	4	2
di	7	5

EDF bez wywłaszczeń:



70 / 91

EDF bez wywłaszczeń - własności

W poprzednim przykładzie CPU jest bezczynne w okresie 0-1, mimo gotowości zadania $J_1.$

71/91

EDF bez wywłaszczeń - własności

- W poprzednim przykładzie CPU jest bezczynne w okresie 0-1, mimo gotowości zadania $J_1.$
- \circ Jeśli czasy a_i nie są znane, to niemożliwe jest podjęcie decyzji o tej bezczynności.

Notes		
		_
Notes		
Notes		
		_

EDF bez wywłaszczeń - własności

- ullet W poprzednim przykładzie CPU jest bezczynne w okresie 0-1, mimo gotowości zadania $J_1.$
- Jeśli czasy ai nie są znane, to niemożliwe jest podjęcie decyzji o tej bezczynności.

Jeffay i in. 1991

Przy braku zezwolenia na okresy bezczynności przy gotowym do wykonania zadaniu, EDF jest optymalny również dla modelu bez wywłaszczeń.

73 / 91

Zadania aperiodyczne, asynchroniczne, wersja bez wywłaszczenia, dozwolone czasy bezczynności

74 / 91

Zadania aperiodyczne, asynchroniczne, wersja bez wywłaszczenia, dozwolone czasy bezczynności

- Dany jest zestaw n zadań: J_1, \ldots, J_n , gdzie:
 - a_i dla wszystkich zadań znane;
 - \circ każde zadanie ma określony deadline d_i i czas obliczenia C_i ;
 - zadania są niezależne (nie ma zależności wykonania).
- system bez wywłaszczeń;
- system zezwala na czasy bezczynności przy gotowym zadaniu;
- $\quad \bullet \ \, jedno procesorowy \\$
- zadanie: znajdź plan minimalizujący maksymalne opóźnienie.

75 / 91

Zadania aperiodyczne, asynchroniczne, wersja bez wywłaszczenia, dozwolone czasy bezczynności

- Dany jest zestaw n zadań: J_1, \ldots, J_n , gdzie:
 - a_i dla wszystkich zadań znane;
 - ullet każde zadanie ma określony deadline d_i i czas obliczenia C_i ;
 - o zadania są niezależne (nie ma zależności wykonania).
- system bez wywłaszczeń;
- system zezwala na czasy bezczynności przy gotowym zadaniu;
- jednoprocesorowy
- <u>zadanie:</u> znajdź plan minimalizujący maksymalne opóźnienie.

W ogólności, problem jest NP-trudny.

otes	
otes	
otes	
2422	
otes	

Zadania aperiodyczne, asynchroniczne, wersja bez wywłaszczenia, dozwolone czasy bezczynności

- Dany jest zestaw n zadań: J_1, \ldots, J_n , gdzie:
 - a_i dla wszystkich zadań znane;
 - każde zadanie ma określony deadline d_i i czas obliczenia C_i ;
 - zadania są niezależne (nie ma zależności wykonania).
- system bez wywłaszczeń;
- system zezwala na czasy bezczynności przy gotowym zadaniu;
- jednoprocesorowy
- zadanie: znajdź plan minimalizujący maksymalne opóźnienie.

W ogólności, problem jest **NP-trudny**. Rozwiązywalny przy użyciu heurystyk lub mechanizmów typu *branch-and-bound*.

77 / 91

Algorytm Bratley'a

- Wykorzystując mechanizm branch-and-bound odnajduje realizowalne rozwiązanie, jeśli istnieje.
- ullet Bazuje na odpowiedniej permutacji zestawu zadań J_1,\ldots,J_n .
- Rozpoczyna od pustej listy (kolejności) zadań;
- Branch: wybierz do listy kolejne zadanie (z pozostałych)
- Bound:

78 / 91

Algorytm Bratley'a

- Wykorzystując mechanizm branch-and-bound odnajduje realizowalne rozwiązanie, jeśli istnieje.
- ullet Bazuje na odpowiedniej permutacji zestawu zadań J_1,\ldots,J_n .
- Rozpoczyna od pustej listy (kolejności) zadań;
- Branch: wybierz do listy kolejne zadanie (z pozostałych)
- Bound:
 - 3 znaleziono realizowalny plan \rightarrow bieżąca lista realizowalnym rozwiązaniem;

79 / 91

Algorytm Bratley'a

- Wykorzystując mechanizm branch-and-bound odnajduje realizowalne rozwiązanie, jeśli istnieje.
- ullet Bazuje na odpowiedniej permutacji zestawu zadań J_1,\ldots,J_n .
- Rozpoczyna od pustej listy (kolejności) zadań;
- Branch: wybierz do listy kolejne zadanie (z pozostałych)
- Bound:

 - ② jeśli istnieje zadanie, którego deadline minął, a nie zostało jeszcze zaplanowane → bieżąca ścieżka to nierealizowalny plan (cofnij do ostatniego dokonanego wyboru).

Notes		
Notes		
ivotes		
Notes		
Notes		

Algorytm Bratley'a

- Wykorzystując mechanizm branch-and-bound odnajduje realizowalne rozwiązanie, jeśli istnieje.
- ullet Bazuje na odpowiedniej permutacji zestawu zadań J_1,\ldots,J_n .
- Rozpoczyna od pustej listy (kolejności) zadań;
- Branch: wybierz do listy kolejne zadanie (z pozostałych)
- Bound:
 - 3 znaleziono realizowalny plan \rightarrow bieżąca lista realizowalnym rozwiązaniem;
 - ② jeśli istnieje zadanie, którego deadline minął, a nie zostało jeszcze zaplanowane → bieżąca ścieżka to nierealizowalny plan (cofnij do ostatniego dokonanego wyboru).

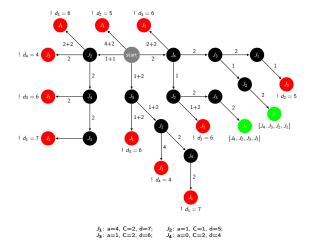
81/91

Algorytm Bratley'a

- Wykorzystując mechanizm branch-and-bound odnajduje realizowalne rozwiązanie, jeśli istnieje.
- ullet Bazuje na odpowiedniej permutacji zestawu zadań J_1,\dots,J_n .
- Rozpoczyna od pustej listy (kolejności) zadań;
- Branch: wybierz do listy kolejne zadanie (z pozostałych)
- Bound:
 - 3 znaleziono realizowalny plan o bieżąca lista realizowalnym rozwiazaniem:
 - ② jeśli istnieje zadanie, którego deadline minął, a nie zostało jeszcze zaplanowane → bieżąca ścieżka to nierealizowalny plan (cofnij do ostatniego dokonanego wyboru).

	J_1	J_2	J_3	J_4
ai	4	1	1	0
C_i	2	1	2	2
d:	7	5	6	4

82 / 91



83/91

Algorytm Bratley'a - własności

• Złożoność wykładnicza – tylko jako algorytm offline;

Notes	
Notes	
Notes	
Notes	
-	

Algorytm Bratley'a - własności Notes • Złożoność wykładnicza – tylko jako algorytm offline; Odkryty schemat wykonania zapisany jako lista i odczytywany w kolejności przez planistę w trakcie działania systemu. 85 / 91 Algorytm Bratley'a - własności Notes • Złożoność wykładnicza – tylko jako algorytm offline; Odkryty schemat wykonania zapisany jako lista i odczytywany w kolejności przez planistę w trakcie działania systemu. • Realizacja – dobry przykład zastosowania struktury drzewa w programowaniu :-) 86 / 91 Planowanie zadań z wymaganą kolejnością wykonania Notes Planowanie bez wywłaszczenia, z asynchronicznymi pojawieniami się zadań, czasami wykonania i wymaganą kolejnością jest NP-trudne. 87 / 91 Planowanie zadań z wymaganą kolejnością wykonania 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 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...
- Np. dla przykładu gdzie wszystkie zadania są dostępne od razu.

89 / 91

Planowanie zadań z wymaganą kolejnością wykonania

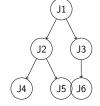
- Planowanie bez wywłaszczenia, z asynchronicznymi pojawieniami się zadań, czasami wykonania i wymaganą kolejnością jest NP-trudne.
- Dla pewnych rozluźnień problemu można znaleźć optymalne rozwiązania...
- Np. dla przykładu gdzie wszystkie zadania są dostępne od razu.

90/91

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...
- Np. dla przykładu gdzie wszystkie zadania są dostępne od razu.

	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



91/91

Notes	
Notes	
Notes	
Nistra	
Notes	