

45 Algorytm RMS (Rate Monotonic Scheduling) szeregowania zadań w systemie operacyjnym czasu rzeczywistego

Algorytm RMS (ang. Rate Monotonic Scheduling) szeregowanie uwzględniające częstość wykonywania danego zadania.

Założenia algorytmu RMS

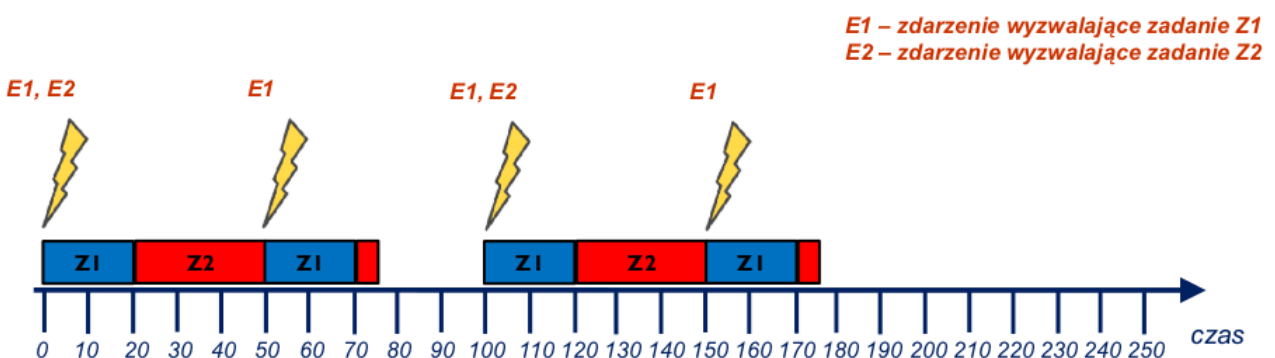
1. Wszystkie zadania są okresowe periodyczne (pojawiają się w regularnych odcinkach czasu) ,
2. Zadania nie synchronizują się między sobą w trakcie współdzielenia zasobów i komunikacji ,
3. Procesor zawsze wykonuje zadanie o najwyższym priorytecie, które jest gotowe do pracy (powinien być wykorzystany system z wywłaszczaniem) ,
4. Procesy są planowane na podstawie statycznego priorytetu równego im krótszy okres zadania, tym wyższy jego priorytet .

(ZRODLA/45/W11_W12_SCR.pdf)

Jądro systemu – Szeregowanie zadań – algorytm RMS

Przykład dwóch zadań o następujących parametrach:

- ▶ zadanie pierwsze – Z1 : $p = 50$; $d = 50$; $t = 20$;
- ▶ zadanie drugie – Z2 : $p = 100$; $d = 100$; $t = 35$;

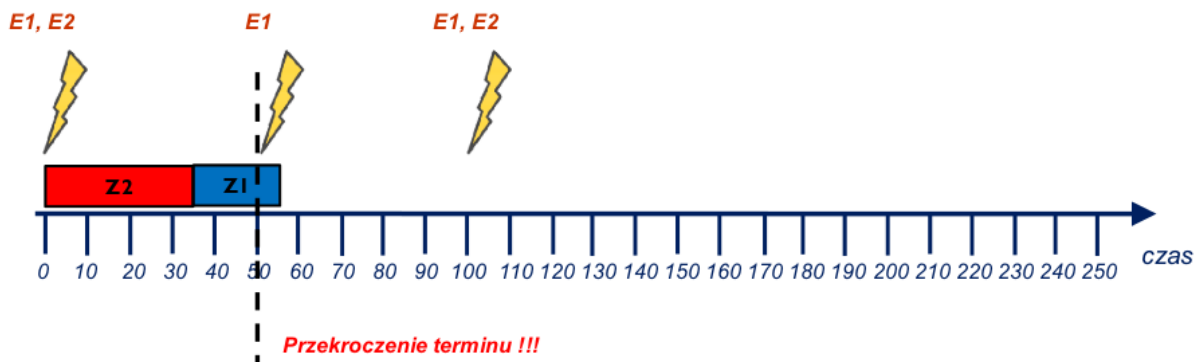


Całkowite obciążenie procesora: $u = (20/50) + (35/100) = 0.75$

Jądro systemu

- Szeregowanie zadań - algorytm RMS

Zakładamy chwilowo, że zadanie drugie ma większy priorytet od zadania pierwszego



Jądro systemu

- Szeregowanie zadań - algorytm RMS

Przykład dwóch zadań o następujących parametrach:

- ▶ zadanie pierwsze – Z1: $p = 50$; $d = 50$; $t = 25$;
- ▶ zadanie drugie – Z2: $p = 80$; $d = 80$; $t = 35$;



Całkowite obciążenie procesora: $u = (25/50) + (35/80) = 0.94$!!!

Jądro systemu
– Szeregowanie zadań – algorytm RMS

Podstawowe twierdzenie RMS:

wszystkie solidne limity czasowe systemu SOCR będą zawsze spełnione jeżeli:

$$\sum_i \frac{t_i}{p_i} \leq n \cdot \left(2^{\frac{1}{n}} - 1 \right)$$

gdzie:

n – dany zestaw zadań z przypisanymi priorytetami RMS,

t_i – maksymalny czas wykonania zadania i ,

p_i – okres wykonania zadania i ,

t_i / p_i – część czasu procesora wymagana do wykonania zadania i .

Zgodnie ze statycznym doбором priorytetów algorytmem RMS, im krótszy okres zadania, tym wyższy jego priorytet.

Jądro systemu
– Szeregowanie zadań – algorytm RMS

Liczba Zadań	$n \cdot \left(2^{\frac{1}{n}} - 1 \right)$
1	1.000
2	0.828
3	0.779
4	0.756
5	0.743
...	...
∞	0.693

Aby spełnić wszystkie limity czasowe systemu o „solidnych” wymaganiach czasowych w oparciu o algorytm RMS, wykorzystanie CPU przez wszystkie zadania krytyczne czasowo powinno być mniejsze niż 70 procent !!!

Nadal można mieć zadania niekrytyczne czasowo, a więc wykorzystać 100 procent czasu CPU
