

Algorytm Baera

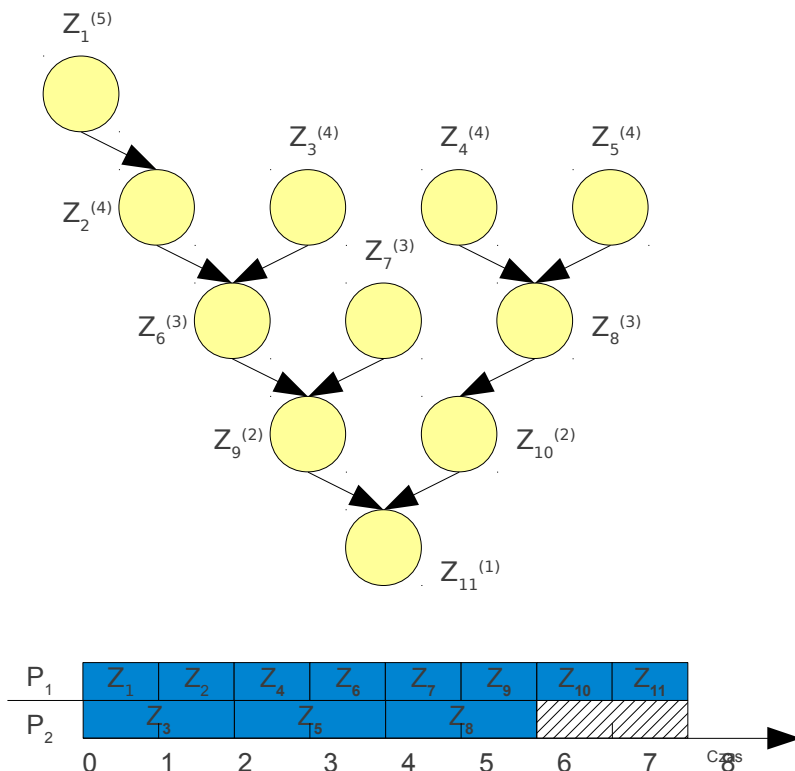
minimalizacja długości uszeregowania dla procesorów jednorodnych zadania zależne, niepodzielne

Założenia:

- w systemie są dwa procesory o współczynnikach prędkości $b_1 = 2, b_2 = 1$
- standardowe czas wykonywania zadań $\tau_j = 2, j = 1, 2, \dots, n$
- zbiór zadań tworzy antydrzewo

Idea tego algorytmu jest oparta na twierdzeniu, że dla tak określonego zbioru zadań i procesorów, w rozwiązaniu optymalnym może wystąpić przestój tylko procesora o mniejszym współczynniku b . Oznaczamy przez k_1 i k_2 liczniki czasu wykonywania zadań odpowiednio dla procesora P_1 (szybszego) i P_2 . Przebieg algorytmu Baera jest następujący.

1. Określ poziomy zadań, licznik zadań $q := n, k_1 := k_2 := 0$
2. Przydziel dostępne w chwili k_1 zadanie o najwyższym poziomie do procesora $P_1, q := q-1, k_1 := k_1 + 1$
3. Jeśli są zadania dostępne w chwili k_2 , to przydziel jedno z nich o najwyższym poziomie do procesora $P_2, q := q-1, k_2 := k_2 + 2$
4. Jeśli są zadania dostępne w chwili k_1 , to przydziel zadanie o najwyższym poziomie do procesora $P_1, q := q-1, k_1 := k_1 + 1$. Jeśli $q = 1$, to przejdź do kroku 5, w przeciwnym przypadku powtórz krok 2
5. Przydziel ostatnie zadania do procesora $P_1, C_{max}^* := k_1 + 1$



Przykład zastosowania algorytmu Baera (na grafie w nawiasach podano poziomy zadań)

Złożoność obliczeniowa tego algorytmu jest $O(n)$, gdyż każde zadania jest rozpatrywane tylko raz, a czas z tym związany jest stały.