

Algorytm Kafury i Shena

minimalizacja długości uszeregowania zadania zależne i podzielne procesory z niezależnymi pamięciami

wprowadzenie

Zakłada się, że każdy z m identycznych procesorów P_1, P_2, \dots, P_m ma swoją własną (dedykowaną) pamięć o rozmiarze odpowiednio $|P_1|, |P_2|, \dots, |P_m|$, przy czym $|P_1| \geq |P_2| \geq \dots \geq |P_m|$. Zadanie Z_j ($j=1, 2, \dots, n$) do swego wykonania potrzebuje jednego procesora oraz $r_1(Z_j)$ jednostek pamięci, może zatem być wykonywane jedynie na tych procesorach, które mają pamięć o wystarczającym rozmiarze. Oznaczmy dalej przez Z_i ($i=1, 2, \dots, m$) zbiór tych zadań, które ze względu na ograniczenie ze strony pamięci mogą być wykonywane tylko na procesorach P_1, P_2, \dots, P_i , a przez t_i sumę czasów wykonywania zadań należących do zbioru Z_i .

algorytm

1. Ustaw zadania na liście w kolejności nie rosnących wartości zapotrzebowania na pamięć
2. Wyznacz zbiory Z_i oraz czasy t_i ($i=1, 2, \dots, m$). Podstaw
$$C_x := \max_i \left\{ \frac{t_i}{i} \right\}, \tau_{max} := \max_j \{ \tau_j \}, C_{max}^* := \max \{ C_x, \tau_{max} \}$$
 oraz $j := 1$
3. Jeśli przydzielono już wszystkie zadania to zakończ procedurę. W przeciwnym razie wybierz następne nieprzydzielone zadanie na liście i przydziel je do procesorów zgodnie z algorytmem Mc Naughtona