

Algorytm Mc Naughtona

minimalizacja długości uszeregowania dla procesorów identycznych zadania niezależne, podzielne

1. Rozpocznij wykonywanie dowolnego zadania na dowolnym procesorze w chwili $t=0$
2. Wybierz dowolne nie uszeregowane zadanie i rozpocznij jego wykonywanie na tym samym procesorze w chwili zakończenia wykonywania poprzedniego zadania. Powtarzaj ten krok do chwili gdy wszystkie zadania zostaną uszeregowane lub $t=C_{max}^*$
3. Część zadania pozostającą do wykonania po osiągnięciu $t=C_{max}^*$ przydziel do innego procesora rozpoczynając jej wykonanie od chwili $t=0$. Wróć do kroku drugiego.

Łatwo zauważyć, że powyższy algorytm zawsze znajduje uszeregowanie, a jego optymalność wynika z faktu, że długość tego uszeregowania jest zawsze określona wzorem

$C_{max}^* = \max \{ \max_j \{ \tau_j \}, \frac{1}{m} \sum_{j=1}^n \tau_j \}$, a zatem jest minimalna. Złożoność tego algorytmu jest

$O(n)$, gdyż każde zadanie jest w nim rozpatrywane tylko raz, a czas z tym związany jest stały, niezależny od liczby zadań.

Przykłady optymalnych uszeregowień dla $n=5, m=3$;

