

**Problem szeregowania zadań na
równoległych procesorach**

Adrian Derda,

Michał Karmański,

Mateusz Adamczak,

Bartosz Kacperkiewicz

Spis treści

2014-04-11

POLITECHNIKA WROCŁAWSKA

WYDZIAŁ ELEKTRONIKI

Zarządzanie w systemach i sieciach komputerowych - projekt

Problem szeregowania zadań na równoległych procesorach

AUTOR:

Adrian Derda,

Michał Karmański,

Mateusz Adamczak,

Bartosz Kacperkiewicz

PROWADZĄCY ZAJĘCIA:

Dr inż. Robert Wójcik, W4/I-6

Problem szeregowania zadań na równoległych procesorach

Adrian Derda,

Michał Karmański,

Mateusz Adamczak,

Bartosz Kacperkiewicz

Spis treści

2014-04-11

OCENA PRACY:

Spis treści

Zarządzanie w systemach i sieciach komputerowych - projekt	1
1. Sformułowanie problemu	1
1.1 Podłoże projektu	1
2. Metoda i algorytmy rozwiązywania problemu.....	1
3. Metoda implementacji	1
4. Sposób testowania i ocena jakości rozwiązań.....	2
Literatura	3

Adrian Derda,
Michał Karmański,
Mateusz Adamczak,
Bartosz Kacperkiewicz

1. Sformułowanie problemu

1.1 Podłoże projektu

Jako temat naszego projektu wybraliśmy szeregowanie zadań na równoległych procesorach. Jest to zagadnienie o tyle ciekawe, że korzystamy z niego na co dzień w każdym komputerze posiadającym więcej niż jeden procesor. Specjalnie opracowane algorytmy umożliwiają taki rozdział procesów względem procesorów, aby uzyskać możliwie jak najkrótszy czas przetwarzania zadania przy jednoczesnym maksymalnym wykorzystaniu procesora.

2. Metoda i algorytmy rozwiązywania problemu

Do rozwiązania problemu zostaną wykorzystane najbardziej rozpowszechnione algorytmy służące szeregowaniu zadań. Będą to:

- a. FCFS (First-Job First-Served) [1]
- b. SJF (Shortest Job First) [2]
- c. SRT (Shortest Remaining Time) [3]
- d. Algorytm priorytetowy [1]
- e. Algorytm rotacyjny [4]

3. Metoda implementacji

Program zostanie zaimplementowany w języku programowania C# wraz z platformą .NET. Na samym początku zostaną zaimplementowane algorytmy służące do szeregowania zadań na procesorach, a następnie po ich przetestowaniu zostanie wykonany graficzny interfejs użytkownika (GUI). W ramach jego działania użytkownik będzie miał dostęp do pełnej funkcjonalności stworzonego uprzednio programu.

Adrian Derda,
Michał Karmański,
Mateusz Adamczak,
Bartosz Kacperkiewicz

2014-04-115

4. Sposób testowania i ocena jakości rozwiązań

Ocena jakości rozwiązań opierać się będzie o wyliczenia parametrów dla poszczególnych instancji problemów. Badane wskaźniki to:

- średni czas wykonania pojedynczego zadania dla różnych instancji problemu.
- średni czas oczekiwania zadania na wykonanie.
- procentowy czas zajętości procesora.
- średni termin zakończenia wszystkich zadań

Taki sposób analizy odzwierciedla aspekty które są obserwowane bezpośrednio przez użytkownika (zauważa on opóźnienia pracy systemu itp.) co pozwoli nam określić jakość pracy z systemem opierającym się o badane rozwiązania, z drugiej strony dane te pozwolą na oszacowanie wynikowego stopnia efektywności procesora.

W przypadku algorytmu karuzelowego analizując opisane wskaźniki postaramy się ustalić optymalny kwant czasu przełączania kontekstu

Sposób testowania opierał się będzie o generowanie instancji problemu, która następnie rozwiązana zostanie przez założone algorytmy, otrzymane dane w postaci wskaźników opisanych powyżej będą ze sobą zestawiane i analizowane pod kątem wspomnianej jakości pracy i efektywności procesora - tak spreparowane informacje pozwolą wysnuć pewne wnioski na temat wad i zalet badanych metod, jak również określać ich przydatność w danym środowisku.

2014-04-115

Adrian Derda,
Michał Karmański,
Mateusz Adamczak,
Bartosz Kacperkiewicz

Literatura

- [1] Chrobot A., Systemy operacyjne – zarządzanie procesami,
<http://achilles.tu.kielce.pl/Members/achrobot/archiwum/i-semestr/pdf/so/> (aktualne na dzień 8.03.2014)
- [2] Nowak S., Dąbrowski W., Michalak G.,
<http://www.isep.pw.edu.pl/~slawekn/info1/lekcja5/segment4/main.html>, (aktualne na dzień 01.04.2014)
- [3] Hasyim N., Shortestjob first algorithm,
<http://naharhasyim.blogspot.com/2013/03/shortest-job-first-algorithm.html>,
(aktualne na dzień 02.04.2014)
- [4] Przykłady algorytmów planowania przydziału procesora,
<http://smurf.mimuw.edu.pl/node/895>, (aktualne na dzień 01.04.2014)
- [5] Pokrywka M., Algorytm karuzelowy (round-robin)
<http://akkd.porubis.pl/bezklasowe/round-robin> (aktualne na dzień 01.04.2014)