

ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

Харківський національний університет
радіоелектроніки

каф. ЕОМ

Система для моделювання багатопроцесорних
алгоритмів планування

Ст. групи КІУКІз-21-1
Олександренко М.А.

Керівник
Радченко В.О.

Мета та задачі роботи

Мета кваліфікаційної роботи створення системи моделювання багатопроцесорних алгоритмів планування

Задачі:

- Розглянути поняття процесу та підходи до багатопроцесорного планування
- Провести аналіз моделей для багатопроцесорного планування
- Розробити та провести тестування системи на основі оновленої моделі в середовищі Matlab

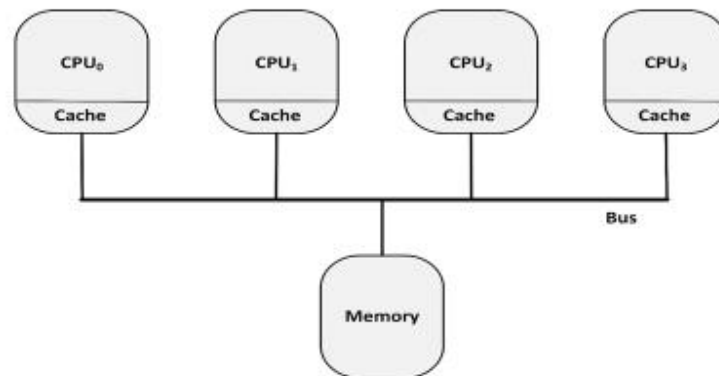
Процес

Операційні системи, такі як Windows та Linux, використовують абстракцію, відому як процес, для виконання програм. Ця абстракція створює ілюзію, що кожен процес має свій власний виділений процесор, навіть якщо він використовує один або кілька процесорів спільно з іншими процесами. Як правило, процес може існувати в одному з трьох різних станів:

- виконується, цей стан вказує на те, що процес активно виконується на процесорі;
- готовий, тут процес готовий до запуску, але ще не був обраний планувальником;
- заблоковано/очікує, у цьому стані процес зазвичай виконав певну операцію вводу/виводу та наразі заблоковано або очікує.

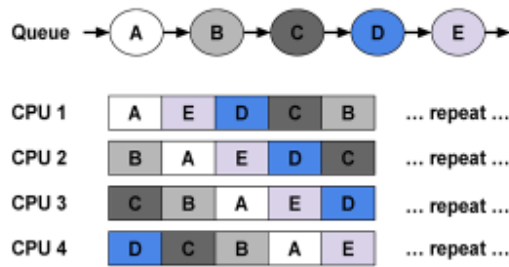
3

Багатопроцесорне планування

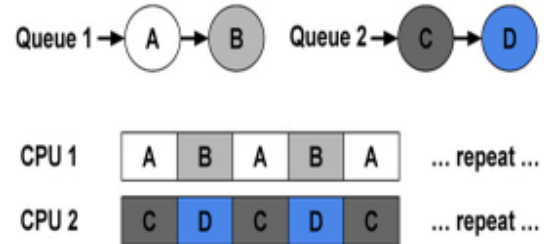


4

Підходи до багатопроцесорного планування



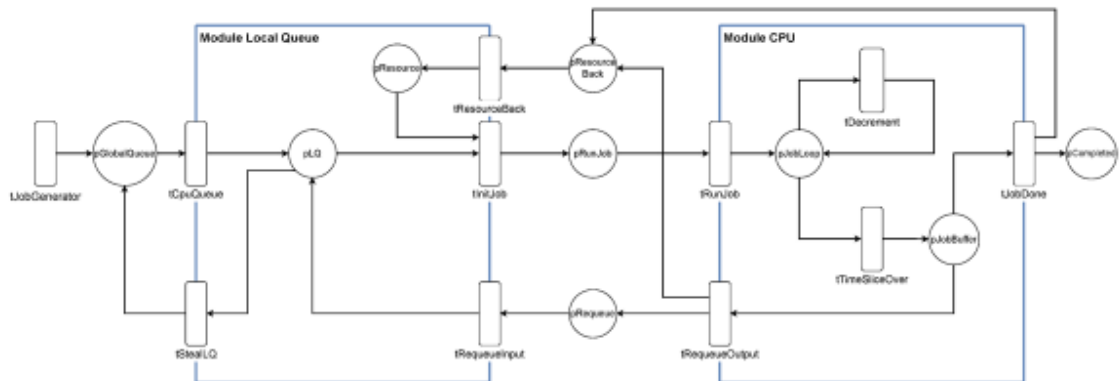
Приклад багатопроцесорної системи з п'ятьма завданнями, що використовує одну чергу.



Приклад багатопроцесорної системи з чотирма завданнями, що використовує багаточергову конфігурацію.

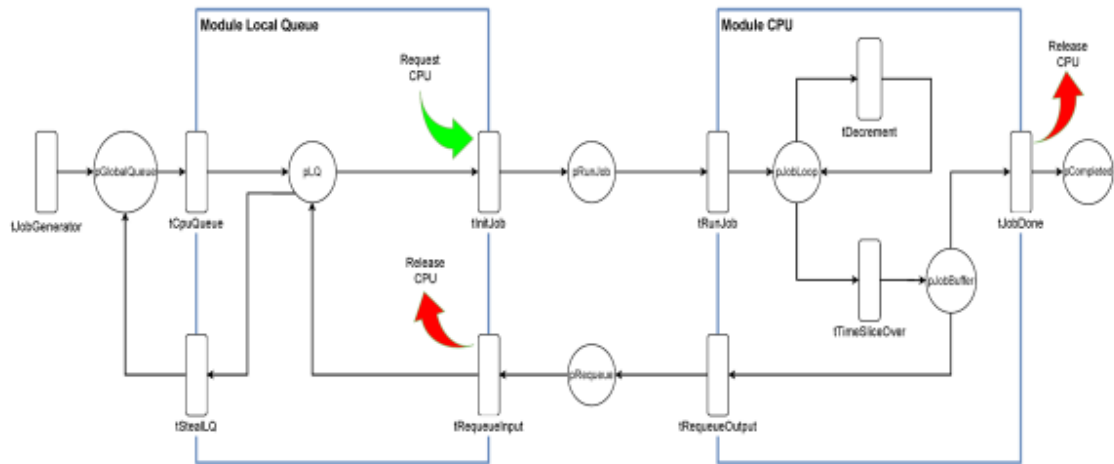
5

Початкова модель



6

Оновлена модель

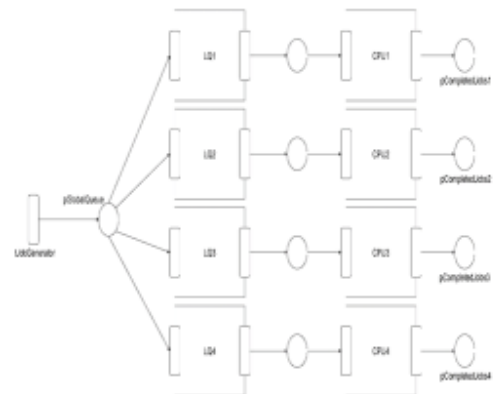


7

Спрощений огляд чотирьох процесорів з використанням попередньої моделі

На рисунку показано конфігурацію моделі з чотирма процесорами. Дублювання компонентів $pGlobalQueue$ та $tJobGenerator$ не є необхідним, оскільки вони повинні бути спільними для процесорів для розподілу завдань.

Загальний потік багатопроцесорної моделі залишається узгодженим з однопроцесорною моделлю, з двома переходами, що потребують модифікації: $tCpuQueue$ та $tStealLQ$. В середовищі з одним процесором $tCpuQueue$ може завжди спрацювати, а $tStealLQ$ може взагалі не спрацювати. Однак ці переходи є критично важливими для розподілу та перерозподілу завдань у багатопроцесорному середовищі. Перехід $tCpuQueue$ для процесора повинен спрацювати лише тоді, коли його локальна черга має найменшу кількість завдань. Це забезпечує справедливий розподіл завдань, мінімізуючи час простою системи.



8

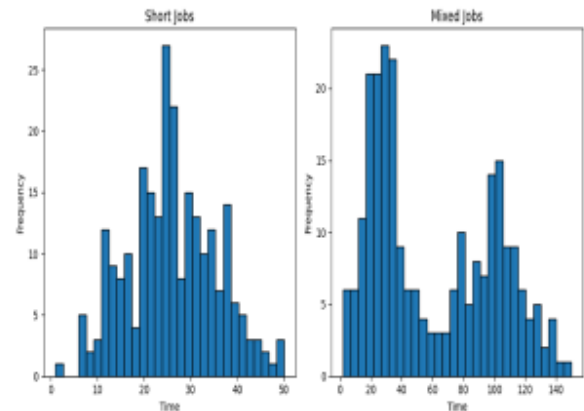
Тестування та результати

Набори даних, що використовуються в симуляціях, складаються з двох різних наборів:

- набір даних коротких завдань, цей набір даних складатиметься із завдань з відносно коротким часом виконання. Ці завдання призначені для імітації процесів, які швидко завершуються.
- змішаний набір даних завдань, навпаки, «змішаний набір даних завдань» забезпечує складніше та реалістичніше тестове середовище. Він моделює сценарій, де процесор повинен керувати поєднанням короткострокових та довгострокових завдань, що дуже нагадує реальні ситуації, де процеси з різним часом виконання постійно конкурують за системні ресурси. Цей набір даних допомагає нам оцінити, наскільки добре наші алгоритми планування адаптуються до викликів, що виникають через різноманітний набір завдань.

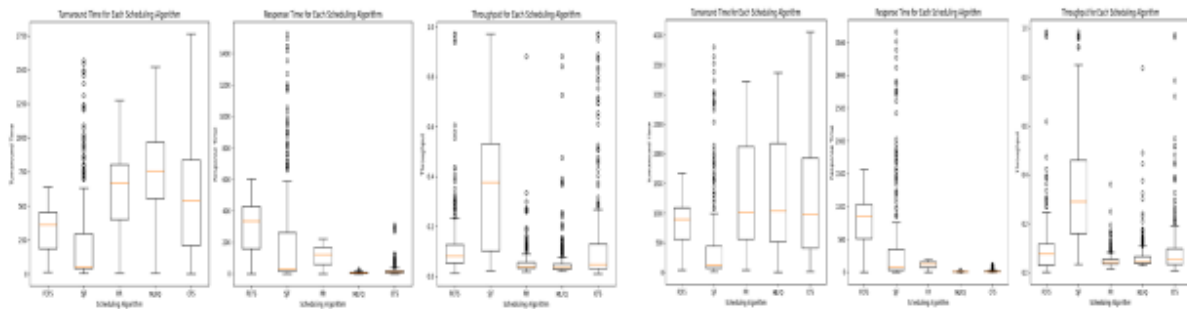
Кожен набір даних містить 250 завдань, причому короткий набір даних містить виключно короткі завдання. Змішаний набір даних складається з рівної кількості 50% коротких і 50% довгих завдань. У змішаних наборах даних короткі завдання представлені непарними ідентифікаторами завдань, тоді як парні ідентифікатори завдань відповідають довгим завданням.

На рисунку наведено візуалізацію розподілу часу виконання завдань для обох наборів даних.



9

Оцінювання різних алгоритмів планування з використанням метрик



Коробчаста діаграма показників з короткими завданнями

Коробчаста діаграма показників зі змішаними завданнями

10

Висновки

В ході кваліфікаційної роботи було створенно систему моделювання багатопроцесорних алгоритмів планування

Задачі які було вирішено:

- Розглянуто поняття процесу та підходи до багатопроцесорного планування
- Проведено аналіз моделей для багатопроцесорного планування
- Розроблено та проведено тестування системи на основі оновленої моделі в середовищі Matlab