DOI: 10.32347/2412-9933.2020.42.43-50
УДК 004. 2

Зайцев Володимир Григорович
Доктор технічних наук, професор, професор кафедри системного програмування і спеціалізованих комп'ютерних систем, *orcid.org/0000-0001-9548-1959*
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ
Цибаєв Євгеній Ігорович
Кандидат технічних наук, кафедра системного програмування і спеціалізованих комп'ютерних систем, *orcid.org/0000-0002-9115-2346*
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ

ОЦІНКА ЧАСОВИХ ХАРАКТЕРИСТИК ЗАДАЧ В БАГАТОПРОЦЕСОРНИХ СИСТЕМАХ РЕАЛЬНОГО ЧАСУ З ВИКОРИСТАННЯМ СІТОК ПЕТРІ

Анотація. *Розглянуто проблему визначення часових характеристик задач у системах реального часу, успішність роботи яких залежить не тільки від їх логічної правильності, а і від часу, за який вони отримують результат. Визначення таких часових характеристик системи на стадії проєктування є досить складною проблемою. Її вирішення на сьогодні засновано на двох основних напрямах: теоретичних розрахунках, пов'язаних з отриманням так званих критеріїв здійсненності і моделюванням роботи системи на моделях. Серед моделей найбільш розповсюдженими є статистичні моделі систем масового обслуговування. Однак, як у першому, так і у другому випадках неможливо отримати гарантований результат, що суттєво ускладнює процес проєктування. Останнім часом з метою моделювання запропоновано використовувати моделі, що засновані на застосуванні апарату сіток Петрі. Але проведені дослідження стосуються лише моделювання однопроцесорних систем. Запропоновано метод оцінювання часових характеристик задач в системах реального часу шляхом аналізу даних, отриманих моделюванням розподілу процесорного часу між задачами згідно обраних алгоритмів планувальника з використанням моделі сіток Петрі для багатопроцесорних систем. Метод гарантує отримання часових характеристик задач при обранні конкретних типів процесорів і планувальника, що потрібно для початку технічного проєктування багатопроцесорних системи реального часу.*

Ключові слова: модель; задача реального часу; сітка Петрі; багатопроцесорна система

Вступ

Робота присвячена проблемі визначення часових характеристик задач у комп'ютерних системах реального часу. Успішність роботи таких систем залежить не тільки від логічної правильності програми, але і від часу виконання та моментів отримання результату [1; 2]. Якщо наперед задані часові обмеження не виконуються, то фіксується похибка у роботі системи.

Актуальність дослідження

Гарантування успішної роботи системи реального часу, що проєктується, є досить складною задачею і пов'язана з оптимізацією комплексного вибору технічних засобів, функціональних алгоритмів і операційної системи.

Розв'язання цієї багатовимірної оптимізаційної задачі на стадії проєктування системи базується на двох основних напрямах: теоретичних дослідженнях, метою яких є отримання так званих критеріїв здійсненності, які мають гарантувати обраному варіанту реалізації дотримання необхідних часових характеристик [3; 6], і підтвердження працездатності системи шляхом моделювання її роботи у різних режимах експлуатації.

Слід зазначити, що отримання критеріїв здійсненності пов'язане з досить складними теоретичними дослідженнями, які часто гарантують лише принципову можливість побудови системи із заданими характеристиками при виборі певних методів реалізації, але не гарантують їх отримання у кожному конкретному випадку. Тому отримання і перевірка критеріїв здійсненності може розглядатись лише як перший етап проєктування конкретної системи[4; 5], а на другому етапі виникає необхідність у моделюванні роботи системи з метою остаточного вибору варіанта реалізації за результатами моделювання.

Такі моделі можуть бути побудовані відповідно до двох основних методів:

1) статистичні моделі, які засновані на використанні моделей масового обслуговування [1];

2) моделі, що побудовані з використанням апарату сіток Петрі[7].

Перший тип моделей гарантує отримання прогнозованих середньостатистичних значень параметрів, що може бути достатнім для м'яких систем реального часу [2], але цього недостатньо для проєктування жорстких систем реального часу.

Другий тип моделей засновано на ідеї надання задачам процесорного часу деякою послідовністю квантів часу, що імітується передачею маркерів з однієї позиції сітки Петрі в іншу. При цьому сумарна кількість квантів у одній позиції імітує наявний сумарний процесорний час, а друга – кількість процесорного часу, що отримала задача, якій відповідає друга позиція[8].

Така роздача квантів часу (маркерів) відбувається за вказівками планувальника і диспетчера операційної системи, робота яких теж імітується.

Останні дослідження

Побудову моделей другого типу наведено у дослідженні [9; 10].В роботі[8] наведено принцип побудови моделей для визначення часових характеристик задачу системах реального часу шляхом моделювання процесу розподілу процесорного часу між задачами за вказівками планувальника. Показано, як змінюються часові характеристики задач при зміні типу планувальника і продуктивності процесора.

Однак це дослідження розглядає особливості побудови тільки моделей однопроцесорних систем.

Перспектива широкого впровадження багатоядерних процесорів при створенні систем реального часу вимагає проведення досліджень з розробки ефективних методів моделювання розподілу процесорного часу серед задач багатопроцесорних систем.

Мета дослідження

Метою дослідження є створення моделей, що засновані на використанні багатопроцесорних або багатоядерних процесорів. При цьому зауважимо, що при розподілі процесорного часу між задачами можна реалізувати декілька варіантів такого розподілу:

1) статичний розподіл задач між процесорами;

2) міграція задач між процесорами;

3) міграція виконуваної задачі між процесорами.

У першому варіанті процесори працюють незалежно один від одного. Проблеми вибору дисципліни планування та оцінки часових характеристик задач визначаються незалежно на моделях для кожного окремого процесора. Кожна модель – складова у цьому випадку функціонує незалежно. Їх робота повністю збігається з роботою окремої однопроцесорної моделі [8].

Другий варіант більш гнучкий. Тут вважається, що задача може мігрувати між процесорами, обираючи кожного разу інший, якщо це доцільно. Але при цьому, якщо виконання задачі переривається у процесі надання процесорного часу, то поновлення її виконання відбувається тільки на тому самому процесорі. Тобто, у цьому випадку кожна задача жорстко не прив'язується до конкретного процесора, окрім того, різні копії однієї і тієї ж задачі можуть бути поставлені на різні процесори. При цьому час виконання однієї і тієї ж задачі може змінюватись у випадку різної продуктивності процесорів. Така додаткова гнучкість при динамічному розміщенні задач може забезпечити більш високу ефективність використання процесорного часу.

Третій варіант – найбільш гнучкий. Тут є можливість активній задачі мігрувати між процесорами у процесі свого виконання. Така міграція можлива, скажімо, після переривання та поновлення виконання задачі. Реалізація такого варіанта пов'язана з додатковою умовою можливості підтримки відповідною апаратною архітектурою. Наприклад, процесори мають працювати в одному адресному середовищі – коди програм та даних мають бути доступними кожному з процесорів, що опрацьовують відповідний додаток, тобто являти собою симетричну багатопроцесорну систему.

У цьому випадку методи планування, оцінка критеріїв здійсненності та принципи моделювання виконання задач можуть бути використані аналогічно раніше розробленим для цього методам і у подальшому розглядатись не будуть. Тобто у подальшому зупинимось на розгляді моделей, що використовують перші два варіанти.

Зупинимось на модельних параметрах задач реального часу, які будуть розглядатись у подальшому.

Ці модельні параметри: r – (Realize Time) – момент часу, коли виникає необхідність в передачі управління виконуваній задачі A; d – (Absolute Deadline) – абсолютний крайній термін, момент часу, до якого задача повинна завершити роботу; s – (Start Time) – момент часу, коли задача фактично починає виконуватись на процесорі; c – (Completion Time) – момент часу, коли задача закінчила роботу, обробивши подію (розв'язавши задачу); D – (Relative Time) – відносний крайній термін: $D = d - r$; E – (Execution Time) – час виконання задачі: $e = c - s$; R – (Response Time) – час відгуку: $R = c - r$; T – період (квазіперіод) існування задач визначаються на попередніх етапах розроблення системи.

Для конкретного вибору номера задачі та номера процесора, на якому вона виконується, у подальшому застосуємо індексацію, де будемо використовувати індекси *j* для визначення номера процесора та i для визначення номера задачі. Наприклад, e_{ji} – час виконання задачі A_i на процесорі j і т.д.

Загальну структуру моделі задач A_i на процесорі j представлено сіткою Петрі, як на рис. 1. Вона складається із сукупності задач A_i, кожна з яких має у своєму складі загальну для всіх задач обраного процесора позицію P_{jo}. Детально робота такої моделі розглядається у [8].

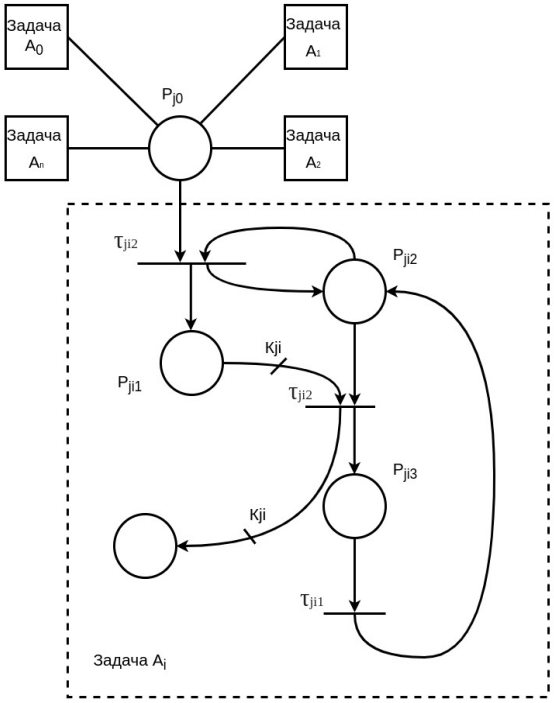

Рисунок 1 –Модель задачі A_i на процесорі j

Нагадуємо, що сітка Петрі являє собою сукупність позицій (кружечків) та планок (рисок) – переходів τ_{jim}, де $m = 1 \div 3$. Орієнтовані дуги поєднують позиції і переходи. Дуга, що направлена від позиції P до переходу τ визначає позицію, що являє собою вхід у перехід. Вихідна позиція вказується дугою від переходу до позиції. Виходи та входи можуть бути кратними, що визначається на схемі кількістю дуг, або (як на наведеному рисунку) цифрою k_{ji} на дузі [7]. Тут індекс i відповідає номеру задачі A_i, а індекс j – номеру процесора. Кожна з позицій може мати певну кількість маркерів, які можуть рухатись з однієї позиції на іншу при виконанні (запуску) переходів [9]. При аналітичному способі визначення сіток Петрі [7] їх задають так:

$$S_j = (P_j, \theta_j, F_j, H_j, \mu_0), \qquad (1)$$

де P_j – множина вершин моделі (процесора) *j*;

θ_j – множина переходів моделі (процесора) *j*;

F_j – множина вхідних позицій моделі (процесора) *j*;

H_j – множина вихідних позицій моделі (процесора)*j*;

μ_0 – початкове маркування(початковий розподіл маркерів по позиціях) кожної з моделей(процесора)*j*).

Маркування сітки S_j – це присвоювання певної кількості маркерів, що перебувають у позиціях P_{ji}. Цю кількість маркерів позначимо [P_{ji}].

Маркування μ може розглядатись як n – мірний вектор $\mu = (\mu_1, \mu_2, \ldots \ldots \mu_n)$, де μ – розподіл фішек серед позицій P у кожному конкретному стані сітки.

Зміна маркування сітки відбувається шляхом запуску переходів. Вона здійснюється відповідно до деяких подій, що визначає процес, який моделюється сіткою.

Переходи можуть бути дозволеними і недозволеними. Перехід називають дозволеним, якщо кожна з його вхідних позицій має кількість маркерів не менше, ніж кількість дуг із позиції в перехід. При запуску дозволених переходів маркування сітки змінюється, а недозволених залишається незмінним(попереднім).

Продемонструємо це на прикладі роботи моделі задачі A_i процесора j, яка представлена на рис.1.

Для цього за аналогією з [8] скористуємось аналітичним представленням сітки Петрі у матричній формі:

$$S_j = (P_j, \theta_j, M_{ji}^+, M_{ji}^-, \mu_0). \qquad (2)$$

Відповідно визначимо нове маркування після запуску переходів:

$$\mu_{i+1} = \mu_i + B_m \times M_{ji}, \qquad (3)$$

де M_{ji} – матриця інцедентності, яка дорівнює:

$$M_{ji} = M_{ji}^+ - M_{ji}^-, \qquad (4)$$

m – номер переходу τ_{jim}; B_m – одиничний вектор – рядок переходу τ_{jim}, усі компоненти якого дорівнюють нулю, за винятком компонента, що відповідає номеру m та дорівнює одиниці.

Виникає питання, яким чином при обчисленнях встановити спрацював перехід при запуску чи ні.

Якщо перехід спрацьовує, то свідченням цього є те, що при обчисленні нового маркування ні в одній позиції не може бути від'ємної кількості маркерів. Наявність таких від'ємних результатів свідчить про

те, що запущений перехід не спрацьовує і попереднє маркування має залишитись незмінним.

Матриця M_{ji}^{+} має вигляд, як табл. 1, M_{ji}^{-} – як табл. 2, а матриця M_{ji}, як табл. 3.

*Таблиця 1 – **Матриця** M_{ji}^{+}*

	P_{j0}	P_{ji1}	P_{ji2}	P_{ji3}	P_{ji4}
τ_{ji1}	0	0	1	0	0
τ_{ji2}	0	1	1	0	0
τ_{ji3}	0	0	0	1	k_{ji}

*Таблиця 2 –**Матриця** M_{ji}^{-}*

	P_{j0}	P_{ji1}	P_{ji2}	P_{ji3}	P_{ji4}
τ_{ji1}	0	0	0	1	0
τ_{ji2}	1	0	1	0	0
τ_{ji3}	0	k_{ji}	1	0	0

*Таблиця 3 –**Матриця** M_{ji}*

	P_{j0}	P_{ji1}	P_{ji2}	P_{ji3}	P_{ji4}
τ_{ji1}	0	0	1	-1	0
τ_{ji2}	-1	1	0	0	0
τ_{ji3}	0	$-k_{ji}$	-1	1	k_{ji}

Уявимо, що ресурс процесорного часу процесора j за період (квазіперіод) роботи задач системи пропорційний кількості маркерів, що перебувають у позиції P_{j0} при початковому маркуванні. Тоді передачу одного маркера з позиції P_{j0} у позицію P_{ji1} можна розглядати, як вилучення задачі i одного кванту процесорного часу розміром Δt процесором j.

Якщо позначити через N_j сумарну початкову кількість маркерів у позиції P_{j0} на початок періоду моделювання роботи задач системи, то кількість маркерів у цій позиції на певний момент можна використати як годинник, що фіксує умовний час передачі чергового маркера деякій задачі, а також час постановки її на виконання і час завершення виконання.

Аналізуючи роботу наведеної моделі задачі A_i (рис. 1), можна встановити, що така передача одного маркеру у позицію P_{ji1} пов'язана з послідовним запуском переходів τ_{ji1} , τ_{ji2} , τ_{ji3}. Кожну таку послідовність запуску у подальшому будемо називати тактом роботи моделі. Таким чином, виконання одного такту буде пов'язано з розходуванням одного кванту певного процесорного часу. Згідно рівняння (3) обчислимо значення компонентів вектора $B_m \times M_{ji}$, які додаються до попереднього маркування, якщо перехід спрацьовує. Вони дорівнюють:

$$\tau_{ji1}:[001\text{-}10],$$
$$\tau_{ji2}:[11000], \qquad (5)$$
$$\tau_{ji3}:[0\text{-}k_{ji}\text{-}11\ k_{ji}].$$

Позиції P_{ji2} та P_{ji3} слугують для визначення стану задачі A_i , що виконується на процесорі j . Якщо в позиції P_{ji3} знаходиться один маркер, а в позиції P_{ji2} нуль маркерів, то це відповідає початку накопичення маркерів у позиції P_{ji1} (початку надання процесором j процесорного часу задачі A_i). Якщо передача чергового кванта часу є продовженням накопичення маркерів (задача A_i продовжує виконання на процесорі j), то кількість маркерів у позиції P_{ji2} стає: [P_{ji2}] = 1, а[P_{ji3}] = 0. Наявність маркерів у позиції P_{ji4} свідчить про те, що задача A_i вже виконувалась на процесорі j протягом цього періоду, причому загальне число виконання α може бути обчислене так:

$$\alpha = [P_{ji4}] / k_{ji}. \qquad (6)$$

Моменти чергового запуску T_{izan} та завершення виконання задачі A_i – $T_{izав}$ можна встановити на кожному такті роботи моделі: $T_{izan} = N_{j0}$ – [P_{j0}] за виконання умови:[P_{j2}] = 1;

$T_{izав} = N_{j0} - [P_{j0}]$ за виконання умов: $[P_{ji3}]=1$; $[P_{ji4}] \neq 0$, де N_{j0} – кількість маркерів у позиціях P_{j0} на початок моделювання (початок циклу).

Звернімо увагу на те, що задачі можуть надходити при моделюванні процесу виконання у довільні моменти часу, тобто можлива ситуація, за якої процесори можуть бути не завантажені. Тим не менше, кількість маркерів в $[P_{j0}]$ має змінюватись, відраховуючи використання часу періоду. Це можна врахувати, якщо у складі кожної підмоделі j передбачити наявність деякої задачі A_0, час виконання якої дорівнює одному такту (k_{j0} =1) і яка

завжди буде виконуватись за відсутності у черзі готових до виконання реальних задач.

На відміну від однопроцесорної моделі у багатопроцесорній (рис. 2) виникає додаткове питання, пов'язане із синхронізацією часу роботи усіх однопроцесорних моделей та різною продуктивністю процесорів. Розподілом задач на виконання різними процесорами займається планувальник, але може виникнути ситуація, коли одна і та ж задача на різних процесорах може виконуватись за різні терміни у зв'язку з їх різною продуктивністю. Врахувати такі особливості можна, регулюючи співвідношення між величинами k_{ji} для різних підмоделей задачі A_i , а для синхронізації роботи підмоделей у часі виконати поєднання запуску послідовностей із трьох переходів τ_{ji1}, τ_{ji2}, τ_{ji3} кожної з підмоделей j в одному такті.

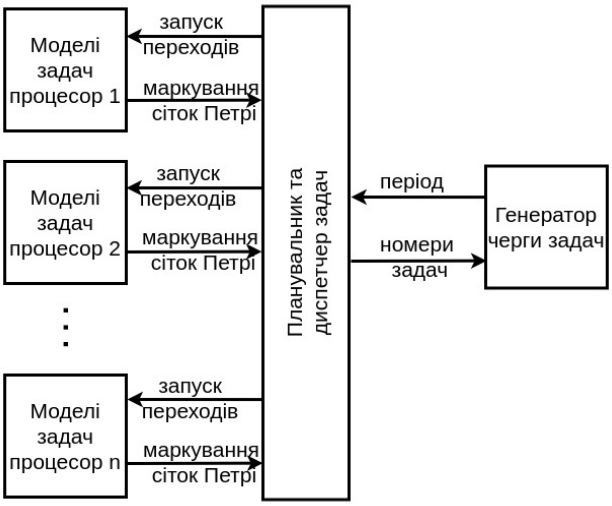

Рисунок 2 –Загальна схема моделювання

Початкові маркування задач, якщо вони обираються на виконання на першому такті, будуть мати такий загальний вигляд:

$$A_i = [N_{j0}\ 0\,0\,1\,0]. \qquad (7)$$

Розглянемо роботу моделі, що складається з трьох задач та двох процесорів, продуктивність яких відрізняється вдвічі. Скористаємось такою формою завдання параметрів задач:

$$A_i : r_i, e_i, d_i. \qquad (8)$$

Задаємо такі параметри: $r_1 = 0$, $r_2 = 1$, $r_3 = 0,\ e_1 = 2$, якщо A_1 виконується на процесорі 1, та $e_1 = 4$, якщо вона виконується на процесорі 2. Відповідно $e_2 = 4,$ або$8, a\ e_3 = 3$ або $6,\ d_1 = 6$, $d_2 = 6$, $d_3 = 6.$

Початкові маркування кожної із задач за умови, що вона запускається першою, а $N_{j0} = 6$ становить:

$$A_i = [6\,0\,0\,1\,0]. \qquad (9)$$

Відповідно до (3) за аналогією з (5) визначимо константи, які додаються до попередніх маркувань сітки за умови спрацювання відповідного переходу:

τ_{ji1}:[001-10];

τ_{ji2}:[11000];

τ_{113} :[0-2-112];

τ_{213} :[0 – 4 -114];

τ_{123}:[0-4-114];

τ_{223}:[0 – 8 -118];

τ_{133} :[0-3-113];

τ_{233}:[0 – 6 -116].

Результати роботи моделі наведено у табл. 4.

Таблиця 4 ***– Виконання сітки Петрі для трьох задач на двох процесорах***

Номер такту	Перехід	Початкове маркування	Результат запуску	Спрацювання переходу	Нове маркування
1	2	3	4	5	6
1	τ_{111} τ_{112} τ_{113}	6 0 0 1 0 6 0 1 0 0 5 1 1 0 0	6 0 1 0 0 5 1 1 0 0 5 -1 0 1 2	+ + -	6 0 1 0 0 5 1 1 0 0 5 1 1 0 0
2	τ_{111} τ_{112} τ_{113}	5 1 1 0 0 5 1 1 0 0 4 2 1 0 0	5 1 2 -1 0 4 2 1 0 0 4 0 0 1 2	- + +	5 1 1 0 0 4 2 1 0 0 4 0 0 1 2
3	τ_{121} τ_{122} τ_{123}	4 0 0 1 0 4 0 1 0 0 3 1 1 0 0	4 0 1 0 0 3 1 1 0 0 3 -3 0 1 4	+ + -	4 0 1 0 0 3 1 1 0 0 3 1 1 0 0
4	τ_{121} τ_{122} τ_{123}	3 1 1 0 0 3 1 1 0 0 2 2 1 0 0	3 1 2 -1 0 2 2 1 0 0 2 -2 0 1 4	- + -	3 1 1 0 0 2 2 1 0 0 2 2 1 0 0
5	τ_{123} τ_{223} τ_{123}	2 2 1 0 0 2 2 1 0 0 1 3 1 0 0	2 2 2 -1 0 1 3 1 0 0 1 -1 0 1 4	- + -	1 2 1 0 0 1 3 1 0 0 1 3 1 0 0

Закінчення табл. 4

1	2	3	4	5	6
6	τ_{121} τ_{122} τ_{123}	1 3 1 0 0 1 3 1 0 0 0 4 1 0 0	1 3 2 -1 0 0 4 1 0 0 0 0 0 1 4	- + +	1 3 1 0 0 0 4 1 0 0 0 0 0 1 4
1	τ_{231} τ_{232} τ_{233}	6 0 0 1 0 6 0 1 0 0 5 1 1 0 0	6 0 1 0 0 5 1 1 0 0 5 -5 0 1 6	+ + -	6 0 1 0 0 5 1 1 0 0 5 1 1 0 0
2	τ_{231} τ_{232} τ_{233}	5 1 1 0 0 5 1 1 0 0 4 2 1 0 0	5 1 2 -1 0 4 2 1 0 0 4 -4 0 1 6	- + -	5 1 1 0 0 4 2 1 0 0 4 2 1 0 0
3	τ_{231} τ_{232} τ_{233}	4 2 1 0 0 4 2 1 0 0 3 3 1 0 0	4 2 2 -1 0 3 3 1 0 0 3 -3 0 1 6	- + -	4 2 1 0 0 3 3 1 0 0 3 3 1 0 0
4	τ_{231} τ_{232} τ_{233}	3 3 1 0 0 3 3 1 0 0 2 4 1 0 0	3 3 2 -1 0 2 4 1 0 0 2 -2 0 1 6	- + -	3 3 1 0 0 2 4 1 0 0 2 4 1 0 0
5	τ_{231} τ_{232} τ_{233}	2 4 1 0 0 2 4 1 0 0 1 5 1 0 0	2 4 2 -1 0 1 5 1 0 0 1 -1 0 1 6	- + -	2 4 1 0 0 1 5 1 0 0 1 5 1 0 0
6	τ_{231} τ_{232} τ_{233}	1 5 1 0 0 1 5 1 0 0 0 6 1 0 0	1 5 2 -1 0 0 6 1 0 0 0 0 0 1 6	- + +	1 5 1 0 0 0 6 1 0 0 0 0 0 1 6

З аналізу таблиці видно, що першою на виконання на процесорі 1 з початку першого такту запускається задача A_1, тобто на нульовій хвилині умовного часу, а на процесорі 2 одночасно (на першому такті) запускається задача A_3. Про це свідчать переходи маркерів з позиції P_{113} у позицію P_{112} та з позиції P_{233} у позицію P_{232}.

Виконання задачі A_i завершується на другому такті, про що свідчить перехід маркера з позиції P_{112} у позицію P_{113}.

Задача A_2 запускається на процесорі 1 на початку третього такту, тобто на другій хвилині умовного часу та завершується на шостому такті, тобто на шостій хвилині.

Задача A_3 завершує своє виконання на процесорі 2 на шостому такті, тобто на шостій хвилині умовного часу.

Отже, усі три задачі успішно виконані, оскільки $T_{1зав} = 2 < 6,\ T_{2зав} = d_2 = 6,\ T_{3зав} = d_3 = 6.$

Легко встановити, наприклад, якщо на першому такті змінити номери задач, що запускаються на процесорах 1 і 2, тобто поміняти місцями запуски задач A_i та A_3, задача A_2 не виконує умову $T_{2зав} = 7 > d_2 = 6.$

Висновки

Запропонована модель є досить ефективним засобом моделювання процесів виконання задач у багатопроцесорних системах реального часу, проста в реалізації, не потребує значних ресурсів машинного часу і гарантує отримання необхідної інформації для вибору оптимального типу планувальника та кількості процесорів необхідної продуктивності.

Список літератури

1 Комп'ютерні системи реального часу: навчальний посібник / Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського". В.Г. Зайцев, Є.І. Цибаєв. – Київ, 2019. Електронний ресурс КПІ ім. Ігоря Сікорського: https://ela.kpi.ua/handle/123456789/29604.

2 Системы реального времени: конспект лекций / Владим. гос. ун-т; сост. А. С. Голубев. – Владимир: Изд-во Владим. гос. ун-та, 2010. – 127 с.

3 Операційні системи: навчальний посібник / Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського": В.Г. Зайцев, І.П. Дробязко. – Київ, 2019. Електронний ресурс КПІ ім. Ігоря Сікорського: https://ela.kpi.ua/handle/123456789/29600.

4 Baker T. Multiprocessors EDF and Deadline Monotonic Shedulability Analysis // Proceeding of 24 IEEE Real – Time Systems Symposium, 2003, p. 120 – 129.

5 Andersen B., Daruah S., Jonson J. Static – Priority Shedulings on Microprocessors // Proccedings of 22 IEEE Real – Time System Symposium. 2001, p. 193 – 202.

6 A.D. Ferrari. Real – Time Scheduling Algorithms // Dr. Dobb's Jornal. 1994, N12, p. 60 – 66.

7 Сети Петри. Электронній ресурс: http://www.hpc-education.ru/files/lectures/2011/ershov/ershov_2011_lectures05.pdf

8 Зайцев В.Г., Цибаєв Є.І. Модель оцінки часових характеристик у комп'ютерних системах реального часу з використанням сіток Петрі // Управління розвитком складних систем, 2019, 40. – С. 76 – 86.

9 Фальк В.Н. Введение в сети Петри и моделирование систем. Учебное пособие. – Москва, 2009. Электронный ресурс: https://studfiles.net/preview/1529418

10 Стеценко І.В. Система імітаційного моделювання засобами сіток Петрі / І.В. Стеценко, О.В. Бойко / Математичні машини і системи. – 2009. – № 1. – С. 117 – 124.

Стаття надійшла до редколегії 03.04.2020

Зайцев Владимир Григорьевич
Доктор технических наук, профессор, профессор кафедры системного программирования и специализированных компьютерных систем, *orcid.org/0000-0001-9548-1959*
Национальный технический университет Украины «Киевский политехнический институт имени Игоря Сикорского», Киев
Цибаев Евгений Игоревич
Кандидат технических наук кафедры системного программирования и специализированных компьютерных систем, *orcid.org/0000-0002-9115-2346*
Национальный технический университет Украины «Киевский политехнический институт имени Игоря Сикорского», Киев

ОЦЕНКА ВРЕМЕННЫХ ХАРАКТЕРИСТИК ЗАДАЧ В МНОГОПРОЦЕССОРНЫХ СИСТЕМАХ РЕАЛЬНОГО ВРЕМЕНИ С ИСПОЛЬЗОВАНИЕМ СЕТЕЙ ПЕТРИ

Аннотация. *Рассмотрена проблема определения временных характеристик задач в системах реального времени, успешность работы которых зависит не только от их логической правильности, но и от промежутка времени, за который они получают результат. Определение таких временных характеристик системы на стадии проектирования является достаточно сложной проблемой. Ее решение в настоящее время основано на двух основных направлениях: теоретических расчетах, связанных с получением так называемых критериев осуществимости, и моделировании работы системы. Среди моделей наиболее распространенными являются статистические модели систем массового обслуживания. Однако, как в первом, так и во втором случаях невозможно получить гарантированный результат, что существенно усложняет процесс проектирования. В последнее время предложено использовать модели, основанные на применении аппарата сетей Петри. Но раньше проведенные исследования касаются только моделирования однопроцессорных систем. Предложен метод оценки временных характеристик задач в системах реального времени путем анализа данных, полученных моделированием распределения процессорного времени между задачами, согласно избранных алгоритмов планировщика, с использованием модели сетей Петри для многопроцессорных систем. Метод гарантирует получение временных характеристик задач при выборе конкретных типов процессоров и планировщика, что необходимо для успешного начала технического проектирования многопроцессорных системы реального времени.*

Ключевые слова: модель; задача реального времени; сетка Петри; многопроцессорная система

Zaitsev Vladimir
DSc (Eng.), Professor. Professor of Department of System Programming and Specialized Computer Systems, Kyiv Polytechnic Institute, *orcid.org/0000-0001-9548-1959*
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv
Tsybaev Evgeniy
PhD (Eng.), Department of System Programming and Specialized Computer Systems, Kyiv Polytechnic Institute, *orcid.org/0000-0002-9115-2346*
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv

EVALUATION OF TIME CHARACTERISTICS OF PROBLEMS IN MULTIPROCESSOR REAL-TIME SYSTEMS USING PETRI NETWORKS

Abstract. *The work is devoted to the problem of determining the temporal characteristics of tasks in real-time systems, the success of which depends not only on their logical correctness but also on the time for which they receive the result. Determining such temporal characteristics of the system at the design stage is a rather difficult problem. Its solution is currently based on two main areas: theoretical calculations related to obtaining the so-called feasibility criteria and modeling the operation of the system on statistical models of queues. It is hard to get a guaranteed result, which significantly complicates the design process. Statistical models guarantee the production of predicted average statistical values of parameters, which may be sufficient for soft real-time systems, but this is not enough when designing hard real-time systems. Within the framework of the created model, it is possible to allow extrusion another task. After a while, the interrupted task can be resumed. Thus there is interference – the mutual influence of active tasks on the time of their execution, which must be taken into account recently. However, the studies carried out concern only the simulation of single-processor systems. .The method of estimation of time characteristics of tasks in real time systems by data analysis is proposed, obtained by simulating the allocation of processor time between tasks according to the selected scheduler algorithms using a Petri net model for multiprocessor systems. The method guarantees the timing of tasks when selecting specific types of processors and floaters, what you need to get started designing real-time multiprocessor systems.*

Keywords**: **model; real-time task; Petri net; multiprocessor system

References

1. Zaitsev, V.G., & Tsybaev, E.I., (2019). Real-time computer systems: a textbook. National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute": Kyiv. Igor Sikorsky KPI Electronic Resource: https://ela.kpi.ua/handle/123456789/29604.

2. Golubev, A.S., (2010). Real-time systems: lecture notes. Vladimir: Publishing house Vladim. State University, 127.

3. Zaitsev, V.G., Drobyazko, I.P. & Tsybaev, E.I., (2019). Operating Systems: Tutorial, 1. Real-time computer systems: a textbook. Kyiv: Igor Sikorsky KPI Electronic Resource: https://ela.kpi.ua/handle/123456789/29600.

4. Baker, T., (2003). Multiprocessors EDF and Deadline Monotonic Shedulability Analysis. Proceeding of 24 IEEE Real – Time Systems Symposium, Pp. 120 – 129.

5. Andersen, B., Daruah, S. & Jonson, J., (2003). Static – Priority Shedulings on Microprocessors. Proccedings of 22 IEEE Real – Time System Symposium, Pp. 193 – 202.

6. Ferrari, A.D., (1994). Real – Time Scheduling Algorithms. Dr. Dobb's Jornal, 12, 60 – 66.

7. Petri nets, (2011). [Electronic resource]: Access mode : http://www.hpc-education.ru/files/lectures/2011/ershov/ershov_2011_lectures05.pdf

8. Zaitsev, V.G. & Tsybaev, E.I., (2019). A model for estimating time characteristics in real-time computer systems using Petri nets. Management of Complex Systems Development, 40, 76 – 86.

9. Falk, V.N., (2009). Introduction to Petri Nets and System Modeling. Tutorial. Moscow. [Electronic resource]: https://studfiles.net/preview/1529418.

10. Stetsenko, I.V., & Boyko, O.V., (2009). System of simulation modeling by means of Petri nets. Mathematical Machines and Systems, 1, 117 – 124.

Посилання на публікацію

APA *Zaitsev, Vladimir, & Tsybaev, Evgeniy. (2020). Evaluation of time characteristics of problems in multiprocessor real-time systems using Petri networks. Management of Development of Complex Systems, 42, 43 – 50, [in Ukrainian]; dx.doi.org\10.32347/2412-9933.2020.42.43-50.*

ДСТУ *Зайцев В.Г. Оцінка часових характеристик задач в багатопроцесорних системах реального часу з використанням сіток Петрі [Текст] / В.Г. Зайцев, Є.І. Цибаєв // Управління розвитком складних систем. – 2020. – № 42. – С. 43 – 50; dx.doi.org\10.32347/2412-9933.2020.42.43-50.*