

УДК 004.451

О. С. Савенко, к. т. н., доц.;

С. В. Мостовий, асп.

МЕТОД ПРОГНОЗУВАННЯ СТАНУ ПРОЦЕСІВ В ПЕРСОНАЛЬНИХ КОМП'ЮТЕРАХ

Розроблено метод прогнозування стану процесів в персональних комп'ютерах, який дає можливість передбачати настання ситуації взаємоблокування процесів на основі аналізу взаємодії сигнатур процесів.

Вступ

У наш час все більшої уваги приділяється паралельним обчисленням та паралельним системам, що в свою чергу спонукає до ефективнішого вирішення проблем пов'язаних з ними. При паралельному виконанні можуть виникати такі ситуації, при яких два і більше процеси весь час знаходяться в стані блокування. Про такі процеси говорять, що вони знаходяться в стані взаємного блокування, безвиході, дедлока (deadlock) або клінчу (clinch) (смертельних обіймів — deadly embrace). В даний час розглядаються можливі компромісні рішення з погляду накладних витрат на включення засобів боротьби з взаємоблокуваннями і очікуваних від цього вигод. В деяких випадках ціна, яку доводиться платити за те, щоб зробити систему вільною від взаємоблокувань, дуже висока. В системах реального часу виникнення ситуації взаємоблокування може призвести до катастрофічних наслідків.

Постановка задачі

Для вирішення проблеми взаємоблокування відомо багато методів і алгоритмів [1—4]. Вони поділяються на наступні групи:

— Запобігання взаємоблокуванню. Полягають у забезпеченні невиконання однієї з чотирьох умов для взаємоблокування.

— Уникнення взаємоблокування. Полягають у тому, щоб не задовільняти запит на ресурс, якщо його виділення може потенційно спричинити взаємоблокування.

— Виявлення взаємоблокувань. Полягає у тому, щоб завжди задовільняти запити на ресурси, коли це можливо, і періодично перевіряти систему на наявність взаємоблокувань. Якщо взаємоблокування має місце, то вирішити проблему.

— Ігнорування взаємоблокувань. Має сенс коли ймовірність виникнення взаємоблокування дуже низька.

Вагома частка взаємоблокувань припадає на взаємоблокування процесів, що виконуються в операційних системах (ОС) для персональних комп'ютерів (ПК). Проте, методи вирішення задачі взаємоблокування процесів в ОС ПК мають ряд недоліків [4] (блокування роботи ОС, наявність циклів активного очікування, складність програмної реалізації для багатьох процесів, необхідність використання спеціалізованої команди процесора) і є складними для їх реалізації. Тому більшість сучасних ОС не містять засобів для вирішення проблеми взаємоблокування процесів.

Для усунення виявлених недоліків та вирішення задачі взаємоблокування процесів в [5] запропонована модель прогнозування стану процесів в персональних комп'ютерах, яка базується на використанні сигнатур процесів.

До складу моделі входять такі структурні частини (рис. 1):

— підсистема виявлення зміни стану процесу — призначена для контролю за поточними параметрами процесів, що вже присутні на ПК в певній ОС, та запуском нових процесів;

— підсистема визначення характеристик ОС даного ПК — призначена для контролю за зміною основних характеристик, що суттєво впливають на наближення процесів до стану взаємоблокування;

— підсистема побудови сигнатури процесу — призначена для побудови сигнатури процесу, що виконує перехід у стан готовності і ще не має сигнатури, та модифікації сигнатури процесів, що змінили значення своїх параметрів та мають відповідні їм сигнатури;

— множина сигнатур процесів, що присутні на ПК в певній ОС — призначена для зберігання сигнатур процесів, які в даний момент виконуються на ПК;

— підсистема аналізу та логічного висновку — за допомогою множини правил взаємодії сигнатур процесів виділяє із множини сигнатур всіх процесів підмножину сигнатур процесів, що наближаються до стану взаємоблокування, і робить висновок про можливість продовження роботи цих процесів.

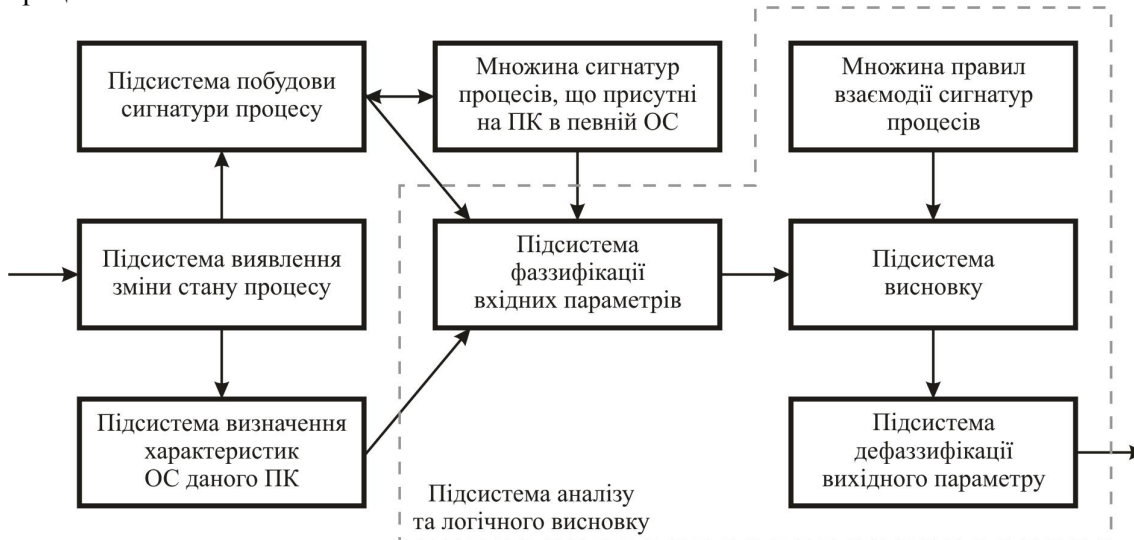


Рис. 1. Модель процесу прогнозування стану процесів в персональних комп'ютерах

Для здійснення аналізу та логічного висновку використаємо нечітку експертну систему. Згідно з [5,6] включимо до складу підсистеми аналізу та логічного висновку такі компоненти:

— підсистема фазифікації вхідних параметрів — визначає ступені впевненості в тому, що вихідні лінгвістичні змінні отримують конкретні значення;

— підсистема висновку — на основі множини правил взаємодії сигнатур процесів та набору вхідних лінгвістичних змінних проводиться оцінка істинності для кожного правила та формується єдина нечітка множина;

— підсистема дефазифікації вихідного параметру — перетворює нечіткий набір значень вихідної лінгвістичної змінної до точного значення.

Прогнозування стану процесів в ПК

Розроблений метод прогнозування стану процесів в ПК включає такі етапи:

- 1) виявлення змін в системі;
- 2) побудова сигнатури процесів;
- 3) визначення основних характеристик ПК та ОС;
- 4) зведення до нормованого вигляду вхідних параметрів та їх передача на підсистему аналізу та логічного висновку;
- 5) здійснення висновку про настання взаємоблокування процесів

На першому етапі виявляємо зміни, що відбуваються в системі, а саме виявляємо запити на створення нового процесу чи зміну параметрів вже існуючого процесу. Ця робота покладена на підсистему виявлення зміни стану процесу.

На другому етапі здійснюємо побудову (перебудова для вже існуючих процесів) сигнатури для виявленого процесу. Побудову сигнатури здійснимо таким чином:

1. Перевіряємо, чи має процес раніше сформовану сигнатуру. Якщо так, то переходимо до дії 5.
2. Отримуємо інформацію про процес, яка необхідна для формування його сигнатури.
3. Формуємо сигнатуру процесу. На цьому кроці відбувається перетворення значень необхідних параметрів процесу до заданого вигляду і утворення його сигнатури (рис. 2).

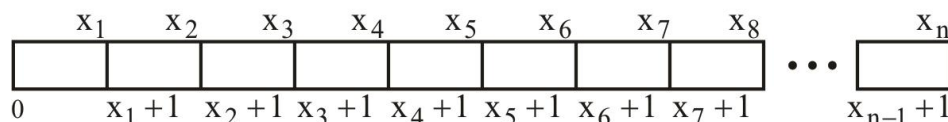


Рис. 2. Зображення сигнатури у машинному форматі

До необхідних параметрів входять наступні: ідентифікатор процесу (x_1 біт); ідентифікатор батьківського процесу ($x_2 - x_1$ біт); ідентифікатор користувача ($x_3 - x_2$ біт); пріоритет процесу ($x_4 - x_3$ біт); кількість дескрипторів файлів, що використовуються процесом ($x_5 - x_4$ біт); обсяг віртуальної пам'яті, що використовує процес ($x_6 - x_5$ біт); час виконання процесу ($x_7 - x_6$ біт); додаткові параметри процесу ($x_n - x_7$ біт).

4. Створену сигнатуру додаємо до множини сигнатур процесів.

5. Перевіряємо, чи змінились значення складових сигнатури процесу. Якщо так, то переходимо до дії 6, інакше переходимо до дії 7.

6. Модифікуємо сигнатуру процесу згідно поточних значень складових.

7. Зберігаємо поточну множини сигнатур процесів.

Ці дії покладені на підсистему побудови сигнатури.

На третьому етапі визначаємо поточні характеристики ПК та ОС. Ці дії проводяться підсистемою визначення характеристик ПК та ОС.

Оскільки взаємоблокування настає по причині конкуренції процесів за системні ресурси, то включаємо їх кількісні характеристики до бази знань експертної системи:

- кількість процесів користувача, присутніх на ПК в певній ОС;
- кількість процесів ядра системи, присутніх на ПК в певній ОС;
- обсяг ОП на ПК в певній ОС (загальний та вільної в даний момент);
- обсяг ЗП на ПК в певній ОС (загальний та вільної в даний момент);
- кількість пристроїв вводу-виводу інформації (загальна та вільних у даний момент);
- загальну кількість файлів на ПК в певній ОС.

На четвертому етапі передаємо множину сигнатур процесів та множину характеристик ПК та ОС для проведення аналізу на предмет настання ситуації взаємоблокування в системі. Ці дані подаються на підсистему аналізу та логічного висновку.

Оскільки всі вхідні характеристики знаходяться у різних числових межах, що є незручним для опрацювання результатів, то на даному етапі проводиться нормування даних величин. Зведемо усі показники до меж $[0; 1]$. Для цього необхідно виконати ділення показника на максимально допустиме значення цього показника в конкретній комп'ютерній системі (КС). Наприклад, загальний обсяг ОП складає 256 Мб, а вільної в даний момент пам'яті — 60 Мб. Тоді показник вільної в даний момент пам'яті складе $60/256 = 0,234$.

Проте такий підхід до нормування показників не буде враховувати ступінь впливу даного показника на результат системи прогнозування стану процесів, оскільки для показників із однаковим нормованим значенням, але з різними поточними та максимально допустимими значеннями, буде різний вплив на КС. Наприклад, при нормованому значенні показника вільної пам'яті 0,1 для КС із загальним обсягом ОП 2Гб та КС із загальним обсягом ОП 128 Мб кількісні показники значно відрізняються і є більш критичними для КС із меншим обсягом ОП. Тому необхідно при нормуванні враховувати максимально допустиме значення показника. Для цього нормування показників проведемо за такою формулою:

$$P_n = 1 - \frac{P_d + P_m}{P_d \cdot P_m}; \quad P_d \neq 0, \quad (1)$$

де P_n — черговий нормований показник; P_d — поточне значення показника в конкретній КС; P_m — максимальне значення показника в конкретній КС.

Такий підхід до нормування показників дозволяє врахувати ступінь впливу даного показника на результат системи прогнозування стану процесів.

На п'ятому етапі відбувається аналіз взаємодії процесів на основі аналізу їхніх сигнатур та характеристик ПК та ОС і здійснюється висновок про настання ситуації взаємоблокування процесів. Для

здійснення даних дій використовується нечітка експертна система [6].

Позначимо через U множину всіх можливих сигнатур процесів (універсум). Виділимо множину сигнатур процесів, що є активними в даний момент в КС і позначимо її через A ($A \subseteq MU$). Процеси, що є активними в даний момент, можна розділити на 2 підмножини: процеси, запущені користувачем, та процеси, запущені ядром системи. Отже, множину A можна розбити на дві підмножини: K та S ($K \subseteq MU, S \subseteq MU, K \cup S = A$). Всі перераховані множини належать до чітких. Їх характеристична функція приймає 1, якщо елемент належить множині, і 0 в протилежному випадку.

При наближенні стану блокування процесів в КС із множини A можна виділити підмножину A^* сигнатур процесів, які найімовірніше приведуть до цього стану. Відповідно множину A^* також можна розділити на 2 підмножини: K^* — множина процесів, запущених користувачем, які ймовірно призведуть до стану блокування, та S^* — множина процесів, запущених ядром системи, які ймовірно призведуть до блокування ($A^* \subseteq MU, K^* \subseteq MU, S^* \subseteq MU, K^* \cup S^* = A^*$). Множини A^*, K^*, S^* є нечіткими, оскільки сигнатури процесів входять у них з певними ймовірностями.

Якщо новий процес за висновком системи не призводить до взаємоблокування, то йому дозволяється виконання на ПК і його сигнатура додається до бази сигнатур працюючих процесів. В протилежному випадку система виявляє множину процесів, які наближаються до стану взаємоблокування, та приймає рішення, який із процесів зняти із виконання.

Послідовність наведених вище дій можна подати алгоритмом (рис. 3):

Для оцінки ефективності розробленого методу проведено його порівняння з відомими методами уникнення ситуації взаємоблокування процесів. При порівнянні до уваги були взяті основні недоліки існуючих методів. Проведене дослідження показало, що у розробленому методі вирішується частина суттєвих недоліків відомих методів уникнення взаємоблокування. Результати дослідження подані в табл. 1. Для оцінки ефективності розробленого методу було реалізовано програмне забезпечення під ОС сімейства Linux за даним методом і за алгоритмом банкіра [7]. Результати дослідження подані в табл. 2

Таблиця 1

Результати порівняння розробленого методу та відомих методів

№ п/п	Параметр для порівняння	Назва методу								
		Заборона на переривання	Строге чередування	Алгоритм Деккера	Алгоритм Петерсона	Алгоритм банкіра	Семафори	Монітори	М'ютекси	Прогнозування стану процесів
1.	Складність реалізації для 2 і більше процесів	ні	ні	так	так	так	ні	ні	ні	ні
2.	Прив'язка до кількості та видів ресурсів ПК	ні	ні	ні	ні	так	ні	ні	ні	ні
3.	Активне очікування	так	так	так	так	так	так	ні	ні	так
4.	Необхідність підтримки на рівні компіляторів	ні	ні	ні	ні	ні	ні	ні	так	ні
5.	Наявність спеціалізованої команди процесора	ні	ні	ні	ні	ні	ні	так	ні	ні
6.	Зниження продуктивності роботи обчислювальної системи	ні	так	ні	ні	ні	ні	ні	ні	ні
7.	Блокування роботи операційної системи	так	ні	ні	ні	ні	так	ні	ні	ні
8.	Спосіб реалізації методу	апаратний	програмний	програмний	програмний	програмний	програмний	програмний	програмно-апаратний	програмний

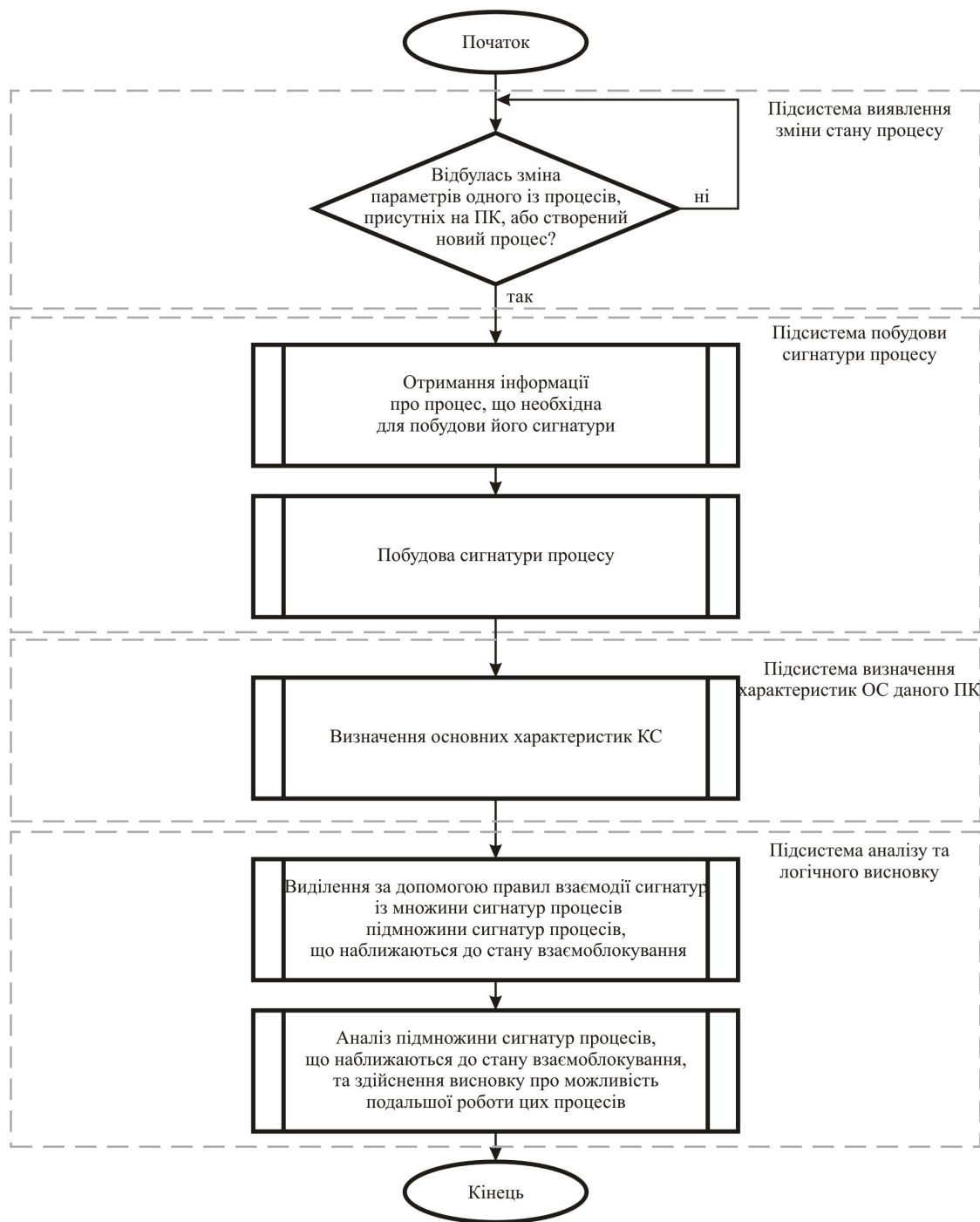


Рис. 3. Алгоритм визначення стану процесу

Таблиця 2

Результати порівняння програмного забезпечення за розробленим методом та алгоритмом «банкіра»

№ п/п	Параметр для порівняння	Алгоритм «банкіра»	Розроблені алгоритми
1	Складність реалізації для 2 і більше процесів	так	ні
2	Прив'язка до кількості та видів ресурсів ПК	так	ні
3	Завантаження центрального процесора	12...14 %	15...18 %
4	Кількість виявлених ситуацій взаємоблокування	5 з 10	9 з 10

Висновок

Розроблений метод прогнозування стану процесів в персональних комп'ютерах дає можливість передбачати взаємоблокування процесів на основі аналізу взаємодії сигнатур процесів та усуває основні недоліки відомих методів, зокрема не потребує спеціалізованої команди процесора, не допускає блокування роботи ОС, є простим у реалізації для багатьох процесів під різні типи операційних систем.

СПИСОК ЛІТЕРАТУРИ

1. Coffman E. G. System deadlocks / E. G. Coffman, M. J. Elphick, A. Shoshani. // Computing Surveys. — June 1971. — Vol.3, No.2. — Pages 67—78.
2. Kaveh N. Deadlock detection in distribution object systems / Nima Kaveh, Wolfgang Emmerich // Software Engineering Notes. — September 2001. — Vol. 26, No. 5. — Pages 44—51.
3. Confirmation of deadlock potentials detected by runtime analysis / Saddek Bensalem, Jean-Claude Fernandez, Klaus Havelund, Laurent Mounier // International Symposium on Software Testing and Analysis — 2006. — Pages 41—50.
4. Савенко О. С., Кльоц Ю. П., Мостовий С. В. Дослідження та аналіз блокування процесів в комп'ютерній системі / О. С. Савенко, Ю. П. Кльоц, С. В. Мостовий // Вісник ХНУ. — 2007. — № 3, т. 1. — С. 248—251
5. Савенко О. С. Модель прогнозування стану процесів в комп'ютерній системі / О. С. Савенко, С. В. Мостовий // Радіоелектронні і комп'ютерні системи — 2008. — № 5 (32). — С. 109—115
6. Система прогнозування стану процесів в персональному комп'ютері: збірник праць VIII міжнародної конференції, 14—17 травня 2008р., Київ / відп. ред. С. В. Сирота. — К.: Просвіта, 2008. — С. 308—314
7. Мостовий С. В. Алгоритми і програмні засоби прогнозування стану процесів в персональному комп'ютері / С. В. Мостовий // Вісник ХНУ. — 2008. — № 5. — С. 124—130.

Рекомендована кафедрою автоматики та інформаційно-вимірювальної техніки

Надійшла до редакції 21.10.08
Рекомендована до друку 20.11.08

Савенко Олег Станіславович — декан факультету комп'ютерних систем та програмування; **Мостовий Сергій Володимирович** — асистент кафедри системного програмування.

Хмельницький національний університет