УДК 621.3

С.В. Осієвський[1], І.Є. Кужель[1], А.П. Литвин[2]

[1]*Харківський університет Повітряних Сил ім. І. Кожедуба, Харків*
[2]*В/ч А-4031, Умань*

ЗАСТОСУВАННЯ ЗАСОБІВ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОЕКТУВАННЯ ДЛЯ РІШЕННЯ ЗАДАЧІ УСУНЕННЯ ТУПИКОВИХ СИТУАЦІЙ В БАГАТОПРОЦЕСОРНИХ ОБЧИСЛЮВАЛЬНИХ КОМПЛЕКСАХ

В статті розглядається можливість застосування засобів об'єктно-орієнтованого проектування для рішення одного класу задач в багатопроцесорних обчислювальних комплексах. Показана актуальність застосування засобів об'єктно-орієнтованого проектування при розробці програмних додатків для багатопроцесорних обчислювальних комплексів. Перевагою такого підходу являється можливість проведення завчасної оцінки несуперечності функцій та процедур, що передбачаються для реалізації в таких комплексах.

Ключові слова: *тупик, варіант використання, діаграма послідовності.*

Вступ

Постановка проблеми. Активне впровадження в експлуатацію багатопроцесорних (БПОК) і багатомашинних обчислювальних комплексів (БМОК), як правило, має на меті підвищення або рівня надійності, або рівня продуктивності комплексів до значень недоступних або важко реалізованих в традиційних ЕОМ.

На більшості класів вирішуваних задач для досягнення високої продуктивності найбільш ефективними виявляються БПОК. Це пов'язано з великою інтенсивністю інформаційних обмінів між підзадачами, яка приводить до дуже високих накладних витрат в БМОК. Хоча слід зазначити, що БМОК, дозволяють досягти досить високої продуктивності завдяки кращій масштабованості, проте ця перевага виявляється тільки при відповідності вирішуваних задач умові максимального подовження незалежних гілок програми, що не завжди можливо [1, 2].

В той же час однією із найбільш критичних проблем, що має місце в БПОК являється проблема наявності тупикових ситуацій між процесами, що вимагають однойменних ресурсів [1, 3].

В даній статті розглянуті можливі шляхи уникнення тупикових ситуацій виходячи з можливостей застосування засобів об'єктно-орієнтованого проектування та програмування.

Основна частина

Розглянемо приклад тупикової ситуації, виходячи з умови одночасного захоплення ресурсів БПОК декількома процесами. Нехай двом процесам, що виконуються в режимі мультипрограмування, для виконання їх роботи потрібно два ресурси, наприклад, сервер друку та файловий сервер. На рис. 1 показані фрагменти реалізації відповідних дій процесів.

Нехай після того, як процес А захватив сервер друку, він був перерваний. Управління отримав процес В, який спочатку захватив файловий сервер, але при виконанні наступної команди був заблокований, оскільки сервер друку виявився вже зайнятим процесом А. Управління знову отримав процес А, який відповідно до своєї програми зробив спробу захватити файловий сервер і був заблокований: файловий сервер вже розподілений процесу В. В такому положенні процеси А і В можуть перебувати скільки завгодно довго.

Залежно від співвідношення швидкостей процесів, вони можуть або абсолютно незалежно використовувати розподілені ресурси (г) або утворювати черги до розподілених ресурсів (в), або взаємно блокувати один одного (б). Тупикові ситуації слід відрізняти від простих черг, хоча і ті і інші виникають при сумісному використанні ресурсів і зовні виглядають схоже: процес припиняється і чекає звільнення ресурсу. Проте черга - це нормальне явище, невід'ємна ознака високого коефіцієнта використання ресурсів при випадковому надходженні запитів. Вона виникає тоді, коли ресурс недоступний в даний момент, але через деякий час він звільняється, і процес продовжує своє виконання. Тупик же є в деякому розумінні нерозв'язною ситуацією.

У розглянутих прикладах тупик був утворена двома процесами, але взаємно блокувати один одного може і значно більше число процесів.

Отже, проблема тупика вимагає вирішення включає наступних задач:

– запобігання виникненню тупикових ситуацій;

– розпізнавання тупикової ситуації;

– відновлення системи після виникнення тупикової ситуації.

Стосовно вирішення першої задачі слід зазначити, що вона може бути вирішена ще на стадії написання програмних додатків реалізуємих в БПОК, тобто програми мають бути написані таким чином, щоб тупикова ситуація була виключена при будь якому співвідношенні співвідношенні взаємних

швидкостей процесів. Так, для наведеної ситуації на рис. 1, якби процес А і процес В запрошували ресурси в однаковій послідовності, то тупик був би неможливим. Другий підхід щодо запобігання тупику являється і полягає у використанні певних правил при призначенні ресурсів процесам, наприклад, ресурси можуть виділятися в певній послідовності, що є загальноприйнятою для всіх процесів.

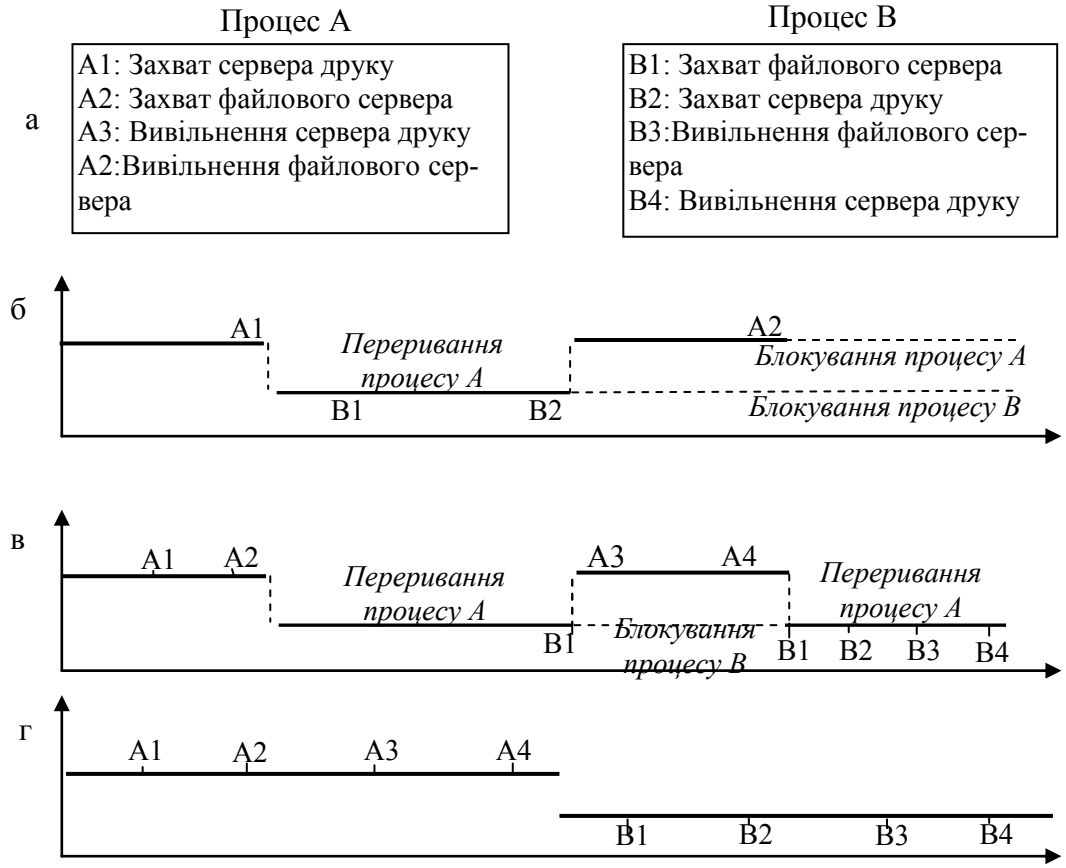

Рис. 1. Реалізація процесів в БПОК при одночасному захопленні ресурсів

В деяких випадках, коли тупикова ситуація утворена багатьма процесами, що використовують багато ресурсів, розпізнавання тупику є складною задачею. Існують формальні, програмно - реалізовані методи розпізнавання тупику, засновані на веденні таблиць розподілу ресурсів і таблиць запитів до зайнятих ресурсів. Аналіз цих таблиць дозволяє виявити взаємні блокування.

Якщо ж тупикова ситуація виникла, то не обов'язково знімати з виконання всі заблоковані процеси. Можна зняти тільки частину з них, при цьому звільняються ресурси, очікувані рештою процесів. Крім цього як окрема можливість усунення тупикової ситуації являється повернення деяких процесів в область свопінгу.

Зі всього вищесказаного зрозуміло, що використання структурного підходу при написанні програмних додатків реалізуємих на БПОК, являється досить складною задачею, що не може бути відслідкованою в процесі розробки і, як наслідок, одна незначна помилка може привести до остановки системи. Для того, щоб полегшити написання коректних програм на основі структурного підходу було запропоновано використання таких засобів управління процесами як семафори та монітори програмних додатків.

Як показав аналіз існуючих задач та методів їх реалізації в БПОК, що складаються з декількох процесорів, кожен з яких має власну оперативну пам'ять, семафори і монітори виявляються непридатними. У таких системах синхронізація може бути реалізована тільки за допомогою обміну повідомленнями.

З погляду на можливості сучасних засобів розробки програмних додатків та інформаційних систем пропонується рішення даної задачі можливостями програмного середовища Rational Rose. Можливості даного середовища розглянуто в [4].

Нехай всі ресурси системи повністю впорядковані від 1 до r. Накладемо наступне обмеження: процес не може захватити ресурс R_k, якщо він утримує ресурс R_h і при цьому $k< h$.

Зрозуміло, що, використовуючи це правило, неможливо потрапити в тупик. Розглянемо застосування даного правила до практичної реалізації задачі. Нехай в системі існує процес, який використовує, впорядковані ресурси: A, B, C, D, E, тоді діаграма послідовності стратегій може бути представлена таким чином, як на рис. 2.

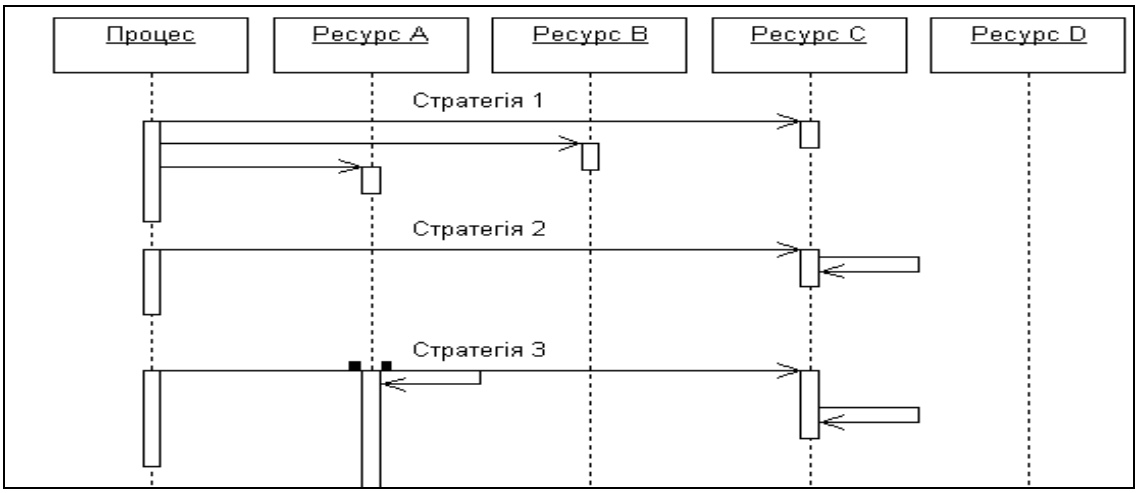

Рис. 2. Фрагмент діаграми послідовності стратегій

Виходячи з наведеної діаграми можна зазначити наступні стратегії реалізації процесу в системі:

Стратегія 1: захоплення (A); захоплення (B); захоплення (C);.

Стратегія 2: використання C;

Стратегія 3: використання A, C;

Стратегія 4: використання A, B, C;

Стратегія 5: звільнення (A); звільнення (B); захоплення (E);

Стратегія 6: використання C і E;

Стратегія 7: звільнення (C); звільнення (E); захоплення (D);

Стратегія 8: використання D;

Стратегія 9: звільнення (D).

Наведений перелік стратегій являється несуперечним, що підтверджує перевірка коректності діаграми послідовності (характеристична оцінка 3,19).

Сукупність стратегій наведеної послідовності може бути використана коли в системі є в наявності декілька ресурсів. Слід зазначити, що в даному випадку ступінь паралелізму зменшується не дуже сильно.

Інший підхід до рішення наведеної задачі базується на дослідженні ієрархічних структур. Тобто у випадку, якщо ресурси ієрархічно структуровані і блокування повинне застосовуватися тільки в ієрархічному порядку. Нехай утримування ресурсів представлене діаграмою взаємодії (рис. 3).

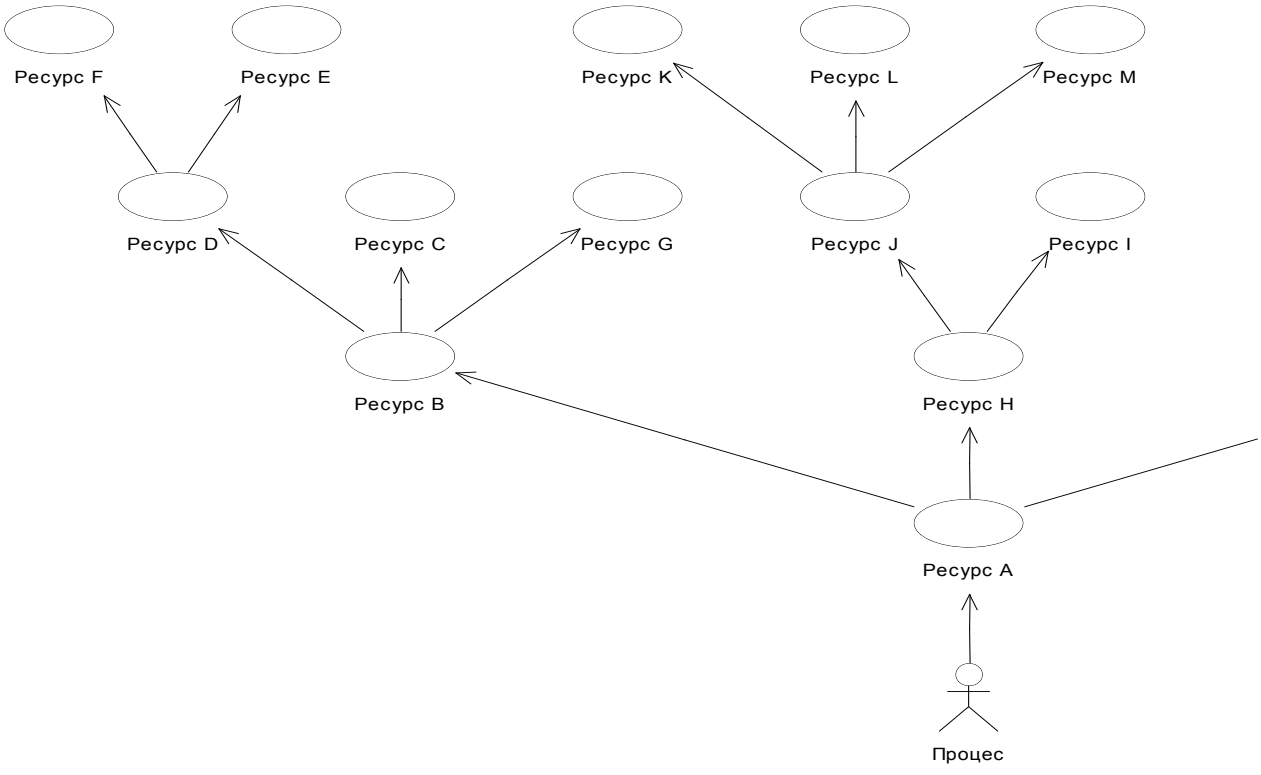

Рис. 3. Діаграма варіантів взаємодії між ресурсами та процесом системи

Користувач може блокувати будь-який варіант використання або групу варіантів використання, що представлені на діаграмі. Тоді наступне правило гарантує запобігання тупикам:

> варі*анти використання, що блоковані в поточний момент часу певним процесом, повинні розташовуватися на всіх шляхах до бажаних ресурсів.*

Приклад використання цього правила, з блокуванням одиночного ресурсу одночасно наведено на рис. 3.

Якщо процесу необхідно використати ресурси e, f, i, k він повинен виконати наступну послідовність операцій:

– блокування (a),
– блокування (b),
– блокування (h),
– звільнення (a),
– блокування (d),
– звільнення (b),
– блокування (i),
– блокування (j),
– звільнення (h),
– блокування (k),
– звільнення (j),
– блокування (e),
– блокування (f),
– звільнення (d).

Дана послідовність операцій досліджена при визначенні оцінки несуперечності діаграми варіантів використання (оціночна характеристика 2,34).

Висновок

Таким чином наведені рішення свідчать про актуальність застосування засобів об'єктно-орієнтованого проектування при розробці програмних додатків для багатопроцесорних обчислювальних комплексів.

Перевагою такого підходу являється можливість проведення завчасної оцінки несуперечності функцій та процедур, що передбачаються для реалізації в багатопроцесорних обчислювальних комплексах.

Список літератури

1. Microsoft Corporation. Распределенные системы. Книга 1. Ресурсы Microsoft Windows 2000 Server: пер. с англ. – М.: Издательско-торговый дом «Русская Редакция», 2001. – 864 с.: ил.

2. Microsoft Corporation Управление сетевой средой Microsoft Windows 2000. Учебный курс MCSA/MCSE: пер. с англ. – М.: Издательско-торговый дом «Русская редакция». 2003. – 896 стр.

3. Таненбаум Э. Современные операционные системы 2-е изд. – СПб.: Питер, 2002.– 1040с

4. Тимочко О.І., Осієвський С.В., Гурін О.С. Методика кількісної оцінки діаграм UML // Системи обробки інформації. – Х: ХУПС, 2005. – Вип. 8(48). – С. 146-150.

Надійшла до редколегії 12.12.2009

Рецензент: д-р фіз..-мат. наук, проф. С.В. Смеляков, Харківський університет Повітряних Сил ім. І. Кожедуба, Харків.

ПРИМЕНЕНИЕ СРЕДСТВ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОЕКТИРОВАНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧИ УСТРАНЕНИЯ ТУПИКОВЫХ СИТУАЦИЙ В МНОГОПРОЦЕССОРНЫХ ВЫЧИСЛИТЕЛЬНЫХ КОМПЛЕКСАХ

С.В. Осиевский, И.Е. Кужель, А.П. Литвин

В статье рассматривается возможность применения средств объектно-ориентированного проектирования для решения одного класса задач в многопроцессорных вычислительных комплексах. Показана актуальность применения средств объектно-ориентированного проектирования при разработке программных дополнений для многопроцессорных вычислительных комплексов. Преимуществом такого подхода является возможность проведения заблаговременной оценки несуперечності функций и процедур, которые предусматриваются для реализации в таких комплексах.

Ключевые слова: *тупик, вариант использования, диаграмма последовательности.*

APPLICATION OF OBJECT-ORIENTED PLANNING FACILITIES FOR DECISION OF REMOVAL TASK OF DEADLY EMBRACES IN MULTIPROCESSORS CALCULABLE COMPLEXES

C.B. Osievskiy, I.E. Kuzhel', A.P. Litvin

In the article possibility of application of facilities of the object-oriented planning is examined for a decision one class of tasks in multiprocessors calculable complexes. Actuality of application of facilities of the object-oriented planning is rotined at development of programmatic additions for multiprocessor calculable complexes. Advantage of such approach is possibility of leadthrough of the done early estimation of несуперечності functions and procedures which are foreseen for realization in such complexes.

Keywords: *deadlock, variant of the use, diagram of sequence.*