УДК 004.032.26, 004.8

В.В. Струшинська
В.О. Денисюк

МОДЕЛЮВАННЯ ТА АНАЛІЗ МЕХАНІЗМУ БАГАТОЗНАЧНИХ СЕМАФОРІВ У ПАРАЛЕЛЬНИХ ОБЧИСЛЕННЯХ

Вінницький національний технічний університет

Анотація

Розглянуто розробку програмної моделі механізму багатозначних семафорів для синхронізації процесів у багатопотокових системах. Проаналізовано принципи роботи семафорів, обґрунтовано вибір мови програмування C++ та використаних примітивів синхронізації. У роботі створено програмну реалізацію алгоритму та проведено тестування продуктивності при змінному навантаженні на спільний ресурс. Використання результатів дозволить візуалізувати процеси очікування та підвищити надійність управління спільними ресурсами.

Ключові слова: *багатозначний семафор, паралельні обчислення, синхронізація, м'ютекс, продуктивність..*

Abstract

The development of a software model of the multi-valued semaphore mechanism for process synchronization in multithreaded systems is considered. The principles of semaphores are analyzed, the choice of C++ programming language and synchronization primitives used is justified. In the work, a software implementation of the algorithm is created and performance testing is carried out under variable load on a shared resource. Using the results will allow visualizing waiting processes and improving the reliability of shared resource management.

Keywords: *multi-valued semaphore, parallel computing, synchronization, mutex, performance.*

Вступ

Актуальність реалізації та дослідження механізму багатозначних семафорів полягає у необхідності ефективної організації паралельних процесів в сучасних операційних системах. Зростання потреби у багатопотокових системах вимагає надійних методів керування доступом до спільних ресурсів для уникнення конфліктів та стану гонки даних [1, 2].

Багатозначні семафори залишаються одним із найгнучкіших засобів, які дозволяють керувати обмеженням кількості процесів, що мають одночасний доступ до ресурсу.

Метою роботи є розробка моделі та програмна реалізація алгоритму багатозначного семафора для аналізу ефективності синхронізації та оцінки часових характеристик виконання потоків.

Постановка задачі дослідження

Задачі дослідження полягають у вирішенні наступних питань:

- аналіз архітектури механізму семафора та методів синхронізації;
- розробка алгоритму та UML-діаграм діяльності для моделювання роботи семафора;
- програмна реалізація моделі з використанням мови C++ та стандартних бібліотек потоків;
- тестування програми та аналіз залежності часу очікування від кількості потоків та дозволів.

Виклад основного матеріалу

У теорії паралельних обчислень семафор визначається як змінна спеціального типу, над якою допустимі дві атомарні операції: P (Wait/Acquire) та V (Signal/Release). Назви цих операцій походять від нідерландських слів *Proben* (перевірити) та *Verhogen* (підвищити). Механізм семафорів є

універсальним засобом синхронізації, що дозволяє вирішувати як задачі взаємного виключення, так і загальні проблеми координації процесів.

Розрізняють два основні типи семафорів: двійкові та багатозначні. Двійковий семафор може приймати лише значення 0 або 1, працюючи подібно до м'ютекса, де 0 означає «зайнято», а 1 — «вільно». Натомість багатозначний (лічильний) семафор може приймати цілі значення від 0 до заданого числа N. Це дозволяє використовувати його для керування пулом однотипних ресурсів, де N визначає кількість одиниць ресурсу (дозволів), доступних для одночасного використання.

Функціонування багатозначного семафора базується на лічильнику. Операція P(S) перевіряє стан лічильника: якщо S>0, значення зменшується на одиницю, і потік продовжує виконання (входить у критичну секцію); якщо S=0, потік блокується і стає в чергу очікування. Операція V(S) збільшує значення лічильника і, якщо в черзі є заблоковані потоки, розблоковує один з них.

Для формалізації роботи системи було використано математичне моделювання. Продуктивність системи з N ресурсами можна оцінити через показники інтенсивності запитів (λ) та інтенсивності звільнення ресурсів (μ). Максимальна пропускна здатність системи λmax визначається як добуток кількості ресурсів на швидкість їх обробки: $\lambda_{max} \approx N \cdot \mu$. Важливим показником є ефективність використання ресурсів E, яка розраховується як відношення прискорення (Sp) до кількості паралельних ресурсів (N). В ідеальному випадку $E \to 1$, проте на практиці $E < 1$ через накладні витрати на синхронізацію та очікування в черзі.

Програмна реалізація моделі виконана мовою C++ з використанням бібліотек <*thread*>, <*mutex*> та <*condition_variable*> [3]. Архітектура системи включає клас *CountingSemaphore*, який інкапсулює логіку синхронізації. Використання *std::condition_variable* дозволяє уникнути активного очікування (busy wait), переводячи потоки в стан сну до моменту звільнення ресурсу, що значно знижує навантаження на процесор. Для збору статистики (час очікування, кількість одночасних потоків) використано атомарні змінні (*std::atomic*), що забезпечує коректність даних без додаткового блокування.

Аналіз результатів тестування наведено у таблицях 1 та 2.

Таблиця 1 – Залежність часу виконання від кількості дозволів (при 20 потоках)

Кількість дозволів (N)	Кількість потоків	Середній час очікування, мс	Загальний час виконання, мс
3	20	1 308,51	3 267
5		389,29	1 432
10		107,91	846
20		2,36	630

Як видно з табл. 1, збільшення кількості дозволів N з 3 до 20 призвело до скорочення загального часу виконання більш ніж у 5 разів.

Таблиця 2 – Залежність часу виконання від кількості потоків (при 4 дозволах)

Кількість дозволів (N)	Кількість потоків	Середній час очікування, мс	Загальний час виконання, мс
4	4	1,76	574
	15	350,29	1 405
	30	943,54	2 691
	50	1 726,31	4 224

Аналіз даних (табл.2) показує, що при фіксованому ресурсі збільшення кількості потоків призводить до різкого зростання середнього часу очікування.

Висновки

У результаті виконання роботи було успішно змодельовано та програмно реалізовано механізм багатозначного семафора як ключового засобу синхронізації у багатопотокових системах. Дослідження підтвердило, що семафори є ефективним інструментом для керування доступом до обмежених ресурсів, забезпечуючи атомарність операцій та запобігаючи стану гонки даних.

Аналіз результатів тестування дозволяє встановити чітку залежність між параметрами системи та її продуктивністю.

1. ***Вплив кількості дозволів (N)***. При фіксованому навантаженні збільшення кількості дозволів семафора призводить до суттєвого зменшення середнього часу очікування та загального часу виконання програми. Зокрема, збільшення N з 3 до 20 при 20 потоках дозволило скоротити загальний час виконання більш ніж у 5 разів (з 3267 мс до 630 мс). Це підтверджує, що для задач з високою інтенсивністю звернень до ресурсу збільшення ступеня паралелізму є критичним фактором підвищення швидкодії.
2. ***Вплив кількості потоків***. В умовах обмежених ресурсів (фіксоване N) зростання кількості потоків призводить до експоненційного збільшення часу очікування. Тестування показало, що при N=4 збільшення кількості потоків з 4 до 50 призвело до зростання загального часу виконання у понад 7 разів, що свідчить про утворення «вузького місця» та зростання накладних витрат на обслуговування черги очікування.

Отримані результати мають практичну цінність для проектування високонавантажених систем. Розроблена модель демонструє, що просте збільшення кількості робочих потоків без відповідного масштабування доступних ресурсів (дозволів семафора) є неефективним і може призвести до деградації продуктивності. Подальші дослідження можуть бути спрямовані на порівняльний аналіз семафорів з іншими примітивами синхронізації, такими як монітори, а також на реалізацію механізмів пріоритетних черг для оптимізації доступу до ресурсів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Коцовський В. М. Теорія паралельних обчислень: навчальний посібник. Ужгород: ПП «АУТДОР-Шарк», 2021. 188 с.
2. Луцків А.М., Лупенко С.А., Пасічник В.В. Паралельні та розподілені обчислення: Навч. посіб. Львів: ПП “Магнолія 2006”, 2024. 565 с.
3. Anthony Williams. C++ Concurrency in Action. 2nd Edition. Manning Publishing, 2019. 592 p. URL: https://readingtoday.site/download/c-concurrency-in-action-second-edition.

Струшинська Вероніка Віталіївна – студентка кафедри комп`ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: strusinskaveronika@gmail.com;

Денисюк Валерій Олександрович – канд. техн. наук, доцент, доцент кафедри комп’ютерних наук, Вінницький національний технічний університет, м.Вінниця, e-mail: vad64@i.ua.

Strushynska Veronika Vitaliivna– student of Computer Science Department, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: strusinskaveronika@gmail.com;

Denysiuk Valerii Olexandrovich – Ph.D., Assistant Professor, Assistant Professor of the Chair of Computer Science, Vinnytsia National Technical University, Vinnytsia, e-mail: vad64@i.ua.