Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра СШІ



**Звіт**

до лабораторної роботи № 4

з дисципліни

​*Операційні системи* ​

на тему:

“ Синхронізація потоків в ОС Windows ”

Виконав: студент КН-217

Мельничук Борис

Прийняв: Дмитро Петренко

Львів – 2022

## Лабораторна робота №4

**Тема.** Синхронізація потоків в ОС Windows

**Мета.** Ознайомитися зі способами синхронізації потоків. Навчитися організовувати багатопоточність з використанням синхронізації за допомогою функцій WinAPI.

**Завдання**.

1. Ознайомитися із синхронізаційними механізмами:
   1. М’ютекс
   2. Семафор
   3. Interlocked-функції
   4. Критична секція
   5. Спін-блокування
   6. Wait-Функції
2. Якщо реалізація задач складності 5 лабораторної роботи №2 вимагала використання синхронізаційних конструкцій – завдання лабораторної роботи №4 зводиться до обґрунтування вибору та пояснення можливих альтернатив.
3. Для алгоритму із лабораторної роботи №3 реалізувати завдання обраної складності із нижченаведеного списку.
4. Обгрунтувати вибір синхронізаційних конструкцій.
5. Результати виконання роботи відобразити у звіті.

## Варіанти.

**Для кожного рівня складності – усі попередні рівні теж входять у завдання**.

* [Складність 1] Щосекундне оновлення прогресу (у відсотках) виконання задачі.
* [Складність 2] Щосекундне оновлення проміжного результату (лише для задач зі скалярним результатом).
* [Складність 2,5] Щосекундне оновлення координат проміжного результату (лише для задач пошуку).
* [Складність 3] Можливість обмеження кількості потоків, які одночасно працюють із даними (щоб, наприклад, із 100 запущених потоків одночасно виконували обчислення лише 8, а інші чекали).

Проаналізуйте вплив обмеження на швидкодію виконання обчислень.

* [Складність 4] Реалізуйте альтернативний варіант синхронізації потоків.

Проаналізуйте переваги та недоліки обох використаних механізмів, а також – порівняйте їх швидкодію.

* [Складність 5] Забезпечте максимальну синхронізацію часу старту та прогресу опрацювання даних усіма потоками. Щосекунди оновлюйте на екрані прогрес опрацювання кожним потоком та фіксуйте розсинхронізацію (різницю мінімального та максимального прогресу). При цьому прогрес у пам’яті можна оновлювати і частіше для точнішої синхронізації, - проте знайдіть баланс між швидкодією та точністю синхронізації прогресу.

Порівняйте максимальну розсинхронізацію (у відсотках прогресу та часі закінчення роботи першого і останнього потоку) розробленого алгоритму із стандартним плануванням ОС Windows. Порівняйте, також, час роботи «максимально синхронного» алгоритму із стандартною версією.

Підказка для оптимізації швидкодії: кілька потоків можуть одночасно зчитувати одні і ті ж дані, не створюючи проблем, проте запис зазвичай вимагає бути ексклюзивним. Ця властивість особливо важлива, коли «читачів» є значно більше, ніж модифікацій.

## 1)

## 

## 2)

## 

## 

## 3)

Будемо досліджувати час виконання з к-стю потоків 20

|  |  |  |
| --- | --- | --- |
| К-сть потоків | Час виконання функції (в мілісекундах) | Час виконання функції (паралел.) (в мілісекундах) |
| 20 | 18464 | 13722 |

|  |  |  |
| --- | --- | --- |
| К-сть одночасних потоків | Час виконання функції (в мілісекундах) | Час виконання функції (паралел.) (в мілісекундах) |
| 2 | 20184 | 41955 |
| 3 | 20195 | 30783 |
| 4 | 20107 | 23031 |
| 5 | 20588 | 20135 |
| 6 | 20373 | 20734 |
| 7 | 20641 | 17850 |
| 8 | 20228 | 17435 |
| 9 | 20337 | 18160 |
| 10 | 20404 | 14725 |
| 11 | 20239 | 14482 |
| 12 | 20497 | 16157 |
| 13 | 20409 | 61808 |
| 14 | 20637 | 15577 |
| 15 | 20297 | 15902 |
| 16 | 20762 | 15869 |
| 17 | 20464 | 16522 |
| 18 | 20190 | 16420 |
| 19 | 20107 | 17729 |
| 20 | 20128 | 14122 |