**Лабораторна робота №2**

Шевченко Н. В.

**Тема:** Засоби синхронізації потоків.

**Мета:** Познайомитись із стандартними засобами синхронізації потоків різними мовами програмування.

**Ada**

Цей код на мові програмування Ada реалізує пошук мінімального елемента у масиві, розділяючи завдання пошуку мінімума між кількома потоками.

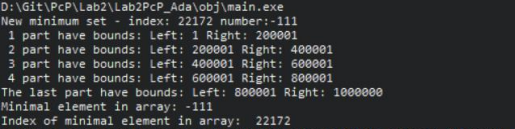
Починається програма з оголошення необхідних модулів, таких як Ada.Text\_IO для виводу у консоль, Ada.Integer\_Text\_IO для взаємодії з цілими числами та Ada.Float\_Text\_IO для взаємодії з числами з плаваючою комою. Крім того, використовується Ada.Numerics.Discrete\_Random для генерації випадкових цілих чисел.

Далі визначений тип завдання MinElementSearchTask, яке буде відповідати за пошук мінімального елемента. У тілі цього завдання визначено змінні для кількості потоків, кількості елементів у масиві, сам масив, а також тип protected, який використовується для збереження мінімального індексу та кількості завдань. Також визначено тип завдання ElementSearcher, який буде відповідати за пошук мінімального елемента в межах певного діапазону.

У тілі завдання MinElementSearchTask описано процедуру пошуку мінімального індексу, яка виконується в кожному потоці ElementSearcher. Також визначені процедури для ініціалізації масиву та встановлення випадкового мінімального елемента.

Далі визначена процедура ParallelSearch, яка розділяє завдання пошуку мінімума між кількома потоками та викликається в тілі MinElementSearchTask.

В кінці програми оголошений екземпляр завдання MinElementSearchTask, який виконується при запуску програми.



**Java**

Цей код на Java демонструє пошук мінімального елемента в масиві, розділяючи цю операцію між кількома потоками.

У класі Main в методі main спочатку задається довжина масиву arrayLength та кількість потоків numThreads. Створюється масив array за допомогою методу generateArray класу ArrayProcessor, а також отримується індекс мінімального елемента за допомогою методу getMinIndex того ж класу.

Потім виводиться згенерований масив і інформація про мінімальний елемент та його індекс. Створюється масив потоків threads розміром numThreads. Задається розмір блоку обробки blockSize, який визначається як arrayLength поділена на кількість потоків.

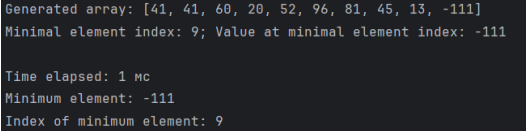
Починається вимірювання часу виконання операції. Для кожного потоку створюється об'єкт MinElementFinderRunnable, який отримує підмасив масиву array для обробки, початковий та кінцевий індекси, а також посилання на об'єкт minElementFinder. Потім кожний потік запускається.

Після цього викликається метод join() для кожного потоку, щоб чекати їх завершення. Після завершення виконання потоків обчислюється час, що пройшов. Виводиться інформація про час виконання, мінімальний елемент та його індекс за допомогою методів об'єкта minElementFinder.

Клас ArrayProcessor відповідає за генерацію масиву. Метод generateArray генерує масив випадкових чисел та встановлює випадковий елемент як мінімальний. Метод getMinIndex повертає індекс цього мінімального елемента.

Клас MinElementFinder відстежує мінімальний елемент та його індекс. Метод updateMin використовується для оновлення мінімального елемента та його індексу, якщо новий елемент менший за поточний. Методи getMin та getMinIndex повертають відповідно мінімальний елемент та його індекс.

Клас MinElementFinderRunnable реалізує інтерфейс Runnable і містить логіку для пошуку мінімального елемента в підмасиві масиву. Під час виконання потоку, він шукає мінімальний елемент та його індекс у своїй частині масиву та оновлює значення в об'єкті MinElementFinder.



**Висновок**

Розглянуто синхронізацію потоків на прикладі паралельного пошуку мінімуму в масиві мовами Ada та Java. В Ada використано типи task та protected object, в Java - Thread та Runnable. Масив розбито на частини, кожен потік шукав мінімум у своїй частині. В Ada синхронізація через захищений об'єкт, в Java - через методи об'єкту. Обидві реалізації показали прискорення завдяки розпаралелюванню порівняно з послідовним виконанням.