

Лабораторна робота №3

Тема: Розширені засоби синхронізації потоків (семафори).

Мета: Познайомитись із використанням семафорів як засобів синхронізації потоків у різних мовах програмування.

Теоритичні відомості:

Для організації коректної роботи декількох потоків, які мають доступ до спільних ресурсів, використовують синхронізацію потоків. Одним із засобів є семафори — механізм, який дозволяє контролювати кількість потоків, що можуть одночасно працювати з ресурсом.

У Java та C# є клас Semaphore, який реалізує цю ідею.

Завдання:

Реалізувати класичну задачу "Виробник-Споживач" з обмеженим буфером, використовуючи семафори для синхронізації:

- Буфер має максимальну місткість (5 елементів).
- Є довільна кількість виробників і споживачів.
- Виробники не додають у заповнений буфер.
- Споживачі не забирають з порожнього буфера.
- Загальна кількість "продукції" — 20 елементів.

Хід роботи:

1. Створено клас ProducerConsumer з внутрішніми класами Producer та Consumer.
2. Для синхронізації використано три семафори: **empty** для контролю порожніх місць у буфері, **full** для контролю кількості доступних елементів, **mutex** для взаємного виключення при доступі до буфера.
3. Запуск кількох потоків-виробників і споживачів.
4. Кожен потік виробляє/споживає задану кількість продукції.

Результат:

Java:

```
Thread-0 produced 0
Thread-0 produced 1
Thread-0 produced 2
Thread-0 produced 3
Thread-1 produced 0
Thread-3 consumed 0
Thread-3 consumed 1
Thread-3 consumed 2
Thread-4 consumed 3
Thread-2 consumed 0
Thread-0 produced 4
Thread-0 produced 5
Thread-0 produced 6
Thread-0 produced 7
Thread-1 produced 1
Thread-3 consumed 4
Thread-3 consumed 5
Thread-3 consumed 6
Thread-4 consumed 7
Thread-2 consumed 1
Thread-0 produced 8
Thread-0 produced 9
Thread-1 produced 2
Thread-1 produced 3
Thread-1 produced 4
Thread-4 consumed 8
Thread-4 consumed 9
Thread-2 consumed 4
Thread-1 produced 5
Thread-1 produced 6
Thread-1 produced 7
Thread-2 consumed 5
Thread-2 consumed 6
Thread-2 consumed 7
Thread-1 produced 8
Thread-1 produced 9
```

C#:

```
P1 produced 0
P2 produced 0
P1 produced 1
P1 produced 2
P2 produced 1
C3 consumed 0
C3 consumed 0
C3 consumed 1
P1 produced 3
P1 produced 4
P2 produced 2
C2 consumed 2
C2 consumed 1
C1 consumed 3
C3 consumed 4
P1 produced 5
P1 produced 6
P1 produced 7
C2 consumed 2
C2 consumed 5
P2 produced 3
P2 produced 4
P1 produced 8
C1 consumed 6
C1 consumed 7
C1 consumed 3
C2 consumed 4
P2 produced 5
P2 produced 6
P2 produced 7
C3 consumed 8
C3 consumed 5
P1 produced 9
C2 consumed 6
C1 consumed 7
C1 consumed 9
P2 produced 8
P2 produced 9
```

Висновок:

Лабораторна робота дозволила закріпити навички використання семафорів у Java і C# для синхронізації потоків, забезпечити правильний доступ до

Гаврилюк А.В.

спільного ресурсу (буфера) та реалізувати класичну задачу "Виробник-Споживач" з довільною кількістю потоків.

Посилання на репозиторії з вихідним кодом:

<https://github.com/AnnaHavryliuk4/ParallelProcessesCourse/tree/main/lab3>