

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА

Факультет прикладної математики та інформатики

Кафедра програмування

Паралельні та розподілені обчислення

ЛАБОРАТОРНА РОБОТА №4

Тема: «Форми»

Виконав:

Ст. Лук'янчук Денис

Група ПМІ-33

2025

Тема: «Форми»

Мета роботи: Ознайомитися з принципами багатопоточності та синхронізації потоків, навчитися відображати час виконання та простою потоків на графічних елементах для аналізу ефективності взаємодії потоків.

Хід роботи

1. Підготовка графічного інтерфейсу

Я створив форму з трьома вертикальними панелями, які символізують циліндри для кожного потоку. Для зручності користувача додав текстові поля, куди можна вводити початкові тривалості роботи потоків (T1,1, T2,1, T3,1). Також на панелі розмістив кнопки для запуску симуляції, скидання результатів і генерації випадкових інтервалів. Додатково підписав кожен циліндр і встановив кольорову легенду: синій — робота, червоний — очікування.

2. Ініціалізація та запуск потоків

Після натискання кнопки «Старт» програма зчитує введені значення часу або генерує випадкові інтервали. Кожен з трьох потоків запускається у власному циклі: перший потік працює заданий час, потім передає наступний час виконання другому потоку. Другий потік, завершивши свою роботу, передає величину часу третьому потоку, а третій — першому. Така ланцюгова передача створює взаємодію потоків та показує, як один потік чекає на завершення іншого.

3. Синхронізація та відображення активності

Для синхронізації потоки обмінюються даними через спільні змінні. Якщо потік ще не отримав новий час виконання від іншого потоку, він перебуває у стані очікування. Час активної роботи відображається синім кольором на відповідному циліндрі, а час простою — червоним. Висота сегментів динамічно змінюється під час виконання, що дозволяє наочно бачити прогрес роботи потоку. Завершені сегменти додаються до списку, щоб зберегти історію виконання кожного потоку.

4. Детальна робота циклів і очікування

Кожен потік виконує кілька циклів роботи. Якщо під час циклу потік повинен чекати, він періодично перевіряє, чи отримав нове значення часу. У цей момент на циліндрі з'являються червоні сегменти, що показують простої. Це дозволяє оцінити, наскільки ефективно потоки взаємодіють і чи виникають затримки.

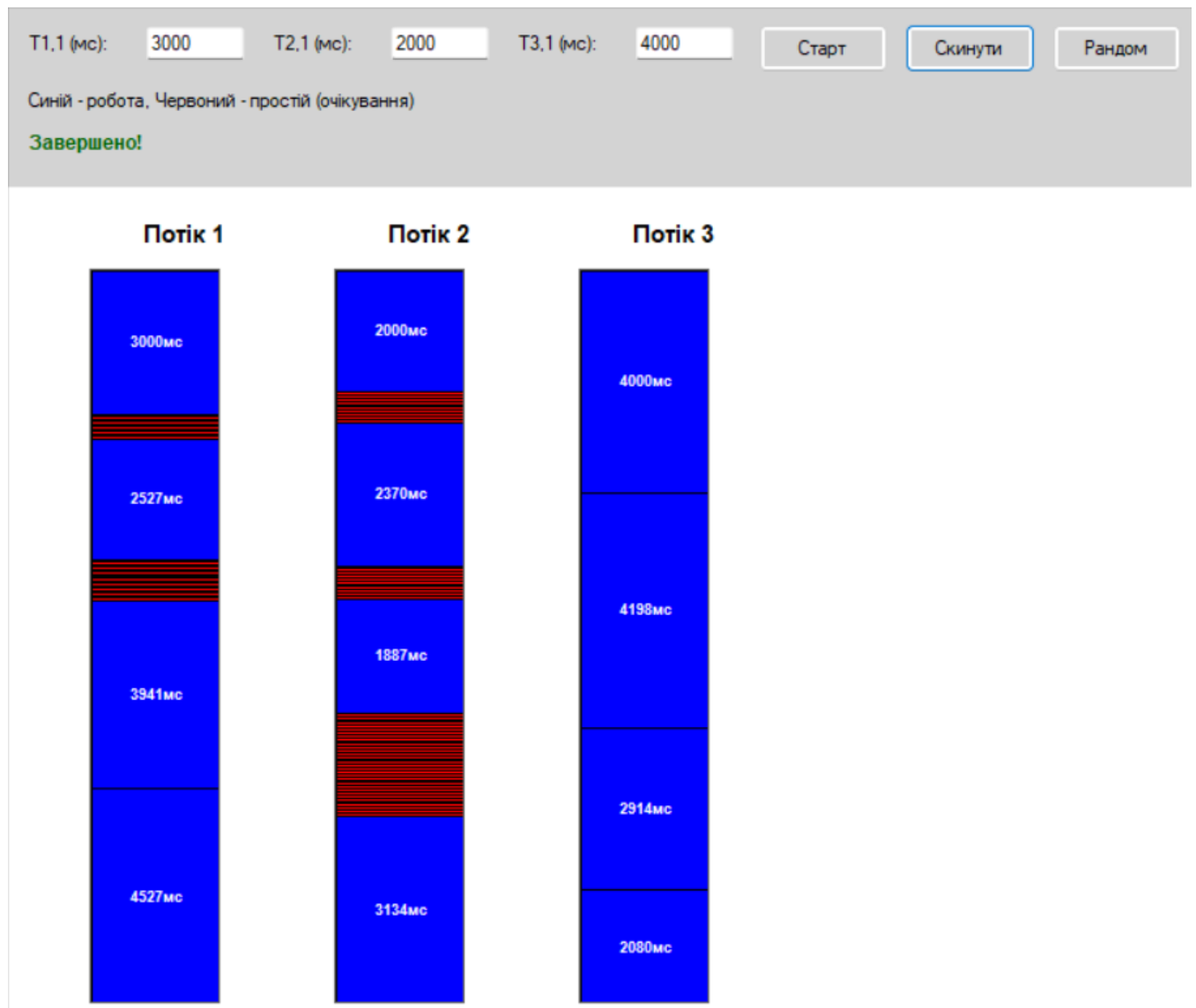
5. Візуалізація та оновлення інтерфейсу

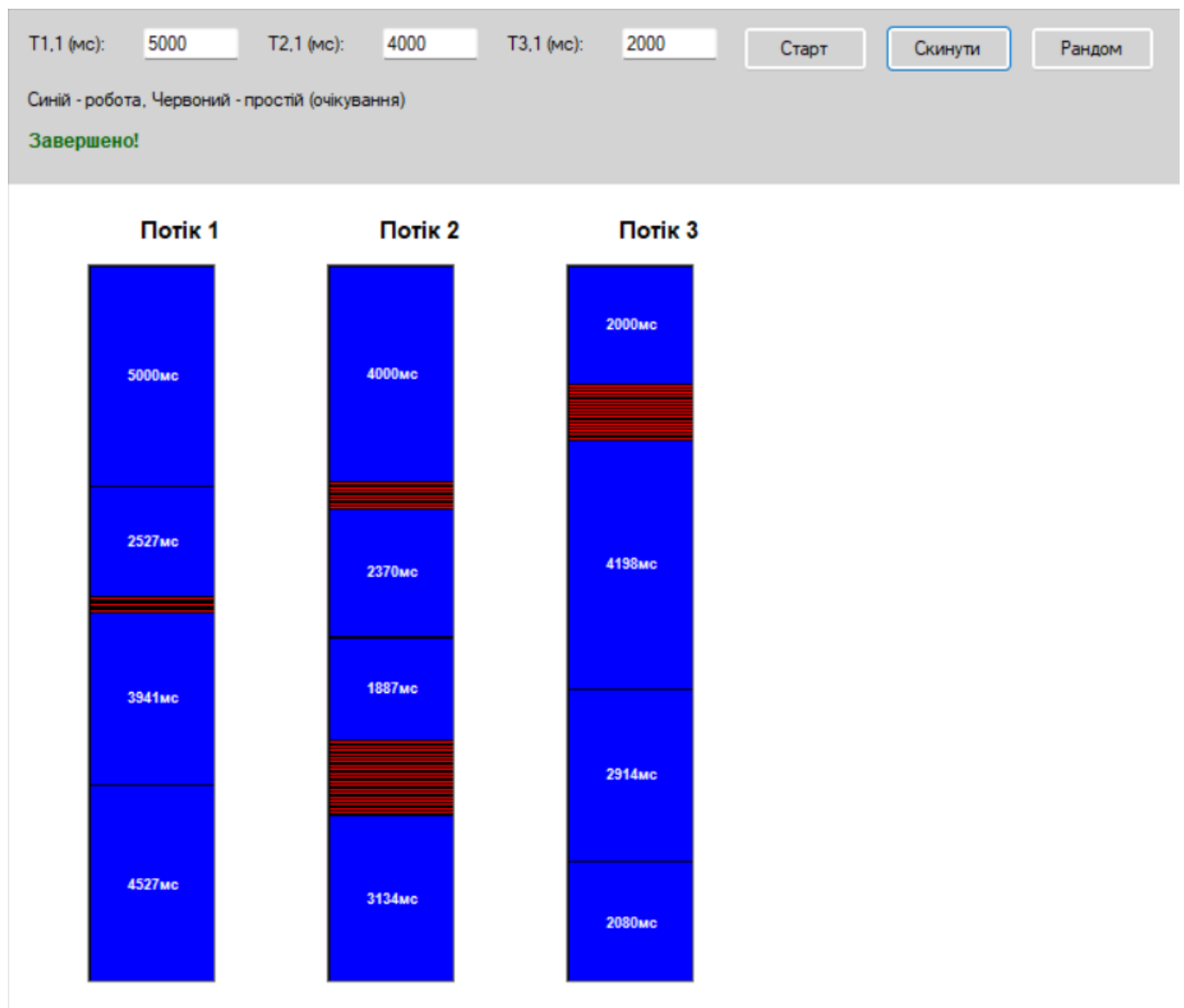
Візуалізація оновлюється за допомогою таймера, який періодично перерисовує циліндри. Поточний стан роботи кожного потоку відображається у реальному часі, а завершені сегменти залишаються на циліндрі. Для незавершених сегментів використовується пунктирна рамка, щоб відрізнити активний процес від ще не виконаної частини.

6. Перевірка та аналіз результатів

Програму запускали кілька разів із різними фіксованими та випадковими інтервалами часу. Це дало змогу наочно побачити чергування роботи та простою потоків, оцінити ефективність їхньої взаємодії та порівняти результати для різних сценаріїв. Завдяки графічній візуалізації можна легко відслідкувати моменти очікування і активного виконання кожного потоку.

Приклади:





Висновок: У лабораторній роботі я ознайомився з принципами багатопоточності та синхронізації потоків у програмуванні. Було реалізовано графічний інтерфейс із трьома прямокутниками, що наочно відображають активну роботу потоків синім кольором та час простою червоним.

Під час виконання програми потоки працювали циклічно та передавали один одному час виконання, що дозволило побачити їхню взаємодію та затримки у реальному часі. Запуск програми з різними фіксованими та випадковими інтервалами дав змогу проаналізувати ефективність роботи потоків та оцінити моменти очікування.

Таким чином, лабораторна робота дозволила на практиці зрозуміти механізми синхронізації, чергування роботи та простою потоків, а також навчитися візуалізувати їхню діяльність для подальшого аналізу ефективності взаємодії.