

## Лабораторна робота №1

### Процеси та потоки

**Мета:** ознайомитись з процесами та потоками в операційній системі, вивчити основні принципи взаємодії між процесами, розподіл даних між процесами та роботу з файлами, які відображуються у пам'ять.

**Завдання 1.** Створив три проекти у VS: два консольних (створення даних та сортування) та один віконний (відображення даних). Таким чином рішення має наступний вигляд:

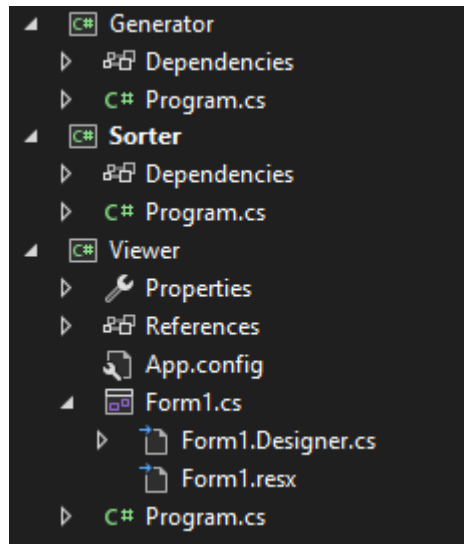


Рисунок 1. Структура рішення.

Перша програма створює дані для сортування та має наступний вигляд під час роботи:

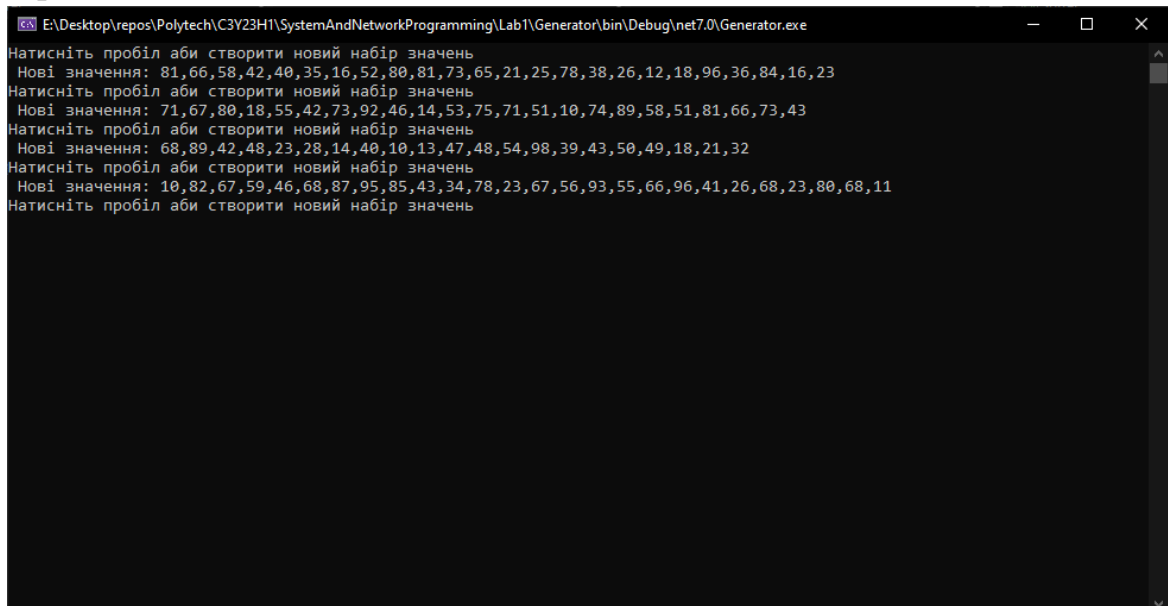


Рисунок 2. Проєкт Generator.

					ДУ «Житомирська політехніка».23.121.06.000 – Лр1			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Каліновський Д.			Звіт з лабораторної роботи		Лім.	Арк.
Перевір.		Петросян А.Р.						1
Керівник								5
Н. контр.							ФІКТ Гр. ІПЗк-21-1	
Зав. каф.								

Друга програма сортує дані та має наступний вигляд:

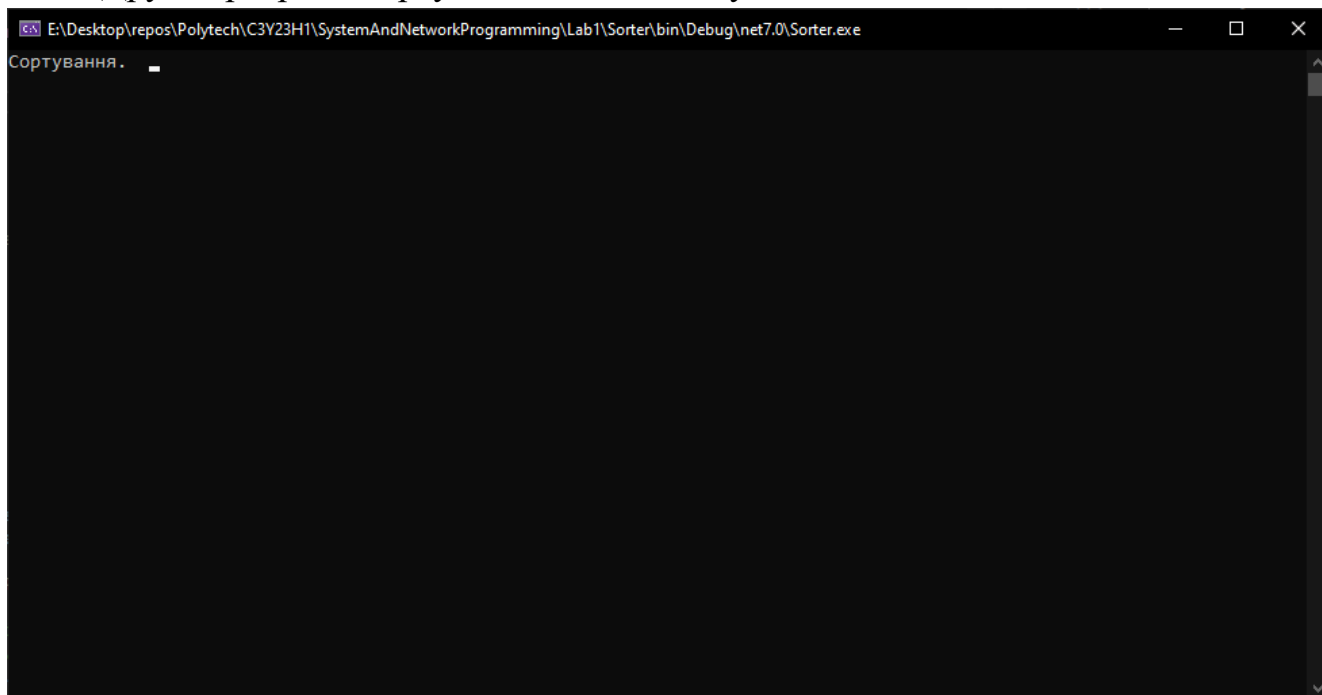


Рисунок 3. Проєкт Sorter.

Третя програма відображає масив у вигляді списку зірочок:

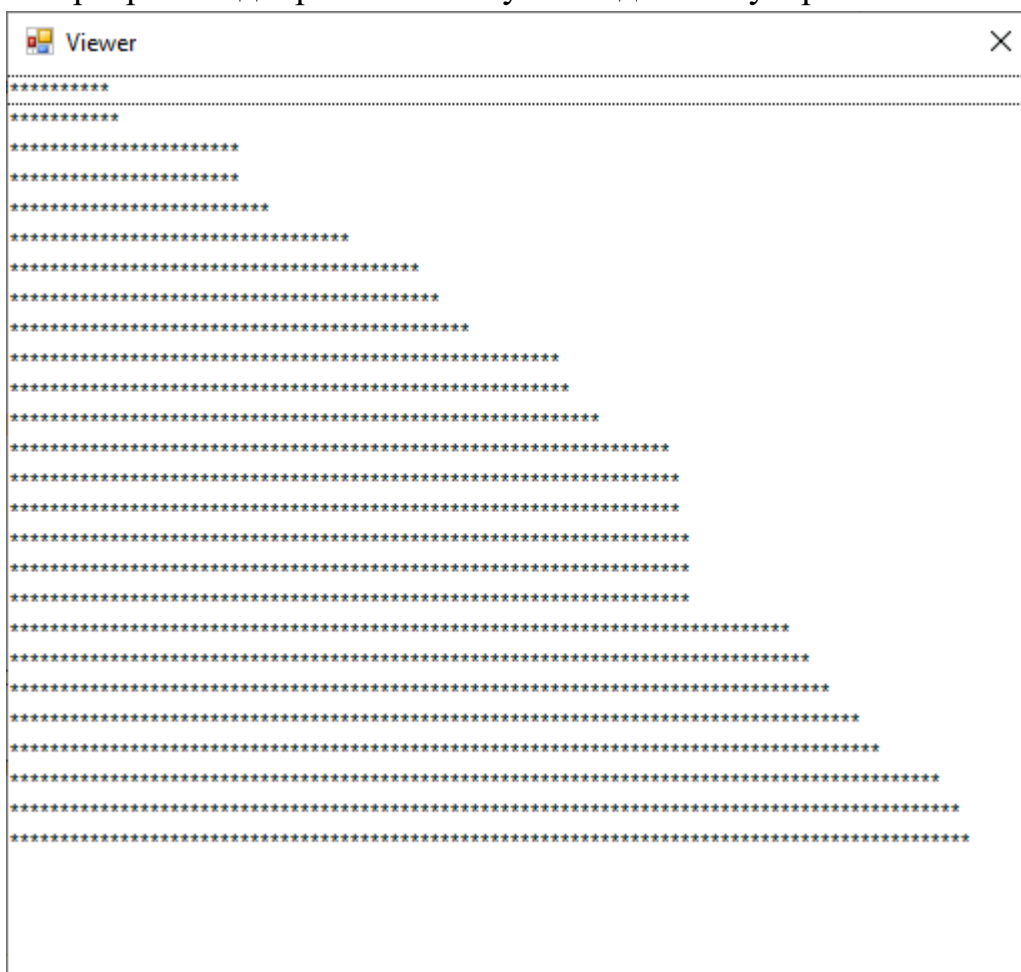


Рисунок 4. Відсортований масив у проєкті Viewer.

		Каліновський Д.			ДУ «Житомирська політехніка».23.121.06.000 – Лр1	Арк.
		Петросян А.Р.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

**Завдання 2.** До існуючих проєктів додав синхронізацію за допомогою м'ютекса та обробку виключень. Наприклад, сортувальник отримав наступний блок коду, що забезпечує завершення виконання методу у випадку неможливості отримання даних чи м'ютекса:

```
try
{
    data = MemoryMappedFile.OpenExisting("Data", MemoryMappedFileRights.ReadWrite);
}
catch (FileNotFoundException ex)
{
    Debug.WriteLine(ex.Message);
    var new_ex = new Exception("Не знайдено даних для сортування");
    throw new_ex;
}

bool mutex_status;
try
{
    mutex_status = mutex.WaitOne(1500);
}
catch (AbandonedMutexException ex)
{
    Debug.WriteLine(ex.Message);
    var new_ex = new Exception("М'ютекс назавжди заблоковано іншим процесом");
    throw new_ex;
}
if (!mutex_status)
{
    var new_ex = new Exception("Неможливо заблокувати м'ютекс");
    throw new_ex;
}
```

Ці блоки не єдині зі схожим функціоналом у цьому проєкті. Подібні блоки отримали й інші проєкти.

**Додаткове завдання.** Додав четвертий проєкт, використавши у ньому алгоритм QuickSort. Перевіряю правильність роботи проєкту:

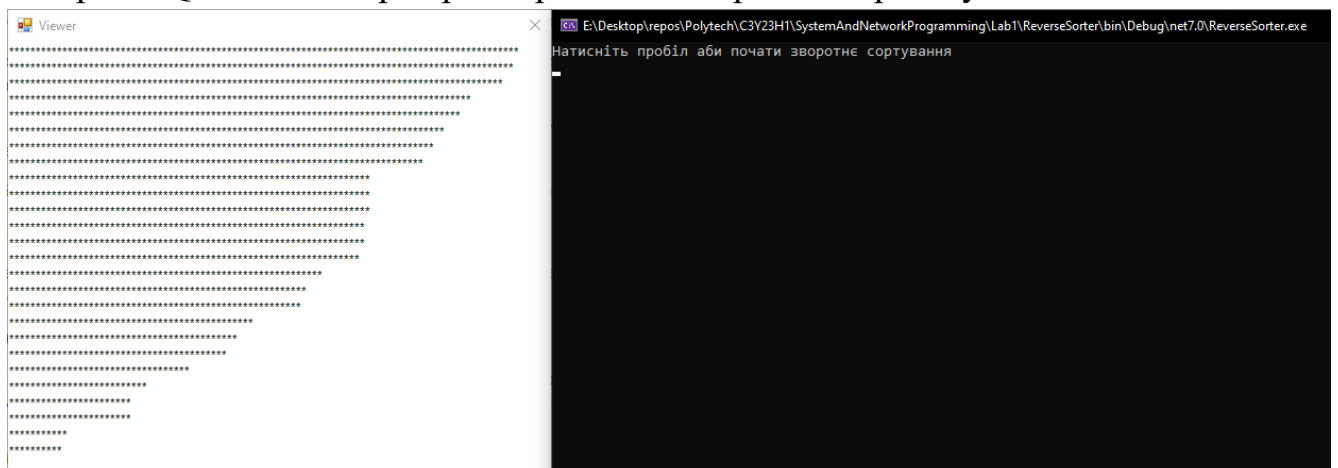


Рисунок 5. Масив відсортовано у зворотному порядку.

У результаті отримав 4 застосунки, один з яких створює значення та записує їх у відповідний файл, два інші намагаються відсортувати ці значення за власними правилами, а четвертий відображає масив у наочній формі.

Приклади роботи усіх 4-х застосунків одночасно:

		Каліновський Д.			ДУ «Житомирська політехніка».23.121.06.000 – Пр1	Арк.
		Петросян А.Р.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

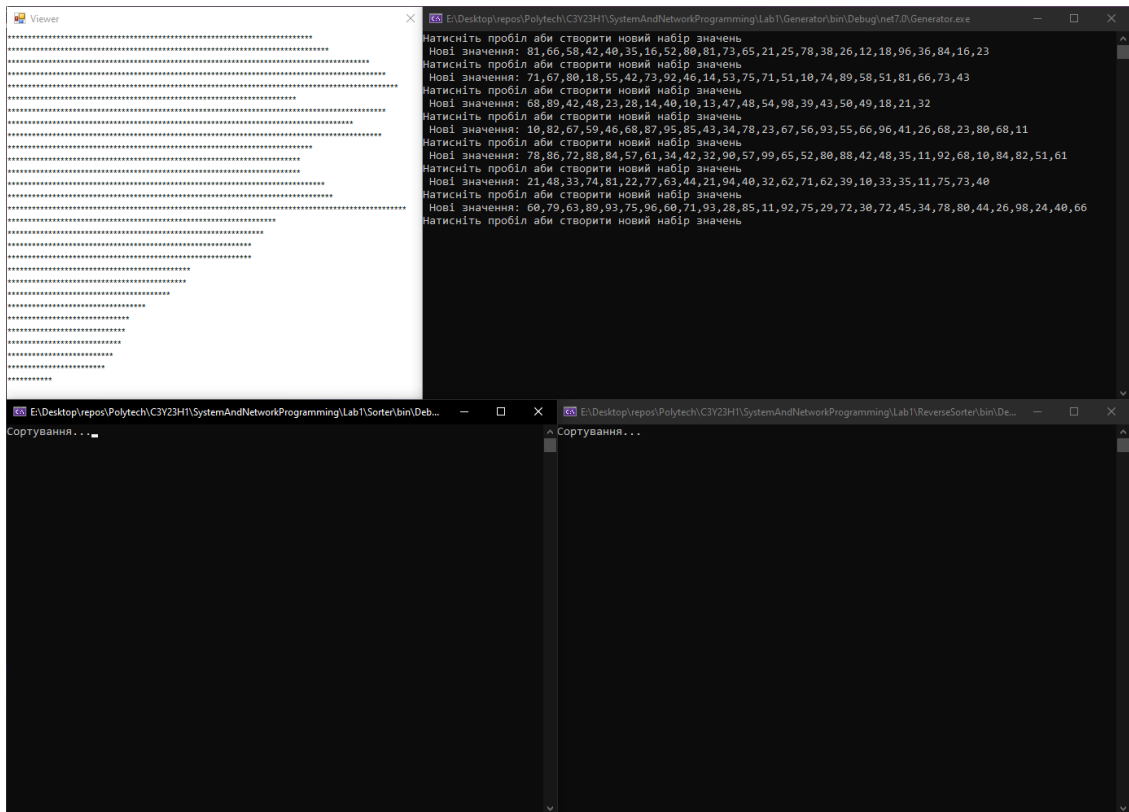


Рисунок 6. Вигляд застосунків після кількох ітерацій.

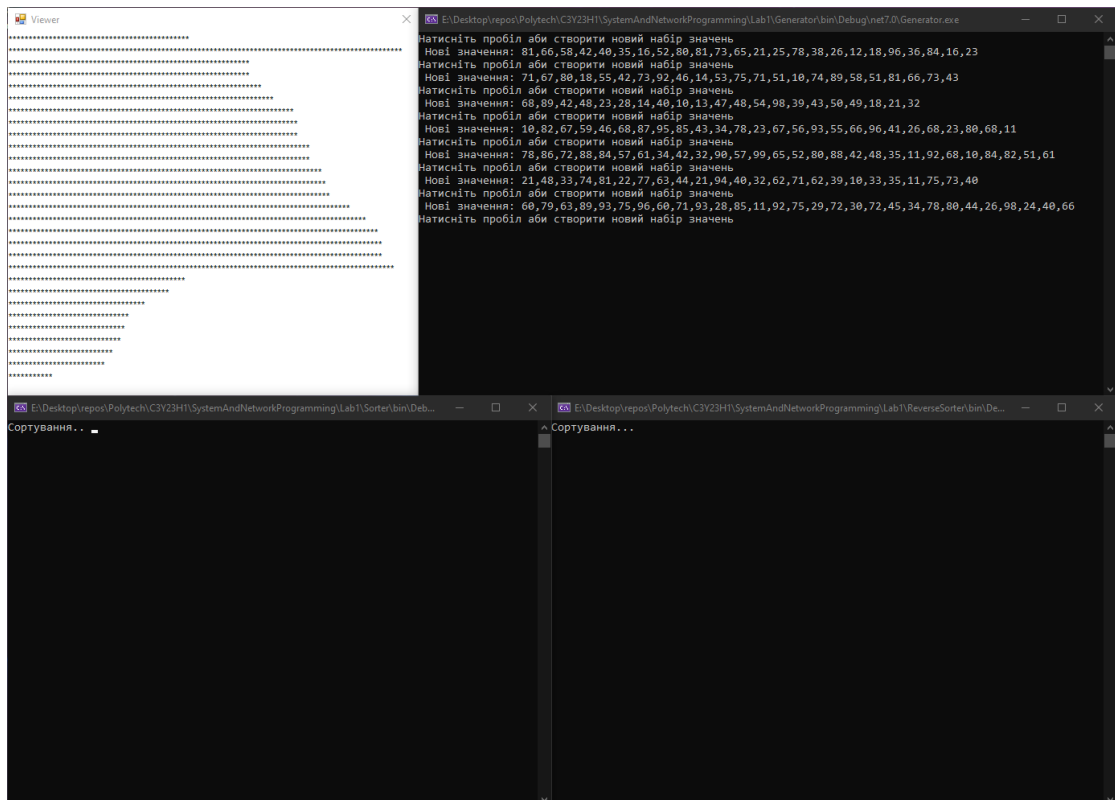


Рисунок 7. Вигляд застосунків після ще декількох ітерацій.

		Каліновський Д.			ДУ «Житомирська політехніка».23.121.06.000 – Пр 1	Арк.
		Петросян А.Р.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

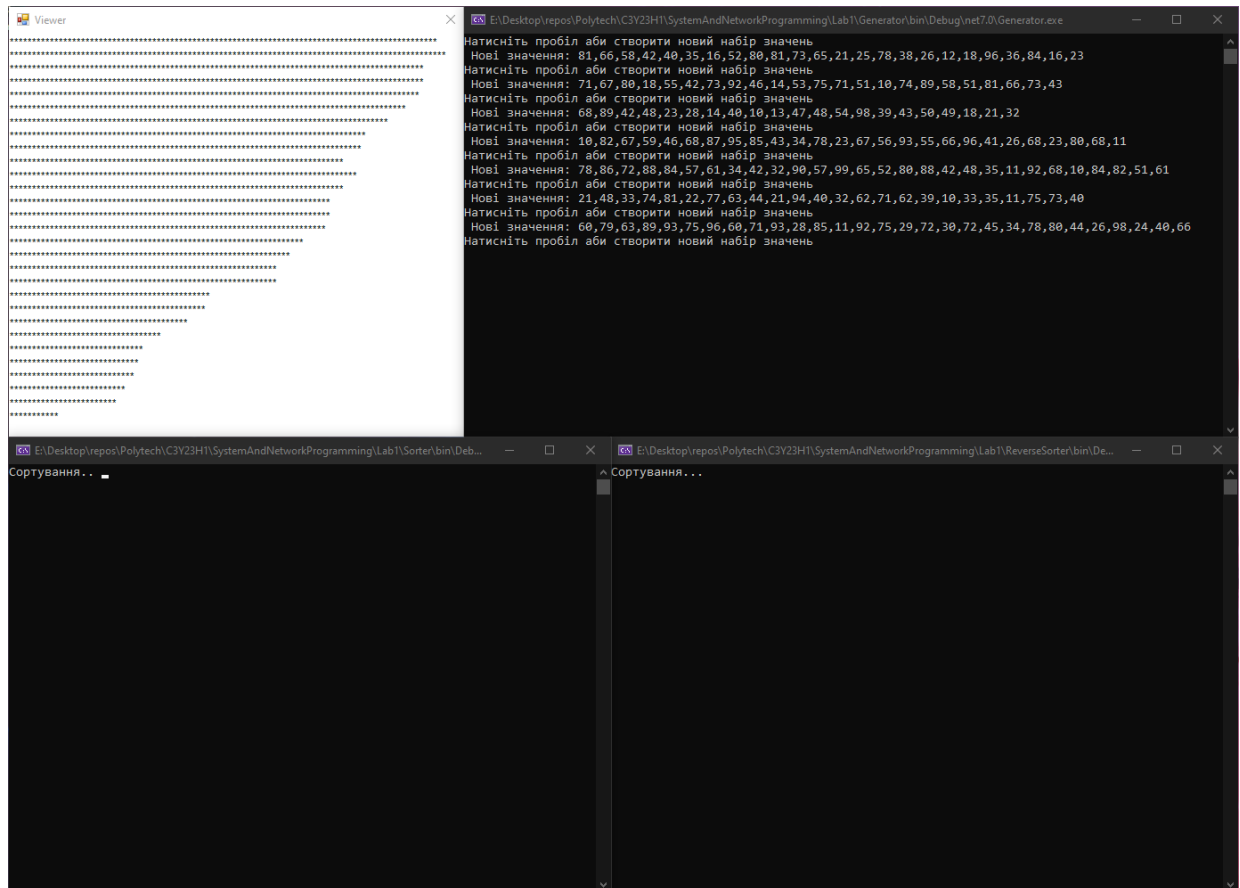


Рисунок 8. Вигляд застосунків наприкінці тестування.

Оскільки я змінив період сну для другого сортувальника таким чином, аби боротьба тривала умовно вічно, то за весь час тестування жодна з програм не змогла повністю відсортувати масив за власними правилами.

**Висновок:** підчас виконання даної лабораторної роботи я ознайомився з процесами та потоками в операційній системі, вивчив основні принципи взаємодії між процесами, розподіл даних між процесами та роботу з файлами, які відображуються у пам'ять.

		Каліновський Д.			ДУ «Житомирська політехніка».23.121.06.000 – Пр 1	Арк.
		Петросян А.Р.				5
Змн.	Арк.	№ докум.	Підпис	Дата		