

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

ІКНІ
Кафедра ПЗ



ЗВІТ

До лабораторної роботи №5

на тему: “Багатопоточність в операційній сисемі WINDOWS. Створення, керування та синхронізація потоків.”

з дисципліни: “Операційні системи”

Лектор:
старший викладач кафедри ПЗ
Грицай О.Д.

Виконав:
студент групи ПЗ-24
Губик А. С.

Прийняв:
доцент кафедри ПЗ
Горечко О. М.

Тема роботи:Багатопоточність в операційній системі WINDOWS. Створення, керування та синхронізація потоків.

Мета роботи:Ознайомитися з багатопоточністю в ОС Windows. Навчитись реалізовувати розпаралелення алгоритмів за допомогою багатопоточності в ОС Windows з використанням функцій WinAPI. Навчитись використовувати різні механізми синхронізації потоків.

Теоретичні відомості

Розглядаючи поняття процесу, визначають ще одну абстракцію для запущеного процесу: потік. У класичному уявленні існує єдина точка виконання в рамках програми (тобто єдиний потік контролю, на якому збираються та виконуються інструкції), багатопотокова програма має більш ніж одну точку виконання (тобто кілька потоків контролю, кожен з яких який отримується та виконується). Кожен потік дуже схожий на окремий процес, за винятком однієї відмінності: вони мають спільний адресний простір і, отже, мають доступ до одних і тих же даних. Таким чином, стан одного потоку дуже подібний до стану процесу. Він має лічильник програм (PC), який відстежує, звідки програма отримує інструкції. Кожен потік має свій власний приватний набір реєстрів, який він використовує для обчислень; таким чином, якщо на одному процесорі працюють два потоки, при переході від запуску одного (T1) до запуску іншого (T2) має відбутися перемикання контексту. Контекстний перемикач між потоками дуже подібний до перемикання контекстів між процесами, оскільки перед запуском T2 необхідно зберегти регістр стану T1 і відновити стан реєстру T2. За допомогою процесів ми зберегли стан до блоку управління процесами (PCB); тепер нам знадобиться один або кілька блоків управління потоками (TCB) для збереження стану кожного потоку процесу. Однак є одна істотна відмінність у перемиканні контексту, який ми виконуємо між потоками порівняно з процесами: адресний простір залишається незмінним (тобто немає необхідності змінювати, яку таблицю сторінок ми використовуємо). Ще одна істотна відмінність між потоками та процесами стосується стека. У простій моделі адресного простору класичного процесу (однопотокового) є єдиний стек, який зазвичай знаходиться внизу адресного простору. Однак у багатопотоковому процесі кожен потік працює окремо і, звичайно, може залучати різні підпрограми для виконання будь-якої роботи. Замість одного стека в адресному просторі буде по одному на кожен потік.

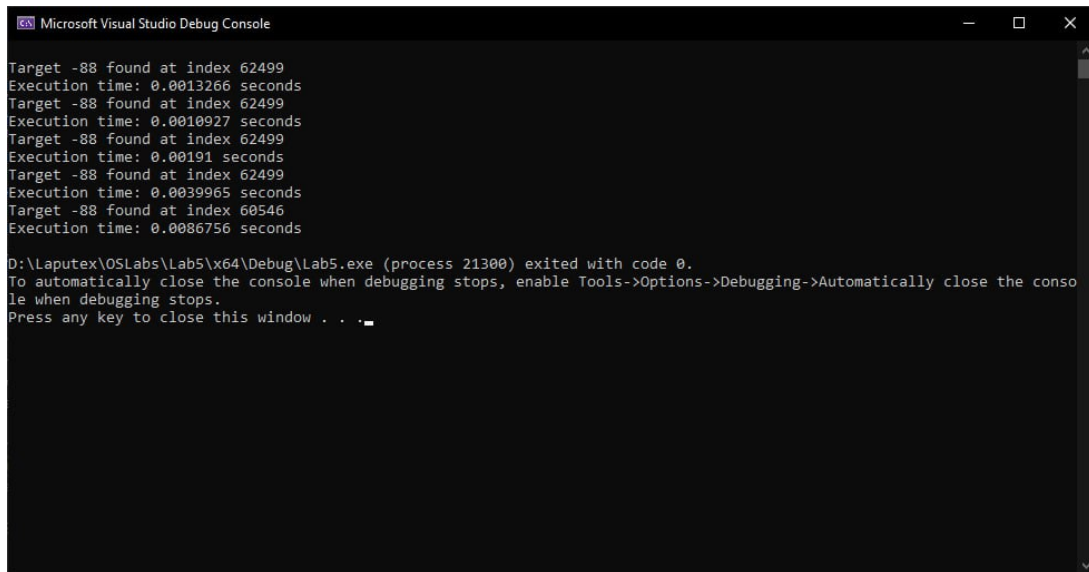
Індивідуальне завдання

Бінарний пошук максимального елемента масиву (кількість елементів >10000, елементи випадкові). Вивести значення та індекс. Синхронізація: 2, 8

1. Мьютекс
2. Семафор
3. Інтерлок-функції
4. Критична секція
5. Події
6. Функції очікування
7. Алгоритм активного очікування
8. Алгоритм монітору
9. Таймери очікування

Хід роботи

I. Розпаралелення бінарного пошуку на 2, 4, 8, 16 потоків

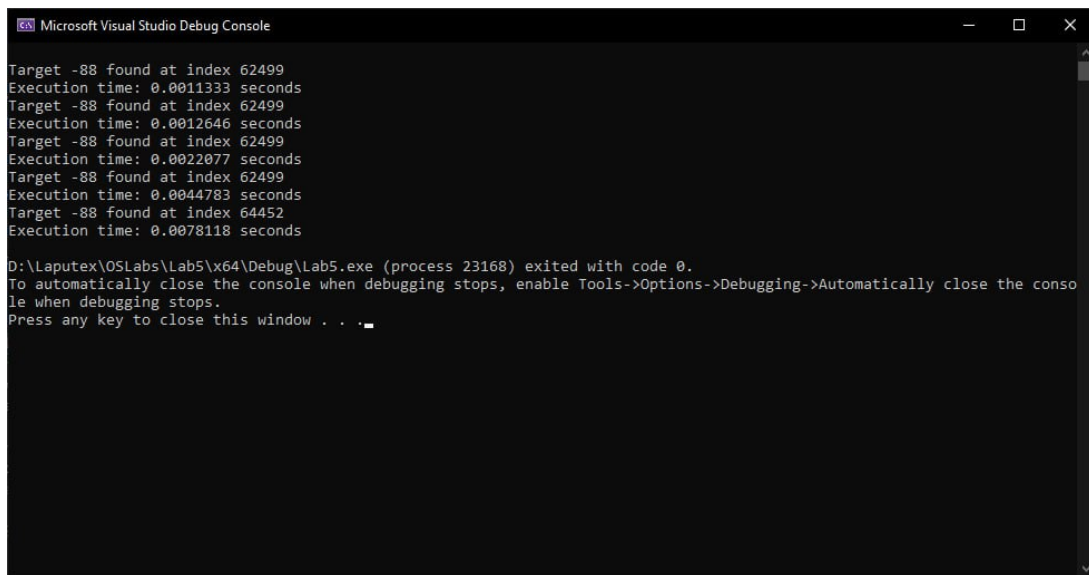


```
Microsoft Visual Studio Debug Console

Target -88 found at index 62499
Execution time: 0.0013266 seconds
Target -88 found at index 62499
Execution time: 0.0010927 seconds
Target -88 found at index 62499
Execution time: 0.00191 seconds
Target -88 found at index 62499
Execution time: 0.0039965 seconds
Target -88 found at index 60546
Execution time: 0.0086756 seconds

D:\Laputex\OSLabs\Lab5\x64\Debug\Lab5.exe (process 21300) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Рис. 1: Без синхронізації

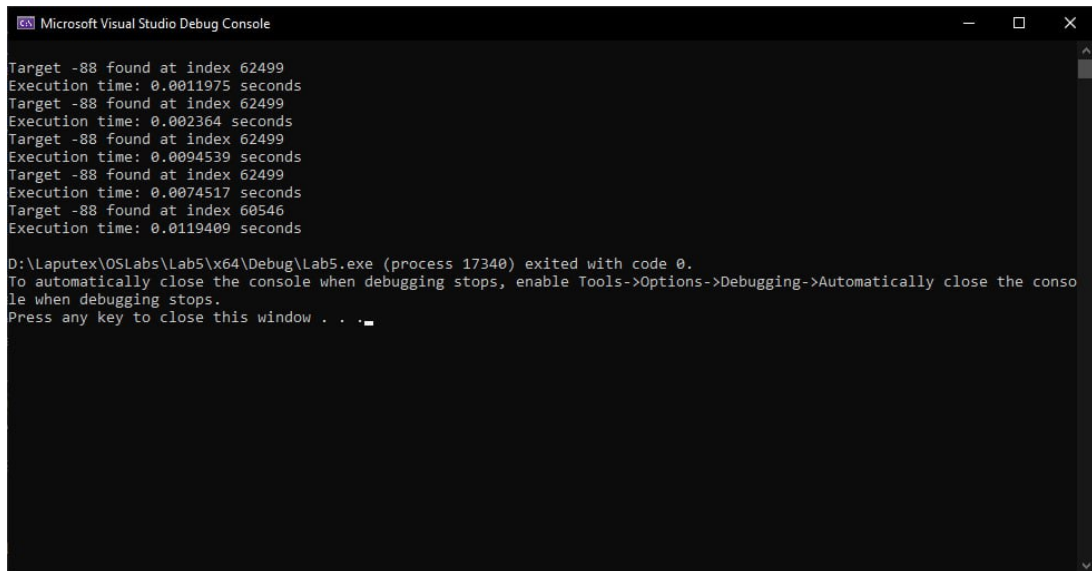


```
Microsoft Visual Studio Debug Console

Target -88 found at index 62499
Execution time: 0.0011333 seconds
Target -88 found at index 62499
Execution time: 0.0012646 seconds
Target -88 found at index 62499
Execution time: 0.0022077 seconds
Target -88 found at index 62499
Execution time: 0.0044783 seconds
Target -88 found at index 64452
Execution time: 0.0078118 seconds

D:\Laputex\OSLabs\Lab5\x64\Debug\Lab5.exe (process 23168) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Рис. 2: З використанням семафори



```
Microsoft Visual Studio Debug Console

Target -88 found at index 62499
Execution time: 0.0011975 seconds
Target -88 found at index 62499
Execution time: 0.002364 seconds
Target -88 found at index 62499
Execution time: 0.0094539 seconds
Target -88 found at index 62499
Execution time: 0.0074517 seconds
Target -88 found at index 60546
Execution time: 0.0119409 seconds

D:\Laputex\OSLabs\Lab5\x64\Debug\Lab5.exe (process 17340) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Рис. 3:

Як бачимо синхронізація сповільнює роботу і не впливає на правильність виконання бінарного пошуку.

II. Зупинка потоків

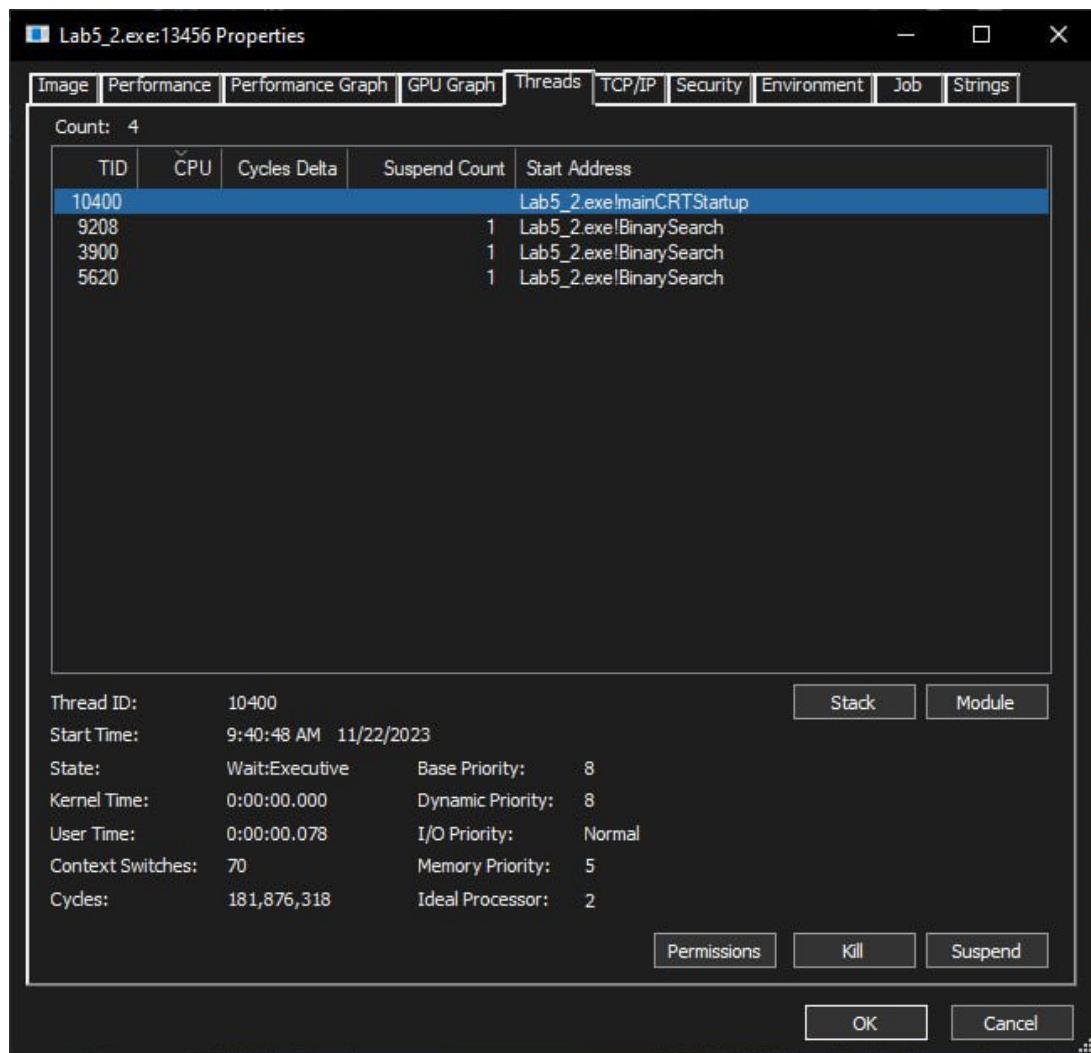


Рис. 4: Process explorer properties

Висновок: Ефективність синхронізації залежить від поставленої перед нами задачі. У випадку з бінарним пошуком воно не потрібне і сповільнює роботу.