

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»  
Факультет прикладної математики  
Кафедра прикладної математики

Звіт  
з лабораторної роботи №3  
з дисципліни Розподілені і хмарні обчислення на тему:  
«OpenMP. Обчислення визначеного інтегралу»

Виконав:  
студент групи КМ-81  
Піткевич І.В.

Керівник:  
Ст.викладач  
Ліскін В. О.

## ЗМІСТ

ВСТУП	3
I ПОСТАНОВКА ЗАДАЧІ	4
II РЕЗУЛЬТАТ РОБОТИ ПРОГРАМИ	5
ВИСНОВОК	6

## ВСТУП

OpenMP (Open Multi-Processing) — це набір директив компілятора, бібліотечних процедур та змінних середовища, які призначені для програмування багатопоточних застосунків на багатопроцесорних системах із спільною пам'яттю на мовах C, C++ та Fortran.

### Переваги OpenMP:

- За рахунок ідеї «інкрементального розпаралелювання» OpenMP ідеально підходить для розробників, що прагнуть швидко розпаралелювати свої обчислювальні програми з великими паралельними циклами. Розробник не створює нову паралельну програму, а просто послідовно додає в текст програми OpenMP-директиви.
- При цьому, OpenMP — досить гнучкий механізм, що надає розробникові великі можливості контролю над поведінкою паралельної програми.
- Передбачається, що OpenMP-програма на однопроцесорній платформі може бути використана як послідовна програма, тобто немає необхідності підтримувати послідовну та паралельну версії. Директиви OpenMP просто ігноруються послідовним компілятором, а для виклику процедур OpenMP можуть бути підставлені заглушки (stubs), текст яких приведений в специфікаціях.
- Однією з переваг OpenMP розробники вважають підтримку так званих «orphan» (відірваних) директив, тобто директиви синхронізації і розподілу роботи можуть не входити безпосередньо в лексичний контекст паралельної області.

## І ПОСТАНОВКА ЗАДАЧІ

- Завдання 1

Розпаралелити процес обчислення визначеного інтегралу, використовуючи редукцію.

- Завдання 2

Обчислити значення визначеного інтеграла відповідно до варіанту. Реалізацію програми виконувати таким чином:

1. Створити клас “Функція” (з єдиним методом “обчислити”) для реалізації підінтегральної функції.
2. Створити клас “Обчислювач інтегралів”, який може працювати у багатопотоковому режимі і має метод “обчислити” з параметрами: a, b - кінці інтервалу, n - кількість кроків та f - підінтегральна функція.
3. Для цих класів розробити модульні тести і виконати тестування.
4. Створити віконну програму, яка буде дозволяти вводити кількість інтервалів розбиття відрізка інтегрування і кількість потоків виконання.
5. Як результати роботи програми вивести обчислене значення інтегралу і час, який знадобився для її виконання.
6. Виконати обчислення декілька разів для різних (від 1 до 20 кількостей потоків виконання) при малій (менше  $10^3$ ) та великій (більше  $10^6$ ) кількості інтервалів розбиття відрізка.
7. Зробити висновки.

- Варіант 6. Інтеграл  $\int_0^1 \ln(t + 1) dt$ . Метод правих прямокутників.

## II РЕЗУЛЬТАТ РОБОТИ ПРОГРАМИ

```
Tests section:
Second task time work: 0.000110 Number of thread: 0
Second task time work: 0.130839 Number of thread: 0
Second task time work: 0.000006 Number of thread: 0

Threads: n=1
Integration interval:
a=0
b=1
Steps: s=10000000
First task time work: 0.077642
First task: 0.386294
Second task time work: 0.113512 Number of thread: 0
Second task result : 0.386294
```

```
Tests section:
Second task time work: 0.000071 Number of thread: 0
Second task time work: 0.146301 Number of thread: 0
Second task time work: 0.000007 Number of thread: 0

Threads: n=3
Integration interval:
a=0
b=1
Steps: s=100000
First task time work: 0.004284
First task: 0.386291
Second task time work: 0.002098 Number of thread: 0
Second task time work: 0.002098 Number of thread: 1
Second task time work: 0.002099 Number of thread: 2
Second task result : 0.386291
```

## ВИСНОВОК

Після виконання програми можемо зробити висновок, що чим більша кількість кроків, тим більша точність, а більша кількість потоків забезпечує вищу швидкість.