

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА
Факультет електроніки і комп'ютерних технологій

Звіт про виконання лабораторної роботи №3
З курсу “ Основи паралельних і розподілених обчислень”
« Синхронізація в OpenMP програмах»

Виконав:
Ст. групи Фес-32
Молібожко Олександр
Перевірів:
Кулик П.Р.

Хід роботи

Написати та запустити програму згідно індивідуального завдання з синхронізацією потоків на основі виразів явної синхронізації та технології замків. Завдання може бути реалізоване як дві програми, або як одна програма з вибором методу синхронізації.

Виконання завдання

1 – Завдання згідно варіанту

10. Написати програму, яка наближено обчислює значення ряду

$$\sum_{n=1}^{\infty} \frac{(-1)^n}{n!} = \frac{1-e}{e};$$

2 – Реалізація

```
1  #include <stdio.h>
2  #include <math.h>
3  #include <omp.h>
4
5  int main(int argc, char **argv) {
6      int N = 1000;
7      double sum = 0.0;
8      double term, factorial;
9      int i, method;
10
11     printf("Choose synchronisation method:\n1 - atomic operations\n2 - locks\n");
12     scanf("%d", &method);
13
14     if (method == 1) {
15         #pragma omp parallel for private(term, factorial) shared(sum)
16         for (i = 1; i <= N; i++) {
17             factorial = 1.0;
18             for (int j = 1; j <= i; j++) {
19                 factorial *= j;
20             }
21             term = pow(-1.0, i) / factorial;
22
23             #pragma omp atomic
24             sum += term;
25         }
26     } else if (method == 2) {
27         omp_lock_t lock;
28         omp_init_lock(&lock);
29
30         #pragma omp parallel for private(term, factorial) shared(sum)
31         for (i = 1; i <= N; i++) {
32             factorial = 1.0;
33             for (int j = 1; j <= i; j++) {
34                 factorial *= j;
35             }
36         }
37     }
```

```

        term = pow(-1.0, i) / factorial;

        omp_set_lock(&lock);
        sum += term;
        omp_unset_lock(&lock);
    }

    omp_destroy_lock(&lock);
} else {
    printf("Invalid choice.\n");
    return 1;
}

printf("The result of a row is: %.15f\n", sum);

double result = (1 - exp(1)) / exp(1);
printf("Theoretic result: %.15f\n", result);

return 0;
}

```

В цьому коді є 2 реалізації – за допомогою директиви `atomic` та замків. Користовачу необхідно вибрати спосіб обрахування та далі буде виведений результат.

```
Choose synchronisation method:  
1 - atomic operations  
2 - locks  
2  
The result of a row is: -0.632120558828558  
Theoretic result: -0.632120558828558  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

Висновок:

На даній роботі я ознайомився з методами синхронізації потоків в openmp та реалізував програму обрахування ряду двома різними способами.