

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

Звіт
Про виконання лабораторної роботи №1
“Основні конструкції OpenMP програм”

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

Львів 2024

Мета роботи: вивчити основні конструкції OpenMP програм

Хід роботи:

1. Ознайомитись з основними конструкціями OpenMP програм.
2. Написати та запустити OpenMP програму, котра створює n потоків кожен потік виводить на екран кількість процесорів, доступних для використання програмою, свій номер та загальну кількість потоків у програмі. I am thread from threads! Навести приклади використання модифікаторів private, shared, тощо.
3. Задати кількість потоків.
4. Створити в паралельній області декілька незалежних секцій коду.
5. Оформити звіт про виконання лабораторної роботи.

Виконання роботи:

1. Ознайомився з основними конструкціями OpenMP програм.
2. Код програми:

```
1 #include <stdio.h>
2 #include <omp.h>
3
4 int main(int argc, char *argv[]) {
5     int n = 4; // кількість потоків
6     int tid;
7
8     // Починаємо паралельну область
9     #pragma omp parallel num_threads(n) private(tid)
10    {
11        tid = omp_get_thread_num(); // Отримуємо номер потоку
12        int num_threads = omp_get_num_threads(); // загальна кількість потоків
13
14        // Виводимо інформацію про потік
15        printf("I am %d thread from %d threads! Available processors: %d\n", tid, num_threads, omp_get_num_procs());
16
17        // Розділяємо паралельну область на секції
18        #pragma omp sections
19        {
20            // Перша секція
21            #pragma omp section
22            {
23                printf("Hello from section 1, thread %d\n", tid);
24            }
25
26            // Друга секція
27            #pragma omp section
28            {
29                printf("Hello from section 2, thread %d\n", tid);
30            }
31
32            // Третя секція
33            #pragma omp section
34            {
35                printf("Hello from section 3, thread %d\n", tid);
36            }
37        }
38    }
39
40    return 0;
41 }
```

Виконання програми:

```
I am 2 thread from 4 threads! Available processors: 8  
Hello from section 1, thread 2  
Hello from section 2, thread 2  
Hello from section 3, thread 2  
I am 0 thread from 4 threads! Available processors: 8  
I am 1 thread from 4 threads! Available processors: 8  
I am 3 thread from 4 threads! Available processors: 8
```

Висновок: вивчив основні конструкції OpenMP програм.