

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

Звіт про виконання лабораторної роботи №2

з початкової дисципліни

«Паралельне програмування»

на тему:

«Паралельні цикли в OpenMP програмах»

Виконав:
студент групи ФЕП-22

Линва В. А.

Львів – 2021

Хід роботи

1. Ознайомився з директивою `#pragma omp for`, та її опціями.
2. Написав програму згідно індивідуального завдання (Завдання №9). За допомогою функції `omp_get_wtime()` заміряв час роботи програми за різної кількості потоків та розміру вхідних даних.
3. Використовуючи опцію `schedule` модифікував програму таким чином, щоб на екран виводилось повідомлення про те, який потік, яку ітерацію виконує.

[<Номер потоку>] : calculation of the iteration number <Номер ітерації>

Індивідуальне завдання – «Програма котра знаходить максимальний елемент матриці.»

Код програми:

```
#include <iostream>
#include <omp.h>
//Variant-9
using namespace std;
int main()
{
#pragma omp parallel private(i)
    double start_time, end_time;
    start_time = omp_get_wtime();
    int const n = 100;
    long double a[n][n];
    int Nmax{ }, Nind{ }, i, j = 0;
    int d;
#pragma omp parallel for private(i, j, Nmax) shedule(static)
    {
        for (int i = 0; i < n; i++) {
            #pragma omp parallel for shedule(static)
            {
```

```

        for (int j = 0; j < n; j++) {
            a[i][j] = rand() % 10;
            cout << "|  " << a[i][j] << " ";
        }
    }

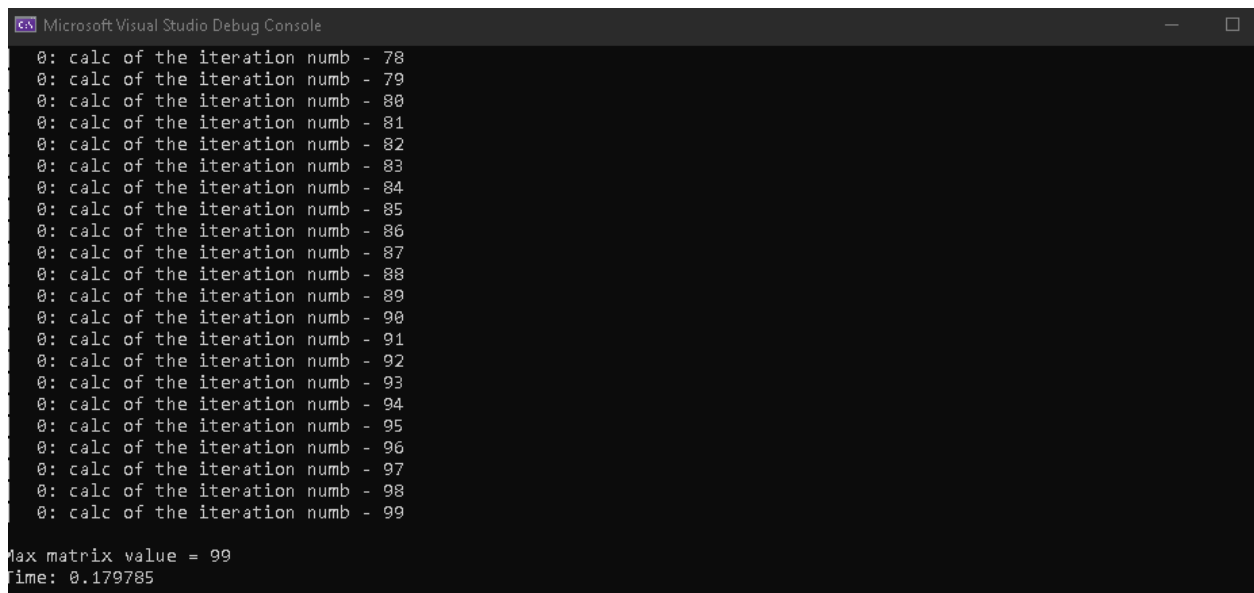
    if (Nmax < a[i][j]) {
        Nmax = a[i][j];
        Nind = i;
    };

    printf("| %d: calc of the iteration numb - %d", omp_get_thread_num(), i);
    cout << " " << endl;
}
}

cout << "\nMax matrix value = " << Nmax << endl;
end_time = omp_get_wtime();
cout << "Time: " << end_time - start_time << endl;
}

```

Результат виконання:



The screenshot shows the Microsoft Visual Studio Debug Console with the following output:

```

0: calc of the iteration numb - 78
0: calc of the iteration numb - 79
0: calc of the iteration numb - 80
0: calc of the iteration numb - 81
0: calc of the iteration numb - 82
0: calc of the iteration numb - 83
0: calc of the iteration numb - 84
0: calc of the iteration numb - 85
0: calc of the iteration numb - 86
0: calc of the iteration numb - 87
0: calc of the iteration numb - 88
0: calc of the iteration numb - 89
0: calc of the iteration numb - 90
0: calc of the iteration numb - 91
0: calc of the iteration numb - 92
0: calc of the iteration numb - 93
0: calc of the iteration numb - 94
0: calc of the iteration numb - 95
0: calc of the iteration numb - 96
0: calc of the iteration numb - 97
0: calc of the iteration numb - 98
0: calc of the iteration numb - 99

Max matrix value = 99
Time: 0.179785

```

Висновок: працюючи над цією лабораторною роботою, я ознайомився з директивою `#pragma omp for` та її опціями. Зміг застосувати її під час виконання реалізації індивідуального завдання. Ця директива дає змогу розпаралелити однотипну задачу із циклів, пришвидшує роботу програми на деякий час (в залежності від розміру вхідних даних для циклу) порівняно з послідовною програмою.