

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
**"Южно-Уральский государственный университет
(национальный исследовательский университет)"**
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

ОТЧЕТ
о выполнении практического задания № 3
по дисциплине
«Теория, методы и средства
параллельной обработки информации»

Выполнил:
студент группы КЭ-303
Старостенок Д.В.

Проверил:
доцент кафедры СП
Маковецкая Т.Ю.

Задание:

1. Напишите программу, в которой создается k нитей, и каждая нить выводит на экран свой номер и общее количество нитей в параллельной области в формате: I am thread from threads!

2. Модифицируйте программу таким образом, чтобы строку I am thread from threads! выводили только нити с четным номером.

Листинг программы:

```
#include <iostream>
#include <omp.h>

int main()
{
    int number_threads_2 = 5;

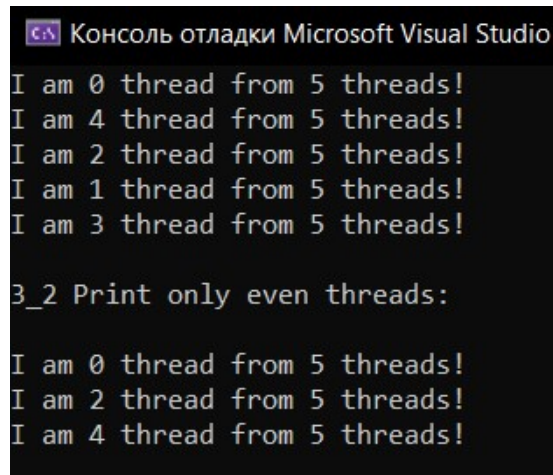
    int number_threads_2 = 5; // Quantity threads for lab 3
    // lab 3_1
    #pragma omp parallel num_threads(number_threads_2)
    {
        int id = omp_get_thread_num(); // get num of current
        thread
        int num_threads = omp_get_num_threads(); // get total
        number of threads in region
        printf("I am %d thread from %d threads!\n", id,
        num_threads);
        //std::cout << "I am " << id << " thread from " <<
        num_threads << " threads!\n";
    }

    printf("\n3_2 Print only even threads: \n\n");

    // lab 3_2
    #pragma omp parallel num_threads(number_threads_2)
    {
        int id = omp_get_thread_num(); // get num of current
        thread
        if (id % 2 == 0) {
            int num_threads = omp_get_num_threads(); // get total
            number of threads in region
            printf("I am %d thread from %d threads!\n", id,
            num_threads);
        }
    }

    return 0;
}
```

Результат выполнения программы и модификации этой программы (Рис. 1).



```
Консоль отладки Microsoft Visual Studio
I am 0 thread from 5 threads!
I am 4 thread from 5 threads!
I am 2 thread from 5 threads!
I am 1 thread from 5 threads!
I am 3 thread from 5 threads!

3_2 Print only even threads:
I am 0 thread from 5 threads!
I am 2 thread from 5 threads!
I am 4 thread from 5 threads!
```

Рис. 1 – Результат выполнения программы

Ответы на вопросы:

1) Как идентифицируются нити в OpenMP? Для чего это нужно?

Приведите содержательный пример.

Каждая нить (thread) идентифицируется своим номером в рамках параллельной области. Это нужно, чтобы можно было контролировать выполнение конкретных нитей и выполнять различные операции в зависимости от номера нити.

```
#include <iostream>
#include <omp.h>

int main() {
    #pragma omp parallel
    {
        int thread_num = omp_get_thread_num();
        std::cout << "Hello from thread " << thread_num << std::endl;
    }
    return 0;
}
```

2) Совпадают ли эти идентификаторы с идентификаторами потоков в ОС?

Идентификаторы нитей в OpenMP обычно не совпадают с идентификаторами потоков в операционной системе. Как правило, нити OpenMP создаются внутри процесса операционной системы и используют

ресурсы процесса, в том числе и идентификаторы потоков операционной системы.

3) Каков порядок вывода сообщений нитями? Всегда ли он одинаков? Чем определяется этот порядок?

Порядок вывода сообщений нитями в OpenMP может быть различным в разных запусках программы и зависит от многих факторов, включая процессор, настройки системы, наличие других запущенных процессов и т.д.

Он также может зависеть от порядка выполнения инструкций в коде программы. В общем случае порядок выполнения нитей не определен и не гарантирован.