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. |

Челябинск-2023

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).

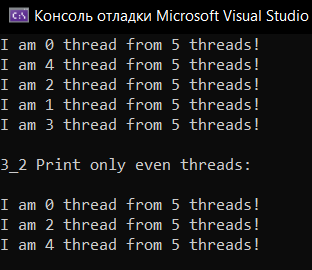


Рис. 1 – Результат выполнения программы

Ответы на вопросы:

1) Как идентифицируются нити в OpenMP? Для чего это нужно? Приведите содержательный пример.

В OpenMP каждая нить идентифицируется уникальным номером, который можно получить с помощью функции omp\_get\_thread\_num().

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 не обязательно совпадают с идентификаторами потоков в операционной системе (ОС). OpenMP является высокоуровневой библиотекой параллельного программирования, которая скрывает от программиста детали реализации многопоточности на уровне ОС.

OpenMP создает нити (threads) на уровне пользовательского пространства, которые могут выполняться на разных физических потоках (cores) процессора, в зависимости от настроек ОС и архитектуры процессора. При этом, идентификаторы нитей в OpenMP могут меняться во время выполнения программы, в зависимости от порядка создания и завершения нитей.

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.