Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего профессионального образования

**«Национальный исследовательский ядерный университет «МИФИ»**

**ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ**

**Кафедра №42 (криптологии и кибербезопасности)**

ЛАБОРАТОРНАЯ РАБОТА №3  
по дисциплине: «Параллельное программирование»  
на тему: «Реализация алгоритма с использованием технологии OpenMP»

**Выполнил:**студент группы Б19-515  
Родионов Дмитрий Александрович

**Москва, 2021**

Оглавление

[Используемая рабочая среда 3](#_Toc85575385)

[Анализ приведённого алгоритма 4](#_Toc85575386)

[Заключение 5](#_Toc85575387)

# Используемая рабочая среда

**Процессор:** Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.59 GHz

**Оперативная память:** 16.0 ГБ

**Тип системы:** 64-разрядная операционная система, процессор x64

**Версия:** 21H1

**Среда разработки:** WSL2 Ubuntu 20.04 LTS

**Поддерживаемая версия OpenMP:** 4.5

# Анализ приведённого алгоритма

**Структура алгоритма сортировки Шелла:**

1) последовательный участок кода – в данном блоке программы происходит инициализация цикла, изменяющего значение сдвига *gap* на каждой своей итерации и выполняющего разбиение исходного массива на списки элементов, отстающих друг от друга на величину сдвига *gap*.

2) параллельный участок кода – в данном блоке программы происходит распараллеливание цикла, выполняющего распределение элементов каждого списка с помощью сортировки вставками и объединяющего целевые списки обратно в массив, т. е. в каждом используемом потоке осуществляется определённое число сравнений элемента, выбранного из неотсортированной части массива, с элементами, находящимися в отсортированной части массива, а также вставка обработанного элемента в отсортированную часть массива; множество итераций цикла в рассматриваемом алгоритме разделяется на куски примерно одинакового размера, которые распределяются между используемыми нитями.

**Назначение используемых директив OpenMP и их опций:**

1) *parallel* – задаёт параллельную область программы (строки №19-21):

1. *for* – распределение итераций цикла между различными нитями: относится к идущему следом блоку, включающему оператор *for*;
2. *num\_threads(threads)* – явное задание количества нитей, которые будут выполнять параллельную область;
3. *shared(array, count, gap)* – задаёт список переменных, общих для всех нитей;
4. *default(none)* – всем переменным в параллельной области класс должен быть назначен явно.

В случае отсутствия данной директивы в приложенном коде алгоритма, всё множество итераций цикла выполнялось бы единственным потоком.

# Заключение

В данной лабораторной работе проведена статическая и динамическая отладка параллельного алгоритма, реализованного с использованием технологии параллельных вычислений OpenMP. Для достижения поставленной цели раскрыты понятия временной сложности, ускорения и эффективности программы; применены базовые способы математического описания алгоритма. Также в ходе выполнения работы были косвенно определены характеристики используемого процессора.

Цель, поставленная в начале работы, достигнута; задачи выполнены.