Министерство науки и образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных технологий

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**ОТЧЕТ**

по лабораторной работе № 1

по дисциплине «Параллельное программирование»

**OpenMP**

ОГУ 09.03.04.4024.704 ПЗ

Руководитель

канд. техн. наук, доцент

\_\_\_\_\_\_\_\_\_\_\_\_\_ Н. А. Тишина

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

Исполнитель

Студент группы 22ПИнж(б)РПиС-1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.В. Федоров

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

Оренбург 2024

# Содержание

[Содержание 2](#_Toc179140167)

[Практическая часть 4](#_Toc179140168)

[Список литературы 2](#_Toc179140169)

Задание

**Цель работы**

ознакомление с технологией разработки параллельных программ для систем с общей памятью OpenMP.

**Задание**

* научиться компилировать и запускать OpenMP–программы;
* научиться использовать основные директивы OpenMP.

**Порядок выполнения работы**

1. Изучить принцип работы OpenMP.
2. Разработать программы, скомпилировать, настроить.
3. Подготовить отчет в твердой копии и в электронном виде

# Практическая часть

*Задание 1:* Знакомство с OpenMP

В данном задании необходимо собрать и запустить параллельную программу, написанную при помощи OpenMP



Рисунок 1. Первая программа

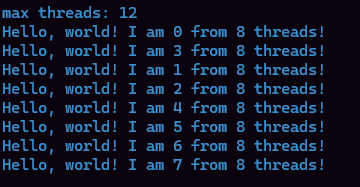


Рисунок 2. Результат

*Задание 2*: Видимость переменных

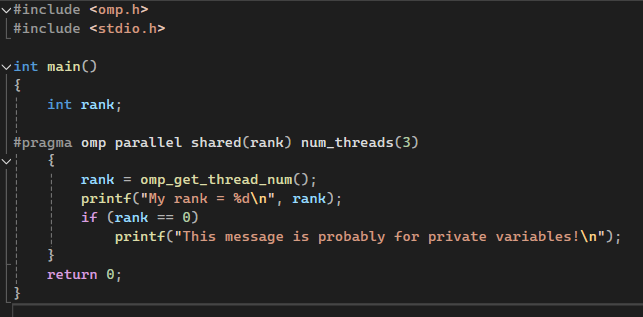


Рисунок 3. Вторая программа с shared

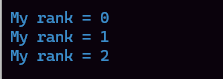


Рисунок 4. Результат программы с shared

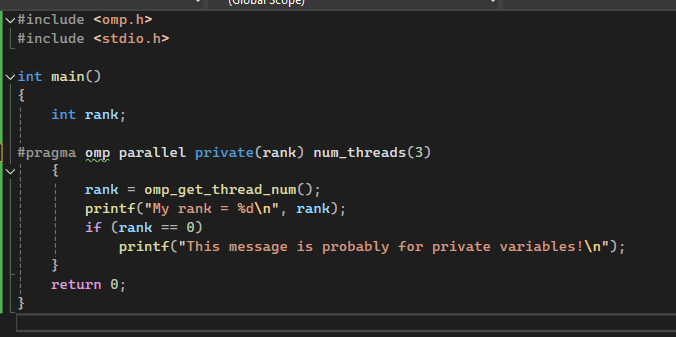


Рисунок 3. Вторая программа с private

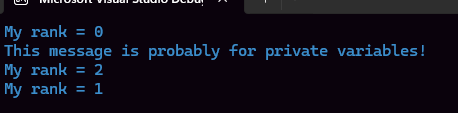


Рисунок 5. Результат программы с private

*Задание 3*: Распараллеливание цикла for

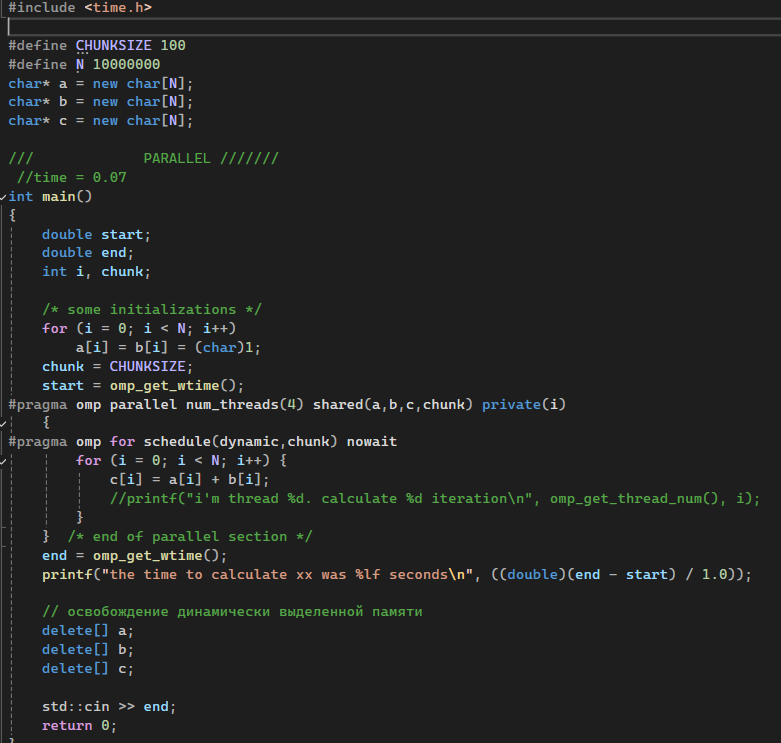


Рисунок 6. forVector параллельное выполнение

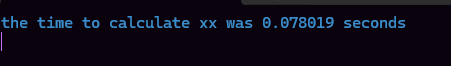


Рисунок 7. forVector параллельное выполнение результат

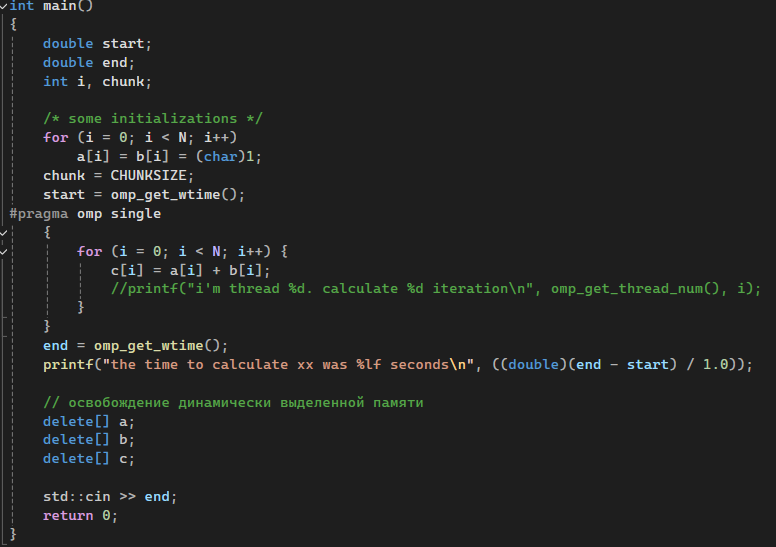


Рисунок 8. forVector последовательное выполнение

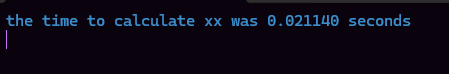


Рисунок 9. forVector последовательное выполнение результат

Если мы увеличим размер чанка с 3 до, например, 100, то сможем добиться значительного ускорения исполнения программы с параллельной реализацией работы.

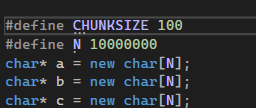


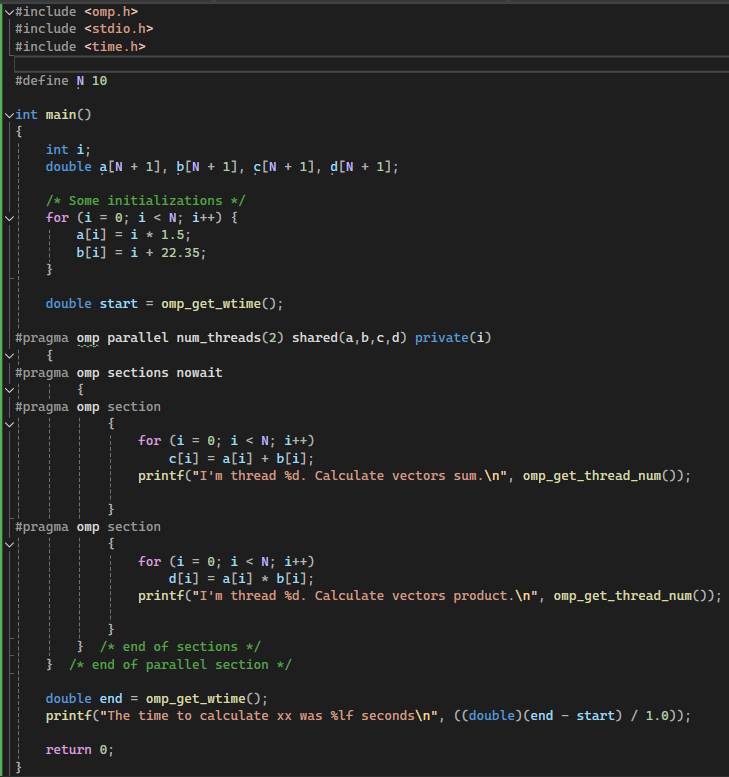
Рисунок 10. Увеличили размер чанка



Рисунок 11. forVector результат с параллельным способом

Как видим значительно выросла скорость компиляции программы в параллельном варианте. На последовательный никак не повлияло. В разработке программ на параллельной реализации необходимо правильно выбирать размер чанка и количество потоков, иначе программа не только будет не быстрее выполняться, а то и медленнее.

*Задание 4:* Распределение заданий с помощью секций

**Рисунок 12. sectionsVector параллельное выполнение**

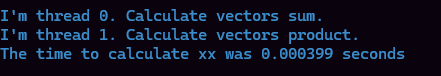


Рисунок 13. sectionsVector параллельное выполнение результат

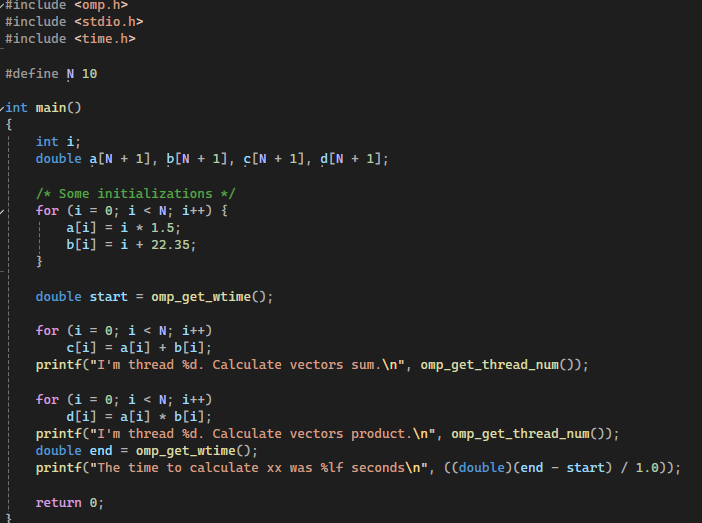


Рисунок 14. sectionsVector последовательное выполнение

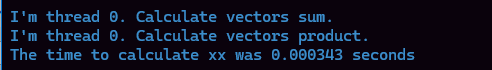


Рисунок 15. sectionsVector последовательное выполнение результат

# Список литературы

1. Антонов, А. С. Технологии параллельного программирования MPI и OpenMP [Текст] : учебное пособие для студентов высших учебных заведений, обучающихся по направлениям ВПО 010400 "Прикладная математика и информатика" и 010300 "Фундаментальная информатика и информационные технологии" / А. С. Антонов; Моск. гос. ун-т им. М. В. Ломоносова. - Москва : Изд-во Моск. ун-та, 2012. - 340 с. : ил. - (Суперкомпьютерное образование). - Библиогр.: с. 333-334. - ISBN 978-5-211-06343-3 (ен тл-26; кх-4).
2. Биллиг, В. А. Параллельные вычисления и многопоточное программирование [электронный ресурс] / В.А.Билиг Национальный Открытый Университет «ИНТУИТ», 2016. - 311 c. – Режим доступа <http://biblioclub.ru/index.php?page=book_view&book_id=428948>