

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной

инженерии

**КОНСОЛЬНОЕ ПРИЛОЖЕНИЕ ВЫЧИСЛЯЮЩЕЕ ПРЯМОЕ
ПРОИЗВЕДЕНИЕ МНОЖЕСТВ**

Вариант № 6

Пояснительная записка

Исполнитель

Студент группы БПИ199

_____/Галанов А. С./

«__»_____2020 г.

Москва 2020

Оглавление

Текст задние	3
Краткое описание программы	4
— Модель построения программы	4
— Этапы программы	4
— Описание работы программы	4
— Описание формата входных данных	5
○ Формат входных данных для консоли	5
○ Формат входных данных в файле с данными	5
Тестовые примеры	7
Список использованной литературы.....	9

Текст задние

Вычислить прямое произведение множеств A_1, A_2, A_3, A_4 . Входные данные: множества чисел A_1, A_2, A_3, A_4 , мощности множеств могут быть не равны между собой и мощность каждого множества больше или равна 1. Количество потоков является входным параметром.

Краткое описание программы

— Модель построения программы

В приложении использовалась следующая модель многопоточных вычислений: ”Управляющий и рабочие”[1], посредством использования OpenMP. То есть в программе управляющий поток координирует действия вспомогательных потоков, выдавая им задачи, собирая информацию о выполнении поставленных задач и прочее.

— Этапы программы

Программа делится на следующие этапы работы:

1. Запуск программы с определенными параметрами для работы
2. Проверка корректности введенных данных.
3. Установка количества потоков для решения задач.
4. Создание задач для будущих потоков.
5. Распределение задач между потоками.
6. Создание файла с результатом.

— Описание работы программы

Пользователь при запуске программы передает в нее 3 параметра: “Количество вспомогательных потоков”; ”Путь к файлу с данными”; “Путь к файлу, куда запишется результат”.

Если число “Количество потоков” больше 6 или меньше 1, то программа сообщает об ошибке пользователю. Количество потоков должно лежать в диапазоне от 1 до “Количество задач”. Одна задача — это прямое произведение двух случайных множеств. Так как на входе множеств 4, то задач будет равно $3!=6$ (Количество перестановок).

После определения количества потоков и задач, устанавливается количество дополнительных потоков командой `omp_set_num_threads()[3]`, которые в будущем будут параллельно выполнять задачи.

Управляющий поток в начале создает массив данных, который является будущими задачи для вспомогательных потоков, после чего доходит до секции

`#pragma omp parallel[3]`, в которой задачи раздаются между потоками и выполняются параллельно. Так как в данной секции присутствует цикл, потокам надо об этом сообщить путем `#pragma omp for[3]`.

После распределения всех задач, и окончания секции `#pragma omp parallel[3]` куда основной поток придет после завершения работы дополнительных потоков в данной секции, основной поток запишет в файл результат выполнения программы и программа завершится.



Рис.1 – Модель “Управляющий и рабочие”

— Описание формата входных данных

○ Формат входных данных для консоли

В качестве входных данных пользователь в командной строке должен написать “Количество вспомогательных потоков”, “Полный путь к файлу с данными”, “Полный путь к файлу, куда запишется результат” через пробел. Пример: “3 C:\MyFiles\test1.txt C:\MyFiles\answer1.txt”

○ Формат входных данных в файле с данными

В файле формат данных организован по следующему принципу:

1. С каждой новой строки записывается множество
2. Перед элементами самого множества записывается мощность множества
3. Все числа записываются через пробел

Пример для множеств $A_1\{1,2,3,4\}$, $A_2\{2,4\}$, $A_3\{123,4344,3\}$, $A_4\{123\}$ приведен на рисунке 2.

```
4 1 2 3 4
2 2 4
3 123 4344 3
1 123
```

Рис.2 – пример входного файла

Тестовые примеры

1. Ввод некорректного числа потоков.

```
Командная строка имела не правильный формат. Количество потоков может быть от 1 до 6!  
Пример командной строки:{countTheards} {pathToInputFile} {pathToOutputFile}  
Для продолжения нажмите любую клавишу . . .
```

2. Пример корректной работы для 1 потока.



test1.txt



answer1.txt

Входной файл

Выходной файл

3. Пример корректной работы для 2 потоков.



test2.txt



answer2.txt

Входной файл

Выходной файл

4. Пример корректной работы для 3 потоков.



test3.txt



answer3.txt

Входной файл

Выходной файл

5. Пример корректной работы для 4 потоков.



test4.txt



answer4.txt

Входной файл

Выходной файл

6. Пример корректной работы для 5 потоков.



test5.txt



answer5.txt

Входной файл

Выходной файл

7. Пример корректной работы для 6 потоков.



test6.txt



answer6.txt

Входной файл

Выходной файл

Пояснительная записка написана на основании инструкции [2].

Список использованной литературы

1. Статья про модели многопоточных вычислений [Электронный ресурс]
//URL: <https://studfile.net/preview/4419687/page:3/> (Дата обращения: 28.11.2020, режим доступа: свободный)
2. Инструкция по составлению пояснительной записки [Электронный ресурс].
//URL: <http://www.softcraft.ru/edu/comparch/tasks/t04/> (Дата обращения: 28.11.2020, режим доступа: свободный)
3. Документация по OpenMP [Электронный ресурс]. //URL:
<https://docs.microsoft.com/ru-ru/cpp/parallel/openmp/reference/openmp-library-reference?view=msvc-160> (Дата обращения: 28.11.2020, режим доступа: свободный)