**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](#_Toc55748255)

[Краткое описание программы 4](#_Toc55748256)

[ Модель построения программы 4](#_Toc55748257)

[ Этапы программы 4](#_Toc55748258)

[ Описание работы программы 4](#_Toc55748259)

[ Описание формата входных данных 5](#_Toc55748260)

[o Формат входных данных для консоли 5](#_Toc55748261)

[o Формат входных данных в файле с данными 5](#_Toc55748262)

[Тестовые примеры 7](#_Toc55748263)

[Список использованной литературы 9](#_Toc55748264)

Текст задние

Вычислить прямое произведение множеств А1, А2, А3, А4. Входные данные: множества чисел А1, А2, А3, А4, 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]. То есть в программе управляющий поток координирует действия вспомогательных потоков, выдавая им задачи, собирая информацию о выполнении поставленных задач и прочее.

* Этапы программы

Программа делиться на следующие этапы работы:

1. Запуск программы с определенными параметрами для работы
2. Проверка корректности введенных данных.
3. Распределение задач между потоками.
4. Создание файла с результатом, по указанному пользователем пути.

* 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, управляющий поток начинает выдавать задачи потокам(Thread)[3], пока они не закончатся. Если задач больше, чем вспомогательных потоков, то управляющий поток проверяет контейнер с индексами свободных потоков. Если там нет ни одного индекса, то он начинает ждать первый освободившийся поток посредством функции wait() у condition\_variable, и выдает новую задачу потоку. Вспомогательные потоки в свою очередь, как только освобождаются кладут свой индекс в контейнер и сигнализируют переменную условия, о том, что они отработали.

Каждая задача имеет в себе критическую секцию, поэтому в коде используется класс mutex, для блокирования всех потоков, кроме действующего. В критическую секцию входят: Создание результирующей строки; добавление индекса потока в контейнер освободившихся потоков; сигнализирование переменной условия.

После распределения всех задач, основной поток ожидает выполнения всех вспомогательных потоков, после чего записывает в файл результат выполнения программы и программа завершается.

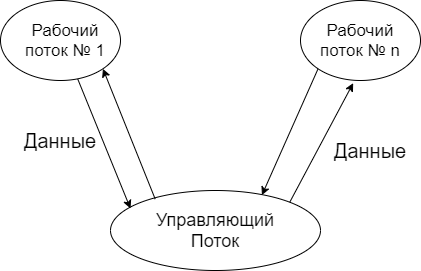


Рис.1 – Модель “Управляющий и рабочие”

* 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 A1{1,2,3,4}, A2{2,4}, A3{123,4344,3}, A4{123}приведен на рисунке 2.

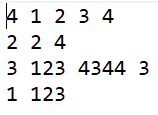
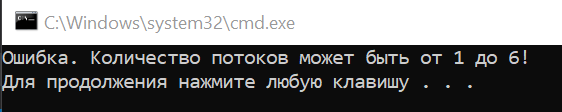


Рис.2 – пример входного файла

Тестовые примеры

1. Ввод некорректного числа потоков.



1. Пример корректной работы для 1 потока.

Входной файл Выходной файл

1. Пример корректной работы для 2 потоков.

Входной файл Выходной файл

1. Пример корректной работы для 3 потоков.

Входной файл Выходной файл

1. Пример корректной работы для 4 потоков.

Входной файл Выходной файл

1. Пример корректной работы для 5 потоков.

Входной файл Выходной файл

1. Пример корректной работы для 6 потоков.

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: 08.11.2020, режим доступа: свободный)
2. Инструкция по составлению пояснительной записки [Электронный ресурс]. //URL:<http://softcraft.ru/edu/comparch/tasks/mp01/>(Дата обращения: 08.11.2020, режим доступа: свободный)
3. Документация по thread [Электронный ресурс]. //URL: https://docs.microsoft.com/ru-ru/cpp/standard-library/thread-class?view=msvc-160&viewFallbackFrom=vs-2019 (Дата обращения: 08.11.2020, режим доступа: свободный)