Практические приемы построения многопоточных приложений

Отчет

Выполнила:
Гурова Екатерина
студентка группы БПИ198

1. Описание задачи

1.1. Текст задания

Вариант 9. Определить, является ли множество C объединением множеств A и B (A U B), пересечением множеств (A ∩ B), разностью множеств A и B (A\B), разностью множеств B и A (B \ A). Входные данные: множества целых положительных чисел A, B, C. Оптимальное количество потоков выбрать самостоятельно.

1.2. Порядок выполнения

1. Выбрать модель приложения, наиболее точно отвечающую специфике задачи или применить указанную. Изучить используемую модель по дополнительным источники информации.
2. Разработать алгоритм решения задания, с учетом разделения вычислений между несколькими потоками. Избегать ситуаций неуправляемого изменения одних и тех же общих данных несколькими потоками. Если же избежать этого невозможно, необходимо использовать мьютексы и критические секции.
3. Разработать программу с применением функций библиотеки POSIX Threads или стандартной библиотеки C++ и протестировать ее на нескольких примерах.

2. Используемая модель вычислений

Моделью построения программы является итеративный параллелизм [3][4]. Для проверки каждой из операций отводится отдельный поток, содержащий циклы. Потоки программы описываются итеративными функциями и работают совместно над решением одной задачи.

В основе алгоритмы программы лежат определения операций над множествами [1].
Объединение:

$$A \cup B = \{x | (x \in A) \vee (x \in B)\}$$

Пересечение:

$$A \cap B = \{x | (x \in A) \wedge (x \in B)\}$$

Разность:

$$A \setminus B = \{x | (x \in A) \wedge (x \notin B)\}$$

Для проверки, является ли множество C результатом применения этих операций ко множествам A и B, строятся соответствующие множетсва согласно их определениям.
Таким образом, для пересечения итеративно перебираются все элементы множества A, и в результирующее множество попадают только те, которые содержатся в B.
Для объединения в результирующее множество попадают все элементы множества A, а затем только те элементы множества B, которые не содержатся в пересечении множеств A и B.
Для разности итеративно перебираются элементы множества A, затем проверяется, не содержится ли данный эелемент во множестве B. Если да, то элемент добавляется в результат.
Затем для проверки совпадения результирующего множества и множества C применяется определение биекции [2]. Между результирующим множеством и множеством C строится биекция, сопоставляющая совпадающие элементы множества. Если их мощности не равны, возвращается отрицательный результат. Если равны, то далее итерациями по эелментам результирующего множества проверяется, содержатся ли все его элементы во множестве C. Если найден один не совпадающий элемент, возвращается отрицательный результат. Во всех остальных случаях рузльтат положительный.

3. Требования к входным и вызодным данным

На вход программе принимаются три текстовых файла, содержащие целые положительные числа в диапазоне от 1 до 4 294 967 294. В каждом текстовом файле должны содержаться элементы множеств A,B,C на отдельных строках. Файлы находиться в той же директории, что и исполняемый файл программы. Результат записывается в текстовый файл, находящийся в той же директории, что и исполняемый файл программы.

Имена всех четырех файлов указываются в командной строке в формате:

<name_of_A> <name_of_B> <name_of_C> <name_of_result>

Файла с входными и выходными данными не должны использоваться во время выполнения программы.

ПРИЛОЖЕНИЕ 1

Источники информации

1. Множества. Операции над множествами. [Электронный ресурс] // URL: http://mathportal.net/index.php/matematicheskaya-logika-i-teoriya-mnozhestv/mnozhestva-operatsii-nad-mnozhestvami (дата обращения: 15.11.2020)
2. Биекция. [Электронный ресурс] // URL: https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B5%D0%BA%D1%86%D0%B8%D1%8F (дата обращения: 15.11.2020)
3. Средства разработки параллельных программ. // URL: https://studfile.net/preview/4419687/page:3/ (дата обращения: 15.11.2020)
4. Итеративный параллелизм: умножение матриц. // URL: http://www.soft.architecturenet.ru/70/index-iterativnyj-parallelizm-umnozhenie-matric.htm (дата обращения: 15.11.2020)