Сахабутдинов Эрнест БПИ202

Домашнее задание 5

Описание задания

Вариант 20. Задача о программистах:

В отделе работают три программиста. Каждый программист пишет свою программу и отдает ее на проверку другому программисту. Программист проверяет чужую программу, когда его собственная уже написана. По завершении проверки, программист дает ответ: программа написана правильно или написана неправильно. Программист спит, если не пишет свою программу и не проверяет чужую программу. Программист просыпается, когда получает заключение от другого программиста. Если программа признана правильной, программист пишет другую программу, если программа признана неправильной, программист исправляет ее и отправляет на проверку тому же программисту, который ее проверял. Создать многопоточное приложение, моделирующее работу программистов.

Цель задания:

Разработать многопоточное консольное приложение на языке C/C++ с использованием **библиотеки POSIX Threads** языка программирования C или **стандартной библиотеки** языка программирования C++, стиль написания произвольный. При реализации программы использовалась модель вычисления “*взаимодействующие равные*”.

Также нужно:

- Описать вид входных данных программы и взаимодействия с программой
- подробно описать используемую модель вычислений и привести источники информации, в которых описана данная модель
- зафиксировать количество заголовочных и программных файлов, общий размер исходных текстов, полученный размер исполняемого кода

Описание входных данных и взаимодействия с программой

Данные вводятся в виде аргументов командной строки при запуске исполняемого файла. Существует два режима работы программы: генерация случайных данных при заданном размере и чтение входных данных из файла. Формат командной строки выглядит следующим образом:

-f infile

Здесь **-f** это флаг, который означает, что данные будут считываться из файла, а **infile** – это непосредственно сам путь до файла. Например, **-f input.txt** означает, что чтение будет производиться из файла input.txt.

Формат входного файла следующий: на каждой строке вводится два целых числа, первое число должно быть в отрезке от 1 до 4, а второе – от 1 до 3. Первое число обозначает тег программы, которую выполняет программист, и по нему определяется, правильно ли написана программа. Второе число обозначает номер программиста, который обязан написать эту программу.

-n size

Здесь **-n** это флаг, который означает, что данные будут генерироваться случайным образом, а **size** означает количество создаваемых программ. Например, **-n 50** означает, что будет сгенерировано 50 случайных элементов данных.

Ограничения установлены следующие: максимальное число вводимых программ – 100. Столь малые ограничения обусловлены тем, что в меру большей реалистичности программист пишет и проверяет программу не мгновенно, а за случайный промежуток времени от 1 до 5 секунд, а также из-за того, что вероятность программиста написать правильную программу составляет только 25%. Также, чтобы не возникало ситуаций, когда один программист должен проверить несколько программ одновременно (такая ситуация указана в условии задачи), программы первого программиста всегда проверяются вторым программистом, программы второго – третьим, а программы третьего – первым.

Описание используемой модели вычислений

Определение модели “*взаимодействующие равные*”, взятое из лекции по Архитектуре Вычислительных Систем:

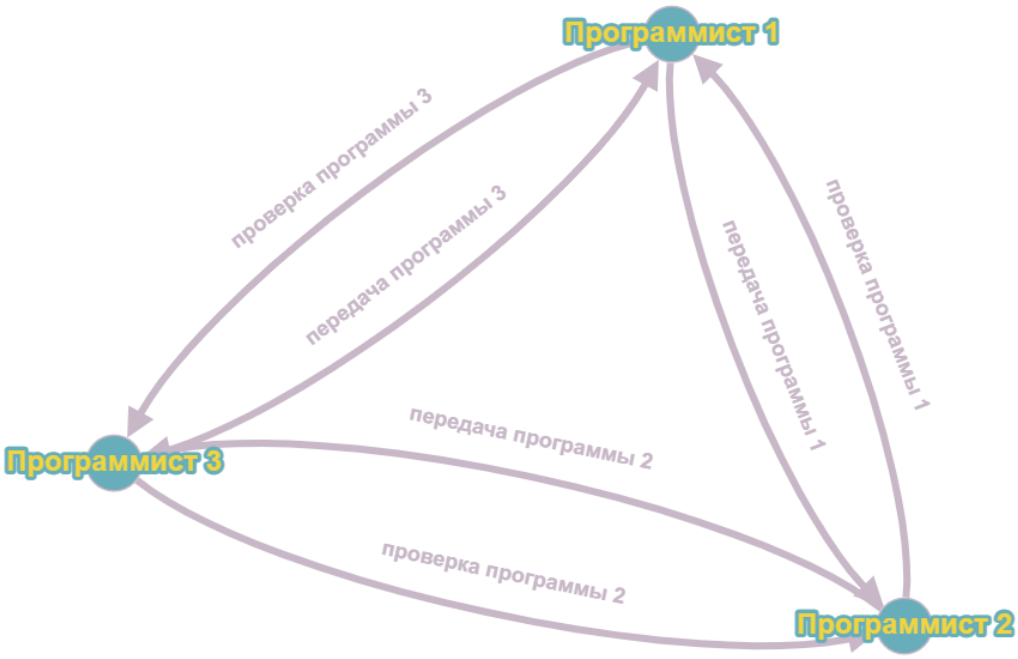
Взаимодействующие равные – модель, в которой исключен не занимающийся непосредственными вычислениями управляющий поток. Распределение работ в таком приложении либо фиксировано заранее, либо динамически определяется во время выполнения.

Эта модель вычислений используется для реализации распределенных параллельных программ, зачастую несколько процессов для решения задачи выполняют один и тот же код и обмениваются сообщениями. Каждый рабочий процесс выполняет один и тот же алгоритм и взаимодействует с другими рабочими, чтобы вычислить свою часть необходимого результата.

(Источник: Основы многопоточного, параллельного и распределенного программирования, Грегори Р. Эндрюс, глава 1)

Действительно, реализация данного задания использует данную модель. Каждый из потоков (каждый из программистов) передает другому программисту результаты написанной работы (в данном случае – тег написанной программы и статус, что программу можно проверить). Другой программист, после того как он отправил свою программу третьему программисту, он проверяет программу с необходимым тегом и отправляет первому программисту информацию относительно корректности программы, и первый программист исходя из переданной информации снова начинает работать над этой программой, либо переходит к новой. Выходит, что модель для данной задачи можно представить в кольцевом виде, которая идет в обоих направлениях: в одну сторону – передача выполненных программ, в другую - передача проверенных программ.

Значит, работу программы можно представить в виде данной схемы:



Все потоки используют мьютексы и примитивы синхронизации для правильной последовательности работы потоков.

ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ПРОГРАММЫ:

- Число заголовочных файлов – 2
- Число модулей реализации – 2
- Размер исполняемого файла – 69,3 кб.
- Общий размер исходных текстов – 202 строки кода, их вес – 7,42 кб.