

Нестеренко Сергей

БПИ201

Отчет

АВС ДЗ 5

Многопоточность

Вариант 23

16.12.2021

## 1 Постановка задачи

Задача о наследстве. У старого дона Энрике было два сына, у каждого из сыновей – еще по два сына, каждый из которых имел еще по два сына. Умирая, дон Энрике завещал все свое богатство правнукам в разных долях. Адвокат дон Хосе выполнил задачу дележа наследства в меру своих способностей. Правнуки заподозрили адвоката в укрывательстве части наследства. Требуется создать многопоточное приложение, которое при известных сумме завещания дона Энрике и доле каждого наследника, проверяет честность адвоката. При решении использовать принцип дихотомии.

**Дедлайн:** 15.12.2021 23:59

**Цели и задачи:** изучить работу с потоками. Научиться разбивать задачу на части, для последующего их выполнения различными потоками.

1. В соответствии с **новым** выданным вариантом задания разработать **консольное** приложение с использованием библиотеки **POSIX Threads** языка программирования **C** или **стандартной библиотеки** языка программирования **C++**. Программа должна быть написана на языке C или C++, компилироваться и выполняться в ОС Linux. Стиль написания программы – свободный.
2. Выбрать модель приложения, наиболее точно отвечающую специфике задачи или применить **указанную**. Изучить используемую модель по дополнительным источникам информации и привести в пояснительной записке ее описание.
3. 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.
4. Провести отладку и тестирование разработанной программы. Программа должна правильно обрабатывать входные данные в соответствии с условием задания и реагировать на некорректно вводимые исходные данные. Ввод основных данных должен осуществляться в допустимом для условия задачи диапазоне без введения искусственных ограничений.
5. Отчет должен содержать титульный лист с идентификацией студента, номер варианта, описание задачи.
6. В отчете необходимо подробно описать используемую модель вычислений. Привести источники информации, в которых описана данная модель.
7. Для удобства ввод-вывод данных можно организовать с использованием файлов и (или) использовать аргументы командной строки. При этом способы ввода, должны быть тщательно документированы и понятны пользователю программы. Описание работы с данными программы должно быть приведено в отчете и сопровождаться соответствующей информацией при работе с программой.
8. Для порождения массивов данных допускается использовать генераторы случайных чисел. При этом генераторами случайных чисел должны порождаться семантически корректные данные.
9. В ходе выполнения программа должна выводить в консоль все события, доступно (в терминах предметной области) объясняющие последовательность моделируемых действия. При необходимости нужно выводить модельное время, формируемое на основе принятых единиц его измерения.
10. При решении задачи обязательно использовать синхропримитивы из указанных в условии (если они указаны).

*Примечание. Хотелось бы обратить внимание на достаточно неформальную постановку задач, что позволяет каждому проявить фантазию и инициативу в его интерпретации и реализации. В связи с этим любое копирование описания, исходных данных, алгоритмов и программного кода будет рассматриваться как плагиат. При этом, в рамках условия задачи, допускаются собственные интерпретации тех или иных событий. При описании этих событий возможно использование как русского, так и английского языка, так как практически во всех задачах нет обработки текстов, а текстовая информация используется только для вывода.*

## 2 Принцип дихотомии

## 3 Реализация

Условие можно упростить до:

Есть 8 чисел, проверить что их сумма равна некому числу  $N$ .

Которое можно обобщить до:

Есть последовательность из  $n$  натуральных чисел, проверить что их сумма равна некому числу  $S$ .

Используем принцип дихотомии.

Будем разделять массив на отрезки и распределять их по потокам, в которых будем опять разделять отрезок на более мелкие части или считать сумму элементов отрезка.

Будем думать, что создание двух потоков для подсчета в каждом из них суммы двух элементов более выгодное решение чем просто подсчет суммы 4 элементов (очевидно, что это не так, но так велит задание). Следовательно будем делить отрезки до того момента пока они не станут длины меньше или равной двум.

Будем использовать язык C++ с использованием потоков из `pthread.h`