

ОТЧЁТ

Работу выполнила студент БПИ 198 Скрыпина Дарья Кирилловна

Номер варианта: 26

Описание задачи: *«Вторая задача об Острове Сокровищ. Шайка пиратов под предводительством Джона Сильвера высадилась на берег Острова Сокровищ. Несмотря на добытую карту старого Флинта, местоположение сокровищ по-прежнему остается загадкой, поэтому искать клад приходится практически на ощупь. Так как Сильвер ходит на деревянной ноге, то самому бродить по джунглям ему не с руки. Джон Сильвер поделил остров на участки, а пиратов на небольшие группы. Каждой группе поручается искать клад на нескольких участках, а сам Сильвер ждет на берегу. Группа пиратов, обшарив одну часть острова, переходит к другой, еще необследованной части. Закончив поиски, пираты возвращаются к Сильверу и докладывают о результатах. Требуется создать многопоточное приложение с управляющим потоком, моделирующее действия Сильвера и пиратов. При решении использовать парадигму портфеля задач.»*

1) Описание используемой модели вычислений

Моделью, указанной в условии задачи, является парадигма портфеля задач. Каждый поток обращается к переменной-счётчику (своеобразному «портфелю задач»), которая отражает количество уже решённых задач (в условиях данного задания – количество уже исследованных частей острова). Это разделяемая переменная, доступ к которой одновременно имеет только один поток. Он выясняет номер текущей задачи, и если в «чемодане» остались свободные задачи, то поток «забирает» себе данную задачу и увеличивает счётчик на 1, после чего выполняет задачу (исследует данную часть острова на предмет сокровищ). Доступ к портфелю задач является потокобезопасным благодаря использованию мьютексов. Выполнение задач прекращается, когда количество свободных задач в «портфеле» становится равным 0.

Выполнение алгоритма начинается с получения от пользователя двух положительных целых чисел – количества команд пиратов (количество потоков) и количества участков на острове, которые необходимо исследовать. Осуществляется проверка пользовательского ввода на корректность. Вводимые значения должны иметь возможность быть преобразованными в тип `int`, что проверяется с помощью функции `std::stoi()`. Если введённое число нельзя преобразовать в `int` (число введено в неверном формате или превышено максимальное значение типа `int`) или введённое число ≤ 0 , то программа выводит сообщение об ошибке и завершает свою работу. То же самое происходит, если пользователь вводит в качестве количества потоков число, большее или равное 1000 (в целях безопасности и экономии памяти установлено верхнее ограничение на количество потоков).

Запускается необходимое число потоков, равное 1-ому введённому пользователем аргументу (кол-во команд пиратов). Каждый поток инициализирован функцией `LookForTreasure(int team_number)`, где `team_number` – номер потока (команды пиратов), нумеруемый с 1. Каждый поток пытается закрыть мьютекс `island`, ответственный за синхронизацию потоков при обращении к счётчику задач и его изменении. Поток, который «успел» закрыть мьютекс (остальные потоки ждут, когда мьютекс снова окажется открытым), обращается к переменной-счётчику `locations_explored`, отвечающей за количество выполненных задач, и сравнивает его с общим количеством задач (2-ой введённый пользователем аргумент). Если счётчик \geq общему количеству задач, то поток прекращает свою работу, иначе увеличивает счётчик выполненных задач на 1 и приступает к выполнению задачи.

При помощи генератора случайных чисел определяется время (от 0 до 4 секунд), которое поток будет спать (в условиях данной задачи – за это

время пираты обшаривают часть острова). После этого при помощи того же генератора вычисляется факт того, нашли ли пираты сокровище или нет (True, если случайное число от 0 до 25 оказалось ≥ 15 , и False в противном случае). Поток закрывает мьютекс callout на вывод текстовой информации в консоль, оповещает о результате своей деятельности (выдаёт разное сообщение в зависимости от того, нашла ли команда пиратов под номером team_number сокровище на данном участке острова), открывает мьютекс callout и возвращается к проверке переменной-счётчика. Так происходит до того, пока все потоки не завершат выполнение своих задач.

2) Источники информации с описанием данной модели

<https://pro-prof.com/forums/topic/parallel-programming-paradigms>
http://window.edu.ru/catalog/pdf2txt/971/67971/41350?p_page=20

3) Прочие источники информации, использованные при работе с данным заданием

<ru.wikipedia.org/wiki/Многопоточность>
<ru.wikipedia.org/wiki/Мьютекс>
<https://en.cppreference.com/w/cpp/thread/mutex>

4) Входные данные

Для задания входных параметров используются аргументы, заданные командной строкой. Формат командной строки:

1-ый аргумент – имя исполняемого файла (HW3_Task26)

2-ой аргумент – целое положительное число, количество потоков (групп пиратов), может принимать значения в диапазоне [1; 999]

3-ий аргумент – целое положительное число, количество участков, на которые Сильвер поделил остров, может принимать значения от 1 до максимального значения типа int (2147483647).

5) Выходные данные

Выходные данные – строковые сообщения, выводимые в консоль. Представляют собой доклады пиратов о результатах обследования конкретной части острова.

6) Дополнительные комментарии

Программа разработана с использованием стандартной библиотеки C++.

7) Тестовые наборы и примеры

Тестовые примеры располагаются в той же директории, где лежит настоящий файл, и именуются test*.png, где * - номер теста.