

ИДЗ №4 по АВС. Харитонов Кирилл Александрович БПИ 214. Вариант 16.

Претендую на оценку 9.

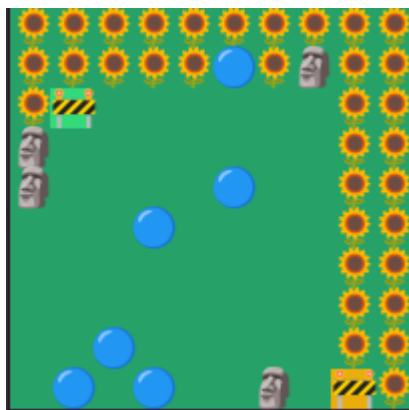
Критерии:

4 балла

- Условие задачи:
16. Задача о нелюдимых садовниках. Имеется пустой участок земли (двумерный массив) и план сада, разбитого на отдельные квадраты. От 10 до 30 процентов площади сада заняты прудами или камнями. То есть недоступны для ухода. Эти квадраты располагаются на плане произвольным образом. Уход за садом выполняют два садовника, которые не хотят встречаться друг другом (то есть, одновременно появляться в одном и том же квадрате). Первый садовник начинает работу с верхнего левого угла сада и перемещается слева направо, сделав ряд, он спускается вниз и идет в обратном направлении, пропуская обработанные участки. Второй садовник начинает работу с нижнего правого угла сада и перемещается снизу вверх, сделав ряд, он перемещается влево и также идет в обратную сторону. Если садовник видит, что участок сада уже обработан другим садовником или является не обрабатываемым, он идет дальше. Если по пути какой-то участок занят другим садовником, то садовник ожидает когда участок освободится, чтобы пройти дальше. Садовники должны работать одновременно со скоростями, определяемыми как параметры задачи. Прохождение через любой квадрат занимает некоторое время, которое задается константой, меньшей чем времена обработки и принимается за единицу времени. Создать многопоточное приложение, моделирующее работу садовников. При решении задачи использовать мьютексы.
- Модель параллельных вычислений: есть 3 потока, первый и второй моделируют работу садовников, третий является основным, управляющим, в нем происходит создание объектов сада и садовников, ввод параметров программы, запуск первого и второго потоков, вывод сада на экран или в файл (при необходимости) при возникновении изменений в саду, завершение работы первых и вторых потоков. Для синхронизации действий потоков используется mutex, запрещающий двум потокам одновременно изменять состояние квадрата поля сада (с “необработанный” на “находится под обработкой”) и одновременно считывать состояние квадрата, который может быть необработанным. Таким образом достигается поставленное в условии задачи требование о том, что если один садовник обрабатывает квадрат, то второй садовник ждет, пока первый закончит обработку прежде, чем пройти на этот квадрат.
- Входные данные: “скорость первого садовника”, “скорость второго садовника”, измеряемые в клетках в секунду, скорость описывает число необрабатываемых

клеток, которые садовник проходит за секунду. На обработку клетки тратиться в два раза больше времени, чем на проход по необрабатываемой. В целом скорость движения садовников ограничивается скоростью вывода данных в консоль на машине, на которой выполняется программа. В ходе тестирования не было замечено проблем при использовании чисел типа double в диапазоне [0.1; 10].

- Реализовано консольное приложение, решающее поставленную задачу с использованием одного варианта синхропримитивов: исходный код приложения содержится в файле main.cpp, исполняемый файл консольного приложения называется main. В приложении при выводе поля сада на экран и в файл приняты следующие обозначения: пустой квадратик зеленого цвета - необработанная клетка сада, квадратик зеленого цвета с эмодзи 🌻 - обработанная клетка сада, квадратик с эмодзи 🟦 - пруд, квадратик с эмодзи 🗿 - камни, квадратик с эмодзи ⚠️ - квадратик, находящийся под обработкой садовником, квадратик светло зеленого цвета - первый садовник, квадратик желтого цвета - второй садовник, квадратик коричневого цвета - оба садовника находятся на одном квадратике. Пример вывода поля сада:



Согласно условию задачи использовался синхропримитив мьютекс, более подробно его использование описано в предыдущем пункте и в обобщенном алгоритме.

- Ввод данных в приложение реализован через аргументы консоли: скорость_первого_садовника скорость_второго_садовника, например ./main 1 1.

5 баллов

- В программу добавлены комментарии, поясняющие выполняемые действия и описание используемых переменных.
- Сценарий одновременного поведения сущностей квадрат сада и садовники. Квадрат сада бывает 5 типов: неухоженный, ухоженный, камни, пруд, находящийся под уходом. Если садовник попадает на неухоженный квадрат, то он начинает его уход, длящийся некоторое время, иначе проходит по квадрата, проход по квадрату длится некоторое время, меньшее чем, обработка квадрата. Перед переходом на новый квадрат садовник проверяет, не занят ли новый квадрат другим садовником (я, в данном случае, понимаю “занят”, как “обрабатывается в данный момент другим садовником”, если же другой садовник просто проходит по этому квадрату, то занятым он не является), если

квадрат занят, то садовник ждет на предыдущем квадрате, пока новый квадрат не освободится, если квадрат не занят, то садовник делает ход на него и в зависимости от того, является ли квадрат обрабатываемым либо начинает обработку, либо нет.

6 баллов

- Обобщенный алгоритм: есть класс `Gardener`, который моделирует работу садовника, так же есть класс `Field`, который моделирует сущность поля сада. Сначала создается поле `field` класса `Field`, потом от 10 до 30% его заполняется камнями, либо прудами с помощью метода `placeObstacles`, который заполняет случайные клетки поля случайными препятствиями в нужном количестве. Далее создаются объекты `first_gardener` и `second_gardener` класса `Gardener`, которые являются субъектами “первый садовник” и “второй садовник”, имеют соответствующие скорости, точки старта и направления движения, далее создаются два потока, выполняющих функцию `startGardening`, аргументом которой является объект садовника. Эта функция вызывает у переданного ей садовника метод `move`, пока флаг садовника `haveFinished` (флаг сигнализирует о том, что садовник обошел все поле) не станет положительным. В методе `move` сначала вызывается метод `garden` у объекта `field`, с аргументами описывающими длительность шага по квадрату и локацию квадрата, на котором садовник сейчас находится. В методе `garden` сначала ставится мьютекс, потом проверяется состояние квадрата, если он является необработанным, то его состояние меняется на “находится под обработкой”, мьютекс снимается, вызывается функция `usleep`, поток простаивает время в 2 раза большее длительности шага, после чего состояние квадрата меняется на “обработан”, иначе, если квадрат является не обрабатываемым, то мьютекс разблокируется, и вызывается `usleep`, который тормозит поток на время шага. Далее возвращаемся в метод `move`, где проверяем является ли текущий квадрат конечным (здесь конечный квадрат - квадрат, после попадания, в который на поле сада не остается квадратов, на которых этот садовник еще не побывал), если он конечный, то выставляем флаг `haveFinished` на `true` и выходим из метода, иначе продолжаем выполнение метода. Далее проверяем может ли садовник походить в своем обычном направлении или он уперся в край поля с помощью метода `checkIfTheWayIsBlocked`, если может, то меняем координаты на которых находится садовник методом `moveInDirection`, в котором садовник в зависимости от переданного в качестве аргумента направления, ходит в него, предварительно проверив, что квадрат перед ним не является занятым, перед проверкой занятости квадрата включается мьютекс, чтобы другой садовник не мог изменить состояние квадрата, в тот момент, когда его состояние проверяет другой садовник, если квадрат занят, то садовник ждет, пока он освободится, далее мьютекс разблокируется, и происходит изменение координат садовника, иначе если садовник не может продолжать движение в том же направлении, то он с помощью метода `moveInDirection`, совершает движение в бок, относительно своего изначального направления движения, и направление движения меняется на противоположное. Далее метод `move` завершается и мы попадаем обратно в `startGardening`, который в зависимости от флага `haveFinished` либо вызывает `move` снова, либо выходит из `startGardening`. Параллельно с этим в основном

потоке через промежутки равные длительности шага, выводится в консоль, либо в файл поле сада, пока у обоих садовников haveFinished не станет true. Далее потоки останавливаются и программа завершается.

- Ввод данных из командной строки происходит при исполнении программы с флагом -s, например ./main -s. В командной строке появляется просьба ввести 2 числа на стандартный поток ввода, после ввода этих чисел запускается программа с соответствующими скоростями садовников.

7 баллов

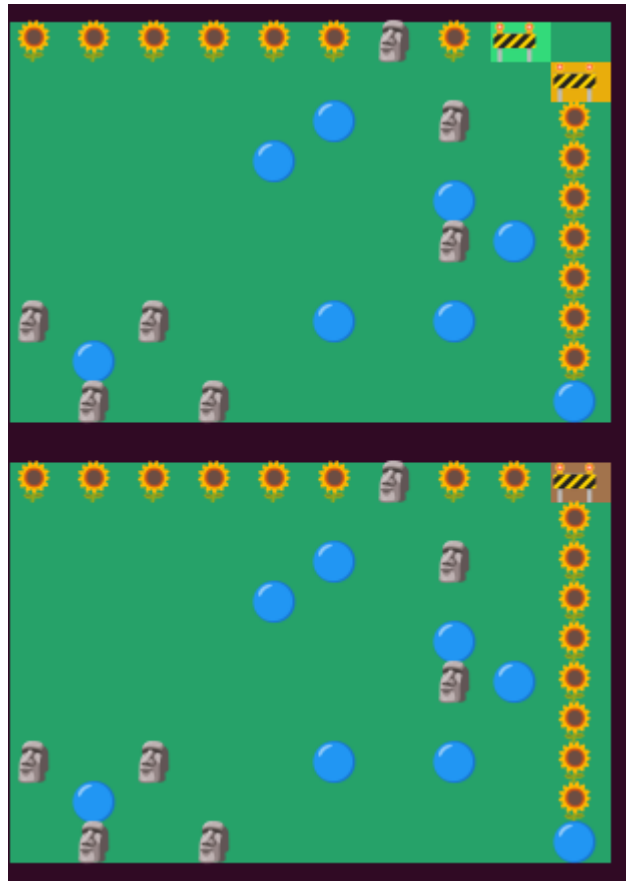
- С помощью флага -f с аргументами название_файла_с_входными_данными:название_файла_с_выходными_данными, например ./main -f input:output, реализован ввод данных из файлов и вывод данных в файлы. При выводе результатов работы программы в файл цвета клеток не отображаются ввиду ограничений текстового формата.
- Входные файлы с данными: input, input2, input3. Выходные файлы с данными с соответствующими результатами выполнения программы: output, output2, output3.
- Результаты работы программы выводятся на экран всегда и записываются в файл, при использовании соответствующего флага.
- Ввод данных из командной строки расширен - добавлен флаг -f.


8 баллов

- В программу добавлен флаг -r, пример использования: ./main -r. При его использовании программа генерирует 2 случайных числа от 0.1 до 9.1, которые выступают в качестве скоростей садовников и выполняет программу с заданными скоростями.
- Файлы с результатами работы программы с генерацией случайных данных (в начале файла указаны сгенерированные данные): output_rand1, output_rand2, output_rand3. Данные отображаются иначе из-за того, что текстовый формат не поддерживает цвета.
- Ввод данных расширен - добавлен флаг -r.

9 баллов. Вариант 1.

- Если отключить пары закрытия и открытия мьютекса в функции moveInDirection, то становятся возможными ситуации, когда садовники одновременно посмотрели на квадрат, увидели, что он не занят, походили на него, увидели что он не обработан, и один принялся его обрабатывать (тот поток, который вперед активировал мьютекс в функции garden), а другой поток просто стоит на клетке время равное длительности хода. Если же отключить мьютексы в функции garden, то возможны ситуации, когда оба потока одновременно увидели, что статус квадрата - "необработан", поменяли статус квадрата на "обрабатывается" и теперь одновременно вдвоем обрабатывают его. Входными данными для возникновения такого поведения должны быть поля сада, где, чтобы добраться до какого либо из квадратов для каждого садовника требуется одно и тоже время. Пример вывода программы в случае возникновения подобной ситуации:



Здесь коричневый квадрат в верхнем правом углу, на нижнем поле сада показывает, что оба садовника находятся на этом квадрате, а  обозначает, что данный квадрат сейчас находится под обработкой, так же сверху можно увидеть поле сада в момент предыдущего хода садовников и саму расстановку препятствий, скорости садовников 1 и 1.