Coroutines

Ваша задача - реализовать Conway's Game of Life, с использованием Kotlin корутин и каналов.

1. Для хранения состояния игрового поля реализуйте immutable класс Grid. Экземпляр этого класса, хранящий текущее состояние будет разделяться между всеми корутинами.
2. Расчет нового состояния игрового поля должен выполняться набором корутин, выполняющихся параллельно в некотором пуле потоков. Каждая корутина должна обрабатывать свою область поля и генерировать область с новым состоянием. Количество корутин должно быть пропорционально количеству логических процессоров в системе (например, по 2 корутине на процессор).
3. Расчитанное состояние каждая корутина должна сохранять в некоторый канал, из которого затем оно будет извлекаться читателем, формирующим новое состояние игрового поля в виде нового экзепляра Grid. Таким образом, ваше решение будет реализовывать паттерн fan-in.
4. Программа должна считывать параметры из стандартного потока ввода, выполнять моделирование и выводить полученное состояние в стандартный поток вывода.

Формат ввода

Первая строка ввода содержит целые числа **W**, **H**, **I** (**1 ⇐ W, H ⇐ 10^5**; **1 ⇐ I ⇐ 100**), ширину и высоту игрового поля и количество итераций соответственно, и символ **'R'** или **'I'** (без кавычек), разделенные одним символом пробела. Если в первой строке указан символ **'R'**, это означает, что начальное состояние игры должно генерироваться случайным образом, в противном случае, следующие **H** строк содержат начальное состояние игрового поля. Каждая строка состоит из **W** символов **'*'** или **'.'** (без кавычек). Символ **“*”** означает, что клетка занята, **'.'** - пуста.

Формат вывода

Выведите состояние игрового поля после **I** итераций в том же формате: в виде **H** строк по **W** символов в каждой.

Пример 1

```
10 5 18 R
```

Необходимо вывести состояние игрового поля размером 10 * 5, после выполнения 18 итераций, с начальным состоянием, сгенерированным случайным образом.

Пример 2

```
6 3 10 I
*..**.
.*.*..
....**
```

Необходимо вывести состояние игрового поля размером 6 * 3, после выполнения 10 итераций, с начальным состоянием, заданным явно в потоке ввода.

From:
http://wiki.osll.ru/ - **Open Source & Linux Lab**

Permanent link:
http://wiki.osll.ru/doku.php/courses:high_performance_computing:coroutines

Last update: **2019/12/11 11:24**