

Отчет «Game of Life»

Пантелеева Наталья

Характеристики процессора: 2,7 GHz 2-ядерный процессор Intel Core i5

Память: 8 ГБ 1867 MHz DDR3

Правила игры:

- Место действия этой игры — «вселенная» — это размеченная на клетки поверхность или плоскость — безграничная, ограниченная, или замкнутая (в пределе — бесконечная плоскость).
- Каждая клетка на этой поверхности может находиться в двух состояниях: быть «живой» (заполненной) или быть «мёртвой» (пустой). Клетка имеет восемь соседей, окружающих её.
- Распределение живых клеток в начале игры называется первым поколением. Каждое следующее поколение рассчитывается на основе предыдущего по таким правилам:
  - в пустой (мёртвой) клетке, рядом с которой ровно три живые клетки, зарождается жизнь;
  - если у живой клетки есть две или три живые соседки, то эта клетка продолжает жить; в противном случае, если соседей меньше двух или больше трёх, клетка умирает («от одиночества» или «от перенаселённости»)
- Игра прекращается, если
  - на поле не останется ни одной «живой» клетки
  - конфигурация на очередном шаге в точности (без сдвигов и поворотов) повторит себя же на одном из более ранних шагов (складывается периодическая конфигурация)
  - при очередном шаге ни одна из клеток не меняет своего состояния (складывается стабильная конфигурация; предыдущее правило, вырожденное до одного шага назад)

Описание алгоритма:

Для случая поля 9x9 разбиваем поле на ячейки 3x3

Для случая 100x100 разбиваем поле на ячейки 10x10.

Для случая 1000x1000 разбиваем поле на ячейки 100x100.

Каждый поток выполняет свою клетку. Допустим на поле 9x9 одновременно запускаются 9 потоков. Каждый поток ждет оставшиеся потоки, после чего полученный результат передается на следующий шаг.

Сначала я создавала новую матрицу на каждом шаге. Это было медленнее, а потом я решила использовать только две матрицы и чередовать их на каждом шаге. Из одной матрицы я считывала, а в другую записывала, причем каждый поток работал со своей клеткой. Так обеспечивается потокобезопасность в моей программе, не нужно использовать специальные инструменты для синхронизации, т.к. мы не считываем изменяемые данные.

В ходе работы, у меня возникли сложности с определением границ. Пришлось немного вспомнить математику. В итоге, пришлось добавлять условие, когда индексы выходили за пределы массива. Возникли сложности с разделением поля на клетки. Проблема была в целочисленном делении и в некоторых случаях приходилось добавлять остаток от деления чтобы покрыть поле полностью.

Результаты:

Для случая поля 9x9 (input.txt) время выполнения - 618 мс

Для случая 100x100 (input100.txt) - 1 500 мс

Для случая 1000x1000 (input1000.txt) - 28 000 мс

В сравнении с однопоточным первые два случая не сильно отличаются по времени, а для третьего случая увеличиваются примерно в 1.5-2 раза.