

Java Core. Продвинутый уровень

Урок 5

Многопоточность

Потоки в Java, способы создания и управления состояниями потоков, основные проблемы при работе с потоками и синхронизация



План урока

- Как несколько приложений одновременно работали на одноядерных системах
- Что такое процессы и потоки, как многопоточность позволяет ускорить работу приложения
- Синхронизация



Как несколько приложений
одновременно работали на
одноядерных системах;



Одноядерный процессор и несколько приложений



Двухядерный процессор и несколько приложений



Что такое процессы и потоки, как
многопоточность позволяет
ускорить работу приложения



Преимущество многопоточности

Очень большое
изображение
10000x10000

На обработку фильтром этого
изображения уходит 60 сек.
Как можно ускорить это?



Создание и запуск потоков

Поток main

Фильтрация изображения в один поток(~60 сек)



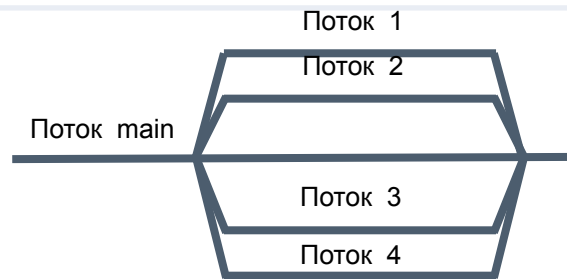
Преимущество многопоточности



Разбиваем на 4 части, и
обрабатываем каждую из них
параллельно, после чего
собираем обратно



Создание и запуск потоков



Разбивка изображения на 4 части, параллельная обработка и сшивка (~18 сек)



Синхронизация





Переходим к практике в
IntelliJ IDEA



Практическое задание

Необходимо написать два метода, которые делают следующее:

1) Создают одномерный длинный массив, например:

```
static final int SIZE = 10 000 000;  
static final int HALF = size / 2;  
float[] arr = new float[size].
```

2) Заполняют этот массив единицами.

3) Засекают время выполнения: long a = System.currentTimeMillis().

4) Проходят по всему массиву и для каждой ячейки считают новое значение по формуле:

```
arr[i] = (float) (arr[i] * Math.sin(0.2f + i / 5) * Math.cos(0.2f + i / 5) *  
Math.cos(0.4f + i / 2)).
```

5) Проверяется время окончания метода System.currentTimeMillis().

6) В консоль выводится время работы: System.out.println(System.currentTimeMillis() - a).



Практическое задание

Отличие первого метода от второго:

- Первый просто бежит по массиву и вычисляет значения.
- Второй разбивает массив на два массива, в двух потоках высчитывает новые значения и потом склеивает эти массивы обратно в один.

Пример деления одного массива на два:

- `System.arraycopy(arr, 0, a1, 0, h);`
- `System.arraycopy(arr, h, a2, 0, h);`

Пример обратной склейки:

- `System.arraycopy(a1, 0, arr, 0, h);`
- `System.arraycopy(a2, 0, arr, h, h);`

Примечание:

`System.arraycopy()` — копирует данные из одного массива в другой.
`System.arraycopy(массив-источник, откуда начинаем брать данные из массива-источника, массив-назначение, откуда начинаем записывать данные в массив-назначение, сколько ячеек копируем)`

По замерам времени:

Для первого метода надо считать время только на цикл расчета. Для второго метода измеряете время разбивки массива на 2, просчета каждого из двух массивов и склейки.



Вопросы участников . . .

