Параллельное программирование: Лекция 7

Учебный курс

Кензин Максим Юрьевич
Кафедра Информационных технологий
ИМИТ ИГУ / Институт динамики
систем и теории управления СО РАН

ИМИТ ИГУ, 2022

Семафоры

Несколько особенностей:

- Порядок Acquire() и Release();
- Semaphore(0).

Задача 2: Парикмахерская.

Есть парикмахерская, в которой работают 3 парикмахера.

В парикмахерской есть салон, состоящий из зоны стоячего ожидания и дивана вместительностью 7, и комната парикмахеров.

Всего салон вмещает 10 клиентов:

- Если салон полный, новые клиенты уходят.
- Если есть место на диване, то новый клиент садится на диван.
- Если диван занят, клиент встает в зоне стоячего ожидания.

К парикмахеру всегда идет клиент, сидящий на диване дольше всех, его место на диване занимает клиент, стоявший на ногах дольше всех.

После завершения стрижки клиент платит деньги. Платить можно любому парикмахеру, но так как кассовый аппарат всего один, то оплата производится клиентами по очереди.

Блокировка (Локи)

- Класс **ReentrantLock** (*java.util.concurrent.locks*) – альтернатива synchronized:
  - void lock(): пытается получить блокировку; если не удалось, ожидает, пока не будет получена блокировка;
  - boolean tryLock(): пытается получить блокировку, если блокировка получена, то возвращает true. Если блокировка не получена, то возвращает false (не ожидает ее получения);
  - int void unlock(): снимает блокировку (всегда в finally);
  - Condition newCondition(): возвращает объект Condition, который связан с текущей блокировкой.

Lock (L) vs Synchronized (S)

1. S не гарантирует сохранения порядка получения доступа потоками, ждущими в очереди;

2. L позволяет задавать таймаут для попытки получения доступа к критическому блоку;

3. В случае с S критический блок должен выполняться целиком внутри одного метода.
   В случае с L можно получить доступ в одном методе, а отдать в другом.

Блокировка: Conditions

- Conditions – альтернатива wait/notify/notifyAll:
  - await(), signal(), signalAll().
  - Нужен объект Condition:

```
ReentrantLock locker = new ReentrantLock();
Condition condition = locker.newCondition();
condition.await();
condition.signal();
...
```

У одного Lock может быть несколько разных Condition, и разные потоки могут ожидать выполнения разных условий (Condition) на одном и том же локе!

Задание: Конвейерная лента 1.0

На некотором заводе есть конвейерная лента.

Два *одинаковых* робота-погрузчика по готовности ставят на ленту детали, собранные разными цехами.

Есть робот-разгрузчик, который с заданной периодичностью снимает с ленты детали.

В тот момент, когда общий вес всех деталей на конвейере превышает заданную критическую массу, погрузка и разгрузка останавливаются, после чего:

1. В консоль выводится сообщение о перегрузке, содержащее данные о текущем количестве деталей на конвейере и их весе.
2. Конвейер полностью очищается от всех деталей.
3. Погрузка и разгрузка возобновляются.

Сделать многопоточную реализацию программы.

Задание: Конвейерная лента 1.1

На некотором заводе есть конвейерная лента.

Два *одинаковых* робота-погрузчика по готовности ставят на ленту детали, собранные разными цехами.

Есть робот-разгрузчик, который с заданной периодичностью снимает с ленты детали.

В тот момент, когда общий вес всех деталей на конвейере превышает заданную критическую массу, погрузка и разгрузка останавливаются, после чего:

1. Включаются два робота по контролю качества;
2. Робот А находит самую легкую и самую тяжелую детали, и снимает их с ленты на переработку.
3. Затем Робот В находит самую легкую и самую тяжелую детали из оставшихся, и отправляет их в отдел качества.
4. Погрузка и разгрузка возобновляются.

Задание: Конвейерная лента.

Требования к реализации:

1. Все роботы – потоки.
2. Деталь имеет индекс и случайный вес 300-500 грамм.
3. Роботы-погрузчики поставляют детали раз в 100-1000мс.
4. Робот-разгрузчик забирает деталь раз в 400мс.
5. Максимально допустимый вес деталей на конвейере – 4кг.
6. Реализация конвейера через ArrayList.
7. Реализация работы всех всех роботов через ReentrantLock и condition.signal / condition.awake.
8. Сделать полное логирование происходящего.
9. (для простой задачи) Длительности очистки конвейера 500мс.