# Теория и практика многопоточного программирования

ЛЕКЦИЯ 8. ЕЩЁ ПРО КОНСЕНСУС

# В прошлый раз...

Консенсус

Число консенсуса

Протокол, состояние протокола

Валентность состояния

Создание консенсуса из примитивов

# Темы лекции

Универсальный объект

Невозможность консенсуса в системе со сбоями

# Число консенсуса

Атомарные регистры – 1

RMW (get-&-set, get-&-add), очередь – 2

(n, n*(n+1)/2)-регистры – N

Compare-and-set - ∞

# Универсальность объектов

В системе N потоков объект называется **универсальным**, если, используя его экземпляры и RW-регистры, можно сконструировать неожидающую линеаризуемую реализацию любого разделяемого объекта.

Класс является универсальным в системе N потоков, если и только если его *число консенсуса* больше либо равно N.

# Последовательная нотация

```
public interface SequentialObject {

    abstract ResponseType
                    apply(InvocationType type);

}
```

InvocationType = вызываемый метод (делегат) + аргументы

ResponseType = результат + состояние выхода (ok, exception)

# Представление детерминированного объекта

**Детерминированный объект** – для любого состояния и любых входных параметров результат выполнения метода предсказуемо неизменен.

Представление разделяемого объекта: объект в начальном состоянии + лог (упорядоченная последовательность вызовов методов).

Исполнение метода потоком – добавление вызова в «голову» списка. Поток может наблюдать результат только последнего вызванного метода

# Реализация универсального детерминированного объекта

1. Сформировать очередь вызовов методов объекта
   - Используя консенсус, предлагать свой «вариант» вызова, пока он не будет помещён в список
2. Создать экземпляры по числу потоков
3. Выполнить в порядке очереди над каждым объектом вызовы
4. Всё это можно сделать, используя объекты консенсуса и массивы RW-регистров
5. Существуют lock-free и wait-free реализации

***Вывод***: можно воссоздать любой линеаризуемый последовательно-описанный объект с сохранением свойства wait freedom через универсальный объект

# Консенсус в «сбоящих» системах

Определение wait freedom подразумевает устойчивость к сбоям – если другой поток остановился или прервался, всё равно ожидается завершение за конечное число шагов.

# FLP impossibility: Асинхронная система

Существует теорема ***Фишер-Линч-Патерсон***, показывающая, что для асинхронной системы N потоков с  хотя бы одним “прерывающимся” потоком нельзя построить решение задачи консенсуса

# FLP: Основные понятия

Система описывается состоянием **C** потоков **{p}** и сообщениями $x_i$=**(p,m)**, передаваемыми через буфер.

$\sigma = \ldots C_i, x_i, C_{i+1}, x_{i+1}, \ldots$ - последовательность исполнения (м.б. бесконечная)

Корректный поток **p** – поток, который не прерывается и все отправленные ему сообщения прочитаны.

Протокол консенсуса может быть «**частично корректным**», если выполены 2 условия:

- Ни одно из достижимых состояний не может **привести** к разным решениям
- **Существуют** сценарии, которые приводят ко всем предложенным значениям

# FLP: исполнение

«**Решающее исполнение протокола**» - хотя бы один достиг решающего состояния (отработал протокол)

**Полностью корректный консенсус** – консенсус, который частично корректен, и каждое исполнение протокола является решающим

Доказательство **невозможности** существования ПКК для системы потоков с 1 «некорректным» состоит из 2 частей:

- Покажем, что начальное состояние бивалентно (доказано)
- Построим такую последовательность исполнения, что для каждого очередного состояния ход $\mathbf{x}_i$ будет переводить систему в бивалентное состояние

# FLP: схема доказательства 2

**Лемма 1 (о важности каждого шага):**

Если есть бивалентная конфигурация C, для которой допустимо событие ***e***, и есть множество ***L*** конфигураций, достижимых из C без применения ***e***, то среди порождённых состояний e(***L***) найдётся бивалентное (док-во от противного)

То есть существуют состояния, для которых событие ***e*** не является «решающим»

Конструирование базируется на последовательных этапах, при которых каждый очередной переходит в бивалентное состояние.

# FLP: схема доказательства 2

**Теорема-следствие**:

Существует (сконструирован) частично корректный протокол консенсуса, в котором корректные потоки достигают решения, если ни один из потоков не умирает по ходу их исполнения и строгое большиство потоков на старте было «живо».