

# Теория и практика многопоточного программирования

---

ЛЕКЦИЯ 7. СОГЛАСОВАННОСТЬ, ИСТОРИЯ,  
ЛИНЕАРИЗУЕМОСТЬ

# В прошлый раз

---

Математическая модель многопоточной программы

Терминология теории параллельного программирования

Реализация объектов блокировки

# Темы лекции

---

Свойства согласованности и регистры

История

Линеризация истории

# Объект

---

Объект – совокупность **состояния** (данных) и **методов**. Методы выполняют вычисления и манипулируют состоянием.

Регистр – объект чтения-записи

```
interface IRegister<T>
{
    T read();
    void write(T val);
}
```

# Метод

---

Формальное описание метода – **последовательная спецификация** (sequential specification):

{состояние  $Q$ , параметры  $e$ }  $\Rightarrow$  {состояние  $Q^*$ , результат  $r$ }

Неформально, *метод* – это совокупность сигнатуры и ожидаемого поведения

У метода есть *события*:

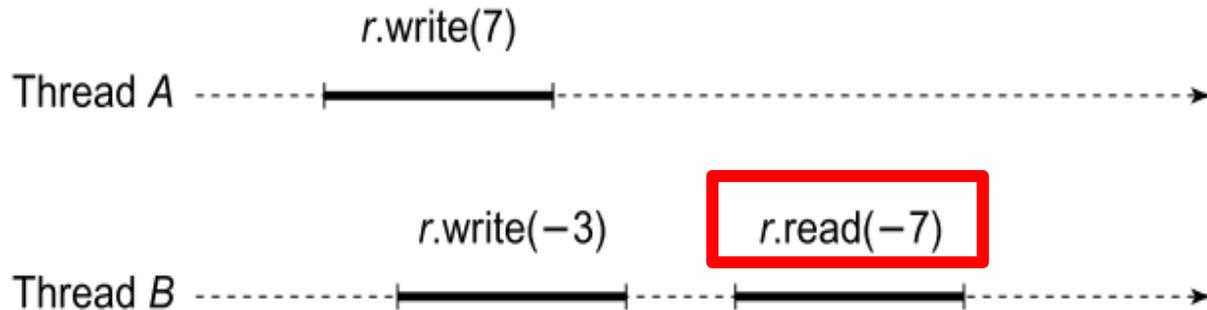
**начало** (invocation), **завершение** (response) и видимый **результат** (result)

Метод называется **тотальным**, если он определён для всех состояний объекта

# Согласованность (consistency) по периодам покоя (quiescent)

Система потоков согласована по периодам покоя, если:

- В одном потоке одновременно может выполняться один метод
- Если вызовы разделены периодом покоя, то их результаты будут иметь место в порядке выполнения.

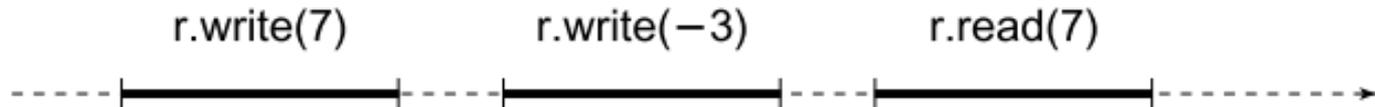


- Регистр, удовлетворяющий свойству согласованности по периодам покоя, называется **безопасным** (safe). При перекрывающихся чтении-записи не гарантирует ничего

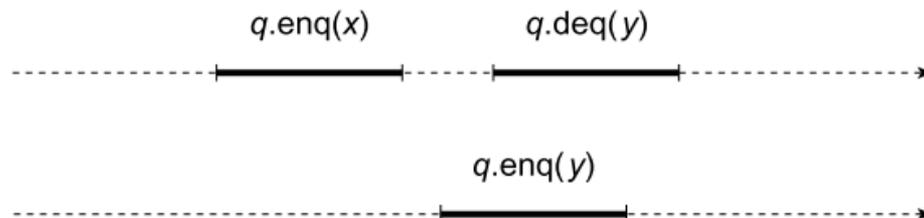
# Упорядоченная (sequential) согласованность

Система потоков упорядоченно согласована, если:

- В одном потоке одновременно может выполняться один метод
- В одном потоке методы выполняются в порядке вызова



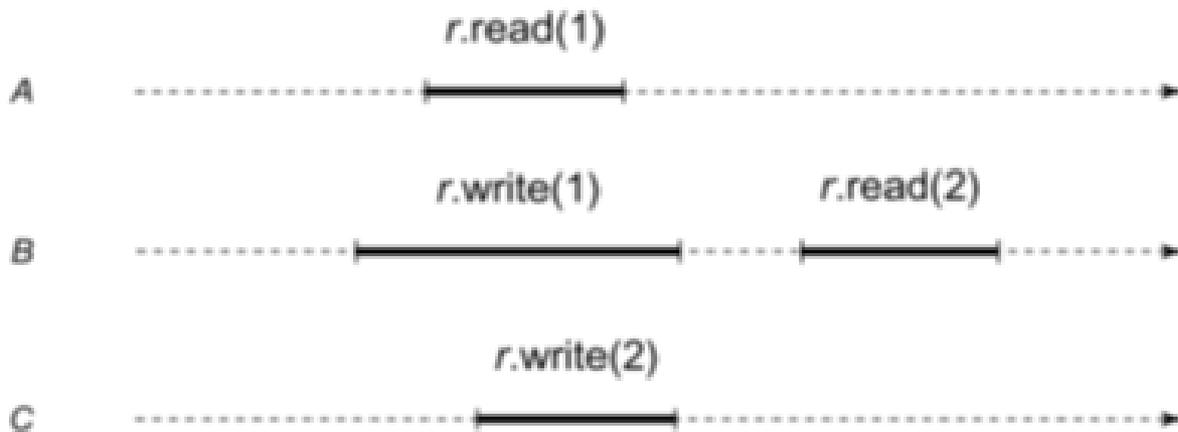
- Регистр, удовлетворяющий свойству упорядоченной согласованности, называется **регулярным** (regular). При перекрывающихся чтении-записи гарантирует возврат старого или нового значения.



# Линеаризуемость

---

Метод является линеаризуемым, если **результат** его выполнения лежит между **вызовом** и **завершением**. Момент наступления результата называется **точкой линеаризации**.



Линеаризуемый регистр называется **атомарным**.

# СВОЙСТВО КОМПОЗИЦИОННОСТИ

---

Свойство корректности  $P$  является **композиционным** для системы объектов в том случае, что если каждый из объектов в системе удовлетворяет  $P$ , то и система в целом удовлетворяет  $P$ .

Неблокирующая (по периодам покоя) корректность и линеаризуемость являются композиционными.  
Упорядоченная не является.

# История

---

**История** многопоточной системы – конечная последовательность **вызовов** и **возвратов** из методов (позитивных, или с исключением).

**Подыстория** – подпоследовательность истории.  
Например,  $H|A$  – подпоследовательность истории, состоящая из методов, выполнявшихся в потоке  $A$ .  
Подыстория  $H|x$  – подпоследовательность для методов объекта  $x$ .

Истории  $H$  и  $H'$  эквивалентны, если  $\forall A: H|A \equiv H'|A$

# История: расширение

---

Метод **ожидает**, если для него в истории нет последующего возврата.

**Расширение** *extension*(**H**) истории **H** – дополнение истории возвратами из ожидающих методов (необязательно всех).

*complete*(**H**) – завершённая история, подпоследовательность **H** за исключением ожидающих методов.

История называется **последовательной**, если исполнение методов в ней не перекрывается (за каждым вызовом метода следует соответствующий ему возврат, кроме, может быть, последнего).

Назовём историю «**оформленной**» (well formed), если любая подыстория *по потоку* в ней последовательна (это необязательно для объектов).

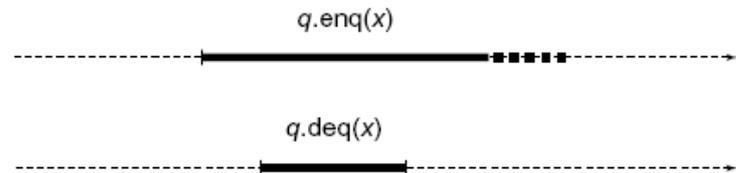
История является **легальной**, если для каждого объекта его подыстория легальна (т.е. логична, допустима)

# История: линеаризуемость

История  $H$  является **линеаризуемой**, если для неё существует расширение  $H' = extension(H)$ , и существует **последовательная** история  $S$  такая, что:

- 1.  $complete(H')$  эквивалентна  $S$
- 2. Если исполнение метода  $m_0$  предшествовало исполнению метода  $m_1$  в истории  $H$ , то то же верно и для истории  $S$ .

$S$  называется **линеаризацией**  $H$ .



*Трактовка условий:*

[1] Некоторые ожидающие вызовы могли уже опубликовать свой результат, даже если они ещё не завершены.

[2] Если в оригинальной истории метод полностью выполнен раньше другого, то это отразится в линеаризации.

# История: композиционность линеаризуемости

---

История  $H$  линеаризуема, **если и только если** все подыстории по объектам  $H|x$  линеаризуемы.

## Доказательство

- 1) **если** (объекты линеаризуемы, история - ?)
- 2) **Только если** (история линеаризуема, объекты - ?).