



# ТЕОРИЯ И ПРАКТИКА МНОГОПОТОЧНОГО ПРОГРАММИРОВАНИЯ

---

## Тема 9

Задача о консенсусе (согласии).

Д.ф.-м.н., профессор А.Г. Тормасов  
Базовая кафедра «Теоретическая и Прикладная Информатика», МФТИ

# Тема

- Задача о консенсусе (согласии).
- Использование понятия “валентности” для доказательства.
- Задача о соглашении для  $k$ -набора как обобщение задачи о консенсусе

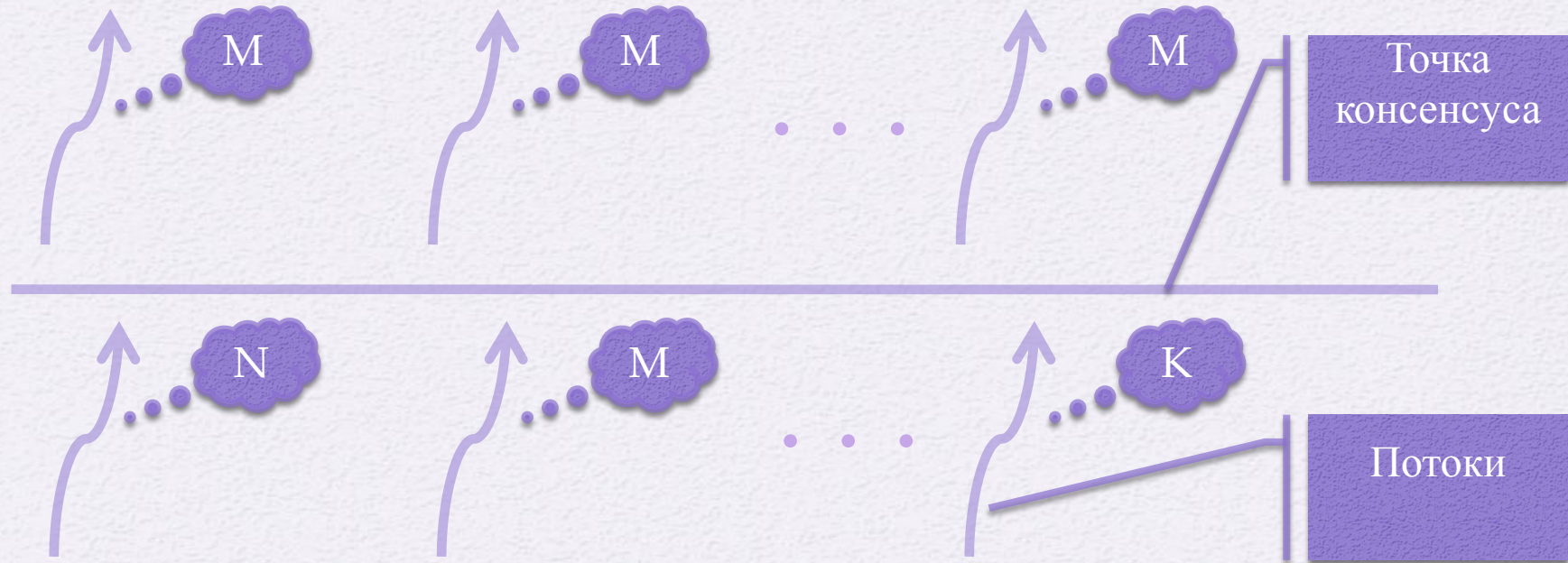
# Как...

- Как можно сравнить разные примитивы синхронизации между собой?
- Какие нужно реализовать аппаратные примитивы для нового микропроцессора?
- Можно ли реализовать один примитив из другого?
- Для решения этих вопросов необходимо создание какой-то формы или механизма сравнения, лучше всего в виде формализации, пригодной затем для создания и доказательства каких либо утверждений.

# КОНСЕНСУС

- Задача о консенсусе – это некоторая абстрактная задача, которая признается многими исследователями как удобный способ оценки синхронизационных проблем.

# Задача о консенсусе



# Задача о консенсусе...

- пусть есть  $n$  потоков
- Все работают с одной разделяемой областью
- каждый поток пытается поместить туда значение  $v_i$
- По достижению консенсуса:
  - существует значение  $j$ , когда все участвующие (без исключения) потоки считают, что значение разделяемой области есть  $v_j$ .
  - Значение  $j$ , возможно, не единственное

# Задача о консенсусе...

- Объект консенсуса предоставляет 1 метод
  - `value decide(value in);`
- Каждый поток вызывает его со своим значением
  - `decide(vi);`
- Метод ВО ВСЕХ потоках возвращает всегда одно и то же значение, с которым поток идет далее

# Задача о консенсусе...

- решение непротиворечиво
  - Метод возвращает одно значение, одинаковое для всех потоков
- решение корректно
  - предложено всегда одним из потоков
- завершенность
  - каждый корректный процесс когда нибудь принимает решение



# Задача о консенсусе...

- соисполнимый объект консенсуса линеаризуется в последовательный объект консенсуса
- тот поток, значение которого было принято, завершает вызов `decide()` первым
- `decide()` вызывается каждым потоком не более одного раза
  - по определению, реализация протокола консенсуса, обладающая свободой от блокировок, будет свободна от ожидания
  - и наоборот.

# Определение

- Определение.  
“Протокол консенсуса” есть реализация объекта консенсуса, свободная от ожидания.
- Далее ограничимся детерминированными объектами с последовательной спецификацией
  - каждый последовательный вызов метода имеет единственный результат

# Объект консенсуса

## Определение

Класс  $C$  решает проблему консенсуса для  $n$  потоков, если существует протокол консенсуса, использующий для принятия решения любое количество объектов класса  $C$  и любое количество атомарных регистров

## Определение

Числом консенсуса класса  $C$  является максимальное количество потоков  $n$ , для которых этот класс решает проблему консенсуса. Число консенсуса может быть бесконечно большим.

## Следствие

Если существует реализация объекта  $C$  из одного или более объектов класса  $D$  и атомарных регистров, то объект  $D$  имеет число консенсуса, как минимум, равное числу консенсуса объекта  $C$ .

# Консенсус - применение

- «похожа» на то, что решается при обеспечении синхронизации
  - Пронумеруем потоки перед заходом в критическую секцию
  - Работает в ней только тот, чье решение принято
  - Остальные – ждут!
  - Не все потоки обязаны посетить точку консенсуса

# Типичная задача

- Бинарный консенсус
  - Два потока предлагают значения 0 или 1
  - обобщение на произвольный набор потоков и значений – оставим в качестве упражнения
- Каждый поток предпринимает какие то действия (ходы), которые в конечном итоге приводят к решению.
  - Ход есть вызов какого либо метода разделяемого объекта
- Состояние протокола состоит из состояний всех потоков и разделяемых объектов.
- Начальное состояние есть состояние протокола перед ходами всех участвующих потоков.
- Конечное состояние – состояние протокола после того как все ходы всех потоков закончились.
- Решением назовем значение всех потоков в финальном состоянии.

# Типичная задача...

- Дерево состояний
  - Узел (вершина) - состояние
  - Ребро – ход из одного состояния в другое
  - Лист (решение) помечается принятым значением
- Для бинарного дерева
  - 2 ребра для каждого узла
  - Листья помечаются только 0 или 1
  - Ребро соединяет предшествующие узлы (и состояния)

# Валентность

- Возможность попасть в какое то состояние
- Состояние является бивалентным, если двигаясь из него, можно попасть в листы, помеченные разными значениями
- Состояние является унивалентным, если все листы помечены одним значением
  - 1-валентное
  - 2-валентное
- Пунктирные и сплошные линии на рисунке – ходы разных потоков





# начальное бивалентное состояние

Лемма.

Для любого бинарного протокола консенсуса  $n$  потоков, при  $n > 1$ , существует начальное бивалентное состояние.

Доказательство

каждый из участвующих потоков в принципе может не «видеть» другого и завершить протокол исключительно со своим состоянием – запуск *соло* (поскольку это единственное состояние, которое он может увидеть при отсутствии ходов второго потока). Тогда, если хотя бы два потока предлагают разные состояния, то, очевидно, начальное состояние бивалентно, так как выходя из него мы можем попасть в два возможных конечных состояния.

# Критическое состояние

## Определение

*критическое состояние* является бивалентным, и при любом ходе становится унивалентным.

Лемма. Каждый свободный от ожидания протокол имеет критическое состояние.

- свободный от ожидания протокол обязан завершиться за конечное число ходов
- в начале у нас есть бивалентное состояние, тогда как в конце оно становится унивалентным

# Соглашение для $k$ -набора

- пусть есть  $n$  потоков
- Все работают с одной разделяемой областью
- каждый поток пытается поместить туда значение  $v_i$
- По достижению консенсуса:
  - существует набор не более чем  $k$  значений  $\{j_1, \dots, j_k\}$ , когда все участвующие (без исключения) потоки считают, что значение разделяемой области есть  $v_j$ , где  $j$  принадлежит  $\{j_1, \dots, j_k\}$ .
- Каждый поток принимает одно из не более чем  $k$  значений

# Соглашение для $k$ -набора

## Теорема

создание свободного от ожидания алгоритма, использующего только атомарные регистры чтения-записи для любого  $k < n$ , невозможно.

## Следствие

число консенсуса не является исчерпывающей характеристикой вычислительной мощности объекта

Так как, согласно теореме, регистрами, у которых число консенсуса 1, решить задачу для  $k=2$  нельзя, но и решить задачу о консенсусе для 2 процессов этим объектом тоже нельзя (это легко показать – просьба проверить самостоятельно), то условное число консенсуса для объекта соглашения для  $k$ -набора лежит между 1 и 2!

# Выводы

- Задача о консенсусе является «обобщением» практики работы процедур и методов синхронизации
- Обычно не используется в явном виде в реальном программировании («непрактична»), но из нее можно построить решения типовых задач
- Будем рассматривать только свободные от ожидания протоколы консенсуса
- Для формализации понятий и доказательство введена задача бинарного консенсуса, введены понятия валентностей и нотация записи состояний
  - Начальное состояние
  - Критическое состояние
  - Конечное состояние – решение
- Задача о соглашении для  $k$ -набора является обобщением задачи о консенсусе и может являться свидетельством того факта, что число консенсуса не является исчерпывающей характеристики вычислительной мощности какого то объекта

**(с) А. Тормасов, 2010-11 г.**

Базовая кафедра «Теоретическая и Прикладная Информатика» ФУПМ МФТИ  
tor@cres.mipt.ru\_

Для коммерческого использования курса просьба связаться с автором.