

# Задание №9. Консенсус

Рубаненко Евгений

Май 2017

## 1 Консенсус с помощью пустых очередей

Итак, тут мы с помощью двух очередей хотим как-то определить *winner* для задачи консенсуса. Как будет происходить выбор:

*Утверждается, что dequeue() на пустой очереди вернет какое-то выделенное значение*

*Псевдокод:*

```
if thread_id = 0
  then queue1.enqueue(0)
  if queue2.dequeue() == empty
    then winner = 0
  else
    if queue1.dequeue() == empty
      then winner = 0
    else
      winner = 1
else
  queue2.enqueue(1)
  if queue1.dequeue() == empty
    then winner = 1
  else
    winner = 0
```

Идея такая - мы с помощью очередей как-то обозначаем, кто пришел первым, а потом проверяем, кто действительно был первым. Проверка корректности есть рассмотрение различных вариантов исполнения. Например, если второй поток быстро установил значение во вторую очередь и прочитал *empty* из первой - то победителем объявляется 1. При таком раскладе первый поток так же объявит победителем 1, потому что все *if* вернут *false*. Остальные случаи проверяются аналогично.

## 2 Консенсус с помощью memory-to-memory swap

Далее будет показано, что *consensus number* =  $\infty$ .

Будем считать, что у нас есть много потоков и каждый поток может обращаться к ячейкам памяти *input<sub>i</sub>* и *competitor<sub>i</sub>*. (Названия, вроде бы, говорящие, и отражают смысл этих ячеек; *competitor<sub>i</sub>* изначально есть ноль) Будем также предполагать, что у нас есть еще одна ячейка памяти

$prize$  - и она изначально инициализированна единицей. Далее все потоки делают  $swap(prize, competitor_i)$  и что-то происходит. Победитель определится так: мы посмотрим, где же оказался 'приз', и этот поток объявим победителем. (Получилась какая-то жизненная программа)

### 3 Консенсус с помощью memory-to-memory copy

Далее будет показано, что  $consensus\ number = \infty$ .

Будем считать, что у нас есть много потоков и каждый из них может обращаться к ячейкам памяти  $p_i$  и  $q_i$ . Как и в предыдущих заданиях, происходит соревнование за звание победителя. Каждый поток делает это так: сначала он копирует значение из  $p_i$  в  $q_i$ , а затем для всех  $i' > i$  меняет  $p_{i'}$  на 0 (изначально там были, например, 1). После этого он идет назад до первой ненулевой ячейки. Она (имеется в виду, ее индекс) и является победителем. Действительно, назвать такую ячейку (имеется в виду индекс ячейки) победителем будет 'правильно', так как она была проинициализирована раньше того, как в нее записали 0. Также при таком подходе не появятся проблемы в процессе исполнения: нули в  $p_{i'}$  означают, что теперь  $q_{i'}$  никак не изменится ( $i' > i$ ); а при  $i' < i$  нам вообще все равно, что происходит, так как на самом первом *copy* мы точно увидим единицу.