

# Распределённые алгоритмы

mk.cs.msu.ru → Лекционные курсы → Распределённые алгоритмы

## Блок 25

Кольцевой волновой алгоритм

Лектор:

**Подымов Владислав Васильевич**

E-mail:

**valdus@yandex.ru**

Это централизованный волновой алгоритм для топологии кольца

- ▶ или для любой топологии, в которой распределённо задано кольцо, проходящее через все узлы
- ▶ т.е. **ГАМИЛЬТОНОВ ЦИКЛ**

Для начала рассмотрим **однаправленный** вариант алгоритма, предназначенный для произвольного **однаправленного** кольца

Считаем, что каждому узлу  $p$  известны узлы

- ▶  $next_p$  — следующий за  $p$  в кольце по направлению дуг
- ▶  $prev_p$  — тот, за которым следует  $p$  в кольце

Алгоритмом по кольцу передаётся **фишка** (**маркер**; **токен**; англ. token) — сообщение типа **tok**, не содержащее ничего, кроме этого типа

Основная идея алгоритма:

- ▶ Инициатор отправляет фишку
- ▶ Фишка проходит по кольцу и возвращается к инициатору
- ▶ Получив фишку, инициатор принимает решение

Код инициатора  $p$ :

1.  $\text{send}_{next_p}(\mathbf{tok})$
2.  $\text{receive}_{prev_p}(\mathbf{tok})$
3.  $\text{decide}$

Код последователя  $p$ :

1.  $\text{receive}_{prev_p}(\mathbf{tok})$
2.  $\text{send}_{next_p}(\mathbf{tok})$

**Пример** диаграммы событий вычисления для четырёх узлов:

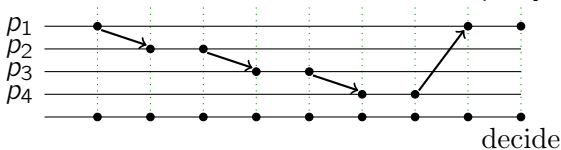
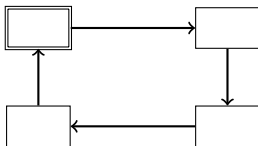


Иллюстрация этого вычисления:



**Утверждение (Д.з. 1).** Однонаправленный вариант кольцевого алгоритма является волновым

Код инициатора  $p$ :

1.  $\text{send}_{\text{next}_p}(\text{tok})$
2.  $\text{receive}_{\text{prev}_p}(\text{tok})$
3.  $\text{decide}$

Код последователя  $p$ :

1.  $\text{receive}_{\text{prev}_p}(\text{tok})$
2.  $\text{send}_{\text{next}_p}(\text{tok})$

**Пример** диаграммы событий вычисления для четырёх узлов:

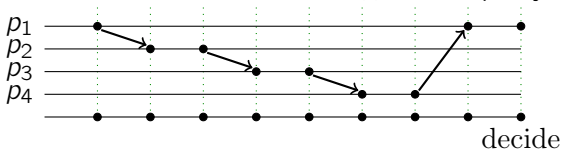
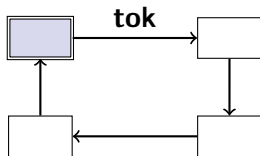


Иллюстрация этого вычисления:



**Утверждение (Д.з. 1).** Однонаправленный вариант кольцевого алгоритма является волновым

Код инициатора  $p$ :

1.  $\text{send}_{next_p}(\mathbf{tok})$
2.  $\text{receive}_{prev_p}(\mathbf{tok})$
3.  $\text{decide}$

Код последователя  $p$ :

1.  $\text{receive}_{prev_p}(\mathbf{tok})$
2.  $\text{send}_{next_p}(\mathbf{tok})$

**Пример** диаграммы событий вычисления для четырёх узлов:

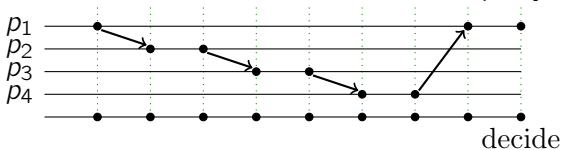
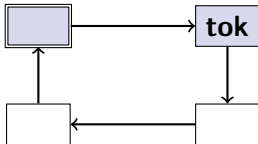


Иллюстрация этого вычисления:



**Утверждение (Д.з. 1).** Однонаправленный вариант кольцевого алгоритма является волновым

Код инициатора  $p$ :

1.  $\text{send}_{\text{next}_p}(\text{tok})$
2.  $\text{receive}_{\text{prev}_p}(\text{tok})$
3.  $\text{decide}$

Код последователя  $p$ :

1.  $\text{receive}_{\text{prev}_p}(\text{tok})$
2.  $\text{send}_{\text{next}_p}(\text{tok})$

**Пример** диаграммы событий вычисления для четырёх узлов:

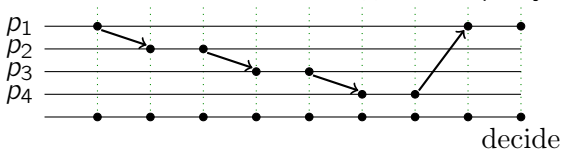
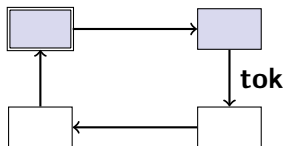


Иллюстрация этого вычисления:



**Утверждение (Д.з. 1).** Однонаправленный вариант кольцевого алгоритма является волновым

Код инициатора  $p$ :

1.  $\text{send}_{\text{next}_p}(\mathbf{tok})$
2.  $\text{receive}_{\text{prev}_p}(\mathbf{tok})$
3.  $\text{decide}$

Код последователя  $p$ :

1.  $\text{receive}_{\text{prev}_p}(\mathbf{tok})$
2.  $\text{send}_{\text{next}_p}(\mathbf{tok})$

**Пример** диаграммы событий вычисления для четырёх узлов:

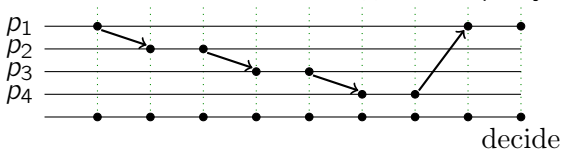
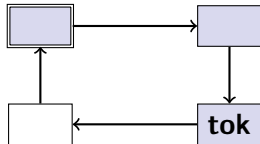


Иллюстрация этого вычисления:



**Утверждение (Д.з. 1).** Однонаправленный вариант кольцевого алгоритма является волновым

Код инициатора  $p$ :

1.  $\text{send}_{next_p}(\mathbf{tok})$
2.  $\text{receive}_{prev_p}(\mathbf{tok})$
3.  $\text{decide}$

Код последователя  $p$ :

1.  $\text{receive}_{prev_p}(\mathbf{tok})$
2.  $\text{send}_{next_p}(\mathbf{tok})$

**Пример** диаграммы событий вычисления для четырёх узлов:

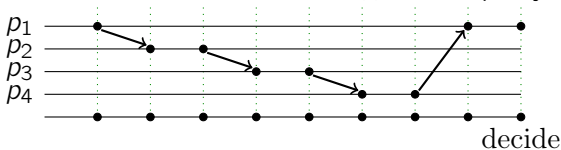
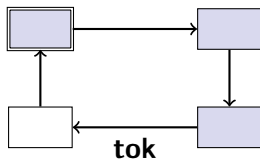


Иллюстрация этого вычисления:



**Утверждение (Д.з. 1).** Однонаправленный вариант кольцевого алгоритма является волновым



Код инициатора  $p$ :

1.  $\text{send}_{next_p}(\mathbf{tok})$
2.  $\text{receive}_{prev_p}(\mathbf{tok})$
3.  $\text{decide}$

Код последователя  $p$ :

1.  $\text{receive}_{prev_p}(\mathbf{tok})$
2.  $\text{send}_{next_p}(\mathbf{tok})$

**Пример** диаграммы событий вычисления для четырёх узлов:

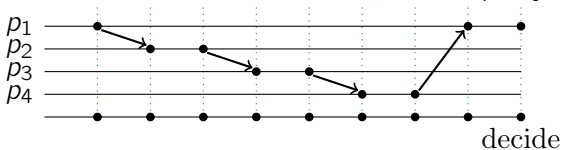
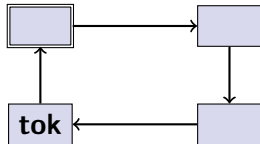


Иллюстрация этого вычисления:



**Утверждение (Д.з. 1).** Однонаправленный вариант кольцевого алгоритма является волновым

Код инициатора  $p$ :

1.  $\text{send}_{next_p}(\mathbf{tok})$
2.  $\text{receive}_{prev_p}(\mathbf{tok})$
3.  $\text{decide}$

Код последователя  $p$ :

1.  $\text{receive}_{prev_p}(\mathbf{tok})$
2.  $\text{send}_{next_p}(\mathbf{tok})$

**Пример** диаграммы событий вычисления для четырёх узлов:

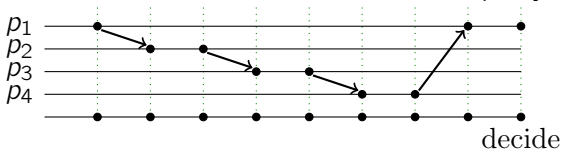
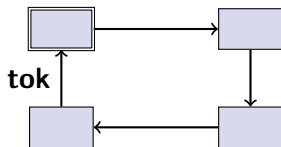


Иллюстрация этого вычисления:



**Утверждение (Д.з. 1).** Однонаправленный вариант кольцевого алгоритма является волновым

Код инициатора  $p$ :

1.  $\text{send}_{\text{next}_p}(\text{tok})$
2.  $\text{receive}_{\text{prev}_p}(\text{tok})$
3.  $\text{decide}$

Код последователя  $p$ :

1.  $\text{receive}_{\text{prev}_p}(\text{tok})$
2.  $\text{send}_{\text{next}_p}(\text{tok})$

**Пример** диаграммы событий вычисления для четырёх узлов:

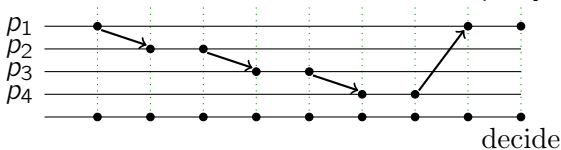
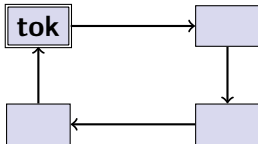


Иллюстрация этого вычисления:



**Утверждение (Д.з. 1).** Однонаправленный вариант кольцевого алгоритма является волновым

Код инициатора  $p$ :

1.  $\text{send}_{\text{next}_p}(\text{tok})$
2.  $\text{receive}_{\text{prev}_p}(\text{tok})$
3.  $\text{decide}$

Код последователя  $p$ :

1.  $\text{receive}_{\text{prev}_p}(\text{tok})$
2.  $\text{send}_{\text{next}_p}(\text{tok})$

**Пример** диаграммы событий вычисления для четырёх узлов:

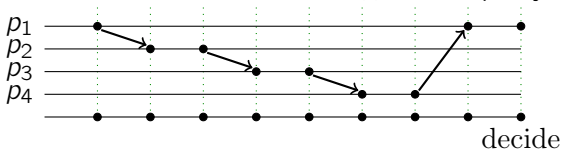
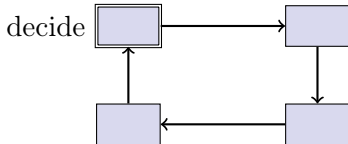


Иллюстрация этого вычисления:



**Утверждение (Д.з. 1).** Однонаправленный вариант кольцевого алгоритма является волновым

В **двунаправленном** варианте кольцевого алгоритма, в отличие от однонаправленного,

- ▶ кольцо не ориентировано (двунаправленное) и
- ▶ узел  $p$  знает соседей:  $Neigh_p$  — и то, что их двое, но не их порядок

Код инициатора  $p$ :

1. Произвольно выбрать  $q \in Neigh_p$   
(пусть  $Neigh_p \setminus \{q\} = \{r\}$ )
2.  $send_q(\mathbf{tok})$
3.  $receive_r(\mathbf{tok})$
4.  $decide$

Код последователя  $p$ :

1.  $receive_q(\mathbf{tok})$  для любого  $q \in Neigh_p$
2.  $send_r(\mathbf{tok})$ , где  $Neigh_p \setminus \{q\} = \{r\}$

Тот факт, что этот вариант алгоритма является волновым, следует из того, что вычисления этого алгоритма являются вычислениями однонаправленного варианта для одной из ориентаций кольца