

# Распределённые алгоритмы

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

## Семинар 1

Псевдокод, системы переходов и справедливость  
на примере передачи данных  
с обеспечением надёжности

Лектор:

**Подымов Владислав Васильевич**

E-mail:

**valdus@yandex.ru**

## Задача 1.1 (псевдокод и системы переходов)

Требуется **передать данные**  $d$  от узла  $A$  узлу  $B$  по каналу связи  $C$  (узлу  $A$  — **отправить данные** узлу  $B$  — **принять данные**)

Сообщения в  $C$  могут доставляться сколь угодно долго, но (если не оговорено иного) после отправки в  $C$  рано или поздно становятся доступны для чтения из  $C$  (доставляются)

До начала передачи каждый узел находится в выделенном **начальном состоянии**, не содержащем никакой информации о передаче

## Задача 1.1 (псевдокод и системы переходов)

$\text{done}_A(x)$ ; — команда узла  $A$ , аналогичная  $\text{por}$ ;, но отмечающая уверенность узла  $A$  в том, что данные доставлены

$\text{done}_B(x)$ ; — команда узла  $B$ , аналогичная  $\text{por}$ ;, но отмечающая то, что данные приняты узлом  $B$

Считается, что произошла **потеря данных**, если  $A$  уверен, что данные доставлены, а  $B$  их не принял и никогда не примет

Считается, что произошло **дублирование данных**, если  $B$  принял эти данные более одного раза

**Конечная передача** — это передача, в которой узел  $A$ , начав отправку данных, рано или поздно уверяется в доставке этих данных

**Надёжная передача** — это конечная передача, не допускающая ни потери данных, ни их дублирования

## Задача 1.1 (псевдокод и системы переходов)

Рассмотрим такую схему передачи данных  $d$ :

1.  $A$  отправляет  $d$  в  $C$  и уверяется в доставке данных
2.  $B$  получает  $d$  из  $C$  и принимает данные

### Требуется:

1. Предложить псевдокод узлов, реализующий такую схему передачи
2. Изобразить систему переходов, отвечающую этому псевдокоду
3. Обосновать надёжность передачи

## Задача 1.2 (псевдокод и системы переходов)

Положим теперь, что сообщение может быть **потеряно в канале**:

- ▶ Потерянное сообщение удаляется из канала (и никогда не будет доставлено)
- ▶ Если отправка сообщения повторяется неограниченно много раз, то рано или поздно хотя бы раз сообщение будет доставлено (и не будет потеряно)

**Требуется** показать конечность и ненадёжность передачи, организованной согласно коду из (1.1)

## Задача 2.1 (справедливость)

Добавим в псевдокод булево выражение `await( $v_1, \dots, v_n$ )`:

- ▶ Узел блокируется, пока не пройдёт некоторое время (далее обозначаемое как «перерыв») или пока не будет доставлено сообщение, которое можно принять выполнением команды `receive( $v_1, \dots, v_n$ )`
- ▶ Если такое сообщение появилось, то выражение вычисляется со значением  $\dagger$
- ▶ Если перерыв закончился и сообщение не появилось, то выражение вычисляется со значением  $\text{ff}$

## Задача 2.1 (справедливость)

Чтобы преодолеть ненадёжность, упомянутую в (1.2), попробуем организовать передачу так:

- ▶  $A$  многократно с перерывами отправляет  $d$  в  $C$ , пока не получит отклик от  $B$  об получении соответствующего сообщения
- ▶  $B$  получает  $d$  из  $C$ , отправляет в  $C$  отклик и принимает данные
- ▶  $A$ , получив из  $C$  отклик от  $B$ , уверяется в доставке данных

### Требуется:

- ▶ Предложить псевдокод узлов, реализующий такую схему передачи
- ▶ Изобразить систему переходов, отвечающую этому псевдокоду
- ▶ Показать ненадёжность такой передачи

## Задача 2.2 (справедливость)

Усложним задачу, попытавшись избавиться от ненадёжности в (2.1):

- ▶ Требуется передать не один блок данных  $d$ , а неизвестное узлам произвольное число блоков  $d_1, \dots, d_n$
- ▶ Узел  $A$  последовательно выполняет свой код **однократной** отправки данных для каждого передающегося блока
- ▶ Узел  $B$  после завершения кода **однократного** приёма данных переходит в **начальное состояние** — т.е. «забывает всё» и ещё раз начинает приём

Положим, что однократные отправка и приём устроены согласно коду из (2.1), и что передаётся один блок данных

**Требуется** показать, что в таких условиях передача конечна и невозможна потеря данных, но возможно их дублирование



## Задача 2.3 (справедливость)

Положим, что отправка и приём данных устроены согласно (2.2), но передаются два блока данных

**Требуется** показать, что в таких условиях передача конечна и возможна потеря данных

## Задача 3.1

Чтобы преодолеть дублирование данных из (2.2), попробуем организовать однократные отправку и приём данных  $d$  так:

- ▶  $A$  многократно с перерывами отправляет  $d$  в  $C$ , пока не получит отклик от  $B$  об получении соответствующего сообщения
- ▶  $B$  получает  $d$  из  $C$  и многократно с перерывами отправляет в  $C$  отклик, пока не получит из  $C$  подтверждение отклика
- ▶  $A$ , получив из  $C$  отклик от  $B$ , отправляет в  $C$  подтверждение отклика и уверяется в доставке данных
- ▶  $B$ , получив из  $C$  подтверждение отклика, принимает данные

### Требуется:

- ▶ Предложить псевдокод узлов, реализующий однократную передачу по этой схеме
- ▶ Показать, что при передаче одного блока данных передача конечна и невозможно дублирование данных, но возможна их потеря

## Задача 3.2

Чтобы преодолеть потерю данных из (3.1), попробуем добавить в каждый узел  $X$  идентификатор сеанса  $s_X$ :

- ▶ Это переменная, способная принимать неограниченно много разных значений
- ▶ В начале выполнения и после завершения каждой однократной передачи данных в  $s_X$  присваивается новое уникальное значение, известное только узлу  $X$
- ▶ В сообщения между узлами добавляются значения идентификаторов сеанса для различения сообщений текущего сеанса передачи и остальных
- ▶ При доставке сообщения, не относящегося к текущему сеансу, оно принимается, помечается как ошибочное и отправляется обратно
- ▶ Доставка ошибочного сообщения может повлиять на поток управления узла-адресата

## Задача 3.2

### Требуется:

1. Изменить код решения (3.1), добавив в него идентификаторы сеансов, и описать «разумную» обработку ошибочных сообщений, в совокупности обеспечивающие надёжную передачу одного блока данных
2. Обосновать надёжность такой передачи

## Задача 3.3

**Требуется** проанализировать надёжность передачи двух блоков данных согласно способу, получившемуся в (3.2)

## Задача 4

### Требуется:

- ▶ Добавить в решение (2.3) идентификаторы сеансов и обработку ошибочных сообщений так, чтобы была устранена указанная там потеря данных (хотя, быть может, и появилось дублирование данных) и сохранилась конечность передачи
  
- ▶ Обосновать свойства получившейся передачи

## Задача 5 (трудная)

**Требуется** предложить конечный способ передачи одного блока данных без потери (хотя и с возможным дублированием) по каналу, допускающему потерю сообщений, учитывающий возможный выход из строя узла  $B$  — принудительный переход к началу приёма с новым сеансом (если используются идентификаторы сеансов) вместо выполнения следующей команды однократного приёма