

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Департамент программной инженерии  
Дисциплина: «Архитектура вычислительных систем»

**Микропроект 2**

**Задача об обедающих философах**

**ВАРИАНТ 4**

Отчет

Листов 7

**Выполнил:**  
Асатрян Эмин,  
студент гр. БПИ198

**Москва**  
2020

## СОДЕРЖАНИЕ

<b>1. ТЕКСТ ЗАДАНИЯ .....</b>	<b>3</b>
<b>2. ОПИСАНИЕ ВХОДНЫХ ДАННЫХ .....</b>	<b>4</b>
<b>3. ОПИСАНИЕ ВЫХОДНЫХ ДАННЫХ .....</b>	<b>5</b>
<b>4. ОПИСАНИЕ ПРИМЕНЯЕМЫХ РАССЧЕТНЫХ МЕТОДОВ .....</b>	<b>6</b>
<b>4.1. Теория решения основной задачи .....</b>	<b>6</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....</b>	<b>7</b>

## 1. ТЕКСТ ЗАДАНИЯ

Пять философов сидят возле круглого стола. Они проводят жизнь, чередуя приемы пищи и размышления. В центре стола находится большое блюдо спагетти. Спагетти длинные и запутанные, философам тяжело управляться с ними, поэтому каждый из них, чтобы съесть порцию, должен пользоваться двумя вилками. К несчастью, философам дали только пять вилок. Между каждой парой философов лежит одна вилка, поэтому эти высококультурные и предельно вежливые люди договорились, что каждый будет пользоваться только теми вилками, которые лежат рядом с ним (слева и справа). Написать многопоточную программу, моделирующую поведение философов с помощью семафоров. Программа должна избегать фатальной ситуации, в которой все философы голодны, но ни один из них не может взять обе вилки (например, каждый из философов держит по одной вилке и не хочет отдавать ее). Решение должно быть симметричным, то есть все потоки-философы должны выполнять один и тот же код.

## **2. ОПИСАНИЕ ВХОДНЫХ ДАННЫХ**

Данная программа не принимает на вход никакие данные.

### **3. ОПИСАНИЕ ВЫХОДНЫХ ДАННЫХ**

На протяжении работы, программа выводит сообщения о действиях, совершаемых каждым философом (размышление, изъявление желания взять вилку, удачное взятие вилки, успешный прием пищи). После достаточного (заранее определенного) количества приемов пищи, совершенных каждым философом, программа выводит сообщение об удачном завершении программы без появления фатальных ситуаций.

## 4. ОПИСАНИЕ ПРИМЕНЯЕМЫХ РАССЧЕТНЫХ МЕТОДОВ

### 4.1. Теория решения основной задачи

Самое очевидное решение задачи представимо следующим алгоритмом, исполняемым каждым философом:

- Ждать пока освободиться левая вилка, взять ее при освобождении
- Ждать пока освободиться правая вилка, взять ее при освобождении
- Есть
- Положить левую вилку
- Положить правую вилку

Однако данное решение имеет потенциальную фатальную ситуацию – например, если все 5 философов возьмут левую вилку. В таком случае, для каждого философа будет занята его правая вилка, и каждый философ будет бесконечно ждать, пока она освободиться.

Решить данную проблему можно, присвоив частичный порядок вилкам. Поскольку наши вилки пронумерованы от 0 до 4, то можно использовать их индексы в качестве порядка. Философ будет брать вилку с наименьшим номером, и только потом с наибольшим номером, а при высвобождении вилок, будет сначала класть вилку с наибольшим номером, а затем с наименьшим, позволив другому философу приступить к еде. Таким образом, получается, что первые четыре философа берут сначала левую вилку, а затем правую, а последний по индексу философ, так как его вилка слева имеет индекс 4, а правая – 0, будет сначала брать правую, а только потом уже левую. Благодаря этому, в случае когда, например, первые 4 философа возьмут левую вилку, 5ый философ будет ожидать освобождения своей правой вилки (которая занята) вместо взятия левой, тем самым не создавая фатальную ситуацию.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. std::counting\_semaphore, std::binary\_semaphore [Электронный ресурс] – [https://en.cppreference.com/w/cpp/thread/counting\\_semaphore](https://en.cppreference.com/w/cpp/thread/counting_semaphore) (дата обращения: 12.12.20).
2. Архитектура параллельных вычислительных систем. Многопоточность [Электронный ресурс] – URL: <http://softcraft.ru/edu/comparch/lect/07-parthread/> (дата обращения: 10.12.20).
3. Задача об обедающих философах [Электронный ресурс] – URL: [https://ru.wikipedia.org/wiki/Задача\\_об\\_обедающих\\_философах](https://ru.wikipedia.org/wiki/Задача_об_обедающих_философах) (дата обращения: 12.12.20).
4. Многопоточное программирование. Простые программы [Электронный ресурс] – URL: <http://softcraft.ru/edu/comparch/practice/thread/01-simple/> (дата обращения: 10.12.20).
5. Поток выполнения [Электронный ресурс] – URL: [https://ru.wikipedia.org/wiki/Поток\\_выполнения](https://ru.wikipedia.org/wiki/Поток_выполнения) (дата обращения: 10.12.20).