

Национальный исследовательский университет "Высшая школа экономики".  
Факультет компьютерных наук.

Пояснительная записка к микропроекту №2  
по предмету "Архитектура вычислительных систем".

Выполнила студентка 2 курса  
группы БПИ192  
Цимбалистая Ульяна Игоревна

13 декабря 2020 г.

## **Содержание**

1. Текст задания
2. Описание решения
3. Текст программы
4. Примеры выполнения работы

## 1. Текст задания

Вариант 4: *Задача об обедающих философах*. 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 решения

Задача о обедающих философов - проблема синхронизации, которую сформулировал Эдсгер Дейкстра. Им же было представлено решение, когда философы берут сначала левые вилки, а потом (при условии их не занятости) правые и только последний философ берет сначала правую вилку, а потом левую. Однако данный алгоритм не является симметричным, так как философ 5 выполняет другой код.

Для решения данной задачи применим алгоритм предложенный Э. Таненбаумом. Пять философов сидят за обеденным столом и у каждого есть три возможных состояния: ест, размышляет, голоден. Для блокировки секции доступа к вилке используется семафор. Также используется массив из семафоров для блокировки философов, если вилки рядом заняты.

В начале философы размышляют, затем пытаются взять обе вилки возле себя одновременно, их состояние меняется на "HUNGRY". Если философ получает доступ к вилкам, то выполняются условия: он голоден, философ слева не ест, и философ справа не ест, то его состояние меняется на "EATING", иначе он блокируется пока не освободятся вилки. После того, как философ поел, он кладет вилки и размышляет. Действия происходят циклически. Взятие вилок и их возврат происходят в критической секции. Данный алгоритм помогает избежать ситуация блокировки потоков, а также в один момент времени могут обедать два философа.

При запуске программы пользователь должен ввести желаемое время выполнения программы от 20 секунд до 2 минут. При верно введенном параметре происходит инициализация семафоров и запуск потоков для выполнения метода действий философа. После этого основной поток засыпает на заданное время, по истечении которого сообщается, что философы закончили собрание и программа завершается. При неверно введенном параметре он будет запрашиваться пока не будет введено корректное число. (рис. 2).

### 3. Текст программы

```
1. #include <unistd.h>
2. #include <stdlib.h>
3. #include <stdio.h>
4. #include <pthread.h>
5. #include <semaphore.h>
6. #include <iostream>
7.
8. // Состояния философов.
9. const int THINK = 0;
10. const int HUNGRY = 1;
11. const int EATING = 2;
12.
13. const int N = 5;           // Число философов.
14. int state[N];             // Массив состояний философов.
15. int left;                 // Философ слева.
16. int right;                // Философ справа.
17. int eatingTime;           // Время на еду.
18. int thinkingTime;         // Время на размышления.
19.
20. sem_t semaphore = 1;      // Блокировка для критических секций.
21. sem_t sem[N];             // Семафоры каждого из философов.
22.
23. /**
24.  * Философ размышляет в течение 1 - 5 секунд.
25.  * @param i Номер философа.
26.  */
27. void think(int i) {
28.     state[i] = THINK;
29.     printf("Philosopher %d now thinks about nature of existence\n", i);
30.
31.     thinkingTime = rand() % 6 + 2;
32.     sleep(thinkingTime);
33. }
34.
35. /**
36.  * Философ ест в течение 1 - 5 секунд.
37.  * @param i Номер философа.
38.  */
39. void eat(int i) {
40.     if (state[i] == EATING){
41.         printf("Philosopher %d took forks and now consumes spaghetti instead of spiritual
42.                                     nourishment\n", i);
43.
44.         eatingTime = rand() % 6 + 2;
45.         sleep(eatingTime);
46.     }
47. }
48. /**
49.  * Проверка свободны ли правая и левая вилка у философа, если да, то он ест.
50.  * @param i Номер философа.
51.  */
52. void checkForks(int i) {
53.     left = (i + N - 1) % N;
54.     right = (i + 1) % N;
55.     if (state[i] == HUNGRY && state[left] != EATING && state[right] != EATING) {
56.         state[i] = EATING;
57.         sem_post(&sem[i]); // Разблокируем философа.
58.     }
59. }
60.
61. /**
62.  * Процесс взятия вилок философom.
63.  * @param i Номер философа.
64.  */
65. void takeForks(int i) {
66.     sem_wait(&semaphore); // Блок секции
67.     state[i] = HUNGRY;    // Философ хочет есть
68.     checkForks(i);        // Проверяет наличие вилок
69.     sem_post(&semaphore); // Разблок секции
70.
71.     sem_wait(&sem[i]);     // Если вилок нет, то философ блокируется
72. }
```

```

73.
74. /**
75.  * Философ кладет вилки на стол, философы справа и слева могут начать есть, если обе вилки
    свободны.
76.  * @param i Номер философа.
77.  */
78. void putForks(int i) {
79.     sem_wait(&semaphore); // Блок секции
80.     state[i] = THINK; // После еды философ размышляет
81.
82.     // Левый и правый проверяют вилки
83.     left = (i + N - 1) % N;
84.     right = (i + 1) % N;
85.     checkForks(left);
86.     checkForks(right);
87.
88.     sem_post(&semaphore); // Разблок секции
89. }
90.
91. /**
92.  * Представляет собой работу мозга философа за столом.
93.  * @param i Номер философа.
94.  */
95. void *philosopher(void *i_) {
96.     auto i = (uint64_t) i_;
97.     while (true) {
98.         think(i);
99.         takeForks(i);
100.        eat(i);
101.        putForks(i);
102.    }
103.}
104.
105. /**
106.  * Считывает и проверяет время работы программы введенное пользователем.
107.  * @return Время в секундах.
108.  */
109. int getTime() {
110.     while (true) {
111.         std::cout << "\nHow long philosophers are going to have dinner? (20 - 120 seconds)\n";
112.
113.         int k;
114.         std::cin >> k;
115.
116.         if (std::cin.fail() || k < 20 || k > 120) {
117.             std::cin.clear();
118.             std::cin.ignore(32767, '\n');
119.         } else {
120.             return k;
121.         }
122.     }
123.}
124.
125. /**
126.  * Инициализация семафоров и создание потоков.
127.  */
128. void semaphoreInit() {
129.     sem_init(&semaphore, 0, 1); // Задаем семафор для блокировок (его начальное значение 1)
130.
131.     // Задаем семафоры для философов (начальное значение 0)
132.     for (int i = 0; i < N; ++i) {
133.         sem_init(&sem[i], 0, 0);
134.     }
135.
136.     // Создаем потоки
137.     pthread_t thread[N];
138.     for (int i = 0; i < N; ++i){
139.         pthread_create(&thread[i], nullptr, philosopher, reinterpret_cast<void *>(i));
140.     }
141.}
142.
143. int main() {
144.     srand(time(0));
145.
146.     std::cout << "Five philosophers gathered at a round table to reflect and consume pasta.\n"
147.                 "However they only have 5 forks and need 2 for person to eat spaghetti.\n";
148.

```

```

149. // Начиаем выполнение работы
150. int t = getTime();
151. semaphoreInit();
152.
153. // Основной поток засыпает на заданное время.
154. sleep(t);
155. std::cout << "Philosophers finished dinning." << std::endl;
156.
157. return 0;
158. }

```

#### 4. Примеры выполнения программы

```

/Users/ulyanatsimbalistaya/CLionProjects/homeworks/tsimbalistaya_mp2/cmake-build-debug/tsimbalistaya_mp2
Five philosophers gathered at a round table to reflect and consume pasta.
However they only have 5 forks and need 2 for person to eat spaghetti.

How long philosophers are going to have dinner? (20 - 120 seconds)
20
Philosopher 0 now thinks about nature of existence
Philosopher 1 now thinks about nature of existence
Philosopher 2 now thinks about nature of existence
Philosopher 3 now thinks about nature of existence
Philosopher 4 now thinks about nature of existence
Philosopher 2 took forks and now consumes spaghetti instead of spiritual nourishment
Philosopher 4 took forks and now consumes spaghetti instead of spiritual nourishment
Philosopher 1 now thinks about nature of existence
Philosopher 3 now thinks about nature of existence
Philosopher 0 now thinks about nature of existence
Philosopher 0 now thinks about nature of existence
Philosopher 4 now thinks about nature of existence
Philosopher 2 now thinks about nature of existence
Philosopher 1 took forks and now consumes spaghetti instead of spiritual nourishment
Philosopher 0 now thinks about nature of existence
Philosopher 2 now thinks about nature of existence
Philosopher 3 took forks and now consumes spaghetti instead of spiritual nourishment
Philosopher 2 now thinks about nature of existence
Philosopher 1 now thinks about nature of existence
Philosopher 0 took forks and now consumes spaghetti instead of spiritual nourishment
Philosopher 4 now thinks about nature of existence
Philosopher 2 now thinks about nature of existence
Philosopher 3 now thinks about nature of existence
Philosopher 1 now thinks about nature of existence
Philosopher 2 took forks and now consumes spaghetti instead of spiritual nourishment
Philosophers finished dinning.

Process finished with exit code 0

```

Рис. 1 Выполнение при верном входном параметре.

```

/Users/ulyanatsimbalistaya/CLionProjects/homeworks/tsimbalistaya_mp2/cmake-build-debug/tsimbalistaya_mp2
Five philosophers gathered at a round table to reflect and consume pasta.
However they only have 5 forks and need 2 for person to eat spaghetti.

How long philosophers are going to have dinner? (20 - 120 seconds)
4
How long philosophers are going to have dinner? (20 - 120 seconds)
sss
How long philosophers are going to have dinner? (20 - 120 seconds)
444
How long philosophers are going to have dinner? (20 - 120 seconds)
-34
How long philosophers are going to have dinner? (20 - 120 seconds)
21rrrrr
Philosopher 0 now thinks about nature of existence
Philosopher 1 now thinks about nature of existence
Philosopher 2 now thinks about nature of existence
Philosopher 4 now thinks about nature of existence
Philosopher 3 now thinks about nature of existence
Philosopher 4 took forks and now consumes spaghetti instead of spiritual nourishment
Philosopher 3 now thinks about nature of existence
Philosopher 4 now thinks about nature of existence
Philosopher 1 took forks and now consumes spaghetti instead of spiritual nourishment
Philosopher 3 took forks and now consumes spaghetti instead of spiritual nourishment
Philosopher 2 now thinks about nature of existence
Philosopher 0 now thinks about nature of existence
Philosopher 4 now thinks about nature of existence
Philosopher 1 now thinks about nature of existence
Philosopher 0 took forks and now consumes spaghetti instead of spiritual nourishment
Philosopher 3 now thinks about nature of existence
Philosopher 0 now thinks about nature of existence
Philosopher 2 took forks and now consumes spaghetti instead of spiritual nourishment
Philosopher 4 took forks and now consumes spaghetti instead of spiritual nourishment

```

Рис. 2 Реакция на неверный входной параметр.