



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 4
по курсу «Разработка параллельных и распределенных
программ»

Задач о пяти обедающих философях

Студент: Пишикина М.В.

Группа: ИУ9-51Б

Преподаватель: Царев А.С.

Москва 2024

Содержание

1	Постановка задачи	3
2	Практическая реализация	4
3	Характеристика устройства	6
4	Работа программы	6

1 Постановка задачи

Суть задачи следующая. Пять философов сидят за круглым столом. Они проводят жизнь, чередуя приёмы пищи и размышления. В центра стола находится большое блюдо спагетти. Чтобы съесть порцию, каждому философу нужно две вилки. Однако, вилок всего пять: между каждой парой рядом сидящих философов лежат по одной вилке, и каждый философ может пользоваться только теми вилками, которые лежат рядом с ним, слева и справа. Философ не может брать две вилки одновременно: сначала он тратит некоторое время на то, чтобы взять одну, затем вторую. Однако, он может одновременно положить их на место. Задача заключается в том, чтобы написать программу, моделирующую поведение философов. Очевидно, что раз вилок всего пять, то одновременно есть могут не более двух философов, и два сидящих рядом философа не могут есть одновременно. Для имитации периодов раздумий и приёмов пищи можно использовать генератор случайных чисел, позволяющий задавать времена их действий в определённом интервале. Имитация поведения каждого философа, по сути, разбивается на то, что в любой момент времени философ находится в одном из пяти состояний: размышляет, берёт левую вилку, берёт правую вилку, ест, кладёт вилки на место. Таким образом, вилки являются разделяемым ресурсом. На программу накладываются условия:

1. Каждый философ, по сути, является потоком, и модель поведения у каждого из них должна быть одинаковой, кроме того, какие вилки они могут брать.
 2. Накладывание блокировки по сути является действием по взятию вилки, поэтому накладывать блокировку сразу на обе вилки нельзя; последовательность действий должна быть «наложить блокировку – взять вилку – наложить вторую блокировку – взять вторую вилку».
 3. Программа должна избегать ситуации взаимоблокировки: ситуации, в которой все философы голодны, то есть ни один из них не может взять себе две вилки (например, когда каждый держит по одной и не хочет её отдавать).
- Запрограммировать остановку алгоритма по достижении контрольного времени (например, атомарной операцией над булевым флагом).

2 Практическая реализация

```
import threading
import time
import random

class Philosopher(threading.Thread):
    def __init__(self, index, left_fork, right_fork, stop_event, log):
        super().__init__()
        self.index = index
        self.left_fork = left_fork
        self.right_fork = right_fork
        self.stop_event = stop_event    # флаг остановки
        self.log = log

    def run(self):
        while not self.stop_event.is_set():
            self.think()
            self.eat()

    def log_state(self, state):
        timestamp = time.time()
        self.log.append((timestamp, self.index + 1, state))

    def think(self):
        self.log_state("размышляет")
        time.sleep(random.uniform(0.5, 1.5))

# Про взаимоблокировку: чет философы начинают с левой вилки, нечет
↪ - с правой
def eat(self):
    right_fork, left_fork = (self.left_fork, self.right_fork) if self.index % 2
    ↪ == 0 else (self.right_fork, self.left_fork)
```

```

        with right_fork:
            self.log_state("берёт правую вилку")
            time.sleep(0.1)

        with left_fork:
            self.log_state("берёт левую вилку")
            self.log_state("ест")
            time.sleep(random.uniform(0.5, 1.0))

        self.log_state("кладёт вилки на место")

def main(philosopher_count=5, run_time=10,
↪   output_file="philosophers_log.txt"):
    forks = [threading.Lock() for _ in range(philosopher_count)]
    stop_event = threading.Event()
    log = []

    # Каждый философ - это поток
    philosophers = [
        Philosopher(i, forks[i], forks[(i + 1) % philosopher_count], stop_event, log)
        for i in range(philosopher_count)
    ]

    # Запуск потоков
    for p in philosophers:
        p.start()

    time.sleep(run_time)
    stop_event.set()

    # Ждем завершения всех потоков
    for p in philosophers:

```

```

p.join()

# Запись логов в файл
with open(output_file, "w") as f:
    f.write("Время, Философы, Состояние\n")
    for entry in log:
        timestamp, philosopher, state = entry
        formatted_time = time.strftime("%Y-%m-%d %H:%M:%S",
            ↪ time.localtime(timestamp))
        f.write(f"{formatted_time}, Философ {philosopher}, {state}\n")

main()

```

3 Характеристика устройства

Устройство: MacBook Pro 2020

Операционная система: macOS Sonoma

Процессор: Intel Core i5

Характеристика процессора: 4-ядерный чип, частота 2 ГГц

Оперативная память: 16GB LPDDR4X

4 Работа программы

Время, Философы, Состояние

2024-11-22 22:45:31, Философ 1, размышляет

2024-11-22 22:45:31, Философ 2, размышляет

2024-11-22 22:45:31, Философ 3, размышляет

2024-11-22 22:45:31, Философ 4, размышляет

2024-11-22 22:45:31, Философ 5, размышляет

2024-11-22 22:45:31, Философ 1, берёт правую вилку

2024-11-22 22:45:31, Философ 1, берёт левую вилку

2024-11-22 22:45:31, Философ 1, ест
2024-11-22 22:45:31, Философ 5, берёт правую вилку
2024-11-22 22:45:32, Философ 2, берёт правую вилку
2024-11-22 22:45:32, Философ 1, кладёт вилки на место
2024-11-22 22:45:32, Философ 1, размышляет
2024-11-22 22:45:32, Философ 2, берёт левую вилку
2024-11-22 22:45:32, Философ 2, ест
2024-11-22 22:45:32, Философ 5, берёт левую вилку
2024-11-22 22:45:32, Философ 5, ест
...
2024-11-22 22:45:41, Философ 5, берёт правую вилку
2024-11-22 22:45:41, Философ 1, кладёт вилки на место
2024-11-22 22:45:41, Философ 5, берёт левую вилку
2024-11-22 22:45:41, Философ 5, ест
2024-11-22 22:45:42, Философ 3, кладёт вилки на место
2024-11-22 22:45:42, Философ 5, кладёт вилки на место