Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

Московский государственный технический университет имени Н.Э.Баумана
(МГТУ им. Н.Э.Баумана)

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

«Задача о пяти обедающих философах»

Выполнила: Митрошкин А.А.
(Фамилия И.О. студента)

ИУ9-51Б
(Индекс группы)

Преподаватель: Царев.А. С.
(Фамилия И.О. преподавателя)

(Подпись)

Москва, 2023

Оглавление

1. Цель работы

   Написать программу, моделирующую поведение философов из условия задачи

2. Условия 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). В отчёте построить некоторый результат работы алгоритма, которая может быть в виде графика, таблицы, лога или чего угодно ещё; главное условие состоит в том, чтобы по результатам можно было однозначно определить, чем в каждый момент времени был занят каждый философ (одно из пяти состояний). Также рассмотреть вариант программы с увеличением количества философов до произвольного N. На основании полученных

результатов сделать вывод о целесообразности использования одного или второго варианта программы

3. Код решения

```
import threading
import time
import random

class Philosopher(threading.Thread):
    def __init__(self, name, left_fork, right_fork, stop_event):
        super().__init__(name=name)
        self.left_fork = left_fork
        self.right_fork = right_fork
        self.stop_event = stop_event

    '''
    Метод run представляет действия философа, выполняемые в отдельном потоке.
    Философ поочередно размышляет и обедает, пока не поступит сигнал остановки.
    '''
    def run(self):
        while not self.stop_event.is_set():
            self.think()
            self.dine()

    # Пауза
    '''
    Метод think представляет процесс размышления философа.
    Выводится сообщение и выполняется случайная пауза.
    '''
    def think(self):
        print(f"{self.name} Размышляет.")
        time.sleep(random.uniform(1, 3))

    # Обед. Берем левую и правую вилку, а потом кладем на место
    '''
      Метод dine представляет процесс обеда философа.
      Философ пытается взять левую вилку и, если успешно, пытается взять правую.
      Если обе вилки взяты, философ ест, а затем кладет вилки на место.
    '''
    def dine(self):
        print(f"{self.name} голоден и пытается взять вилки")
        acquired_left = False
        acquired_right = False

```

```
        while not acquired_left and not self.stop_event.is_set():
            acquired_left = self.left_fork.acquire(timeout=random.randint(1, 3))

        if acquired_left:
            print(f"{self.name} взял левую вилку.")
            # Попытка взять правую вилку
            while not acquired_right and not self.stop_event.is_set():
                acquired_right = self.right_fork.acquire(timeout=1)

            if acquired_right:
                print(f"{self.name} взял правую вилку и ест.")
                time.sleep(random.uniform(1, 3))
                print(f"{self.name} опустить правую вилку.")
                self.right_fork.release()

            print(f"{self.name} опустить левую вилку.")
            self.left_fork.release()

def main():
    num_philosophers = 10
    forks = [threading.Lock() for _ in range(num_philosophers)] # Список мьютексов
    stop_event = threading.Event()

    philosophers = [Philosopher(f"Философ {i}", forks[i], forks[(i + 1) % num_philosophers], stop_event) for i in
                    range(num_philosophers)] # Потоки

    try:
        for philosopher in philosophers:
            philosopher.start()

        time.sleep(10)  # Программа работает 10 секунд

    finally:
        stop_event.set()
        for philosopher in philosophers:
            philosopher.join()

if __name__ == "__main__":
    main()

```

4. Результаты

```
PS C:\Users\Alex\Desktop\bmstu_labs\parallel_programming\lab4> python lab4.py
Философ 0 Размышляет.
Философ 1 Размышляет.
Философ 2 Размышляет.
Философ 3 Размышляет.
Философ 4 Размышляет.
Философ 5 Размышляет.
Философ 6 Размышляет.
Философ 7 Размышляет.
Философ 8 Размышляет.
Философ 9 Размышляет.
Философ 5 голоден и пытается взять вилки
Философ 5 взял левую вилку.
Философ 5 взял правую вилку и ест.
Философ 8 голоден и пытается взять вилки
Философ 8 взял левую вилку.
Философ 8 взял правую вилку и ест.
Философ 7 голоден и пытается взять вилки
Философ 7 взял левую вилку.
Философ 3 голоден и пытается взять вилки
Философ 3 взял левую вилку.
Философ 3 взял правую вилку и ест.
Философ 4 голоден и пытается взять вилки
Философ 2 голоден и пытается взять вилки
Философ 2 взял левую вилку.
Философ 9 голоден и пытается взять вилки
Философ 0 голоден и пытается взять вилки
Философ 0 взял левую вилку.
Философ 0 взял правую вилку и ест.
Философ 6 голоден и пытается взять вилки
Философ 8 опустить правую вилку.
Философ 8 опустить левую вилку.
Философ 8 Размышляет.
```

Характеристики компьютера

12th Gen Intel(R) Core(TM) i5-12450H 2.50 GHz
16,0 ГБ (доступно: 15,7 ГБ)
64-разрядная операционная система, процессор x64

Итог: Задача реализована