



Министерство науки и высшего образования Российской Федерации  
Калужский филиал  
федерального государственного бюджетного  
образовательного учреждения высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК2 "Информационные системы и сети"

## ЛАБОРАТОРНАЯ РАБОТА №5

«МНОГОПОТОЧНОЕ ПРОГРАММИРОВАНИЕ НА PYTHON »

ДИСЦИПЛИНА: «Перспективные языки программирования»

Выполнил: студент гр. ИУК4-33Б \_\_\_\_\_ (\_\_\_\_ Сароян Н.А.\_\_\_\_)  
(Подпись) (Ф.И.О.)

Проверил: \_\_\_\_\_ (\_\_\_\_ Осипова О.В.\_\_\_\_)  
(Подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2024

**Цель:** Целью выполнения лабораторной работы является формирование практических навыков многопоточного программирования, разработки и отладки программ, овладение методами и средствами разработки.

**Задачи:** Основными задачами выполнения лабораторной работы являются:

1. Изучить особенности создания потоков и процессов;
2. Научиться создавать многопоточные программы;
3. Изучить типовые алгоритмы решения задач с использованием принципов многопоточного программирования;

Результатами работы являются:

1. Реализация разработанных алгоритмов на языке программирования Python;
2. Подготовленный отчет;

**Задание:**

## Вариант 8

### Задание 1.

Статистический анализ. Имеется несколько массивов данных (разного размера). Требуется определить максимальное значение в каждом массиве, без использования встроенной функции. Обработка каждого массива выполняется в отдельном процессе (потоке).

### Задание 2.

Разработайте многопоточное приложение, выполняющее вычисление произведения матриц  $A (m \times n)$  и  $B (n \times k)$ . Элементы  $c_{ij}$  матрицы произведения  $C = A \times B$  вычисляются параллельно  $p$  однотипными потоками. Если некоторый поток уже вычисляет элемент  $c_{ij}$  матрицы  $C$ , то следующий приступающий к вычислению поток выбирает для расчета элемент  $c_{i,j+1}$ , если  $j < k$ , и  $c_{i+1,k}$ , если  $j = k$ . Выполнив вычисление элемента матрицы-произведения, поток проверяет, нет ли элемента, который еще не рассчитывается. Если такой элемент есть, то приступает к его расчету. В противном случае отправляет (пользовательское) сообщение о завершении своей работы и приостанавливает своё выполнение. Главный поток, получив сообщения о завершении вычислений от всех потоков, выводит результат на экран и запускает поток, записывающий результат в конец файла-протокола. В каждом потоке должна быть задержка в выполнении вычислений (чтобы дать возможность поработать всем потокам). Обеспечить синхронизацию потоков

## Листинг программы задания 1:

```
import multiprocessing

def find_max(array, result, index):
    max_value = array[0]
    for value in array:
        if value > max_value:
            max_value = value
    result[index] = max_value

def main():
    arrays = [
        [1, 5, 3, 9, 2],
        [4, 7, 1, 8],
        [10, 15, 3, 7, 9]
    ]

    processes = []
    manager = multiprocessing.Manager()
    result = manager.list(range(len(arrays)))

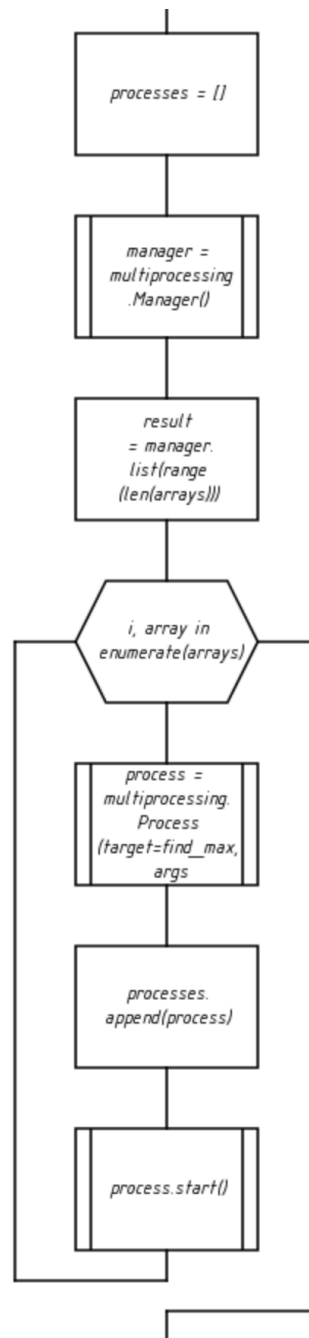
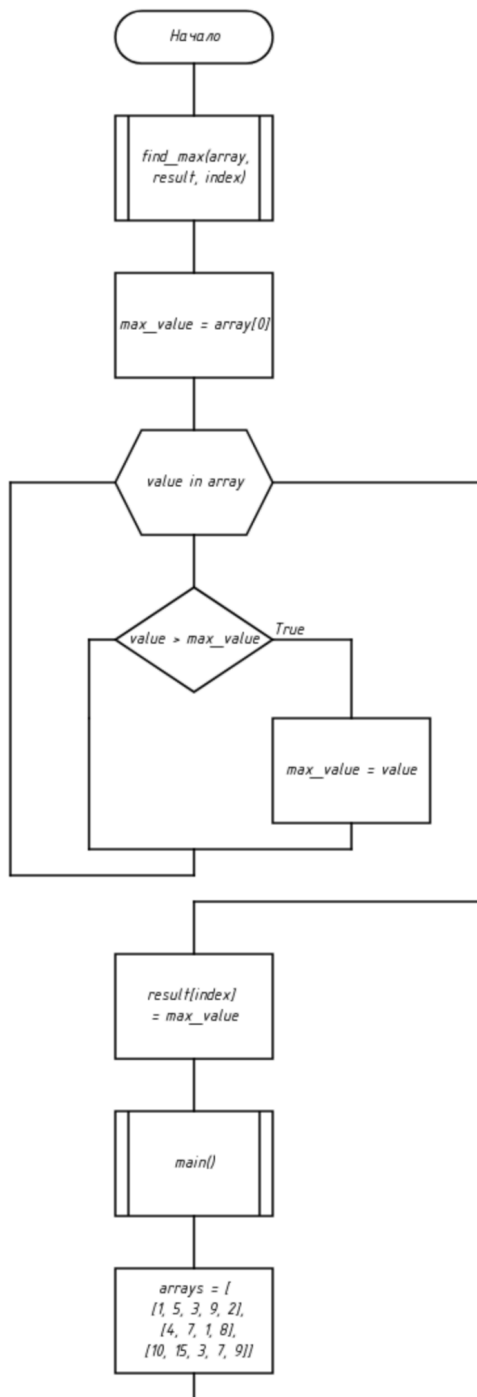
    for i, array in enumerate(arrays):
        process = multiprocessing.Process(target=find_max, args=(array, result,
i))
        processes.append(process)
        process.start()

    for process in processes:
        process.join()

    print("Максимальные значения в каждом массиве:", list(result))

if __name__ == "__main__":
    main()
```

## Блок схема:





## Результаты выполнения программы

```
/Users/noriksaroyan/PycharmProjects/PPL_HW1/.venv/bin/python /Users/noriksaroyan/PycharmProjects/PPL_HW1/lab5/Task1.py
Максимальные значения в каждом массиве: [9, 8, 15]

Process finished with exit code 0
```

## Листинг программы задания 2.

```
import threading
import time

# Матрицы A и B
A = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]

B = [
    [9, 8, 7],
    [6, 5, 4],
    [3, 2, 1]
]

m = len(A)
n = len(A[0])
k = len(B[0])

# Матрица результата C
C = [[0 for _ in range(k)] for _ in range(m)]

# Мьютекс для синхронизации потоков
lock = threading.Lock()
current_element = [0, 0] # [i, j] текущего элемента для вычисления

def calculate_element():
    global current_element
    while True:
        with lock:
            i, j = current_element
            if i >= m:
                break
            current_element[1] += 1
            if current_element[1] >= k:
                current_element[0] += 1
                current_element[1] = 0

        # Вычисляем элемент C[i][j]
        C[i][j] = sum(A[i][p] * B[p][j] for p in range(n))
```

```

        # Задержка для наглядности работы потоков
        time.sleep(0.1)

    print(f"Поток завершил работу.")

if __name__ == '__main__':
    # Количество потоков
    p = 4
    threads = []

    # Создаем и запускаем потоки
    for _ in range(p):
        thread = threading.Thread(target=calculate_element)
        threads.append(thread)
        thread.start()

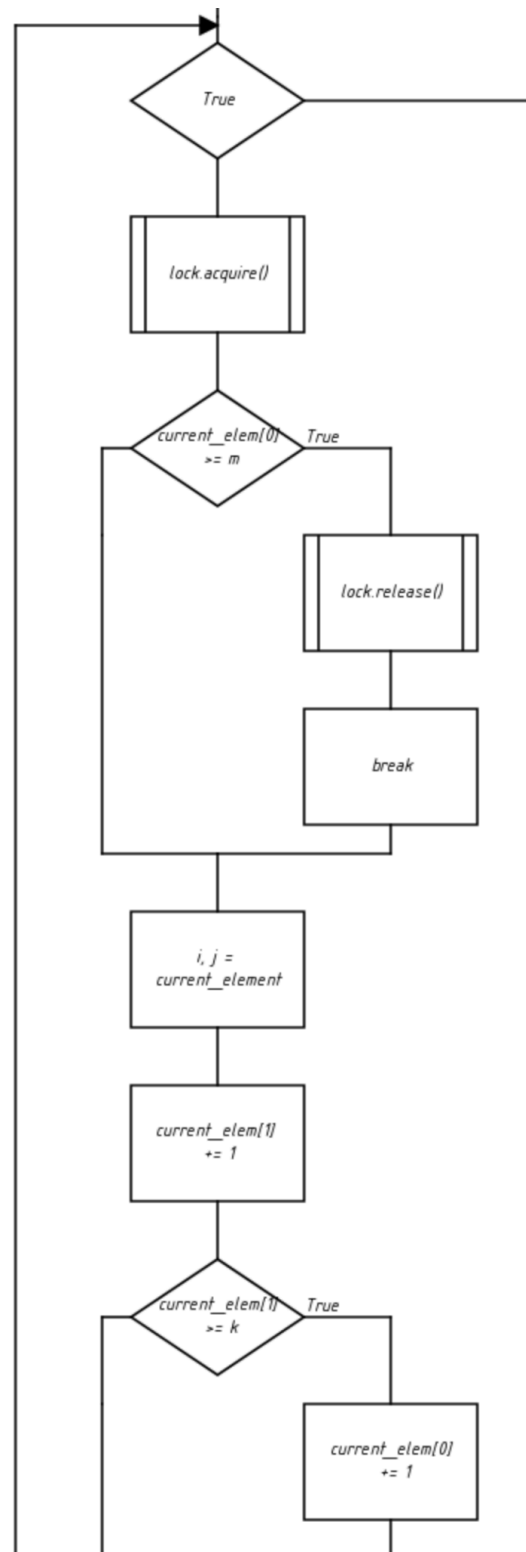
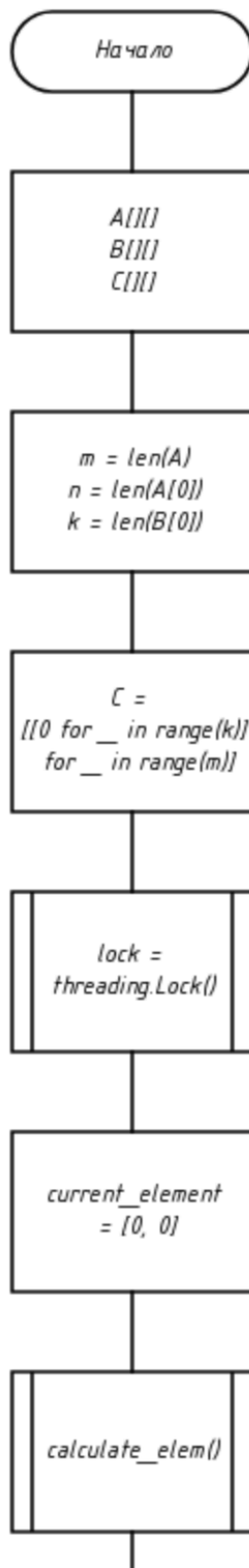
    # Ожидаем завершения всех потоков
    for thread in threads:
        thread.join()

    # Вывод результата
    print("Результат произведения матриц:")
    for row in C:
        print(row)

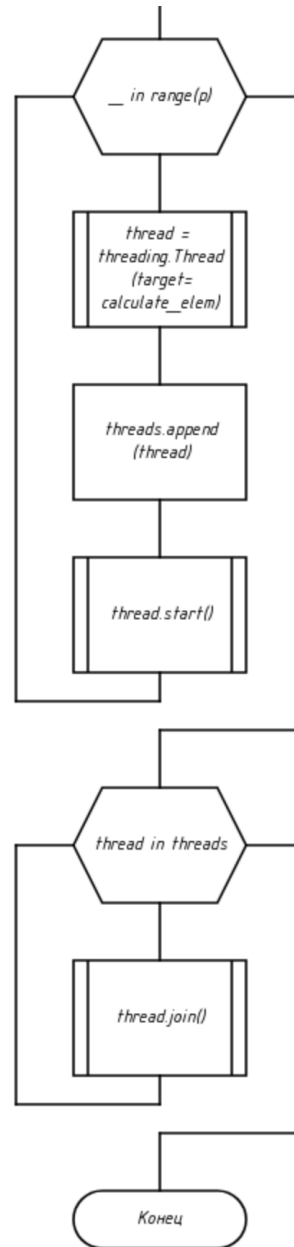
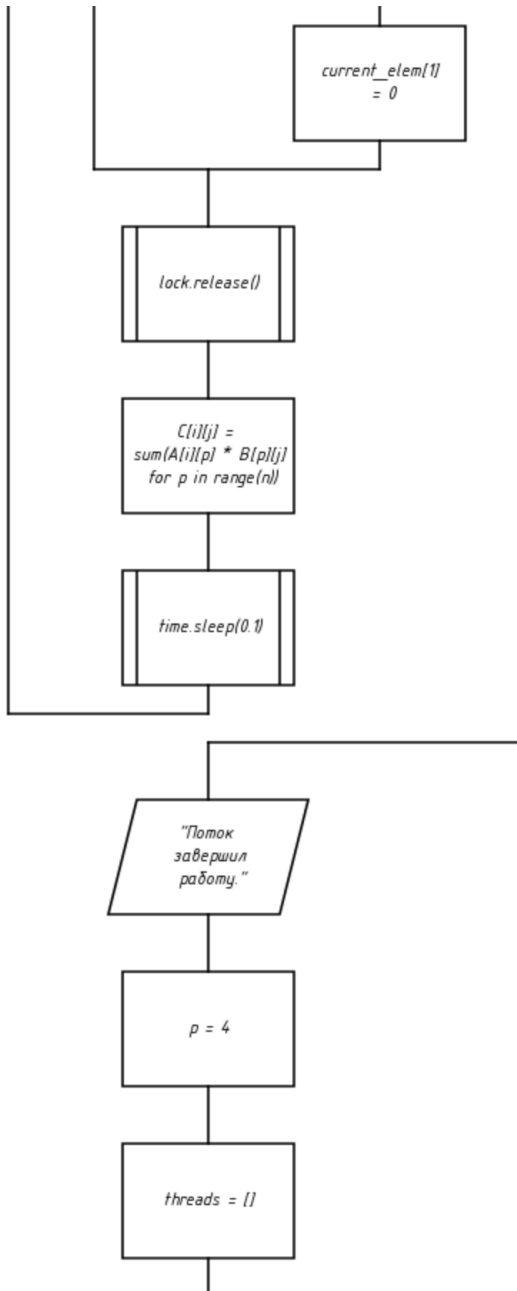
    # Запись результата в конец файла-протокола
    with open("matrix_result.txt", "a") as f:
        f.write("Результат произведения матриц:\n")
        for row in C:
            f.write(" ".join(map(str, row)) + "\n")

```

## Блок Схема







## Результат выполнения программы

```
/Users/noriksaroyan/PycharmProjects/PPL_HW1/.venv/bin/python /Users/noriksaroyan/PycharmProjects/PPL_HW1/lab5/Task2.py
Поток завершил работу.
Поток завершил работу.
Поток завершил работу.
Поток завершил работу.
Результат произведения матриц:
[30, 24, 18]
[84, 69, 54]
[138, 114, 90]

Process finished with exit code 0
```

**Вывод:** в ходе выполнения работы были приобретены практические навыки необходимые для разработки задач, решение которых предполагает использование многопоточности средствами языка Python.