Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №2 по курсу**

**«Операционные системы»**

**УПРАВЛЕНИЕ ПОТОКАМИ**

Студент: Корнев Максим Сергеевич

Группа: М8О–212Б–20

Вариант: 16

Преподаватель: Соколов Андрей Алексеевич

Оценка: \_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_

Москва, 2023.

**Постановка задачи**

## Цель работы

Приобретение практических навыков в:

* Управление процессами в ОС
* Обеспечение обмена данных между процессами посредством каналов

## Задание

Составить программу на языке Си, обрабатывающую данные в многопоточном режиме. При обработки использовать стандартные средства создания потоков операционной системы (Windows/Unix). Ограничение максимального количества потоков, работающих в один момент времени, должно быть задано ключом запуска вашей программы.

Так же необходимо уметь продемонстрировать количество потоков, используемое вашей программой с помощью стандартных средств операционной системы.

В отчете привести исследование зависимости ускорения и эффективности алгоритма от входных данных и количества потоков. Получившиеся результаты необходимо объяснить.

Вариант 16: Задаётся радиус окружности. Необходимо с помощью метода Монте-Карло рассчитать её площадь

**Общие сведения о программе**

Программа компилируется из файла «main.c». Внутри программы 2 функции: int main(int argc, char \*\*argv), принимающая аргументы командной строки, и функция void\* monteCarlo(void\* arg), которая принимает из командной строки величину радиуса и которая вычисляет площадь круга с заданным радиусом по методу Монте-Карло.

**Общий метод и алгоритм решения**.

Для реализации поставленной задачи необходимо:

1. Изучить принципы работы потоков в си.
2. Реализовать функцию, вычисляющую площадь по заданному методу: функция должна принимать все заданные параметры игры и рассчитывать число точек, которые будут обрабатываться каждым потоком numPointThread = Points / numThreads, где Points – общее количество точек, numThreads – количество потоков.
3. Реализовать функцию Монте-Карло, считающая количество точек, попавших во внутрь окружности.
4. Проверить, какой вариант решения задачи в данном случае лучше: однопоточное решение или многопоточное.

**Основные файлы программы**

**main.c:**

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <time.h>

#include <sys/time.h>

#include <math.h>

typedef struct ThreadData {

    int insidePoints;

    int totalPoints;

    double radius;

} Spot;

void\* monteCarlo(void\* arg) {

    struct ThreadData\* data = (struct ThreadData\*)arg;

    double radius = data->radius;

    unsigned int seed = (unsigned int)time(NULL);

    for(int i = 0; i < data->totalPoints; ++i) {

        double x = (((double)rand\_r(&seed) / RAND\_MAX) \* 2 \* radius) - radius;

        double y = (((double)rand\_r(&seed) / RAND\_MAX) \* 2 \* radius) - radius;

        if(x \* x + y \* y <= radius \* radius) {

            data->insidePoints++;

        }

    }

    pthread\_exit(NULL);

}

int main(int argc, char\* argv[]) {

    if(argc != 3) {

        printf("Usage: %s <radius> <number of threads>\n", argv[0]);

        return 1;

    }

    int Points = 1000;

    double radius = atof(argv[1]);

    int numThreads = atoi(argv[2]);

    pthread\_t threads[numThreads];

    Spot alpha[numThreads];

    int numPointThread = Points / numThreads;

    struct timeval start, end;

    int seconds, microSeconds;

    gettimeofday(&start, NULL);

    for(int i = 0; i < numThreads; ++i) {

        alpha[i].insidePoints = 0;

        alpha[i].totalPoints = numPointThread;

        pthread\_create(&threads[i],   NULL, monteCarlo, (void\*)&alpha[i]);

    }

    for(int i = 0; i < numThreads; ++i) {

        pthread\_join(threads[i], NULL);

    }

    gettimeofday(&end, NULL);

    int pointInsideCircle = 0;

    for(int i = 0; i < numThreads; ++i) {

        pointInsideCircle += alpha[i].insidePoints;

    }

   //printf("%d   ,   %.6lf\n", pointInsideCircle, (double)pointInsideCircle / Points);

    double area = 4 \* radius \* radius \* pointInsideCircle / Points ;

    printf("Area of the cirlce with radius %.3lf: %.3lf \n", radius, area);

    seconds = end.tv\_sec - start.tv\_sec;

    microSeconds = end.tv\_usec - start.tv\_usec;

    double time = seconds + microSeconds;

    printf("Execution time: %.3f ms\n", time);

    return 0;

}

**Пример работы**

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 3 10

Area of the cirlce with radius 3.000: 31.572

Execution time: 1719.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 3 10

Area of the cirlce with radius 3.000: 32.400

Execution time: 1611.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 3 10

Area of the cirlce with radius 3.000: 31.788

Execution time: 1526.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 3 10

Area of the cirlce with radius 3.000: 32.400

Execution time: 1668.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 3 10

Area of the cirlce with radius 3.000: 32.400

Execution time: 623.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 3 20

Area of the cirlce with radius 3.000: 35.676

Execution time: 1467.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 3 20

Area of the cirlce with radius 3.000: 35.676

Execution time: 1892.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 3 20

Area of the cirlce with radius 3.000: 35.676

Execution time: 1765.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 3 20

Area of the cirlce with radius 3.000: 35.820

Execution time: 1505.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 3 20

Area of the cirlce with radius 3.000: 35.460

Execution time: 2526.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 30 20

Area of the cirlce with radius 30.000: 3553.200

Execution time: 1813.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 30 20

Area of the cirlce with radius 30.000: 3560.400

Execution time: 1645.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 30 20

Area of the cirlce with radius 30.000: 3556.800

Execution time: 1708.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 30 20

Area of the cirlce with radius 30.000: 3556.800

Execution time: 1945.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 30 20

Area of the cirlce with radius 30.000: 3546.000

Execution time: 1461.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 30 50

Area of the cirlce with radius 30.000: 3600.000

Execution time: 3663.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 30 50

Area of the cirlce with radius 30.000: 3600.000

Execution time: 3868.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 30 50

Area of the cirlce with radius 30.000: 3600.000

Execution time: 4190.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 30 50

Area of the cirlce with radius 30.000: 3600.000

Execution time: 3037.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 30 50

Area of the cirlce with radius 30.000: 3600.000

Execution time: 3121.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 1 8

Area of the cirlce with radius 1.000: 3.500

Execution time: 1292.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 1 8

Area of the cirlce with radius 1.000: 3.500

Execution time: 1457.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 1 8

Area of the cirlce with radius 1.000: 3.500

Execution time: 518.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 1 8

Area of the cirlce with radius 1.000: 3.500

Execution time: 1368.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 1 8

Area of the cirlce with radius 1.000: 3.500

Execution time: 1038.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 5 1

Area of the cirlce with radius 5.000: 100.000

Execution time: 355.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 5 1

Area of the cirlce with radius 5.000: 100.000

Execution time: 390.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 5 1

Area of the cirlce with radius 5.000: 100.000

Execution time: 417.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 5 1

Area of the cirlce with radius 5.000: 100.000

Execution time: 352.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 5 1

Area of the cirlce with radius 5.000: 100.000

Execution time: 340.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 25 1

Area of the cirlce with radius 25.000: 2500.000

Execution time: 507.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 25 1

Area of the cirlce with radius 25.000: 2500.000

Execution time: 405.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 25 1

Area of the cirlce with radius 25.000: 2500.000

Execution time: 368.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 25 1

Area of the cirlce with radius 25.000: 2500.000

Execution time: 376.000 ms

radioactive@DESKTOP-RNP2IGB:/mnt/d/labs/os\_lab\_2/src$ ./main 25 1

Area of the cirlce with radius 25.000: 2500.000

Execution time: 318.000 ms

**Вывод**

Анализируя результаты выполненной работы, можно сказать, что увеличение количества потоков не даёт мне выигрыша во времени, с количеством точек, равным 1000. Взглянем на результаты:

|  |  |
| --- | --- |
| **Количество потоков** | **Время выполнения, секунд** |
| 1 | 0.370805 |
| 10 | 1.433480 |
| 20 | 1.831451 |
| 50 | 3.575365 |

Из данной таблицы следует, что программа, работающая с одним потоком, выполняется эффективнее всего. Также из таблицы можно выяснить, что программа работает менее эффективно с 10 и 20 потоками, чем с 50. При увеличении количества потоков, производительность падает. Предположительно, это возникает скорее всего из-за нагруженности системой другими процессами, однако ситуация может измениться, если мы увеличим количество точек, в программе я использовал 1000 точек, если увеличить их количество, то, вероятно, многопоточность ускорит работу программы.