**«ОПРЕДЕЛЕНИЕ ВРЕМЕНИ РАБОТЫ ПРИКЛАДНЫХ ПРОГРАММ»**

## Отчёт по лабораторной работе №1

## Выполнил: Ланин Д.М. ФИТ НГУ 2 курс

## Преподаватель: Артюхов А. А.

## Цели лабораторной работы:

1. Рассмотреть один из способов измерения времени подпрограммы.
2. Измерить время работы прикладной программы.

## Вариант задания:

**Алгоритм вычисления числа Пи методом Монте-Карло.**

*Далее подробно об алгоритме* : cначала в квадрат с центром в начале координат и со стороной два вписывается круг с единичным радиусом. Затем в этом квадрате случайным образом с равномерным распределением генерируются N точек. Точка может попасть в окружность или нет (условие попадания x2 + y 2 ≤ 1). Далее определяется число M точек, попавших в круг. При достаточно большом числе бросков N, по значениям M и N вычисляется число Пи: Pi = 4 \* M / N.

**Описание методики для определения времени работы программы:**

Для измерения времени исполнения кода я использовал функцию

clock()

 из модуля

time.h

 — она возвращает число таков, измеряемое процессором от начала выполнения программы.  
Глобальная константа

CLOCKS\_PER\_SEC

 хранит число тактов, выполняемое процессором в секунду. Соответственно, чтобы получить время работы программы в секундах достаточно результат работы функции разделить на эту константу:

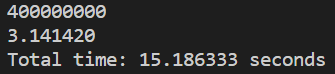
clock() / CLOCKS\_PER\_SEC;

## Результаты измерений

Измерения проводились для процессора на архитектуре AMD Picasso при зафиксированной частоте на уровне 2.10GHz.

При входных данных в размере 1 миллиарда операций, программа завершила работу за 37 секунд, с верно подсчитанной точность числа pi до 4го знака после запятой.

Рассмотрим меньшие входные данные, на примере 400 млн. На первом тесте результат был получен за 15.186 сек.



Абсолютная погрешность функции clock() состовляет 55мс, найдём относительную погрешность програмы, разделив 55 / 15186 \* 100 = 0.36% - погрешность. Для нахождения более точного времени исполнения программы запустим её несколько раз, и найдём среднее значение.

Среднее время вышло:



## Вывод:

Мы научились измерять время работы программы. Научились улучшать точность измерения программы. А так-же сделали вывод, что программа работающая на случайных входных данных не является хорошей по отношению точность/время.