



Министерство науки и высшего образования Российской Федерации  
Мытищинский филиал  
Федерального государственного автономного образовательного  
учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МФ МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ Космический

КАФЕДРА К3

## отчет

### *К ЛАБОРАТОРНОЙ РАБОТЕ*

*№* 3

по дисциплине

**«ОПЕРАЦИОННЫЕ СИСТЕМЫ»**

Студент К3-76Б  
(Группа)

В. Д. Чернов  
(И.О.Фамилия)

Преподаватель

А. В. Чернышов  
(И.О.Фамилия)

## **Задание к лабораторной работе**

Написать на языке Си программу, выполняющую вычисление математической операции, с распределением вычислений на компьютерном кластере, используя библиотеку OpenMPI.

## РЕШЕНИЕ ЗАДАНИЯ

```
#include <mpi.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define NUM_ITERATIONS 1500000000LL
#define FRAC_SCALE 1000000000000000LL // 1e15

int main(int argc, char** argv) {
    int rank, size;
    long long local_iters, local_sum_int = 0, local_sum_frac = 0;
    long long global_sum_int = 0, global_sum_frac = 0;
    double start_time, end_time, pi_approx;

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    if (rank == 0) start_time = MPI_Wtime();

    local_iters = NUM_ITERATIONS / size + (rank < (NUM_ITERATIONS %
size));
    long long start_idx = rank * (NUM_ITERATIONS / size) +
        (rank < (NUM_ITERATIONS % size) ? rank :
NUM_ITERATIONS % size);

    double scale = 1.0 / NUM_ITERATIONS;
    long long end_idx = start_idx + local_iters;

    for (long long i = start_idx; i < end_idx; i++) {
        double x = (i + 0.5) * scale;
        double x2 = x * x;
```

```

double denom = 1.0 + x2;
double term = 4.0 / denom;

long long term_int = (long long)term;
long long term_frac = (long long)((term - term_int) * FRAC_SCALE);

local_sum_int += term_int;
local_sum_frac += term_frac;

if (local_sum_frac >= FRAC_SCALE) {
    local_sum_int += local_sum_frac / FRAC_SCALE;
    local_sum_frac %= FRAC_SCALE;
}
}

MPI_Reduce(&local_sum_int, &global_sum_int, 1, MPI_LONG_LONG, MPI_SUM, 0,
MPI_COMM_WORLD);

MPI_Reduce(&local_sum_frac, &global_sum_frac, 1, MPI_LONG_LONG, MPI_SUM,
0, MPI_COMM_WORLD);

if (rank == 0) {

    global_sum_int += global_sum_frac / FRAC_SCALE;
    global_sum_frac %= FRAC_SCALE;

    pi_approx = (double)global_sum_int + (double)global_sum_frac / FRAC_SCALE;
    pi_approx /= NUM_ITERATIONS;

    end_time = MPI_Wtime();

    printf("Pi ≈ %.15f \n", pi_approx);
    printf("Время: %.6f сек \n", end_time - start_time, );
}

MPI_Finalize();
return 0;
}

```