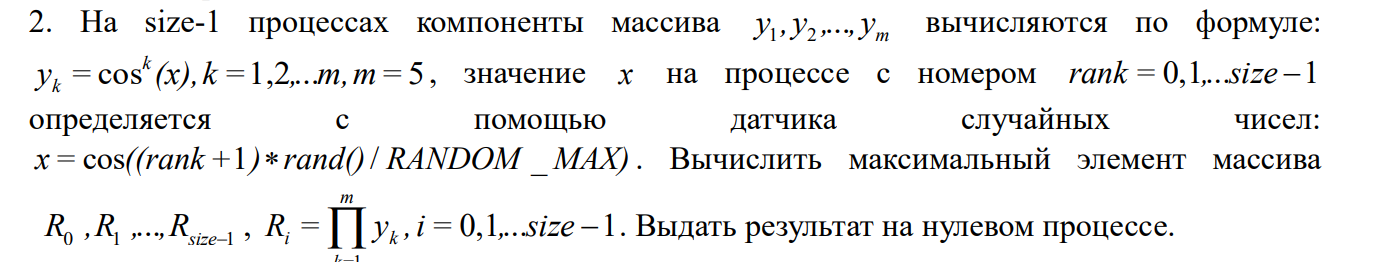
Отчет №1

Гренц Елизавета Алексеевна группа 932220



Описание работы:

С начало мы создаем массив из y (кол-во 5) на каждом процессе так же на каждом процессе генерируется X

После чего находим R. И находим максимальный R, передаем ответ на 0 процесс.

Код для вставки :

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <mpi.h>

#include <time.h>

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

int rank, size;

MPI\_Status status;

MPI\_Init(&argc, &argv);

MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank);

MPI\_Comm\_size(MPI\_COMM\_WORLD, &size);

srand(time(0) + rank);

int m = 5;

double R\_local;

double rec = -INFINITY;

int n = 4;

double x = ((rank + 1) \* rand()) / (double)RAND\_MAX;

double y[m + 1];

double R\_element = 1;

for (int k = 1; k < m + 1; k++) {

y[k] = pow(cos(x), k);

printf("Process: %d y[%d]= %f\n",rank,k, y[k]);

R\_element \*= y[k];

}

printf("Process: %d R = %f\n",rank,R\_element);

if (R\_element > rec) {

rec = R\_element;

}

if (rank != 0) {

MPI\_Send(&rec, 1, MPI\_DOUBLE, 0, 0, MPI\_COMM\_WORLD);

printf("\n");

} else {

printf("\n");

for (int i = 1; i < size; i++) {

double local\_max;

MPI\_Recv(&local\_max, 1, MPI\_DOUBLE, i, 0, MPI\_COMM\_WORLD, &status);

if (local\_max > rec) {

rec = local\_max;

}

}

printf("Global maximum: %lf\n", rec);

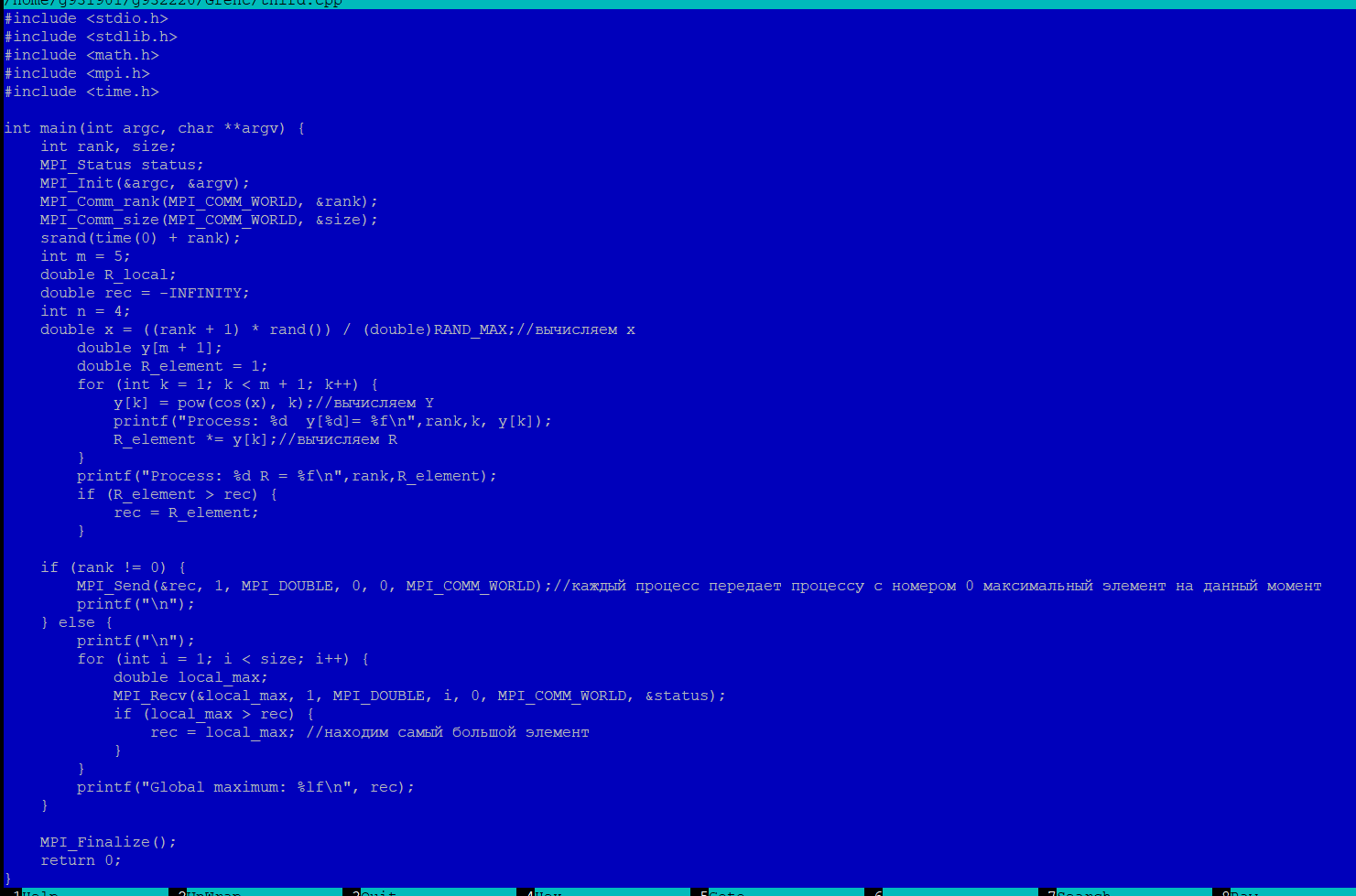
}

MPI\_Finalize();

return 0;

}

Код скрином(смотреть удобно)



Результаты

1-процесс

Process: 0 y[1]= 0.984329

Process: 0 y[2]= 0.968905

Process: 0 y[3]= 0.953721

Process: 0 y[4]= 0.938776

Process: 0 y[5]= 0.924065

Process: 0 R = 0.789055

Global maximum: 0.789055

1. 2-процесса

Process: 1 y[1]= 0.969228

Process: 1 y[2]= 0.939403

Process: 1 y[3]= 0.910496

Process: 1 y[4]= 0.882478

Process: 1 y[5]= 0.855323

Process: 1 R = 0.625734

Process: 0 y[1]= 0.965157

Process: 0 y[2]= 0.931529

Process: 0 y[3]= 0.899072

Process: 0 y[4]= 0.867746

Process: 0 y[5]= 0.837511

Process: 0 R = 0.587452

Global maximum: 0.625734

4-процесса

Process: 3 y[1]= 0.889793

Process: 2 y[1]= 0.996905

Process: 1 y[1]= 0.941975

Process: 1 y[2]= 0.887317

Process: 1 y[3]= 0.835830

Process: 1 y[4]= 0.787331

Process: 1 y[5]= 0.741646

Process: 1 R = 0.407934

Process: 3 y[2]= 0.791731

Process: 2 y[2]= 0.993819

Process: 2 y[3]= 0.990743

Process: 2 y[4]= 0.987676

Process: 2 y[5]= 0.984619

Process: 2 R = 0.954562

Process: 3 y[3]= 0.704476

Process: 3 y[4]= 0.626838

Process: 3 y[5]= 0.557755

Process: 3 R = 0.173513

Process: 0 y[1]= 0.690542

Process: 0 y[2]= 0.476848

Process: 0 y[3]= 0.329283

Process: 0 y[4]= 0.227384

Process: 0 y[5]= 0.157018

Process: 0 R = 0.003871

Global maximum: 0.954562

1. 6 -процессов

Process: 4 y[1]= 0.960585

Process: 4 y[2]= 0.922724

Process: 4 y[3]= 0.886356

Process: 4 y[4]= 0.851420

Process: 4 y[5]= 0.817862

Process: 4 R = 0.547067

Process: 1 y[1]= 0.795311

Process: 1 y[2]= 0.632519

Process: 1 y[3]= 0.503049

Process: 1 y[4]= 0.400080

Process: 1 y[5]= 0.318188

Process: 1 R = 0.032214

Process: 3 y[1]= 0.901606

Process: 3 y[2]= 0.812893

Process: 3 y[3]= 0.732909

Process: 3 y[4]= 0.660795

Process: 3 y[5]= 0.595776

Process: 3 R = 0.211471

Process: 2 y[1]= 0.995654

Process: 2 y[2]= 0.991326

Process: 2 y[3]= 0.987018

Process: 2 y[4]= 0.982728

Process: 2 y[5]= 0.978456

Process: 2 R = 0.936751

Process: 0 y[1]= 0.951589

Process: 5 y[1]= 0.935524

Process: 0 y[2]= 0.905522

Process: 0 y[3]= 0.861685

Process: 0 y[4]= 0.819970

Process: 0 y[5]= 0.780274

Process: 0 R = 0.475053

Process: 5 y[2]= 0.875205

Process: 5 y[3]= 0.818775

Process: 5 y[4]= 0.765983

Process: 5 y[5]= 0.716596

Process: 5 R = 0.367978

Global maximum: 0.936751

1. 8-процессов

Process: 4 y[1]= 0.786962

Process: 0 y[1]= 0.946114

Process: 0 y[2]= 0.895131

Process: 0 y[3]= 0.846896

Process: 0 y[4]= 0.801260

Process: 0 y[5]= 0.758083

Process: 0 R = 0.435663

Process: 2 y[1]= 0.990301

Process: 2 y[2]= 0.980697

Process: 2 y[3]= 0.971186

Process: 2 y[4]= 0.961766

Process: 2 y[5]= 0.952439

Process: 2 R = 0.863995

Process: 1 y[1]= 0.925108

Process: 7 y[1]= 0.754794

Process: 1 y[2]= 0.855825

Process: 1 y[3]= 0.791730

Process: 1 y[4]= 0.732436

Process: 1 y[5]= 0.677582

Process: 1 R = 0.311090

Process: 7 y[2]= 0.569713

Process: 7 y[3]= 0.430016

Process: 7 y[4]= 0.324573

Process: 7 y[5]= 0.244986

Process: 7 R = 0.014704

Process: 4 y[2]= 0.619309

Process: 4 y[3]= 0.487372

Process: 4 y[4]= 0.383543

Process: 4 y[5]= 0.301834

Process: 4 R = 0.027498

Process: 5 y[1]= 0.972106

Process: 5 y[2]= 0.944990

Process: 5 y[3]= 0.918631

Process: 5 y[4]= 0.893007

Process: 5 y[5]= 0.868098

Process: 5 R = 0.654192

Process: 3 y[1]= 0.932636

Process: 3 y[2]= 0.869809

Process: 3 y[3]= 0.811215

Process: 3 y[4]= 0.756568

Process: 3 y[5]= 0.705602

Process: 3 R = 0.351301

Process: 6 y[1]= 0.642739

Process: 6 y[2]= 0.413114

Process: 6 y[3]= 0.265525

Process: 6 y[4]= 0.170663

Process: 6 y[5]= 0.109692

Process: 6 R = 0.001320

Global maximum: 0.863995

Вывод :

В ходе решения практического задания, я использовала принципы двухточечного обмена (MPI\_Send и MPI\_Recv) , с помощью этого у меня получилось эффективно согласовать взаимодействия процессов.