# Отчёт по разработке и тестированию алгоритма заполнения массива псевдослучайными числами

Выполнил Скопецкий Анатолий, Б9118-09.03.04прогин

Количество элементов: 100.000.000.

## Таблица результатов параллельных вычислений

|  |  |  |
| --- | --- | --- |
| Результат вычисления | Время вычисления | Ускорение |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Краткое объяснение результатов

## Код алгоритма заполнения массива

#define localOffsetB offsetB \* (vecMulA[t] - 1) / (vecMulA[0] - 1)

#define parSeedI vecMulA[t + 1] \* seedI + localOffsetB

void vecRandParallel(unsigned seedGlobal, vector<vectorType>\* vec, unsigned min, unsigned max) {

//---> Создаем массив множителей A

vector<int> vecMulA(numOfThreads + 1);

vecMulA[0] = mulA;

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

vecMulA[i] = (vecMulA[i - 1] \* mulA) % mod;

}

//<---

//---> Создаем массив сумм всех элементов массива vec

accInit;

accResInit;

//<---

//---> Процесс каждого из ядер

auto thread\_proc = [=, &vec accThreadProc](unsigned t) {

unsigned seedI = vecMulA[t] \* seedGlobal + localOffsetB;

for (unsigned i = t; i < vec->size(); i += numOfThreads) {

vec->at(i).value = seedI % valBounds;

accInc;

seedI = parSeedI;

}

};

//<---

//---> Создаем массив исполняющих потоков и инициализируем исполнение каждого из потоков

vector<thread> threads;

for (unsigned t = 1; t < numOfThreads; t++)

threads.emplace\_back(thread\_proc, t);

thread\_proc(0);

for (auto& thread : threads)

thread.join();

//<---

accSum;

accOutput;

}