**Цель работы:** написать код по поставленной задаче.

**Задание:** Перемножить 2 квадратные матрицы размера 4096x4096 с элементами типа double. Выполнить 3 варианта перемножения и их анализ и сравнение

**Листинг программы:**

**Файл main.cpp**

#include <iostream>

#include <vector>

#include <random>

#include <chrono>

#include <omp.h>

#include <cblas.h>

using namespace std;

using namespace chrono;

const int N = 4096;

const int BLOCK\_SIZE = 64;

void generateMatrix(vector<double>& mat) {

mt19937 gen(42);

uniform\_real\_distribution<double> dist(0.0, 1.0);

for (int i = 0; i < N \* N; ++i)

mat[i] = dist(gen);

}

void multiplyClassic(const vector<double>& A, const vector<double>& B, vector<double>& C) {

fill(C.begin(), C.end(), 0.0);

for (int i = 0; i < N; ++i)

for (int j = 0; j < N; ++j)

for (int k = 0; k < N; ++k)

C[i \* N + j] += A[i \* N + k] \* B[k \* N + j];

}

void multiplyBlocked(const vector<double>& A, const vector<double>& B, vector<double>& C) {

fill(C.begin(), C.end(), 0.0);

#pragma omp parallel for collapse(2)

for (int ii = 0; ii < N; ii += BLOCK\_SIZE)

for (int jj = 0; jj < N; jj += BLOCK\_SIZE)

for (int kk = 0; kk < N; kk += BLOCK\_SIZE)

for (int i = ii; i < min(ii + BLOCK\_SIZE, N); ++i)

for (int j = jj; j < min(jj + BLOCK\_SIZE, N); ++j)

for (int k = kk; k < min(kk + BLOCK\_SIZE, N); ++k)

C[i \* N + j] += A[i \* N + k] \* B[k \* N + j];

}

void multiplyBLAS(const vector<double>& A, const vector<double>& B, vector<double>& C) {

fill(C.begin(), C.end(), 0.0);

cblas\_dgemm(CblasRowMajor, CblasNoTrans, CblasNoTrans,

N, N, N, 1.0, A.data(), N, B.data(), N, 0.0, C.data(), N);

}

template <typename Func>

double measureTime(Func func) {

auto start = high\_resolution\_clock::now();

func();

auto end = high\_resolution\_clock::now();

return duration<double>(end - start).count();

}

int main() {

cout << "Святухин Данил Алексеевич 02.03.03-АИСа-о24\n" << endl;

vector<double> A(N \* N);

vector<double> B(N \* N);

vector<double> C(N \* N, 0.0);

generateMatrix(A);

generateMatrix(B);

cout << "A[0]: " << A[0] << ", B[0]: " << B[0] << endl;

double complexity = 2.0 \* N \* N \* N;

double t\_classic = measureTime([&]() { multiplyClassic(A, B, C); });

double mflops\_classic = complexity / (t\_classic \* 1e6);

cout << "Classic time: " << t\_classic << "MFLOPS: " << mflops\_classic << endl;

cout << "C[0]: " << C[0] << endl;

double t\_blas = measureTime([&]() { multiplyBLAS(A, B, C); });

double mflops\_blas = complexity / (t\_blas \* 1e6);

cout << "BLAS time: " << t\_blas << "MFLOPS: " << mflops\_blas << endl;

cout << "C[0]: " << C[0] << endl;

double t\_blocked = measureTime([&]() { multiplyBlocked(A, B, C); });

double mflops\_blocked = complexity / (t\_blocked \* 1e6);

cout << "blocked time: " << t\_blocked << "MFLOPS: " << mflops\_blocked << endl;

cout << "C[0]: " << C[0] << endl;

double performance\_ratio = mflops\_blocked / mflops\_blas;

cout << "\nblocked " << performance\_ratio \* 100 << "% of BLAS performance" << endl;

return 0;

}

**Вывод:** В ходе выполнения лабораторной работы была сделана программа перемножающая 2 квадратные матрицы размера 4096x4096 с элементами типа double.