#include <omp.h>

#include <iostream>

#include <ctime>

using namespace std;

void PrintM(int\*\* a, int n)

{

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

{

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

printf("%d ", a[i][j]);

printf("\n");

}

}

void PrintV(int\* a, int n)

{

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

printf("%d ", a[i]);

printf("\n");

}

void VvodV(int\* a, int n)

{

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

{

printf("Введите элемент вектора v[%d]=", i);

cin >> a[i];

}

}

void VvodM(int\*\* a, int n)

{

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

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

{

printf("Введите элемент матрицы A[%d][%d]=", i, j);

cin >> a[i][j];

}

}

void Mult(int\*\* A, int\* b, int\* c, int n)

{

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

{

c[i] = 0;

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

c[i] += A[i][j] \* b[j];

}

}

int main()

{

int i, j, n, c\_size=2;

setlocale(LC\_ALL, "Russian");

printf("Введите размер n=");

cin >> n;

int\*\* A = new int\* [n];

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

A[i] = new int[n];

int\* v = new int[n];

int\*b = new int[n];

//Вводим данные

VvodM(A, n);

VvodV(v, n);

printf("Для начала умножим матрицу на вектор последовательно\n");

unsigned int start1 = clock();

for ( i = 0; i < n; i++)

{

b[i] = 0;

for ( j = 0; j < n; j++)

b[i] += A[i][j] \* v[j];

}

printf("Матрица А:\n");

PrintM(A, n);

printf("Вектор V:\n");

PrintV(v, n);

printf("Результат умножения матрицы на вектор:\n");

PrintV(b, n);

unsigned int end1 = clock();

cout << "Время последовательной программы = "<< (end1-start1/ CLOCKS\_PER\_SEC)<< endl << "Конец фрагмента последовательной программы." << endl;

printf("\nУмножение матрицы на вектор параллельно\n");

unsigned int start2 = clock();

omp\_set\_num\_threads(2);

#pragma omp parallel for schedule(static, c\_size)

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

{

b[i] = 0;

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

b[i] += A[i][j] \* v[j];

printf("Кол-во = %d. Нить = %d. b[%d] = %d.\n", omp\_get\_num\_threads(), omp\_get\_thread\_num(), i, b[i]);

}

printf("Матрица А:\n");

PrintM(A, n);

printf("Вектор V:\n");

PrintV(v, n);

printf("Результат умножения матрицы на вектор:\n");

PrintV(b, n);

unsigned int end2= clock();

cout << "Время параллельной программы = "<< (end2 - start2 / CLOCKS\_PER\_SEC)<< "s" << endl << "Конец фрагмента параллельной программы." << endl;

for (i = 0; i<n; i++)

delete [] A[i];

delete [] v;

delete [] b;

return 0;

}