Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Алтайский государственный технический университет

им. И.И. Ползунова»

Факультет (институт) Информационных технологий

Кафедра Прикладная математика

Отчет защищен с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_

А.И.Потупчик

(подпись преподавателя) (инициалы, фамилия)

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Отчет

по лабораторной (практической) работе №\_\_9\_\_

Реализация алгоритма на основе программирования OpenMP

(название лабораторной (практической работы)

по дисциплине Архитектура ЭВМ

(наименование дисциплины)

ЛР 09.03.04.21.000 ОТ

(обозначение документа)

Студент группы ПИ-02 Р.А.Чередов

(инициалы, фамилия)

Преподаватель доцент А.И.Потупчик

(должность, учебное звание) (инициалы, фамилия)

Барнаул 2022

Задание:

**21. Дан двумерный массив b. Рассматривая его как матрицу, транспонировать b.**

Решение:

#include <iostream>

#include <omp.h>

#include <ctime>

#include <deque>

using namespace std;

class matrix {

deque<int> ar;

int row;

int column;

public:

// Конструктор. Принимает параметры row - количество строк и column - кол-во столбцов

matrix(int row, int column) {

this->row = row;

this->column = column;

}

// Функция заполнения матрицы

void input() {

for (int i = 0; i < row \* column; ++i) {

ar.emplace\_back(i+1);

}

}

// Функция транспонирования матрицы

void Trans() {

deque<int> temp; // Массив который мы заполняем в виде транспонированной матрицы

int tempNum = 0; // переменная для обмена числа столбцов и колонок местами

for (int i = 0; i < column; ++i) {

temp.emplace\_back(ar[i]); // Добавляем i элемент в массив

for (int j = 1; j < row; ++j) {

temp.emplace\_back(ar[j \* column + i]); // Добавляем следующий элемент колонки

}

}

ar = temp;

tempNum = row;

row = column;

column = tempNum;

}

// Вывод матрицы в консоль

void print() {

for (int i = 0; i < row; ++i) {

for (int j = 0; j < column; ++j) {

cout << ar[i \* column + j] << "\t";

}

cout << endl;

}

cout << endl;

}

};

int main() {

const int numRow = 3; // Количество строк в матрице

const int numColumn = 4; // Количество столбцов в матрице

double t1, t2, dt; // Переменные для хранения времени выполнения

matrix A(numRow, numColumn);// Матрица A для работы без openMP

matrix B(numRow, numColumn);// Матрица B для работы c openMP

cout << endl;

t1 = omp\_get\_wtime(); // Запуск таймера

A.input(); // Заполнение матрицы

A.print(); // Вывод матрицы в консоль

A.Trans(); // Транспонирование матрицы

A.print(); // Вывод матрицы в консоль

t2 = omp\_get\_wtime(); // Конец таймера

dt = t2 - t1; // Расчитываем время работы области программы

cout << "Work time = " << dt << endl << endl;

t1 = omp\_get\_wtime(); // Запуск таймера

#pragma omp parallel // Выполнения программы с использованием многопоточности

{

B.input(); // Заполнение матрицы

B.print(); // Вывод матрицы в консоль

B.Trans(); // Транспонирование матрицы

B.print(); // Вывод матрицы в консоль

}

t2 = omp\_get\_wtime(); // Конец таймера

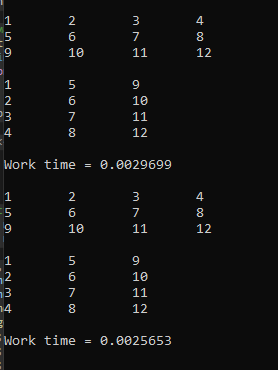
dt = t2 - t1; // Расчитываем время работы области программы

cout << "Work time = " << dt << endl << endl;

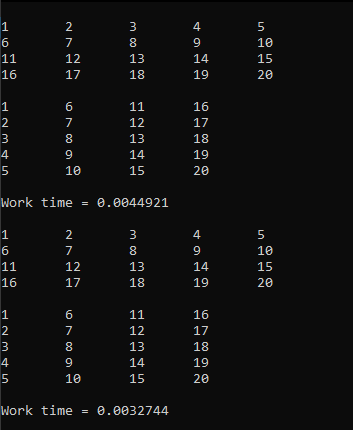
}

Тесты программы:

Матрица 3 на 4



4 на 5



5 на 5

