

ДСОГЛАСОВАНО

Научный руководитель,
доцент департамента программной инженерии
факультета компьютерных, канд. техн. наук

_____ А.И. Легалов

« ____ » _____ 2020 г.

УТВЕРЖДАЮ

Академический руководитель образовательной
программы «Программная инженерия» профессор
департамента программной инженерии, канд. техн.
наук

_____ В. В. Шилов

« ____ » _____ 2020 г.

| | | | | |
|-------------|--------------|--------------|--------------|--------------|
| Инв. № подл | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |
| | | | | |

Вариант 12

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.01-01 81 01-1 ЛУ

Исполнитель

студент группы БПИ197

_____ / П. О. Кулешова /

« ____ » _____ 2020 г.

| Инв. № подл | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |
|-------------|--------------|--------------|--------------|--------------|
| | | | | |

Вариант 12
Пояснительная записка
RU.17701729.04.01-01 81 01-1
Листов 13

RU.17701729.04.01-01 81 01-1

СОДЕРЖАНИЕ

| | |
|--|----------|
| ВВЕДЕНИЕ | 3 |
| 1.1. Наименование программы | 3 |
| 1.2. Документ, на основе которого ведется разработка | 3 |
| 1. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ | 4 |
| 1.1 Назначение разработки | 4 |
| 1.2 Краткая характеристика области применения | 4 |
| 2. Описание программы | 5 |
| Условие задачи | 5 |
| Метод решения | 5 |
| Алгоритм решения | 5 |
| Замечание | 5 |
| ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ | 7 |
| ПРИЛОЖЕНИЕ 2 Код программы | 8 |
| ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ | 8 |

| | | | | |
|------------------------------|--------------|--------------|--------------|--------------|
| | | | | |
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.04.01-01 81 01-1 | | | | |
| Инв. № подл. | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

RU.17701729.04.01-01 81 01-1

ВВЕДЕНИЕ**1.1.Наименование программы**

Наименование программы: Практические приемы построения многопоточных приложений

Краткое наименование программы: ДЗ

1.2.Документ, на основе которого ведется разработка

<http://www.softcraft.ru/edu/comparch/tasks/t03/>

| | | | | |
|------------------------------|--------------|--------------|--------------|--------------|
| | | | | |
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.04.01-01 81 01-1 | | | | |
| Инв. № подл. | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

RU.17701729.04.01-01 81 01-1

1. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ**1.1 Назначение разработки**

Разработать многопоточную программу для нахождения наибольшей длины

1.2 Краткая характеристика области применения

Научно-исследовательская область.

| | | | | |
|------------------------------|--------------|--------------|--------------|--------------|
| | | | | |
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.04.01-01 81 01-1 | | | | |
| Инв. № подл. | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

2. ОПИСАНИЕ ПРОГРАММЫ

Условие задачи

Определить индексы i , j , для которых существует наиболее длинная последовательность $A[i] < A[i+1] < A[i+2] < A[i+3] < \dots < A[j]$. Входные данные: массив чисел A , произвольной длины большей 1000. Количество потоков является входным параметром

Метод решения

Задача решалась с использованием **итеративного параллелизма**. Так как при поиске подпоследовательности мы не записываем никакие дополнительные данные и все потоки равноправны.

Алгоритм решения

1. Разделяем массив на число промежутков равное числу потоков
2. Каждый поток проходит по своему промежутку и запоминает наибольшую длину возрастающей последовательности.
 - а. Если к концу промежутка он обнаружил, что следующий элемент существует и он больше текущего, то он продолжает идти по последовательности до окончания возрастающей последовательности.
3. Когда все потоки закончили работу, то проходимся по всему массиву результатов и находим наилучший. Результат записываем в файл.

Замечание

Считалось, что последовательность из одного символа является возрастающей.
Искалась первая наибольшая последовательность.

| | | | | |
|------------------------------|--------------|--------------|--------------|--------------|
| | | | | |
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.04.01-01 81 01-1 | | | | |
| Инв. № подл. | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

RU.17701729.04.01-01 81 01-1

3. ФОРМАТ ВХОДНЫХ ДАННЫХ

Входные данные командной строки

Программа на вход получает данные из входной строки в следующем формате: *<Путь до файла с тестом>_<Путь до файла с ответом>_<Число потоков>*

Значок _ означает пробел, то есть входные данные разделены одинарным пробелом.

Входные данные для тестов

Тест должен состоять из строки с числом элементов в массиве. На второй строке – элементы массива через пробел.

4. ФОРМАТ ВЫХОДНЫХ ДАННЫХ

Текстовый файл, где на первой строке записан индекс начала последовательности, на второй – индекс окончания, на третьей – длина последовательности.

| | | | | |
|------------------------------|--------------|--------------|--------------|--------------|
| | | | | |
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.04.01-01 81 01-1 | | | | |
| Инв. № подл. | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

RU.17701729.04.01-01 81 01-1

ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

1. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению// Единая система программной документации. – М.: ИПК Стандартиформ, 2010.
2. Сайт «SoftCraft». URL: <http://www.softcraft.ru/edu/comparch/practice/thread/01-simple/> [<http://www.softcraft.ru>] Просмотрено: 17.11.2020
3. Сайт «Основы многопоточного, параллельного и распределенного программирования ГРЕГОРИ Р. ЭНДРЮС». URL: https://l.wzm.me/_coder/custom/parallel.programming/001.htmПросмотрено: 17.11.2020

| | | | | |
|------------------------------|--------------|--------------|--------------|--------------|
| | | | | |
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.04.01-01 81 01-1 | | | | |
| Инв. № подл. | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

ПРИЛОЖЕНИЕ 2

Код программы

```
// Threads.cpp : Этот файл содержит функцию "main". Здесь начинается и заканчивается
// выполнение программы.
#include <pthread.h>
#include <string>
#include <ctime>
#include <fstream>
#include <iostream>
#include <sstream>

using namespace std;

/// <summary>
/// Структура данных для потоков
/// </summary>
struct Package {
    int* array; // Указатель на начало области обработки
    unsigned int threadNum; // Номер потока
    unsigned int i; //Начало последовательности
    unsigned int j; //Конец последовательности
    unsigned int max = 1; //Размер последовательности
    unsigned int start; //Начало сектора потока
    unsigned int finish; //Конец сектора
    unsigned int number; //Число элементов в массиве
};

/// <summary>
/// Функция для подсчёта максимальной длины последовательности на промежутке потоком
/// </summary>
/// <param name="param">Структура данных для работы потока</param>
/// <returns>nullptr</returns>
void* func(void* param);

/// <summary>
/// Функция для вывода результата в файл
/// </summary>
/// <param name="threadNumber">Количество потоков</param>
/// <param name="pack">Структура данных</param>
/// <param name="answer">Путь к файлу с ответом</param>
void Write_answer(unsigned int threadNumber, Package* pack, const std::string& answer);

/// <summary>
/// Передача параметров в структуру данных
/// </summary>
/// <param name="pack">Структура данных</param>
/// <param name="q">Номер потока</param>
/// <param name="num">Число элементов в массиве</param>
/// <param name="threadNumber">Число потоков</param>
/// <param name="array">Массив для изучения</param>
void Pack_params(Package* pack, unsigned int q, unsigned int num, unsigned int
threadNumber, int* array);
```

| | | | | |
|------------------------------|--------------|--------------|--------------|--------------|
| | | | | |
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.04.01-01 81 01-1 | | | | |
| Инв. № подл. | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

RU.17701729.04.01-01 81 01-1

```

/// <summary>
/// Основная программа
/// </summary>
/// <param name="args">Число аргументов командной строки</param>
/// <param name="argv">Массив аргументов командной строки</param>
/// <returns>Код ошибки</returns>
int main(int args, char* argv[])
{
    const string test = argv[1]; //путь до теста
    const string answer = argv[2]; //путь до ответа
    const string threds = argv[3]; //число потоков
    //создание потока для чтения
    ifstream fin(test);
    if (!fin.is_open()) {
        throw runtime_error("IO Exception");
    }
    unsigned int threadNumber = stoi(threds);
    int s;
    unsigned int num;
    fin >> num; //число элементов в массиве
    int* array = new int[num];
    unsigned int w = 0;
    //Чтение элементов массива и запись их в массив
    while (!fin.eof() && w < num)
    {
        fin >> s;
        array[w] = s;
        w++;
    }
    fin.close(); //закрытие потока
    pthread_t* thread = new pthread_t[threadNumber];
    Package* pack = new Package[threadNumber];
    //Создание потоков
    for (unsigned int q = 0; q < threadNumber; q++) {
        Pack_params(pack, q, num, threadNumber, array);
        pthread_create(&thread[q], nullptr, func, (void*)&pack[q]);
    }
    pthread_join(*thread, nullptr); //ожидание завершения всех потоков
    Write_answer(threadNumber, pack, answer); //запись ответа
    //очистка памяти
    delete[] thread;
    delete[] array;
    delete[] pack;
    return 0;
}

/// <summary>
/// Передача параметров в структуру данных
/// </summary>
/// <param name="pack">Структура данных</param>
/// <param name="q">Номер потока</param>
/// <param name="num">Число элементов в массиве</param>
/// <param name="threadNumber">Число потоков</param>
/// <param name="array">Массив для изучения</param>

```

| | | | | |
|------------------------------|--------------|--------------|--------------|--------------|
| | | | | |
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.04.01-01 81 01-1 | | | | |
| Инв. № подл. | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

```

void Pack_params(Package* pack, unsigned int q, unsigned int num, unsigned int
threadNumber, int* array)
{
    //Если число потоков меньше числа элементов массива
    if (threadNumber <= num)
    {
        pack[q].start = (num / threadNumber) * (q); //начать с ячейки массива
равной данному номеру
        pack[q].finish = (num / threadNumber) * (q + 1); //закончить соответственно
на начале для следующего потока
    }
    //Данная ситуация невозможно при корректном использовании программы, так как
создание большого числа потоков
    //только до определённого момента увеличивает скорость работы программы
    //Так как на вход подаётся больше 1000 чисел, то создавать 1001 поток не имеет
смысла
    else //Если число потоков больше числа элементов массива, то, наоборот,
распределять относительно числа потоков
    {
        pack[q].start = (threadNumber / num) * (q) > num ? num : (threadNumber /
num) * (q);
        pack[q].finish = (threadNumber / num) * (q + 1) > num ? num : (threadNumber
/ num) * (q);
    }
    //Запись параметров в структуру данных
    pack[q].threadNum = q;
    pack[q].array = array;
    pack[q].number = num;
}

/// <summary>
/// Функция для подсчёта максимальной длины последовательности на промежутке потоком
/// </summary>
/// <param name="param">Структура данных для работы потока</param>
/// <returns>nullptr</returns>
void* func(void* param)
{
    //Восстанавливаем структуру
    Package* p = (Package*)param;
    //Начинаем идти с start
    unsigned int i = p->start;
    //Записываем первый для нас элемент массива
    int c = p->array[i];
    //Будем считать, что один элемент является последовательностью
    p->i = i;
    p->j=i;
    unsigned int max = 0;
    unsigned int i2 = 0;
    //Идём до конца нашего промежутка
    //Или пока наша последовательность не перестанет возрастать
    while (i <= (p->finish) || ((i + 1 < p->number) && (p->array[i + 1] > c)))
    {
        //"Обнуляем" значения
        max = 1;
        i2 = i;
    }
}

```

| | | | | |
|------------------------------|--------------|--------------|--------------|--------------|
| | | | | |
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.04.01-01 81 01-1 | | | | |
| Инв. № подл. | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

RU.17701729.04.01-01 81 01-1

```

c = p->array[i];
//Идём пока наша последовательность возрастает
while ((i + 1 < p->number) && (p->array[i + 1] > c))
{
    max++;
    i++;
    c = p->array[i];
}
//Записываем результат
if (max > p->max)
{
    p->i = i2 == p->number ? i2 - 1 : i2;
    p->j = i == p->number ? i - 1 : i;
    p->max = max;
}
i++;
}
return nullptr;
}

/// <summary>
/// Функция для вывода результата в файл
/// </summary>
/// <param name="threadNumber">Количество потоков</param>
/// <param name="pack">Структура данных</param>
/// <param name="answer">Путь к файлу с ответом</param>
void Write_answer(unsigned int threadNumber, Package* pack, const std::string& answer)
{
    unsigned int i = 0, j = 0, max = 0;
    //Проходимся в цикле по результатам потоков
    for (unsigned int q = 0; q < threadNumber; q++)
    {
        if (pack[q].max > max)
        {
            max = pack[q].max;
            i = pack[q].i;
            j = pack[q].j;
        }
    }
    //Записываем наилучший результат в файл
    fstream out(answer, ios::out);
    out << "i = ";
    out << i;
    out << "\n";
    out << "j = ";
    out << j;
    out << "\n";
    out << "length : ";
    out << max;
    out.close();
}

```

| | | | | |
|------------------------------|--------------|--------------|--------------|--------------|
| | | | | |
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.04.01-01 81 01-1 | | | | |
| Инв. № подл. | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

[illegible]

| | | | | |
|------------------------------|--------------|--------------|--------------|--------------|
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.04.01-01 81 01-1 | | | | |
| Инв. № подл. | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |



This document was created with the Win2PDF "print to PDF" printer available at
<http://www.win2pdf.com>

This version of Win2PDF 10 is for evaluation and non-commercial use only.

This page will not be added after purchasing Win2PDF.

<http://www.win2pdf.com/purchase/>