

Архитектура вычислительных систем
Отчёт по заданию «Практические приемы построения многопоточных
приложений»
Вариант 24

Петров Иван Андреевич, БПИ195

Содержание

1	Описание задачи	1
2	Реализация	1
3	Формат входных данных	2
4	Работа программы	2
4.1	Пример 1	2
4.2	Пример 2	2
4.3	Пример 3	3
4.4	Пример 4	3
4.5	Некорректные данные	3

1 Описание задачи

Задача о Пути Кулака. На седых склонах Гималаев стоят два древних буддистских монастыря: Гуань-Инь и Гуань-Янь. Каждый год в день сошествия на землю боддисатвы Араватти монахи обоих монастырей собираются на совместное празднество и показывают свое совершенствование на Пути Кулака. Всех соревнующихся монахов разбивают на пары, победители пар бьются затем между собой и так далее, до финального поединка. Монастырь, монах которого победил в финальном бою, забирает себе на хранение статую боддисатвы. Реализовать многопоточное приложение, определяющего победителя. В качестве входных данных используется массив, в котором хранится количество энергии Ци каждого монаха. При решении использовать принцип дихотомии.

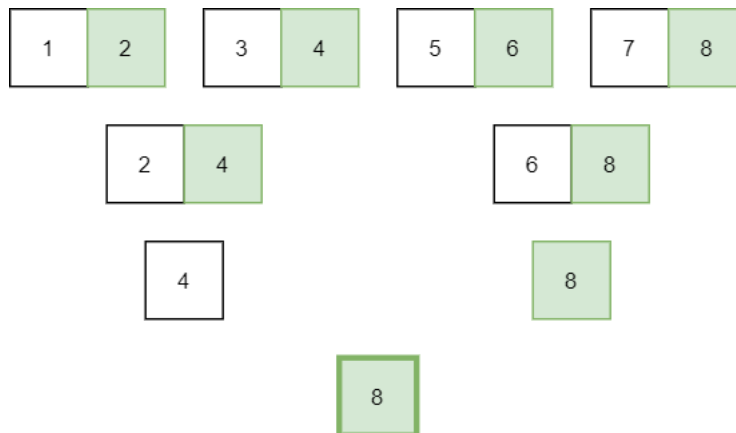
2 Реализация

Задача сводится к поиску наибольшего значения в одномерном массиве с помощью алгоритма, основанного на подходе «разделяй-и-властвуй». Код данного алгоритма на языке C++:

```
int find_max(int *array, int size) {  
    if (size == 1) return array[0];  
    int mid = size % 2 == 0 ? size / 2 : size / 2 + 1;  
    int v1 = find_max(array, mid);  
    int v2 = find_max(array + mid, mid);  
    return v1 >= v2 ? v1 : v2;  
}
```

Таким образом, сначала будет найден максимальный элемент в каждой паре, затем максимальный элемент среди найденных пар и так далее, пока не останется один элемент - максимальный в массиве.

Пример работы алгоритма изображен на рисунке:



Для обеспечения многопоточности используется модель «Рекурсивный параллелизм». Исходный массив разбивается на несколько подмассивов, количество которых зависит от количества указанных потоков. Затем каждый дочерний поток находит максимальный элемент в своем подмассиве, после чего в родительском потоке находится максимальный элемент среди результатов работы дочерних потоков.

3 Формат входных данных

Входные данные задаются с помощью аргументов командной строки в следующем формате:

\$ <путь до исполняемого файла> <размер массива> <левая граница> <правая граница> <количество потоков>

Под границами подразумеваются минимальные и максимальные значения для генератора случайных чисел.

4 Работа программы

4.1 Пример 1

Результат работы программы со следующими входными данными:

- размер массива = 10^6
- левая граница = -10^5
- правая граница = 10^5
- количество потоков = 2

```
delta@DESKTOP-I9D6ULO:~/projects/monkbattle$ ./main 1000000 -100000 100000 2
Maximum value = 100000
Time = 4431
```

4.2 Пример 2

Результат работы программы со следующими входными данными:

- размер массива = 10^6
- левая граница = -10^5
- правая граница = 10^5
- количество потоков = 8

```
delta@DESKTOP-I9D6ULO:~/projects/monkbattle$ ./main 1000000 -100000 100000 8
Maximum value = 100000
Time = 455
```

4.3 Пример 3

Результат работы программы со следующими входными данными:

- размер массива = 10^7
- левая граница = -10^5
- правая граница = 10^5
- количество потоков = 1

```
delta@DESKTOP-I9D6ULO:~/projects/monkbattle$ ./main 10000000 -100000 100000 1
Maximum value = 100000
Time = 155842
```

4.4 Пример 4

Результат работы программы со следующими входными данными:

- размер массива = 10^7
- левая граница = -10^5
- правая граница = 10^5
- количество потоков = 8

```
delta@DESKTOP-I9D6ULO:~/projects/monkbattle$ ./main 10000000 -100000 100000 8
Maximum value = 100000
Time = 110917
```

4.5 Некорректные данные

При вводе некорректных данных программа выдаст сообщение с указанием формата входных данных.

```
delta@DESKTOP-I9D6ULO:~/projects/monkbattle$ ./main a b c d
Usage: <path to executable file> <array size> <min value> <max value> <threads number>
```