

Архитектура вычислительных систем
Отчёт по заданию «Разработка многопоточных приложений с
использованием OpenMP»
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	2

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.

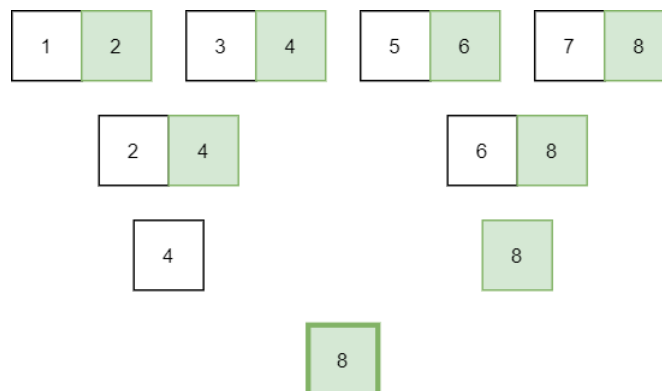
2 Реализация

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) {
    int v1, v2, res;
    if (size == 1) return array[0];
    int mid = size % 2 == 0 ? size / 2 : size / 2 + 1;
    #pragma omp task shared(v1)
    v1 = find_max(array, mid);
    #pragma omp task shared(v2)
    v2 = find_max(array + mid, mid);
    #pragma omp taskwait
    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» и директивы *OpenMP*. Исходный массив разбивается на несколько подмассивов, количество которых зависит от результата функции `omp_get_max_threads()`. Затем для каждого подмассива находится свой максимальный элемент, после чего в родительском потоке находится максимальный элемент среди максимальных элементов подмассивов.

3 Формат входных данных

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

\$ <путь до исполняемого файла> <размер массива>

4 Работа программы

4.1 Пример 1

Результат работы программы со следующими входными данными:

- размер массива = 8

```
[Thread 0] Monk (1) is fighting against Monk (2). The winner is Monk (2)
[Thread 0] Monk (3) is fighting against Monk (4). The winner is Monk (4)
[Thread 0] Monk (2) is fighting against Monk (4). The winner is Monk (4)
[Thread 0] Monk (5) is fighting against Monk (6). The winner is Monk (6)
[Thread 0] Monk (7) is fighting against Monk (8). The winner is Monk (8)
[Thread 0] Monk (6) is fighting against Monk (8). The winner is Monk (8)
[Thread 0] Monk (4) is fighting against Monk (8). The winner is Monk (8)
Monk (8) has won the battle!
```

4.2 Пример 2

Результат работы программы со следующими входными данными:

- размер массива = 16

```
[Thread 2] Monk (1) is fighting against Monk (2). The winner is Monk (2)
[Thread 1] Monk (3) is fighting against Monk (4). The winner is Monk (4)
[Thread 5] Monk (5) is fighting against Monk (6). The winner is Monk (6)
[Thread 1] Monk (7) is fighting against Monk (8). The winner is Monk (8)
[Thread 0] Monk (9) is fighting against Monk (10). The winner is Monk (10)
[Thread 2] Monk (11) is fighting against Monk (12). The winner is Monk (12)
[Thread 5] Monk (13) is fighting against Monk (14). The winner is Monk (14)
[Thread 2] Monk (15) is fighting against Monk (16). The winner is Monk (16)
[Thread 0] Monk (2) is fighting against Monk (4). The winner is Monk (4)
[Thread 0] Monk (6) is fighting against Monk (8). The winner is Monk (8)
[Thread 0] Monk (4) is fighting against Monk (8). The winner is Monk (8)
[Thread 0] Monk (10) is fighting against Monk (12). The winner is Monk (12)
[Thread 0] Monk (14) is fighting against Monk (16). The winner is Monk (16)
[Thread 0] Monk (12) is fighting against Monk (16). The winner is Monk (16)
[Thread 0] Monk (8) is fighting against Monk (16). The winner is Monk (16)
Monk (16) has won the battle!
```

4.3 Пример 3

Результат работы программы со следующими входными данными:

- размер массива = 65536

```
[Thread 4] Monk (60048) is fighting against Monk (60064). The winner is Monk (60064)
[Thread 4] Monk (60064) is fighting against Monk (60096). The winner is Monk (60096)
[Thread 4] Monk (60096) is fighting against Monk (60160). The winner is Monk (60160)
[Thread 4] Monk (60032) is fighting against Monk (60160). The winner is Monk (60160)
[Thread 5] Monk (60160) is fighting against Monk (60416). The winner is Monk (60416)
[Thread 5] Monk (59904) is fighting against Monk (60416). The winner is Monk (60416)
[Thread 1] Monk (60416) is fighting against Monk (61440). The winner is Monk (61440)
[Thread 1] Monk (59392) is fighting against Monk (61440). The winner is Monk (61440)
[Thread 7] Monk (61440) is fighting against Monk (65536). The winner is Monk (65536)
[Thread 0] Monk (8192) is fighting against Monk (16384). The winner is Monk (16384)
[Thread 0] Monk (24576) is fighting against Monk (32768). The winner is Monk (32768)
[Thread 0] Monk (16384) is fighting against Monk (32768). The winner is Monk (32768)
[Thread 0] Monk (40960) is fighting against Monk (49152). The winner is Monk (49152)
[Thread 0] Monk (57344) is fighting against Monk (65536). The winner is Monk (65536)
[Thread 0] Monk (49152) is fighting against Monk (65536). The winner is Monk (65536)
[Thread 0] Monk (32768) is fighting against Monk (65536). The winner is Monk (65536)
Monk (65536) has won the battle!
```