

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

**Программа, демонстрирующая многопоточную работу на  
C++ с использованием библиотеки OpenMP**

**Вариант 20**

Пояснительная записка

Исполнитель:

Студент группы БПИ198

Мелехин Денис Антонович

29.11.2020

Москва 2020

## Текст задания

*Задача о производстве булавок.* В цехе по заточке булавок все необходимые операции осуществляются тремя рабочими. Первый из них берет булавку и проверяет ее на предмет кривизны. Если булавка не кривая, то рабочий передает ее своему напарнику. Напарник осуществляет собственно заточку и передает заточенную булавку третьему рабочему, который осуществляет контроль качества операции. Требуется создать многопоточное приложение, моделирующее работу цеха. При решении использовать парадигму «производитель-потребитель».

## Описание решения

Программа генерирует массив pins из случайных чисел из полуинтервала  $[0; \text{GOOD\_CURVE} * 2)$ , который мы будем считать булавками. Первый рабочий (worker1) будет считать все булавки меньше GOOD\_CURVE искривленными. Для обмена булавками между рабочими будет использоваться буфер buffer12. Второй рабочий (worker2) будет затачивать все иголки. Я реализовал это как прибавление 100 к каждому элементу (т.е. Если у нас была не искривлённая булавка со значением A, то после второго рабочего ее значение будет  $A + 100$ ). Для обмена булавками между вторым и третьим рабочим используется буфер buffer23. Третий рабочий (worker3) проверяет качество заточки. Я реализовал это как проверку на то, что значение булавки  $\geq 120$ . После чего третий рабочий для каждой итоговой булавки выводит сообщение: "(worker3) One of final pins ( $\geq 120$ ) : {значение булавки после заточки}". Для демонстрации того, что программа действительно работает параллельно был оставлен вывод об промежуточных итогах каждого рабочего.

Сообщения:

(worker1) Pin with good curve ( $\geq \text{GOOD\_CURVE}$ ): {значение булавки} означает, что булавка была не искривлена и передана второму рабочему.

(worker1) Pin with bad curve ( $< \text{GOOD\_CURVE}$ ) to garbage: {значение булавки} означает, что булавка была искривлена и первый рабочий ее выкинул.

(worker2) Made {значение булавки} pin ( +100 ) показывают какая булавка стала после заточки (к ней прибавили 100)

(worker3) One of final pins ( $\geq 120$ ): {значение булавки} показывает булавки, которые прошли через всех рабочих

## Тестирование программы

```
ABC_task04 x
C:\Users\dmele\CLionProjects\ABC_task04\cmake-build-debug\ABC_task04.exe
Pins:
29 28 18 4 26 9 4 28 22 1
(worker1) Pin with good curve ( >= GOOD_CURVE): 29
(worker2) Made 129 pin ( +100 )
(worker3) One of final pins (>=120): 129
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with good curve ( >= GOOD_CURVE): 26
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with good curve ( >= GOOD_CURVE): 28
(worker2) Made 128 pin ( +100 )
(worker3) One of final pins (>=120): 128
(worker1) Pin with good curve ( >= GOOD_CURVE): 28
(worker2) Made 128 pin ( +100 )
(worker3) One of final pins (>=120): 128
(worker1) Pin with good curve ( >= GOOD_CURVE): 22
(worker2) Made 122 pin ( +100 )
(worker3) One of final pins (>=120): 122
(worker2) Made 126 pin ( +100 )
(worker3) One of final pins (>=120): 126
(worker1) Pin with good curve ( >= GOOD_CURVE): 18
(worker2) Made 118 pin ( +100 )
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1

Process finished with exit code 0
|
```

```
ABC_task04 x
C:\Users\dmele\CLionProjects\ABC_task04\cmake-build-debug\ABC_task04.exe
Pins:
11 7 24 9 9 11 2 24 9 16
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with good curve ( >= GOOD_CURVE): 24
(worker2) Made 124 pin ( +100 )
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with good curve ( >= GOOD_CURVE): 24
(worker2) Made 124 pin ( +100 )
(worker3) One of final pins (>=120): 124
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker3) One of final pins (>=120): 124
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with good curve ( >= GOOD_CURVE): 16
(worker2) Made 116 pin ( +100 )

Process finished with exit code 0
```

```
ABC_task04 x
C:\Users\dmele\CLionProjects\ABC_task04\cmake-build-debug\ABC_task04.exe
Pins:
0 11 17 19 20 13 9 11 25 5
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with good curve ( >= GOOD_CURVE): 17
(worker2) Made 117 pin ( +100 )
(worker1) Pin with good curve ( >= GOOD_CURVE): 20
(worker2) Made 120 pin ( +100 )
(worker3) One of final pins (>=120): 120
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with good curve ( >= GOOD_CURVE): 25
(worker2) Made 125 pin ( +100 )
(worker3) One of final pins (>=120): 125
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with good curve ( >= GOOD_CURVE): 19
(worker2) Made 119 pin ( +100 )

Process finished with exit code 0
```

```
ABC_task04 x
C:\Users\dmele\CLionProjects\ABC_task04\cmake-build-debug\ABC_task04.exe
Pins:
16 7 11 3 7 19 1 22 7 23
(worker1) Pin with good curve ( >= GOOD_CURVE): 16
(worker2) Made 116 pin ( +100 )
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with good curve ( >= GOOD_CURVE): 19
(worker2) Made 119 pin ( +100 )
(worker1) Pin with good curve ( >= GOOD_CURVE): 22
(worker2) Made 122 pin ( +100 )
(worker3) One of final pins (>=120): 122
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with good curve ( >= GOOD_CURVE): 23
(worker2) Made 123 pin ( +100 )
(worker3) One of final pins (>=120): 123

Process finished with exit code 0
```

```

C:\Users\dmele\CLionProjects\ABC_task04\cmake-build-debug\ABC_task04.exe
Pins:
5 3 10 25 6 22 22 11 27 16
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with good curve ( >= GOOD_CURVE): 25
(worker2) Made 125 pin ( +100 )
(worker3) One of final pins (>=120): 125
(worker1) Pin with good curve ( >= GOOD_CURVE): 22
(worker2) Made 122 pin ( +100 )
(worker3) One of final pins (>=120): 122
(worker1) Pin with good curve ( >= GOOD_CURVE): 16
(worker2) Made 116 pin ( +100 )
(worker1) Pin with good curve ( >= GOOD_CURVE): 22
(worker2) Made 122 pin ( +100 )
(worker3) One of final pins (>=120): 122
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): -1
(worker1) Pin with good curve ( >= GOOD_CURVE): 27
(worker2) Made 127 pin ( +100 )
(worker3) One of final pins (>=120): 127

Process finished with exit code 0

```

## Список литературы

1. <http://softcraft.ru/>
2. <https://iwomp2016.riken.jp/wp-content/uploads/2016/10/OpenMP-Tasking.pdf>
3. [http://rsuib.cc.rsu.ru/tutor/high\\_performance\\_computing/chapter3/page08.html](http://rsuib.cc.rsu.ru/tutor/high_performance_computing/chapter3/page08.html)
4. <https://www.viva64.com/en/a/0054/>

## Текст программы

```

#include <mutex>
#include <condition_variable>
#include <iostream>
#include <mutex>

```

```
#include <omp.h>
```

```
using namespace std;
```

```
const int PINS_NUM = 10;
```

```
const int GOOD_CURVE = 15;
```

```
const int BUFFER12_SIZE = 10;
```

```
const int BUFFER23_SIZE = 10;
```

```
struct Buffer {
```

```
    int size;
```

```
    int* ar;
```

```
    int index_put = 0;
```

```
    int index_take = 0;
```

```
    Buffer(int _size) {
```

```
        this->size = _size;
```

```
        ar = new int[this->size];
```

```
    }
```

```
    void put(int elem) {
```

```
        ar[index_put] = elem;
```

```
        index_put = (index_put + 1) % this->size;
```

```
    }
```

```
    int take() {
```

```
        int temp_answer = ar[index_take];
```

```
        index_take = (index_take + 1) % this->size;
```

```
        return temp_answer;
```

```
    }
```

```
};
```

```

int worker1_one(int pin, Buffer buffer) {
    if (pin >= GOOD_CURVE) {
#pragma omp critical
        cout << "(worker1) Pin with good curve ( >= GOOD_CURVE): " << pin << endl;
    }
    else {
        pin = -1;
#pragma omp critical
        cout << "(worker1) Pin with bad curve (<GOOD_CURVE) to garbage(-1): " << pin <<
endl;
    }

    buffer.put(pin);
}

```

```

void worker2_one(int pin, Buffer buffer) {
    if(pin == -1) {
        buffer.put(-1);
        return;
    }
    pin += 100;
#pragma omp critical
        cout << "(worker2) Made " << pin << " pin ( +100 )" << endl;

    buffer.put(pin);
}

```

```

void worker3_one(int pin) {
    if(pin == -1) return;
    if (pin >= 120) {
#pragma omp critical
        cout << "(worker3) One of final pins (>=120): " << pin << endl;
    }
}

```

```
}  
}
```

```
int main() {  
    srand((unsigned int)time(NULL));  
    // Исходные булавки для первого рабочего [0, PINS_NUM)  
    int* pins; // значения кривизны булавки из полуинтервала [0, GOOD_CURVE*2)  
  
    pins = new int[PINS_NUM];  
    cout << "Pins:" << endl;  
    for (int i = 0; i < PINS_NUM; i++) {  
        pins[i] = rand() % (GOOD_CURVE * 2);  
        cout << pins[i] << " ";  
    }  
    cout << endl;  
    int pin_x = 0;  
    int pin_y = 0;  
    int pin_z = 0;  
    int index = 0;  
    #pragma omp parallel num_threads(PINS_NUM)  
    {  
        Buffer buffer12(BUFFER12_SIZE);  
        Buffer buffer23(BUFFER23_SIZE);  
        #pragma omp task depend(in: pin_x) depend(out: pin_y)  
            worker1_one(pins[index++], buffer12);  
  
        #pragma omp task depend(in: pin_y) depend(out: pin_z)  
            worker2_one(buffer12.take(), buffer23);  
  
        #pragma omp task depend(in: pin_z) depend(out: pin_z)  
            worker3_one(buffer23.take());  
    }  
}
```



