

ГУАП

КАФЕДРА №14

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

ДОЦ., К.Т.Н.

должность

подпись, дата

К.А. Курицин

инициалы, фамилия

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

по курсу: ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛИ

СТУДЕНТЫ ГРУППЫ 1441

подпись, дата

А.А. Протасов, Н.А. Соловьев

инициалы, фамилия

Санкт-Петербург 2016

# 1. Постановка задачи

Реализовать класс синхронизации семафор. Класс инкапсулирует работу объекта синхронизации, который содержит счетчик между нулем и заданным максимальным значением. Значение счетчика увеличивается каждый раз, когда поток завершает ожидание освобождения семафора, и уменьшается, когда поток освобождает семафор. В случае, если значение счетчика достигает максимального значения, потоки ожидают освобождения семафора. Если указано максимальное значение счетчика, равное 1, то семафор функционально должен вести себя как критическая секция (critical section).

## 2. Листинги

---

```
1  #include <iostream>
2  #include <chrono>
3  #include <mutex>
4  #include <thread>
5  #include <vector>
6
7  using namespace std;
8  using ::this_thread::sleep_for;
9  using ::chrono::seconds;
10
11 class Semaphore
12 {
13     uint32_t max_count;
14     uint32_t count;
15     mutex CS, CS_2;
16
17 public:
18     Semaphore(int max_count) : max_count(max_count), count(0){}
19     ~Semaphore(){}
20     void lock(){
21         CS.lock();
22         if(count < max_count)
23         {
24             count++;
25             cout << "\033[31mCounter start " << count << "\033[0m" << endl;
26             CS.unlock();
27         }
28         else
29         {
30             count++;
31             CS.unlock();
32             CS_2.lock();
33             cout << "Counter start " << count << endl;
34             CS_2.unlock();
35         }
36     }
37     void unlock()
38     {
39         CS.lock();
40         //cout << "Counter finish " << count << endl;
41         count--;
```

```
42         CS.unlock();
43     }
44 };
45
46 Semaphore *sema;
47
48 void exec(){
49     sema->lock();
50     sleep_for(seconds(1));
51     sema->unlock();
52 }
53
54 int main()
55 {
56     vector<thread>pt;
57     sema = new Semaphore(1);
58
59     for(int i = 1; i <= 2; i++){
60         pt.push_back(thread(exec));
61     }
62     for(auto &x : pt) x.join();
63     return 0;
64 }
```

---

### 3. Примеры работы программы

```
Counter start 1  
Counter start 2  
Counter start 3  
Counter start 4  
Counter start 5  
Counter start 6  
Counter start 7  
Counter start 8  
Counter start 10  
Counter start 10  
Press <RETURN> to close this window...  
■
```

Семафоров 5, потоков 10

```
Counter start 1  
Counter start 2  
Counter start 3  
Counter start 4  
Counter start 5  
Counter start 6  
Counter start 7  
Counter start 8  
Counter start 9  
Counter start 10  
Counter start 11  
Counter start 12  
Counter start 13  
Counter start 14  
Counter start 15  
Counter start 16  
Counter start 17  
Counter start 18  
Counter start 19  
Counter start 20  
Press <RETURN> to close this window...  
■
```

Семафоров 10, потоков 20