

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

М.А. Конаков, А.В. Рещенко

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 <windows.h>
3  #include <conio.h>
4  using namespace std;
5  class Semaphore{
6      CRITICAL_SECTION critical_sec;
7      int max_count;
8      int count;
9      HANDLE event;
10 public:
11     Semaphore(int max_count) : max_count(max_count), count(0){
12         InitializeCriticalSection(&critical_sec);
13         event=CreateEvent(NULL,TRUE,FALSE,NULL);
14     }
15     ~Semaphore(){
16         DeleteCriticalSection(&critical_sec);
17         CloseHandle(event);
18     }
19     void lock(){
20         EnterCriticalSection(&critical_sec);
21         if(count<max_count){
22             count++;
23             cout<<"(lock())critical_sec is unlocked,"
24                  "total number of threads in semaphore:"<<count<<endl;
25             LeaveCriticalSection(&critical_sec);
26             return;
27         }
28         count++;
29         LeaveCriticalSection(&critical_sec);
30         cout<<"(lock())critical_sec is unlocked,"
31              "total number of threads:"<<count<<endl;
32         WaitForSingleObject(event,INFINITE);
33     }
34     void unlock(){
35         EnterCriticalSection(&critical_sec);
36         count--;
37         if(count!=0){
38             SetEvent(event);
39             cout<<"(unlock())critical_sec is locked,"
40                  "total number of threads:"<<count<<endl;
41         }
```

```
42         LeaveCriticalSection(&critical_sec);
43     }
44 };
45
46 Semaphore*sema;
47
48 DWORD WINAPI foo(PVOID){
49     sema->lock();
50     Sleep(1000);
51     sema->unlock();
52     return 0;
53 }
54
55 int main(){
56     sema=new Semaphore(6);
57     HANDLE thread[12];
58     for(int i=0;i<12;i++){
59         thread[i]=CreateThread(NULL,0,foo,NULL,0,NULL);
60     }
61     WaitForMultipleObjects(12,thread,TRUE,INFINITE);
62     for(int i=0;i<12;i++){
63         CloseHandle(thread[i]);
64     }
65     cout<<endl<<"Done. Press any key to exit."<<endl;
66     _getch();
67     return 0;
68 }
```

3. Примеры работы программы

```
<lock(>>critical_sec is unlocked, total number of threads in semaphore:1
<lock(>>critical_sec is unlocked, total number of threads in semaphore:2
<lock(>>critical_sec is unlocked, total number of threads in semaphore:3
<lock(>>critical_sec is unlocked, total number of threads in semaphore:4
<lock(>>critical_sec is unlocked, total number of threads in semaphore:5
<lock(>>critical_sec is unlocked, total number of threads in semaphore:6
<lock(>>critical_sec is unlocked, total number of threads:7
<lock(>>critical_sec is unlocked, total number of threads:10
<lock(>>critical_sec is unlocked, total number of threads:8
<lock(>>critical_sec is unlocked, total number of threads:12
<lock(>>critical_sec is unlocked, total number of threads:9
<lock(>>critical_sec is unlocked, total number of threads:11
<unlock(>>critical_sec is locked, total number of threads:11
<unlock(>>critical_sec is locked, total number of threads:10
<unlock(>>critical_sec is locked, total number of threads:9
<unlock(>>critical_sec is locked, total number of threads:8
<unlock(>>critical_sec is locked, total number of threads:7
<unlock(>>critical_sec is locked, total number of threads:6
<unlock(>>critical_sec is locked, total number of threads:5
<unlock(>>critical_sec is locked, total number of threads:4
<unlock(>>critical_sec is locked, total number of threads:3
<unlock(>>critical_sec is locked, total number of threads:2
<unlock(>>critical_sec is locked, total number of threads:1
<unlock(>>critical_sec is locked, total number of threads:0
Done.Press any key to exit.
```

Семафоров 6, потоков 12