Белорусский государственный университет
Факультет прикладной математики и информатики
Кафедра технологии программирования
Доц. Побегайло А.П.

Лабораторная работа №4. (4 часа: 01.04.19 – 15.04.19)

Тема: “Семафоры. Разработка классов, безопасных для потоков”.

Задача. Написать класс для управления доступом параллельных потоков к кольцевой очереди.

Требования к реализации.

1. Интерфейс класса для управления доступом параллельных потоков к кольцевой очереди:

```
class SyncQueue
{
public:
        // конструктор
        SyncQueue(int nSize);         // nSize - размер очереди
        // деструктор
        ~ SyncQueue();
        // функции для доступа к очереди
        void  insert(int nElement);   // добавить элемент в хвост очереди
        int   remove();               // удалить головной элемент очереди
};
```

2. Кольцевая очередь реализуется массивом, элементы которого имеют тип `int`. Размер кольцевой очереди задается в конструкторе.

3. Если поток вызывает метод `insert`, а кольцевая очередь оказывается полной, то метод `insert` должен переводить этот поток в состояние ожидания до удаления из кольцевой очереди, хотя бы одного элемента другим потоком.

4. Если поток вызывает метод `remove`, а кольцевая очередь оказывается пустой, то метод `remove` должен переводить этот поток в состояние ожидания до записи в кольцевую очередь, хотя бы одного нового элемента другим потоком.

5. Работа с примитивами синхронизации должна быть организована внутри объектов класса.

Дополнительные требования.

Для тестирования класса `SyncQueue` написать программу для консольного процесса, который состоит из потока `main` и нескольких потоков `consumer` и `producer`.

Поток `main` должен выполнять следующие действия:

- создать объект кольцевой очереди, размер очереди вводится пользователем с клавиатуры;
- ввести с клавиатуры количество потоков `producer` и количество потоков `consumer`, которые он должен запустить;
- создать требуемое количество потоков `producer` и `consumer`;
- ждать сигнал на готовность к работе от всех потоков `producer` и `consumer`;
- подать сигнал на начало работы потоков `producer` и `consumer`;
- завершить свою работу после окончания работы всех потоков `producer` и `consumer`.

Поток `producer` должен выполнять следующие действия:

- запрашивает с консоли количество и значение целого числа, которое он должен производить;
- подает сигнал на готовность к работе;
- ждет сигнал на начало работы;
- циклически выполняет следующие действия (цикл по количеству производимых чисел):
  - добавляет в кольцевую очередь произведенное целое число;
  - выводит на консоль сообщение: "Произведено число: `N` ", где `N` - значение числа, помещенного в очередь.
  - спит 15 мс.

Поток `consumer` должен выполнять следующие действия:

- запрашивает с консоли количество потребляемых чисел;
- подает сигнал на готовность к работе;
- ждет сигнал на начало работы;
- циклически извлекает из кольцевой очереди целые числа с интервалом в 15 мс (цикл по количеству потребляемых чисел);
- при извлечении числа из кольцевой очереди, выводить на консоль сообщение: "`\t`Употреблено число `N` ", где `N` - значение числа, извлеченного из очереди.