

Лабораторная 15. Многопоточность III

Сдать до 21.12

Задача о медведе и пчелах (4 балла)

Есть N пчел и один медведь. Все они пользуются одним единственным горшком меда, вмещающим X порций меда.

Сначала горшок пустой. Пока горшок не наполнится, медведь спит, потом съедает весь мед и засыпает. Каждая пчела многократно собирает по одной порции меда и кладет ее в горшок. Сбор меда каждый раз занимает случайное количество времени. Пчела, которая приносит последнюю порцию меда и заполняет горшок, будит медведя.

Смоделируйте поведение пчел и медведя с использованием задач (**Tasks**).

Бараны и волк (4 балла)

Создайте игру, где будут 3 барана и 1 волк. При совпадении координат волка с бараном баран исчезает. При совпадении координат баранов появляется новый баран. Все движутся хаотически. Поле игры – матрица $N \times N$. Пусть каждая сущность (бараны и волк) будут реализованы в отдельном потоке. Все двигаются со случайной задержкой в случайном направлении на одну клетку (всего может быть 8 направлений). Продемонстрируйте работу программы.

Задача о спящем парикмахере (4 балла)

В парикмахерской имеется одно кресло для стрижки и N кресел для ожидающих посетителей. Если нет посетителей, парикмахер спит. Если приходит посетитель и кресло для стрижки свободно, посетитель садится в него и парикмахер начинает его стричь. В противном случае посетитель садится в кресло для ожидающих. Если все кресла заняты, посетитель уходит. Реализуйте данную задачу и создайте несколько тестов, имитирующих различные ситуации в парикмахерской.

Многопоточный Merge Sort (4 балла)

Реализуйте сортировку слияниями, модифицированную для работы на нескольких потоках.

У вас есть массив целых чисел длины n и r потоков. Вам необходимо разбить массив на части и отсортировать каждую из частей в отдельном потоке. Далее из отсортированных отдельных частей необходимо получить общий массив в отсортированном виде. Все потоки должны работать с одним и тем же массивом, получая диапазон индексов для сортировки (без создания отдельных подмассивов). Введите искусственную задержку сортировки внутри каждого потока.

1. Реализуйте слияние отсортированных подмассивов в общий отсортированный массив в главном потоке, сперва дождавшись завершения каждого из r потоков.
2. Предположим, что массив очень большой и что потоки заканчивают работу в разное время. Главный поток начинает слияние отсортированных подмассивов в промежуточный массив по мере того, как сортирующие потоки заканчивают свою работу. Если ни один из потоков не готов, то главный поток засыпает до того момента, пока не закончит работу один из сортирующих потоков. Когда все потоки завершаются, главный поток заканчивает слияние и возвращает отсортированный массив.

Вы можете вместо потоков использовать в своем решении задачи (**Tasks**).

MyWaitAll (4 балла)

Реализуйте с помощью стандартных примитивов аналог WaitAll.

Пример сигнатуры класса

```
public class CMyWaitAll : IDisposable
{
    public CMyWaitAll(Int32 atomsNumber);
    public void SetAtomSignaled(Int32 atomId);

    public bool Wait(TimeSpan timeout);
    public void Dispose();
}
```

При создании указывается кол-во atomsNumber. AtomId могут быть в пределах от 0 до atomsNumber включительно. SetAtomSignaled переводит указанный atomId в сигнальное состояние. Wait блокируется до тех пор, пока все atoms не будут в сигнальном состоянии. Выделяемые ресурсы должны освободиться в методе Dispose. Реализовать обработку различных передаваемых параметров. Разрешается использовать только AutoResetEvent, ManualResetEvent, Mutex, Semaphore, lock(). Не должно быть ограничения в 64 atomsNumber.

Реализуйте проверку работы создаваемого класса в различных ситуациях.