

## Курс: Системное программирование

### Модуль 4. Синхронизация

#### Тема: События. Критические секции

##### Задание 1

Создайте приложение, использующее механизм событий. Создайте в коде приложения несколько потоков. Первый поток генерирует и сохраняет в файл некоторое количество пар чисел. Второй поток ожидает завершения генерации, после чего подсчитывает сумму каждой пары. Результат записывается в файл. Третий поток тоже ожидает завершения генерации, после чего подсчитывает произведение каждой пары. Результат записывается в файл.

##### Задание 2

Создайте имитацию работы автобусной конечной остановки за день. Количество автобусов и их размер по величине перевозимых пассажиров фиксирован. Число людей на остановке выбирается случайным образом. Новое полученное значение нужно прибавлять к уже существующему количеству на остановке. При приходе автобуса число людей на остановке уменьшается, так как автобус заполняется пассажирами. Однако, количество пассажиров в автобусе не может быть больше максимального значения. Люди, которые не поместились в автобус

остаются на конечной и ждут следующего автобуса. В рамках этой задачи считаем, что по маршруту, у нас ездит автобус только с одним номером. Например, автобус номер 175. Используйте механизм потоков и событий. Если вам необходимо, используйте и другие механизмы синхронизации.

## Задание 3

Добавьте возможность работы автобусов с разными номерами. Каждый пассажир приходит на остановку с желанием сесть в автобус определенного номера. Если приходит автобус другого номера, пассажир остаётся на остановке.

## Задание 4

Модифицируйте приложение, сделав остановку не конечной. Это значит, что автобус может приехать на остановку с каким-то количеством пассажиров на борту.