

Текстовый вариант видеоурока из предыдущего шага

Теперь мы с вами живем в мире Go, где горутины могут выполняться параллельно. Мы даже видели, что есть возможность запускать горутины на разных процессорах. Мы рассмотрели каналы, которые помогают отправлять сообщения нашим горутинам и выстраивать процесс общения между ними. Сейчас мы разберем немножко теорию примитивов синхронизации, и какие еще есть инструменты, встроенные в язык, которые позволяют нам синхронизировать наши данные.

При изучении параллельного программирования, чаще всего сталкиваются с такими понятиями, как мьютекс и семафор. Это основные примитивы синхронизации. Давайте разберемся, что они из себя представляют.

Семафор можно представить себе в виде счетчика, который занимается подсчетом количества потоков, например, и разрешает какому-то количеству доступ к ресурсам. То есть он может сказать, что в данный момент времени к этому ресурсу могут получить доступ всего 2 потока.

Мьютекс, в свою очередь, немножко отличается. Мьютекс блокирует ресурс для одного потока или одного пользователя, скажем так. Его можно сравнить с дверью, которую можно закрыть только изнутри. То есть кто-то один заходит в эту дверь, закрывает ее, внутри делает все, что ему нужно, потом открывает и выходит. В эту дверь можно входить только по одному. Ну, стоит тут сказать, что мьютекс — это частный случай семафора, поэтому, в принципе, они похожи друг на друга.

С помощью реализации данных механизмов мы, например, можем, запретить горутинам выполнять функцию, пока эту функцию выполняет какая-то другая горутина. Тем самым, мы предотвращаем ситуацию гонки за одни и те же ресурсы, так как у нас горутины выполняются параллельно.