

Дедлоки

Давайте вспомним из терминологии в начале лекции три понятия: starvation или ресурсное голодание, deadlock – взаимная блокировка и livelock. С ресурсным голоданием в целом понятно: операционная система должна как-то следить за тем, чтобы все были «накормлены». Процессы постоянно меняются, постоянно появляются новые, постоянно изменяются условия, но не должен быть такой процесс, который постоянно ждет и никак не получает своей порции ресурсов. Решений может быть много, зависит, естественно, всё от ситуации и от платформы – того, для чего эта операционная система сделана.

Задачи, которые встают перед инженерами операционной системы очень близки к нашим повседневным задачам. И **starvation** – это ни что иное, как голодание в семье. Если, допустим, у мамы 12 детей и пусть они кушают в разное время и всё время хотят разных вещей – нужно сделать так, чтобы никто из них не умер от голода. Можно по-разному это сделать. Можно присоединить какой-то датчик каждому ребенку, чтобы если он находится без еды больше, допустим, 8 часов, то нужно обязательно его покормить. Может быть такого датчика не нужно, потому что сам процесс кормежки всегда обеспечивает всех едой и так далее (зависит, опять же, от ситуации, от свойств детей и так далее). В целом, можно придумать много решений этой проблемы

С **livelock** не то, чтобы все понятно, но такие проблемы тоже можно обнаружить.

Поговорим про **deadlock**, про взаимную блокировку. Вспомним тот не плохой пример из дорожного движения. У нас есть пробка на перекрестке дорог где оба потока переплелись, и никто не может двигаться, потому что, чтобы один поток подвинулся нужно, чтобы второй уехал, а второму нужно, чтобы первый уехал, и они ждут друг друга и ни что им не может помочь.