

LAB 3 "Электростанция"

(взаимодействие между нитями с помощью сообщений)

1. Электростанция состоит из следующих элементов: хранилище топлива (1 шт.), транспортное средство (1 шт.), котлы (4 шт.). Элементы станции работают параллельно, каждый по своей программе (что может быть реализовано с помощью нитей). Транспортное средство доставляет топливо из хранилища к котлам. Топливо имеет различные марки (от 1 до 10). Топливо марки 10 горит в котле 10 с (условно), в то время как топливо марки 1 горит всего 1 с. Необходимо написать программу, моделирующую работу электростанции и показывающую на экране процесс ее функционирования.

Методические указания

1. Начните с написания на бумаге словесных алгоритмов работы всех элементов станции. Первый алгоритм составляется более или менее произвольно. Алгоритмы же работы других элементов должны по своей логике "стыковаться" с ним. Если вы чувствуете, что схема будет работать, то можно переходить к следующему шагу. Иначе нужно пересмотреть все алгоритмы, начиная с первого. Может оказаться проще начать с другого элемента станции.
2. Теперь подумайте, где нужно использовать Send, а где Receive или Reply (исходя из принципов работы протокола Send-Receive-Reply). Определите, сколько каналов и соединений вам потребуется. Внесите все это в ваши словесные описания алгоритмов.
3. Теперь переходите к написанию кода. Все должно получиться.
4. Все действия элементов станции должны производиться с временными задержками. Иначе мы не увидим динамики. Например, если транспортное средство будет доставлять топливо мгновенно (что в реальности не возможно), то все будет происходить так, как если бы мы вообще исключили его из схемы. Такой вариант можно позволить только на каком-то промежуточном этапе разработки программы.
5. Любое изменение состояния элемента станции должно как-то отображаться на экране. Тогда мы сможем видеть всю логику процесса. Для наглядности рекомендуется даже развертывать во времени некоторые состояния. Например, транспортное средство в состоянии движения может быть изображено с помощью перемещающейся по экрану фигуры.
6. Чтобы все лучше понять и прочувствовать, запустите ер.demo. Чтобы понять логику работы отдельного элемента, нужно смотреть только на него. Способ изображения элементов станции может быть совершенно другой. Вы также можете выбрать другую схему размещения элементов. Важно только, чтобы путь от хранилища к котлу был индивидуален и требовал некоторого времени для прохождения (это потребуется при последующих модификациях вашей программы).

LAB 3 "Электростанция-2(3)"

2. А теперь добавьте второе транспортное средство.

3. (использование импульсов) Регулируя скорости работы элементов электростанции, вы можете создать ситуацию, когда котлы будут простаивать из-за низкой скорости подвоза топлива. Создайте такую ситуацию. Теперь сделайте так, чтобы топливо подвозилось к котлам заранее, до момента их полной остановки. Это можно реализовать, если котлы будут сообщать о том, что топливо скоро кончится (например, его осталось на 2 с работы). Ясно, что котлы могут это сделать с помощью импульса, т.к. обычное сообщение их заблокировало бы, в то время как они должны продолжать работать.

LAB 3 "Электростанция-4"

4. А теперь сделайте сетевой вариант разработанной программы. Пусть теперь хранилище работает на одной машине, котлы на другой, а транспортные средства переносят топливо между этими двумя машинами. Вам понадобится разделить вашу программу на две, которые вы будете запускать на разных узлах сети. Чтобы не надоедать соседу с просьбами о запуске вашей программки, которая к тому же будет постоянно подвешивать его компьютер, отладку можно вести на своем компьютере. Напомним, что имя узла содержится в переменной HOSTNAME.