Санкт-Петербургский Национальный Исследовательский Университет

Информационных Технологий, Механики и Оптики

Кафедра Информатики и Прикладной Математики

Распределенные системы

Лабораторная работа №1

«Разработка и реализация консервативного алгоритма синхронизации»

Работу выполнил

студент группы Р4115

Назукин Дмитрий

Санкт-Петербург

2017

**Постановка задачи**

Разработать консервативный алгоритм синхронизации с нулевыми сообщениями для имитации взаимодействия покупателя, банка и магазина:

Предположим, что имитационная модель представляет собой совокупность трёх взаимодействующих процессов.

Покупатель в магазине делает покупку в кредит. Магазин сообщает о сделке банку, в котором открыт счёт покупателя. Покупатель приходит в банк, чтобы снять со счёта некоторую сумму. Если деньги на счёте есть, он получает эту сумму, если нет, то ему отказывают и рекомендуют пополнить счёт.

**Описание консервативного алгоритма с нулевыми сообщениями**

Алгоритм предполагает:

* Топология процессов, которые посылают сообщения друг другу, известна и фиксирована.
* Каждый логический процесс посылает сообщения с неубывающими временными метками.
* Коммуникационная среда обеспечивает доставку сообщений в том порядке, в котором они были посланы.

Исходя из этих предположений, можно сделать следующее заключение:

* Временная метка сообщения, которое было получено последним на линии связи, является нижней границей временных меток (LBTS) всех будущих сообщений, передаваемых по этой линии связи.
* Нижняя временная метка (LBTS) логического процесса определяется как минимальная из всех нижних временных меток, полученных процессом по всем входным линиям связи.

Сообщения каждой линии связи находятся в очереди, которая обрабатывается по дисциплине FIFO. Очередь упорядочена также в соответствии с временными метками. Каждая линия связи имеет своё локальное время, которое равно временной отметке первого сообщения в очереди (если таковые имеются) или временной отметке последнего принятого сообщения. Все события, которые планирует сам процесс для себя, находятся в другой очереди. Алгоритм периодически выбирает линию связи с наименьшим временем и, если в ней есть события, то он обрабатывает это событие. Если очередь пуста, то процесс блокируется. Процесс никогда не блокируется при проверке состояния очереди сообщений, которые он планирует для себя. Итак, этот алгоритм гарантирует, что каждый логический процесс будет обрабатывать события в хронологическом порядке.

Для избегания тупиков процессы отсылают нулевые сообщения по всем выходным дугам после обработки очередного события. Нулевое сообщение служит дополнительной информацией для того, чтобы определить очередное безопасное событие.

Нулевое сообщение обрабатывается как обычное сообщение за исключением того факта, что ни одна переменная процесса и никакое событие не будут запланированы. Локальное время логического процесса продвигается до временной отметки нулевого сообщения.

**Реализация алгоритма**

Основная часть алгоритма реализована в следующих строчках. Процесс (в данном случае Банк) выбирает из входящих очередей (от покупателя и из магазина) сообщение с наименьшей временной меткой – метод getMinMessage(). Причем если хотя бы одна из очередей пуста, процесс не выбирает сообщение. Далее процесс банка обрабатывает выбранное сообщение и отправляет «нулевые сообщения» по всем выходным дугам.



Каждое сообщение наследуется от абстрактного класса Message и содержит временную метку, определяющую порядок обработки данного сообщения.



В данной имитационной модели покупатель имеет определенную сумму денег в банке (20000 у.е.), совершает покупку в магазине на эти деньги, а потом приходит в банк чтобы снять деньги.



Результат работы программы с использованием консервативного алгоритма с нулевыми сообщениями:



**Заключение**

В ходе выполнения лабораторной работы реализован консервативный алгоритм синхронизации с нулевыми сообщениями на примере взаимодействия покупателя, магазина и банка. Данный алгоритм позволяет выполнять запланированные операции в хронологическом порядке, однако недостатком алгоритма является большое количество «нулевых сообщений».