

Лекция №7

Раздел 2. Модельная динамика систем с дискретными событиями

Существует несколько концепций дискретно-событийного моделирования, определяющих различные подходы к декомпозиции динамических процессов, протекающих в моделируемой системе.

1. Концепция событий.
2. Концепция состояний.
3. Концепция параллельных процессов.

4 Процессно-ориентированный подход к моделированию систем

4.1 Понятие процесса

Понятие процесса обычно связывается с изменением состояния системы во времени. Применительно к классу дискретных систем, процесс удобно интерпретировать как перемещение по графу состояний системы некоторого абстрактного элемента (например, указателя текущего состояния системы, в схеме транзактов — транзакта и т.д.). Понятие процесса применительно к алгоритмам имитации может быть определено в рамках схемы событий. Такой подход связан с разбиением множества событий $E = \{e_1, e_2, \dots, e_k\}$, возникающих в системе, на подмножества $\{E_1, E_2, \dots, E_k\}$ (может быть пересекающиеся), каждое из которых ассоциируется с определенным процессом $\{PR_1, PR_2, \dots, PR_k\}$. Возникновение события $e_1 \in E_k$ в некоторый момент времени означает активизацию процесса PR_k , т.е. в течение временного интервала между метками ближайших событий $e_l \in E_k$ и $e_m \in E_k$ процесс PR_k «ждет» активизации. В такой схеме внутри того или иного подмножества событий можно имитировать переход системы из одного состояния в другое за счет выполнения определенных действий над атрибутами динамических объектов, определенных в системе.

Процесс — это ориентированная во времени последовательность событий, наступление которых приводит к изменению состояния. Понятие процесса является концептуально более емким, чем понятие события или состояния. Состояние — это моментальный снимок процесса, а событие — управляющее воздействие, заставляющее процесс изменить состояние.

Особое значение в концепции параллельных процессов приобретают вопросы синхронизации их взаимодействия. Основной аппарат такой синхронизации — события, управляемые интеррогативно.

4.2 Схема слабо связанных процессов.

Интеррогативное управление специального вида с помощью сигналов

С помощью схемы слабо связанных процессов организуется взаимодействие квазипараллельных процессов. Каждый процесс представляет собой последовательную программу, выполнение нескольких таких программ осуществляется за счет передачи управления между ними.

Термин «слабо связанные процессы» означает, что связь между ними осуществляется через монитор моделирования, причем вместо функций условий для интеррогативного управления используются объекты специального вида — сигналы синхронизации. Объект «сигнал» может находиться в одном из двух состояний: «сигнал послан» и «сигнал не послан». Посылка сигнала рассматривается как выполнение условия

CN = (сигнал послан?) = TRUE. На уровне реализации сигнал связан с множеством процессов, ожидающих этот сигнал (Рисунок 16).

Сигнал



Рисунок 16 – Объект «сигнал»

Все существующие в системе процессы представляются дескрипторами (Рисунок 17), которые связаны в кольцевой список (Рисунок 18).

Дескриптор процесса

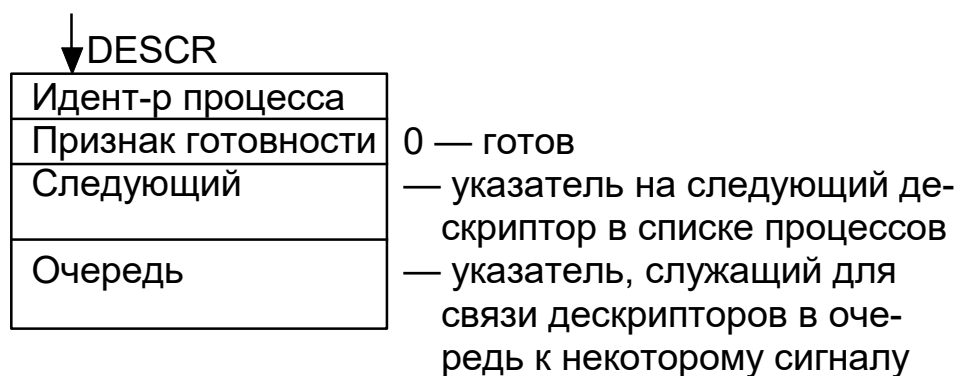


Рисунок 17 – Структура дескриптора процесса

Кольцо готовности процессов

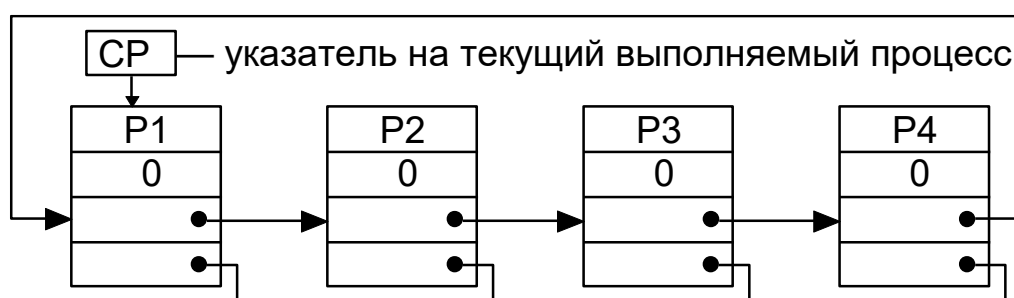


Рисунок 18 – Кольцо готовности процессов

Признак готовности процесса принимает два значения: 0 – готов, 1 – не готов.

Каждый процесс может посылать сигнал (оператор SEND) и ожидать послышки сигнала (оператор WAIT). Управление пересылкой сигналов от одного процесса к другому осуществляет монитор моделирования. Когда текущий процесс выдает команду WAIT(*S*), его признак готовности устанавливается в 1, он включается в конец очереди к сигналу *S*,

текущим становится процесс, у которого признак готовности 0. Когда текущий процесс подает команду $SEND(S)$, то если очередь к S пуста, ничего не происходит, если же очередь не пуста, из нее удаляется первый процесс, признак готовности у него устанавливается в 0, и он становится текущим. Процесс, выдавший команду $SEND$, перестает быть текущим, но признак готовности у него сохраняется 0. Одновременно могут существовать очереди к нескольким сигналам, но они не пересекаются. Каждый процесс может ожидать послылки единственного сигнала.

Схема синхронизации слабо связанных процессов с помощью сигналов показана на Рисунке 19.

Сигнальная синхронизация слабо связанных процессов

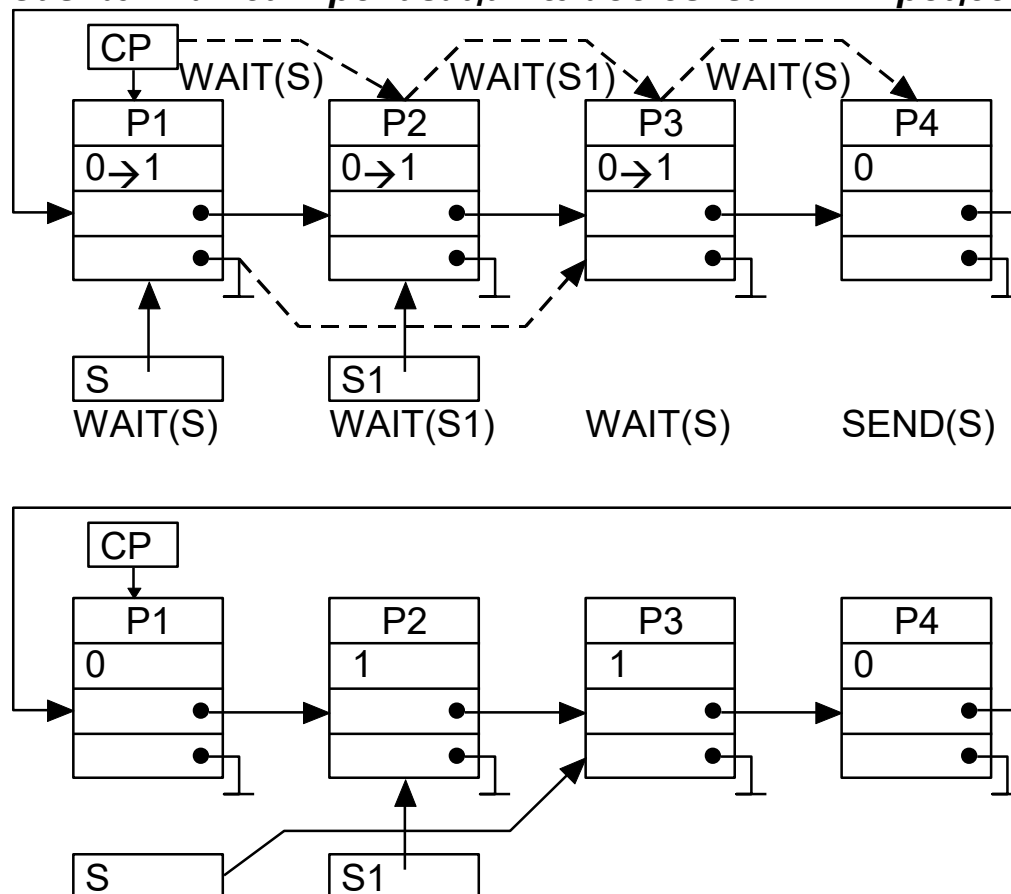


Рисунок 19 – Схема синхронизации слабо связанных процессов с помощью сигналов

Взаимодействие квазипараллельных процессов организуется по *схеме сопрограмм*, когда тело процесса выполняется не с самого начала, а с той точки, где выполнение было приостановлено (Рисунок 20).

Управление пересылкой сигналов между процессами

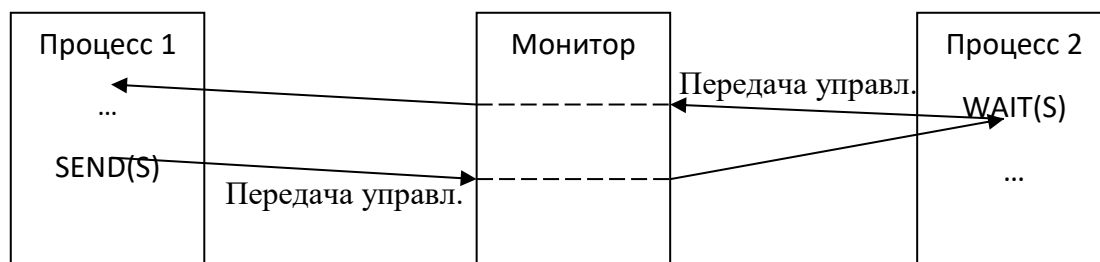


Рисунок 20 – Схема управления пересылкой сигналов между процессами

Пример. Взаимодействие двух процессов «Поставщик» и «Потребитель» происходит с использованием общего буфера сообщений. Тела процессов написаны на языке Модуля 2. Кольцевой буфер организован с использованием массива.

```
IMPLEMENTATION MODULE BUFFER;
Const N=128;   (*размер буфера*)
Var n:[0..N];  (*текущее число элементов в буфере*)
    НЕПОЛОН: SIGNAL;    (*n<N*)
    НЕПУСТ: SIGNAL;     (*n>0*)
    IN, OUT: [0..N-1];  (*индексы*)
    BUF: array[0..N-1] of char; (*буфер*)

procedure ПОМЕСТИТЬ(x:char);
begin
    If n=N then WAIT (НЕПОЛОН);
    n:=n+1; BUF[IN]:=x;
    IN:=(IN+1)mod N; SEND (НЕПУСТ);
end;

procedure ИЗВЛЕЧЬ (var x:char);
begin
    If n=0 then WAIT(НЕПУСТ);
    n:=n-1; x:=BUF[OUT];
    OUT:=(OUT+1)mod N;
    SEND (НЕПОЛОН);
end;

begin
    n:=0; IN:=0; OUT:=0;
    INITSIGNAL (НЕПОЛОН);
    INITSIGNAL (НЕПУСТ);
end.
```

4.3 Концепция транзактов как частный случай схемы параллельных процессов

Одним из средств описания параллельных динамических процессов, имитируемых движением абстрактного элемента-транзакта, по графу состояний системы, является **схема транзактов**. Каждый транзакт движется в пространстве и во времени. Путь движения транзакта – абстракция, рассматриваемая как последовательность секций, называемых блоками. Каждый блок определяется специальными свойствами, например, «начало движения», «останов», «задержка на время», «задержка до выполнения условия» и т.п. Выбирая блоки в соответствии со своими представлениями о закономерностях функционирования моделируемой системы, пользователь конструирует последовательность блоков – модель пути движения транзакта.

Транзакт – это абстрактный динамический объект, который может продвигаться по программе модели, останавливаться, уничтожаться. Транзакт может быть динамически введен в модель и выведен из модели. Каждый транзакт имеет свои собственные атрибуты, например, «время рождения», «длительность пребывания в системе» и т.п.

Структура динамического объекта «транзакт» показана на Рисунке 21.

идентификатор транзакта (номер)
время рождения
продолжительность пребывания в модели
идентификатор блока, в котором находится транзакт
параметры (до 100)
ссылка на соседний транзакт в семействе




Рисунок 21 – Структура динамического объекта «транзакт»

Схема транзактов строится на событийной основе. В календарь собраны транзакты, управляемые императивно, т.е. по времени (*список будущих событий*), транзакты, управляемые интеррогативно, т.е. по условиям, собраны в *список текущих событий*. Понятие «событие» остается прежним: изменение состояния системы, происходящее мгновенно во времени. Однако концепция транзакта как основного элемента динамического процесса связывает каждое событие с передвижкой транзакта по блокам модели пути, и состояние моделируемой системы в каждый момент времени характеризуется тем, *где (в каком блоке)* находится транзакт, и тем, *какую часть модели пути* он прошел.

Передвижка транзакта есть пара событий: «транзакт вошел в блок» («IN») и «транзакт вышел из блока» («OUT»). Движение транзакта по модели пути можно представить как поток событий «IN» и «OUT». Событие «OUT» и следующее событие «IN» в схеме транзактов рассматриваются как одновременные, поэтому поток событий следует представить в виде:

«**OUT**-IN»; «OUT-IN»; «OUT-IN»; ... ; «OUT-**IN**».

OUT – событие «транзакт вышел из особого (первого) блока, являющегося блоком генерации транзактов».

IN – событие «транзакт вошел в особый (последний) блок, являющийся блоком уничтожения транзактов».

Транзакт не может войти в блок генерации и не может выйти из блока уничтожения, транзакт не может находиться между блоками, поэтому движение транзакта – это серии переходов от блока к блоку. Если транзакт не может немедленно войти в блок, он находится в специальной списковой структуре монитора моделирования. В модели могут присутствовать несколько транзактов одновременно, причем каждый из них инициирует в модельном времени свой цикл действий, свой динамический процесс. В компьютерном времени подобные процессы выполняются квазипараллельно, разделяя время центрального процессора.

Блок генерации может генерировать регулярные или стохастические потоки транзактов. Нерегулярные стохастические потоки порождают две основные проблемы: появление очередей и появление тупиков.

4.4 Модели очередей, возникающих при задержке движения транзакта

Возникновение очередей транзактов является результатом многих причин, среди них основные – типы потоков событий, интенсивности потоков, структура модели и т.д. Каждый блок, который может задержать движение транзакта – возможное место возникновения очереди. Основные дисциплины управления очередями (в широком смысле) – FIFO, LIFO, приоритетная, внутри одного приоритетного класса используется дисциплина FIFO. Основные характеристики очереди – средняя длина и среднее время пребывания транзакта в очереди.

Важнейшими видами очередей являются *очереди ограниченной длины* и *очереди ограниченной длины с ограниченным временем пребывания в модели*.

Функционирование очереди ограниченной длины иллюстрирует Рисунок 22. Для такой очереди существует ограничение на количество транзактов, которые могут находиться в очереди.

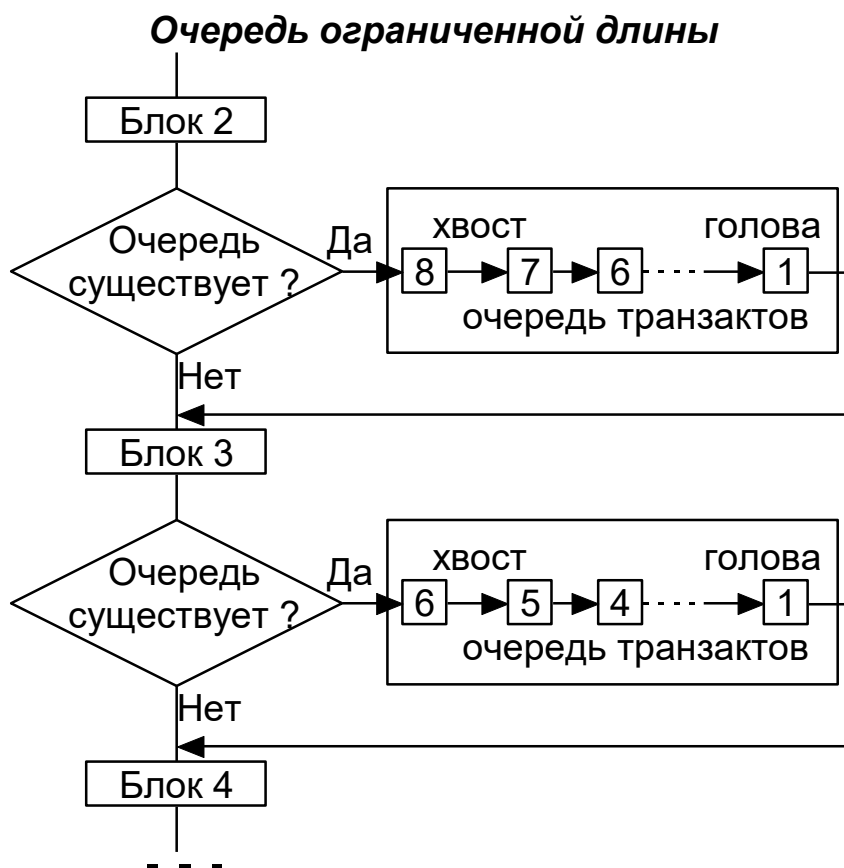


Рисунок 22 – Структура очереди ограниченной длины

Функционирование очереди ограниченной длины с ограниченным временем пребывания в модели иллюстрирует Рисунок 23. Условие ограничения времени пребывания проверяется для всех элементов простой очереди, и один или несколько транзактов должны покинуть ограниченную очередь. Направление дальнейшего движения транзактов зависит от дополнительных условий и структуры модели. Транзакт может быть удален из модели, либо вновь направлен в очередь с некоторой временной задержкой.

Таким образом, модели ограниченных очередей имеют собственное интеррогативное управление.

Очередь ограниченной длины с ограниченным временем пребывания

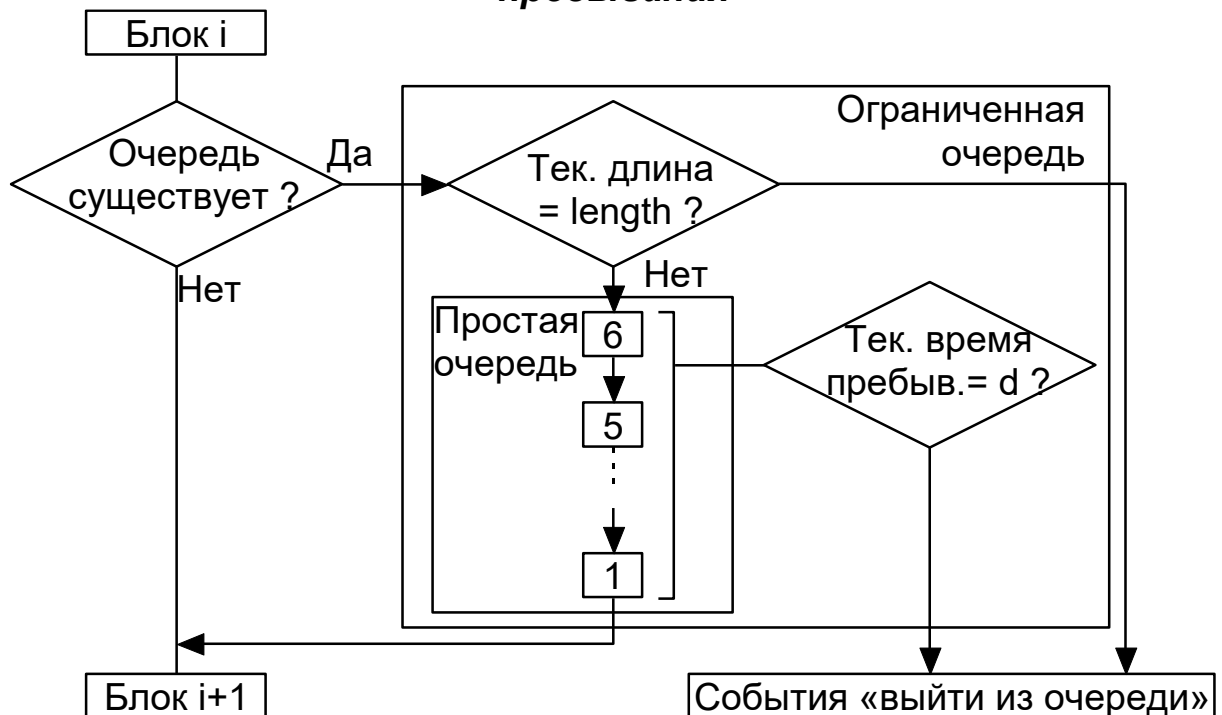


Рисунок 23 – Структура очереди ограниченной длины с ограниченным временем пребывания в модели

В системе моделирования модели очередей инкапсулированы в мониторе моделирования, поэтому работа с моделью очереди выполняется автоматически, и пользователь такой системы может не знать о деталях реализации моделей очередей.

Очередь является сложной стохастической структурой, которая может появляться и исчезать в модели исследуемой системы.