Лекция № 7

Описание процессов с наличием ресурсов

Ремарка -напоминание о ресурсах и конфликтах

Процессы Zi в системе Q развиваются параллельно. Это значит, что они изменяют значения параметров системы в течение одного и того же интервала времени. Достаточно типичны ситуации, когда по логике функционирования системы накладываются ограничения на изменение некоторых параметров несколькими процессами одновременно в течение заданного либо обусловленного интервала времени. Это возможно лишь для пересекающихся объектов. Ограничения развития процесса, накладываемые другими процессами, назовем конфликтными ситуациями, или конфликтами.

Общую область параметров для пересекающихся процессов Ok и Om назовем ресурсом. Таким образом, ресурс $R_{k,m}$ определяется, как

$$\mathbf{R_{k,m} = O_k \cap O_m}$$

Конфликт возможен только при наличии ресурса. Таким образом, необходимо добиться согласования процессов в этой области. В основе способов разрешения конфликтов лежит утверждение об обязательном разделении доступа процессов к ресурсу во времени. Рассмотрим следующие способы разрешения конфликтных ситуаций.

А. Явное разделение по времени (синхронизация). При этом способе разрешения конфликта раз/несение во времени производится явным указанием интервалов времени, определенных для каждого процесса. На рисунке 4.9 показан пример выделения таких интервалов для случая конфликта трех процессов.

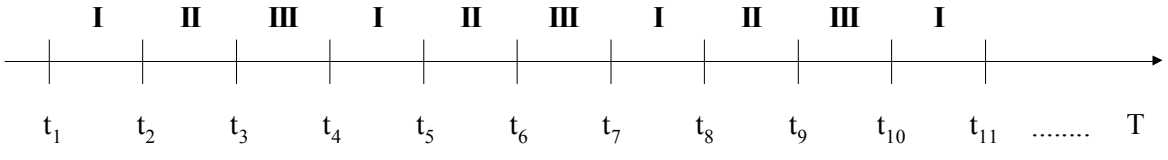
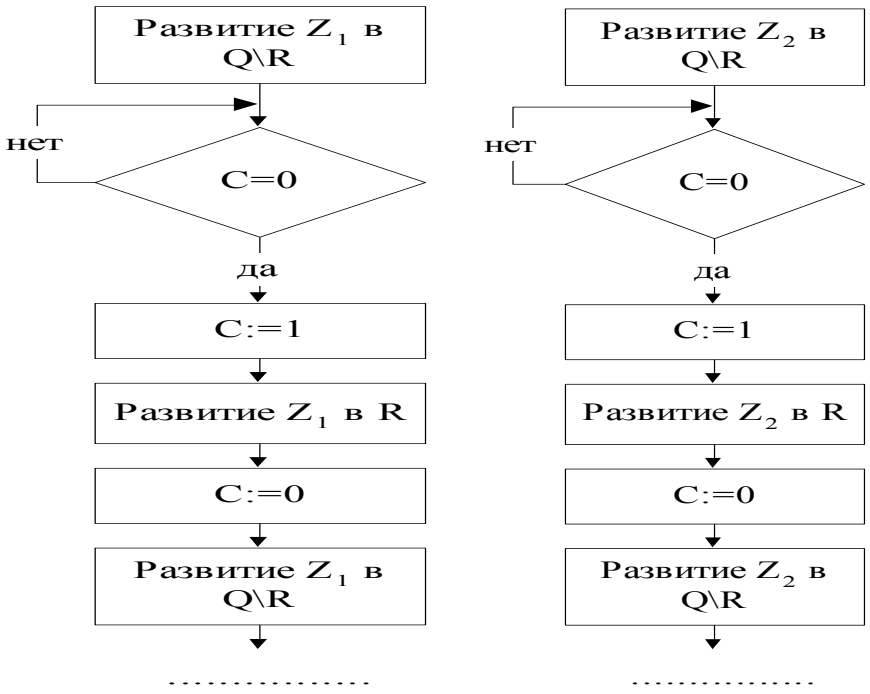


Рисунок 4.9.

Разделение по времени

Б. Логический способ. В частности – использование семафоров. Если условие захвата ресурса не ограничивает время использования этого ресурса захватившим его процессом, то в этом случае удобно использовать семафоры. Семафор относится к логическим способам разрешения конфликтов. Семафор есть простая логическая переменная, однозначно соответствующая ресурсу. Значение семафора ‘0’ означает, что ресурс может быть захвачен процессом, значение семафора ‘1’ блокирует захват ресурса. На рисунке 4.10 показан пример использования семафора C при захвате ресурса R двумя процессами Z1 и Z2. Следует заметить, что в качества семафора может выступать любая логическая функция.

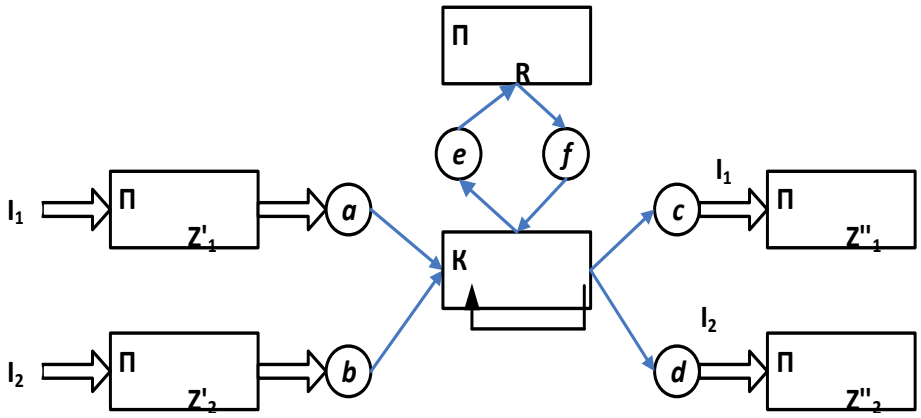
Рисунок 4.10. Использование семафоров



Алгоритмический способ

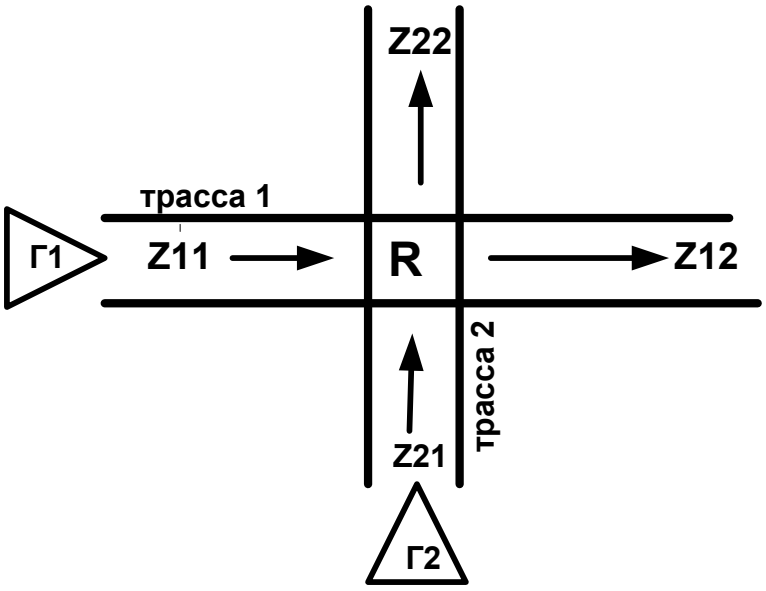
Универсальный способ разрешения конфликтов на ресурсах состоит в использовании блоков – агрегатов и блоков - контроллеров. Блоки – агрегаты используются при организации управления через параметры, а блоки – контроллеры используются при организации управления через инициаторы.

Контроллер. Рассмотрим вновь блок типа агрегат. Как было показано выше, он не имеет возможности взаимодействовать с внешними инициаторами. С тем, чтобы снять это ограничение, введем над инициаторами операции *пассивизации и активизации*. Операция пассивизации переводит инициатор в класс обычных параметров. Операция активизации, наоборот, обычный параметр переводит в класс инициаторов. Если агрегат содержит операторы, выполняющие указанные операции, то такой агрегат назовем контроллером, или К-блоком. Контроллер, таким образом, представляет собой агрегат, выполняющий операции над внешними инициаторами в соответствии с собственным алгоритмом функционирования. Операции над инициаторами суть операции над процессами. Таким образом, контроллер исполняет роль управляющего звена в некоторой блочной схеме.



Примеры процессов с наличием ресурсов

Рассмотрим перекресток



R есть ресурс, т.к. Z1∩Z2 = R (R является общим параметром обоих процессов)

Г1 и Г2 генераторы ; генерируют инициаторы в свои процессоры;

трасса 1 и трасса 2 , которые реализуют процессы Z1 и Z2 соответственно

А. Реализация семафором Этот вариант захвата ресурса годится для случая слабой загрузки трасс, когда на трассах нет очередей. Поэтому мы можем использовать принцип «кто первый захватил»

блок процессор ТРАССА 1;
описания;
r – **скаляр;** --отражает состояние ресурса R
ТПЕР – **скаляр**; -- время переезда через перекресток
ТКR – **скаляр**; --момент времени покидания перекрестка
…….

все описания

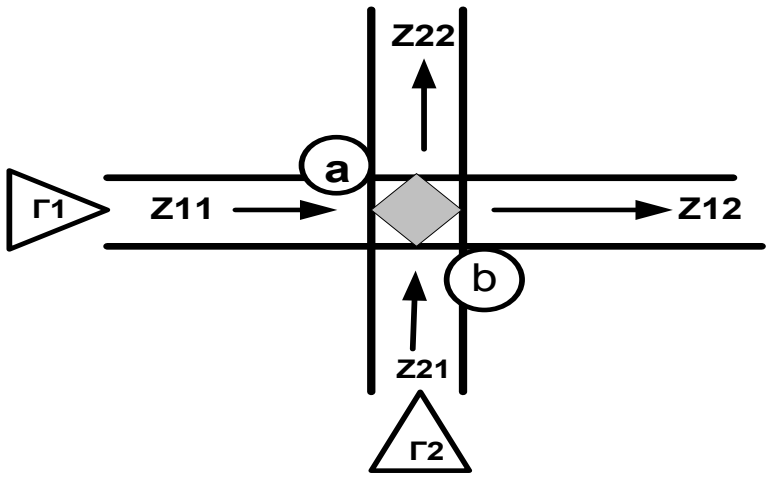
алгоритм

ВХОД : **ждать** r =0;
r := 1;
TKR := ВРЕМЯ+ТПЕР;
ждать ВРЕМЯ = TKR;
r := 0;
направить инициатор на ,,,,,(продолжение процесса Z1)
все алгоритм;
все блок;

блок процессор ТРАССА 2;
описания;
r **– скаляр; --**отражает состояние ресурса R
ТПЕР – **скаляр**; -- время переезда через перекресток
TKR – **скаляр**; --момент времени покидания перекрестка
…….
все описания
алгоритм
ВХОД : **ждать** r =0;
r := 1;
TKR := ВРЕМЯ+ТПЕР;
ждать ВРЕМЯ = TKR;
r := 0;
направить инициатор на ,,,,,(продолжение процесса Z2)
все алгоритм;
все блок;

Б. В случае интенсивной загрузки трасс и появления очередей предыдущий вариант не пройдет, он приведет к авариям. Поэтому рассмотрим вариант алгоритмического управления доступа к ресурсу.

Реализация агрегатом (с управлением через параметры)



Управление осуществляется введением параметров a и b, которые отражают **состояние ресурса – перекресток.**

блок процессор ТРАССА 1;
описания;
a**– скаляр; --** внешний параметр**,** отражает состояние ресурса R

ТПЕР – **скаляр**; -- время переезда через перекресток
TKR – **скаляр**; --момент времени покидания перекрестка
…….
все описания
алгоритм
ВХОД : **ждать** a = «зеленый»;
        TKR := ВРЕМЯ+ТПЕР;
        ждать ВРЕМЯ = TKR;
        направить инициатор на ,,,,,(продолжение процесса Z1)
все алгоритм;
все блок;

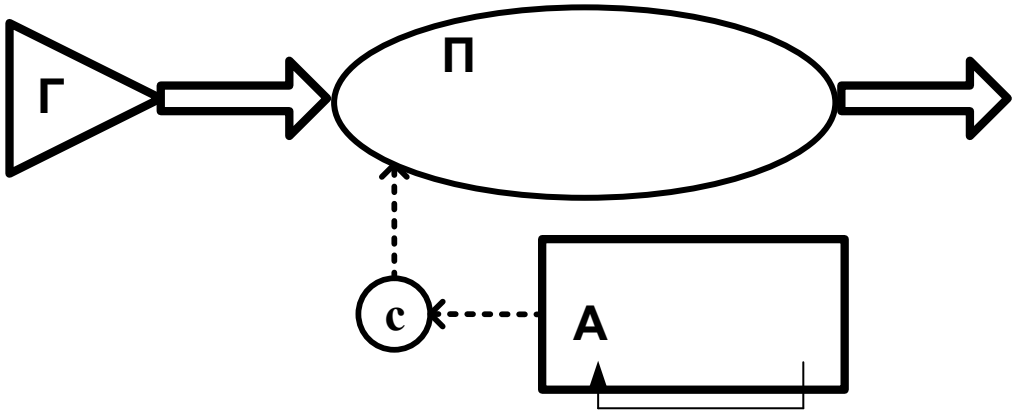
блок процессор ТРАССА 2;
описания;
b **– скаляр; --** внешний параметр, отражает состояние ресурса R
ТПЕР – **скаляр**; -- время переезда через перекресток
TKR – **скаляр**; --момент времени покидания перекрестка
…….
все описания
алгоритм
ВХОД : **ждать** b = «зеленый»;
        TKR := ВРЕМЯ+ТПЕР;
        ждать ВРЕМЯ = TKR;
        направить инициатор на ,,,,,(продолжение процесса Z2)
все алгоритм;
все блок;

блок контроллер СЕМАФОР;
описания
a,b – **скаляры**:
…….
все описания
алгоритм
НАЧ : a := «зеленый»;
        b := «красный»;
        ТПЕР := ВРЕМЯ+Т1;
        ждать ВРЕМЯ = ТПЕР;
        a := «красный»;
        b := «зеленый»;
        ТПЕР := ВРЕМЯ+Т2;
        ждать ВРЕМЯ = ТПЕР;
        направить инициатор на НАЧ;
все алгоритм;
все блок;

Процессы с прерыванием

Речь идет о процессе, реализация которого может прерываться периодически по тем или иным причинам. В этом случае необходимо описать процесс, описывающий прерывания основного процесса. Как правило, такой процесс представляется *агрегатом*.

Рассмотрим процесс передачи пакетов. Добавим следующую особенность его работы: канал периодически отключается на тестирование с параметрами ПТЕСТ – период тестирования, ТТ-время тестирования. На время тестирования пакеты скапливаются в канале, а потом передаются по очереди. В этом случае в работу процессора вводим параметр **с**, определяющий включение процесса тестирования.



блок процессор ИМЯ;
описания;
ТОПК – **скаляр**; --момент времени окончания передачи пакета в канале
КАН - **скаляр**; --признак состояния канала
…….
все описания;
алгоритм;
НАЧ: **ждать** (КАН=0) * (с = 0); -- * - символ операции конъюнкции
КАН := 1;
ТОПК:= ВРЕМЯ + RAND(10,50);
ждать ВРЕМЯ = ТОПК;
КАН := 0;
направить инициатор на ….. –во внешний блок
все алгоритм;
все блок;

блок агрегат ТЕСТ; **описания**
ПТЕСТ , ТТ , ТОТ, с – **скаляры**:
…….
все описания
алгоритм
НАЧ : с := 0;
ТОТ := ВРЕМЯ+ ПТЕСТ;
ждать ВРЕМЯ = ТОТ;
с := 1;
ТОТ := ВРЕМЯ+ ТТ;
ждать ВРЕМЯ = ТОТ;
направить инициатор на НАЧ;
все алгоритм;
все блок;

Схемы описаний функционирования систем

Агрегативная схема. Агрегативная схема описания функционирования системы предполагает использование только А-блоков . Как показано выше, в такой схеме взаимодействие может осуществляться лишь через параметры. В результате формируется информационная сеть агрегатов: агрегаты - вершины сети, дуги - информационные связи. Для агрегативных схем показано, что каждый А-блок в такой модели может представляться некоторым конечным автоматом. Если функционирование системы может быть описано совокупностью конечных автоматов, взаимодействующих между собой через множество входных и выходных параметров, то применение агрегативных схем представляется наиболее рациональным.

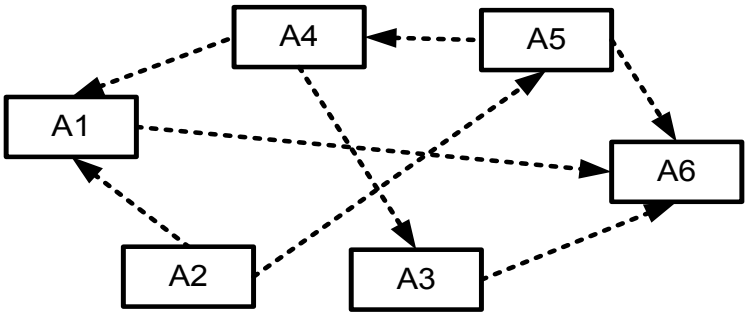


Рисунок Пример агрегативной схемы

На рисунке показаны 6 агрегатов, выполняющих функции сбора и обработки информации. Например, это могут быть какие-либо агентные структуры. Пунктирными линиями показано взаимодействие через параметры, относящиеся к тому или иному агрегату. Из схемы видно, что агрегат А2 сообщает информацию, агрегаты А1, А4, А3, А5 преобразуют полученную информацию, агрегат А6 собирает всю полученную информацию.

Процессная схема. Если в основе описания функционирования системы лежит использование П-блоков, то такое описание назовем процессным. Взаимодействие в процессных схемах осуществляется как через параметры, так и через локальные среды процессов. Процессный подход наиболее эффективен, когда имеем дело с множеством явно выраженных локальных процессов, например, при описании информационных, биологических, экономических, социальных и т.п. систем. Процессный подход реализован в языках Simula, GPSS и др. Пример процессной схемы приведен на рисунке 15.

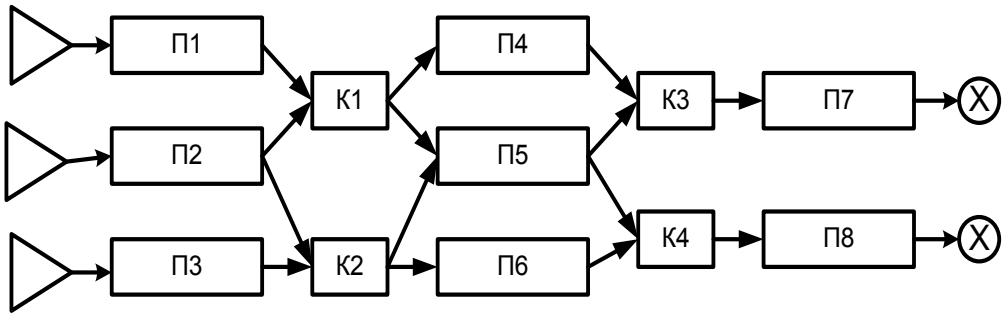


Рисунок. Пример процессной схемы

На рисунке показаны процессоры и контроллеры, обеспечивающие управление потоками инициаторов и разрешение конфликтных ситуаций. Сплошными линиями показаны пути движения инициаторов. Как видно из схемы, инициаторы, порожденные генераторами, проходят все треки, заданные процессорами и уничтожаются к конце пути. В процессных схемах реализация всех процессов осуществляется только инициаторами, возникшими из генераторов. Контроллеры лишь меняют направление и условия продвижения инициаторов.

Потоковая схема. Потоковая схема возникает из агрегативной, когда среди связей агрегатов появляются связи с управляющей информацией. Когда эта информация поступает в агрегат, она производит инициацию процессов внутри агрегата. Если на графе информационных связей агрегативной схемы выделить лишь связи с управляющей информацией, то получим потоковую сеть агрегатов. Управляющая информация в этом случае может рассматриваться, как *инициаторы сетевого уровня*.

Таким образом, в потоковых схемах существуют два вида инициаторов: сетевые инициаторы, инициирующие запуск агрегата, и блочные инициаторы, реализующие процесс функционирования блока. Пример потоковой схемы приведен на рисунке 16.

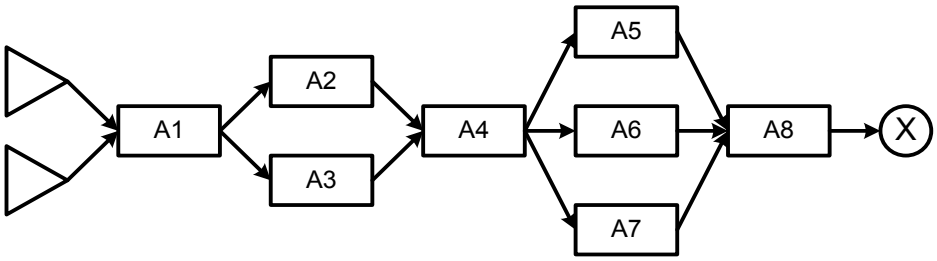


Рисунок Пример потоковой схемы

На рисунке стрелками показаны пути движения сетевых инициаторов. Блочные (внутренние инициаторы) находятся внутри агрегатов. Например, сетевой инициатор, поступающий на агрегат А4 от агрегата А2, вызывает выполнение некоторого процесса в А4. Сам инициатор после этого может быть уничтожен или перемещен блоком А4 дальше по сети.

К потоковым схемам относятся сети Наура, Е-сети, сети Петри. Наибольшее распространение среди потоковых схем получили сети массового обслуживания (СеМО). Эти сети возникают из процессных схем, когда процессоры очень просты, либо отсутствуют вообще. Тогда основным элементом сети становятся контроллеры. Объединение контроллера с простым процессором или агрегатом формируют объект, называемый *системой массового обслуживания* (СМО). Таким образом, возникает сеть из СМО. Согласно определению потоковой схемы, по этой сети перемещаются инициаторы сетевого уровня, которые в теории СеМО называются сообщениями или требованиями.