

**ФОРМАЛЬНОЕ ОПИСАНИЕ АЛГОРИТМА  
УПРАВЛЕНИЯ ВЗАИМОДЕЙСТВУЮЩИМИ  
ПАРАЛЛЕЛЬНЫМИ ПРОЦЕССАМИ В ЗАДАЧЕ  
ПРОИЗВОДИТЕЛИ–ПОТРЕБИТЕЛИ С ИСПОЛЬЗОВАНИЕМ  
СОГЛАСУЮЩЕГО КОЛЬЦЕВОГО БУФЕРА**

В статье рассматриваются вопросы формального описания алгоритма управления синхронизацией параллельных асинхронных процессов при обмене сообщениями в задаче *производители–потребители* с использованием согласующего кольцевого буфера. Рассматривается простой базовый вариант задачи, когда имеет место один производитель и один потребитель. Формальное описание алгоритма управления процессами базируется на использовании логики недетерминированных автоматов.

**Введение**

Задача *производители–потребители* является одной из классических задач синхронизации процессов. Процессы взаимодействуют через некоторую обобщенную область памяти – согласующий буфер сообщений. В данной работе, в отличие от [1], будем использовать согласующий кольцевой буфер сообщений, в который информация записывается и считывается по очереди.

Производитель передает сообщение потребителю, помещая его в конец очереди. Потребитель получает сообщение, извлекая его из начала очереди. Чтобы сообщение нельзя было извлечь из пустой очереди или поместить в заполненный буфер, нужна синхронизация.

Для формального описания алгоритма управления взаимодействующими процессами в рассматриваемой задаче используется язык логики недетерминированных автоматов (НДА), с помощью которого описываются все реализуемые в алгоритме частные события [1].

В работе рассматривается алгоритм функционирования кольцевого согласующего буфера при обмене сообщениями между производителем и потребителем, позволивший определить условия, для которых буфер будет полон, пуст или займет промежуточное состояние.

На основании этих условий, общих требований к алгоритму управления межпроцессного взаимодействия и методики формализации функций взаимного исключения критических участков [1] строится граф НДА, представляющий этот алгоритм, с описанием всех основных событий, реализуемых в алгоритме.

**1. Функционирование кольцевого согласующего буфера при обмене сообщениями между производителем и потребителем**

Производитель помещает сообщения в разделяемый кольцевой буфер [2], потребитель извлекает их оттуда. Буфер содержит очередь уже помещенных, но еще не извлеченных сообщений. Эта очередь может быть представлена связным списком или массивом.

Для представления очереди сообщений используются две целочисленные переменные, которые указывают соответственно на первую заполненную

и первую пустую ячейку буфера. Эти переменные организуются с помощью счетчиков буфера записи СчБз и счетчика буфера чтения СчБчт. Показания счетчика СчБз будут определять номер первой незаполненной (пустой) ячейки, а счетчика СчБчт – номер первой заполненной ячейки. Для этой цели при инициализации системы управления в указанные счетчики записываются единицы, т.е. при  $t = 0$  СчБз = 1; СчБчт = 1.

Если принять количество  $n$  ячеек в кольцевом буфере, например, равным 8, то счетчики буфера будут работать по модулю 8. Тогда при переполнении счетчиков может быть образован сигнал переполнения, который может быть использован для определения состояний буфера, в которых буфер будет полон или пуст.

Для нашего варианта кольцевого буфера событие «буфер полон» ( $S_{BP} = 1$ ) будет иметь место при равенстве значений счетчиков буфера (СчБз = СчБчт > 1) и переносе из счетчика буфера записи (СчБз = 0).

Событие «буфер пуст» ( $S_{BO} = 1$ ) будет иметь место также при равенстве значений счетчиков буфера (СчБз = СчБчт) и если не было сигнала переноса из СчБчт или при условии, когда в процессе работы счетчики буфера приняли исходное состояние (СчБз = СчБчт = 1) и был перенос из СчБчт.

Рассмотрим некоторые варианты записи и чтения в кольцевой буфер.

Представим буфер массивом, в котором имеется  $n = 8$  ячеек (рис. 1), и в каждую ячейку записывается одно сообщение.

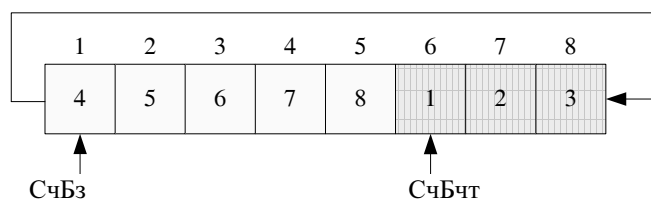


Рис. 1 Произвольное положение массива в буфере

Будем считать, что номер ячейки буфера совпадает с номером сообщения. Тогда верхний ряд цифр (рис. 1) указывает на исходное состояние буфера перед началом работы системы, когда все ячейки пусты.

Кольцевой буфер при записи информации работает следующим образом. Такт записи в одну ячейку буфера состоит из двух частей: в первой части осуществляется запись в ячейку, номер которой указывает счетчик СчБз, во второй части производится кольцевой сдвиг содержимого буфера влево. На рисунке 1 показаны результаты записи трех сообщений, когда положение счетчика СчБз стало равным 4, тогда как в исходном состоянии СчБз было равно единице. Таким образом, ячейка буфера под номером 4 будет первой пустой ячейкой после сообщений в конце очереди.

Пусть после трех тактов записи выполняется чтение из буфера до полной его очистки. Тогда состояние счетчиков буфера записи и чтения будет одинаковыми СчБз = СчБчт = 4. Это свидетельствует о том, что буфер пуст, т.к. не было сигнала переполнения из СчБчт.

Допустим, что после этого было выполнено 5 тактов записи в буфер. Тогда состояние буфера будет иметь вид (рис. 2), а состояние счетчика записи после пяти тактов определится сложением по модулю 8, т.е. СчБз =  $4 \oplus 5 = 1$ , а состояние счетчика чтения не изменится – СчБчт = 4.

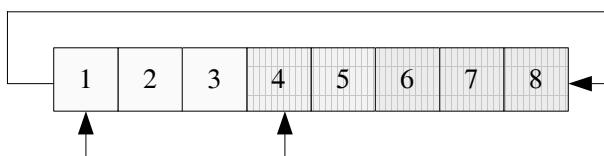


Рис. 2 Состояние буфера

Пусть в следующем такте выполняется запись в буфер трех порций информации, тогда состояние буфера примет вид (рис. 3).

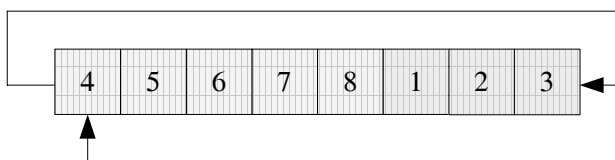


Рис. 3 Состояние буфера

Для этого состояния буфер будет полон, т.к. состояния счетчиков после операции записи имеют вид  $СчБз = 1 + 3 = 4$ ,  $СчБчт = 4$ , и был перенос из счетчика записи (в предыдущем такте записи).

В следующем такте возможна лишь операция чтения, т.к. буфер полон. Пусть считывается пять порций информации. Тогда состояние буфера будет иметь вид (рис. 4).

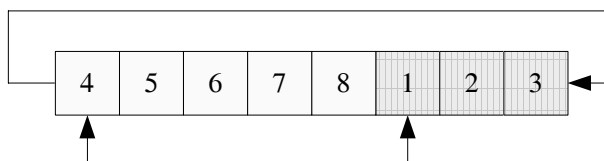


Рис. 4 Состояние буфера

Показания счетчиков буфера после операции чтения имеют следующие значения:  $СчБз = 4$ ;  $СчБчт = 4 \oplus 5 = 1$ .

## 2. Представление алгоритма управления взаимодействующими параллельными процессами в задаче *производитель–потребитель* в виде графа НДА и системы канонических уравнений

Для обеспечения бесконфликтного взаимодействия процессов в задаче *производитель–потребитель* с использованием согласующего буфера сообщений к алгоритму управления процессами предъявляются следующие требования:

а) выполнение условий взаимного исключения по отношению к критическому ресурсу – буферу сообщений;

б) учет состояния буфера сообщений, характеризующего возможность и невозможность посылки (принятия) очередного сообщения, а именно процесс-производитель при попытке поместить очередное сообщение в полностью заполненный буфер должен быть полностью заблокирован; попытка процесса-потребителя чтения из пустого буфера также должна быть заблокирована.

Как было рассмотрено в предыдущем разделе, для определения состояния согласующего кольцевого буфера сообщений используются два основ-

ных счетчика буфера: СчБз и СчБчт, в которые при инициализации ( $t = 0$ ) записываются единицы. По показаниям этих счетчиков определяются события:  $S_{БП} = 1$  – буфер полон и  $S_{БО} = 1$  – буфер пуст.

Введение двух событий  $S_{БП}$  и  $S_{БО}$  позволяет охарактеризовать промежуточное состояние буфера сообщений, когда он может быть и не полон, и не пуст ( $S_{БП} = S_{БО} = 1$ ). Это может иметь место, когда при очередной записи буфер был заполнен не полностью или когда после очередного чтения буфер не пуст. Для этого варианта использования буфера сообщений используются два дополнительных счетчика: Сч\*з и Сч\*чт, в которые перед операциями с буфером сообщений заносится число порций информации, записываемой или считываемой из буфера за один раз. По показаниям этих счетчиков определяются события  $S_{кз}$  и  $S_{кч}$ , свидетельствующие о том, что должны быть прекращены операции записи или чтения соответственно.

Учитывая рассмотренные выше требования к алгоритму управления межпроцессного взаимодействия для параллельных процессов через согласующий кольцевой буфер, а также методику формализации функций взаимного исключения критических областей (участков) в [1], граф НДА, представляющий рассматриваемый алгоритм управления, можно представить в следующем виде (рис. 5).

Граф НДА (рис. 5) совместно с системой канонических уравнений, описывающих все реализуемые в алгоритме события, дает полное представление о структуре алгоритма управления взаимодействующими процессами в задаче производитель–потребитель, когда имеется один производитель и один потребитель.

Опишем назначение событий, представленных на графе (рис. 5):

$S_3^1$  и  $S_3^2$  – события, свидетельствующие о поступлении заявок на обслуживание производителя (сервера) и потребителя (клиента) на запись и чтение из согласующего буфера соответственно.

$S_1$  и  $S_2$  – события, определяющие прием заявок на обслуживание производителя и потребителя на запись и чтение из согласующего буфера соответственно.

$S_k^1$  и  $S_k^2$  – события, определяющие входы производителя и потребителя в свои критические участки, эти события при их истинности свидетельствуют также о том, что соответствующие процессы (производитель и потребитель) находятся внутри своих критических участков (областей).

$S_{pz}^1$  и  $S_{pч}^2$  – события, определяющие непересекающиеся операции записи и чтения из согласующего буфера соответственно.

$S_n^1$  и  $S_n^2$  – события, свидетельствующие об окончании одноразовых операций записи и чтения из согласующего буфера соответственно, эти события при их истинности обеспечивают также выход взаимодействующих процессов из соответствующих критических участков.

$S_ч^1$  и  $S_з^2$  – события, символизирующие ожидание взаимодействующими процессами окончания процедуры записи и чтения из согласующего буфера, чтобы продолжить свою работу, прерванную обращением к согласующему буферу соответственно.

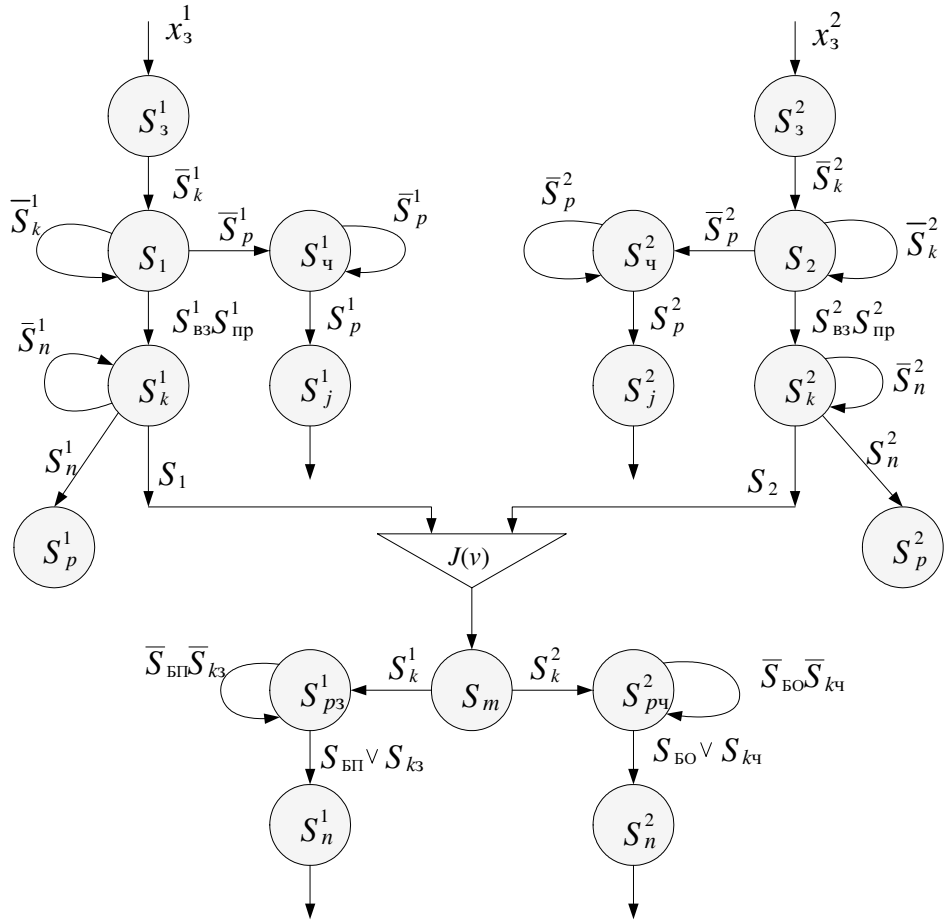


Рис. 5 Граф НДА алгоритма управления межпроцессного взаимодействия в задаче *производитель–потребитель* с согласующим кольцевым буфером сообщений (для двух процессов: клиент–сервер)

$S_j^1$  и  $S_j^2$  – события, обеспечивающие начало выполнения своих программ производителем и потребителем после выхода из критических участков соответственно.

$S_m$  – событие, определяющее обращение к согласующему буферу и обеспечивающее начало процедуры реализации обращения к согласующему буферу при записи и чтении.

$S_{БП}$  и  $S_{БО}$  – события, свидетельствующие о том, что согласующий буфер полон и пуст соответственно.

$S_{кз}$  и  $S_{кч}$  – события, свидетельствующие о том, что должна быть прекращена запись и чтение из согласующего буфера соответственно. В этом случае буфер может находиться в промежуточном состоянии – не пуст и не полон.

$S_p^1$  и  $S_p^2$  – события, обеспечивающие окончание ожидания процессами выхода из своих критических участков соответственно.

$S_{пр}^1$  и  $S_{пр}^2$  – события, определяющие приоритет записи и чтения из согласующего буфера соответственно.

$S_{вз}^1$  и  $S_{вз}^2$  – события, обеспечивающие взаимоисключение критических участков (интервалов) производителя и потребителя соответственно.

$J(v)$  – дизъюнктивный соединитель.

Основными событиями, определяющими бесконфликтные взаимодействия в любых задачах, когда имеет место использование разделяемых ресурсов (данных), в том числе и в задаче *производители–потребители*, являются события, определяющие прием (восприятие) заявок на обслуживание (в нашем случае это события  $S_1$  и  $S_2$ ), и события, определяющие входы процессов в свои критические интервалы (в нашем случае это события  $S_k^1$  и  $S_k^2$ ).

Отмеченные события формализуют исходя из основных трех требований к критическим интервалам, обеспечивающих отсутствие конфликтных ситуаций при взаимодействии параллельных процессов во время организации доступа к разделяемым данным (ресурсу). Эти требования были определены Деккером и Дейкстра [3, 4] и формулируются таким образом:

1. В любой момент времени только один процесс может находиться внутри критического интервала.

2. Ни один процесс не может оставаться внутри критического интервала бесконечно долго.

3. Ни один процесс не должен бесконечно долго ждать входа в критический интервал.

Анализируя указанные требования к критическим интервалам, можно сформировать условия их выполнения, используя логику НДА.

Первое требование должно обеспечиваться таким описанием условий зарождения события  $S_k^i$ , определяющих вход в свой критический интервал  $i$ -го процесса, которые учитывают наличие воспринятой заявки на обслуживание, взаимоисключение (несовместимость) критических интервалов и приоритетность процессов.

Второе требование должно обеспечиваться таким описанием условий сохранения событий  $S_k^i$ , для которого событие  $S_k^i$  будет существовать только до окончания одноразовой процедуры обращения к разделяемым данным.

Третье требование должно обеспечиваться таким описанием события  $S_i$ , определяющего прием заявки процесса на обслуживание, когда заявка воспринимается и сохраняется воспринятой лишь в том случае, когда данный процесс не находится в своем критическом интервале.

Учитывая сформулированные условия, обеспечивающие выполнение требований к критическим интервалам, события  $S_k^i$  и  $S_i$  можно описать в следующем виде:

$$S_k^i(t+1) = S_i S_{вз}^i S_{пр}^i \vee S_k^i \bar{S}_n^i; \quad (1)$$

$$S_i(t+1) = (S_{i,3} \vee S_i) \bar{S}_k^i. \quad (2)$$

В уравнении (1)  $S_{вз}^i$  – комбинационное событие, обеспечивающее взаимоисключение критических интервалов на основе несовместимости событий  $S_k^i$  с другими событиями из их общего числа, равного  $n$ ,

$$S_{вз}^i = \bigwedge_{(\forall \alpha)[\alpha \neq i]} \bar{S}_k^\alpha, \quad i = \overline{1, n}.$$

Для нашего примера, когда  $n = 2$ , имеем:

$$S_{вз}^1 = \bar{S}_k^2, \quad S_{вз}^2 = \bar{S}_k^1. \quad (3)$$

$S_{пр}^i$  – комбинационное событие, обеспечивающее заданное приоритетное обслуживание  $i$ -го процесса. Например, для дисциплины обслуживания с относительным приоритетом  $S_{пр}^i$  может быть представлено следующим выражением:

$$S_{пр}^i = S_3^i \bigwedge_{(\forall \alpha)[\alpha < i]} \bar{S}_3^\alpha, \quad i = \overline{1, n}.$$

В нашем конкретном случае для  $n = 2$  имеем два события:

$S_{пр}^1$  – событие, определяющее приоритет записи в согласующий буфер;

$S_{пр}^2$  – событие, определяющее приоритет чтения из согласующего буфера.

Описания событий  $S_{пр}^1$  и  $S_{пр}^2$  могут быть составлены исходя из анализа состояния согласующего буфера и наличия воспринятых заявок на обслуживание (запись и чтение). Так как состояния буфера характеризуются двумя событиями  $S_{БП}$  и  $S_{БО}$ , то возможны следующие четыре комбинации значений этих событий:

1. Для  $S_{БП} = S_{БО} = 0$  буфер не пуст и не полон – возможна или запись, или чтение из буфера: при  $S_1 = 1$  будет запись, а при  $S_1 = 0$  и  $S_2 = 1$  будет чтение.

2. Для  $S_{БП} = 1$  и  $S_{БО} = 0$  при  $S_1 = S_2 = 1$  возможно только чтение.

3. Для  $S_{БП} = 0$  и  $S_{БО} = 1$  при  $S_1 = S_2 = 1$  возможна только запись.

4. Для  $S_{БП} = S_{БО} = 1$  имеем невозможную комбинацию, когда для любых значений событий  $S_1$  и  $S_2$  значение операции в буфере будет неопределенным (т.е. булева функция  $S_{пр}^2$  будет не полностью определенной).

При рассмотренных возможных комбинациях событий, характеризующих состояние буфера и наличие воспринятых заявок на запись и чтение из буфера, события  $S_{пр}^1$  и  $S_{пр}^2$  будут иметь вид

$$S_{пр}^1 = \bar{S}_{БП}, \quad S_{пр}^2 = \bar{S}_{БО} \bar{S}_1 \vee S_{БП}. \quad (4)$$

Учитывая полученные соотношения (3) и (4), уравнения (1) и (2) запишем в следующем виде:

$$\begin{aligned} S_k^1(t+1) &= S_1 \bar{S}_k^2 \bar{S}_{БП} \vee S_k^1 \bar{S}_n^1, \\ S_k^2(t+1) &= S_2 \bar{S}_k^1 (S_{БП} \vee \bar{S}_{БО} \bar{S}_1) \vee S_k^2 \bar{S}_n^2, \end{aligned} \quad (5)$$

$$\begin{aligned} S_1(t+1) &= (S_{1,3} \vee S_1) \bar{S}_k^1, \\ S_2(t+1) &= (S_{2,3} \vee S_2) \bar{S}_k^2. \end{aligned} \quad (6)$$

Представление событий  $S_1$  и  $S_2$  в виде формул (6) свидетельствует о том, что заявки процессов на операции записи или чтения могут быть приняты только в том случае, если процессы не находятся в своих критических интервалах.

События  $S_{\text{ч}}^1$  и  $S_{\text{ч}}^2$ , символизирующие ожидание взаимодействующими процессами окончания процедуры записи и чтения из буфера, чтобы продолжить свою работу, прерванную обращением к согласующему буферу, формализуются так:

$$\begin{aligned} S_{\text{ч}}^1(t+1) &= (S_1 \vee S_{\text{ч}}^1) \bar{S}_p^1, \\ S_{\text{ч}}^2(t+1) &= (S_2 \vee S_{\text{ч}}^2) \bar{S}_p^2. \end{aligned} \quad (7)$$

Остальные события, реализуемые алгоритмом межпроцессного взаимодействия в задаче *производители–потребители*, представленные на графе НДА (рис. 5), формализуются таким образом:

$$\begin{aligned} S_m(t+1) &= S_k^1 S_1 \vee S_k^2 S_2, \\ S_{p3}^1(t+1) &= S_m S_k^1 \vee S_{p3}^1 (\bar{S}_{\text{БП}} \bar{S}_{k3}), \\ S_{pч}^2(t+1) &= S_m S_k^2 \vee S_{pч}^2 (\bar{S}_{\text{БО}} \bar{S}_{kч}), \\ S_n^1(t+1) &= S_{p3}^1 (S_{\text{БП}} \vee S_{k3}), \\ S_n^2(t+1) &= S_{pч}^2 (S_{\text{БО}} \vee S_{kч}), \\ S_p^1(t+1) &= S_k^1 S_n^1, \\ S_p^2(t+1) &= S_k^2 S_n^2, \\ S_j^1(t+1) &= S_{\text{ч}}^1 S_p^1, \\ S_j^2(t+1) &= S_{\text{ч}}^2 S_p^2. \end{aligned} \quad (8)$$

### Заключение

Полученная система канонических уравнений (5)–(8) отвечает требованиям описания алгоритмов управления параллельными взаимодействующими процессами в формальной стандартной форме, когда функции переходов представлены в явном виде, что позволяет в дальнейшем просто решать задачи структурной реализации алгоритмов управления. В частности в качестве примера такая структурная реализация представлена в [5], где рассматривается аппаратная реализация модуля синхронизации асинхронных процессов с использованием ПЛИС.

Рассмотренная в данной работе методика формализации алгоритма управления межпроцессным взаимодействием для простейшего варианта за-



дачи *производители–потребители* является основой для решения более сложных вариантов решения этой задачи, когда имеет место несколько производителей и несколько потребителей.

**Список литературы**

1. **Вашкевич, Н. П.** Недетерминированные автоматы в проектировании систем параллельной обработки: учебное пособие / Н. П. Вашкевич. – Пенза : Изд-во Пенз.гос.ун-та, 2004. – 280 с.
2. **Эндрюс, Г. Р.** Основы многопоточного параллельного и распределенного программирования : пер. с англ. / Г. Р. Эндрюс. – М. : Вильямс, 2003. – 512 с.
3. **Дейтел, Г.** Введение в операционные системы : в 2-х т. : пер. с англ. / Г. Дейтел. – М. : Мир, 1987. – 1 т. – 359 с.
4. **Кейлингерт, П.** Элементы операционных систем : пер. с англ. / П. Кейлингерт. – М. : Мир, 1985. – 295 с.
5. **Вашкевич, Н. П.** Аппаратная реализация функций синхронизации параллельных процессов при обращении к разделяемому ресурсу на основе ПЛИС / Н. П. Вашкевич, Р. А. Бикташев, Е. И. Гурин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2007. – № 2. – С. 3–12.