

УДК 62.507

## ПРИМЕНЕНИЕ МЕТОДОЛОГИИ РАСКРАШЕННЫХ СЕТЕЙ ПЕТРИ К РЕШЕНИЮ ЗАДАЧИ УСТРАНЕНИЯ ВЗАИМНОЙ БЛОКИРОВКИ ПАРАЛЛЕЛЬНЫХ ПРОЦЕССОВ

С.В. Яковлев, Д.В. Мочалов, М.В. Ильясова

*Предложены модели, реализованные с применением методологии раскрашенных сетей Петри, позволяющие решить задачу устранения взаимных блокировок параллельных процессов.*

*In this work the model is implemented using the methodology of colored Petri nets allow us to resolve the problem of removing deadlocks parallel processes.*

*Ключевые слова: параллельные процессы, блокировка, цветные сети Петри, CPN Tools.*

*Keywords: parallel processes, blocking, colored Petri nets, CPN Tools.*

Современные вычислительные системы предполагают наличие сложных алгоритмов функционирования с параллельно выполняющимися процессами. При этом в таких системах имеется некоторое множество параллельных процессов, которые являются взаимозависимыми ввиду использования ими одних и тех же ресурсов (аппаратные средства, память, программное обеспечение и т. д.). Наличие совместно используемых ресурсов требует организации взаимодействия параллельных процессов, т. к. независимое их выполнение может привести к ошибкам или к возникновению тупиковых ситуаций [1].

Мы предлагаем способы устранения взаимных блокировок параллельных процессов, вызываемых наличием разделяемых ресурсов с использованием методологии раскрашенных сетей Петри [2] и программной среды CPN Tools [3].

Реализуемые параллельные процессы состоят из последовательности  $K$ -операций, осуществляемых программными компонентами (ПК), каждый из которых соответствует переходу сети Петри  $t_1, \dots, t_k$ . Рассматриваемое множество асинхронных параллельных процессов задается совокупностью сетей Петри, использующих  $m$  ресурсов (процессорных модулей, ПМ) (рис. 1), и является интерпретацией классического примера разделения файла двумя процессами [4].

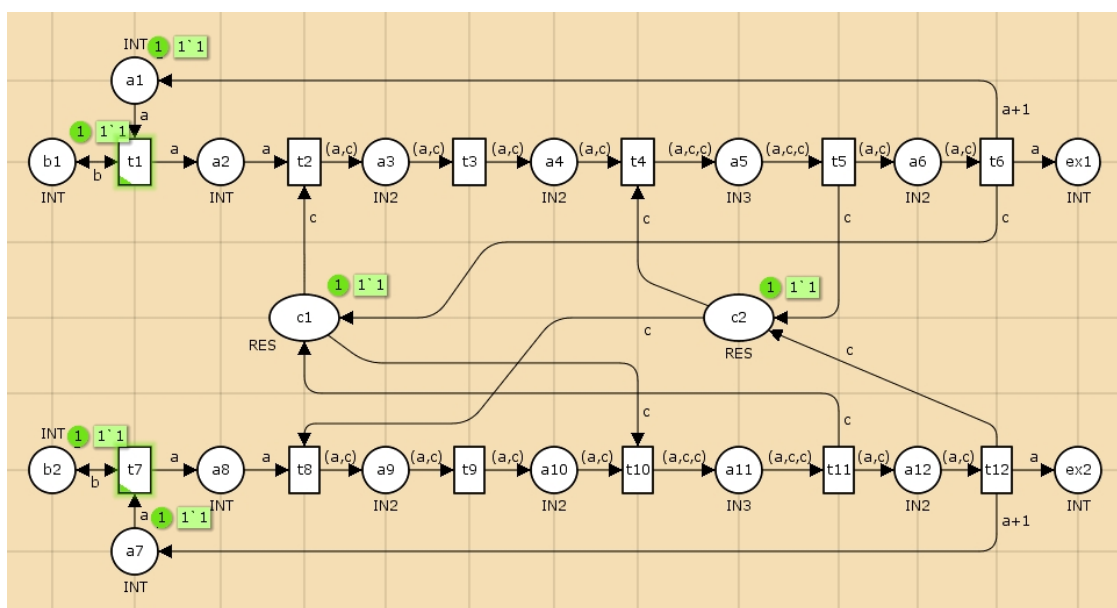


Рисунок 1 – Сеть Петри реализующая модель выполнения двух параллельных процессов

В состоянии, когда процесс владеет ресурсом, выполняется последовательность операций этого процесса и ресурс считается занятым. Поскольку один и тот же ПМ может потребоваться для выполнения нескольких процессов, возможны блокировки их выполнения, тупиковые состояния (рис. 2). Для анализа рассматриваемой модели будем использовать граф достижимых маркировок сети Петри представленный на рис. 3.

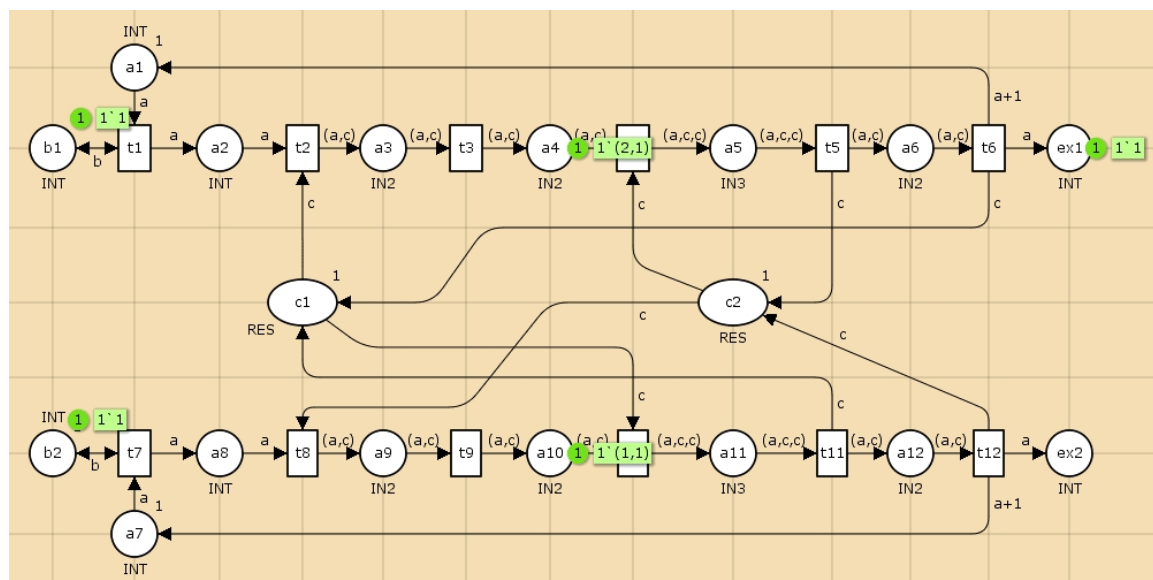


Рисунок 2 – Модель выполнения двух параллельных процессов в состоянии блокировки

Состояния сети Петри, из которых все пути графа достижимых маркировок ведут в тупиковое состояние  $S_t$  (в данном случае  $S_{33}$ ), считают предтупиковыми  $S_{пт}$  ( $S_{22}$ ,  $S_{23}$ ,  $S_{32}$ ). Состояния  $S_t$  и  $S_{пт}$  объединенные в одно множество состояний называют множеством запрещенных состояний.

Очевидно, что для ускорения реализации процессов необходимо устранять все конфликтные ситуации, т. е. максимально использовать имеющийся в системе параллелизм. Рассмотрим несколько способов устранения взаимных блокировок процессов и тупиковых состояний.

Первый подход, который состоит в использовании своевременной принудительной блокировки процессов [1]. При этом определяется минимальное число процессов, которое необходимо заблокировать и минимальное число состояний, в которых эта блокировка должна сохраняться.

Для устранения опасных состояний осуществляют преобразование графа, исключая из него все ребра, соединяющие опасные и запрещенные состояния, что приводит к формированию графа, содержащего только безопасные состояния.

Для графа достижимых маркировок рассматриваемого примера состояние  $S_{11}$  является корнем двух опасных отрезков:  $S_{21} - S_{31}$  и  $S_{12} - S_{13}$ . Это состояние переводится в состояние конфликта введением в сеть Петри (см. рис. 4) позиции  $S_b$ , являющейся входной для переходов  $t_2$  и  $t_8$ . Тогда после срабатывания перехода  $t_2$  ( $t_8$ ), что соответствует развитию процесса  $P_1$  ( $P_2$ ), удаляется метка из позиции  $S_b$  и процесс  $P_2$  ( $P_1$ ) блокируется. Как видно из графа на рис. 3, состояние  $S_{31}$  ( $S_{13}$ ) является последним состоянием выбранного опасного отрезка, поэтому при реализации перехода  $t_4$  ( $t_{10}$ ) блокировка может быть снята. Для этого позиция  $c_b$  должна быть выходной позицией переходов  $t_4$  и  $t_{10}$ .

Преобразованный граф достижимых маркировок представлен на рисунке 5, где в множествах  $Q$  всех состояний, кроме тех, которые соответствуют опасным отрезкам, содержится позиция  $c_b$ .

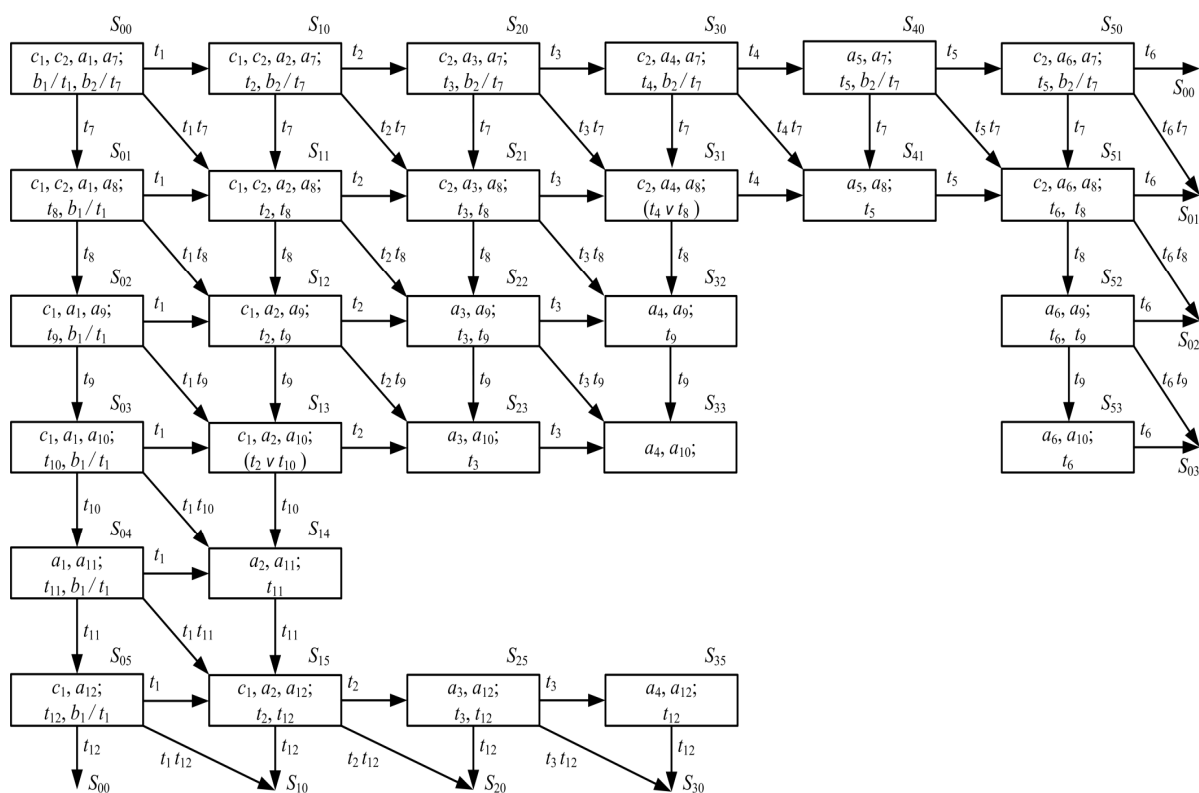


Рисунок 3 – Граф достижимых маркировок исходной сети Петри

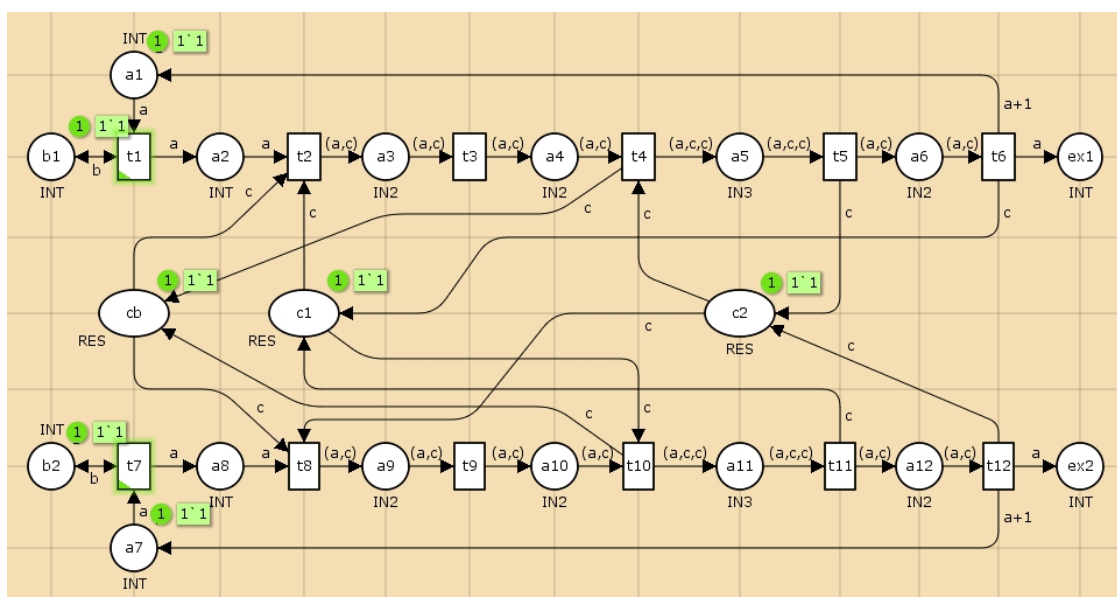


Рисунок 4 – Преобразованная сеть Петри

Состояния  $S_{12}$ ,  $S_{13}$ ,  $S_{21}$ ,  $S_{31}$ , которые в исходном графе были опасными, становятся безопасными состояниями блокировки, а состояние  $S_{11}$  – состоянием конфликта. Ясно, что при этом запрещенные в графе состояния  $S_{22}$ ,  $S_{32}$ ,  $S_{23}$ ,  $S_{33}$  станут недостижимыми. Результат работы преобразованной сети Петри после выполнения ста шагов представлен на рис. 6.

Вторым из возможных способов устранения взаимных блокировок процессов и тупиковых состояний рассмотрим способ обеспечения дополнительных ресурсов, позволяющих реализовать потребности процессов одновременно. Вариант решения для рассматриваемого примера приведен на рис. 7.

В данном случае в позициях  $c_1$  и  $c_2$  добавлено по одной метке, что соответствует наличию двух экземпляров однотипных ресурсов. Результат работы данной сети представлен на рис. 8.

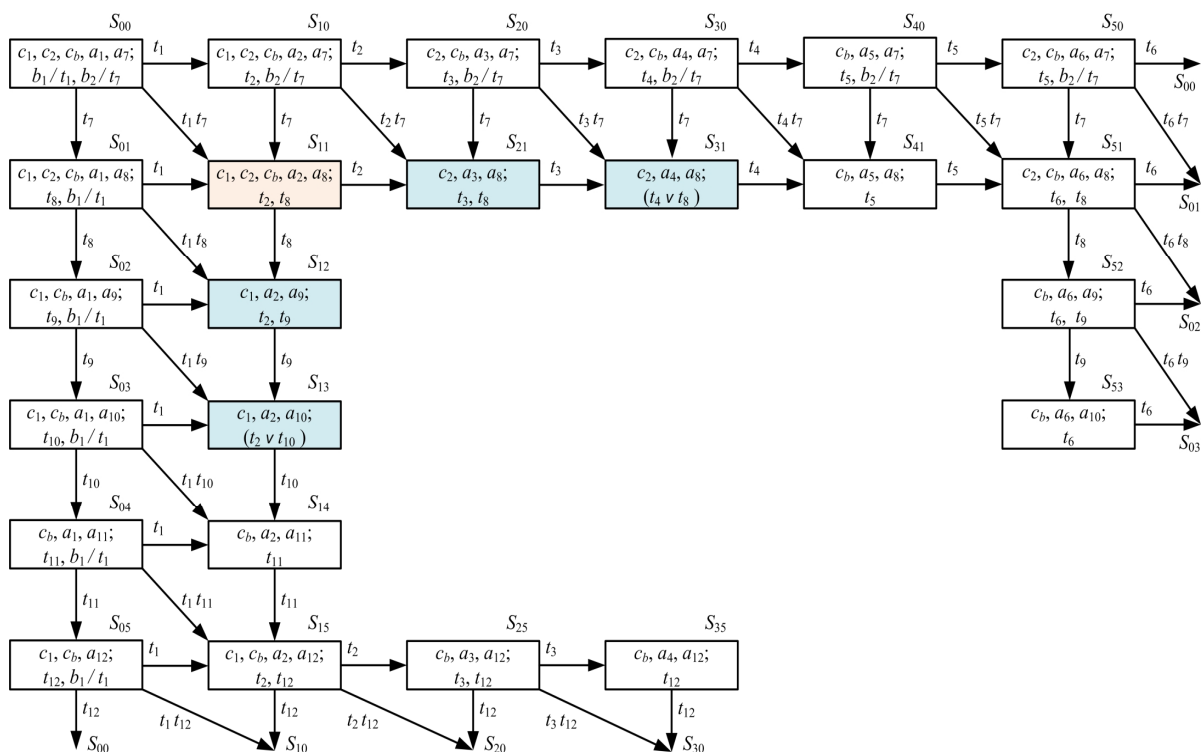


Рисунок 5 – Преобразованный граф достижимых маркировок

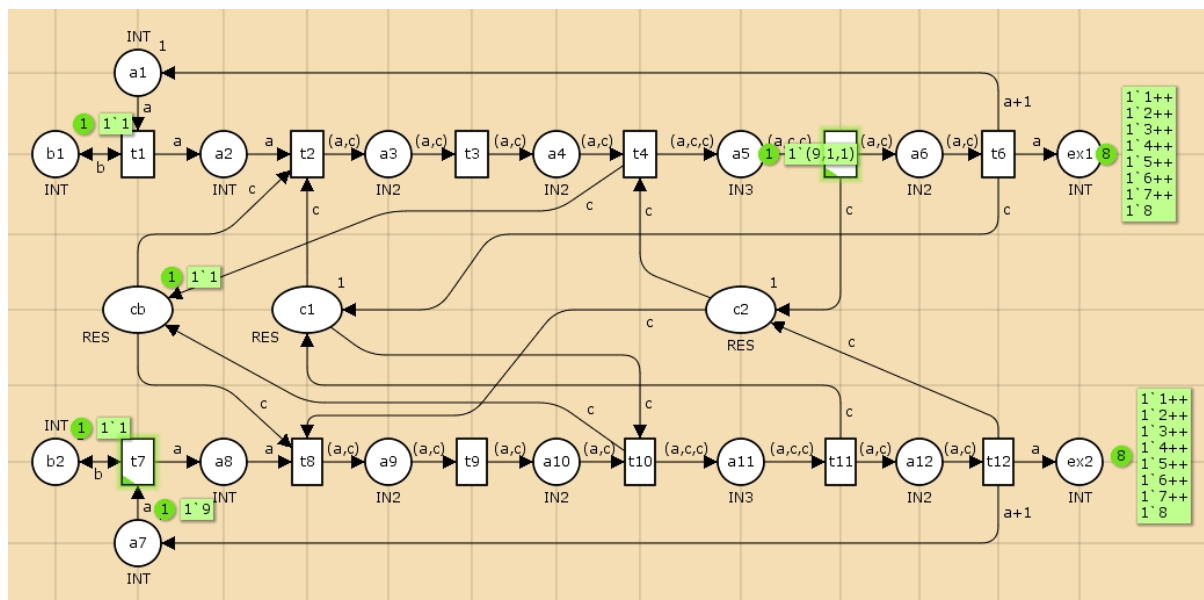


Рисунок 6 – Результаты работы преобразованной модели

Третий способ устранения взаимных блокировок процессов и тупиковых состояний предполагает обеспечение захвата «сразу» всех ресурсов необходимых для выполнения процесса, а не постепенно по мере необходимости. Данный принцип реализован в сети Петри представленной на рисунке 9.

Для устранения блокировки в исходную сеть добавлена позиция  $c_{21}$  обеспечивающая «временное хранение» второго ресурса. Результат работы данной сети представлен на рис. 10.

Последним – четвертым – из предложенных способов устранения взаимных блокировок процессов и тупиковых состояний является метод упорядочивания захвата ресурсов. Всем типам ресурсов задаются порядковые номера и обеспечивается дисциплина выборки ресурсов процессами. В данном примере этот подход реализуется поочередной сменой номеров ресурсов и заданием правила выбора в переходах  $t_2$ ,  $t_4$  и  $t_8$ ,  $t_{10}$  (рис. 11).

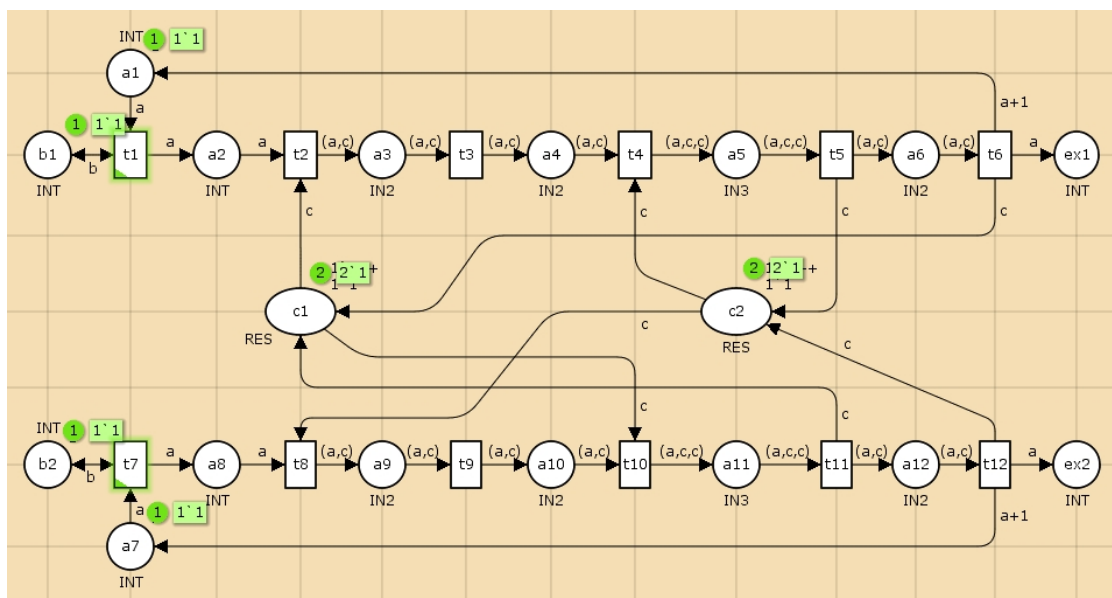


Рисунок 7 – Модель выполнения процессов с дополнительными ресурсами

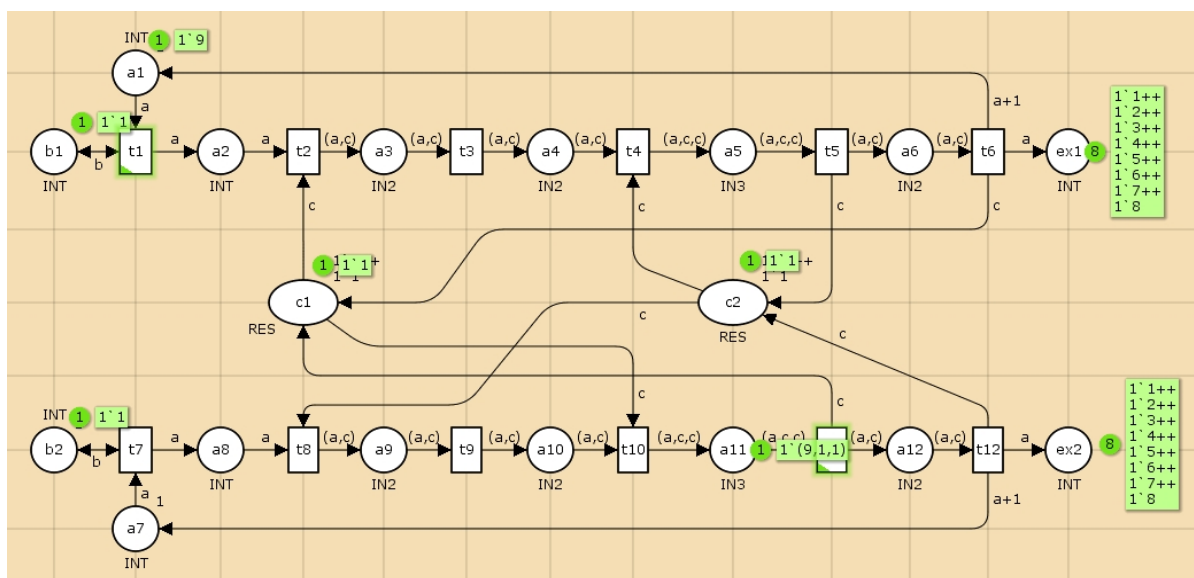


Рисунок 8 – Результаты работы модели с дополнительными ресурсами

Разработанные модели позволяют решить задачу устранения взаимной блокировки выполнения параллельных процессов. Методология раскрашенных сетей Петри позволяет проводить анализ полного пространства состояний модели для повышения надежности функционирования вычислительной системы и обеспечения соответствия требованиям. Предложенные методы и программные решения позволяют ускорить и облегчить процесс создания программ данного класса, органично вписываются в принятые в индустрии ПО подходы и методы разработки и готовы к реальному применению.

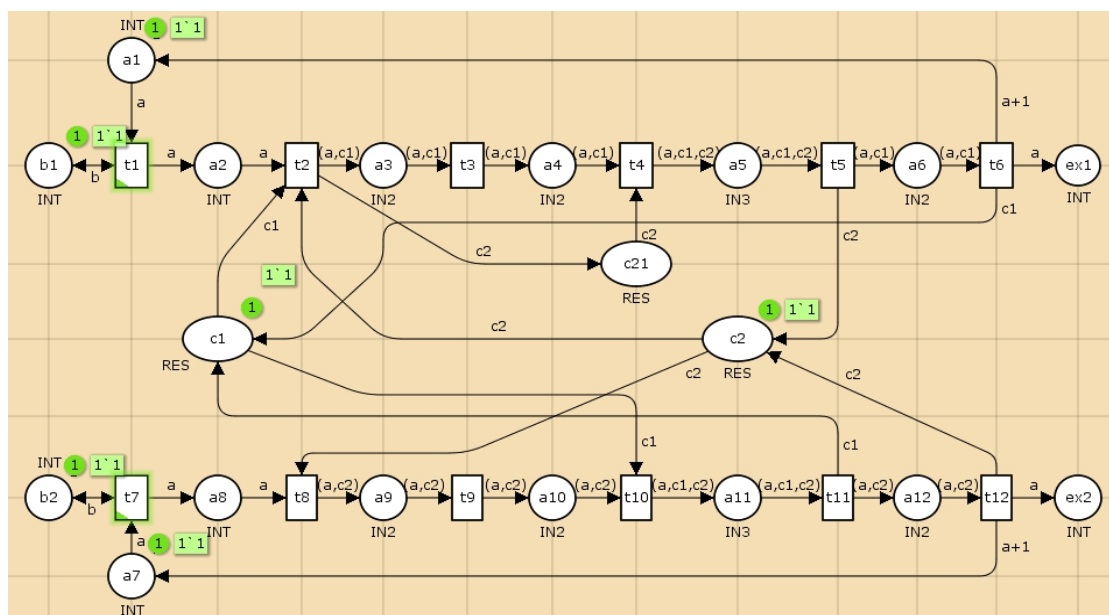


Рисунок 9 – Модель выполнения процессов с монопольным захватом ресурсов

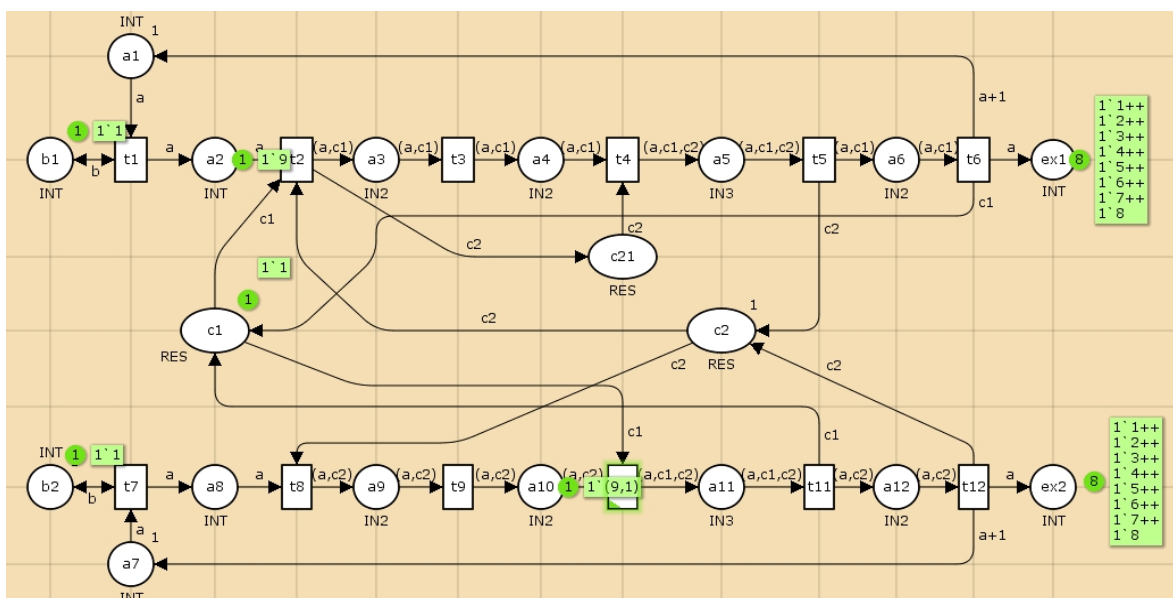


Рисунок 10 – Результаты работы модели с монопольным захватом ресурсов

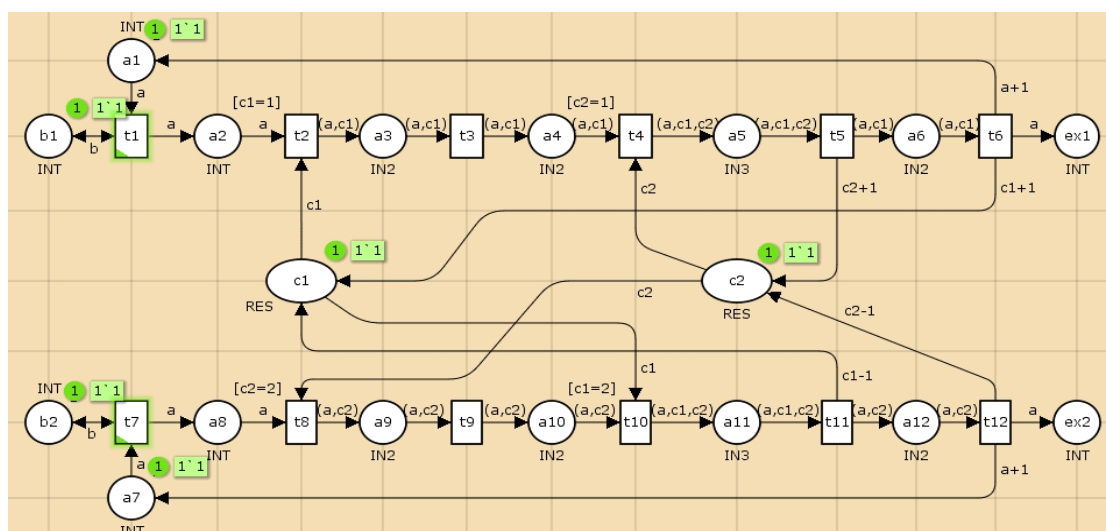


Рисунок 11 – Модель выполнения процессов с упорядочиванием захвата ресурсов



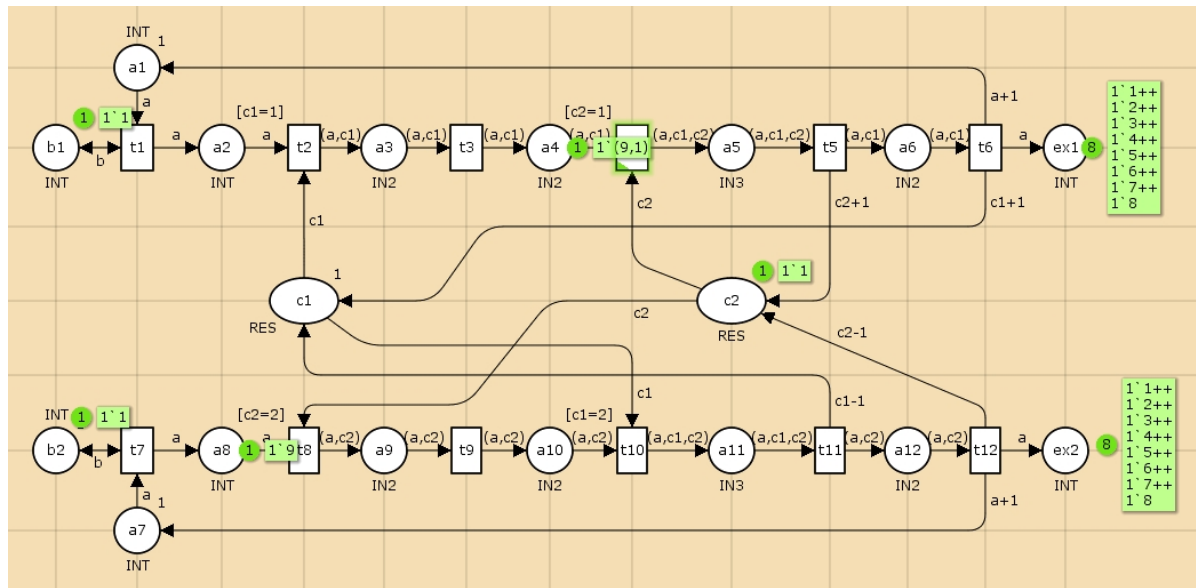


Рисунок 12 – Результаты работы модели с упорядочиванием захвата ресурсов

### Литература

1. Пийль, Е. И. Устранение взаимной блокировки параллельных процессов при статистическом распределении ресурсов [Текст] / Е. И. Пийль // Сетевые протоколы и управление в распределенных вычислительных системах : сборник / Акад. наук СССР – М. : Наука, 1986. – С. 116 – 125.
2. Jensen, K. Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use / K. Jensen. – Berlin : Springer/ Vol.1 . 1996, Vol.2 . 1997, Vol.3 . 1997.
3. Coloured Petri Nets. [Электронный ресурс]. Режим доступа : <http://cs.au.dk/CPnets/> (04.02.2013).
4. Шоу, А. Логическое проектирование операционных систем [Текст] / А. Шоу. – М. : Мир, 1981.