

ЧЕБЫШЕВСКИЙ СБОРНИК

Том 20. Выпуск 2.

УДК 004.942

DOI 10.22405/2226-8383-2019-20-2-504-516

Моделирование компьютерных систем с FIFO-дисциплиной обработки прерываний

Е. В. Ларкин, А. Н. Привалов

Ларкин Евгений Васильевич — доктор технических наук, профессор, заведующий кафедрой, Тульский государственный университет. (г. Тула).

e-mail: elarkin@mail.ru

Привалов Александр Николаевич — доктор технических наук, профессор, профессор кафедры информатики и информационных технологий, Тульский государственный педагогический университет им. Л. Н. Толстого. (г. Тула).

e-mail: privalov.61@mail.ru

Аннотация

Дисциплина FIFO (First In First Out) обработки прерываний достаточно широко используется в ЭВМ Фон Неймановского типа, применяемых в информационных и цифровых управляющих системах. Цель реализации подобного режима работы — оптимизация времени доступа к данным — достижима только при наличии адекватной модели, описывающей систему. Аналитическая модель построена с использованием фундаментального математического аппарата сетей Петри-Маркова. Первичная Петри-Марковская модель разделена на иерархические уровни, соответствующие количеству заявок на обработку прерываний в очереди. Показано, что с текущего уровня возможно переключение, как на предыдущий, так и на последующий уровни прерываний. Получены зависимости для определения времени пребывания на текущем уровне и вероятностей переключения на сопряженные уровни. Предложен метод преобразования Петри-Марковской модели в полумарковский процесс. Показано, что структура подобного полумарковского процесса представляет собой бинарное дерево. Получены зависимости для определения временных и вероятностных характеристик блужданий по бинарному дереву.

Ключевые слова: FIFO-дисциплина, прерывание, временные характеристики, сеть Петри-Маркова, полумарковский процесс, «соревнование», бинарное дерево

Библиография: 18 названий.

Для цитирования:

Е. В. Ларкин, А. Н. Привалов. Моделирование компьютерных систем с FIFO-дисциплиной обработки прерываний // Чебышевский сборник. 2019. Т. 20, вып. 2, С. 504–516.

CHEBYSHEVSKII SBORNIK

Vol. 20. No. 2.

UDC 004.942

DOI 10.22405/2226-8383-2019-20-2-504-516

Simulation of computer systems with FIFO-discipline interrupt handling

E. V. Larkin, A. N. Privalov

Larkin Eugene Vasilyevich — doctor of technical science, professor, head of chair, Tula State University. (Tula).

e-mail: elarkin@mail.ru

Privalov Alexander Nikolaevich — Doctor of Technical Sciences, Professor, Professor of the Department of Informatics and Information Technologies, Tula State Pedagogical University. L.N. Tolstoy. (Tula).

e-mail: privalov.61@mail.ru

Abstract

The FIFO (First In First Out) discipline of interrupt processing is widely used in Von Neumann type computers of informational and digital control systems. The goal of implementing such modes of operation - optimization time to data access - is achievable only when there is an adequate model, which describes data processing in the system. The analytical model is worked out with use the fundamental mathematical apparatus of Petri-Markov nets. The initial Petri-Markov model is divided into hierarchical levels in accordance with the number of interrupts in queue for processing. It is shown, that from the current level it is possible to switch both to the previous and to the next interrupt. Dependencies for determine the time of residence on the current level, and the probabilities of switching to conjugate levels are obtained. The method of Petri-Markov model transformation into the semi-Markov process is proposed. It is shown, that semi-Markov process obtained has the binary tree structure. Dependences for determining the time and probabilistic characteristics of wandering through a binary tree, are obtained.

Keywords: FIFO discipline, interruption, time characteristics, Petri-Markov net, semi-Markov process, "competition", binary tree

Bibliography: 18 titles.

For citation:

E. V. Larkin, A. N. Privalov, 2019, "Simulation of computer systems with FIFO-discipline interrupt handling", *Chebyshevskii sbornik*, vol. 20, no. 2, pp. 504–516.

1. Введение

ЭВМ Фон Неймановского типа, в которых реализован режим прерываний, широко используются при обмене данными в цифровых управляющих и информационных системах [1, 2, 3]. Прерывания, как правило, поступают от внешних источников сигналов, функционирующих независимо от самой ЭВМ. При поступлении соответствующего запроса процессор прекращает интерпретацию текущего (фонового) алгоритма и переходит к интерпретации алгоритма обработки прерываний, по завершении которой возвращается к интерпретации отложенного алгоритма [4, 5]. При этом возможен возврат как к алгоритму обработки самого раннего запроса при FIFO-дисциплине (First In First Out) обработки, так и предыдущего поступившего запроса при LIFO-дисциплине (Last In First Out).

Несмотря на широкое распространение систем с прерываниями, предварительное моделирование процесса обработки данных в них развито мало, что объясняется отсутствием подходов, которые позволяли бы, с одной стороны, адекватно описывать функционирование системы на физическом уровне, а с другой стороны приводили бы к достаточно простым расчетам временных характеристик системы. Вместе с тем, существует практически универсальный подход, основанный на теории полумарковских/Марковских процессов [6, 7, 8, 9, 10], позволяющий провести оценку временных интервалов с точностью до плотностей распределения и развитие теории полумарковских процессов - математический аппарат сетей Петри-Маркова (СПМ) [11], позволяющий проводить анализ «соревнований» в системе [12, 13, 14, 15], а следовательно оценивать величину временных интервалов при взаимодействии систем. Для формирования адекватной Петри-Марковской модели системы с FIFO-дисциплиной обработки прерываний ниже сделаны следующие допущения:

процессы в системе развиваются в реальном физическом времени;

как интервалы времени между прерываниями, так и интервалы времени, затрачиваемого на обработку прерываний, являются случайными, задаются с точностью до плотностей распределения и не зависят от предыстории процесса.

Это позволило построить первичную модель, учитывающую эффект «соревнования» в системе, и упростит модель до ординарного полумарковского процесса, позволяющего проводить оценку временных интервалов.

2. Модели компонентов системы

Сложность любого математического описания системы, включающей ряд взаимодействующих элементов, определяется сложностью описания ее компонентов, поэтому модели компонентов системы с прерываниями должны быть предельно простыми. В [16] показано, что интерпретация алгоритма любой сложности в физическом времени может быть представлена в виде ординарного полумарковского процесса μ_m следующего вида:

$$\mu = \{A, h(t)\}, \quad (1)$$

где t - время; $A = \{a_0, \dots, a_j, \dots, a_J\}$ - множество состояний; a_0 - стартовое состояние, моделирующее начало интерпретации алгоритма; a_J - поглощающее состояние, моделирующее окончание интерпретации алгоритма; $h(t)$ - полумарковская матрица;

$$h(t) = [h_{j,k}(t)] = p \otimes f(t); \quad (2)$$

$p = [p_{j,k}]$ и $f(t) = [f_{j,k}(t)] - (J+1) \times (J+1)$ стохастическая матрица и матрица плотностей распределения времени пребывания полумарковского процесса в состояниях, соответственно.

В полумарковской матрице (2), описывающей алгоритм, имеющий начало и конец элементы нулевого столбца, и J -й строки являются нулевыми, что вытекает из особенностей интерпретируемого процессором алгоритма (рис. 1 а). Для элементов, с нулевого по $(J-1)$ -й, справедливо следующее выражение:

$$\sum_{k=1}^J \int_0^\infty h_{j,k}(t) dt = 1, 0 \leq j < J. \quad (3)$$

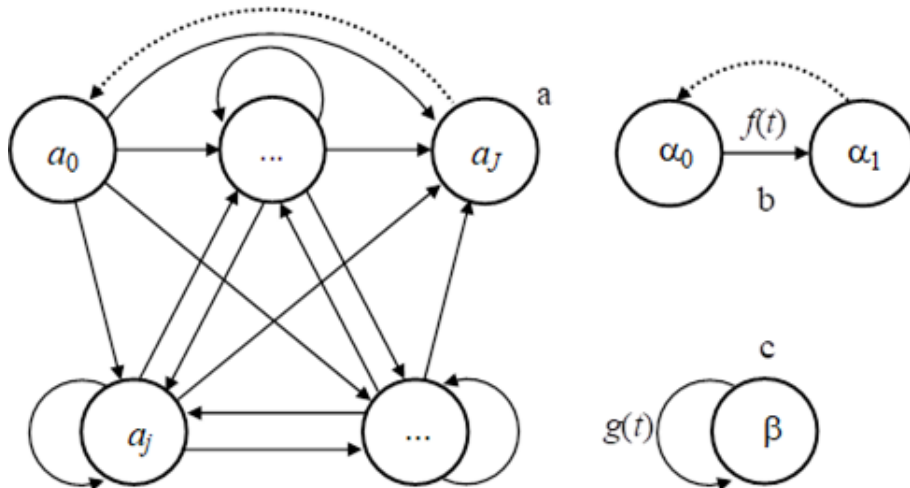


Рис. 1. Структуры элементов модели

На практике возможен случай, когда, например, фоновая программа является циклической, т.е. после окончания выполнения снова перезапускается сначала. В этом случае нулевой

элемент последней строки матрицы $h(t)$, $h_{J,0}(t) = \delta(t)$, где $\delta(t)$ - δ -функция Дирака. Эта связь показана пунктирной стрелкой на рис. 1 а.

При упрощении модель общего вида (2) преобразуется в модель, структура которой показана на рис. 2 б:

$$\mu \rightarrow \left\{ \begin{array}{l} \left\{ \{\alpha_0, \alpha_1\}, \begin{bmatrix} 0 & f(t) \\ 0 & 0 \end{bmatrix} \right\} - 4; \text{O A; CG0O} := 5\text{G} \Rightarrow 3 > 0; 3 > @8\text{B} < 0; \\ \left\{ \{\alpha_0, \alpha_1\}, \begin{bmatrix} 0 & f(t) \\ \delta(t) & 0 \end{bmatrix} \right\} - 4; \text{O A; CG0O F8} :: 8\text{G5A} :> 3 > 0; 3 > @8\text{B} < 0; \end{array} \right. \quad (4)$$

где $f(t)$ - плотность распределения времени интерпретации алгоритма;

В общем случае плотность распределения $f(t)$ для конечного алгоритма определяется по следующей зависимости [16]:

$$f(t) = L^{-1} \left[I_0^R \cdot \sum_{w=1}^{\infty} \{L[h(t)]\}^w \cdot I_J^C \right], \quad (5)$$

где I_0^R - $(J+1)$ -мерный вектор строка, в котором нулевой элемент равен единице, а все остальные элементы равны нулю; I_J^C - $(J+1)$ -мерный вектор-столбец, J -й элемент которого равен единице, а все остальные элементы равны нулю; $L[\dots]$ и $L^{-1}[\dots]$ - прямое и обратное преобразования Лапласа.

Для определения плотности распределения $f(t)$ в случае циклического алгоритма, в $h(t)$ необходимо выполнить подстановку $h_{J,0}(t) \Leftarrow 0$, после чего к сформированной таким образом матрицы применить зависимость (5)..

Внешний генератор прерываний, представленный на рис. 2 с, имеет вид

$$\nu \rightarrow \{\{\beta\}, [g(t)]\}, \quad (6)$$

где β - единственное возвратное состояние; $g(t)$ - плотность распределения времени возврата в состояние.

Для описания рекурсивного «соревнования» процессов, описываемых плотностями $f(t)$ и $g(t)$, вводятся функции ${}^k\varphi_f(t)$ и ${}^k\varphi_g(t)$, которые соответственно обозначают плотности распределения времени, оставшегося до окончания обработки прерывания и до поступления очередного прерывания на k -м этапе рекурсии.

Взвешенные плотности распределения времени завершения этапа «победителем» определяются по зависимостям:

$$\begin{cases} {}^k\varphi_f(t) / {}^k\varphi_g(t) = {}^k\eta_{f/g}(t) = {}^k\varphi_f(t) \cdot [1 - {}^k\Phi_g(t)]; \\ {}^k\varphi_g(t) / {}^k\varphi_f(t) = {}^k\eta_{g/f}(t) = {}^k\varphi_g(t) \cdot [1 - {}^k\Phi_f(t)], \end{cases} \quad (7)$$

где $\dots\Phi_{\dots}(t) = \int_0^t \dots\varphi_{\dots}(\tau) d\tau$.

Вероятности и плотности распределения времени завершения этапа «победителем» равны, соответственно

$$\begin{cases} {}^k\pi_{f/g} = \int_0^{\infty} {}^k\eta_{f/g}(t) dt; \\ {}^k\pi_{g/f} = \int_0^{\infty} {}^k\eta_{g/f}(t) dt; \end{cases} \quad \begin{cases} {}^k\varphi_{f/g}(t) = \frac{{}^k\eta_{f/g}(t)}{{}^k\pi_{f/g}}; \\ {}^k\varphi_{g/f}(t) = \frac{{}^k\eta_{g/f}(t)}{{}^k\pi_{g/f}}. \end{cases} \quad (8)$$

Плотности распределения времени ожидания «победителями» завершения этапа «побежденными» определяется по зависимостям

$$\begin{cases} {}^k\varphi_f(t) \rightarrow {}^k\varphi_g(t) = {}^k\varphi_{f \rightarrow g}(t) = \frac{1(t) \int_0^{\infty} {}^k\varphi_f(\tau) {}^k\varphi_g(t+\tau) d\tau}{\int_0^{\infty} {}^k\Phi_f(t) d{}^k\Phi_g(t)}; \\ {}^k\varphi_g(t) \rightarrow {}^k\varphi_f(t) = {}^k\varphi_{g \rightarrow f}(t) = \frac{1(t) \int_0^{\infty} {}^k\varphi_g(\tau) {}^k\varphi_f(t+\tau) d\tau}{\int_0^{\infty} {}^k\Phi_g(t) d{}^k\Phi_f(t)}, \end{cases} \quad (9)$$

где $1(t)$ - единичная функция Хевисайда.

С использованием зависимостей (7), (8), (9) может быть построена аналитическая модель прерываний.

3. Петри-Марковская модель прерываний

Каждый по отдельности полумарковский процесс (4) и (5) не зависит от другого, но в системе исследуемого типа они вступают в «соревнование», а после «победы» одного из процессов происходит переключение обоих компонентов системы в новое состояние, определяемое «победителем». Для моделирования подобного явления использован математический аппарат сетей Петри-Маркова (СПМ) [11].

Структура СПМ, описывающей «соревнование» в системе с прерываниями, приведена на рис. 2. За ее основу была принята СПМ, описанная в [17]. Точечной стрелкой на структуре показана связь, возникающая при моделировании циклического алгоритма. В алгоритме, имеющем начало и конец, указанная связь отсутствует. Система с прерываниями описывается следующей шестеркой:

$$\Pi = \{A, Z, X(Z), Y(Z), U(t), \Lambda\}, \quad (10)$$

где $A = a_0 \cup \bigcup_k A_k$ - множество мест; включающее a_0 - место, моделирующее начало функционирования FIFO-системы; и $A_k = \{^k a_1, ^k a_2, ^k a_3\}$ $k = 1, 2, \dots$ - подмножества мест, моделирующих процессы, участвующие в «соревновании» k -го уровня рекурсии; $^k a_1$ - место, моделирующее процесс интерпретации фоновой программы, или процесс обработки прерываний (для простоты, не нарушая общности, программы обработки прерываний и фоновая считаются одинаковыми); $^k a_2$ - место, моделирующее функционирование генератора прерываний, $^k a_3$ - место, моделирующее возврат на предыдущий уровень рекурсии; $Z = \{z_1, \dots, z_k, \dots\}$ - множество переходов, определяющих уровень рекурсии; $X(Z) = \{X(z_1), \dots, X(z_k), \dots\}$ - входная функция переходов, описывающая множество мест, из которых можно попасть в соответствующий переход за один полушаг; $Y(Z) = \{Y(z_1), \dots, Y(z_k), \dots\}$ - выходная функция переходов, описывающая множество мест, в которые можно попасть из соответствующего перехода за один полушаг; $U(t) = \{U_1(t), \dots, U_k(t), \dots\}$ - множество плотностей распределения времени выполнения полушагов из мест; $\Lambda = \{\Lambda_1, \dots, \Lambda_k, \dots\}$ - множество логических условий выполнения полушагов из переходов;

$$X(z_1) = \{a_0, {}^3 a_3, {}^1 a_3\}; X(z_k) = \{^{k-1} a_1, {}^{k-1} a_2, {}^{k+2} a_3\}, k = 2, 3, \dots; \quad (11)$$

$$Y(z_k) = \{^{k+1} a_1, {}^{k+1} a_2, {}^{k+1} a_3\}, k = 2, 3, \dots; \quad (12)$$

$$U_k(t) = \{^k \varphi_f(t), ^k \varphi_g(t), \delta(t)\}; \quad (13)$$

$$\Lambda_1 = \{\Lambda_1(z_1, {}^1 a_1), \Lambda_1(z_1, {}^1 a_2)\};$$

$$\Lambda_1(z_1, {}^1 a_1) = \Lambda_1(z_1, {}^1 a_2) = \lambda(a_0, z_1) \vee \lambda({}^2 a_3, z_1) \vee \lambda({}^1 a_3, z_1);$$

$$\Lambda_k = \{\Lambda_k(z_k, {}^k a_1), \Lambda_k(z_k, {}^k a_2), \Lambda_k(z_k, {}^k a_3)\};$$

$$\Lambda_k(z_k, {}^k a_1) = \Lambda_k(z_k, {}^k a_2) = \lambda({}^{k-1} a_2, z_k) \vee \lambda({}^{k+1} a_3, z_k);$$

$$\Lambda_k(z_k, {}^k a_3) = \lambda({}^{k-1} a_1, z_k), k = 2, 3, \dots \quad (14)$$

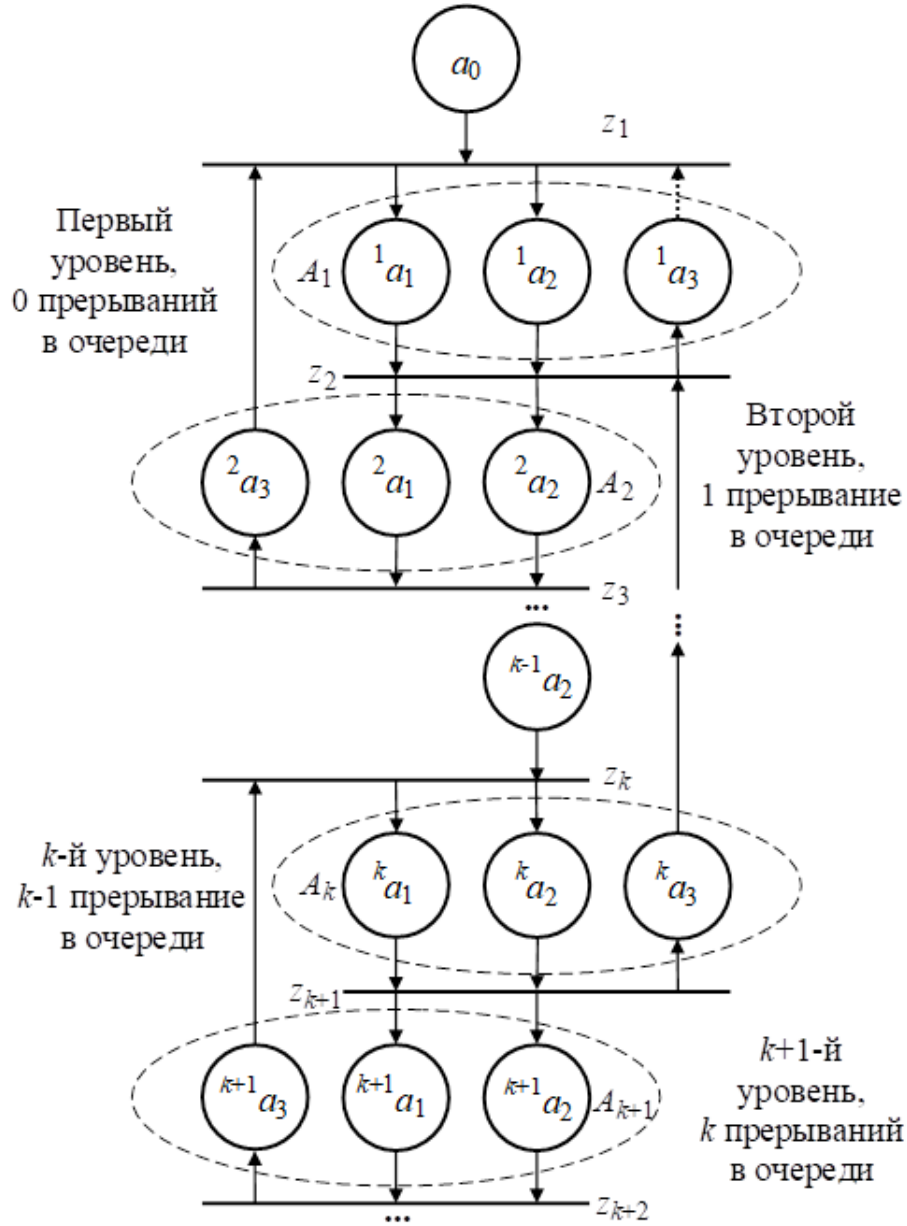


Рис. 2. Сеть Петри-Маркова, моделирующая алгоритм с прерываниями

Множество $X(z_1)$ имеет мощность 2 и имеет вид $X(z_1) = \{a_0, {}^3a_3\}$, если моделируется FIFO-система с прерываниями, основной (фоновый) алгоритм которой имеет начало и конец. Мощность множества $X(z_1)$ имеет мощность 3 и имеет вид, как показано в (11), если моделируется FIFO-система с прерываниями, основной (фоновый) алгоритм которой является циклическим.

В (13), (14) ${}^k\varphi_f(t)$, ${}^k\varphi_g(t)$ - плотности распределения, которые получаются в соответствии с зависимостями (7), (8), (9) в результате анализа «соревнования» на соответствующем уровне рекурсии; $\delta(t)$ - δ -функция Дирака; $\Lambda_k(z_k, {}^ka_i)$ - логические условия выполнения полшага $(z_k, {}^ka_i)$, $k = 1, 2, \dots, i = 1, 2, 3$; (a_0, z_1) , $\lambda({}^{k-1}a_1, z_k)$, $\lambda({}^{k-1}a_2, z_k)$, $\lambda({}^{k+1}a_3, z_k)$, $k = 1, 2, \dots$ - логические переменные, принимающие значение

$$\lambda(\dots) = \begin{cases} 0, & \text{when corresponding semi - step is not executed;} \\ 1, & \text{when corresponding semi - step is executed.} \end{cases} \quad (15)$$

Кроме того, дизъюнкция в логическом выражении для $\Lambda_1(z_1, {}^1a_1) = \Lambda_1(z_1, {}^1a_2)$ содер-

жит две логические переменные, если моделируется FIFO-система с прерываниями, основной (фонový) алгоритм которой имеет начало и конец, и три логические переменные, если алгоритм является циклическим.

Структура СПМ описывает многоуровневую. процедуру парных «соревнований» генератора и обработчика прерываний.

Первый уровень «соревнований» реализуется следующим образом.

1.1. Производится подстановка

$${}^1\varphi_f(t) \Leftarrow f(t); {}^1\varphi_g(t) \Leftarrow g(t). \quad (16)$$

1.2. Проверяется выполнение полушага (a_0, z_1) . Если $(a_0, z_1) = 0$, то система находится в режиме ожидания запуска, в противном случае запускается «соревнование» плотностей распределения ${}^1\varphi_f(t)$ и ${}^1\varphi_g(t)$.

1.3. Определяется «победитель» «соревнования». Если «победителем» является плотность ${}^1\varphi_f(t)$, то взвешенная плотность распределения времени выполнения полушага $({}^1a_1, z_2)$ определяется по зависимости (7) для случая ${}^1\varphi_f(t)/{}^1\varphi_g(t)$, вероятность ${}^1\pi_{f/g}$ и чистое распределение времени выполнения полушага $({}^1a_1, z_2)$, ${}^1\varphi_{f/g}(t)$, определяются по зависимостям (8).

1.4. Выполняется полушаг $({}^1a_1, z_2)$ в соответствии с логическими условиями (14), который инициирует выполнение полушага $(z_2, {}^1a_3)$. Место 1a_3 моделирует поглощающее состояние полумарковского процесса в случае, если фоновый алгоритм имеет начало и конец, и перезапускает процесс в случае, если фоновый алгоритм является циклическим.

1.5. При перезапуске процесса определяется плотность распределения времени, оставшегося до поступления очередного запроса на прерывание, ${}^1\varphi_f(t) \rightarrow {}^1\varphi_g(t)$, по зависимости (9),

1.6. Производится подстановка

$${}^1\varphi_f(t) \Leftarrow {}^1f(t); {}^1\varphi_g(t) \Leftarrow {}^1\varphi_{f \rightarrow g}(t), \quad (17)$$

и производится перезапуск процесса.

1.7. Если после определения «победителя» в «соревновании», которое анализируется в соответствии с п. 1.3. является плотность ${}^1\varphi_g(t)$, то взвешенное распределение времени выполнения полушага $({}^1a_2, z_2)$ определяется по зависимости (7) для случая ${}^1\varphi_g(t)/{}^1\varphi_f(t)$, вероятность ${}^1\pi_{f/g}$ и чистая плотность распределения времени выполнения полушага $({}^1a_2, z_2)$, ${}^1\varphi_{g/f}(t)$, определяются по зависимостям (8). Кроме того, в этом случае определяется ${}^1\varphi_g(t) \rightarrow {}^1\varphi_f(t)$.

1.8. Производится подстановка

$${}^2\varphi_f(t) \Leftarrow {}^1\varphi_{g \rightarrow f}(t) \rightarrow {}^1\varphi_f(t); {}^2\varphi_g(t) \Leftarrow g(t), \quad (18)$$

которая подготавливает следующий шаг рекурсии.

«Соревнование» на k -м уровне, если для попадания на указанный уровень был выполнен полушаг $({}^{k-1}a_2, z_k)$, реализуется следующим образом.

$k.1.1.$ Определяется плотность распределения времени, оставшегося до выполнения полушага $({}^ka_1, z_{k+1})$, ${}^{k-1}\varphi_{g \rightarrow f}(t)$, по зависимости (9) и выполняется подстановка

$${}^k\varphi_f(t) \Leftarrow {}^{k-1}\varphi_{g \rightarrow f}(t); {}^k\varphi_g(t) \Leftarrow g(t). \quad (19)$$

$k.1.2.$ Запускается «соревнование» плотностей распределения ${}^k\varphi_f(t)$ и ${}^k\varphi_g(t)$.

$k.1.3.$ Определяется «победитель» «соревнования». Если «победителем» является плотность ${}^k\varphi_f(t)$, то взвешенное распределение времени выполнения полушага $({}^ka_1, z_{k+1})$ определяется по зависимости (7) для случая ${}^k\varphi_f(t)/{}^k\varphi_g(t)$, вероятность ${}^k\pi_{f/g}$ и чистое распределение времени выполнения полушага $({}^ka_1, z_{k+1})$, ${}^k\varphi_{f/g}(t)$, определяются по зависимостям (8).

k.1.4. Выполнение полушага $(^k a_1, z_{k+1})$ в соответствии с логическими условиями (14) инициирует выполнение полушагов $(z_{k+1}, ^k a_3)$, $(^k a_3, z_{k-1})$, и система возвращается на $(k-1)$ -й уровень прерываний.

k.1.5. В случае, если в п. *k.1.3* «победителем» является плотность $^k \varphi_g(t)$, то взвешенное распределение времени выполнения полушага $(^k a_2, z_{k+1})$ определяется по зависимости (7) для случая $^k \varphi_g(t) / ^k \varphi_f(t)$, вероятность $^k \pi_{g/f}$ и чистое распределение времени выполнения полушага $(^k a_2, z_{k+1})$, $^k \varphi_{g/f}(t)$, определяются по зависимостям (8). Кроме того, в этом случае определяется $^k \varphi_g(t) \rightarrow ^k \varphi_f(t)$.

k.1.6. Производится подстановка

$$^{k+1} \varphi_f(t) \Leftarrow ^k \varphi_{g \rightarrow}(t) \rightarrow ^k \varphi_f(t); ^{k+1} \varphi_g(t) \Leftarrow g(t), \quad (20)$$

которая подготавливает следующий шаг рекурсии.

«Соревнование» на k -м уровне, если для попадания на указанный уровень был выполнен полушаг $(^{k+1} a_3, z_k)$ (в «соревновании» на $(k+1)$ -м, уровне «победила» плотность распределения $^{k+1} \varphi_f(t)$) реализуется следующим образом.

k.2.1. Определяется плотность распределения времени, оставшегося до выполнения полушага $(^k a_2, z_{k+1})$, $^{k+1} \varphi_{f \rightarrow g}(t)$, по зависимости (9) и выполняется подстановка

$$^k \varphi_1(t) \Leftarrow f(t), ^k \varphi_g(t) \Leftarrow ^{k+1} \varphi_{f \rightarrow g}(t); . \quad (21)$$

k.2.2. Запускается «соревнование» плотностей распределения $^k \varphi_f(t)$ и $^k \varphi_g(t)$.

k.2.3. Определяется «победитель» «соревнования». Если «победителем» является плотность $^k \varphi_f(t)$, то взвешенное распределение времени выполнения полушага $(^k a_1, z_{k+1})$ определяется по зависимости (7) для случая $^k \varphi_f(t) / ^k \varphi_g(t)$, вероятность $^k \pi_{f/g}$ и чистое распределение времени выполнения полушага $(^k a_1, z_{k+1})$, $^k \varphi_{f/g}(t)$, определяются по зависимостям (8).

k.2.4. Выполнение полушага $(^k a_1, z_{k+1})$ в соответствии с логическими условиями (14) инициирует выполнение полушагов $(z_{k+1}, ^k a_3)$, $(^k a_3, z_{k-1})$, и система возвращается на $(k-1)$ -й уровень прерываний.

k.2.5. В случае, если в п. *k.1.3* «победителем» является плотность $^k \varphi_g(t)$, то взвешенное распределение времени выполнения полушага $(^k a_2, z_{k+1})$ определяется по зависимости (7) для случая $^k \varphi_g(t) / ^k \varphi_f(t)$, вероятность $^k \pi_{g/f}$ и чистое распределение времени выполнения полушага $(^k a_2, z_{k+1})$, $^k \varphi_{g/f}(t)$, определяются по зависимостям (8). Кроме того, в этом случае определяется $^k \varphi_g(t) \rightarrow ^k \varphi_f(t)$.

k.2.6. Производится подстановка

$$^{k+1} \varphi_f(t) \Leftarrow ^k \varphi_{g \rightarrow}(t) \rightarrow ^k \varphi_f(t); ^{k+1} \varphi_g(t) \Leftarrow g(t), \quad (22)$$

которая подготавливает следующий шаг рекурсии.

Приведенная рекурсивная модель системы FIFO с прерываниями отличается от модели LIFO (Last In First Out) с прерываниями [17] тем, что при переключении с текущего на другой уровень в системе производятся разные подстановки.

4. Модель прерываний как полумарковский процесс

Процесс переключений в системе с прерываниями, в общем случае является бесконечным, но не является ни Марковским, ни полумарковским. Это происходит из-за необходимости при моделировании выполнения подстановок (16) - (22), которые определяются предысторией переключений. Однако, если на каждом уровне рекурсии оценивать плотности распределения

времени пребывания на нем и вероятности «победы» и «поражения» в «соревновании» генератора и обработчика прерываний, то может быть сформирован полумарковский процесс переключений в системе. Вследствие того, что при переключениях с различными предысториями оценки плотностей распределения и вероятностей является различными, полумарковский процесс имеет древовидную структуру (рис. 3). Вследствие того, что «соревнования» на каждом уровне являются парными (см. (7)), дерево является бинарным.

В структуре, приведенной на рис. 3, вершина, отмеченное двойной линией, является стартовым состоянием полумарковского процесса и моделирует начало функционирования FIFO-системы (место a_0 на рис. 2). Вершины, отмеченные штриховкой, являются поглощающими состояниями полумарковского процесса и моделируют окончание интерпретации фонового алгоритма FIFO-системой (место 1a_3 на рис. 2). В остальные вершины вписаны целые числа, обозначающие номер уровня рекурсии, из структуры, приведенной на рис. 2.

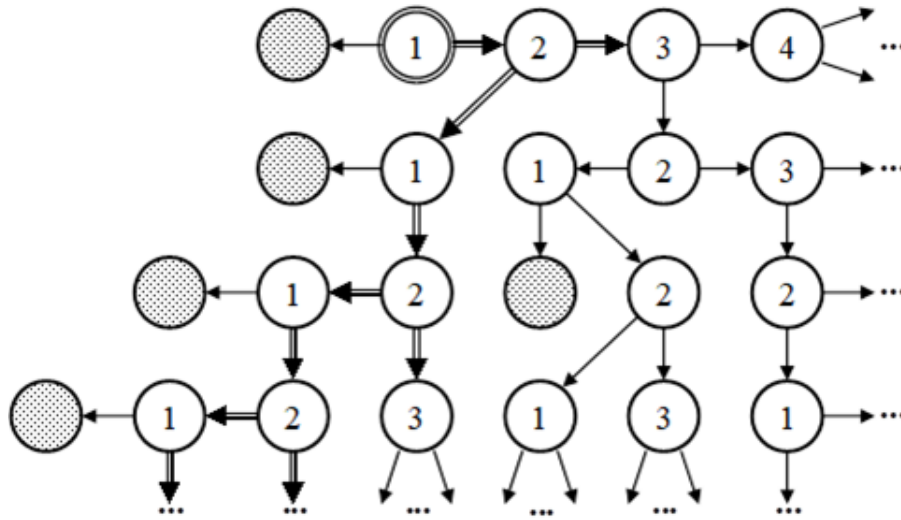


Рис. 3. Структура полумарковского процесса

Основными задачами, решаемыми с использованием модели FIFO-системы, являются:
задача оценки времени достижения системой состояния с R заявками на обработку прерываний из стартового состояния системы;

задача оценки времени переключения из состояния с R заявками на обработку прерываний в состояние с S заявками на обработку прерываний.

Для решения первой задачи необходимо:

- 1) проложить все возможные маршруты из стартового состояния в состояния с R заявками на обработку прерываний, встречающимися первый раз;
- 2) определить взвешенные плотности распределения времени достижения каждого из этих состояний из стартового состояния;
- 3) оценить вероятность и время достижения системой состояния с K заявками на обработку прерываний из стартового состояния системы.

Общепринятым методом поиска маршрутов с заданными свойствами является метод ветвей и границ [18] (на рис. 3 маршруты от стартового состояния до состояния с двумя заявками в очереди показаны стрелками с двойными линиями). Взвешенные плотности распределения, вероятности и чистые плотности распределения времени достижения каждого из состояний определяются в соответствии с зависимостями

$$\eta_m^{0-R}(t) = L^{-1} \left[\prod_n L \left[\eta_{g/f,m,n}^{0-R}(t) \right] \right]; \quad (23)$$

$$\pi_m^{0-R} = \int_0^\infty \eta_m^{0-R}(t) dt, \quad (24)$$

$$\varphi_m^{0-R}(t) = \frac{\eta_m^{0-R}(t)}{\pi_m^{0-R}}; \quad (25)$$

где $\eta_m^{0-R}(t)$ и π_m^{0-R} - соответственно взвешенная плотность распределения и вероятность достижения состояния с R прерываниями в очереди из стартового состояния по m -му маршруту; $\eta_{g/f,m,n}^{0-R}(t)$ - n -я взвешенная плотность распределения, лежащая на m -м маршруте, определяемая по зависимости (7); $\varphi_m^{0-R}(t)$ - чистая плотность распределения достижения состояния с R прерываниями в очереди из стартового состояния по m -му маршруту.

Оценка вероятности и времени достижения системой состояния с K заявками производится с помощью стохастического суммирования плотностей, полученных по зависимости (25):

$$\eta^{0-R}(t) = \sum_m \eta_m^{0-R}(t); \quad (26)$$

$$\pi^{0-R} = \sum_m \pi_m^{0-R}; \quad (27)$$

$$\varphi^{0-R}(t) = \frac{\eta^{0-R}(t)}{\pi^{0-R}}, \quad (28)$$

где $\eta^{0-R}(t)$, π^{0-R} - соответственно, взвешенная плотность распределения и вероятность достижения любого состояния с R прерываниями в очереди из стартового состояния; $\varphi^{0-R}(t)$ - чистая плотность распределения достижения состояния с R прерываниями в очереди из стартового состояния.

Для решения второй задачи необходимо:

- 1) проложить все возможные маршруты из стартового состояния в состояния с R заявками на обработку прерываний, встречающимися первый раз;
- 2) определить вероятности достижения каждого состояния с R заявками из стартового состояния;
- 3) для каждого найденного в соответствии с п. 1) состояния с R прерываниями положить все возможные маршруты, ведущие в состояния с S прерываниями и оценить вероятность и время их достижения;
- 4) оценить общее время достижения состояния с S прерываниями из состояния с R прерываниями.

В результате может быть получена зависимость, имеющая вид:

$$\varphi^{R-S}(t) = \frac{\sum_m \pi_m^{0-R} \sum_l L^{-1} \left[\prod_n L \left[\eta_{g/f,m,l,n}^{R-S}(t) \right] \right]}{\sum_m \pi_m^{0-R} \sum_l \prod_n \int_0^\infty \eta_{g/f,m,l,n}^{R-S}(t) dt}; \quad (29)$$

где $\eta_{g/f,m,l,n}^{R-S}(t)$ - взвешенная плотность распределения времени переключения на следующий уровень при движении из состояния с R прерываниями в состояние с S прерываниями на n -м этапе при движении по l -му маршруту из m -го состояния; π_m^{0-R} - вероятность попадания в m -е состояние из стартового, рассчитанная по зависимости (24).

Кроме предложенного, для определения временных интервалов может быть предложен любой другой метод анализа полумарковских процессов.

5. Заключение

В результате сформирована модель системы с прерываниями, построенная с использованием математического аппарата сетей Петри-Маркова и предложен метод упрощения первичной модели до строго полумарковского процесса, что позволило получить достаточно простые зависимости для расчета временных интервалов в системе. Общий подход является основой для

построения моделей более сложных систем, в том числе многозадачных, многопроцессорных и т.п. Дальнейшим развитием систем обработки данных с прерываниями является решение оптимизационной задачи, которая позволяет, например, минимизировать время ответа на прерывания

Исследования были проведены при поддержке Госпрограммы Минобрнауки РФ (№ 2.3121.2017/ПЧ).

СПИСОК ЦИТИРОВАННОЙ ЛИТЕРАТУРЫ

1. Tzafestas S.G. Introduction to Mobile Robot Control. Elsevier, 2014. 750 p.
2. Landau I.D., Zito G. Digital Control Systems, Design, Identification and Implementation. Springer, 2006. 484 p.
3. Астроцм J., Wittenmark B., Computer Controlled Systems: Theory and Design. Tsinghua University Press. Prentice Hall, 2002. 557 p.
4. Regehr J. Duongsaa U. 2005. Preventing interrupt overload // Proceedings of the Conference on Languages, Compilers, and Tools for Embedded Systems. 2005. - Pp. 50 – 58.
5. Czerwinski, M., Cutrell, E., Horvitz, E. Instant messaging and interruption: Influence of task type on performance // Proceedings of OZCHI. Sydney, Australia - 2000, - <https://interruptions.net/literature/Czerwinski-OZCHI00.pdf>.
6. Bielecki T.R., Jakubowski J., Niewkgiowski M. Conditional Markov chains: Properties, construction and structured dependence // Stochastic Processes and their Applications. V. 127, N. 4. 2017. Pp. 1125–1170.
7. Ching W.K., Huang X., Ng M.K., Siu T.K. Markov Chains: Models, Algorithms and Applications / International Series in Operations Research & Management Science. V. 189. Springer Science + Business Media NY, 2013. 241 p.
8. Markov A.A. Extension of the law of large numbers to dependent quantities // Izvestiia Fiz.-Matem. Obsch. Kazan Univ., (2-nd Ser.), - 1906, - Pp. 135–156
9. Howard R. A. Dynamic Probabilistic Systems. Vol. 1: Markov Models. Vol. II: Semi-Markov and Decision Processes. Courier Corporation, 2012.
10. Janssen J., Manca R. Applied Semi-Markov processes. Springer US, 2006. 310 p.
11. Larkin E.V., Malikov A.A., Ivutin A.N. Petri-Markov model of fault-tolerant computer systems // 4th International Conference on Control, Decision and Information Technologies (CoDIT). – 5-7 April 2017, Barcelona, Spain – IEEE, 2017. – P. 416-420.
12. Heymann M. Concurrency and Discrete Event Control // IEEE Control Syst. Mag. 1990. Vol. 10. Pp. 103 – 112.
13. Valk R. Concurrency in Communicating Object Petri Nets // Concurr. object-oriented Program. Petri nets. 2001. Pp. 164 – 195.
14. Ivutin A.N, Larkin E.V. Simulation of Concurrent Games // Bulletin of the South Ural State University. Series: Mathematical Modeling, Programming and Computer Software. – Chelyabinsk, 2015. – Vol. 8, № 2. – Pp. 43 - 54.

15. Larkin E.V., Ivutin A.N., Kotov V.V., Privalov A.N. Simulation of Relay-races // Bulletin of the South Ural State University. Mathematical Modelling, Programming & Computer Software. - 2016. - Vol. 9. - No 4. - Pp. 117 - 128.
16. Larkin E.V., Ivutin A.N. Estimation of Latency in Embedded Real-Time Systems // 3-rd Mediterranean Conference on Embedded Computing (MECO-2014). - 2014. - Budva, Montenegro, 2014. - Pp. 236 - 239.
17. Larkin E.V., Ivutin A.N., Troshina A. Model of interruptions in Swarm unit // Advances in swarm intelligence. Proceedings of 8-th International conference ICSI 2017. Fukuoka, Japan.. 2017. Part 1. Pp. 50 - 59.
18. Bienstock D., Graph searching, path-width, tree-width and related problems (a survey) // Reliability of Computer and Communication Networks, New Brunswick, NJ, 1989, in: DIMACS Ser. Discrete Math. Theoret. Comput. Sci. Vol. 5, Amer. Math. Soc., Providence, RI, 1991. Pp. 33–49.

REFERENCES

1. Tzafestas S.G. Introduction to Mobile Robot Control. Elsevier, 2014. 750 p.
2. Landau I.D., Zito G. Digital Control Systems, Design, Identification and Implementation. Springer, 2006. 484 p.
3. Astrum J., Wittenmark B., Computer Controlled Systems: Theory and Design. Tsinghua University Press. Prentice Hall, 2002. 557 p.
4. Regehr J. Duongsaa U. 2005. Preventing interrupt overload // Proceedings of the Conference on Languages, Compilers, and Tools for Embedded Systems. 2005. - Pp. 50 – 58.
5. Czerwinski, M., Cutrell, E., Horvitz, E. Instant messaging and interruption: Influence of task type on performance // Proceedings of OZCHI. Sydney, Australia - 2000, - <https://interruptions.net/literature/Czerwinski-OZCHI00.pdf>.
6. Bielecki T.R., Jakubowski J., Niewkigowski M. Conditional Markov chains: Properties, construction and structured dependence // Stochastic Processes and their Applications. V. 127, N. 4. 2017. Pp. 1125–1170.
7. Ching W.K., Huang X., Ng M.K., Siu T.K. Markov Chains: Models, Algorithms and Applications / International Series in Operations Research & Management Science. V. 189. Springer Science + Business Media NY, 2013. 241 p.
8. Markov A.A. Extension of the law of large numbers to dependent quantities // Izvestiia Fiz.-Matem. Obsch. Kazan Univ., (2-nd Ser.), - 1906, - Pp. 135–156
9. Howard R. A. Dynamic Probabilistic Systems. Vol. 1: Markov Models. Vol. II: Semi-Markov and Decision Processes. Courier Corporation, 2012.
10. Janssen J., Manca R. Applied Semi-Markov processes. Springer US, 2006. 310 p.
11. Larkin E.V., Malikov A.A., Ivutin A.N. Petri-Markov model of fault-tolerant computer systems // 4th International Conference on Control, Decision and Information Technologies (CoDIT). – 5-7 April 2017, Barcelona, Spain – IEEE, 2017. – P. 416-420.

12. Heymann M. Concurrency and Discrete Event Control // IEEE Control Syst. Mag. 1990. Vol. 10. Pp. 103 – 112.
13. Valk R. Concurrency in Communicating Object Petri Nets // Concurr. object-oriented Program. Petri nets. 2001. Pp. 164 – 195.
14. Ivutin A.N., Larkin E.V. Simulation of Concurrent Games // Bulletin of the South Ural State University. Series: Mathematical Modeling, Programming and Computer Software. – Chelyabinsk, 2015. – Vol. 8, № 2. – Pp. 43 - 54.
15. Larkin E.V., Ivutin A.N., Kotov V.V., Privalov A.N. Simulation of Relay-races // Bulletin of the South Ural State University. Mathematical Modelling, Programming & Computer Software. - 2016. - Vol. 9. - No 4. - Pp. 117 - 128.
16. Larkin E.V., Ivutin A.N. Estimation of Latency in Embedded Real-Time Systems // 3-rd Mediterranean Conference on Embedded Computing (MECO-2014). - 2014. - Budva, Montenegro, 2014. - Pp. 236 - 239.
17. Larkin E.V., Ivutin A.N., Troshina A. Model of interruptions in Swarm unit // Advances in swarm intelligence. Proceedings of 8-th International conference ICSI 2017. Fukuoka, Japan.. 2017. Part 1. Pp. 50 - 59.
18. Bienstock D., Graph searching, path-width, tree-width and related problems (a survey) // Reliability of Computer and Communication Networks, New Brunswick, NJ, 1989, in: DIMACS Ser. Discrete Math. Theoret. Comput. Sci. Vol. 5, Amer. Math. Soc., Providence, RI, 1991. Pp. 33–49.

Получено 18.03.2019 г.

Принято в печать 12.07.2019 г.