

ПРИБОРЫ И УПРАВЛЕНИЕ

УДК 004.942

ПЕТРИ-МАРКОВСКАЯ МОДЕЛЬ ПРЕРЫВАНИЙ

Е.В. Ларкин

Исследуется время интерпретации алгоритмов на ЭВМ Фон Неймановского типа. С использованием фундаментального аппарата полумарковских процессов построена аналитическая математическая модель для оценки времени выполнения программы при отсутствии прерываний. Показано, что внешние прерывания представляют собой результат работы независимого случайного процесса, который развивается параллельно с интерпретацией алгоритма. Для описания взаимодействия двух процессов применен аппарат сетей Петри-Маркова. Построена базовая структурно-параметрическая модель функционирования ЭВМ при наличии внешних прерываний. Показано, что в общем случае Петри-Марковская модель является бесконечной. Разработана рекурсивная процедура блуждания по сети Петри-Маркова для данного конкретного случая. Показано, что процесс блуждания не является полумарковским. Предложен метод трансформации Петри-марковской модели в строго полумарковский процесс.

Ключевые слова: ЭВМ Фон Неймановского типа, прерывание, временные характеристики, полумарковский процесс, сеть Петри-Маркова, модель, блуждание.

1. Введение. Режим прерываний широко используется в системах цифрового управления объектами [1, 2]. В соответствии с этим режимом, при поступлении прерывания центральный процессор прекращает интерпретацию текущего алгоритма и переходит к интерпретации алгоритма обработки прерываний, по завершении которой возвращается к выполнению отложенной программы. При практической реализации алгоритмов возможны прерывания двух типов, программные и аппаратные. Программные прерывания приводят к обычным ветвлениям в алгоритмах [3], поэтому учет их влияния на общее время интерпретации является достаточно несложной инженерной задачей. Аппаратные прерывания генерируются внешним источником, функционирование которых не зависит от ЭВМ, поэтому в компьютерной системе с аппаратными прерываниями имеет место

параллельный процесс, разворачивающийся во времени: с одной стороны, это ЭВМ, интерпретирующая текущий алгоритм, а с другой стороны - это внешний источник прерываний. Как и в любом параллельном процессе, в исследуемой системе между субъектами возникает «соревнование» [4, 5, 6] за пользование аппаратными ресурсами, в частности, процессором. Наличие «соревнования», и «победа» в «соревновании» генератора прерываний увеличивает время интерпретации основного алгоритма. Увеличение времени зависит от плотности потока команд на прерывание и от временных характеристик алгоритма обработки прерываний. Оценка времени интерпретации алгоритма в этом случае должна сводиться к анализу результатов «соревнования» между текущим алгоритмом и генератором прерываний. Модели, позволяющей производить подобную оценку, в настоящее время не существует, что объясняет необходимость и актуальность исследований в данной области.

2. Временные характеристики алгоритма без прерываний. Рассмотрим процесс выполнения алгоритма на ЭВМ Фон Неймановского типа. Он сводится к последовательной интерпретации операторов, которая разворачивается во времени. После выполнения очередного оператора переход к одному из следующих возможных операторов осуществляется для внешнего наблюдателя случайным образом. Время, в течение которого выполняется оператор, является случайным. В силу этого естественной моделью, описывающей интерпретацию алгоритма, является полумарковский процесс [7, 8, 9] вида

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

где $B = \{\beta_1, \dots, \beta_j, \dots, \beta_J, \beta_{J+1}, \dots, \beta_M\}$ - множество состояний;
 $h(t) = [h_{j,m}(t)]$ - полумарковская матрица, имеющая размер $M \times M$.

Типовая структура алгоритма самого общего вида приведена на рис. 1.

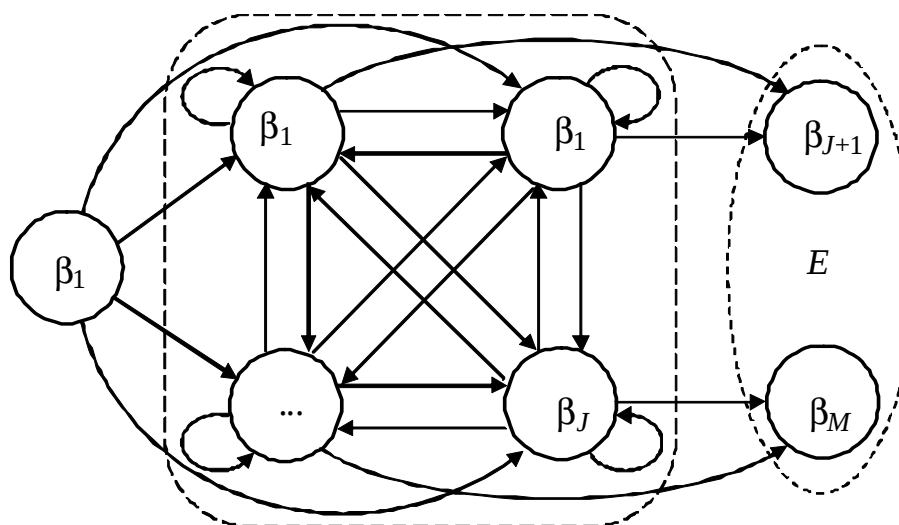


Рис. 1. Типовая структура алгоритма

Состояния полумарковского процесса, показанного на рис. 1, имеют следующий физический смысл:

β_1 - стартовое состояние, являющееся математическим аналогом оператора «начало» алгоритма;

$B \supset E = \{\beta_{J+1}, \dots, \beta_m, \dots, \beta_M\}$ - подмножество поглощающих состояний, являющихся математическим аналогом операторов «конец» алгоритма.

Структура полумарковского процесса такова, что для любого $\beta_j \notin E$, $j \neq 1$ всегда существует хотя бы один путь $\beta_1 \rightarrow \beta_j$ и хотя бы один путь $\beta_j \rightarrow E$. В силу этого структура и параметры полумарковского процесса (1) таковы, что в полумарковской матрице:

первый столбец содержит только нулевые элементы;

поглощающими состояниями являются только состояния подмножества E , т.е. строки с $(J + 1)$ -й по M -ю содержат только нулевые элементы;

для остальных строк

$$\sum_{m=1}^M \int_0^{\infty} h_{jm}(t) dt = 1, 1 \leq j \leq J. \quad (2)$$

Таким образом, все возможные пути $\beta_1 \rightarrow E$ составляют полную группу несовместных событий. Для этого случая плотность распределения времени достижения одного из состояний подмножества E из состояния β_1 определяется по зависимости

$$f(t) = L^{-1} \left[{}^r I_1 \cdot \sum_{k=1}^{\infty} \{L[h(t)]\}^k \cdot {}^c I_E \right], \quad (3)$$

где L и L^{-1} - соответственно прямое и обратное преобразования Лапласа; ${}^r I$ - вектор-строка, включающий M элементов, первый элемент которого равен единице, а остальные элементы равны нулю; ${}^c I_E$ - вектор-столбец, включающий M элементов, элементы с первого по J -й которого равны нулю, а остальные элементы равны единице.

Плотность распределения $f(t)$ определяет время выполнения алгоритма. По указанной плотности могут быть найдены минимальное, среднее и максимальное время выполнения:

$$T_{\min} = \min \arg f(t); T_{\max} = \max \arg f(t); \quad (4)$$

$$T = \int_0^{\infty} t f(t) dt; D = \int_0^{\infty} (t - T)^2 f(t) dt.$$

Полумарковский процесс после упрощения приведен на рис. 2 а.

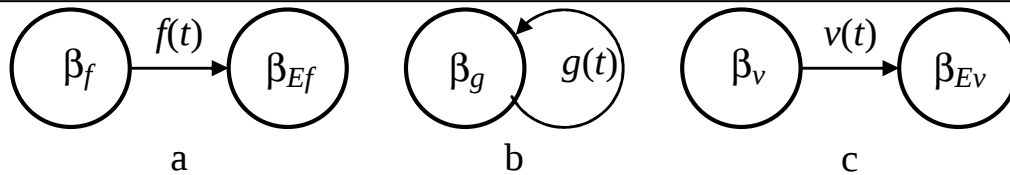


Рис. 2. Структура упрощенного полумарковского процесса *a*, генератора *b* и обработчика с прерываний

Внешний генератор прерываний представлен на рис. 2 б, а на рис. 2 с показан полумарковский процесс обработки прерываний. Естественно, что наличие внешних прерываний изменяет время выполнения алгоритма, поскольку вычислительные ресурсы ЭВМ Фон Неймановского типа в этом случае тратятся на интерпретацию алгоритма обработки прерываний¹.

3. Модель выполнения алгоритма при наличии прерываний. Рассмотрим случай, когда прерывания в ЭВМ Фон-Неймановского типа поступают от внешнего источника. Полумарковский процесс, описывающий интерпретацию алгоритма при наличии прерываний, является 3-параллельным и имеет следующий вид:

$$\tilde{\mu} = \{\{\beta_g, \beta_f, \beta_v, \beta_{Ef}, \beta_{Ev}\}, \tilde{h}(t)\}, \quad (5)$$

где t - время; $\beta_g, \beta_f, \beta_E$ - множество состояний; β_g - состояние, моделирующее генерацию одного прерывания; β_f - состояние, моделирующее выполнение алгоритма; β_v - состояние, моделирующее алгоритм обработки прерываний; β_{Ef} - поглощающее состояние полумарковской модели алгоритма; β_{Ev} - поглощающее состояние полумарковской модели обработчика прерываний;

$$\tilde{h}(t) = \begin{bmatrix} h_g(t) & 0_{12} & 0_{13} \\ 0_{21} & h_f(t) & 0_{23} \\ 0_{31} & 0_{32} & h_v(t) \end{bmatrix}; \quad (6)$$

$$h_g(t) = [g(t)]; \quad h_f(t) = \begin{bmatrix} 0 & f(t) \\ 0 & 0 \end{bmatrix}; \quad h_v(t) = \begin{bmatrix} 0 & v(t) \\ 0 & 0 \end{bmatrix}; \quad (7)$$

$$0_{12} = 0_{13} = (0, 0); \quad 0_{21} = 0_{31} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}; \quad 0_{23} = 0_{32} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

Полумарковский процесс (5) является не просто 3-параллельным. Все три элементарных процесса находятся во взаимодействии. Взаимодействие осуществляется по следующему правилу. При старте одновременно запускаются процессы $h_f(t)$ и $h_g(t)$, которые вступают в «соревнование» [10, 11, 12]. Если «побеждает» процесс $h_f(t)$, то интерпретация алгоритма заканчивается, в противном случае запускается процесс $h_v(t)$, который

«соревнуется» с процессом $h_g(t)$. Если в этом случае «побеждает» процесс $h_g(t)$, то вновь запускается обработчик прерываний. В противном случае происходит возврат в процесс $h_f(t)$, который продолжается с того момента, на котором был прерван, при этом «соревнование» происходит с продолжением ожидания срабатывания генератора прерываний. Теория графов не позволяет описать разделение процессов («fork») и слияние процессов («joint»). Поэтому далее для моделирования использован аппарат сетей Петри-Маркова [10, 11, 12].

Сеть Петри-Маркова (СПМ), описывающая функционирование алгоритма с прерываниями, приведена на рис. 3.

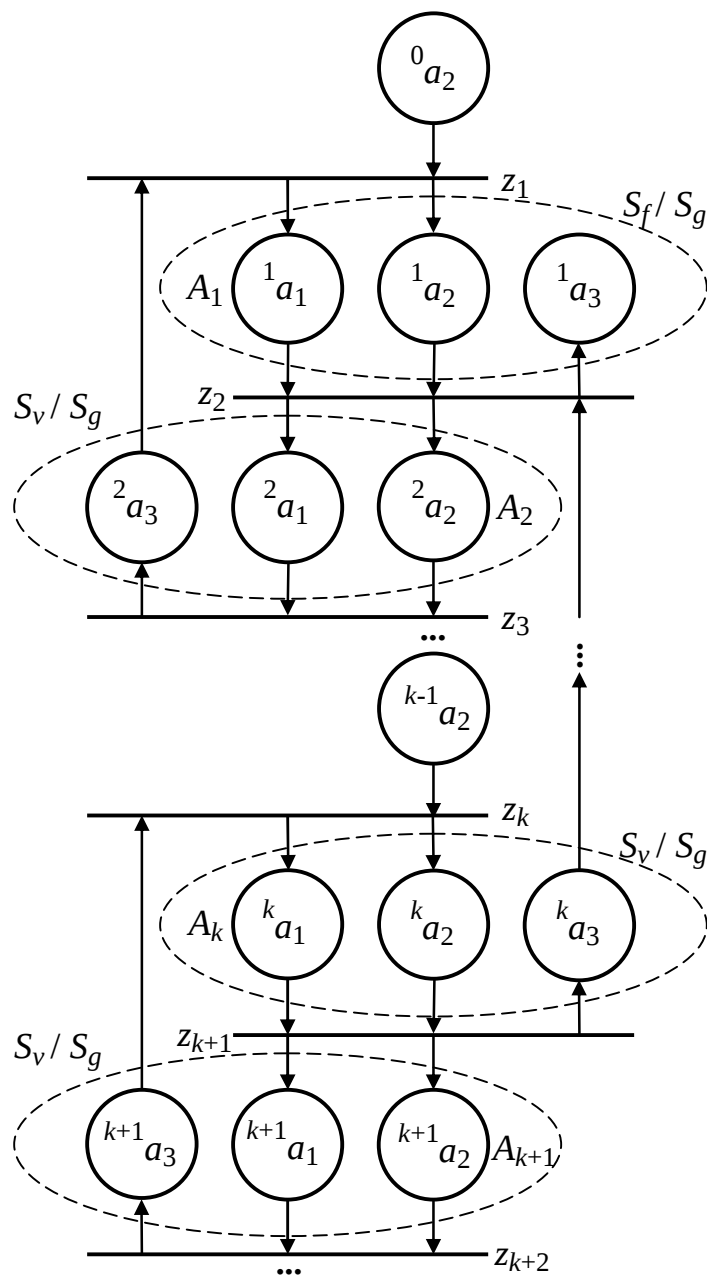


Рис. 3. Сеть Петри-Маркова, моделирующая алгоритм с прерываниями

СПМ имеет следующий вид:

$$\Pi = \{A, Z, \Phi(t), A\}, \quad (8)$$

где $A = \{^0a_2, ^1A, \dots, ^kA_k, \dots\}$ - множество мест, включающее место 0a_2 , моделирующее старт процесса, и подмножества $^kA = \{^ka_1, ^ka_2, ^ka_3\}$ $k = 1, 2, \dots$ мест, моделирующих процессы, участвующие в «соревновании» k -го уровня; ka_1 - место, моделирующее текущий алгоритм, интерпретируемый ЭВМ, которым может быть алгоритм (плотность распределения времени выполнения $v(t)$) обработки прерываний, или основной алгоритм (плотность распределения времени выполнения $f(t)$); ka_2 - место, моделирующее функционирование генератора прерываний, ka_3 - место, моделирующее возврат на предыдущий уровень обработки прерываний; $Z = \{z_1, \dots, z_k, \dots\}$ - множество переходов, определяющих уровень обработки прерываний; $\Phi(t)$ - матрица плотностей распределения; A - матрица логических условий выполнения полушагов из переходов.

Структура СПМ описывает многоуровневую. процедуру парных «соревнований» трех субъектов: генератора прерываний S_g с основным алгоритмом S_f и генератора прерываний S_g с обработчиком прерываний S_v

Первый уровень «соревнований» представлен подсетью от места 0a_2 до мест $^1a_3, ^2a_3$, т.е.

$$\Pi_1 = \{A_1, Z_1, \Phi_1(t), A_1\}, \quad (10)$$

где $A_1 = \{^0a_2, ^1a_1, ^1a_2, ^1a_3, ^2a_1, ^2a_2, ^2a_3\}$ - подмножество мест; $Z_1 = \{z_1, z_2, z_3\}$ - подмножество переходов; $\Phi_1(t) = [\varphi_{ij}(t)]$ - 7×3 матрица плотностей распределения, задающая временные интервалы «соревнований»; $A_1 = [\lambda_{ji}]$ - 3×7 матрица логических условий выполнения полушагов из переходов;

$$\Phi_1(t) = \begin{bmatrix} \delta(t) & 0 & 0 \\ 0 & ^1\varphi_1(t) & 0 \\ 0 & ^1\varphi_2(t) & 0 \\ 0 & 0 & 0 \\ 0 & 0 & ^2\varphi_1(t) \\ 0 & 0 & ^2\varphi_2(t) \\ \delta(t) & 0 & 0 \end{bmatrix}; \quad (11)$$

$$A_1 = [\lambda_{ji}]; \quad (12)$$

$$\begin{aligned} \lambda_{11} = \lambda_{14} = \lambda_{16} = \lambda_{17} = \lambda_{21} = \lambda_{22} = \lambda_{23} = \\ = \lambda_{27} = \lambda_{31} = \lambda_{32} = \lambda_{33} = \lambda_{34} = \lambda_{35} = \lambda_{36} = 0; \\ \lambda_{12} = \lambda_{13} = \binom{0}{a_2, z_1} \vee \binom{2}{a_3, z_1}; \lambda_{15} = \binom{1}{a_2, z_2}; \\ \lambda_{24} = \lambda_{25} = \binom{1}{a_2, z_2} \vee \binom{3}{a_3, z_2}; \lambda_{26} = \binom{1}{a_1, z_2}; \lambda_{37} = \binom{2}{a_1, z_2}. \end{aligned}$$

Рекурсивная процедура первого уровня «соревнований» реализуется следующим образом. После полушага $\binom{0}{a_2, z_1}$ логические условия становятся равными $\lambda_{12} = 1$; $\lambda_{13} = 1$. При этом, в соответствии с (12), разрешается выполнение полушагов $\binom{z_1}{1, a_1}$ и $\binom{z_1}{1, a_2}$. После выполнения указанных полушагов запускаются процессы S_f и S_g , и вступают между собой в «соревнование», которое разворачивается в физическом времени. На первом шаге рекурсии

$${}^1\varphi_1(t) := f(t); {}^1\varphi_2(t) := g(t). \quad (13)$$

Взвешенные плотности распределения времени выполнения полушагов $\binom{1}{a_1, z_2}$ и $\binom{1}{a_2, z_2}$ первыми определяются по зависимостям:

$$\begin{cases} {}^1\eta_1(t) = {}^1\varphi_1(t) \cdot [1 - {}^1\Phi_2(t)] \\ {}^1\eta_2(t) = {}^1\varphi_2(t) \cdot [1 - {}^1\Phi_1(t)] \end{cases} \quad (14)$$

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

Вероятности и плотности распределения времени выполнения полушагов $\binom{1}{a_1, z_2}$ и $\binom{1}{a_2, z_2}$ первыми равны соответственно

$$\begin{aligned} {}^1\pi_1 = \int_0^\infty {}^1\eta_1(t) dt, \quad {}^1\pi_2 = \int_0^\infty {}^1\eta_2(t) dt. \\ {}^1\psi_1(t) = \frac{{}^1\eta_1(t)}{{}^1\pi_1}; \quad {}^1\psi_2(t) = \frac{{}^1\eta_2(t)}{{}^1\pi_2}. \end{aligned} \quad (15)$$

В том случае, если первым выполняется полушаг $\binom{1}{a_1, z_2}$, СПМ переключается в место 1a_3 , являющееся аналогом поглощающего состояния, и работа алгоритма завершается. В том случае, если первым выполняется полушаг $\binom{1}{a_2, z_2}$, СПМ переключается в места 2a_1 , 2a_2 , и по зависимости [11]

$${}^1\varphi_1(t) := \frac{{}^1(t) \int_0^\infty {}^1\varphi_2(\tau) {}^1\varphi_1(t+\tau) d\tau}{\int_0^\infty {}^1\Phi_2(t) d{}^1\Phi_1(t)}, \quad (16)$$

оценивается время, оставшееся до окончания пребывания в месте 1a_1 (${}^1(t)$ - единичная функция Хэвисайда). Время (16) подставляется в процесс S_f .

После выполнения в первый раз полушагов $(z_2, {}^2a_1)$ и $(z_2, {}^2a_2)$ в местах ${}^2a_1, {}^2a_2$ начинается «соревнование» процесса обработки прерывания и генератора прерываний, который запускается вновь (${}^2\varphi_2(t) := g(t)$). Поскольку «соревнование» запускается в первый раз, то ${}^2\varphi_1(t) := v(t)$. Взвешенные плотности распределения времени выполнения полушагов $({}^2a_1, z_3)$ и $({}^2a_2, z_3)$ первыми, вероятности и плотности распределения определяются по зависимостям:

$$\begin{cases} {}^2\eta_1(t) = {}^2\varphi_1(t) \cdot [1 - {}^2\Phi_2(t)] \\ {}^2\eta_2(t) = {}^2\varphi_2(t) \cdot [1 - {}^2\Phi_1(t)] \end{cases} \quad (17)$$

$${}^2\pi_1 = \int_0^\infty {}^2\eta_1(t) dt, \quad {}^2\pi_2 = \int_0^\infty {}^2\eta_2(t) dt, \quad {}^2\psi_1(t) = \frac{{}^2\eta_1(t)}{{}^2\pi_1}, \quad {}^2\psi_2(t) = \frac{{}^2\eta_2(t)}{{}^2\pi_2}.$$

Если в результате «соревнования» первым выполняется полушаг $({}^2a_1, z_3)$ (завершается обработка прерывания), то

а) выполняется подстановка (подготавливается продолжение работы генератора прерываний)

$${}^2\varphi_2(t) := \frac{{}^1(t) \int_0^\infty {}^2\varphi_1(\tau) {}^2\varphi_2(t+\tau) d\tau}{\int_0^\infty {}^2\Phi_1(t) d{}^2\Phi_2(t)}; \quad (18)$$

б) выполняются полушаги, сначала $(z_3, {}^2a_3)$, а затем $({}^2a_3, z_1)$;

в) выполняется пара полушагов $(z_1, {}^1a_1)$ и $(z_1, {}^1a_2)$ и запускается процесс «соревнования в местах ${}^1a_1, {}^1a_2$ с новыми значениями плотностей распределения ${}^1\varphi_1(t), {}^1\varphi_2(t)$.

Если в результате «соревнования» первым выполняется полушаг $({}^2a_2, z_3)$ (срабатывает генератор прерываний), то

а) выполняется подстановка

$${}^2\varphi_1(t) := \frac{{}^1(t) \int_0^\infty {}^2\varphi_2(\tau) {}^2\varphi_1(t+\tau) d\tau}{\int_0^\infty {}^2\Phi_2(t) d {}^2\Phi_1(t)}; \quad (19)$$

б) процесс переключается на следующий уровень прерываний.

k -й уровень «соревнований» представлен подсетью от места ${}^{k-1}a_2$ до мест ${}^ka_3, {}^{k+1}a_3$, т.е.

$$\Pi_k = \{A_k, Z_k, \Phi_k(t), A_k\}, \quad (20)$$

где $A_k = \{{}^{k-1}a_2, {}^{k1}a_1, {}^ka_2, {}^ka_3, {}^{k+1}a_1, {}^{k+1}a_2, {}^{k+1}a_3\}$ - подмножество мест; $Z_k = \{z_k, z_{k+1}, z_{k|2}\}$ - подмножество переходов; $\Phi_k(t) = [\varphi_{ij}(t)]$ - 7×3 матрица плотностей распределения, задающая временные интервалы «соревнований»; $A_k = [\lambda_{ji}]$ - 3×7 матрица логических условий выполнения полушагов из переходов;

$$\Phi_k(t) = \begin{bmatrix} {}^{k-1}\eta_2(t) & 0 & 0 \\ 0 & {}^k\varphi_1(t) & 0 \\ 0 & {}^k\varphi_2(t) & 0 \\ 0 & 0 & 0 \\ 0 & 0 & {}^{k+1}\varphi_1(t) \\ 0 & 0 & {}^{k+1}\varphi_2(t) \\ \delta(t) & 0 & 0 \end{bmatrix}; \quad (21)$$

$$A_k = [\lambda_{ij}]; \quad (22)$$

$$\begin{aligned} \lambda_{11} = \lambda_{14} = \lambda_{16} = \lambda_{17} = \lambda_{21} = \lambda_{22} = \lambda_{23} = \\ = \lambda_{27} = \lambda_{31} = \lambda_{32} = \lambda_{33} = \lambda_{34} = \lambda_{35} = \lambda_{36} = 0; \end{aligned}$$

$$\begin{aligned} \lambda_{12} = \lambda_{13} = \left({}^{k-1}a_2, z_k \right) \vee \left({}^{k+1}a_3, z_k \right); \lambda_{15} = \left({}^ka_2, z_{k+1} \right); \\ \lambda_{24} = \lambda_{25} = \left({}^ka_2, z_{k+1} \right) \vee \left({}^{k+2}a_3, z_{k+1} \right); \lambda_{26} = \left({}^ka_1, z_{k+1} \right); \lambda_{37} = \left({}^{k+1}a_1, z_{k+1} \right). \end{aligned}$$

Рекурсивная процедура на k -м уровне «соревнований» реализуется следующим образом. После полушага $\left({}^{k-1}a_2, z_k \right)$ логические условия становятся равными $\lambda_{12} = 1; \lambda_{13} = 1$. При этом, в соответствии с (22), разрешается выполнение полушагов $\left(z_k, {}^ka_1 \right)$ и $\left(z_k, {}^ka_2 \right)$. После выполнения указанных полушагов вступают в «соревнование» процессы S_f и S_g . На данном уровне «соревнований» величина ${}^k\varphi_1(t)$ представляет собой ре-

зультат предыдущих нисходящих (от первого уровня к более высоким) и восходящих (от более высоких уровней к первому) переключений. Если восходящее переключение на k -м уровне происходит первый раз, то

$${}^k\varphi_1(t) := f(t); {}^2\varphi_2(t) := g(t), \quad (23)$$

в противном случае ${}^k\varphi_1(t)$: представляет собой результат предыдущих переключений, а ${}^2\varphi_2(t) := g(t)$.

Взвешенные плотности распределения времени выполнения полушагов $\binom{k}{a_1, z_{k+1}}$ и $\binom{k}{a_2, z_{k+1}}$ первыми определяются по зависимостям:

$$\begin{cases} {}^k\eta_1(t) = {}^k\varphi_1(t) \cdot [1 - {}^k\Phi_2(t)]; \\ {}^k\eta_2(t) = {}^k\varphi_2(t) \cdot [1 - {}^k\Phi_1(t)]. \end{cases}, \quad (24)$$

Вероятности и плотности распределения времени выполнения полушагов $\binom{k}{a_1, z_{k+1}}$ и $\binom{k}{a_2, z_{k+1}}$ первыми равны, соответственно

$${}^k\pi_1 = \int_0^\infty {}^k\eta_1(t) dt, \quad {}^k\pi_2 = \int_0^\infty {}^k\eta_2(t) dt. \quad (25)$$

$${}^k\psi_1(t) = \frac{{}^k\eta_1(t)}{{}^k\pi_1}; \quad {}^k\psi_2(t) = \frac{{}^k\eta_2(t)}{{}^k\pi_2}.$$

В том случае, если первым выполняется полушаг $\binom{k}{a_1, z_{k+1}}$, СПМ повторно переключается в место ${}^k a_3$, с последующим выполнением полушага $\binom{k}{a_3, z_{k-1}}$ (переход z_{k-1} не входит в подсеть (20), поэтому в данном случае это переключение не описывается).

В том случае, если первым выполняется полушаг $\binom{k}{a_2, z_{k+1}}$, СПМ переключается в места ${}^{k+1}a_1, {}^{k+1}a_2$. При этом по зависимости

$${}^k\varphi_1(t) := \frac{1(t) \int_0^\infty {}^k\varphi_2(\tau) {}^k\varphi_1(t + \tau) d\tau}{\int_0^\infty {}^k\Phi_2(t) d{}^k\Phi_1(t)}, \quad (26)$$

оценивается время, оставшееся до окончания пребывания в месте ${}^k a_1$, и подставляется в процесс S_f .

.После выполнения полушагов $\binom{k+1}{z_{k+1}, a_1}$ и $\binom{k+1}{z_{k+1}, a_2}$ в местах ${}^{k+1}a_1, {}^{k+1}a_2$ начинается «соревнование» процесса обработки прерывания и генератора прерываний, который запускается вновь

$(^{k+1}\varphi_2(t) := g(t))$. При этом, процесс обработки прерываний, в том случае, если этот уровень проходил на предыдущих этапах, может продолжаться, поэтому

$$^k\varphi_1(t) := \begin{cases} v(t), \text{ when step } (z_k, ^ka_1) \text{ is exequter for the first time;} \\ ^k\varphi_1(t), \text{ dipending of pre – history in all other caces.} \end{cases} \quad (27)$$

Взвешенные плотности распределения времени выполнения полушагов $(^{k+1}a_1, z_{k+2})$ и $(\ddot{e}^{+1}a_2, z_{\ddot{e}+2})$ первыми, вероятности и плотности распределения определяются по зависимостям:

$$\begin{cases} ^{k+1}\eta_1(t) = ^{k+1}\varphi_1(t) \cdot [1 - ^{k+1}\Phi_2(t)] \\ ^{k+1}\eta_2(t) = ^{k+1}\varphi_2(t) \cdot [1 - ^{k+1}\Phi_1(t)] \end{cases} \quad (28)$$

$$^{k+1}\pi_1 = \int_0^\infty ^{k+1}\eta_1(t) dt, \quad ^{k+1}\pi_2 = \int_0^\infty ^{k+1}\eta_2(t) dt;$$

$$^{k+1}\psi_1(t) = \frac{^{k+1}\eta_1(t)}{^{k+1}\pi_1}; \quad ^{k+1}\psi_2(t) = \frac{^{k+1}\eta_2(t)}{^{k+1}\pi_2}.$$

Если в результате «соревнования» первым выполняется полушаг $(^{k+1}a_1, z_{k+2})$ (завершается обработка прерывания), то

а) выполняется подстановка (подготавливается продолжение работы генератора прерываний)

$$^{k+1}\varphi_2(t) := \frac{1(t) \int_0^\infty ^{k+1}\varphi_1(\tau) ^{k+1}\varphi_2(t + \tau) d\tau}{\int_0^\infty ^{k+1}\Phi_1(t) d^{k+1}\Phi_2(t)}; \quad (29)$$

б) выполняются полушаги, сначала $(z_{k+2}, ^{k+1}a_3)$, затем $(^{k+1}a_3, z_k)$;

в) выполняется пара полушагов $(z_k, ^ka_1)$ и $(z_k, ^ka_2)$ и запускается процесс «соревнования в местах $^ka_1, ^ka_2$ с новыми значениями плотностей распределения $^k\varphi_1(t), ^k\varphi_2(t)$.

Если в результате «соревнования» первым выполняется полушаг $(^{k+1}a_2, z_{k+2})$ (срабатывает генератор прерываний), то

а) выполняется подстановка

$${}^{k+1}\varphi_1(t) := \frac{{}_1(t) \int_0^{\infty} {}^{k+1}\varphi_2(\tau) {}^{k+1}\varphi_1(t+\tau) d\tau}{\int_0^{\infty} {}^{k+1}\Phi_2(t) d {}^{k+1}\Phi_1(t)}; \quad (30)$$

б) процесс переключается на следующий уровень прерываний.

Аналогичным образом рекурсивная процедура может быть продолжены для всех последующих уровней прерываний.

4. Модель прерываний как полумарковский процесс. Строго говоря, процесс, структура которого показана на рис. 3, а последовательность действий описана зависимостями (10) ÷ (30), обладает двумя свойствами:

он является бесконечным по двум параметрам, по количеству уровней прерывания и по количеству возвратов на один и тот же уровень;

процесс не является ни полумарковским, ни, тем более, Марковским за счет подстановок (16), (17), (18), (19), (26), (29), (30), которые зависят от предыстории.

Для анализа временных и вероятностных характеристик сеть Петри-Маркова, приведенная на рис. 3, должна быть преобразована в строго полумарковский процесс по следующей методике.

1. На СПМ выделяются подсети, моделирующие «соревнование», вида

$$\Pi_k^c = \left\{ \left\{ {}^{k-1}a_2, {}^ka_1, {}^ka_2 \right\}, \left\{ z_k, z_{k+1} \right\}, \Phi_k^c, A_k^c \right\}, \quad k = 1, 2, \dots, \quad (31)$$

где Φ_k^c, A_k^c , соответственно, подматрицы текущих плотностей распределения и логических условий выполнения полушагов.

2. Строится подграф, представляющий собой фрагмент бинарного дерева вида

$$\mu_k^c = \left\{ B_k^c, h_k^c(t) \right\} = \left\{ \left\{ \beta_{k-1,1(2)}^c, \beta_{k,1}^c, \beta_{k,2}^c \right\}, \begin{bmatrix} 0 & {}^k\eta_1(t) & {}^k\eta_2(t) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right\}, \quad k = 1, 2, \dots, \quad (32)$$

где $\left\{ \beta_{k-1,1}^c, \beta_{k,1}^c, \beta_{k,2}^c \right\}, \left\{ \beta_{k-1,12}^c, \beta_{k,1}^c, \beta_{k,2}^c \right\}$ - состояния полумарковского процесса, формируемые при «победе» в «соревновании» программы обработки прерываний (или основной программы) и генератора прерываний, соответственно; ${}^k\eta_1(t)$ и ${}^k\eta_2(t)$ - плотности распределения, рассчитываемые по зависимостям (24).

Вид бесконечного бинарного дерева показан на рис. 4.

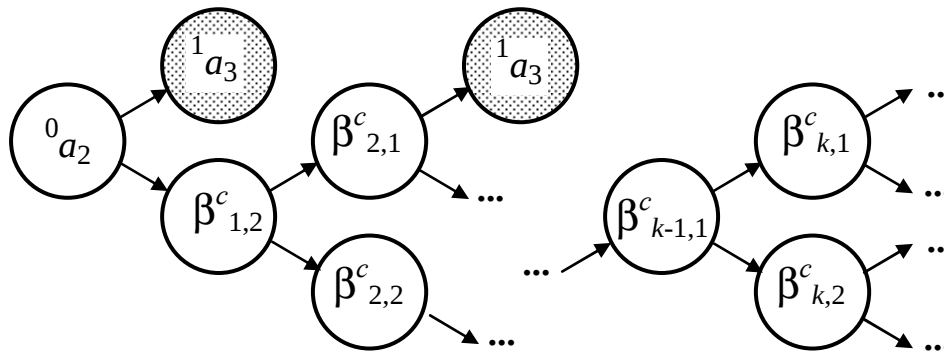


Рис. 4. Бинарное дерево полумарковского процесса

Состояние 0a_2 моделирует единственное стартовое состояние полумарковского процесса, а заштрихованные вершины графа моделируют бесконечное количество поглощающих состояний.

Вследствие того, что вся предыстория процесса была учтена при рекурсивном определении значений $^k\eta_1(t)$ и $^k\eta_2(t)$, процесс является строго полумарковским. Для определения временных интервалов, например, для определения времени достижения множества поглощающих состояний из стартового состояния, может быть использован любой известный метод.

5. Заключение. Полученная модель на базе фундаментального математического аппарата сетей Петри-Маркова описывает систему простейшего вида, когда после поступления сигнала прерывания ЭВМ Фон Неймановского типа полностью переключается на интерпретацию алгоритма обработки прерывания, а количество уровней прерывания является бесконечным. Очевидно, что в подобной системе возможен вариант, когда время интерпретации основного алгоритма увеличивается до бесконечности. Тем не менее, математическая модель и метод оценки временных характеристик, предложенный в работе, является продуктивным и позволяет в перспективе моделировать более сложные случаи. В частности, может быть исследован случай, когда интервал между прерываниями делится на окна, в каждом из которых производится часть обработки данных по основному алгоритму и алгоритмам обработки уже поступивших прерываний [13, 14]. Кроме того, с использованием предложенной модели могут быть решены оптимизационные задачи, с использованием в качестве критерия временной фактор.

Список литературы

1. Regehr J. Duongsaa U. Preventing interrupt overload // Proceedings of the Conference on Languages. Compilers, and Tools for Embedded Systems. 2005. P. 50 – 58.

2. Czerwinski M., Cutrell E., Horvitz E. Instant messaging and interruption: Influence of task type on performance // Proceedings of OZCHI. Sydney, Australia - 2000.
3. Bouchhima A., Yoo, S., Jeraya A. Fast and accurate timed execution of high level embedded software using HW/SW interface simulation model // Proceedings of the Conference on Asia South Pacific Design Automation, 2004. P. 469 – 474.
4. Cleaveland R., Smolka S. Strategic directions in concurrency research // ACM Comput. Surv. 1996. Vol. 28, № 4.
5. Heymann M. Concurrency and Discrete Event Control // IEEE Control Syst. Mag, 1990. Vol. 10. P. 103 – 112.
6. Valk R. Concurrency in Communicating Object Petri Nets // Concurr. object-oriented Program. Petri nets, 2001. P. 164 – 195.
7. Korolyuk V., Swishchuk A. Semi-Markov random evolutions. Springer-Science+Business Media, 1995. 309 p.
8. Larkin E.V., Ivutin A.N. Estimation of Latency in Embedded Real-Time Systems // 3-rd Mediterranean Conference on Embedded Computing (MECO-2014), Budva, Montenegro, 2014. P. 236 - 239.
9. Ivutin A., Larkin E., Lutskov Yu. Evaluation of Program Controlled Objects States // 4rd Mediterranean Conference on Embedded Computing (MECO). Budva, Montenegro, 2015. P. 250 - 253.
10. Larkin E.V., Lutskov Yu.I., Ivutin A.N., Novikov A.S. Simulation of concurrent process with Petri-Markov nets // Life Science Journal, 2014. N. 11 (11). P. 506 - 511.
11. 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. P. 43 - 54.
12. 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. P. 117 - 128.
13. Mark S. Squillante Stochastic Analysis and optimization of multi-server systems / Danilo Ardagna, Li Zhang (ed.) Run-Time Models for Self-managing Systems and Applications. Mathematic Subject Classification. Springer Based AG, 2010. P. 1 - 25.
14. Duda K.J., Cheriton D.R. Borrowed-virtual-time (BVT) scheduling: supporting latency-sensitive threads in a general-purpose scheduler // Proceedings of the ACM Symposium on Operating Systems Principles, 1999. P. 261–276.

Ларкин Евгений Васильевич, д-р техн. наук, проф., зав. кафедрой, elarkin@mail.ru, Россия, Тула, Тульский государственный университет

E.V. Larkin

Time characteristics of algorithm interpretation by Von-Neumann computers are investigated. With use of semi-Markov process fundamental apparatus the analytical model of program runtime evaluation, when absence of interruption is worked out. It is shown that external interruptions are the result of functioning of independent random process, which develops in parallel with algorithm interpretation. For description of interaction of two processes apparatus of Petri-Markov nets is used. Basic structural-parametric model of computer functioning in the presence of interruptions is worked out. It is shown that in common case Petri-Markov model is an infinite one. The recursive procedure of wandering through Petri-Markov net for case under investigation is worked out. It is shown that process of wandering through the net is not quite semi-Markov one. The method of transformation of Petri-Markov model onto strong semi-Markov process is proposed.

Key words: Von-Neumann computer, Interruption, runtime, semi-Markov process, Petri-Markov model, wandering.

Larkin Eugene Vasilyevich, doctor of technical science, professor, head of chair, elarkin@mail.ru, Russia, Tula, Tula State University

УДК 621.78

НАГРЕВ ПОВЕРХНОСТИ МИШЕНИ ПОДВИЖНЫМ ЛАЗЕРНЫМ ЛУЧОМ

Т.А. Акименко

Создана модель нагрева поверхности мишени с помощью подвижного лазерного луча, которая охлаждается потоком воздуха. Уравнение решено для случая динамического изменения энергии лазерного луча при постоянных параметрах воздушного потока.

Ключевые слова: нагрев, лазер, мишень, метод конечных элементов, метод конечных разностей, модель, конечный элемент.

Минешь представляет собой плоскую пластину, заданной толщины δ . Декартова система координат, в которой решается задача, неподвижна. Луч движется по поверхности пластины по заданному закону со скоростью v (рис.1).