**BC/NW 2017 № 1 (30)**

**ОСНОВЫ МЕТОДИКИ ПОСТРОЕНИЯ ИМИТАЦИОННЫХ GPSS-МОДЕЛЕЙ ПЕРЕСТРАИВАЕМЫХ СТРУКТУР ДИСКРЕТНЫХ ПРОЦЕССОВ И СИСТЕМ**

Дорошенко А.Н.

В работе [1] кратко рассмотрены принципы, механизмы и операционные средства системы GPSS WORLD, позволяющие строить и исследовать компактные представления моделей сложных регулярных и перестраиваемых структур и алгоритмов выполнения процессов в вычислительных устройствах, системах сбора, обработки и передачи данных и сетях телекоммуникаций. Проблема состоит в правильной (адекватной) семантической интерпретации средств языка GPSS и применении заложенных в систему GPSS механизмов событийного моделирования и имитации параллельных процессов моделируемых дискретных систем. В данной работе делается попытка конкретизировать эти возможности на ряде примеров использования языковых средств, механизмов идентификации и косвенной адресации объектов GPSS.

**Тема1: Идентификация параметров транзактов и элементов памяти**

**Задача 1.** На вход многопроцессорной системы (МС) поступает два потока заявок, имеющих пуассоновское распределение с интенсивностями Lp1 и Lp2 и требующих обработки на любом из процессоров. Времена обработки заявок tо1 и tо2 - случайные величины с экспоненциальным распределением и параметрами 1/То1 и 1/То2. Построить GPSS-модель работы МС, предусмотрев вычисление пропускной способности МС как количество обработанных заявок первого и второго потока за единицу времени.

*Примечания.*

1. *Схема рассматриваемой системы отличается от типовых СМО наличием двух потоков заявок с различными интенсивностями входа в систему и различными интенсивностями обслуживания, поэтому оценка численных значений характеристик системы возможна только имитационным моделированием.*
2. *Рассматриваемая модель относится к классу многоканальных СМО с неограниченной очередью и для обеспечения требования стационарности случайных процессов должно выполняться условие ненасыщенного режима работы моделируемой системы: (Lp1+Lp2)/[ (То1+То2)/(2\*То1\*То2)\*N ] < 1, где N – количество процессоров в МС. С учётом этого условия в GPSS-модели задаём, например, следующие значений исходных данных: Lp1=10 заявок в сек.; Lp2=15 заявок в сек ; To1= 18 сек; To2= 7сек; N >252, пусть N=256 процессоров.*

*\* объявление объектов*

MC equ 1

MC storage 256 ; количество процессоров в МС

OCH equ 1

Lp1 equ 10

Lp2 equ 15

То1 equ 18

То2 equ 7

Tsim equ 86400 ; время моделирования (имитации работы МС) – в секундах

Simulate

*\* генерация первого потока заявок*

Generate (exponential (1,0,1/L1)); *1/L1=0.1сек = Твх1*

Assign 1,1; *в первом параметре транзакта идентифицируется номер потока 1*

Transfer ,*proc*; *безусловное перемещение заявки в очередь к процессорам*

*\* генерация второго потока заявок*

Generate (exponential (2,0, 1/L2)); *1/L2= Твх2*

Assign 1,2; *в первом параметре транзакта идентифицируется номер потока 2*

*proc* Queue OCH; *занять очередь к процессорам*

Enter MC*; занять один из свободных процессоров*

Depart OCH; *выйти из очереди*

Test E p1,1,*potok2*; *проверка условия принадлежности заявки к потоку 1 или 2*

Advance (exponential (3,0,To1)); *время обработки t1 заявки потока 1*

Savevalue sch1+,1; *счёт числа обработанных заявок потока 1*

Transfer ,*vyhproc*; *безусловное перемещение заявки из процессора*

*potok2* Advance (exponential (3,0,To2)); *время обработки t2 заявки потока 2*

Savevalue sch2+,1; *счёт числа обработанных заявок потока 2*

*vyhproc* Leave MC; освободить процессор

Savevalue proizv1, sch1/Tsim; вычисление доли производительности блока

\*процессоров при обработке заявок первого потока

Savevalue proizv2, sch2/Tsim; вычисление доли производительности блока \*процессоров при обработке заявок второго потока

Savevalue proizv0, (sch1 + sch2 )/Tsim; вычисление общей производительности \*блока процессоров при обработке заявок первого и второго потока

Terminate; *удаление заявки из системы*

*\* сегмент задаёт время окончание процесса моделирования, равное 24 часам с \* точностью до 1 сек.*

Generate Tsim;

Terminate 1

Start 1

*Примечание: получаемые по результатам имитационного моделирования значения производительности МС следует скорректировать, разделив эти значения на среднее значение коэффициента занятости блока процессоров, формируемое системой GPSS в файле результатов.*

**Тема 2: Применение расширенного понятия функции – FUNCTION**

Функция в GPSS должна быть задана таблицей, имеющей следующую структуру:

Name FUNCTION A,B

x1,y1/x2,y2/…xn,yn

где А – аргумент функции – может быть **именем,** целым положительным числом, строкой, выражением в скобках, СЧА, **СЧА\*параметр;** В – тип функции, состоящий из одной буквы, определяющей собственно тип, и числа, задающего количество пар значений аргумента и функции{ хi,yi} – координат точек i=1..n на плоскости графика функции.

В GPSS существует пять типов функций: С – непрерывная, D – дискретная, L – списковая числовые функции и E и M – дискретная и списковая атрибутивные функции.

Для наших целей предпочтительными являются варианты применения функций типа D с аргументами **имя** и **СЧА\*параметр**. Рассмотрим варианты применения этих функций на примерах.

 **Пример1.** Пусть требуется входной поток заявок распределить случайным образом с заданными вероятностями 0.2, 015, 0.3, 0.25, 0.1на n=5 одноканальных устройств (OKU)некой системы. Положения устройств в модели системы не упорядочены, поэтому обращение к ним задается метками met1,…, met5. Тогда фрагмент модели будет иметь вид:

…

\*объявление функции Rpd; аргументом функции является встроенная функция RN, \*генерирующая числа в диапазоне (0,1), 29 – начальное – произвольное положительное \*целое число, задаваемое при написании программы

Rpd FUNCTION RN29,D5

.2,met1/.35,met2/.65,met3/.9,met4/1,met5

 Simulate

 Generate (Exponential(13,0,25))

 TRANSFER FN,Rpd; блок изменения маршрута транзакта Transfer здесь \*работает в режиме функции, которая реализует выборку меток в соответствии с \*заданными вероятностями

 Met1 seize OKU1…………….

 …

 Met2 seize OKU2……………

 …

 Met3 seize OKU3…………….

 …

 Met4 seize OKU4 …………….

 …

 Met5 seize OKU5…………….

**Пример2.** Пусть в одном и том же ОКУ требуется имитировать блоком ADVANCE интервал времени обслуживание заявок с различными средними значениями, задаваемыми функцией Tobsl. Тогда настройка на заданное время определяется этой функцией, аргументом которой назначается параметр Р1активного транзакта, поступающего в блок ADVANCE:

Rpd FUNCTION RN29,D5

.2,met1/.35,met2/.65,met3/.9,met4/1,met5

\*объявление функции времени обслуживания; аргумент функции – первый параметр \*транзакта

Tobsl FUNCTION P1,D5

1,4.5/2,2.7/3,2.5/4,0.7/5,3.3

 Simulate

 Generate (exponential(13,0,25))

 …

Advance (Exponential(17,0,FN$Tobsl)); среднее значение времени \*обслуживания определяется первым параметром транзакта, поступающего в данный \*блок

Идея косвенной адресации состоит в том, что каждый транзакт в некотором своем параметре содержит номер того или иного объекта, а в полях блоков соответствующих объектов записывается ссылка на этот параметр транзакта. При косвенной адресации СЧА определяются как СЧА\*параметр, где параметр - это номер или имя параметра активного транзакта, содержащего номер нужного блока. Например, Q\*2 – текущее значение длины очереди, номер которой является значением второго параметра активного транзакта, SR\*Npm – коэффициент использования памяти (storage), номер которой содержится в параметре с именем Npm активного транзакта.

**Пример 3.**

Seize P\*X1; Занять ОКУ, номер которого содержится в параметре, номер

\*которого определяется значением ячейки Х1.

Пример 2.

Var1 Variable 1+RN1/250; переменная принимает случайные целые из 1..4

 …

Assign 2,V$Var1; параметру 2 присваивается значение переменной Var1

Test L Q\*2,10,UDAL; проверяется условие, меньше ли числа 10 текущее

\*значение длины очереди Q, номер которой является значением второго параметра \*активного транзакта.

Пример 3.

Savevalue 1, X\*P3; поместить в ячейку с номером 1 значение, содержащееся в

\*ячейке Х, номер которой определяется значением параметра 3 активного транзакта, \*вошедшего в блок Savevalue.

**Тема3: Применение SELECT – блока выбора направления перемещения транзакта по условию состояния отдельного устройства, памяти, очереди или по условию соотношения состояний этих объектов.**

 Структура блока: **SELECT условие A,B,C,D,E,F**, где **условие** – это:

1) логическая переменная типа стандартный логический атрибут (СЛА):

* объекта типа Facility: U –занято, NU – не занято, I – прервано, NI – не прервано устройство, номер j которого находится последовательным сканированием номеров из диапазона (**В..С**) номеров устройств Facility,
* объекта типа Storage : SE – память пуста, SNE – память не пуста, SF – память заполнена, SNF – не заполнена память, номер j которого находится последовательным сканированием номеров из диапазона (**В..С**) номеров устройств Storage,

2)или логическое выражение типа отношения: E – равно, NE – не равно, G – больше, GE – больше или равно, L – меньше, LE – меньше или равно величине **D** значение некоторого стандартного числового атрибута (СЧА), указываемого в поле **Е** блока SELECT,

3) или MIN, MAX – операции поиска номера j из диапазона номеров (**B..С**) объектов с минимальным или максимальным значением СЧА (в поле **Е**).

В поле **А** блока SELECT записывается номер параметра транзакта, в который заносится номер объекта, удовлетворяющего заданному условию.

Таким образом**, если** при поступлении текущего активного транзакта в результате выполнения подпрограммы блока SELECT находится устройство, для которого **выполняется условие, то параметру А этого транзакта присваивается значение номера соответствующего объекта (устройства или памяти или очереди), а сам транзакт перемещается в следующий блок GPSS-модели.**

**Если условие не выполняется**, то текущий транзакт перемещается по метке в поле **F** блока SELECT, а при отсутствии метки транзакт переходит в следующий блок модели.

Рассмотрим варианты применения блока SELECT для имитации работы перестраиваемой структуры МС на конкретном примере.

**Пример 4.** Пусть вычислительная система представлена совокупностью m процессорных блоков, каждый из которых содержит некоторое число N1,N2,…,Nm процессорных модулей (процессоров). В процессе моделирования в текущий момент работы ВС требуется найти процессорный блок по одному из признаков:

1. процессорный блок с максимальным числом свободных процессоров;
2. процессорный блок с количеством свободных процессоров не менее 3;
3. процессорный блок, к которому минимальная длина очереди запросов на выполнение заданий.

Процессорный блок в модели представим многоканальным устройством (МКУ) (по терминологии GPSS это объект Storage) и в i-том блоке задаём Ni – количество параллельно работающих процессорных модулей. В модели нумерация процессорных блоков должна быть сквозная от 1 до m.

Предполагается, что к каждому из процессорных блоков может быть организована неограниченная очередь запросов.

Решение варианта 1 (пусть m=16):

 **…**

**SELECT max 4,1,16,,R;** поле F в этом режиме не требуется

\*комментарий: в четвёртый параметр транзакта заносится номер процессорного блока

\* (из диапазона 1..16), в котором **наибольшее** количество свободных процессорных \*модулей, где R – СЧА – количество свободных элементов памяти (storage) и каждый из \*элементов имитирует процессорный модуль выбранного процессорного блока.

**…**

Решение варианта 2:

 **…**

 **SELECT GE 4,1,16,3,R,?**

\*комментарий: в четвёртый параметр транзакта заносится номер первого процессорного \*блока (из диапазона 1..16), в котором **не менее трёх** свободных процессорных модулей, \*если же такой процессорный блок не найден, то если поле F=? пустое, то транзакт \*переходит в блок модели, следующий за блоком SELECT, иначе по метке F.

**…**

Решение варианта 3:

**…**

**SELECT min 2,1,16,,Q;** поле F в этом режиме не требуется

\*комментарий: во второй параметр транзакта заносится номер той очереди, в которой в \*данный момент находится наименьшее количество запросов на выполнение задания, где \*Q – СЧА – текущая длина очереди (queue).

**…**

Рассмотрим применение блока SELECT на примере следующей задачи.

**Задача 2.** На вход вычислительной сети (ВС), содержащей m независимых и равнозначных по характеристикам узлов (одноканальных устройств ОКУ) обслуживания заявок, поступает пуассоновский поток заявок с интенсивностью Lam. Заявки по некоторому признаку делятся на три группы и поступают в сеть с вероятностями 0.25, 0.6 и 0.15. Любая заявка поступает на любой свободный узел ОКУi, i=1..m на обслуживание, а если нет свободного, то к тому узлу, к которому очередь заявок, ожидающих обслуживание, меньше. Время обслуживания – экспоненциальное с заданным средним Tobs. Очереди к узлам не ограничены . После завершения обслуживания заявки первой группы удаляются, а заявки второй и третьей группы требуют обращения к серверу, время обслуживания на котором имеет нормальное распределение с заданными средним Ts временем и стандартным отклонением Dn. Разработать GPSS-модель функционирования ВС, предусмотрев, кроме типовой статистики, предусмотренной в системе GPSS, вычисление количества заявок, обслуженных каждым узлом ВС за сутки.

*Примечание: Для запуска программы следует задать числовые значения исходных данных. По умолчанию предполагается сквозная нумерация объектов каждого типа, например, всем ОКУ автоматически системой моделирования присваиваются номера 1..m, очереди к этим устройствам также имеют номера 1..m. Число m должно быть задано – в данной программе – в блоках SELECT.*

Grup FUNCTION Rn555, D3

0.25,1/0.85,2/1,3

 Simulate

Generate (Exponential(1,0,1/Lam);

Assign 1,FN$Grup;

SELECT NU 2,1,m,,,*vibor\_och;выбор одного из 1..m свободных устройств (ОКУ)*

*\* и запись его номера в параметр 2 транзакта, поступившего в данный блок; при*

*\*отсутствии свободного ОКУ транзакт передается по метке на блок выбора очереди*

Transfer ,*Obsl;передача транзакта на обслуживание*

*vibor\_och* SELECT min 2,1,m,,Q; *выбор очереди с минимальной длиной*

*Obsl* Queue P2;транзакт поступает в очередь с номером, содержащемся в параметре 2 \*транзакта

Seize P2; транзакт пытается войти в ОКУ с номером, содержащемся в параметре 2

Depart P2; транзакт освобождает очередь с номером, содержащемся в параметре 2 \*транзакта

Advance (exponential(2,0,Tobs)); имитируется время обслуживания

Release P2; освобождается ОКУ с номером, содержащемся в параметре 2

Savevalue P2+,1;подсчитывается количество заявок, обслуженных каждым узлом ВС

Test NE P1,1,*Vyhod;*отбираются заявки второго и третьего типа для обработки в \*сервере: ниже – постановка в очередь к серверу и обработка

Queue Och\_Serv;

Seize Server;

Depart Och\_Serv;

Advance (Normal (3,Ts,Dn));

Relese Server;

Terminate; удаление из модели заявок второго и третьего типа

*Vyhod*  Terminate; удаление из модели заявок первого типа

Generate 24\*60\*60\*1000; время моделирования задано с точностью до ms

Terminate 1;

Start 1

*Примечание. Применение косвенной адресации в этой модели позволяет имитировать одной парой блоков Queue и Depart не одну, а* ***m*** *очередей, в блоках Seize и Release –* ***m*** *устройств ОКУ (узлов ВС), в блоке Savevalue –* ***m*** *переменных – сохраняемых величин (по терминологии GPSS), в которых подсчитывается количество заявок, обработанных каждым из* ***m*** *узлов.*

**Литература**

1. Дорошенко А.Н. Моделирование перестраиваемых структур многопроцессорных систем средствами GPSS WORLD. Электронный журнал «Вычислительные сети. Теория и практика». BC/NW, 2016, № 1 (28):9.1. – 7 с.
2. Дорошенко А.Н., Федоров В.Н. Моделирование дискретных систем. Метод. пособие. – М.: Изд-во МЭИ, 2001. – 44 с.
3. Рыжиков Ю.И. Имитационное моделирование. Теории и технологии. – СПб.: КОРОНА принт; М.: Альтекс-А, 2004, 384 с.
4. Боев В.Д. Моделирование систем. Инструментальные средства GPSS WORLD: Учеб. Пособие. – СПб.: БХВ-Петербур, 2004. – 368 с.
5. Варжапетян А.Г. Имитационное моделирование на GPSS/H. – М.: Вузовская книга, 2007. – 424 с.
6. Кудрявцев, Е.М. GPSS Word : / Е.М.Кудрявцев. — Москва: ДМК Пресс, 2008. — 317 с
7. Сосновиков Г.К. , Воробейчиков Л.А. Компьютерное моделирование. Практикум по имитационному моделированию в среде GPSS World: Учебное пособие / Г. К. Сосновиков, Л. А. Воробейчиков. — Москва: Инфра-М Форум, 2015. — 112 с