

**СЕВЕРО-КАВКАЗСКИЙ ФИЛИАЛ  
ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ  
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО  
ОБРАЗОВАТЕЛЬНОГО  
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
«Московский технический университет связи и информатики»  
Кафедра «Инфокоммуникационные технологии и системы связи»**

**НЕРСЕСЯНЦ А.А.**

**Методические указания  
к лабораторной работе по теме:**

**Изучение основных приёмов имитационного  
моделирования в системе  
GPSS World**

**Направление подготовки 11.03.02  
«Инфокоммуникационные технологии и системы связи»,  
профиль «Сети связи и системы коммутации»**

**Дисциплины: Сети связи, Мультисервисные сети связи,  
Моделирование**

**Ростов-на-Дону  
2019**

**План УМР на 2019/2020 учебный год**

**Методические указания к лабораторной работе по теме:**

**Изучение основных приёмов имитационного моделирования  
в системе GPSS World**

**Дисциплины: Сети связи, Мультисервисные сети связи.  
Моделирование**

**Направление подготовки 11.03.02  
Инфокоммуникационные технологии и системы связи  
профиль «Сети связи и системы коммутации»**

**Направление подготовки 09.03.01  
Информатика и вычислительная техника  
профиль «Вычислительные машины, комплексы, системы и сети»**

**Автор:  
профессор кафедры ИТСС,  
д.т.н., с.н.с. Нерсисянц А.А.,**

**Рецензент: доцент кафедры ИТСС - к.т.н. Борисов Б.П.**

**Рассмотрено и одобрено на заседании кафедры ИТСС  
Протокол №11 от 26.08.19**

**Целью настоящих практических занятий является изучение на простейших примерах основных приёмов моделирования элементов телекоммуникационных сетей.**

В отчёте по практическим занятиям необходимо по каждому пункту представить программы с комментариями и основные результаты прогонов.

Изучить функции использованных операторов и операндов, а также содержание стандартных отчётов.

## **Практическое занятие № 1.**

### **Модель одноканального устройства с неограниченной очередью.**

Вызвать систему GPSS World щелчком на соответствующем ярлыке на рабочем столе.

Открыть Модуль 1 (рис. 1).

Ввести программу рис. 1 (без комментариев):

```
generate 20 ;генерация транзактов (пакетов) с интервалом 20 е.м.в.
queue fff ;установить транзакт в очередь с именем fff
seize kkk ;занять канал с именем kkk
depart fff ;уйти из очереди fff
advance 19 ;задержать транзакт в канале на 19 е.м.в.
release kkk ;освободить канал kkk
terminate 1 ;вычесть 1 из длины прогона
```

Рис.1. Модуль 1. Модель одноканального устройства с неограниченной очередью.

Е.м.в. – единица модельного времени.

Тщательно изучить каждый оператор и приведенные операнды по комментариям.

Произвести запуск программы (command > start) на длину прогона в 10000 транзактов.

Изучить содержимое стандартного отчета (REPORT), особенно в части характеристик обслуживающего устройства (канала) и очереди.

Обратите внимание на то, что данная модель статистически устойчива только, если длительность задержки транзакта (операнд А оператора advance) меньше интервала между моментами появления транзактов (операнд А оператора generate).

Убедитесь в этом, изменив соотношение операндов на противоположное и проведя два эксперимента с различной длиной прогона. Объясните причину зависимости длины очереди (AVE. CONT) от длины прогона.

Внесите случайность в работу модели:

generate 20,20 - длительность интервала между моментами появления транзактов будет меняться от 0 до 40 .

advance 19,10 - длительность задержки транзакта в канале будет меняться от 9 до 29.

## Практическое занятие № 2.

### Модель одноканальной системы с ограниченной очередью.

Не меняя соотношения операндов А, введите ограничение на длину очереди с помощью оператора test (добавить в модуль 1):

```
test L  Q$fff,10,met      ;очередь меньше десяти?
```

Данный оператор проверяет на «меньшее» текущую длину очереди (стандартный числовой атрибут Q) с именем fff с числом 10 и, если оно не меньше (вспомогательный операнд L), то направляет транзакт к метке met.

Таким образом, обслуживание транзакта (пакета) продолжается только в том случае, если в очереди есть свободные места. В противном случае транзакт прибавляет единицу к переменной otk и покидает систему.

Общий вид программы:

```
generate 20,20      ;генерация транзактов (пакетов) с интервалом 20 е.м.в.
test L    Q$fff,10,met ;есть свободные места в очереди?
queue     fff        ;Да. Установить транзакт в очередь с именем fff
seize     kkk        ;занять канал с именем kkk
depart    fff        ;уйти из очереди fff
advance   19,10      ;задержать транзакт в канале на 19 е.м.в. (в среднем)
release   kkk        ;освободить канал kkk
terminate 1          ;вычесть 1 из длины прогона
met savevalue otk+,1 ;зафиксировать отказ в переменной otk
terminate 1
```

Рис.2. Модуль 2. Модель одноканальной системы с ограниченной очередью.

Запустите программу и проанализируйте отчет.

Снимите зависимости:

-  $T_{ож}(q)$  для 4-х значений  $q$  в пределах от 0,5 до 1,5. Здесь:  $T_{ож}$  – среднее время ожидания в очереди (AVE.TIME в REPORT), а  $q = t_{зад}/t_{инт}$  – загрузка канала, определяемая как отношение времени задержки транзакта в операторе advance к интервалу между моментами появления транзактов в операторе generate.

-  $P(q)$  – вероятность потери транзакта;

## Практическое занятие № 3.

### Построение гистограммы распределения времени ожидания в очереди.

Дополнить программу Модуль 2 (рис. 2) оператором:

```
wtime qtable fff,0,50,40
```

поставив его перед оператором generate. Операнд А (fff) – имя очереди, операнд В (0) – начало гистограммы по оси абсцисс, операнд С (50) – длина интервала, операнд D (40) – число интервалов. Подобрать операнды В, С, и D для наглядного представления гистограммы на экране монитора.

```
wtime qtable fff,0,50,40,
generate 20,20 ;генерация транзактов (пакетов) с интервалом 20 е.м.в.
test L Q$fff,10,met ;есть свободные места в очереди?
queue fff ;Да. Установить транзакт в очередь с именем fff
seize kkk ;занять канал с именем kkk
depart fff ;уйти из очереди fff
advance 19,10 ;задержать транзакт в канале на 19 е.м.в.
release kkk ;освободить канал kkk
terminate 1 ;вычесть 1 из длины прогона
met savevalue otk+,1 ;зафиксировать отказ в переменной otk
terminate 1
```

Рис. 3. Модуль 3.

Зарисовать гистограмму в отчёте.

## Практическое занятие № 4.

### Модель многоканальной системы с потерями (без ожидания)

В модели многоканального устройства (МКУ) вместо операторов seize и release используются операторы enter и leave. При этом до оператора generate должен быть установлен оператор storage с именем МКУ, в котором указывается число каналов (в данном случае определено, что в МКУ ggg имеется 3 канала).

Оператор gate snf передаст транзакт на обслуживание только при незаполненности МКУ (дополнительный операнд snf). В противном случае транзакт будет учтён как потерянный в переменной ot.

Открыть программу Модуль 4.

```
ggg storage 3
generate 30,20
gate snf ggg,met ;многоканальное устройство не заполнено?
enter ggg ;Да. Занять канал
advance 100,100
leave ggg
terminate 1
met savevalue ot+,1
terminate 1
```

Рис. 4. Модуль 4. Модель многоканальной системы с потерями (без ожидания)

Снимите зависимость вероятности потерь  $P(q)$ , аналогично занятию 2, но в данном случае  $q = t_{\text{зад}} / kt_{\text{инт}}$ , где  $k$  – число каналов, задаваемое оператором storage.

## Практическое занятие № 5.

### Модель разделения потока транзактов на 3 разноприоритетных потока.

Открыть программу Модуль 5.

sss	function	rn1,d3	;аргумент – случайное число (rn1)
	.5,1/.8,2/1,3		;функция – числа 1, 2 или 3
	generate	30,20	
	priority	fn\$sss	;присвоение приоритетов по функции sss
	terminate	1	

Рис. 5. Модуль 5. Модель разделения потока транзактов на 3 разноприоритетных потока.

Операторы priority и function с именем sss разделят общий поток транзактов случайным образом (датчик случайных чисел rn1) на 3 потока. При этом 50% всех транзактов получит приоритет №1, 30% - №2 и 20% - №3. Третий приоритет – старший.

Объяснить структуру оператора priority.

## Практическое занятие № 6.

### Модель двухфазной системы с потерями.

Каждый транзакт должен обслуживаться вначале каналом с именем ггг, а затем каналом с именем ккк. Если к моменту поступления транзакта к любому из каналов данный канал занят (проверка оператором gate nu), то транзакт фиксируется как потерянный в переменной ot.

	generate	20,20	;генерация транзактов (пакетов) с интервалом 20 е.м.в.
	gate nu	ггг,otk	;устройство ггг свободно?
	seize	ггг	;Да. Занять канал с именем ггг
	advance	29,20	;задержать транзакт в канале на 29 е.м.в.
	release	ггг	;освободить канал с именем ггг
vtor	gate nu	ккк,otk	;Вторая фаза. Устройство ккк свободно?
	seize	ккк	;Да. Занять канал с именем ккк
	advance	19,19	;задержать транзакт в канале на 19 е.м.в.
	release	ккк	;освободить канал ккк
	terminate	1	;вычесть 1 из длины прогона
otk	savevalue	ot+,1	;Зафиксировать потерю транзакта
	terminate	1	;вычесть 1 из длины прогона

Рис. 6. Модуль 6. Модель двухфазной системы с потерями.

Откройте программу Модуль 6 и запустите её. Определите степень загрузки каждого канала и долю потерянных транзактов.

Измените программу так, чтобы имелась возможность определять число потерянных транзактов в каждом канале.

### **Практическое занятие № 7.**

**Добавление 2-го потока, входящего сразу во 2-ю фазу (во второй канал).**

Допишите к программе Модуль 6 (в начале) операторы генерации 2-го потока и безусловного перехода во 2-ю фазу обслуживания.

```

generate 25,20 ;генерация второго потока
transfer ,vtor ;безусловный переход
generate 20,20 ;генерация транзактов (пакетов)
gate nu rrr,otk ;канал rrr свободен?
seize rrr ;занять канал с именем kkk
advance 29,20 ;задержать транзакт в канале
release rrr ;освободить канал kkk
vtor gate nu kkk,otk ;канал kkk свободен?
seize kkk ;занять канал с именем kkk
advance 19,15 ;задержать транзакт в канале
release kkk ;освободить канал kkk
terminate 1 ;вычесть 1 из длины прогона
otk savevalue ot+,1 ;зафиксировать потерю транзакта
terminate 1 ;вычесть 1 из длины прогона

```

Рис. 7. Модуль 7. Добавление 2-го потока, входящего сразу во 2-ю фазу

Произведите прогон модуля 7 и в отчёте (Report) в самом начале (массив с названием LABEL) по данным колонки ENTRY COUNT проследите за движениями транзактов по блокам программы. Установите и объясните местонахождение всех транзактов, выданных двумя операторами generate. Определите в какой фазе обслуживания вероятность потери транзактов выше и измените входные параметры модели (факторы) так, чтобы соотношение откликов изменилось на обратное.

### **Практическое занятие № 8.**

**Модель системы с 3-я разноприоритетными очередями транзактов.**

Откройте программу Модуль 8.

```

sss function rn1,d3 ;функция случайного распределения приорите
.5,1/.8,2/1,5      тов с вер. 0.5, 0.3 и 0.2
och function pr,d3 ;функция распределения длин очередей по при

```

1,30/2,15/5,7		оритетам
generate	20,20	;генерация транзактов с интервалом 20 е.м.в.
priority	fn\$sss	;выбор номера приоритета
assign	ocher,pr	;присвоение имени очереди номера приоритета
test L	Q*ocher,fn\$och,met	;есть свободные места в очереди?
queue	pr	;Да. Установить транзакт в очередь с именем 1,2 или 5
seize	kkk	;занять канал с именем kkk
depart	pr	;уйти из очереди
advance	19,10	;задержать транзакт в канале на 19 е.м.в.
release	kkk	;освободить канал kkk
terminate	1	;вычесть 1 из длины прогона
met savevalue	otk+,1	;зафиксировать отказ в переменной otk
terminate	1	

Рис. 8. Модуль 8. Модель системы с 3-я очередями транзактов.

Особенностью данной программы является использование в операторах queue и depart в качестве имени очереди номера приоритета (pr). Это дает возможность использовать одни и те же операторы для транзактов любого приоритета. В данном случае именами очереди будут цифры 1, 2 или 5.

Для этого используется оператор assign, который может вошедшему в него транзакту определить имя параметра и присвоить этому параметру заданное значение. В данном случае параметру с именем ocher присваивается числовое значение по номеру приоритета (1, 2 или 5). В дальнейшем этот параметр будет использоваться как имя очереди в операторе test L.

Оператор test L сравнивает на “меньше” текущую длину очереди под номером приоритета (Q\*ocher), хранящегося в параметре транзакта ocher, с числом мест в приоритетных очередях, определяемых функцией с именем och.

Произведите прогон Модуля 8 и проанализируйте отчёт в части очередей с именами 1,2 и 5. Измените длины очередей в функции с именем och, произведите прогон и объясните изменения в отчёте.

Измените программу Модуль 8 так, чтобы она имитировала многоканальную систему. Снимите зависимости средней длительности ожидания  $T_{ожі}$  (значение AVE.TIME в REPORT) и средней длины очереди  $D_{очі}$  (значение AVE.CONT в REPORT) от числа каналов по каждой категории. Если значения откликов  $T_{ожі}$  и  $D_{очі}$  исчезающе малы, измените параметры системы так, чтобы исследуемые зависимости оказались наиболее наглядными.

## Практическое занятие № 9.

**Реализация экспоненциального распределения длительности обслуживания и пуассоновского входящего потока.**

Откройте программу Модуль 9 (рис. 9). В операторе generate транзакты генерируются через экспоненциально распределённые интервалы со средним значением, равным 2 е.м.в. В операторе advance транзакты задерживаются в среднем на 3 е.м.в.



```

tab table    m1,0,0.2,100
eks fvariable -log((1+rn8)/1000)
    generate (2#v$eks)
    advance  (3#v$eks)
    tabulate tab
    terminate 1

```

Рис. 9. Реализация экспоненциального распределения длительности обслуживания и пуассоновского входящего потока.

Операнд `m1` в операторе `table` является СЧА, определяющим время жизни транзакта. Операнды `B`, `C` и `D` определяют формат гистограммы по оси абсцисс.

В операторе `fvariable` датчик случайных чисел `rn8` при каждом обращении выбирает случайное целое число в диапазоне от 0 до 999. Делением на 1000 эти числа переносятся в диапазон от 0 до 0.999. Чтобы избежать  $\log 0$  к `rn8` прибавляется 1 и окончательно диапазон чисел для логарифмирования устанавливается от 0.001 до 1.

## Практическое занятие № 10.

### Изучение стандартных числовых атрибутов.

Ознакомиться на конкретных примерах с основными элементами стандартного отчета и способами использования стандартных числовых атрибутов (СЧА).

Рассмотреть модель многоканальной системы с ограниченной очередью.

<code>vpot</code>	<code>fvariable</code>	<code>N\$met/N\$fin</code>
<code>kan</code>	<code>storage</code>	<code>10</code>
<code>eks</code>	<code>fvariable</code>	<code>-log{(1+rn1)/1000}</code>
	<code>generate</code>	<code>10,v\$eks</code>
	<code>test l</code>	<code>Q*och,7,met</code>
	<code>queue</code>	<code>och</code>
	<code>enter</code>	<code>kan</code>
	<code>depart</code>	<code>och</code>
	<code>advance</code>	<code>11,v\$eks</code>
	<code>leave</code>	<code>kan</code>
	<code>transfer</code>	<code>,fin</code>
<code>met</code>	<code>savevalue</code>	<code>otk+1</code>
<code>fin</code>	<code>test e</code>	<code>tg1,1 final</code>
	<code>savevalue</code>	<code>verp,v\$vpot</code>

final terminate 1

Рис. 10. Изучение стандартных числовых атрибутов.

Произвести запуск модели и выписать в тетрадь содержимое стандартного отчёта аналогично приведённым ниже фрагментам.

Объясните зафиксированное в отчёте число транзактов, поступивших в каждый блок (ENTRY COUNT), и число транзактов, присутствующих в некоторых блоках на момент окончания прогона (CURRENT).

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT
	1	GENERATE	100010	0
	2	TEST	100010	0
	3	QUEUE	85887	5
	4	ENTER	85882	1
	5	DEPART	85881	0
	6	ADVANCE	85881	4
	7	LEAVE	85877	0
	8	TRANSFER	85877	0
MET	9	SAVEVALUE	14123	0
FIN	10	TEST	100000	0
	11	SAVEVALUE	1	0
FINAL	12	TERMINATE	100000	0

Объясните содержание отчёта в части очереди с именем OCH:

QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE.CONT.	AVE.TIME
ОСН	7	6	85887	16103	3.357	7.821

Объясните содержание отчёта в части многоканального устройства с именем KAN:

STORAGE	CAP.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.
KAN	5	0	5	85882	1	4.696	0.939

Объясните содержание отчёта в части переменных с именами ОТК и VTRP:

SAVEVALUE	RETRY	VALUE
ОТК	0	14123.000
VERP	0	0.141

В данном программном модуле использованы следующие стандартные числовые атрибуты:

- СЧА - Q\*och – означает число транзактов, стоящих в очереди с именем och.
- СЧА - N\$met – счетчик входов в оператор с именем met. Переменная vrot подсчитывает вероятность потерь, как отношение числа транзактов, прошедших через оператор с меткой met (СЧА - N\$met) к общему числу транзактов (СЧА - N\$fin).

- с) СЧА –  $tg1$  – содержимое счётчика завершения. В данной программе подсчёт переменной  $vrot$  производится не при всех транзактах, а в целях экономии времени, только при проходе последнего транзакта, т.е. только тогда, когда  $tg1=1$ .
- д) СЧА -  $rn1 \dots rn8$  – случайные числа в интервале  $0 \dots 0,999999$ , используемые как аргументы функций, например, для определения номера узла отправителя (см. пр. занятие № 5). Во всех остальных случаях  $rn1 \dots rn8$  – случайные числа в интервале  $0 \dots 999$ . В данной программе  $rn1$  используется для определения переменной с плавающей запятой, распределенной по экспоненциальному закону. Такое распределение дает функция  $-\log(rn1/1000)$ . В программе используется функция  $-\log\{(1+rn1)/1000\}$ .
- е) СЧА -  $m1$  – время жизни транзакта. Удобно использовать для формирования таблицы распределения времени обслуживания.
- Полный список СЧА приведён в Приложении 2.

### Практическое занятие № 11.

Составить произвольную композицию с использованием наибольшего числа усвоенных программных элементов.

## Приложение 1.

### Краткие сведения из теории

В математических моделях (ММ) сложных объектов, представленных в виде систем массового обслуживания (СМО), фигурируют средства обслуживания, называемые *обслуживающими аппаратами* {ОА), и обслуживаемые заявки, называемые **транзактами**. Так, в модели производственной линии ОА отображают рабочие места, а транзакты - поступающие на обработку детали, материалы, инструмент.

Состояние СМО характеризуется состояниями ОА, транзактов и очередей к ОА.

Состояние ОА описывается двоичной переменной, которая может принимать значения "занят" или "свободен". Переменная, характеризующая состояние транзакта, может иметь значения "обслуживания" или "ожидания". Состояние очереди характеризуется количеством находящихся в ней транзактов.

**Имитационная модель** СМО представляет собой алгоритм, отражающий поведение СМО, т.е. отражающий изменения состояния СМО во времени при заданных потоках заявок, поступающих на входы системы. Параметры входных потоков заявок - внешние параметры СМО. Выходными параметрами являются

величины, характеризующие свойства системы - качество ее функционирования.

Примеры выходных параметров:

- *производительность* СМО - среднее число заявок, обслуживаемых в единицу времени;
- *коэффициенты загрузки оборудования* - отношение времени обслуживания к общему времени в каждом ОА;
- *среднее время обслуживания одной заявки*.

Основное свойство ОА, учитываемое в модели СМО, - это затраты времени на обслуживание, поэтому внутренними параметрами в модели СМО являются величины, характеризующие это свойство ОА. Обычно время обслуживания рассматривается как случайная величина и в качестве внутренних параметров фигурируют параметры законов распределения этой величины.

Имитационное моделирование позволяет исследовать СМО при различных типах входных потоков и интенсивностях поступления заявок на входы, при вариациях параметров ОА, при различных дисциплинах обслуживания заявок. **Дисциплина обслуживания** - правило, по которому заявки поступают из очередей на обслуживание. Величина, характеризующая право на первоочередное обслуживание, называется **приоритетом**. В моделях СМО заявки, приходящие на вход занятого ОА, образуют очереди, отдельные для заявок каждого приоритета.

При освобождении ОА на обслуживание принимается заявка из непустой очереди с наиболее высоким приоритетом.

Основной тип ОА – устройства. Именно в них происходит обработка транзактов с затратами времени. К ОА относятся также **накопители (памяти)**, отображающие средства хранения обрабатываемых деталей в производственных линиях или обрабатываемых данных в вычислительных системах. Накопители характеризуются не временами обслуживания заявок, а емкостью – максимально возможным количеством одновременно находящихся в накопителе заявок.

К элементам имитационных моделей СМО кроме ОА относят также **узлы и источники заявок**. Связи ОА между собой реализуют узлы, т.е. характеризуют правила, по которым заявки направляются к тому или иному ОА. Для описания моделей СМО при их исследовании на ЭВМ разработаны специальные языки имитационного моделирования. Существуют **общецелевые языки**, ориентированные на описание широкого класса СМО в предметных областях, и **специализированные языки**, предназначенные для анализа систем определенного типа. Примером общецелевых языков служит широко распространенный язык GPSS, примером специализированного языка – язык GPSS.

## Основные правила и операторы языка GPSS.

Для описания имитационной модели на языке GPSS полезно представить ее в виде схемы, на которой отображаются элементы СМО - устройства, накопители, узлы и источники. Описание на языке GPSS есть совокупность операторов (блоков), характеризующих процессы обработки заявок. Имеются операторы и для отображения возникновения заявок, задержки их в ОА, занятия памяти, выхода из СМО, изменения параметров заявок (например, приоритетов), вывода на печать

накопленной информации, характеризующей загрузку устройств, заполненность очередей и т.п.

Каждый транзакт, присутствующий в модели, может иметь до 12 параметров. Существуют операторы, с помощью которых можно изменять значения любых параметров транзактов, и операторы, характер исполнения которых зависит от значений того или иного параметра обслуживаемого транзакта.

*Пути продвижения заявок между ОА отображаются последовательностью операторов в описании модели на языке GPSS специальными операторами передачи управления (перехода). Для моделирования используется событийный метод. Соблюдение правильной временной последовательности имитации событий в СМО обеспечивается интерпретатором GPSS – программной системой, реализующий алгоритмы имитационного моделирования.*

## Структура операторов CP58

В записи оператора выделяют три части: метку, название, поле переменных.

Пример оператора:

**L1       GENERATE   30,5                    ; Первый сегмент модели**  
**2....6     8.....18 ..19.....70**  
 <метка><название> <поле переменных> <комментарии>

В поле переменных выделяют **подполя**, разделяемые при записи запятыми и служащие для указания чисел, стандартных числовых атрибутов (СЧА), символов, обозначающих метки, идентификаторов, указателей разновидностей операторов и т.п. Подполя могут быть пустыми. Возможна запись **комментария** после последнего непустого поля через пробел.

Стандартные числовые атрибуты служат для сокращенного указания различных величин, фигурирующих в модели.

Примеры СЧА: K126 - константа, равная 126; V2 - переменная N 2; (Q4 - длина очереди N 4; X5 - хранимая величина N 5; FN7 - функция N 7; P4 – значение параметра N 4 транзакта; \*6 - содержимое параметра N 6 транзакта; S\*3 (или FN\*3 - память (или функция), определенная в параметре N 3 транзакта.

## Основные операторы языка GPSS

Основные операторы языка GPSS приведены в приложении 3 и описаны ниже в виде примеров с конкретными значениями подполей в поле переменных.

**GENERATE 12,4,50, 5,1** - генерация транзактов, интервалы времени между появлениями транзактов распределены равномерно в диапазоне [ 12-4,12+4], первый транзакт появится с задержкой в 50 единиц модельного времени, всего будет создано 5 транзактов, приоритет транзактов равен единице.

**GENERATE 12,4,50,,1** - то же, но количество генерируемых транзактов не ограничено.

**GENERATE 6, FN\$FFF,50,5,1** - то же, но интервал времени между появлениями транзактов есть целая часть произведения числа 6 на значение функции FFF.

### **FFF FUNCTION RN1,C4**

**0,0/0.1,0.8/0.5,1.6/1.0,1.9** - описание функции FFF, ее аргументом является случайная величина (на это указывает значение RN1), равномерно распределенная в диапазоне [0,1], функция является непрерывной числовой (указатель C), заданной таблично четырьмя точками: (0;0), (0.1; 0.8), (0.5,1.6), (1.0; 1.9).

### **FFF FUNCTION \*2,D4**

**0,12/1,9/2,8/3,6** - то же, но аргументом является значение второго параметра транзакта, для которого вычисляется значение дискретной величины (D) числовой функции FFF, заданной таблично четырьмя узловыми точками. Это текущее значение округляется до ближайшего большего значения аргумента в узловой точке.

**SEIZE PLOT** - занятие устройства PLOT приходящим на его вход транзактом; если устройство занято, то транзакт задерживается в очереди к этому устройству.

**RELEASE PLOT** - освобождение устройства PLOT обслуженным транзактом.

**ENTER MEM,12** - занятие транзактом 12 единиц емкости в накопителе MEM.

**LEAVE MEM\*2** - освобождение k единиц памяти в накопителе MEM, где k - значение 2-го параметра транзакта.

**STR STORAGE 4096** - описание накопителя STR емкостью 4096 единиц.

**TERMINATE 3** - удаление транзакта из системы, при этом содержимое итогового счетчика уменьшается на 3 единицы, моделирование заканчивается, если содержимое счетчика станет равным или меньше нуля.

**ADVANCE A,B** - задержка транзакта на время, определяемое содержимым полей A и B, смысл величин, записываемых в этих подполях, такой же, как и в блоке GENERATE.

**SPLIT 3,LLL,6** - копирование транзактов. В данном случае создаются три копии исходного транзакта. Исходный транзакт направляется в следующий по порядку блок, а созданные копии - в блок с меткой LLL. При этом параметр 6 основного транзакта увеличивается на единицу, а транзактов-копий - на 2, 3, 4 соответственно.

**ASSEMBLE 5** - объединение транзактов. Первый из вошедших в блок транзактов продолжит движение в системе после того, как в блок придут еще четыре транзакта.

**ASSIGN 2, NAP** - изменение параметров транзактов, в данном случае второй параметр транзакта получит значение NAP.

**ASSIGN 3+, V4** - изменится значение третьего параметра транзакта - к нему прибавится значение V4.

**TRANSFER ,MET** - безусловная передача управления оператору с меткой (номером) MET.

**TRANSFER ,BOTH, LAB1, UNN** - переход к оператору с меткой LAB1. Если он невозможен, то к оператору с меткой UNN, если и он невозможен, то транзакт задерживается до следующего момента дискретного модельного времени, в который повторяются указанные попытки перехода.

**TRANSFER .4, AAA, LAB** - транзакт с вероятностью 0.4 переходит к оператору с меткой LAB и с вероятностью 0.6 к оператору с меткой AAA.

**TRANSFER PICK, STK7, STK21** - равновероятный переход к операторам с номерами STK7, STK7+1, STK7+2, ..., STK21.

**TRANSFER FN, AAA, 5** - переход к оператору, метка которого равна сумме значения функции AAA и числа 5.

**TRANSFER P, 4, 41** - переход к оператору, метка которого равна сумме значения параметра N 4 транзакта и числа 41.

**TRANSFER SBR, PRC, 7** - переход к оператору PRC с записью в параметр N 7 транзакта метки данного оператора.

**LOOP 6, MET** - организация цикла - переход 5 раз к оператору с меткой MET и на шестой раз - к следующему по порядку оператору.

**TEST E V7, K256, LAB** - переход по условию (условная передача управления): в позициях 13-18 записывается знак отношения, в первых двух подполях поля переменных записываются сравниваемые величины, если условие выполняется, то перехода нет, иначе переход есть к оператору с меткой LAB. Символы отношений: G - больше, L - меньше, E - равно, NE - неравно, LE - меньше или равно, GE - больше или равно. В данном примере перехода нет, если  $V7 = 256$ , иначе переход к оператору с номером LAB.

**QUEUE SQV** - оператор организации очереди, длина очереди SQV увеличивается на единицу.

**DEPART SQV** - то же, но длина очереди уменьшается на единицу.

**PRIORITY 2** - транзакту присваивается приоритет 2.

**TABULATE MAT** - в соответствующий интервал гистограммы с именем MAT добавляется единица.

**MAT TABLE P3,8,1,5** - описание таблицы (гистограммы) MAT, предназначенной для табулирования величины, значения которой находятся в третьем параметре транзакта, входящего в оператор TABULATE MAT, верхний предел левого интервала гистограммы равен 8, ширина каждого последующего интервала равна 1, всего интервалов 5.

**5 VARIABLE X2** - K25 - вычислительный оператор, в данном случае из хранимой величины N 2 вычитается число 25 и результат присваивается переменной N 5.

**SAVEVALUE 5,\*3** - хранимая величина N 5 (X5) получает значение третьего параметра транзакта.

## Приложение 2

### Список стандартных числовых атрибутов

Типы объектов	Символическое обозначение	Диапазон изменения	Описание
Транзакты	PF	$\pm(2^{31}-1)$	Текущее значение параметра (формата F — «слово», H — «полуслово», B — «байт», L — «плавающая точка») транзакта, обрабатываемого в данный момент
	PH	$\pm(2^{15}-)$	
	PB	$\pm(2^7-1)$	
	PL	$\pm(2^{24}-1)$	
	PR	0... 127	Приоритет обрабатываемого транзакта
	MI	$\pm(2^{31}-1)$	Транзактное время
	MP		Параметрическое транзактное время



--	--	--	--

Блоки	N W	$2^{24}-1$ $2^{18}-1$	Счетчик входов в блок Счетчик текущего содержимого блока
Память	F	Логические 1 или 0	Состояние устройства
	FR	0... 999	Коэффициент использования (в долях от 1000)
	FC	$2^{31}-1$	Счетчик входов
	FT	$2^{31}-1$	Среднее время/транзакт
	S	$2^{31}-1$	Текущее содержимое памяти
	R	$2^{31}-1$	Число свободных единиц памяти
	SR	0... 999	Коэффициент использования (в долях от 1000)
	SA	$2^{31}-1$	Среднее содержимое памяти
	SC	$2^{31}-1$	Счетчик входов
	ST	$2^{31}-1$	Среднее время, транзакт
	SM	$2^{31}-1$	Максимальное содержимое памяти
Переменные	V	$\pm(2^{31}-1)$	Значение арифметической переменной
	FV	$10^{-78} \dots 10^{75}$	Переменные с плавающей запятой
	BV	1 или 0	Значение булевской переменной
Функции	FN	$\pm(2^{31}-1)$	Значение функции
Очереди	Q	$2^{31}-1$	Текущая длина очереди
	QA	$2^{31}-1$	Средняя длина очереди
	QM	$2^{31}-1$	Максимальная длина очереди
	QC	$2^{31}-1$	Общее число элементов в очереди
	QZ	$2^{31}-1$	Число элементов, не задержанных в очереди

	QT	$2^{31}-1$	Среднее время пребывания транзакта в очереди
	QX	$2^{31}-1$	Среднее время транзакта в очереди без учета нулевых входов
Таблицы	TB	$\pm(2^{31}-1)$	Среднее значение аргумента
	TC	$2^{31}-1$	Счетчик входов в таблицу
	TD	$2^{31}-1$	Среднее квадратичное отклонение аргумента
Матрицы	MX (a, b)	$\pm\{2^{31}-1\}$	Текущее содержание матрицы (формата X — «слово», H — «полу-слово», B — «байт», L — «плавающая точка») строка- a, столбец- b
	MH (a, b)	$\pm(2^{15}-1)$	
	MB (a, b)	$\pm(2^7-1)$	
	ML (a, b)	$\pm(2^{24}-1)$	
Ячейки	XF	$\pm(2^{31}-1)$	Текущее содержание ячейки (формата F — «слово», H — «полу-слово», B — «байт», L — «плавающая точка»)
	XH	$\pm(2^{15}-1)$	
	XB	$\pm(2^7-1)$	
	XL	$\pm(2^{24}-1)$	
Группы	G	$2^{15}-1$	Число членов группы в текущий момент времени
Список пользователя	CA	$2^{15}-1$	Среднее число элементов в списке
	CH	$2^{15}-1$	Текущее число элементов в списке
	CM	$2^{15}-1$	Максимальное число элементов
	CC	$2^{31}-1$	Общее число элементов в списке
	CT	$2^{31}-1$	Среднее время пребывания элементов в списке
Системные атрибуты	RN1...RN8	0...0,999999	Случайное число, используемое как аргумент функции
		0...999	Случайное число в других случаях
	C1	$2^{31}-1$	Текущее значение относительного времени
	AC1	$2^{31}-1$	Текущее значение абсолютного времени
	TG	$2^{31}-1$	Содержание счетчика завершения

## Приложение 3

## Список блоков GPSS

Операция	Операнды							Примечание
	A	B	C	D	E	F	G	
ADVANCE (задержать)	Среднее время { @ }	[Модификатор разброса [ K, cгаj, cга \$имя cга*j кроме fnj, fn\$ имя fn*j] или модификатор функций [fn fn\$ имя, fn*j]						@→ k,сча j, сча \$ имя сча*j { } - обязательный операнд [ ]- необязательный
ASSEMBLE (соединить)	Счетчик соединения { @ }							
ASSIGN (назначить)	№ параметра { @ }	Назначаемая величина	N модифицирующей					

		{ @ }	функции [ @ ]					
BUFFER (возобновить просмотр)								
DEPART (покинуть очередь)	Имя очереди { имя }	Число единиц [ @ ]						
ENTER (войти)	Имя много- канального устройства { имя }	Число единиц [ @ ]						
GATE { LS } { LR } (впустить)	Имя логического переключател я { имя }	Следующий блок при невыполнен ии логического условия [ имя ]						
GATE { M } { NM } (впустить)	Имя блока ASSEMBLE GATHER или MATCH { имя }	—//—						
GATE { NI, I } { NV, V } (впустить)	Имя прибора { имя }	—//—						

GATE { SE, SF SNE, SNF (впустить)	Имя многоканальн ого устройства {имя}	—//—						
GATHER (собрать)	Счетчик сборки { @ }							
GENERATE (генерировать)	Среднее время интервала {имя}	Модификато р разброса [k,craј, cra\$ имя, кроме FNJ,FN\$имя ] или модификато р функции [FNJ, FN\$имя]	Интервал смещения [k,craј, cra\$ имя]	ограничител ь [k,craј, cra\$ имя]	Уровень приорит ета [k,craј, cra\$ имя]	Число параметров [k,craј, cra\$ имя]	Тип парам етров [F]	
LEAVE (выйти)	Имя многоканальн ого устройства { @ }	Число единиц [ @ ]						
LINK (ввести в цепь)	Имя цепи пользователя {имя}	Критерий присоединен ия {LIFO,FIF O, PJ}	Альтернативн ый блок {имя}					
LOGIC {I,R,S}	Имя логического							

(воздействовать на логический переключатель)	переключатель {имя}							
LOOP (организовать цикл)	Номер параметра { @ }	Следующий блок {имя}						
MARK (отметить)	№ параметра { @ }							
MATCH (синхронизировать)	Адрес сопряженного блока {имя}							
MSAVEVACVE (сохранить значение)	Имя матрицы {имя}[±1]	№ строки { @ }	№ столбца { @ }	Величина, которую следует сохранить { @ }	Тип матрицы [X,H,B или L]	Возможность снятия с обслуживания		
PREEMPT (захватить)	Имя прибора {имя}	Возможность захвата по приоритету [PR]	Блок в который переходит захваченный транзакт [имя]	Параметры захваченного транзакта [ @ ]				
PRINT (напечатать)	Нижний предел [имя]	Верхний предел [имя]	Мнемоническое обозначение элемента <sup>(1)</sup>	Указатель разбивки по страницам [любая буква]				

PRIORITY (назначить приоритет)	Новое значение приоритета { @ }	Возможност ь возобновлен ия просмотра [BUFFER]						
QUEUE (встать в очереди)	Имя очереди { имя }	Число единиц [ @ ]						
RELEASE (освободить)	Имя прибора { имя }							
RETURN (вернуть)	Имя прибора { имя }							
SAVEUALVE (сохранить значение)	Имя сохраняемой величины { имя } [ ± 1 ]	Величина, которую следует сохранить { @ }						
SEIZE (занять)	Имя прибора { имя }							
SELECT (в логическом режиме) { V, NV, I, NI, SE, SNE, SF, SNF, LR, LS } (выбрать)	Параметр в который вносят номер элемента { @ }	Нижний предел { @ }	Верхний предел { @ }			Альтернати вный выход { имя }		

SELECT (в режиме max и min) (выбрать)	—//—	—//—	—//—		Исследу емый атрибут (любой сча, кроме MX,МН )			
SELECT (в режиме отношения) (выбрать)	—//—	—//—	—//—	Величина для сравнения { @ }	Исследу емый атрибут	Альтернати вный выход { имя }		
SPLIT (расщепить)	Число потомком { @ }	Следующий блок для потомков { имя }	Параметр в который заносят порядковый номер { @ }	Число параметров у каждого потомка { @ }				
TABVLATE (табулировать)	Имя таблицы { имя }	Весовой фактор { @ }						
TERMINATE (завершить)	Счетчик числа завершений [ @ ]							
TEST { G,GE,E,NE,LE, L } (проверить)	Первая величина { @ }	Вторая величина { @ }	Следующий блок при невыполнении и условия					



			{имя}					
TRANSFER (в условном режиме)(переда ть)	Режим выбора BOTH	Первый исследуемы й блок {имя}	Второй исследуемый блок {имя}					
TRANSFER (в статическом режиме) (передать)	Режим выбора { @ }	Первый блок {имя}	Второй блок {имя}					
TRANSFER (в безусловном режиме)	Режим выбора не используется	Блок в который переходит транзакт {имя}						
UNLINK (вывести из цепи)	Имя цепи пользователя {имя}	Блок в который входят выведенные транзакты {имя}	Счетчик выводимых транзактов { @,all }	№ параметра [ @ ] или [BACK]	Аргуме нт парност и { @ }	Альтернати вный выход {имя}		