

Оглавление

Введение.....	3
Общие положения	3
Объекты GPSS	3
Именованние объектов.....	4
Типы данных	4
Правила описания стандартных числовых и стандартных логических атрибутов.....	4
Арифметическое выражение	5
Логическое выражение	6
Транзакты.....	6
Продвижение транзактов по модели	7
Блоки генерации и задержки транзактов	7
Блок GENERATE.....	7
Блок PRIORITY	8
Блок MARK.....	8
Блок ASSIGN	9
Блок TERMINATE (базовое описание)	9
Блок ADVANCE	10
Ресурсы.....	10
Блок SEIZE	11
Блок RELEASE	11
Оператор STORAGE	13
Блок ENTER	13
Блок LEAVE.....	14
Блок LOGIC.....	14
Блоки и операторы организации вычислений	15
Блок SAVEVALUE.....	15
Оператор MATRIX.....	15
Блок MSAVEVALUE	16
Оператор VARIABLE.....	17
Оператор FVARIABLE	17
Оператор BVARIABLE.....	17
Оператор FUNCTION (базовое описание)	17
Оператор INITIAL	19
Блоки управления движением транзактов	19
Блок TRANSFER (базовое описание).....	19
Блок TEST	20
Блок GATE	20
Блок LOOP	21
Блоки и операторы сбора статистики.....	22
Блок QUEUE	22
Блок DEPART	22
Оператор TABLE.....	23
Блок TABULATE.....	23
Блоки работы с семейством транзактов	24
Блок SPLIT	24
Блок ASSEMBLE	24
Блок MATCH	25
Останов процесса моделирования	25
ПРИЛОЖЕНИЕ.....	27

Список дополнительных блоков и расширений	27
Блоки работы со списками пользователя	27
Блок LINK	27
Блок UNLINK.....	28
Приборы с прерыванием.....	29
Блок PREEMPT	29
Блок RETURN.....	31
Расширенные описания блоков.....	31
Блок TRANSFER (полное описание).....	31
Блок SELECT	34
Оператор FUNCTION (полное описание).....	35
Сводный список стандартных атрибутов объектов языка GPSS.....	37
Задания по составлению имитационных программ для самостоятельной проработки	41
ЛИТЕРАТУРА	43

Введение

GPSS представляет собой алгоритмический язык, ориентированный на описание параллельных процессов с последующим моделированием путем построения имитационного процесса.

Программа на языке GPSS, как и во многих алгоритмических языках, представляет собой последовательность операторов. Особенностью реализации языка GPSS является описание развития процессов посредством обращений к некоторым фиксированным процедурам посредством движущихся по этим описаниям динамических объектов, называемых *транзактами*. Видимо, в силу задания оператора в виде обращения к некоторой стандартной подпрограмме, исполняемые операторы в GPSS называются *блоками*. Таким образом, программа имитационной модели задается в виде последовательности блоков. Эта последовательность называется в дальнейшем *треком*. Под *оператором* в языке GPSS понимается некоторое описание, не входящее в трек, но описывающее некоторые параметры блоков или характеристики управления. Кроме того, в GPSS существует ряд ключевых слов, выполняющих служебные функции, структура которых фиксирована и оговаривается заранее. Концепция языка GPSS опирается на понятия *объектов*, которые являются основным инструментом создания имитационной модели. Система GPSS имеет развитую библиотеку процедур.

Общие положения

Объекты GPSS

Объекты являются концептуальными единицами, с помощью которых создается имитационная модель. Объекты либо создаются автоматически – при ссылке на них в модели, либо должны быть обязательно описаны перед их использованием.

В языке GPSS существуют следующие объекты:

основные

- транзакты
- блоки
- операторы

ресурсы

- приборы
- памяти
- ключи

группы и списки

- семейства транзактов
- группы транзактов
- числовые группы
- списки пользователя

вычислительные

- датчики случайных чисел
- функции
- переменные
- булевские переменные
- сохраняемые величины
- матрицы сохраняемых величин

[Оглавление](#)

статистические

- очереди
- таблицы

Каждый представленный объект имеет набор стандартных числовых атрибутов (СЧА) и стандартных логических атрибутов (СЛА), определяющий свойства объекта.

Именованние объектов

Для идентификации объектов в модели используются имена. Имена могут быть числовыми и символическими. Основными именами являются числовые имена.

Числовое имя - целое положительное число.

Символическое имя - последовательность символов. Символы включают прописные буквы A-Z, строчные буквы a-z, цифры 0-9 и символ _ (подчеркивание).

Правила создания символических имен:

- длина имени от 1 до 250 символов
- имя должно начинаться с символа
- имя не должно быть ключевым словом GPSS.

Система GPSS не различает в обозначениях верхний и нижний регистры (прописные или строчные буквы)

Нельзя присваивать объектам имена операторов и блоков, а также стандартных числовых атрибутов, используемых в системе.

Имена используются также в качестве меток операторов или блоков GPSS.

Типы данных

В системе выделяют три типа данных: целочисленный (Integer), вещественный (Real) и строковый (String). Первые два относятся к числовому типу данных.

Целочисленный тип - 32-разрядные целые числа. Если во время арифметических операций происходит переполнение целого числа, то выполняется его преобразование к вещественному числу.

Вещественный тип - это числа с двойной точностью с плавающей запятой. Они имеют точность 15 десятичных цифр и диапазон экспоненты от -306 до 306.

Строковая константа - последовательность символов ASCII, взятая в двойные кавычки. Строковая константа может иметь любой размер. Для создания и управления строковыми константами в системе имеются строковые процедуры, которые находятся в библиотеке процедур. Строковые константы используются при выводе результатов моделирования в файл результата и формирования собственных сообщений.

Типы данных преобразуются либо явным образом при помощи вызова соответствующих процедур, либо неявно при вычислении выражения.

Правила описания стандартных числовых и стандартных логических атрибутов

Каждый тип объектов имеет набор стандартных числовых (СЧА) и стандартных логических (СЛА) атрибутов, определяющий свойства объектов данного типа. Как правило, у этих атрибутов имя начинается с *префикса*, однозначно соответствующего типу объекта.

Стандартные числовые и логические атрибуты конкретного объекта данного типа могут быть заданы одним из следующих способов:

1) < имя СЧА/СЛА типа объекта > j - где j – положительное целое число, определяющее числовое имя объекта

Пример:

Q1 – текущая длина очереди с именем 1

2) < имя СЧА/СЛА типа объекта > \$ < имя > - где имя – символическое имя объекта

Пример:

Q\$SERVER – текущая длина очереди с именем SERVER

3) <имя СЧА/СЛА типа объекта >*<j> – где j – положительное целое число, определяющее числовое имя параметра активного транзакта, который содержит числовое имя объекта (косвенная адресация)

Пример:

Q*1 – текущая длина очереди, числовое имя которой содержится в первом параметре транзакта

4) <имя СЧА/СЛА типа объекта >*<имя >(или <имя СЧА,СЛА типа объекта >*\$<имя >) – где имя – символическое имя параметра активного транзакта, который содержит числовое имя объекта (косвенная адресация)

Пример:

Q*\$CHANNEL (или Q*\$CHANNEL) – текущая длина очереди , числовое имя которой содержится в параметре транзакта с именем CHANNEL.

Примечание.

Системные СЧА и СЧА транзактов (кроме СЧА Р (значение параметра активного транзакта) являются «атомарными». Такие СЧА являются законченными, для их вычисления не требуется задания имени объекта.

Системные СЧА

Таблица 1 – Системные СЧА

C1	Текущее значение условного времени. Автоматически изменяется в модели и устанавливается в 0 управляющими операторами CLEAR или RESET. Вещественное значение.
RN №	Число, вычисляемое датчиком случайных чисел (№ - до 7 датчиков). Датчик генерирует последовательность равномерно распределенных целочисленных случайных чисел в интервале 0 – 999. При использовании датчика в качестве аргумента функции или объекта в переменной значение будет дробью от 0 до 0.999999.

Арифметическое выражение

Арифметическое выражение представляют собой комбинацию арифметических операторов, стандартных числовых атрибутов, библиотечных функций и констант, которая удовлетворяет правилам элементарной алгебры. Выражение вычисляется согласно приоритетам арифметических операций. Вычисление происходит слева направо. Порядок вычислений может быть изменен с помощью круглых скобок.

При использовании арифметического выражения в качестве операнда блока, оно должно быть записано в круглых скобках.

Пусть имеем запись некоторой операции в виде: **A<знак операции>B**

Результат такой операции представлен в таблицах 1 и 2.

Таблица 2 - Арифметические операции в порядке возрастания приоритета.

Знак операции	Операция	Результат
+	Сложение	A+B возвращает значение суммы A и B
-	Вычитание	A-B возвращает значение разности A и B
#	Умножение	A#B возвращает значение произведения A и B
/	Деление	A/B возвращает значение частного от деления A на B
\	Целочисленное деление	A\B возвращает значение целочисленного деления A на B

@	Целый остаток	$A@B$ возвращает целый остаток от деления A на B
^	Возведение в степень	A^B возвращает значение A , возведенное в степень B

Логическое выражение

Логическое выражение принимает одно из двух значений: 1 или 0.

Таблица 3 - Логические операции

Знак операции	Операция	Результат
& , 'AND'	Логическое «И»	$A \& B$ возвращает 1, если числовые значения A и B отличны от нуля, в противном случае - 0
, 'OR'	Логическое «ИЛИ»	$A \text{ OR } B$ возвращает 1, если хотя бы одно из числовых значения A или B отлично от нуля, в противном случае - 0

Таблица 4 - Условные операции отношения

Знак операции	Операция	Результат
> , 'G'	Больше	$A > B$ возвращает 1, если числовое значение A больше B , в противном случае - 0
>= , 'GE'	Больше или равно	$A \geq B$ возвращает 1, если числовое значение A больше или равно B , в противном случае - 0
= , 'E'	Равно	$A = B$ возвращает 1, если числовое значение A равно B , в противном случае - 0
!= , 'NE'	Не равно	$A \neq B$ возвращает 1, если числовое значение A не равно B , в противном случае - 0
< , 'L'	Меньше	$A < B$ возвращает 1, если числовое значение A меньше B , в противном случае - 0

Транзакты

Транзакты – это динамические объекты, которые создаются в определенные моменты модельного времени, продвигаются интерпретатором системы через блоки модели и затем уничтожаются. Фундаментальное свойство транзакта – *инициативность*, когда выполнение любого блока в программе возможно лишь при подходе к нему транзакта. Иными словами, развитие процессов в имитационной модели выполняют транзакты. Транзакты не могут непосредственно ссылаться друг на друга. С *каждым* транзактом связана некоторая совокупность параметров, организованная в виде вектора. Эта совокупность называется *параметры транзакта* и имеет префикс **P**.

Таблица 5 – СЧА транзакта

P	Значение параметра активного транзакта: целочисленное, вещественное или строковое значение.
PR	Приоритет активного транзакта: целочисленное значение.
M1	Время пребывания в модели активного транзакта. Разно разности текущего значения абсолютного времени и времени рождения активного транзакта: вещественное значение.
XN1	Номер активного транзакта. Целочисленное значение.

Продвижение транзактов по модели

Если транзакт в активном состоянии, интерпретатор пытается его продвинуть по треку блоков. При этом выполняются операции, соответствующие алгоритму блоков.

Если в выполняемом блоке не задана явным образом задержка транзакта, интерпретатор сразу же пытается продвинуть транзакт через следующий блок. Этот процесс продолжается до тех пор, пока транзакт не получает отказ при попытке войти в блок и будет задержан.

Блоки генерации и задержки транзактов

Блоки – это объекты, описывающие изменение состояния модели. Блок характеризуется именем и параметрами, называемыми операндами блока. Блок выполняется только тогда, когда на него поступает транзакт. Блок может принять транзакт или отказать ему во входе, если не выполняются условия входа транзакта в блок. В случае отказа транзакт остается в предыдущем блоке. Если блок принял транзакт, то выполняются операции, соответствующие данному блоку.

Таблица 6 – СЧА блока

N	Общее число транзактов, которые вошли в блок: целочисленное значение.
W	Текущее число транзактов в блоке: целочисленное значение.

Блок GENERATE

Назначение. Блок генерирует транзакты и отправляет их вниз под себя на начало трека.

Синтаксис. GENERATE [A],[B],[C],[D],[E]

Операнд	Назначение	Значение	Значение по умолчанию
A	<u>Среднее</u> время между моментами генерации новых транзактов	Число, СЧА(кроме СЧА транзактов)	0
B	Модификатор, задающий <u>разброс</u> времени между моментами генерации новых транзактов . Два типа модификаторов: - модификатор-интервал - модификатор-функция. <u>Модификатор-интервал</u> появляется, когда значением B является числовой скаляр. Тогда система предполагает задание <u>равномерного</u> закона распределения времени между моментами генерации новых транзактов. <u>Нижняя</u> граница интервала вычисляется как $(A - B)$, <u>верхняя</u> граница как $(A + B)$. <u>Модификатор-функция</u> появляется, когда значением B является функция. Тогда время между моментами генерации новых транзактов определяется как произведение значения операнда A и значения функции, заданной операндом B	Число, СЧА(кроме СЧА транзактов)	0
C	Начальная задержка. Задаёт момент генерирования первого транзакта в блоке GENERATE при первом	Число, СЧА(кроме	Если операнд пуст или равен 0, то момент появ-

	выполнении модели, и после выполнения операции CLEAR. Поля A и B на задержку транзакта не влияют. Начальная задержка может быть меньше, равна или больше среднего времени, заданного в поле A.	СЧА транзактов)	ления первого транзакта определяется операндами A и B
D	Предел генерации. Задаёт максимальное число транзактов, которое будет создано в блоке GENERATE. Если поле D пусто, блок генерирует неограниченное число транзактов. Предел генерации устанавливается повторно оператором CLEAR.	Число, СЧА(кроме СЧА транзактов)	∞
E	Приоритет транзакта, задаваемый при генерации.	Число, СЧА(кроме СЧА транзактов)	0

Блок GENERATE при генерации транзакта привязывает к нему вектор параметров, извлекаемый из динамической области памяти системы моделирования. Он содержит ряд системных параметров (момент рождения, приоритет и др.) и пользовательские параметры, доступные пользователю.

Примеры.

1. GENERATE 25,10

приращение времени для генерации очередного транзакта будет равно случайному числу, равномерно распределённому в интервале от 15 до 20 единиц модельного времени.

2. GENERATE 25,FN\$EXP

приращение времени для генерации транзакта вычисляется как произведение значения числа 25 и текущего значения функции EXP.

3. GENERATE „,25

генерируются 25 транзактов в момент времени, равный 0.

4. GENERATE 720

генерируются транзакты с интервалом 720 единиц модельного времени, причем первый транзакт появится в момент времени, равный 720.

Блок PRIORITY

Назначение. Блок изменяет приоритет активного транзакта.

Синтаксис. PRIORITY A

Операнд	Назначение	Значение	Значение по умолчанию
A	Задаёт новое значение приоритета.	Имя, число, СЧА	Обязательный параметр

Пример.

PRIORITY 10

Вошедшему в блок транзакту присваивается приоритет, равный 10.

Блок MARK

Назначение. Блок изменяет значение момента рождения транзакта или заносит значение текущего абсолютного модельного времени в заданный параметр транзакта.

Синтаксис. MARK [A]

Операнд	Назначение	Значение	Значение по умолчанию
A	Определяет параметр транзакта, которому присваивается значение абсолютного модельного времени	Имя, число, СЧА	Нет

Примеры.

1. MARK

Значение времени рождения транзакта становится равным значению абсолютного модельного времени C1. СЧА M1 данного транзакта будет равен времени с момента прохождения транзактом блока MARK.

2. MARK TIME

Параметру с именем TIME присваивается значение абсолютного модельного времени C1.

Блок ASSIGN

Назначение. Блок заменяет, увеличивает или уменьшает значение параметра активного транзакта.

Синтаксис. ASSIGN A,B[,C]

Операнд	Назначение	Значение	Значение по умолчанию
A	Определяет номер или имя параметр транзакта, которому присваивается значение. Если значение параметра нужно увеличить или уменьшить, то справа в операнде A ставится знак сложения (+) или вычитания (-)	Имя, число, СЧА	Обязательный параметр
B	Определяет значение, которое следует добавить или вычесть из значения параметра транзакта, заданного операндом A, или заменить его	Число, СЧА	Обязательный параметр
C	Задаёт имя модификатора-функции. При использовании операнда C значение операнда B умножается на значение модификатора-функции. Полученное произведение становится значением, которое изменяет значение параметра транзакта, заданного в операнде A	СЧА функции	Нет

Примеры.

1. ASSIGN 2,15

При поступлении транзакта к этому блоку в параметр номер 2 транзакта будет занесено значение 15.

2. ASSIGN 2+,3

При поступлении транзакта к этому блоку значение параметра 2 этого транзакта будет увеличено на число 3

3. ASSIGN 2, 5, FN\$EXP

При поступлении транзакта к этому блоку происходит обращение функции FN\$EXP, полученное значение функции умножается на 5, а затем результат всей операции загружается во 2-й параметр активного транзакта.

Блок TERMINATE (базовое описание)

Назначение. Блок уничтожает активный транзакт и все его параметры.

Синтаксис. TERMINATE

Полная транскрипция оператора приведена ниже в механизме останова моделирования.

Блок ADVANCE

Назначение. Блок задерживает продвижение активного транзакта на заданное в параметрах блока время

Синтаксис. ADVANCE A[,B]

Операнд	Назначение	Значение	Значение по умолчанию
A	Среднее время задержки транзакта в блоке ADVANCE	Число, СЧА	Обязательный параметр
B	Модификатор, задающий <u>разброс</u> времени задержки Два типа модификаторов: - модификатор-интервал - модификатор-функция. <u>Модификатор-интервал</u> появляется, когда значением B является числовой скаляр. Тогда система предполагает задание <u>равномерного</u> закона распределения времени задержки транзакта. <u>Нижняя граница</u> интервала вычисляется как $(A - B)$, <u>верхняя граница</u> как $(A + B)$. <u>Модификатор-функция</u> появляется, когда значением B является функция. Тогда время задержки определяется как произведение значения операнда A и значения функции, заданной операндом B	Число, СЧА	0

Примеры.

1. ADVANCE 10

продвижение активного транзакта будет задержано на 10 единиц модельного времени

2. ADVANCE 15, 5

время задержки транзакта будет равно случайному числу, равномерно распределенному в интервале от 10 до 20 единиц модельного времени

3. ADVANCE 10, FN\$TOBR

время задержки транзакта будет равно произведению числа 10 и текущего значения функции с именем TOBR

Ресурсы

Ресурсы – это часть трека, содержащая некоторые условия блокировки этого участка для вхождения в него транзактов. В связи с появлением блокировок к этим ресурсам организуется очередь транзактов, управляемые системой моделирования.

Приборы

Под прибором понимается такой участок трека (ресурс), который заблокирован *одним логическим семафором*. Если семафор открыт, то транзакт может войти на этот участок трека (захватить ресурс). Если семафор закрыт, то все подошедшие транзакты выстраиваются в очередь типа FIFO к этому участку трека (очередь к ресурсу).

Параметры, связанные с приборами, имеют префикс **F**.

СЧА приборов

Таблица 7 - СЧА

F	Состояние прибора. Равно 0, если прибор свободен, и 1 - во всех остальных случаях. Целочисленное значение.
FI	Флаг прерывания прибора : 1 - если прибор находится в состоянии прерывания, 0 - в противном случае. Целочисленное значение.
FR	Коэффициент использования прибора в долях тысячи Вещественное значение.
FC	Общее число занятий прибора. Целочисленное значение.
FT	Среднее время использования прибора одним бранзактом. Вещественное значение.

СЛА приборов

Таблица 8 - СЛА

NU	Равен 1, если прибор свободен, иначе 0
U	Равен 1, если прибор занят, иначе 0

Блок SEIZE

Назначение. Блок позволяет активному транзакту занять прибор (захватить ресурс) или встать в очередь к прибору.

Синтаксис. SEIZE A

Операнд	Назначение	Значение	Значение по умолчанию
A	Имя занимаемого прибора	Имя, число, СЧА	Обязательный операнд

Особенности выполнения.

1. При попытке транзакта войти в блок SEIZE всегда проверяется, свободен ли прибор (открыт ли семафор).
2. Если прибор свободен (семафор открыт), то транзакт, занявший прибор, пытается перейти к следующему по номеру блоку, а семафор закрывается. Прибор остается занятым до тех пор, пока занимающий его транзакт не войдет в соответствующий блок RELEASE. Прежде чем освободить прибор, транзакт может пройти через произвольное число блоков.
3. Блок SEIZE отказывает во входе транзакту, если прибор занят (семафор закрыт). При этом транзакт помещается в список задержки прибора в конец своего приоритетного класса

Пример

SEIZE Server

Активный транзакт пытается занять прибор с именем Server.

Блок RELEASE

Назначение. Блок позволяет активному транзакту освободить занятый прибор (открыть семафор).

Синтаксис. RELEASE A

Операнд	Назначение	Значение	Значение по умолчанию
A	Имя освобождаемого прибора	Имя, число, СЧА	Обязательный операнд

[Оглавление](#)

Особенности выполнения.

1. Если прибор занят активным транзактом, он освобождает прибор и пытается войти в следующий по порядку блок.
2. Если активный транзакт освобождает прибор, следующий транзакт выбирается из списка задержки и занимает прибор. Если в списках транзакты отсутствуют, прибор становится не занятым.
3. Активизируются транзакты из списка задержки блоков GATE NU.

Пример.

RELEASE WorkStation

Транзакт покидает ресурс (прибор) с именем WorkStation и открывает семафор.

Памяти

Под памятью понимается такой участок трека (ресурс), который заблокирован *одним арифметическим семафором*. Это означает, что семафор пропускает не более заданного в этом семафоре количества транзактов в ресурс. Если количество подошедших к ресурсу транзактов превышает заданное ограничение, то семафор закрывается, а оставшиеся транзакты выстраиваются к ресурсу (памяти) в очередь типа FIFO.

Параметры, связанные с памятьями, имеют префикс **S**.

Таблица 9 – СЧА памяти

S	Текущее содержимое памяти. Целочисленное значение.
R	Число свободных единиц памяти. Целочисленное значение.
SR	Коэффициент использования памяти в долях тысячи. Вещественное значение.
SA	Взвешенное по времени среднее содержимое памяти. Вещественное значение.
SM	Максимальное содержимое памяти. Целочисленное значение.
SC	Общее количество использовавшихся элементов памяти. Целочисленное значение.
ST	Среднее время пребывания транзактов в памяти. Вещественное значение.
SE	Флаг незанятости памяти. (:1 - свободна, 0 – занята). Целочисленное значение.
SF	Флаг заполненности памяти (:1 - заполнена, 0 - не заполнена). Целочисленное значение.
SV	Флаг готовности памяти (:1 - готова, 0 - не готова). Целочисленное значение.

Таблица 10 – СЛА памяти

SE	Равен 1, если память пуста (нулевое содержимое), иначе 0
SNE	Равен 1, если память не пуста (ненулевое содержимое), иначе 0
SF	Равен 1, если память заполнена, иначе 0
SNF	Равен 1, если память не заполнена, иначе 0
SV	Равен 1, если память используется, иначе 0
SNV	Равен 1, если память не используется, иначе 0

Оператор STORAGE

Назначение. Оператор STORAGE определяет максимальное значение емкости в памяти (размер семафора). Оператор относится к декларирующему типу и ставится в начале программы.

Синтаксис. <имя памяти> STORAGE A

Имя памяти – символическое или числовое имя памяти, ставится в поле меток.

Операнд A определяет емкость данной памяти: обязательный операнд целого типа

Пример.

RAM_Workstation STORAGE 1024

Оператор определяет память с именем RAM_Workstation с общей емкостью 1024 элементов.

Блок ENTER

Назначение. Блок позволяет активному транзакту либо занять определенное число элементов памяти, либо встать в очередь к данной памяти

Синтаксис. ENTER A,[B]

Операнд	Назначение	Значение	Значение по умолчанию
A	Имя занимаемой памяти	Имя, число, СЧА	Обязательный параметр
B	Число занимаемых элементов памяти	Имя, число, СЧА	1

Особенности выполнения.

1. Операнд A должен указывать на заранее определенную оператором STORAGE память.
2. При попытке транзакта войти в блок ENTER всегда проверяется, существует ли необходимое число свободных элементов данной памяти и находится ли память в состоянии «готова к использованию». Для определения необходимого числа свободных элементов памяти используется операнд B.
3. Если необходимое число свободных элементов есть и память готова, она занимает. Число свободных элементов памяти уменьшается на заданную в параметре B величину..
4. Транзакт, занявший память, пытается перейти к следующему по треку блоку.
5. Элементы памяти остаются занятыми до тех пор, пока занимающий их транзакт не войдет в соответствующий блок LEAVE.
6. Блок ENTER отказывает во входе транзакту, если нет необходимого числа свободных элементов данной памяти. При этом устанавливается его индикатор задержки и транзакт помещается в список задержки памяти в конец своего приоритетного класса

Примеры.

1. ENTER RAM

Транзакт пытается занять один элемент памяти с именем RAM

2. ENTER RAM, P\$PAM

Транзакт пытается занять память с именем RAM. Необходимое число элементов памяти содержится в параметре транзакта с именем PAM.

Блок LEAVE

Назначение. Блок позволяет активному транзакту освободить определенное число элементов памяти.

Синтаксис. LEAVE A,[B]

Операнд	Назначение	Значение	Значение по умолчанию
A	Имя освобождаемой памяти	Имя, число, СЧА	Обязательный параметр
B	Число освобождаемых элементов памяти	Имя, число, СЧА	1

Особенности выполнения.

1. Операнд A должен указывать на заранее определенную оператором STORAGE память.
2. Для определения числа освобождаемых элементов памяти используется операнд B.
3. Транзакт, освободивший элементы памяти, пытается перейти к следующему по треку блоку.
4. После освобождения памяти, список задержки памяти просматривается в порядке убывания приоритетов, определяются транзакты, потребность в памяти которых может быть удовлетворена. Используется правило «первый подходящий с пропусками». Успешные транзакты входят в блок ENTER.

Примеры.

1. LEAVE RAM

Транзакт освобождает один элемент памяти с именем RAM

2. LEAVE RAM, P\$ PAM

Транзакт освобождает элементы памяти с именем RAM. Освобождаемое число элементов памяти содержится в параметре транзакта с именем PAM.

Логические ключи

Логические ключи представляют собой логические переменные, предназначенные выполнять функцию логического семафора на любом участке трека.

Таблица 11 – СЛА ключей

LR	Равен 1, если логический ключ "выключен", иначе 0
LS	Равен 1, если логический ключ "включен", иначе 0

Блок LOGIC

Назначение. Блок изменяет состояние логического ключа.

Синтаксис. LOGIC X A

X – логический оператор

A – имя (номер) логического ключа

Операнд	Назначение	Значение	Значение по умолчанию
X	Логический оператор	S,R или I	Обязательный параметр
A	Имя логического ключа	Имя, число,	Обязательный параметр

		СЧА	
--	--	-----	--

Особенности выполнения.

1. Блок LOGIC предназначен для того, чтобы устанавливать, сбрасывать или инвертировать (менять на противоположное) состояние логического ключа.
2. Логический ключ может находиться в двух состояниях: **S** (включен или 1) или **R** (выключен или 0).
3. Если логический оператор равен «S» или «R», то логический ключ, заданный операндом A, устанавливается во включенное или выключенное состояние соответственно.
4. Если логический оператор равен «I», то логический ключ, заданный операндом A, инвертируется. Это значит, что если он был включен, он будет выключен и наоборот.

Примеры.

1. LOGIC S SWITCH
2. LOGIC R SWITCH
3. LOGIC I SWITCH

Блоки и операторы организации вычислений

Вектор сохраняемых величин – это совокупность параметров, предназначенная для сохранения скалярных значений, организованная в виде вектора. Адресация такой величины имеет вид $X\ I$, где X – стандартное имя вектора, I – номер или имя элемента вектора X.

Блок SAVEVALUE

Назначение. Блок присваивает, увеличивает или уменьшает значение сохраняемой величины.

Синтаксис. SAVEVALUE A[±],B

Операнд	Назначение	Значение	Значение по умолчанию
A	Имя изменяемой сохраняемой величины	Имя, число, СЧА	Обязательный операнд
B	Значение, которое присвоено, прибавлено или вычтено из сохраняемой величины	Имя, число, СЧА	Обязательный операнд

Примеры.

1. SAVEVALUE 25, 7
Значение сохраняемой величины X25 становится равным 7.
2. SAVEVALUE WES+, 2
Значение сохраняемой величины X\$WES увеличивается на 2.

Оператор MATRIX

Матрица сохраняемых величин – это совокупность параметров, предназначенная для сохранения скалярных значений, организованная в виде двумерной матрицы. Адресация такой величины имеет вид $MX\ (I, J)$, где MX – стандартное имя матрицы; I, J – номера или имена элемента матрицы MX.

Оператор определяет матрицу сохраняемых величин. Имеет декларирующий характер и располагается выше описаний треков.

Синтаксис. <имя матрицы> MATRIX A, B, C

имя матрицы – числовое или символическое имя, обязательный параметр. Расположен в поле метки.

A - неиспользуемое поле (для совместимости с ранними версиями GPSS).

B - максимальное количество элементов в первом измерении. (количество строк матрицы). Допустимое значение – целое число. Обязательный операнд

C - максимальное количество элементов во втором измерении (количество столбцов матрицы). Допустимое значение – целое число. Обязательный операнд.

Пример.

REZULT MATRIX ,15, 3

Оператор определяет матрицу с именем REZULT с 15 строками и 3 столбцами.

Блок MSAVEVALUE

Назначение. Блок присваивает, увеличивает или уменьшает значение элемента матрицы сохраняемых величин.

Синтаксис. MSAVEVALUE A[±], B, C, D

Операнд	Назначение	Значение	Значение по умолчанию
A	Имя изменяемой матрицы сохраняемых величин	Имя, число, СЧА	Обязательный операнд
B	Номер строки матрицы	Имя, число, СЧА	Обязательный операнд
C	Номер столбца матрицы	Имя, число, СЧА	Обязательный операнд
D	Значение, которое присвоено, прибавлено или вычтено из элемента матрицы	Имя, число, СЧА	Обязательный операнд

Особенности выполнения.

1. Матрица должна быть предварительно определена с помощью оператора определения данных MATRIX.
2. Если операнд A сопровождается знаком «+», то к значению элемента матрицы, определенного операндами A, B и C, прибавляется значение, определенное операндом D.
3. Если операнд A сопровождается знаком «-», то из значения элемента матрицы, определенного операндами A, B и C, вычитается значение, определенное операндом D.
4. Если за операндом A нет знака, то значение элемента матрицы, определенного операндами A, B и C, заменяется значением, определенным операндом D.

Пример.

MSAVEVALUE DATA, 4, 5, P7

Элементу матрицы DATA с номером строки 4 и номером столбца 5 присваивается значение седьмого параметра активного транзакта.

Обращение к этому элементу в арифметических и других выражениях имеет вид **MX(4, 5)**

Оператор VARIABLE

VARIABLE является оператором-функцией, определяющим арифметическую целую переменную, имеющую имя **V I**, где **V** – стандартное обозначение переменной типа оператор-функция, **I** – номер или имя функции **VARIABLE**.

Синтаксис: <имя переменной> **VARIABLE** <арифметическое выражение>

Имя переменной - символическое или числовое имя переменной, ставится в поле меток.

Арифметическое выражение определяет значение данной переменной.

Примеры.

1. 5 **VARIABLE** 46+P6

Арифметическая переменная с числовым именем 5 равна сумме числа 46 и значения параметра 6 активного транзакта.

Обращение к этой переменной имеет вид V5.

2. **SUM VARIABLE** (P3 + P4)/5

Арифметическая переменная с символическим именем SUM равна сумме значений 3-го и 4-го параметров активного транзакта, деленной на 5.

Обращение к этой переменной имеет вид V\$SUM.

Арифметическая переменная может использоваться как:

- элемент другой арифметической переменной, булевой переменной;
- аргумент функции, таблицы;
- операнд блока

Оператор FVARIABLE

FVARIABLE является оператором-функцией, определяющим арифметическую переменную с фиксированной точкой, имеющую имя **FV I**, где **FV** – стандартное обозначение, **I** – номер или имя функции **FVARIABLE**.

Синтаксис. <имя переменной> **FVARIABLE** <арифметическое выражение>

Имя переменной - символическое или числовое имя переменной, ставится в поле меток.

Арифметическое выражение определяет значение переменной.

Пример.

PBR **FVARIABLE** (SI-S\$CAN)/5 + 3.6

Ссылка на арифметическую переменную с фиксированной точкой выполняется так же, как и на арифметическую переменную.

Оператор BVARIABLE

Оператор определяет булеву переменную.

Синтаксис. <имя переменной> **BVARIABLE** <булево выражение>

Выражения в операторе **BVARIABLE** кроме логических операторов могут включать операции отношений и вызовы библиотечных процедур.

Пример.

FLAG **BVARIABLE** BV\$CAN1'AND'BV\$CAN2

Значение переменной FLAG равно 1, если булевы переменные CAN1 и CAN2 имеют значение TRUE, и равна 0 в других случаях

Оператор FUNCTION (базовое описание)

Оператор определяет функцию GPSS, заданную таблично.

Синтаксис. <имя функции> **FUNCTION** A, BN

имя функции – числовое или символическое имя, обязательный параметр.

A - аргумент функции; обязательный операнд. Допустимые значения – имя, число, СЧА.

В - тип функции (одна буква) и N - количество пар данных в списке данных функции. Обязательный операнд.

Обращение к функции выглядит, как FN I, где I – имя функции.

Существует несколько типов функций. Тип определяется операндом В оператора FUNCTION.

За строкой FUNCTION A, BN сразу же должна следовать строка, содержащая список пар данных, разделенных символом «/». Каждая пара данных определяет значения аргумента X и значения функции Y, разделенные запятой. Список данных используются для вычисления значения функции по заданным значениям аргумента.

В данном транскрипции оператора рассмотрим 2 типа функций. Полное изложение вариантов задания функции приведено в приложении.

1. Функции типа C – непрерывные числовые функции.

В списке данных функций типа C значения X и Y должны быть целочисленными (Integer) или вещественными (Real). Значения X и Y хранятся, как числа с плавающей точкой двойной точности.

Вычисление функции начинается с вычисления аргумента. Далее определяется интервал ($X_i; X_{i+1}$), на котором находится вычисленное значение аргумента и на этом интервале выполняется линейная интерполяция двойной точности с использованием соответствующих значений Y_i и Y_{i+1} . Результатом является значение функции двойной точности. Если аргумент попадает за предельные значения области определения функции, возвращается значение функции в ближайшей предельной точке.

Примеры.

- 1) ART FUNCTION X1, C3
1.1,10.1/20.5,98.7/33.3,889.2

Оператор определяет кусочно-линейную функцию с двумя линейными участками. Если мы обращаемся к функции FN\$ART, то по значению сохраняемой величины X1 вычисляется функция в соответствии с заданной совокупностью точек.

Пример приближенного представления обратного экспоненциального распределения со средним, равным 1.

- 2) Xpdis FUNCTION RN1, C24
0,0/1.,104/2, .222/3, .355/4, .509/5, .69/6,915/7,1.2/75,1.38
.8, 1.6/84, 1.83/88, 2.12/9, 2.3/92, 2.52/94, 2.81/95, 2.99/96,3.2
.97, 3.5/98, 3.9/99, 4.6/995, 5.3/998,6.2/999,7/9998,8

2. Функции типа D - дискретные функции.

В списках данных функций типа D значения X должны быть целочисленными или вещественными, а значения Y – целочисленными, вещественными или именами.

Функция типа D задает одно и то же значение функции Y_i для всех значений аргумента $X_{i-1} < X \leq X_i$. Значения X в списке данных функции должны быть неубывающими. Внутренне они сохраняются, как числа двойной точности. Когда вычисляется функция, значения X в списке данных функции просматриваются от наименьшего к наибольшему. Когда найдено значение X, которое больше или равно текущему значению аргумента, возвращается соответствующее ему значение Y. Если такое значение X отсутствует, возвращается значение Y или именованная величина, соответствующая самому большому значению X.

Примеры.

- 1) LIR FUNCTION X\$A2, D5
1.1,6.9/2.1,7/6.33,9.4/7,10/9.9,12.01
- 2) RAF FUNCTION RN1, D5

Оператор INITIAL

Оператор INITIAL задает начальное значение сохраняемым величинам, элементам матрицы, логическим ключам.

Синтаксис. INITIAL A, B

A - логический ключ, сохраняемая величина, элемент матрицы, определенные как СЧА. Операнд A должен иметь форму СЧА классов LS, X, MX или имени матрицы. В операнде A нельзя использовать параметры транзакта.

B - присваиваемое значение

Если операнд A указывает на логический ключ, присваиваются только значения 0 или 1. Если операнд B явно задан как 0, то присваивается значение 0. В противном случае присваивается 1. Если операнд A задает имя матрицы, всем ее элементам присваивается значение, указанное операндом B. По умолчанию это 1. Для задания элемента двумерной матрицы может использоваться СЧА класса MX.

Примеры

INITIAL X21, 17

INITIAL L\$KLU, 1

Блоки управления движением транзактов**Блок TRANSFER (базовое описание)**

Назначение. Блок передает транзакт на указанный блок.

Синтаксис. TRANSFER [A], B

Операнд	Назначение	Значение	Значение по умолчанию
A	Вероятность P продолжения движения транзакта по треку	число, СЧА	Режим безусловной передачи
B	Номер или метка блока, куда переходит транзакт с вероятностью (1-P)	Имя, число, СЧА	Нет

Особенности выполнения.**1. Режим статистической передачи**

В этом режиме активный транзакт переходит к блоку, заданному в операнде B, с вероятностью, обратной заданной в операнде A. Операнд A может быть положительной дробью, меньшей единицы или целым положительным числом. Если операнд A - целое число, оно интерпретируется как доля от тысячи. С вероятностью, заданной в операнде A, транзакт продолжает движение по треку.

Пример

TRANSFER 0.3, LAB1

Транзакт с вероятностью 0.3 продолжает движение дальше по треку, с вероятностью 0.7 – поступает на блок, помеченный меткой LAB1.

2. Режим безусловной передачи

Когда операнд A отсутствует, блок TRANSFER функционирует в режиме безусловной передачи. В этом режиме активный транзакт всегда переходит к блоку, заданному в операнде B.

Пример

TRANSFER ,NO_SERV

При входе транзакта в блок TRANSFER, он передается в блок с меткой NO_SERV.

Полное изложение синтаксиса блока помещено в приложении.

Блок TEST

Назначение. Блок выполняет навигацию транзакта в зависимости от результата операции отношения, заданной в этом же блоке.

Синтаксис. TEST X A, B [, C]

Операнд	Назначение	Значение	Значение по умолчанию
X	Операция отношения (см.выше)	Условная операция отношения	Обязательный операнд
A	Левая часть отношения	Имя, число, СЧА	Обязательный операнд
B	Правая часть отношения	Имя, число, СЧА	Обязательный операнд
C	Метка или номер блока, в который будет передан транзакт	Имя, число, СЧА	Режим отказа

Особенности выполнения.

Блок TEST функционирует в двух режимах:

1. Если операнд C не используется, блок TEST функционирует в режиме отказа. Когда транзакт пытается войти в блок TEST, работающий в режиме отказа, и заданное условие не выполняется, транзакт блокируется, и ему не позволено войти в блок TEST, проверка повторяется до тех пор, пока условие не будет выполнено. После выполнения заданного условия активный переходит к следующему по порядку блоку.
2. Если операнд C используется, блок TEST функционирует в режиме передачи транзакта. Когда транзакт пытается войти в такой блок TEST и проверяемое условие не выполняется, транзакт переходит к блоку, указанному в операнде C. Если проверяемое условие выполняется, активный транзакт входит в блок TEST и затем переходит к следующему по порядку блоку.

Примеры.

1. TEST L Q\$SERVER,100

После входа в блок TEST транзакт при выполнении условия (Q\$SERVER<100) перейдет к следующему блоку по треку, иначе транзакт будет задержан вплоть до выполнения указанного условия.

2. TEST L Q\$SERVER,100, MET1

После входа в блок TEST транзакт при выполнении условия (Q\$SERVER<100) перейдет к следующему блоку по треку, иначе транзакт будет передан на блок, помеченный меткой MET1.

Блок GATE

Назначение. Блок передает транзакт в зависимости от состояния объекта.

Синтаксис. GATE X A[,B]

Назначение операндов.

Операнд	Назначение	Значение	Значение по умолчанию
X	Определяет проверяемый логический атрибут объекта	СЛЖ	Обязательный параметр

A	Имя или номер проверяемого объекта	Имя, число, СЧА	Обязательный параметр
B	Определяет блок для режима перехода	Имя, число, СЧА	Режим отказа

Особенности выполнения.

Блок GATE работает в двух режимах:

1. Если операнд B не используется, блок GATE работает в режиме отказа. Когда транзакт пытается войти в блок GATE, работающий в режиме отказа, и условие, указанное в операнде X не выполняется, транзакт задерживается, пока условие не будет выполнено. Если условие выполняется, активный транзакт переходит к следующему блоку по треку.
2. Если операнд B используется, блок GATE работает в режиме перехода. Когда транзакт пытается войти в блок GATE, и условие не выполняется, транзакт направляется к блоку, помеченному операндом B. Если условие выполняется, активный транзакт входит в блок GATE и затем переходит к следующему блоку по треку.

Примеры.

1. GATE FV SERVER

В режиме отказа активный транзакт войдет в блок GATE, если прибор с именем SERVER доступен. В противном случае транзакт блокируется до выполнения условия.

2. GATE SE RAM, NO_RAM

В режиме перехода, если память с именем RAM пуста, транзакт входит в блок GATE и переходит к следующему блоку по треку. В противном случае транзакт переходит к блоку с меткой NO_RAM.

Блок LOOP

Назначение. Блок передает транзакт и уменьшает значение параметра транзакта. Используется для организации циклических процессов.

Синтаксис. LOOP A , B

Операнд	Назначение	Значение	Значение по умолчанию
A	Имя параметра транзакта, содержащего уменьшаемое число	Имя, число, СЧА	Обязательный параметр
B	Метка или номер блока, в который будет передан транзакт	Имя, число, СЧА	Обязательный параметр

Особенности выполнения.

1. Блок всегда принимает транзакт.
2. После входа транзакта в блок числовое значение параметра транзакта, заданного операндом A , уменьшается на единицу
3. Если новое значение параметра больше нуля, то транзакт передается в блок, номер или имя которого содержится в операнде B. В противном случае транзакт переходит к следующему блоку по треку.

Пример.

LOOP 5, CYCLE

Значение пятого параметра транзакта уменьшается на единицу и, если оно остается больше нуля, транзакт передается на блок с меткой **CYCLE**

Блоки и операторы сбора статистики

Блок **QUEUE**

Блок QUEUE (очередь) – регистратор статистики. Предназначен для сбора и обработки статистики на некотором фрагменте трека.

Назначение. Блок отмечает место входа на треке транзакта в регистратор, что соответствует началу сбора статистики для данного транзакта.

Синтаксис. **QUEUE A**

Операнд	Назначение	Значение	Значение по умолчанию
A	Имя регистратора (очереди)	Имя, число, СЧА	Обязательный операнд

Примеры

QUEUE STAT1

В регистраторе с именем **STAT1** отмечается начало сбора статистики для подошедшего транзакта.

Блок **DEPART**

Назначение. Блок отмечает место выхода транзакта из регистратора на треке, что соответствует концу сбора статистики для данного транзакта.

Синтаксис. **DEPART A**

Операнд	Назначение	Значение	Значение по умолчанию
A	Имя регистратора (очереди)	Имя, число, СЧА	Обязательный операнд

Примеры.

DEPART SYSTEM

В регистраторе с именем **SYSTEM** отмечается окончание сбора статистики для подошедшего транзакта.

Результаты по собранной статистике размещаются в СЧА регистратора (очереди)

Таблица 12 – СЧА очередей

Q	Текущая длина очереди. Целочисленное значение.
QA	Взвешенная по времени средняя длина очереди. Вещественное значение.
QM	Максимальная длина очереди. Целочисленное значение.
QC	Общее число входов в очередь. Целочисленное значение.
QZ	Число нулевых входов в очередь. Целочисленное значение.
QT	Среднее время пребывания транзактов в очереди (включая нулевые входы). Вещественное значение.
QX	Среднее время пребывания сообщения в очереди (без нулевых входов). Вещественное значение.

Оператор TABLE

Оператор TABLE – декларирующий оператор, определяет таблицу плотности распределения случайной величины, ее интегральных относительных частот, среднего и стандартного отклонения.

Синтаксис <имя таблицы> TABLE A, B, C, D

Длина имени таблицы ограничена 32 символами.

A – аргумент таблицы – случайная величина, плотность распределения которой находится, как частота попадания на интервалы таблицы.

B – левая граница таблицы (левая граница первого частотного интервала).

C – размер частотных интервалов

D – количество частотных интервалов (включая интервал от $-\infty$ до левой границы таблицы и интервал от правой границы таблицы до $+\infty$).

Пример.

GIST TABLE P\$TOA, 5.12, 10, 10

В этом примере в таблице с именем GIST регистрируется распределение значений параметра TOA транзакта.

Оператор TABLE создает таблицу с десятью частотными интервалами.

Все значения TOA, меньшие или равные 5.1, приводят к изменению первого частотного интервала таблицы. (Обычно частота увеличивается на 1. Однако в операнде B блока TABULATE может использоваться весовой коэффициент, что приводит к добавлению весового коэффициента к значению частоты. Весовой коэффициент также применяется для среднего и стандартного отклонения, что равнозначно нескольким входам в блок TABULATE).

Если значение параметра TOA больше 5.12, будет изменено значение частоты в десятом (последний) частотном интервале. Если значение параметра TOA не попадает ни в первый, ни в последний частотный интервал, оно используется для изменения частоты в интервалах со 2-го по 9-й.

Статистика, собранная в таблице, выводится в стандартный отчет системы GPSS. Кроме того, часть результатов может быть взята из СЧА табуляции.

Блок TABULATE

Назначение. Блок добавляет данные в таблицу

Синтаксис. TABULATE A [, B]

Операнд	Назначение	Значение	Значение по умолчанию
A	Имя таблицы, в которую заносится табулируемая величина (аргумент) в момент входа транзакта в данный блок	Имя, число, СЧА	Обязательный операнд
B	Весовой коэффициент	Имя, число, СЧА	1

Особенности выполнения.

1. Таблица должна быть предварительно определена с помощью оператора определения данных TABLE.
2. Если задан операнд B, то он задает количество раз, которое табулируемая величина (аргумент) должна быть занесена в таблицу при каждом входе в блок.
3. Операнд B должен быть положительным.

4. В результате моделирования таблица с именем А содержит гистограмму, а также значения среднего и стандартного отклонения регистрируемой величины (аргумента).
5. Блок TABULATE является частью трека и срабатывает лишь при попадании в него транзактов

Пример.

TABULATE GIST

Когда транзакт входит в блок TABULATE, происходит занесение в таблицу GIST величины (аргумента), определенной в операторе TABLE.

Таблица 13 – СЧА табуляции

ТВ	Среднее значение аргументов таблицы. Вещественное значение.
ТС	Общее число аргументов таблицы. Целочисленное значение.
TD	Среднеквадратичное отклонение для аргументов таблицы. Вещественное значение.

Блоки работы с семейством транзактов

Блок SPLIT

Назначение. Блок генерирует транзакты того же семейства, что и активный транзакт.

Синтаксис. SPLIT A, [B], [C]

Операнд	Назначение	Значение	Значение по умолчанию
A	Количество генерируемых транзактов-потомков семейства	Число, СЧА	Обязательный параметр
B	Имя или номер блока, куда переходят транзакты-потомки	Имя, число, СЧА	Следующий блок
C	Параметр транзакта, значение которого увеличивается на единицу для каждого транзакта семейства (для транзакта-родителя на 1, для первого потомка семейства на 2 и т.д.).	Имя, число, СЧА	Нет

Примеры.

1. SPLIT 2

Создаются два транзакта-потомка, которые вместе с транзактом-родителем переходят к следующему блоку.

2. SPLIT 3,fork,5

Создаются три транзакта-потомка, которые переходят к блоку с меткой fork. Параметр 5 родителя и потомков будет модифицирован. Если параметр 5 родителя не был определен, то он будет создан и после выполнения блока SPLIT его значение у родителя станет равным 1, у первого потомка – 2, у второго – 3, у третьего – 4.

Блок ASSEMBLE

Назначение. Блок объединяет транзакты одного семейства в один.

Синтаксис. ASSEMBLE A

Операнд	Назначение	Значение	Значение по умолчанию
A	Количество членов семейства, подлежащих объединению	Число, СЧА	Обязательный параметр

Пример.

ASSEMBLE 5

В результате выполнения блока происходит объединение пяти членов каждого ансамбля, члены которых будут входить в данный блок

Блок MATCH

Назначение. Блок синхронизирует движение транзактов одного семейства.

Синтаксис. **MATCH A**

Операнд	Назначение	Значение	Значение по умолчанию
A	Имя или номер блока MATCH, сопряженного с данным.	Имя, число, СЧА	Обязательный параметр

Пример.

PROC1 MATCH PROC2

:
:
:

PROC2 MATCH PROC1

в этом случае два транзакта одного семейства могут пройти через сопряженные блоки MATCH только одновременно.

Останов процесса моделирования

Процедура останова модели включает 3 компонента:

- счетчик останова
- занесение начального значения в счетчик останова
- изменение значения счетчика останова.

Останов моделирования происходит, когда содержимое счетчика останова принимает значение 0.

Счетчик останова – системный параметр. Работа со счетчиком останова выполняется с помощью оператора START и блока TERMINATE.

Оператор START завершает описание программы моделирования и в параметре A содержит начальное значение счетчика останова.

Изменение содержимого счетчика останова выполняется блоком TERMINATE, содержащим параметр A. При поступлении активного транзакта на блок TERMINATE происходит не только его уничтожение, но и выполняется вычитание из счетчика останова содержимого параметра A блока TERMINATE.

Пример управления окончанием моделирования по числу транзактов:

```

GENERATE      1000
<сегмент модели>
TERMINATE    1
START        20

```

Моделирование завершится, когда через сегмент модели пройдет 20 транзактов. Эта величина первоначально задается оператором START и уменьшается каждый раз при входе транзакта в блок TERMINATE. После поступления 20 транзактов на блок TERMINATE счетчик останова станет равным 0 и моделирование завершится.

Пример управления окончанием моделирования по времени:

```
GENERATE      15,5
<сегмент модели>
TERMINATE
```

```
GENERATE      720
TERMINATE      1
START          1
```

Основной сегмент модели содержит блоки TERMINATE, у которых отсутствует параметр A. Для останова по времени создается отдельный сегмент, содержащий указанные блоки. Моделирование завершится, когда модельное время будет равно 720 единицам модельного времени, поскольку именно в этот момент времени из генератора выйдет первый транзакт, и счетчик останова обнулится.

ПРИЛОЖЕНИЕ

Список дополнительных блоков и расширений

Блоки работы со списками пользователя

Блок LINK

Назначение. Блок помещает активный транзакт в список пользователя.

Синтаксис. LINK A,B[, C]

Операнд	Назначение	Значение	Значение по умолчанию
A	Список пользователя, в который должен быть помещен входящий в блок транзакт.	Имя, число, СЧА	Обязательный параметр
B	Дисциплина помещения нового транзакта в список пользователя.	LIFO, FIFO, СЧА транзакта	Обязательный параметр
C	Имя блока, куда переходит транзакт, если индикатор компоновки списка пользователя находится в выключенном состоянии (сброшен)	Имя, число, СЧА	Нет

Особенности выполнения.

1. При помещении транзакта в список он удаляется из всех других списков, кроме групп транзактов и списков прерываний.
2. Транзакт остается в списке пользователя до тех пор, пока какой-либо другой транзакт не войдет в блок UNLINK и не считает его из списка.
3. Если в операнде B указано FIFO (первым вошел, первым вышел), транзакты помещаются в конец списка.
4. Если в операнде B указано LIFO (последним вошел, первым вышел), вновь прибывшие транзакты помещаются в начало списка.
5. Если в операнде B не используется LIFO или FIFO, то применяется СЧА транзакта - PR, M1 или P. Может быть использована косвенная адресация. Если задано PR, транзакты помещаются в список пользователя в приоритетном порядке. Если задан номер параметра, транзакт помещается в список пользователя позади тех транзактов, значение соответствующего параметра которых меньше, чем у входящего транзакта.
6. Если задан операнд C, то используется флаг «Индикатор компоновки». Если индикатор компоновки списка пользователя выключен(сброшен), блок LINK не поместит транзакт в список. Вместо этого транзакт перейдет к блоку указанному в операнде C, после чего индикатор компоновки будет включен (установлен). Следующие транзакты, входящие в блок LINK, будут помещены в список пользователя. Индикатор компоновки управляется блоками LINK и UNLINK. Он выключается (сбрасывается), когда блок UNLINK определяет, что список пользователя пуст.

Примеры.

1. LINK CHANNEL, FIFO

Транзакт, вошедший в блок LINK, будет добавлен в конец очереди с именем CHANAL.

2. LINK CHANNEL, PR, LABEL_CH

[Оглавление](#)

Если список пользователя CHANNEL пуст (индикатор компоновки списка пользователя выключен), транзакт перейдет к блоку с именем LABEL_CH и индикатор компоновки будет включен. Следующие транзакты, входящие в блок LINK, будут помещены в список пользователя. Индикатор компоновки выключится, когда блок UNLINK определит, что список пользователя пуст.

Блок UNLINK

Назначение. Блок считывает транзакты из списка пользователя.

Синтаксис. UNLINK [X] A,B,[C],[D],[E],[F]

Операнд	Назначение	Значение	Значение по умолчанию
X	Операция сравнения для операндов D и E.	Условная операция отношения	E
A	Список пользователя, из которого будет считан один или несколько транзактов.	Имя, число, СЧА	Обязательный операнд.
B	Блок, куда переходят считанные транзакты.	Имя, число, СЧА	Обязательный операнд
C	Максимальное количество транзактов, которые будут считаны	Имя, число, СЧА, ALL	ALL
D	Определяет условия считывания транзактов из очереди	Имя, число, СЧА, BACK	Нет
E	Используется совместно с операндом D и X для определения условия считывания транзактов из списка	Имя, число, СЧА	Нет
F			Нет

Особенности выполнения.

1. Блок UNLINK считывает транзакты из списка пользователя и направляет их в указанный блок. Транзакты, которые необходимо исключить, можно выбирать, также можно наложить ограничение на количество исключаемых транзактов.
2. Если транзакт входит в блок UNLINK, когда в списке нет транзактов, индикатор компоновки списка пользователя сбрасывается.
3. Можно ограничить число считываемых из списка транзактов с помощью операнда C. Если операнд C опущен, то берется ALL.
4. Если операнды D, E и условный оператор опущены, исключаются все транзакты с начала списка, пока список не будет исчерпан, или не будет достигнут предел исключаемых транзактов (операнд C).
5. Операнд D может быть булевой переменной, номером параметра или словом «BACK».

Если операнд D является булевой переменной, он вычисляется относительно транзакта, находящегося в списке пользователя, и если результат не нулевой, транзакт исключается.

Если в операнде D указано BACK, транзакты исключаются, начиная с конца списка пользователя, пока не будет достигнут предел.

В противном случае операнд вычисляется относительно транзакта, находящегося в списке пользователя, и используется в качестве номера параметра, значение которого возвращается членом списка пользователя, как конечный результат. Это конечное значение сравнивается с результатом вычисления операнда Е.

Если операнд D задает параметр, а Е не используется, параметр транзакта из списка пользователя сравнивается с таким же параметром активного транзакта. Если они равны, транзакт, находящийся в списке, считывается из списка.

6. Операнд Е используется только в том случае, если используется оператор отношения. В этом случае обязательно требуется операнд D. Список пользователя проверяется, начиная с начала. Если условие, заданное оператором отношения, выполняется для операндов D и Е, исключается каждый транзакт, (вплоть до предела (операнд С)). Если в операнде Е используется СЧА транзакта, он вычисляется относительно активного транзакта.
7. оператор отношения используется для определения соотношения между атрибутом транзакта (операнд D) и значением операнда Е. Если отношение выполняется, то транзакт считывается из списка. В качестве оператора отношения могут быть использованы Е, G, GE, L, LE или NE. По умолчанию в качестве оператора отношения используется Е (равно).
8. Операнд F используется для определения блока, куда будет направлен входящий транзакт, в случае, если предел исключения транзактов (операнд С) не может быть достигнут, или из списка пользователя не может быть удален ни один транзакт.

Примеры.

1. UNLINK CHANNEL, LABEL_CH,1

Считывается один транзакт из начала списка CHANNEL и он направляется в блок с именем LABEL_CH. Транзакт, вошедший в UNLINK, переходит к следующему блоку

2. UNLINK BUFFER, FAC_2,1,BACK

Считывается из списка пользователя с именем BUFFER один транзакт с конца списка и он направляется в блок с именем FAC_2.

3. UNLINK E P\$BUF,MET_1,ALL,COND,P\$COND,MET_2

Считываются из списка пользователя, номер которого записан в параметре BUF вошедшего транзакта, и направляются в блок с именем MET_1 все транзакты, содержимое параметра COND которых равно содержимому одноименного параметра вошедшего транзакта. Если таких транзактов в списке нет, то вошедший транзакт будет направлен в блок с именем MET_2, в противном случае - к следующему блоку.

Приборы с прерыванием

Блок PREEMPT

Назначение. Блок позволяет транзакту, в зависимости от условий, заданных в операндах блока, занять прибор, даже если он занят другим транзактом (абсолютный приоритет).

Синтаксис. PREEMPT A,[B],[C],[D],[E]

Операнд	Назначение	Значение	Значение по умолчанию
A	имя занимаемого прибора	Имя,	Обязательный операнд

		число, СЧА	
В	Условия занятия прибора	PR	Режим прерывания
С	Блок, куда должен перейти прерванный транзакт	Имя, число, СЧА	Нет
Д	Параметр прерванного транзакта, в который записывается оставшееся время, если транзакт удаляется из списка будущих событий	Имя, число, СЧА	Нет
Е	Режим удаления	RE	Нет

Особенности выполнения.

1. Если прибор находится в состоянии «не готов к использованию», то транзакт помещается в список задержки прибора в конец своего приоритетного класса.
2. Операнд В задает приоритетный режим (PR) или режим прерывания, если операнд опущен.
3. При работе в приоритетном режиме транзакт, занимающий уже прибор, может быть прерван только транзактом, приоритет которого выше приоритета данного транзакта. Если приоритет входящего транзакта ниже, то он помещается в список задержки в конец своего приоритетного класса.
4. В режиме прерывания, если прибор уже используется, поступивший транзакт помещается в список отложенных прерываний. Транзактам из списка отложенных прерываний право занять прибор предоставляется раньше, чем транзактам из списков прерываний или задержки.
5. Прерванный транзакт теряет управление прибором, но может претендовать на дообслуживание, когда прервавший его транзакт входит в соответствующий блок RETURN(если только не задан операнд Е).
6. Прерванные транзакты помещаются в список прерываний в порядке приоритета.
7. Операнд С задает блок, куда должен попытаться перейти прерванный транзакт в этот же момент модельного времени.
8. Если прерываемый транзакт находится в списке будущих событий(вошел в блоке ADVANCE), то для него выполняются следующие действия:
 - вычисляется остаток времени, в течение которого транзакт должен был находиться в блоке ADVANCE, равный разнице планируемого времени выхода транзакта из блока ADVANCE и текущего значения абсолютного условного времени; Операнд D задает номер параметра прерванного транзакта, куда записывается вычисленный остаток времени. Если такой параметр не существует, то он создается.
 - транзакт удаляется из списка будущих событий;
 - транзакт рассматривается, как находящийся в состоянии прерывания и помещается в список прерывания;
 - счетчик прерываний увеличивается на единицу.
9. Прерываемый транзакт может находиться в списке текущих событий (например, когда блок ADVANCE имеет нулевую задержку). В этом случае удаление транзакта из списка текущих событий и перевод его в состояние прерывания производится не сразу. Сначала устанавливается индикатор состояния прерывания. Транзакт, занимающий прибор, будет обрабатываться интерпретатором как обычно и перейдет в состояние прерывания только тогда, когда оно войдет в блок ADVANCE с ненулевой задержкой.
10. Поле Е задает один из следующих режимов:
 - режим удаления(RE). Задание этого режима означает, что прерванный транзакт более не претендует на пользование прибором. Прерванный транзакт пытается войти в блок, заданный полем С (если в поле Е стоит RE, то должно быть указано и поле С). При ис-

Оглавление

пользовании RE прерванный транзакт не должен входить в блоки RELEASE и RETURN, связанные с прерванным транзактом;

-если режим RE не задан, т.е. поле E - пусто, то прерванный транзакт по возвращении в список текущих событий будет вновь пытаться занять прибор.

11. Прерванный транзакт борется за прибор, даже если он перемещен операндом C (если RE не используется в операнде E). Если прерванный транзакт все еще борется за прибор, то попытка транзакта войти в блок TERMINATE приводит к ошибке. Такой транзакт перед входом в блок TERMINATE должен войти в блок RELEASE или блок RETURN.

12. Транзакт может быть прерван на любом количестве приборов и продолжать циркулировать в модели при выполнении следующих двух условий:

-транзакт вошел в блок ADVANCE с положительным временем;

-транзакту не разрешается покинуть блоки ASSEMBLE, GATHER или MATCH до тех пор, пока в них не войдет заданное число транзактов из того же семейства.

13. Прибор может быть захвачен любое количество раз, но не два раза подряд одним транзактом.

14. При использовании операндов C, D, E следует учитывать следующее:

-при задании полей D и (или) E, поле C также должно быть задано;

-если приоритетный режим не задан (PR в операнде B), то операнды C, D и (или) E игнорируются.

Примеры.

1. PREEMPT UNIT

В режиме прерывания, если прибор UNIT уже занят, поступивший транзакт помещается в список отложенных прерываний

2. PREEMPT UNIT,PR,NODE2,P_Time

В приоритетном режиме, если приоритет поступившего транзакта выше приоритета транзакта, занимающего уже прибор, возникает прерывание. Если приоритет входящего транзакта ниже, то он помещается в список задержки в конец своего приоритетного класса. Прерванный транзакт пытается перейти в блок с меткой NODE2, а остаток времени до окончания обслуживания будет помещен в параметр транзакта с именем P_Time.

Блок RETURN

Назначение. Блок позволяет активному транзакту освободить занятый прибор или исключить транзакт из списка прерываний прибора.

Синтаксис. RETURN A

Операнд	Назначение	Значение	Значение по умолчанию
A	Имя освобождаемого прибора	Имя, число, СЧА	Обязательный операнд

Пример использования.

RETURN WorkStation

Освобождается прибор с именем WorkStation

Расширенные описания блоков

Блок TRANSFER (полное описание)

Назначение. Блок передает транзакт на указанный блок.

Синтаксис. TRANSFER [A],[B],[C],[D]

[Оглавление](#)

Операнд	Назначение	Значение	Значение по умолчанию
A	Режим блока	BOTH, ALL, PICK, FN, P, SBR, SIM, Имя, чис- ло, СЧА	Режим безусловной передачи
B	Номер или метка блока. Номер или имя параметра в режиме P	Имя, число, СЧА	Нет
C	Номер или метка блока. Приращение в режимах FN и P	Имя, число, СЧА	Нет 0 для режима P
D	Приращение номера блока для режима ALL	Имя, число, СЧА	1

Особенности выполнения.

Блок TRANSFER может функционировать в одном из 9 режимов. Операнд A используется для определения режима, в котором функционирует блок. Значение операндов B и C зависит от режима. Если не задан операнд, определяющий блок, куда будет передан транзакт, то используется блок, следующий за блоком TRANSFER.

1. Режим безусловной передачи

Когда операнд A отсутствует, блок TRANSFER функционирует в режиме безусловной передачи. В этом режиме активный транзакт всегда переходит к блоку, заданному в операнде B.

TRANSFER ,NO_SERV

При входе транзакта в блок TRANSFER, он передается в блок с меткой NO_SERV.

2. Режим статистической передачи

В этом режиме активный транзакт переходит к блоку, заданному в операнде C, с вероятностью, заданной в операнде A. Операнд A может быть положительной дробью, меньшей единицы или целым положительным числом. Если операнд A - целое число, оно интерпретируется как доля от тысячи. Альтернативный переход задается в операнде B. Если операнд B пропущен, транзакт переходит к следующему по порядку блоку.

TRANSFER .75,, CHANNEL_2

При входе транзакта в блок TRANSFER, с вероятностью 0.75 он переходит к блоку с именем CHANNEL_2. С вероятностью 0.25 он переходит к следующему по порядку блоку.

3. Режим BOTH (ОБА)

В данном режиме проверяется возможность входа транзакта в блок, метка или номер которого указаны в операнде B. Если транзакту отказано во входе в данный блок, проверяется блок, метка или номер которого указаны в операнде C. Транзакт направляется в первый блок, в который ему будет позволено войти. Если ни один из блоков не принимает транзакт, он остается в блоке TRANSFER до тех пор, пока не сможет войти в один из них.

TRANSFER BOTH, CHANNEL_1, CHANNEL_2

При входе транзакта в блок TRANSFER, проверяется блок с меткой CHANNEL_1. Если транзакт не может войти в него, проверяется блок с меткой CHANNEL_1. Если транзакту отказано во входе и второй блок, он остается в блоке TRANSFER до тех пор, пока не сможет войти в один из блоков.

4. Режим ALL (BCE)

В этом режиме проверяется блок, метка или номер которого указаны в операнде В. Если этот блок не может принять активный транзакт, то последовательно проверяются все блоки до тех пор, пока не будет достигнут блок, метка или номер которого заданы в операнде С, или один из проверенных блоков не примет транзакт до достижения блока, заданного в операнде С. Номер каждого последовательно проверяемого блока вычисляется путем добавления операнда D к номеру ранее проверенного блока. Если операнд D не используется, проверяется каждый блок между блоками, заданными в операндах В и С. Если не используется операнд С, проверяется только один блок. Блоки с большим номером, чем операнд С, не проверяются. Транзакт направляется в первый блок, принявший его. Если блок, принимающий транзакт, отсутствует, транзакт остается в блоке TRANSFER до тех пор, пока не сможет войти в один из блоков.

TRANSFER ALL, CHANNEL_1, CHANNEL_2, 2

Когда транзакт входит в данный блок TRANSFER, проверяется блок с меткой CHANNEL_1. Если транзакт не может войти в него, проверяется каждый блок с номером на 2 больше предыдущего. Если все проверенные блоки отказывают транзакту во входе, проверка заканчивается на блоке с меткой CHANNEL_2 или на блоке перед ним. Если ни один из блоков не принял транзакт, он остается в блоке TRANSFER, пока не сможет покинуть его.

5. Режим PICK (выборочный)

В режиме PICK номер блок для перехода транзакта выбирается случайно из интервала, заданного операндами В и С.

TRANSFER PICK, CHANNEL_1, CHANNEL_2

Когда транзакт входит в этот блок TRANSFER, номер нового блока выбирается случайно в интервале между блоками с именами CHANNEL_1 и CHANNEL_2.

6. Режим FN (функциональный)

В режиме FN номер блок для перехода транзакта выбирается путем вычисления функции, заданной в операнде В, с прибавлением к этому значению необязательного приращения, заданного в операнде С.

TRANSFER FN,Exp,5

Когда транзакт входит в данный блок TRANSFER, номер блок для перехода вычисляется как текущее значение функции с именем Exp плюс 5.

7. Режим P (параметрический)

В параметрическом режиме активный транзакт переходит к блоку, вычисленному, как сумма значения параметра, заданного операндом В, и значения, заданного операндом С.

TRANSFER P,End_Proc,1

Когда транзакт входит в этот блок TRANSFER, он переходит в блок, следующий за блоком, номер которого задан в параметре транзакта с именем End_Proc.

8. Режим SBR (подпрограммный)

В режиме SBR активный транзакт всегда переходит к блоку, заданному в операнде В. Номер блока TRANSFER помещается в параметр, указанный в операнде С.

TRANSFER SBR,Proc, End_Proc

Когда транзакт входит в данный блок TRANSFER, он переходит в блок с меткой Proc. Номер блока TRANSFER помещается в параметр с именем End_Proc.

9. Режим SIM (одновременный)

В режиме SIM активный транзакт переходит к одному из двух местоположений в зависимости от индикатора задержки транзакта. Если индикатор задержки установлен, транзакт переходит к блоку, заданному в операнде C, а если индикатор задержки сброшен, транзакт переходит к блоку, заданному в операнде B.

Индикатор задержки транзакта устанавливается всегда, когда транзакту отказывается во входе в какой-либо блок. Индикатор задержки остается установленным до тех пор, пока транзакт не войдет в блок TRANSFER SIM.

TRANSFER SIM, Nodelay_Place, Delay_Place

Когда транзакт входит в этот блок TRANSFER, он немедленно направляется в блок с меткой Delay_Place, если его индикатор задержки установлен, или в блок с меткой Nodelay_Place, если его индикатор задержки сброшен. После перехода транзакта его индикатор задержки всегда сбрасывается.

Блок SELECT

Назначение. Блок выбирает первый объект определенного типа, который удовлетворяет заданному условию. Номер объекта записывается в параметр активного транзакта.

Синтаксис. SELECT X A,B,C,[D],[E],[F]

Назначение операндов.

Операнд	Назначение	Значение	Значение по умолчанию
X	Определяет условный или логический режим функционирования блока.	Условный оператор или СЧА	Обязательный операнд
A	Номер параметра активного транзакта, в который записывается номер объекта, удовлетворяющего заданному условию	Имя, число, СЧА	Обязательный операнд
B	Минимальный номер объекта данного типа, для которого проверяется заданное условие	Имя, число, СЧА	Обязательный операнд
C	Максимальный номер объекта данного типа, для которого проверяется заданное условие	Имя, число, СЧА	Обязательный операнд
D	Величина для сравнения в режиме отношения. Не используется в режиме выбора максимального или минимального	Имя, число, СЧА	Обязательный операнд для режима отношения
E	СЧА типа объекта для режима отношения и выбора максимального или минимального	СЧА типа объекта	Обязательный операнд для режима отношения и выбора максимального или минимального
F	Определяет блок для перехода транзакта, если объект не выбран. Не используется в режиме выбора максимального или минимального	Имя, число, СЧА	Следующий блок

Примеры.

1. SELECT E 1,10,20,0,F,NO_UNIT

В режиме отношения по стандартному числовому атрибуту F(состояние прибора) среди приборов с номерами от 10 до 20 ищется первый прибор со значением F=0(прибор свободен). Номер найденного прибора записывается в первый параметр транзакта. Если такого прибора нет, то в первый параметр записывается 0 и транзакт переходит в блок с меткой NO_UNIT

2. SELECT MAX 1,5,15,,Q

В режиме выбора максимального элемента по стандартному числовому атрибуту Q(длина очереди) среди очередей с номерами от 5 до 15 ищется очередь с максимальным значением данного атрибута.

3. SELECT SE 3,1,7

В логическом режиме по стандартному логическому атрибуту SE(память пуста) среди памятей с номерами от 1 до 7 ищется первая пустая. Номер найденной памяти записывается в третий параметр транзакта. Если такой памяти нет, то в третий параметр записывается 0.

Оператор FUNCTION (полное описание)

Оператор определяет функцию GPSS, заданную таблично.

Синтаксис. <имя функции> FUNCTION A, B

Имя функции – числовое или символическое имя, обязательный параметр.

A - Аргумент функции; обязательный операнд. Допустимые значения – имя, число, СЧА.

B - Тип функции (одна буква) и количество пар данных в списке данных функции. Обязательный операнд.

Последующие обращения к СЧА типа FN вычисляют функцию и возвращают результат.

Существует несколько типов функций. Тип определяется операндом B команды FUNCTION.

За каждой командой FUNCTION сразу же должен следовать список пар данных, разделенных символами «/». Каждая пара данных определяет значения аргумента X и значения функции Y, разделенные запятой. Списки данных используются для вычисления значения функции по заданным значениям аргумента.

Существует 5 различных типов функций.

1. Функции типа C – непрерывные числовые функции.

В списке данных функций типа C значения X и Y должны быть целочисленными (Integer) или вещественными (Real). Значения X и Y хранятся, как числа с плавающей точкой двойной точности.

Вычисление функции начинается с вычисления аргумента. Далее определяется интервал $(X_i; X_{i+1})$, на котором находится вычисленное значение и на этом интервале выполняется линейная интерполяция двойной точности с использованием соответствующих значений Y_i и Y_{i+1} . Результатом является значение функции двойной точности. Если аргумент попадает за предельные значения области определения функции, возвращается значение функции в ближайшей предельной точке.

Примеры.

1) ART FUNCTION X1, C3
1.1,10.1/20.5,98.7/33.3,889.2

Оператор определяет кусочно-линейную функцию с двумя линейными участками. Если мы обращаемся к функции FN\$ART, то по значению сохраняемой величины X1 вычисляется функция в соответствии с заданной совокупностью точек.

Пример приближенного представления обратного экспоненциального распределения со средним, равным 1.

2) Xpdis FUNCTION RN1, C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2/.75,1.38
.8,1.6/.84,1.83/.88,2.12/.9,2.3/.92,2.52/.94,2.81/.95,2.99/.96,3.2
.97,3.5/.98,3.9/.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8

2. Функции типа D - дискретные функции.

В списках данных функций типа D значения X должны быть целочисленными или вещественными, а значения Y – целочисленными, вещественными или именами.

Функции типа D задают одно и то же значение функции Y[i] для всех значений аргумента $X[i-1] < X \leq X[i]$. Значения X в списке данных функции должны быть неубывающими. Внутренне они сохраняются как числа двойной точности. Когда вычисляется функция, значения X в списке данных функции просматриваются от наименьшего к наибольшему. Когда найдено значение X, которое больше или равно текущему значению аргумента, возвращается соответствующее ему значение Y. Если такое значение X отсутствует, возвращается значение Y или именованная величина, соответствующая самому большому значению X.

Примеры.

1) LIR FUNCTION X\$A2, D5
1.1,6.9/2.1,7/6.33,9.4/7,10/9.9,12.01

2) RAF FUNCTION RN1, D5
0,0/.2,7.2/.4,6.667/.8,9.92/1.0,10

3. Функции типа E - дискретные атрибутивные функции.

В качестве значений Y в списке данных дискретных атрибутивных функций используются стандартные числовые атрибуты.

Функции типа E вычисляются тем же способом, что и функции типа D, за исключением того, что значение функции вычисляется косвенно через стандартный числовой атрибут.

Пример использования.

Edisc FUNCTION P7, E4
1,FR2/2,FR7/3,FR9/4,FR11

Значение функции Edisc равно коэффициенту использования приборов 2, 7, 9, 11 в зависимости от значения аргумента, содержащегося в параметре транзакта с именем 7.

4. Функции типа L - списковые функции.

Функции типа E вычисляются тем же способом, что и функции типа D, за исключением того, что значения X должны начинаться с 1 и увеличиваться на 1 для каждой последующей пары данных.

Если аргумент меньше 1 или превосходит наибольшее заданное значение X, происходит останов по ошибке.

Списковые функции требуют меньшего времени выполнения по сравнению с дискретными.

Пример использования.

Listtype FUNCTION Q\$Barber, L5
1,PAR1/2,PAR2/3,PAR3/4,PAR4/5,PAR5

5. Функции типа M - списковые атрибутивные функции.

Функции типа M вычисляются тем же способом, что и функции типа L, за исключением того, что в качестве значений Y в списке данных используются стандартные числовые атрибуты и значение функции вычисляется косвенно через стандартный числовой атрибут.

Пример использования.

Mlist FUNCTION X\$Name1,M5
1,Q\$Nname1/2,Q\$NameX/3,Q\$Name4/4,Q\$Name6/5,F\$Tan1

Правила использования функций

- Значения X в списке данных функции должны быть неубывающими.
- Спискам данных функций никогда не присваиваются номера строк.
- Все поля команды FUNCTION обязательны.
- Все значения X и Y в списке данных функции обязательны.
- Количество пар данных, указанное в операнде B команды FUNCTION, должно совпадать с количеством пар, разделенных символами «/» в списке данных функции.
- Списки данных функций не имеют полей комментария.
- В списке данных функций за значением X следует запятая, за которой следует значение Y, за ним «/» или перевод строки, затем опять значение X.
- Функции типа C, L и D не могут иметь СЧА в качестве значений Y.
- Функции типа E и M должны содержать СЧА или выражение в качестве значений Y.
- Функции типа L и M не могут иметь случайные аргументы.
- Списки данных функций типа L и M должны иметь последовательно возрастающие значения X, начинающиеся с 1.

Сводный список стандартных атрибутов объектов языка GPSS

Таблица .14 - Стандартные числовые атрибуты (СЧА) объектов GPSS

Типы объектов	Имя СЧА	Назначение
Системные СЧА (атомарные)	C1	Текущее значение условного времени. Автоматически изменяется в модели и устанавливается в 0 управляющими операторами CLEAR или RESET. Вещественное значение.
	AC1	Текущее значение абсолютного времени. Автоматически изменяется в модели. Устанавливается в 0 под действием оператора CLEAR. Вещественное значение.
	TG1	Текущее значение счетчика числа завершений. Целочисленное значение.
	Z1	Размер свободной оперативной памяти в байтах. Целочисленное значение.
Транзакты	P	Значение параметра активного транзакта. Целочисленное, вещественное или строковое значение.
	PR	Приоритет активного транзакта. Целочисленное значение.
	M1	Время пребывания в модели активного транзакта. Равно разности текущего значения абсолютного времени и времени рождения активного транзакта. Вещественное значение.
	MP	Транзитное время пребывания в модели активного транзакта. Равно разности текущего значения абсолютного времени и содержимого параметра активного транзакта. Вещественное значение.
	XN1	Номер активного транзакта. Целочисленное значение.

Блоки	N	Общее число транзактов, которые вошли в блок. Целочисленное значение.
	W	Текущее число транзактов в блоке. Целочисленное значение.
Приборы	F	Состояние прибора. Равно 0, если прибор свободен, и 1 - во всех остальных случаях. Целочисленное значение.
	FI	Флаг прерывания прибора : 1 - если прибор находится в состоянии прерывания, 0 - в противном случае. Целочисленное значение.
	FV	Флаг готовности прибора к использованию : 1 - если готов, 0 - в противном случае. Целочисленное значение.
	FR	Коэффициент использования прибора в долях тысячи. Вещественное значение.
	FC	Общее число занятий прибора. Целочисленное значение.
	FT	Среднее время использования прибора одним занятием. Вещественное значение.
Памяти	S	Текущее содержимое памяти. Целочисленное значение.
	R	Число свободных единиц памяти. Целочисленное значение.
	SR	Коэффициент использования памяти в долях тысячи. Вещественное значение.
	SA	Взвешенное по времени среднее содержимое памяти. Вещественное значение.
	SM	Максимальное содержимое памяти. Целочисленное значение.
	SC	Общее количество использовавшихся элементов памяти. Целочисленное значение.
	ST	Среднее время пребывания транзактов в памяти. Вещественное значение.
	SE	Флаг незанятости памяти. (:1 - свободна, 0 – занята). Целочисленное значение.
	SF	Флаг заполненности памяти : (1 - заполнена, 0 - не заполнена). Целочисленное значение.
	SV	Флаг готовности памяти (:1 - готова, 0 - не готова). Целочисленное значение.
Ключи	LS	Состояние логического ключа: (1 - установлен, 0 - не установлен). Целочисленное значение.
Очереди	Q	Текущая длина очереди. Целочисленное значение.
	QA	Взвешенная по времени средняя длина очереди. Вещественное значение.
	QM	Максимальная длина очереди. Целочисленное значение.
	QC	Общее число входов в очередь. Целочисленное значение.
	QZ	Число нулевых входов в очередь. Целочисленное значение.
	QT	Среднее время пребывания транзактов в очереди (включая нулевые входы). Вещественное значение.
	QX	Среднее время пребывания сообщения в очереди (без нулевых входов). Вещественное значение.
Таблицы	TB	Среднее значение аргументов таблицы. Вещественное значение.
	TC	Общее число аргументов таблицы. Целочисленное значение.

	TD	Вычисленное среднеквадратичное отклонение для аргументов таблицы. Вещественное значение.
Датчики случайных чисел	RN	Число, вычисляемое датчиком случайных чисел (м.б. до 7 датчиков). Датчик генерирует последовательность равномерно распределенных целочисленных случайных чисел в интервале 0 – 999. При использовании датчика в качестве аргумента функции или объекта в переменной значение будет дробью от 0 до 0.999999.
Функции	FN	Вычисленное значение функции . Вещественное значение.
Переменные	V	Вычисленное значение целочисленной переменной или переменной с плавающей точкой. Вещественное значение.
Булевские переменные	BV	Вычисленное значение булевой переменной. Вещественное значение.
Сохраняемые величины	X	Значение сохраняемой величины. Целочисленное, вещественное или строковое значение.
Матрицы сохраняемых величин	MX (a,b)	Содержимое элемента матрицы сохраняемых величин, расположенного в строке a, столбце b. Целочисленное, вещественное или строковое значение.
Числовые группы	GN	Текущее число членов в числовой группе. Целочисленное значение.
Группы транзактов	GT	Текущее число членов в группе транзактов. Целочисленное значение.
Семейства транзактов	A1	Номер семейства активного транзакта. Целочисленное значение.
	MB	Флаг синхронизации : 1 , если транзакт в некотором блоке принадлежит тому же семейству , что и активный транзакт; 0 - в противном случае. Целочисленное значение.
Списки пользователей	CH	Текущее число транзактов в списке пользователя. Целочисленное значение.
	CA	Взвешенное по времени среднее число транзактов в списке пользователя. Вещественное значение.
	CM	Максимальное число транзактов в списке пользователя. Целочисленное значение.
	CC	Общее число входов транзактов в список пользователя. Целочисленное значение.
	CT	Среднее время пребывания транзактов в списке пользователя. Вещественное значение.

Таблица 15 - Стандартные логические атрибуты (СЛА) объектов GPSS

Тип объектов	Имя СЛА	Значение
Приборы	NU	Равен 1, если прибор свободен, иначе 0
	U	Равен 1, если прибор занят, иначе 0
	NI	Равен 1, если прибор не прерван, иначе 0
	I	Равен 1, если прибор прерван, иначе 0
	FV	Равен 1, если прибор доступен, иначе 0
	FNV	Равен 1, если прибор недоступен, иначе 0

Памяти	SE	Равен 1, если память пуста (нулевое содержимое) , иначе 0
	SNE	Равен 1, если память не пуста (ненулевое содержимое) , иначе 0
	SF	Равен 1, если память заполнена, иначе 0
	SNF	Равен 1, если память не заполнена, иначе 0
	SV	Равен 1, если память используется, иначе 0
	SNV	Равен 1, если память не используется, иначе 0
Ключи	LR	Равен 1, если логический ключ "выключен", иначе 0
	LS	Равен 1, если логический ключ "включен", иначе 0

Задания по составлению имитационных программ для самостоятельной проработки

1. ЛВС.

Дано: 1 сервер, 10 рабочих станций (Р/С). Запросы поступают от Р/С на сервер, тот готовит ответ и посылает его на Р/С. Спустя некоторое время обдумывания, Р/С снова посылает запрос на сервер и т.д. (режим замкнутого цикла). Времена обдумывания на Р/С, а также решения на сервере заданы. Сеть обеспечивает в любой момент времени одно и только одно соединение.

Определить: среднее время реакции системы в зависимости от времени решения на сервере (быстродействие сервера).

Законы распределения случайных параметров задать самостоятельно.

2. Справочная служба.

Дано: бригада телефонисток, время ответа у каждой одинаковое, заданы параметры потока звонков от клиентов. Дисциплина: если занято, то клиент уходит.

Определить: какой процент клиентов не обслужен в зависимости от количества телефонисток в бригаде.

Законы распределения случайных параметров задать самостоятельно.

3. Сеть.

Дано: сеть из 2-х вершин, 2-х каналов (по одному в каждую сторону), поток пакетов из каждой вершины пуассоновский. Заданы также:

- пропускные способности каналов,
- распределение времени обработки полученного пакета в вершине для формирования подтверждения,
- значение кредита.

Подтверждение поступает в общую очередь при передаче в соседнюю вершину.

Найти: пропускную способность сети, как функцию от кредита.

Законы распределения случайных параметров задать самостоятельно.

4. Сеть с отказами.

Дано: сеть из 2-х вершин: А и В. Поток пакетов из А в В задан. Пропускная способность канала задана. Задан поток отказов канала и время восстановления отказа.

Найти зависимость времени пребывания пакетов в системе от интенсивности отказов.

Законы распределения случайных параметров задать самостоятельно.

5. Ремонт станков.

Дано : в цеху непрерывно работают 100 станков. Задано распределение времени работоспособности станка до поломки. В цеху есть бригада мастеров по ремонту станков, которая работает с 8 до 18 . Задано распределение времени ремонта станка. Каждый станок ремонтируется одним мастером.

Найти: среднее число работающих станков в цеху в зависимости от численности ремонтной бригады

Законы распределения случайных параметров задать самостоятельно.

6. Сеть из 3-х вершин с отказами.

Дано: сеть из 3-х вершин: А, В, С. Задан поток пакетов из А. Пакеты поступают в В, если канал АВ работает, иначе по каналу АС в С. Заданы пропускные способности каналов. Задан поток отказов на канал АВ и время восстановления отказа. Считаем канал АС безотказным. Пакет, оказавшийся в канале АВ в момент отказа, считается пропавшим.

Найти: коэффициент полезной загрузки канала АВ и число потерянных пакетов в зависимости от интенсивности отказов.

Законы распределения случайных параметров задать самостоятельно.

7. Магазин.

Задан поток покупателей на входе в магазин. В магазине 2 продавщицы. Время обслуживания одного покупателя задано и одинаково для каждой продавщицы.

Сравнить по среднему времени пребывания в магазине покупателей для следующих организационных вариантов:

- две очереди к каждой из продавщиц. При этом покупатели с равной вероятностью занимают очереди;
- одна очередь к двум продавщицам.

Законы распределения случайных параметров задать самостоятельно.

8. ЛВС

Дано: ЛВС из 10 рабочих станций (Р/С) и одного сервера.

Запросы поступают от Р/С на сервер, тот готовит ответ и посылает его на Р/С. Спустя некоторое время обдумывания, Р/С снова посылает запрос на сервер и т.д. (режим замкнутого цикла). Времена обдумывания на Р/С, а также обработки запроса на Р/С (для файл-серверного варианта) заданы. Сеть в любой момент времени допускает лишь одно соединение. Пропускная способность канала задана. Будем считать, что один запрос требует просмотра некоторого фиксированного объема БД.

В зависимости от быстродействия сервера сравнить два варианта организации обработки запроса:

- файл-сервер, когда фиксированный объем БД пересылается из сервера на Р/С и там обрабатывается;
- клиент-сервер, когда запрос обрабатывается на сервере, а ответ отсылается на Р/С. Объемом ответа можно пренебречь.

Быстродействие сервера будем определять относительно быстродействия Р/С: быстрее в 1 раз, в 2 раза и т.д.

Законы распределения случайных параметров задать самостоятельно.

ЛИТЕРАТУРА

1. Шрайбер А. Моделирование на языке GPSS . Издательство Мир .1980 г.
2. Учебное пособие по GPSS World. /Перевод с английского/.-Казань: Изд-во «Мастер Лайн». 2002. -272 с.
3. Томашевский В.,Жданова Е, Имитационное моделирование в среде GPSS.- Киев: Бестселлер, 2003.- 416с.