

Министерство образования Российской Федерации
Дальневосточный государственный технический университет
(ДВПИ им. В. В. Куйбышева)

А. И. Васильев

**Имитационное моделирование
информационных и вычислительных систем
с использованием GPSS**

Рекомендовано Дальневосточным региональным учебно-методическим
центром в качестве учебного пособия для студентов специальности 220200
«Автоматизированные системы обработки информации и управления»
вузов региона

Владивосток
2004

Одобрено методическим советом университета

УДК 681.51.012

Васильев А. И. **Имитационное моделирование информационных и вычислительных систем с использованием GPSS: Учеб. пособие.** – Владивосток: Изд-во ДВГТУ, 2004. – 97 с.

В учебном пособии рассмотрены методы и особенности построения имитационных моделей информационных и вычислительных систем, формализованных в виде систем массового обслуживания, на языке моделирования GPSS.

Пособие предназначено для студентов, обучающихся по направлению “Информатика и вычислительная техника” и специальности “Автоматизированные системы обработки информации и управления”, а также может быть рекомендовано специалистам в области информационных технологий и вычислительных систем.

Учебное пособие печатается с оригинал-макета,
подготовленного автором

Редактор В. В. Сизова
Технический редактор Н. М. Белохонова

Подписано в печать 11.11.03. Формат 60 x 84/16.

Усл. печ. л. 5,58. Уч.-изд. л. 4,12.

Тираж 100 экз. Заказ 009.

Издательство ДВГТУ, 690950, Владивосток, Пушкинская, 10

Типография издательства ДВГТУ, 690950, Владивосток, Пушкинская, 10

ISBN 5-7596-0388-4

© Изд-во ДВГТУ, 2004

ВВЕДЕНИЕ

Настоящая работа предназначена для студентов, обучающихся по направлению "Информатика и вычислительная техника" и специальности 220200 – "Автоматизированные системы обработки информации и управления", в качестве учебного пособия по курсу "Моделирование систем". В ней рассмотрена методика исследования информационных и вычислительных систем, формализованных в виде систем массового обслуживания, на основе имитационных моделей.

Имитационное моделирование является одним из широко используемых методов анализа процессов обработки информации и включает в себя процесс построения модели реальной системы, а также проведения эксперимента на модели с целью изучения поведения системы. При моделировании дискретных систем наибольшее распространение в настоящее время получили инструментальные средства GPSS/PC и GPSS World фирмы Minuteman Software. В настоящей работе в качестве инструмента исследования используется система имитационного моделирования GPSS/PC, основу которой составляет версия языка GPSS, очень близкая языку GPSS V [1,2,3].

В работе рассматривается сокращенный список объектов языка, ориентированный на моделирование систем массового обслуживания. Приведены примеры построения моделей различных систем.

Рассмотрены вопросы планирования имитационных экспериментов, оценки точности моделирования, обработки и анализа результатов моделирования.

Ознакомление с пособием позволит читателю перейти к самостоятельному построению моделей с учетом всех возможностей данной системы моделирования.

1. ОСНОВЫ СТАТИСТИЧЕСКОГО ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

1.1. Основные этапы имитационного моделирования

Процесс имитационного моделирования системы включает следующие *основные этапы*:

- построение математической модели;
- машинная реализация модели;
- получение и интерпретация результатов моделирования.

Содержательное описание процессов, происходящих в исследуемом объекте, и, возможно, существующее их *математическое описание*, представленные в качестве исходной информации для моделирования, носят название *концептуальной модели* [5].

Концептуальная модель предъявляет свои требования к системе моделирования, на которой она будет реализована, и, наоборот, система моделирования оказывает влияние на терминологию, используемую при описании концептуальной модели.

Математическая модель реальной системы является тем формальным описанием объекта, изучение которого возможно математическими методами, в том числе и с помощью имитационного моделирования [1,2].

Переход от концептуальной модели к математической называется *формализацией*.

Процесс построения математической модели включает ряд этапов, при выполнении которых формулируется проблема, анализируются задачи моделирования, определяются требования к исходной информации и организуется ее сбор. На различных этапах моделирования проверяется достоверность модели, составляется техническая документация по отдельным этапам.

Для имитационного моделирования характерно воспроизведение явлений, описываемых математической моделью, с сохранением их логической

структуры, последовательности чередования во времени при помощи средств вычислительной техники. Поэтому не обязательно записывать математическую модель в виде системы уравнений относительно искомых величин. Достаточно описать алгоритмы имитации отдельных подпроцессов, характеризующих систему и алгоритмы связи этих подпроцессов.

Наиболее рационально строить модель функционирования системы по *блочному принципу*. При этом могут быть выделены *три группы* блоков такой модели. Блоки первой группы представляют собой имитацию воздействий внешней среды на систему; блоки второй группы являются собственно моделью процесса функционирования исследуемой системы; блоки третьей группы – вспомогательными и служат для машинной реализации блоков двух первых групп, а также для фиксации и обработки результатов моделирования [1,2,7, 8].

Типовой моделью функционирования элементов информационно-управляющих систем: абонентских пунктов, процессоров, устройств памяти, устройств ввода-вывода является система массового обслуживания (СМО).

В данной работе математическая модель системы задана в виде СМО, и основная задача состоит в машинной реализации этой модели и ее исследовании.

Информационные и вычислительные системы функционируют при действии множества случайных факторов. Поэтому при исследовании таких систем используется *метод статистического моделирования*.

Под *статистическим моделированием* понимают процесс создания и исследования на ЭВМ алгоритмических вероятностных моделей, учитывающих случайный характер внешних воздействий и внутренних законов функционирования объектов моделирования.

Статистическая имитационная модель объекта представляется в виде моделирующей программы для ЭВМ, которую условно можно разделить на три блока (рис. 1).

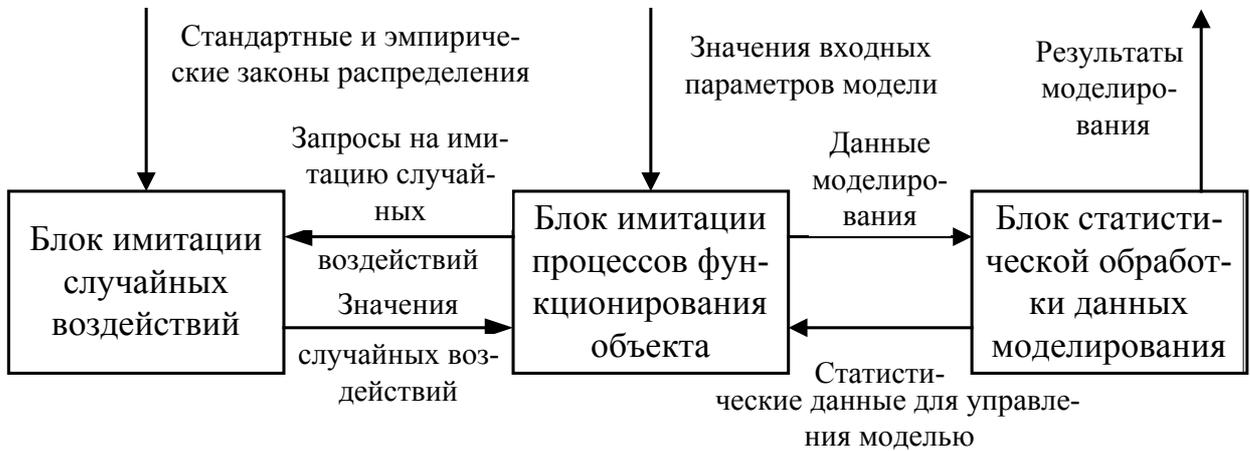


Рис. 1. Схема взаимодействия блоков статистической модели

В блоке имитации процессов функционирования объекта реализуется основная часть *алгоритмической модели*. Здесь воспроизводится логика и динамика взаимодействия элементов модели, организуется отсчет модельного времени, обеспечивается сбор данных моделирования, необходимых для расчета оценок характеристик функционирования объекта.

Блок имитации случайных воздействий служит для воспроизведения параметров случайных явлений и процессов. В его состав входят генераторы базовых случайных величин, а также генераторы случайных воздействий с требуемыми свойствами. В блоке статистической обработки данных моделирования рассчитываются как текущие значения характеристик, так и итоговые характеристики, составляющие результаты экспериментов с моделью.

1.2. Моделирование случайных объектов

Для формализации действия случайных факторов используются статистические объекты: случайные события, случайные величины, случайные вектора и случайные функции [1,2,3].

Для формирования реализаций случайных объектов на ЭВМ используются *случайные числа*.

Обычно используется следующий подход к формированию последовательности случайных чисел с заданным законом: выбирают исходную совокупность равномерно распределенных случайных чисел в интервале (0,1), а затем преобразуют ее в нужную последовательность.

1.2.1. Получение случайных чисел, равномерно распределенных на интервале (0, 1)

В системе моделирования GPSS/PC используются *датчики случайных чисел*, которые позволяют получать случайные числа, равномерно распределенные:

- 1) на интервале (0 – 0,999999), если число является аргументом функции;
- 2) на интервале (0 – 999) в остальных случаях.

Обращение к датчику осуществляется путем указания стандартного числового атрибута RNj, где j – номер датчика .

1.2.2. Моделирование непрерывных случайных величин

При моделировании систем случайными величинами могут быть как временные интервалы (интервалы между моментами поступления заявок, длительность обслуживания), так и численные значения параметров.

Одним из способов получения случайной величины с заданным законом распределения является *метод обратной функции*, который основан на следующей теореме: “Если случайная величина η имеет плотность распределения $f(y)$, то распределение случайной величины $\zeta = \int_{-\infty}^{\eta} f(y)dy$ является

равномерным на интервале [0,1].”

Поскольку функция распределения случайной величины η :

$\Phi(y) = \int_{-\infty}^y f(y)dy$, то для получения случайной величины, принадлежащей

совокупности $[y_i]$, достаточно выполнить преобразование $y_i = \phi^{-1}(x_i)$, где $\phi^{-1}(x_i)$ – функция, обратная $\Phi(y_i)$,

$[x_i]$ -совокупность случайных чисел, равномерно распределенных на интервале $[0,1]$.

Например, если необходимо получить случайную величину, распределенную по *экспоненциальному закону*, $f(y) = \lambda e^{-\lambda y}$, ($y > 0$), то необходимо выполнить преобразование $y_i = \frac{1}{\lambda} [-\ln(1 - x_i)]$.

Если обратная функция не может быть выражена аналитически, то ее можно задать в виде таблицы, а значения функции между точками таблицы вычислять методом линейной интерполяции.

Функции GPSS могут использоваться в модели для вычисления случайных величин, распределенных по определенному закону, а также для формирования значений, зависящих от каких-либо СЧА.

Рассмотрим только дискретные и непрерывные функции. Способ задания этих функций аналогичен. Они задаются в определенных точках в виде пар значений аргумента и функции. При вычислении *непрерывной функции* между точками таблицы осуществляется *линейная интерполяция*.

Для задания функции используются два типа карт: карта определения функции и карты следования функции. Карта определения функции задается следующим образом:

Имя FUNCTION A, B

Имя может быть символическое или числовое. В поле **A** задается СЧА – аргумент функции. В поле **B** – тип функции и число пар точек в виде **Dn** – дискретная, **Cn** – непрерывная, **n** – число пар точек, задаваемых картами следования.

Данные на картах следования располагаются следующим образом, начиная с первой позиции:

X1, Y1 / X2, Y2 / ... / Xn, Yn

Данные могут занимать несколько карт при условии, что каждая карта заканчивается парой значений.

Функция для моделирования случайной величины, распределенной по экспоненциальному закону с $\lambda = 1$, имеет имя XPDIS и задается в 24 точках:

```
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
```

Умножение на множитель $\frac{1}{\lambda}$ можно выполнить в блоках GENERATE, ADVANCE,.

Действительная переменная FVARIABLE и функция LOG (см. 2.1.5) могут быть использованы для получения случайных величин с показательным законом распределения. Для $\lambda = 0,01$ получим переменную V\$TARR, распределенную по экспоненциальному закону:

```
TARR  FVARIABLE  - 100 # LOG (1 - RN1/1000).
```

Обратная функция для получения нормированной случайной величины, распределенной по *нормальному закону* с математическим ожиданием $M = 0$ и дисперсией $\sigma^2 = 1$:

```
SNORM  FUNCTION  RN2, C25
0,-5/.00003,-4/.00135,-3/.00621,-2.5/.02275,-2/.06681,-1.5
.11507,-1.2/.15866,-1/.21186,8/.27425,-.6/.34458,-.4/.42074,-.2
.5,0/.57926,.2/.65542,.4/.72575,.6/.78814,.8/.84134,1/.88493,1.2
.93319,1.5/.97725,2/.9939,2.5/.99865,3/.99997,4/1,5
```

Для получения нормально распределенной случайной величины X с параметрами σ_x и M_x достаточно выполнить преобразование $X = \sigma_x \cdot \text{FN\$NORM} + M_x$.

В этом случае необходимо использовать арифметическую переменную (см. 2.1.5). Арифметическая переменная определяется картой FVARIABLE.

Предположим $\sigma_x = 20$, $M_x = 100$. Тогда карта

NORM FVARIABLE 20#FN\\$NORM+100

определит переменную V\\$NORM, распределенную по требуемому закону.

1.2.3. Моделирование случайных событий и дискретных случайных величин

Для моделирования *случайного события* A , наступающего с заданной вероятностью P , с помощью равномерно распределенных в интервале $(0,1)$ случайных чисел можно использовать следующую процедуру:

- 1) выбирается очередное случайное число X_i ;
- 2) если число $X_i \leq P$, то событие A наступило, в противном случае не наступило.

Действительно, если случайная величина ξ имеет плотность распреде-

ления $f(x) = \begin{cases} 1; 0 \leq x \leq 1 \\ 0; x < 0; x > 1 \end{cases}$, то $P[\xi \leq P] = \int_0^P f(x)dx$. Откуда следует, что

$$P[A] = P[\xi \leq P].$$

Изложенная процедура может быть распространена на группу событий. Пусть A_1, A_2, \dots, A_S - полная группа событий, наступающих с вероятностями P_1, P_2, \dots, P_S . В этом случае $P_1 + P_2 + \dots + P_S = 1$. Событие A_m определяется как событие, состоящее в том, что выбранное значение X_i случайной величины

ξ удовлетворяет неравенству $l_{m-1} < X_i < l_m$, где $l_m = \sum_{i=1}^m P_i$. Процедура моде-

лирования испытаний в этом случае состоит в последовательном сравнении

случайных чисел X_i с величинами l_m . Аналогичная процедура используется при моделировании дискретной случайной величины.

В GPSS для задания возможных значений дискретной случайной величины используется функция. При этом, если функция описана как дискретная, то в случае попадания аргумента между точками таблицы значение берется равным значению в правом конце интервала.

Например, требуется получить дискретную случайную величину

$$X_i = \left(\begin{array}{ccc} 1 & 2 & 3 \\ P_1 & P_2 & P_3 \end{array} \right), P_1 = 0,3; P_2 = 0,2; P_3 = 0,5.$$

Зададим дискретную функцию

```
DIS FUNCTION RN2,D3
.3,1/.5,2/1,3
```

В зависимости от значения случайного числа RN2 будет определено одно из значений дискретной величины X_i . Аналогичная процедура используется при моделировании случайных событий.

В системе моделирования GPSS World [8] для формирования случайных величин наряду с рассмотренными методами могут использоваться стандартные процедуры, написанные на языке PLUS (Приложения 3, 4).

1.3. Сбор и статистическая обработка данных моделирования

Данными моделирования называют первичную информацию о состоянии элементов модели, а также информацию о процессах или событиях определенного типа, необходимую для определения показателей функционирования исследуемого объекта. При статистическом моделировании эти показатели имеют вероятностный характер, и для их расчета применяют статистические методы [1,2, 3,4].

Для сбора данных моделирования используют переменные и операторы (счетчики событий, накопители сумм и т. п.), которые включаются в состав основной части моделирующей программы. В системе моделирования

GPSS/PC данные накапливаются и обрабатываются автоматически при использовании определенных объектов.

При моделировании дискретных систем расчет показателей функционирования сводится к одной из следующих схем:

- оценка вероятности события;
- построение эмпирической функции распределения;
- оценка математического ожидания, дисперсии, корреляционных моментов;
- оценка временных интегралов и характеристик использования ресурсов.

Оценка вероятности события. В модели определяются две переменных-счетчиков: N – счетчик числа испытаний; m – счетчик числа случаев наступления события A в N испытаниях. Для оценки вероятности некоторого события A используется частота события

$$\bar{P}(A) = \frac{m}{N}.$$

Аналогично оцениваются вероятности распределения дискретной случайной величины X , которая может принимать n различных значений

a_1, a_2, \dots, a_n . В модели определяются n счетчиков m_k ($k=\overline{1, n}$), каждый из которых подсчитывает число событий $X=a_k$.

Тогда, соответствующие частоты определяются соотношением

$$\bar{P}(X=a_k) = \frac{m_k}{\sum_{i=1}^n m_i}, \quad k=\overline{1, n}$$

Построение эмпирической функции распределения. Для исследуемой непрерывной случайной величины X диапазон ее возможных значений разбивается на n интервалов. В результате N испытаний можно зафиксировать для каждого интервала m_k ($k=\overline{1, n}$) событий, состоящих в том, что случайная величина принимает значение из этого интервала.

Для оценки вероятности попадания случайной величины в интервал с номером k используется частота

$$\overline{P}_k = \frac{m_k}{N}.$$

По полученным частотам \overline{P}_k строится гистограмма распределения случайной величины X и кумулятивная кусочно-линейная функция распределения, значения которой на границах интервала равны

$$\overline{F}_k = \sum_{i=1}^k \overline{P}_i, \quad k=1, n.$$

Оценка математического ожидания. Для оценки математического ожидания случайной величины используется среднее арифметическое

$$\overline{X} = \frac{1}{N} \sum_{i=1}^N X_i,$$

где X_i - значение случайной величины в i -м испытании.

Оценка дисперсии. Для оценки дисперсии используется статистическая дисперсия

$$S^2 = \frac{1}{N} \sum_{i=1}^N (X_i - \overline{X})^2. \quad (1.1)$$

На практике удобнее пользоваться следующей формулой, полученной из соотношения (1.1):

$$S^2 = \frac{1}{N} \sum_{i=1}^N X_i^2 - \overline{X}^2.$$

Оценка корреляционного момента. Для оценки корреляционного момента двух зависимых случайных величин используется выражение

$$\overline{K} = \frac{1}{N} \sum_{i=1}^N (X_i - \overline{X})(Y_i - \overline{Y}).$$

Его можно преобразовать к более удобному виду

$$\overline{K} = \frac{1}{N} \sum_{i=1}^N X_i \cdot Y_i - \overline{X} \cdot \overline{Y},$$

где X_i, Y_i – значения случайных величин; \bar{X}, \bar{Y} – оценки математических ожиданий случайных величин.

Оценка временных интегралов и характеристик использования ресурсов. В моделях дискретных систем часто используются элементы двух типов:

- а) элементы, аналогичные обслуживающим приборам в системах массового обслуживания (СМО), т. е. имеющие два возможных состояния: “свободен” и “занят”;
- б) элементы, аналогичные очередям в СМО (накопители, устройства памяти), т.е. характеризуемые уровнем использования ресурса.

Временные диаграммы функционирования таких элементов приведены на рис.2.

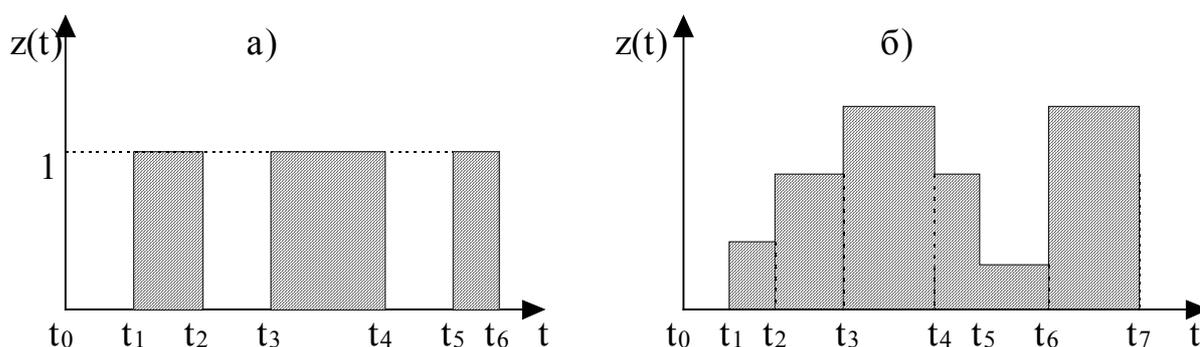


Рис 2. Диаграммы функционирования дискретных элементов:
а) – обслуживающих приборов; б) накопителей

Особенностью рассматриваемых элементов является то, что характеристика их состояния $z(t)$ изменяется скачком и остается постоянной в течение интервалов времени $t_i - t_{i-1}$ ($i=1,2,3,\dots$).

Площади заштрихованных фигур на рис. 2 называют *временными интегралами*, которые рассчитываются в процессе моделирования. Для этого каждому элементу ставится в соответствие две переменные: t_c - момент последнего перехода в новое состояние; $z(t)$ - значение характеристики состояния до текущего момента времени t . Если в момент t элемент изменяет свое

состояние, то значение временного интеграла в момент t получает приращение $I:=I+z*(t-t_c)$, переопределяется значение времени $t_c= t$ и значение характеристики $z(t)$.

Для элементов первого типа временной интеграл I численно равен суммарному периоду занятости, поэтому он используется для расчета коэффициента использования (загрузки) элемента:

$$k_1 = \frac{I(t)}{t}.$$

Для элементов второго типа могут быть определены следующие характеристики:

- среднее число единиц используемого ресурса: $k_2 = \frac{I(t)}{t};$

- коэффициент использования ресурса: $k_3 = \frac{k_2}{z_{\max}} = \frac{I(t)}{t \cdot z_{\max}},$

где z_{\max} – наибольшее допустимое значение $z(t)$, т. е. количество имеющегося ресурса;

- среднее время использования единицы ресурса : $k_4 = \frac{I(t)}{i_1(t)},$

где $i_1(t)$ - суммарное количество занимавшихся единиц ресурса за время t .

1.4. Определение длительности моделирования и количества реализаций

1.4.1 . Типы процессов функционирования систем

Процесс функционирования большинства реальных систем можно разделить на три характерных периода (рис. 3):

$(0, t_1)$ – период от момента запуска системы до момента вхождения в устойчивый (стационарный) режим работы (переходный процесс);

(t_1, t_2) – период функционирования в стационарном режиме;

(t_2, t_3) – период завершения функционирования.

Переходный процесс $(0, t_1)$ возникает из-за того, что начальное состояние элементов системы не соответствует наиболее вероятным состояниям стационарного режима работы.

Период (t_1, t_2) для многих систем является наиболее характерным режимом работы и имеет относительно большую продолжительность.

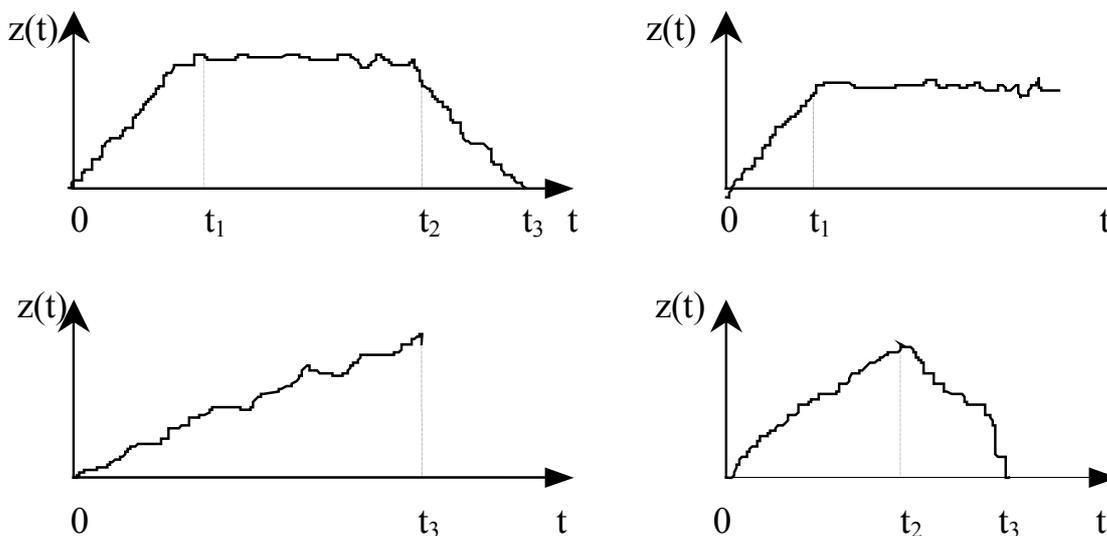


Рис. 3. Изменение характеристик состояния для систем различного типа

В системах с *естественным остановом* этот период ограничен моментом t_2 , который может соответствовать моменту поступления последней заявки в систему массового обслуживания, продолжающую обслуживание всех ранее поступивших заявок до момента t_3 . Естественный останов возникает при завершении выполнения системой заданных функций. В системах без естественного останова продолжительность периода (t_1, t_2) может быть не ограничена.

Методами статистического моделирования можно исследовать все приведенные типы систем. Целью моделирования может являться оценка характеристик функционирования следующих типов:

- усредненных характеристик стационарного режима работы на интервале (t_1, t_2) ;
- временных или интегральных характеристик переходного процесса на интервале $(0, t_1)$ или (t_2, t_3) ;
- временных или интегральных характеристик полного периода функционирования до естественного останова $(0, t_3)$.

Характеристики стационарного режима оцениваются методами статистического моделирования наиболее часто.

1.4.2.. Оценка количества реализаций и точности

Оценка количества реализаций при моделировании может быть проведена на основе методов математической статистики.

Рассмотрим точность оценки вероятности события и математического ожидания с помощью частоты события и среднего арифметического.

Метод статистических испытаний основан на предельных теоремах теории вероятностей, утверждающих, что при большом числе опытов N частота события приближается к его вероятности, а среднее арифметическое – к математическому ожиданию.

Согласно центральной предельной теореме теории вероятностей при большом числе опытов N их средний результат распределяется приближенно по нормальному закону.

Тогда можно считать, что частота события $\bar{P} = m/N$ распределена нормально с математическим ожиданием $M[\bar{P}] = P$ и дисперсией $D[\bar{P}] = P(1-P)/N$. Среднее арифметическое \bar{X} распределено нормально с математическим ожиданием $M[\bar{X}] = A$ и дисперсией $D[\bar{X}] = \sigma_X^2 / N$, где σ_X^2 – дисперсия случайной величины X .

Воспользуемся следующим соотношением:

$$P [|\bar{a} - a| < \varepsilon] = Q, \quad (1.2)$$

где \bar{a} – статистическая оценка рассматриваемой величины; a – математическое ожидание; Q – доверительная вероятность.

Для нормального распределения:

$$Q = 2 \Phi(t); \quad t = \frac{\bar{a} - a}{\sigma}, \quad (1.3)$$

где σ – среднее квадратическое отклонение оцениваемой величины; $\Phi(t)$ – функция Лапласа.

Из соотношений (1.2, 1.3) можно найти точность полученных оценок при заданном количестве реализаций. В результате получим: при оценке ве-

$$\text{роятности } \varepsilon = t_Q \sqrt{P(1-P)/N}; \quad (1.4)$$

$$\text{при оценке математического ожидания } \varepsilon = t_Q \cdot \sigma_X / \sqrt{N}. \quad (1.5)$$

Значения обратной функции Лапласа $t_Q = \Phi^{-1}(Q)$ приведены ниже

Q	0.9	0.95	0.99	0.997
t_Q	1.65	1.96	2.58	3

Для оценки количества реализаций N при заданной точности ε и доверительной вероятности Q необходимо знать значение оцениваемой вероятности P или среднеквадратического значения σ_X . Для этого производят предварительную “пристрелку”, назначая какое-либо число реализацией

$$N_0, \text{ и определяют } P_0 = m_0 / N_0 \text{ или } \sigma_X \approx \sqrt{[\sum_{i=1}^{N_0} X_i^2 - (\sum_{i=1}^{N_0} X_i)^2 / N_0] / N_0}.$$

Затем из формул (1.4) или (1.5) окончательно получают значение N .

Приведенные выше формулы относятся к абсолютной точности. Для количества реализаций N в случае оценки P с относительной точностью $\Delta = \varepsilon / P$ имеем:

$$N = t_Q^2 \frac{1 - P}{P \cdot \Delta^2}. \quad (1.6)$$

При оценке математического ожидания случайной величины с относительной точностью $\Delta = \varepsilon / A$:

$$N = \frac{t_Q^2}{\Delta^2} \left(\frac{\sigma_X}{A} \right).$$

Необходимо обратить внимание на невозможность определения малых вероятностей методом статистического моделирования, вытекающую из соотношения (1.6). Это обстоятельство является принципиальным ограничением метода статистического моделирования.

1.5. Имитационные эксперименты для исследования и оптимизации модели

Имитационным экспериментом называют комплекс прогонов модели, направленных на достижение единой заранее определенной цели [7]. Реализация одного или нескольких параллельных прогонов модели при различных последовательностях псевдослучайных чисел при фиксированных значениях исходных данных называют *моделированием варианта*. На модели могут исследоваться различные варианты структуры системы, значений входных параметров, типов распределений случайных воздействий, начальных условий и т. д. Возможны следующие типы имитационных экспериментов:

Испытательные прогоны

- отладка моделирующей программы;
- проверка адекватности модели;
- определение используемых ресурсов ЭВМ;

Рабочие прогоны

- анализ зависимостей от одного параметра;
- анализ зависимостей от нескольких параметров;
- оптимизация показателей функционирования;
- оценка и выбор вариантов.

Испытательные прогоны модели производятся для ее подготовки к решению основных задач моделирования, которое осуществляется путем реализации *рабочих прогонов* модели.

При отладке моделирующей программы добиваются полного соответствия между собственными представлениями и характером функционирования программы. Для сравнения с результатами ручного счета модель может быть запущена в детерминированном режиме, когда все случайные воздействия модели заменяются константами.

Для проверки адекватности модели можно использовать два подхода:

- 1) прямое сравнение показателей функционирования реального объекта и модели;
- 2) экспертное оценивание модели.

Реализация первого подхода требует сбора большого количества экспериментальных данных о функционировании реального объекта. При обработке этих данных, а также при сравнении выходных показателей модели и реального объекта применяются соответствующие статистические методы. Данный подход нельзя использовать, когда моделированию подлежит проектируемый объект, для которого нет близких аналогов.

Второй подход требует привлечения специалистов - экспертов для оценки адекватности модели. Эксперт должен описать условия функционирования объекта и оценить ожидаемые результаты.

Измерение используемых ресурсов ЭВМ проводят с целью построения эмпирических зависимостей между количеством ресурсов ЭВМ (процессорного времени, объема памяти, объема выводимых данных и т. п.), необходимых для реализации одного прогона модели, и некоторыми параметрами самого прогона (количеством различных событий длительностью прогона количеством моделируемых объектов и т. п.).

Рабочие прогоны модели выполняются с целью решения разнообразных задач анализа и оптимизации (синтеза). Важной количественной характеристикой прогона является требуемое количество реализаций N значений выходного параметра модели. В зависимости от характера моделируемого процесса и способов обработки статистики в качестве отдельной реализации значения выходного параметра модели можно использовать:

- результат каждого отдельного события в модели;
- усредненный результат, полученный по группе событий заданного размера или на отрезке модельного времени заданной длины;
- усредненный результат, полученный на основании данных целого прогона.

Эффективность экспериментов с имитационными моделями зависит от выбора *плана эксперимента*, так как именно план определяет объем и порядок проведения вычислений на ЭВМ, приемы накопления и статистической обработки результатов моделирования.

Математические *методы планирования экспериментов* основаны на представлении процесса в виде следующей модели:

$$Y = \varphi(X), \quad Y = (y_1, y_2, \dots, y_k), \quad X = (x_1, x_2, \dots, x_m),$$

где Y - множество зависимостей выходных переменных, называемых *реакциями*; X - множество векторов входных независимых переменных, называемых *факторами*. Функция φ , связывающая реакцию с факторами, называется *функцией реакции*.

При проведении машинного эксперимента с моделью для оценки характеристик функционирования исследуемой системы необходимо создать такие условия, которые способствовали бы выявлению влияния факторов, находящихся в функциональной связи с искомой характеристикой. Для этого необходимо: отобрать факторы $x_i, i = \overline{1, m}$, влияющие на искомую характеристику; установить диапазон изменения факторов $x_i \min \dots x_i \max$, определить координаты точек факторного пространства $\{x_1, x_2, \dots, x_m\}$, в котором следует проводить эксперимент; оценить необходимое количество реализаций и их порядок в эксперименте.

Получение модели, описывающей реакции изучаемой системы на многофакторное возмущение, - одна из задач математического *планирования эксперимента*. Зависимости вида $Y = \varphi(X)$ можно представить в виде таблиц и графиков. Для аппроксимации этих зависимостей аналитическими выражениями используют методы *регрессионного анализа*.

Задача оптимизации показателей функционирования имитационной модели может быть представлена в виде

$$Q(y_1, y_2, \dots, y_n) \xrightarrow[X \in \bar{X}]{} extr, \quad (1.7)$$

где Q - функционал качества функционирования системы;

$X = \{x_i\}$ - вектор факторов; \bar{X} - область допустимых значений факторов. Для решения задачи (1.7) применяются различные методы оптимиза-

ции. Независимо от того, как организуется выбор наилучшего варианта системы – простым перебором всех проанализированных при экспериментах результатов или с помощью специальных процедур поиска оптимального варианта – элементарной операцией является сравнение статистически усредненных критериев оценки эффективности вариантов систем.

Контрольные вопросы

- 1.1. Охарактеризуйте основные этапы имитационного моделирования.
- 1.2. В чем заключается блочный принцип построения модели?
- 1.3. В чем сущность статистического моделирования на ЭВМ?
- 1.4. Охарактеризуйте структуру статистической имитационной модели.
- 1.5. Какие статистические объекты используются для формализации действия случайных факторов?
- 1.6. Какая последовательность случайных чисел используется в качестве базовой в системе GPSS/PC?
- 1.7. Какие способы моделирования случайных событий, непрерывных и дискретных случайных величин используются в системе GPSS/PC?
- 1.8. Какие показатели функционирования оцениваются при имитационном моделировании?
- 1.10. Каковы особенности имитационного эксперимента на ЭВМ с точки зрения обработки результатов?
- 1.11. Чем определяется длительность моделирования и количество реализаций?
- 1.10. Назовите типы процессов функционирования систем.
- 1.11. Как оценивается точность статистического моделирования?
- 1.12. Что называют имитационным экспериментом?
- 1.13. Назовите типы имитационных экспериментов.
- 1.14. В чем заключается задача математического планирования эксперимента?

2. ПОСТРОЕНИЕ АЛГОРИТМИЧЕСКОЙ МОДЕЛИ НА GPSS/PC

Математическую модель системы необходимо преобразовать в моделирующий алгоритм. Эта работа в значительной степени зависит от средств описания, которыми располагает исследователь.

В качестве такого средства выбрана *система имитационного моделирования GPSS/PC*, основу которой составляет версия языка моделирования GPSS, очень близкая к языку GPSS V [2, 3, 4]. На языке GPSS моделируемая система может быть описана в терминах небольшого набора абстрактных элементов - "объектов". Логические правила, лежащие в основе функционирования системы, могут быть сведены к набору простых операций.

2.1. Краткие сведения о языке моделирования

Язык GPSS построен в предположении, что моделью сложной системы является описание её элементов и логических правил их взаимодействия в процессе функционирования моделируемой системы.

Можно выделить небольшой набор абстрактных элементов, называемых объектами. Причем набор логических правил также ограничен и может быть описан небольшим числом стандартных операций.

Системный язык строится из набора простых объектов, разделяемых на 7 категорий (табл. 1).

Состояния объектов характеризуются величинами, называемыми *Стандартными числовыми атрибутами* (СЧА). СЧА определяют состояние объектов и могут использоваться в качестве переменных в различных блоках модели. Список СЧА приведен в приложении 1.

1. **Динамические объекты** называются *транзактами*. Они "создаются" и "уничтожаются" так, как это нужно в процессе моделирования. С каждым транзактом может быть связано некоторое число "параметров", которые назначаются пользователем для задания характеристик транзактов.

Типы объектов GPSS

Категория GPSS	Типы объектов GPSS
Динамическая	Транзакты
Аппаратная	Устройства (Приборы) Накопители (Многоканальные устройства) Логические переключатели
Статистическая	Очереди Таблицы
Операционная	Блоки
Вычислительная	Переменные: арифметические, булевские Функции
Запоминающая	Ячейки (Сохраняемые величины) Матрицы ячеек (Матрицы сохраняемых величин)
Группирующая	Цепи пользователя Группы

2. **Элементы оборудования** (аппаратная категория), которые управляются транзактами, включают в себя **“Устройства”**, **“Накопители”**, **“Логические переключатели”**. *Устройство* может обслуживать одновременно только один транзакт. *Накопитель* может обслуживать несколько транзактов одновременно, *логический переключатель* является бистабильным индикатором, который, принимая при прохождении одного транзакта состояние "включено" или "выключено", может изменять путь других транзактов.

3. **Статистические объекты**, используемые для оценки поведения системы, двух типов: **"Очереди"** и **"Таблицы"**. Каждой *очереди* соответствует перечень транзактов, задержанных в какой-либо точке системы, и запись длительности этих задержек. *Таблицы* могут использоваться для построения распределений фиксируемых величин.

4. **Операционные объекты**, называемые **"Блоками"**, формируют логику системы, определяя пути продвижения транзактов, их задержку.

5. **Вычислительные объекты** используют для различных вычислений с использованием в качестве переменных СЧА. К этим объектам относятся **"Функции"** и **"Переменные"**.

6. **Запоминающая категория** позволяет сохранять данные в виде отдельных значений или в виде матрицы значений.

7. **Группирующая категория** позволяет объединять отдельные транзакты в группы.

2.1.1. Блок-схемы

Использование *блок-схем* является удобной базой для языка моделирования GPSS. Для этого моделируемая система должна быть представлена в виде последовательности стандартных блоков.

Тогда в процессе моделирования программа будет создавать транзакты, продвигать их через определенные блоки и выполнять действия, соответствующие каждому из этих блоков, т.е. блок-схема показывает путь продвижения транзактов и последовательность их обработки различными блоками,

При построении блок-схемы в первую очередь выделяются логические сегменты системы и блоки, представляющие основные функции, выполняемые системой. После этого производится детализация логики системы добавлением блоков, выполняющих менее важные операции.

Условные обозначения блоков, используемых в языке моделирования GPSS, приведены в приложении 2.

2.1.2. Создание и уничтожение транзактов

Внесение транзактов в модель производится с помощью блока GENERATE (ГЕНЕРИРОВАТЬ). Информация, необходимая для определения случайного интервала между моментами поступления транзактов, задается операндами **A** и **B**. В поле **A** задается среднее значение времени, а в поле **B** - модификатор. При этом аргумент поля **B** может быть задан одним из двух способов:

1. Посредством модификатора интервала. Эта константа определяет интервал, в котором времена распределены равномерно. Любое из $(2B+1)$ целых чисел, заключенных в интервале $(A-B, A+B)$, будут выбираться с вероятно-

стью $1/(2B+1)$.

2. Посредством модификатора-функции. При этом в поле **B** должен быть записан модификатор-функция, а вычисленное значение атрибута, заданного в поле **A**, умножается на значение функции, заданной в поле **B**.

Поле **C** определяет интервал смещения, предшествующий возникновению в блоке первого транзакта.

Поле **D** ограничивает число транзактов, которые должны быть созданы блоком GENERATE. Если поле является пустым, то блок будет генерировать транзакты без ограничения. Уровень приоритета задается в поле **E**.

Каждый транзакт может иметь определенное число параметров, в которые записываются его характеристики. Групповым именем параметра является буква **P**, Например, **P12**, **P\$SLED** означают значения двенадцатого параметра транзакта и параметра с именем **SLED**.

В процессе моделирования параметры транзакта могут изменяться и назначаться при входе в блок ASSIGN. В момент создания транзакта все его параметры равны 0. В поле **A** задается номер или имя параметра. Непосредственно за аргументом поля **A** может не стоять никакого знака или же стоять знак "+" или "-". Эти знаки показывают, как будет использовано значение поля **B**: будет ли оно заменять текущее значение параметра или же будет суммироваться (вычитаться) с текущим значением.

Транзакты удаляются из модели, попадая в блок TERMINATE. В модели может быть любое число блоков TERMINATE. Операнд **A** является указателем уменьшения счетчика завершения. При входе транзакта в блок TERMINATE значение счетчика завершения уменьшается на число, указанное в операнде **A**.

Начальное значение счетчика завершения задается в поле **A** карты START. Как только в процессе моделирования значение счетчика становится равным нулю, моделирование завершается.

2.1.3. Занятие и освобождение приборов. Реализация задержек во времени

В GPSS для моделирования процесса обслуживания транзакта каналами используются специальные элементы: приборы и накопители. Для различия приборов и накопителей, используемых в модели, им присваиваются числовые либо символические имена

В GPSS элементами, которые требуют обслуживания, являются транзакты. Если транзакт должен занять прибор, то он должен войти в блок SEIZE (ЗАНЯТЬ). Если в какой-то момент времени транзакт должен освободить прибор, то он должен войти в блок RELEASE (ОСВОБОДИТЬ). В поле **A** указывается имя занимаемого или освобождаемого прибора. Если прибор занят, то транзакт не может войти в блок SEIZE и ожидает своей очереди, находясь в предыдущем блоке. При этом может образовываться очередь транзактов, ожидающих обслуживания, и обеспечивается дисциплина "Первый пришел - первый обслужен".

Транзакт занимает прибор для того, чтобы немедленно начать обслуживание. В течение времени обслуживания транзакт должен прекратить свое продвижение в модели и только по истечении этого времени войти в блок RELEASE.

Для реализации задержки используется блок ADVANCE (ЗАДЕРЖАТЬ). Информация обслуживания задается в полях **A** и **B** по тем же правилам, что и в блоке GENERATE .

Для моделирования процесса прерывания обслуживания транзактов, занявших прибор (блоки SEIZE- RELEASE), используются блоки PREEMPT (ЗАХВАТИТЬ) и блок RETURN (ВЕРНУТЬ).

Блок FUNAVAIL переводит устройство в состояние недоступности. Недоступность устройства предупреждает занятие или прерывание устройства последующими сообщениями. В поле **A** задается имя устройства. Поля **B**,...,**H** предназначены для задания специальных режимов.

Блок FAVAIL делает устройство доступным. В поле **A** указывается имя устройства.

Для формирования однородных параллельных приборов в GPSS используется специальный элемент – накопитель.

Использование накопителя аналогично использованию прибора. Для занятия накопителя используется блок ENTER (ВОЙТИ), для освобождения – блок LEAVE (ВЫЙТИ). В операнде **A** задается имя накопителя, в поле **B** – число занимаемых или освобождаемых единиц емкости. Если операнд **B** не задан, то емкость изменяется на единицу. Для каждого накопителя с помощью специальной карты задается емкость следующим образом:

```
SERV STORAGE 10
```

Если накопитель имеет числовое имя 1, то для определения его емкости необходимо использовать символическое имя:

```
SERV EQU 1  
SERV STORAGE 10
```

Если накопитель заполнен, то вход транзакту в блок ENTER запрещен, и он будет задержан в предыдущем блоке до момента, пока в накопителе не освободится требуемое число единиц емкости

Кроме блоков ENTER- LEAVE используются блоки SUNAVAIL (сделать накопитель недоступным) и блок SAVAIL (сделать накопитель доступным). В поле **A** указанных блоков задается имя накопителя. В периоды недоступности транзакты не могут войти в накопитель. Однако содержимое накопителя может уменьшаться при вхождении транзактов в блок LEAVE.

Для сбора статистики, описывающей формирование очереди в различных точках модели, используются специальные блоки QUEUE(Встать в очередь) и DEPART (Покинуть очередь). Операнд **A** используется для задания имени очереди. В поле **B** задается количество занимаемых или освобождаемых единиц очереди. Если поле **B** отсутствует, то занимает или освобождается единица очереди.

В GPSS для определения среднего, стандартного отклонения и гистограммы используют специальный блок TABULATE (табулировать).

Таблица определяется с помощью карты TABLE:

Имя таблицы TABLE A,B,C,D

В карте используются операнды **A,B,C,D** :

A – имя табулируемой переменной (СЧА); **B** – левое граничное значение; **C** – ширина всех промежуточных интервалов; **D** – общее число интервалов, включая левый и правый.

В конце моделирования статистика по функционированию приборов, накопителей и статистика по очередям автоматически записывается в файл REPORT.GPS.

2.1.4. Управление потоком транзактов

Для управления потоком транзактов в блок-схеме может использоваться специальный элемент оборудования – ЛОГИЧЕСКИЙ ПЕРЕКЛЮЧАТЕЛЬ и блоки TRANSFER (ПЕРЕДАТЬ), TEST (ПРОВЕРИТЬ), GATE (ВПУСТИТЬ), SELECT (ВЫБРАТЬ).

Для передачи транзакта в блок, отличный от последующего, безусловным образом используется блок TRANSFER (ПЕРЕДАТЬ). Операнд **A** в этом режиме не используется. Операнд **B** определяет номер блока, в который транзакт должен сделать попытку входа.

Например, если определен блок TRANSFER ,COM1, то транзакт будет передан в блок с символическим номером COM1.

Для блока TRANSFER ,FN\$\$LED номер последующего блока, куда переходит транзакт, определяется функцией SLED.

Для изменения направления движения транзактов в зависимости от значений СЧА используется блок TEST.

В этом блоке сравниваются с помощью оператора отношения два СЧА. Оператор отношения (G, GE, E, NE, LE, L), что соответствует

знакам ($>$, \geq , $=$, $<$, \leq , $<$), записывается в специальном поле после слова TEST. В операндах **A** и **B** записываются сравниваемые СЧА. Операнд **C** является необязательным. В этом случае проверка производится в режиме отказа, т.е. транзакт не сможет войти в блок TEST, пока не будет выполнено условие. В случае использования операнда **C** в нем указывается имя блока, куда переходит транзакт в случае невыполнения условия.

Для управления потоком транзактов в зависимости от состояния элементов оборудования используется блок GATE (ВПУСТИТЬ).

Работает блок GATE аналогично блоку TEST. Только вместо операторов отношения используются логические указатели (см. приложение 2). Логический указатель записывается в специальном поле после слова GATE. В поле **A** указывается имя проверяемого элемента (числовое или символическое). В поле **B** – имя блока, куда переходит транзакт при невыполнении условия. Если операнд **B** не используется, блок работает в режиме отказа.

2.1.5. Вычисляемые и запоминающие объекты

К ним относятся переменные, функции и хранимые значения.

Арифметическая переменная в GPSS является СЧА, определяемым пользователем. Ссылка на арифметическую переменную производится с помощью СЧА, Vf или V\$ имя, где f - номер переменной, а "имя" - символическое имя в случае символического определения. Арифметическая переменная определяется с помощью задаваемого пользователем арифметического выражения, являющегося набором СЧА, связанных арифметическими операциями и круглыми скобками. В GPSS/PC используются арифметические операции: + (сложение), - (вычитание), # (умножение), / (деление), \ (целочисленное деление), @ (деление по модулю), при котором результат равен остатку от деления, ^ (возведение в степень). При записи арифметических выражений могут использоваться встроенные функции: ABS(), ATN(), SIN(), COS(), TAN(), SQR(), EXP(), LOG(), INT().

В арифметическом выражении последовательность вычислений определяется приоритетом операций и круглыми скобками.

Арифметическая переменная определяется следующим образом. Имя переменной помещается в поле имени, а в поле операции помещается слово VARIABLE или FVARIABLE для целой или действительной переменной. В поле операндов записывается арифметическое выражение.

Отличие переменных целых и действительных заключается в следующем. При вычислении целых переменных отбрасываются дробные части всех промежуточных результатов. При вычислении действительной переменной действия выполняются с дробными числами и отбрасывается дробная часть только результата. Операция деления по модулю @ для действительной переменной не используется.

Булевские переменные дают возможность принимать решения в зависимости от состояния и значения многих объектов GPSS, используя для этого только один блок. Булевские переменные описываются и используются так же, как и арифметические переменные. Булевская переменная определяется с помощью задаваемого пользователем логического выражения, являющегося набором СЧА, операций отношения: 'L', 'LE', 'G', 'GE', 'E', 'NE', логических указателей, определяющих состояние устройств (F\$ имя, FI\$ имя, FV\$ имя), накопителей (SE\$ имя, SF\$ имя, SV\$ имя), логических ключей (LS\$ имя) и логических операторов ('NOT', 'AND', 'OR'). Булевская переменная принимает значение 1, если логическое выражение ИСТИННО, и 0, если ЛОЖНО. Например, переменная

STOR BVARIABLE 'NOT'F\$CPU'AND'Q\$BUF'LE'5 примет значение 1, если устройство CPU не занято и длина очереди BUF меньше или равна 5.

Хранимыми значениями называются постоянные ячейки памяти, в которые можно обратиться из любого места модели во время моделирования. Они являются СЧА и обозначаются XJ или X\$ имя, где J – число, а "имя" – символическое имя.

Значение СЧА изменяется при входе транзактов в блок SAVEVALUE (СОХРАНИТЬ ВЕЛИЧИНУ). В операнде **A** указывается номер или символическое имя хранимого значения. В поле **B** указывается СЧА, значение которого необходимо сохранить.

Блок SAVEVALUE может использоваться в режиме накопления. Тогда в поле **A** после имени ставится знак + или –.

Матрицы сохраняемых величин дают возможность упорядочить сохраняемые значения в виде матриц $m * n$, где m – число строк, n – число столбцов матрицы. Каждая матрица должна быть перед началом моделирования определена с помощью оператора MATRIX (определить матрицу), имеющего следующий формат:

Имя MATRIX A, B, C.

Поле **A** оператора не используется и сохранено в GPSS/PC для совместимости со старыми версиями GPSS. В полях **B** и **C** указываются соответственно число строк и столбцов матрицы, задаваемые константами, причем общее число элементов, равное произведению B на C , не должно превышать 8191.

Например, оператор MТАВ MATRIX ,10, 2 определяет матрицу с именем MТАВ, содержащую десять строк и два столбца.

В начале моделирования элементы всех определенных матриц устанавливаются равными 0. Для установки отличных от 0 начальных значений отдельных элементов матриц используется оператор INITIAL, имеющий следующий формат:

INITIAL MX \$ имя (a, b), значение или INITIAL MXj (a, b), значение. Здесь имя и j – соответственно имя и номер матрицы; a и b – номера соответственно строки и столбца, задаваемые константами; *значение* – присваиваемое элементу матрицы начальное значение, задаваемое также константой.

Для изменения значений элементов матриц в процессе моделирования служит блок MSAVEVALUE (сохранить значение элемента матрицы), имеющий следующий формат:

Имя MSAVEVALUE A, B, C, D.

В поле **A** указывается имя или номер матрицы, после которого, как и в блоке SAVEVALUE, может стоять знак + или -. В полях **B** и **C** указываются номера соответственно строки и столбца, определяющие изменяемый элемент матрицы. В поле **D** указывается величина, используемая для изменения заданного элемента матрицы.

Например, MSAVEVALUE 5, 3, 2, X1,
MSAVEVALUE MTAB +, P \$ ROW, P \$ COL, 1

Матрицы имеют единственный СЧА с названием MX, ссылка на который записывается в следующем виде MX \$ имя (a, b) или

MXj (a, b). Здесь имя и j - соответственно имя и номер матрицы; a и b - номера соответственно строки и столбца, задаваемые константами или ссылками на СЧА параметров транзактов.

Например:

MX5 (2, 1) ,
MX \$ MTAB (P \$ ROW, P \$ COL)

2.1.6. Косвенная адресация

Иногда при моделировании возникают ситуации, при которых в качестве исходного данного должен использоваться СЧА, номер которого зависит от свойств транзакта, производящего ссылку на этот СЧА.

В GPSS/PC косвенная адресация осуществляется через значения параметров транзактов. При записи между соответствующим СЧА и номером параметра ставится знак *. Например, Q*6 - величина текущего содержимого очереди, номер которой записан в шестом параметре транзакта; FN*3 - значение функции, номер которой записан в третьем параметре транзакта.

2.1.7. Определение характеристик случайной величины

В GPSS/PC для определения среднего, стандартного отклонения и гистограммы используют специальный блок TABULATE.

Таблица определяется с помощью карты TABLE:

№ строки Имя табл. TABLE A,B,C,D,E

В карте используются операнды A,B,C,D:

A – имя табулируемой переменной (СЧА); **B** – верхняя граница первого (левого) частотного интервала; **C** – ширина всех промежуточных интервалов; **D** – общее число интервалов, включая левый и правый.

Например, для описания таблицы SLU TABLE P3,10,5,7 можно дать следующую интерпретацию:



Табулирование требуемой переменной происходит при входе транзакта в блок TABULATE SLU.

Для определения характеристик очередей обычно используют блоки QUEUE – DEPART. Однако часто исследователя интересует не только среднее значение времени ожидания в очереди, но и дисперсия этого времени, а также статистическое распределение выборки времени ожидания, представляемое обычно графически в виде гистограммы. Имея такое распределение, можно оценить вероятность того, что время ожидания превысит или не превысит некоторое заданное значение. Для сбора и обработки данных о выбо-

рочном распределении времени ожидания в очереди служат статистические объекты типа Q-таблица.

Для создания в модели такой таблицы она должна быть предварительно определена с помощью оператора определения QTABLE (Q-таблица), имеющего следующий формат:

имя QTABLE A,B,C,D.

Здесь *имя* – имя таблицы, используемое для ссылок на нее;

A - номер или имя очереди; **B** – верхняя граница первого частотного интервала таблицы; **C** – ширина частотных интервалов; **D** – количество частотных интервалов.

При прохождении транзакта через блоки QUEUE и DEPART его время ожидания фиксируется, и к счетчику частотного интервала таблицы, в который попало это время, добавляется 1. Одновременно в таблице накапливается информация для вычисления среднего значения и среднеквадратического отклонения времени ожидания. По окончании моделирования среднее значение и среднеквадратическое отклонение времени ожидания, а также счетчики попаданий в различные частотные интервалы выводятся в стандартный отчет GPSS/PC.

2.2. Формирование результатов моделирования в GPSS/PC

По завершении *прогона модели* в GPSS/PC производится создание отчета о прогоне модели в неформатированном виде. Отчет записывается в файл со стандартным именем REPORT.GPS. Это имя может быть изменено командой REPORT (**создать отчет**), имеющей следующий формат: >REPORT A:MOD1.GPS, NOW <ENTER> .

Данная команда обеспечит создание *файла с результатами моделирования* MOD1.GPS на дисковом A.

Для обработки файлов с результатами прогона GPSS/PC-модели в неформатированном виде, используется программа GPSSREPT.EXE.

Она вызывается обычной командой MS-DOS. Работа начинается с указания входного файла, содержащего результаты моделирования, подлежащие представлению в форме стандартного вывода GPSS/PC. При этом предполагается, что таким файлом может являться REPORT.GPS. Далее назначается имя выходного файла, в котором результаты будут представлены в форме стандартного вывода.

Отдельные виды объектов языка могут не включаться в состав стандартного вывода. Для этого нужно сделать соответствующие записи в строках файла SETTIGNS.GPS (0 или 1). Для их внесения можно использовать любой текстовый редактор MS-DOS.

Выходной файл, являющийся результатом работы программы GPSSREPT.EXE, можно просматривать на экране дисплея и выводить на печать.

Отчет содержит следующую информацию:

1) *общие сведения* о модели и ее прогоне, включающие модельное время начала (START_TIME) и конца (END_TIME) прогона, количество блоков в модели (BLOCKS), количество устройств (FACILITIES), количество многоканальных устройств (STORAGES), объем памяти, остававшейся свободной при прогоне модели (FREE_MEMORY);

2) *сведения об именах объектов модели*, включающие для каждого имени идентификатор (NAME), присвоенное ему числовое значение (VALUE) и тип имени: 0, если значение имени присвоено пользователем с помощью оператора EQU; 1, если числовое значение имени присвоено системой; 2, если имя является именем блока;

3) *сведения о блоках модели*, включающие для каждого блока номер строки исходной программы (LINE), номер или имя блока (LOC), название блока (BLOCK_TYPE), количество транзактов, прошедших через блок (ENTRY_COUNT), текущее количество транзактов в блоке в момент завершения моделирования (CURRENT_COUNT), количество транзактов, забло-

кированных перед блоком в момент завершения моделирования (RETRY);

4) *сведения об устройствах модели*, включающие для каждого устройства его имя или номер (FACILITY), количество занятий устройства (ENTRIES), коэффициент использования (UTIL.), среднее время на одно занятие (AVE._TIME) и ряд других данных;

5) *сведения о многоканальных устройствах модели (накопителях)*, включающие имя или номер (STORAGE), емкость (CAP.), количество свободных каналов в момент завершения моделирования (REMAIN.), наименьшее (MIN.) и наибольшее (MAX.) количество занятых каналов в процессе моделирования, количество занятий (ENTRIES), среднее количество занятых каналов (AVE.C.), коэффициент использования (UTIL.) и ряд других данных;

6) *сведения об очередях модели*, включающие для каждой очереди ее имя или номер (QUEUE), максимальную длину очереди в процессе моделирования (MAX.), текущую длину очереди в момент завершения моделирования (CONT.), общее количество транзактов, вошедших в очередь в процессе моделирования (ENTRIES), и количество "нулевых" входов в очередь (ENTRIES(0)), среднюю длину очереди (AVE.CONT.), среднее время ожидания в очереди с учетом всех транзактов (AVE.TIME) и без учета "нулевых" входов (AVE.(-0));

7) *сведения о статистических таблицах модели*, включающие для каждой таблицы ее имя или номер (TABLE), среднее значение (MEAN) и среднеквадратическое отклонение (STD.DEV.) табулируемой величины, границы частотных интервалов (RANGE), частоты (FREQUENCY) и накопленные частоты в процентах (CUM.%) попадания наблюдений в эти интервалы;

8) *сведения о списках пользователя модели*, включающие для каждого списка его имя или номер (USER_CHAIN), количество транзактов в списке в момент завершения моделирования (CHAIN_SIZE), среднее количество транзактов в списке (AVE.CONT), общее количество транзактов, вошедших в список в процессе моделирования (ENTRIES), максимальное количество

транзактов, находившихся в списке (MAX), среднее время пребывания транзакта в списке (AVE.TIME);

9) сведения о логических переключателях (ЛП) модели, включающие для каждого ЛП его имя или номер (LOGICSWITCH) и состояние в момент завершения моделирования: 1 - "включен", 0 - "выключен";

10) сведения о сохраняемых величинах модели, включающие для каждой сохраняемой величины ее имя или номер (SAVEVALUE) и значение в момент завершения моделирования (VALUE);

11) сведения о матрицах модели, включающие для каждой матрицы ее имя или номер (MATRIX), а также список всех элементов матрицы в формате: "строка" (ROW), "столбец" (COLUMN), "значение" (VALUE).

Если в операторе START (см. 4.3.) задан вывод в отчет списков текущих и будущих событий, то отчет включает в себя также сведения о транзактах, находившихся в момент завершения моделирования в этих списках. Сведения о транзактах размещаются в отчете в соответствии с размещением транзактов в каждом списке.

Информация о списке текущих событий включает в себя для каждого транзакта его номер (XACT_NUMBER), приоритет (PRI), резидентное время транзакта (M1), номер текущего блока (CURRENT), номер следующего блока (NEXT), перечень всех параметров транзакта в формате: "параметр" (PARAMETER), "значение" (VALUE).

Информация о списке будущих событий включает для каждого транзакта те же данные, однако вместо резидентного времени транзакта (M1) выводится запланированное время выхода транзакта из списка будущих событий (BDT).

Сведения об объектах того или иного типа появляются в отчете только в том случае, если в модели присутствует хотя бы один объект данного типа. В отчетах о прогоне моделей, включающих в себя другие, не рассматривавшиеся здесь объекты GPSS/PC, появляется информация и об этих объектах.

Помимо отчета отдельные результаты моделирования могут быть также выведены в *базу данных* GPSS/PC [6] с помощью команд RESULT. Однофакторный дисперсионный анализ и получение доверительных интервалов для выведенных в базу данных характеристик модели могут быть выполнены с помощью команды ANOVA. Эти команды в данном пособии не рассматриваются.

Контрольные вопросы

- 2.1. Из каких объектов состоит язык моделирования GPSS/PC?
- 2.2. Для чего используются стандартные числовые атрибуты?
- 2.3. Как вводятся в модель и выводятся из модели транзакты?
- 2.4. Какие блоки моделируют работу обслуживающих устройств?
- 2.5. Какие блоки используются для управления потоком транзактов?
- 2.6. Какие объекты используются для вычислений и запоминания в GPSS/PC?
- 2.7. Как определяются при моделировании характеристики случайной величины?
- 2.8. Как формируется в GPSS/PC отчет с результатами моделирования?
- 2.9. Какую информацию содержит отчет с результатами моделирования?

3. ПОСТРОЕНИЕ МОДЕЛЕЙ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ НА GPSS

Рассмотрим СМО, в которую поступает один или несколько потоков заявок. Если в момент поступления заявок каналы заняты, то требования образуют очередь. На длину очереди или на допустимое время ожидания в очереди накладываются ограничения. При этом, если обозначить Tg – допустимое время ожидания в очереди, а Ng – допустимая длина очереди, то можно рассмотреть следующие варианты систем:

1. Системы с ожиданием ($Ng = \infty$, $Tg = \infty$).
2. Системы с отказами ($Ng = 0$, $Tg = 0$).
3. Система с ограниченным ожиданием ($Ng = \infty$, $Tg \geq 0$, $Tg < \infty$).
4. Система с ограниченной длиной очереди $Ng = \text{const}$.

Если длина очереди ограничена (вариант 4), то необходимо задавать дисциплину заполнения очереди, т.е. необходимо определить, что делать с заявкой, застающей очередь заполненной. Из очереди заявки принимаются к обслуживанию в определенной последовательности, т.е. необходимо задать дисциплину обслуживания заявок.

При моделировании многоканальных СМО необходимо различать тип каналов, которые используются для обслуживания требований. Каналы могут быть идентичны, и время использования канала зависит только от типа требований, поступающих в систему, а могут быть различны. В этом случае время использования канала зависит от типа канала.

Кроме того, может различным образом формироваться очередь. Очередь может быть общей ко всем обслуживающим каналам, а может формироваться к каждому каналу отдельно.

В первом случае выбор заявок из очереди производится в соответствии с дисциплиной обслуживания. Во втором случае необходимо задать критерий присоединения к очереди (например, требование присоединять к очереди, имеющей наименьшую длину).

Для моделирования всех перечисленных ситуаций используются различные блоки GPSS .

Если каналы "обезличены" - идентичны, и образуется общая очередь, то для моделирования процесса функционирования можно использовать элемент "накопитель" и блоки ENTER , LEAVE.

Если каждый канал обладает индивидуальными свойствами, или к каждому каналу образуется отдельная очередь, то для моделирования каждого канала используется "прибор" и блоки SEIZE, RELEASE.

Необходимо отметить, что в случае использования "накопителя" мы получаем обобщенную статистику по использованию всех каналов

3.1. Модель одноканальной СМО с ожиданием

Рассмотрим одноканальную СМО, в которую поступает ординарный поток заявок с распределением интервалов между двумя последовательными заявками по экспоненциальному закону с интенсивностью $\lambda = 0,04$. Длина очереди не ограничена, и не накладывается ограничение на время пребывания в очереди. Обслуживаются заявки в порядке поступления ("первый пришел - первый обслужен"). Длительность обслуживания распределена по экспоненциальному закону с интенсивностью $\mu = 0,05$.

Нас интересуют следующие характеристики СМО:

1. Коэффициент использования обслуживающего прибора.
2. Среднее время ожидания в очереди.
3. Средняя длина очереди.

На рис. 4 приведена блок-схема модели. В модели используется прибор с именем FAC1 и очередь с именем BUF. Для получения экспоненциально распределенной величины используется обратная функция $-\text{LN}(1-\text{RNj})$, которая имеет имя XPDIS и задается с помощью специальных карт.

Рассмотрим логику работы модели. Транзакты, имитирующие заявки, вносятся в модель с помощью блока GENERATE. Интервалы времени между последовательными транзактами задаются полями А и В, т. е. определяются соотношением $\frac{1}{\lambda} FN\$XPDIS$. При входе в блок QUEUE происходит присоединение к очереди. Блок SEIZE фиксирует занятие прибора. Блок DEPART

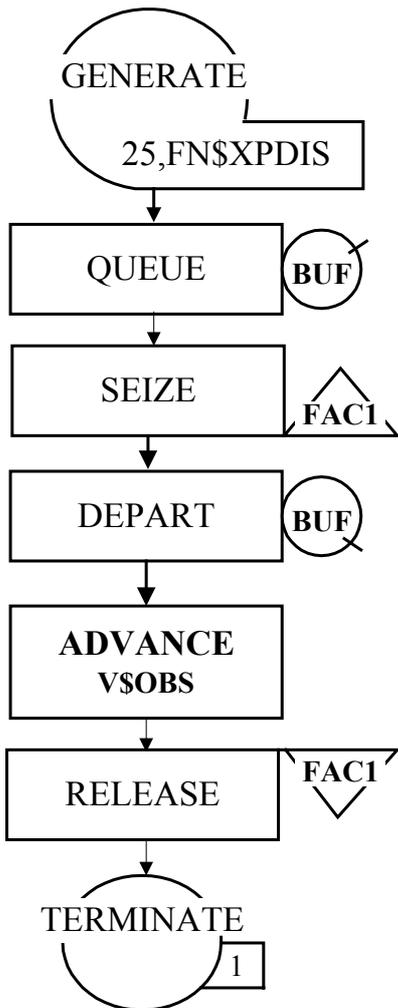


Рис. 4. Блок-схема СМО с обслуживанием заявок в порядке поступления

фиксирует уход одного транзакта из очереди. Напомним, что блоки QUEUE и DEPART не влияют на логику моделирования и используются для сбора статистики очереди. Блок ADVANCE задерживает транзакт на время, необходимое для обслуживания. Это время определяется соотношением $\frac{1}{\mu} FN\$XPDIS$ и задается переменной OBS FVARIABLE 25#FN\$XPDIS. При входе в блок RELEASE фиксируется факт освобождения прибора. Блок TERMINATE выводит транзакт из модели. Единица в поле А блока TERMINATE будет вычитаться из счетчика числа завершений. Если задать карту START 500, то процесс моделирования закончится, когда будет обслужено (войдет в блок TERMINATE) 500 транзактов.

Статистика по использованию прибора и очереди, накапливаемая в процессе моделирования автоматически в файле REPORT.GPS, позволяет получить необходимые характеристики.

3.2. Модель многоканальной СМО с общей очередью

Рассмотрим СМО с несколькими каналами, перед которыми образуется общая очередь. В СМО поступает ординарный поток заявок с распределением интервалов между последовательными заявками по экспоненциальному закону с интенсивностью $\lambda = 0,04$.

Длина очереди ограничена 5 единицами. Поступившая заявка занимает первый свободный канал в порядке нумерации. Если все каналы заняты, заявка присоединяется к очереди. Если очередь заполнена, то заявка получает отказ. Обслуживаются заявки в порядке поступления. Длительность обслуживания не зависит от номера канала и распределена по экспоненциальному закону с интенсивностью $\mu = 0,0125$.

Нас интересуют следующие характеристики СМО:

1. Коэффициенты использования обслуживающих приборов.
2. Средняя длина очереди.
3. Вероятность отказа в обслуживании.
4. Среднее время ожидания в очереди.

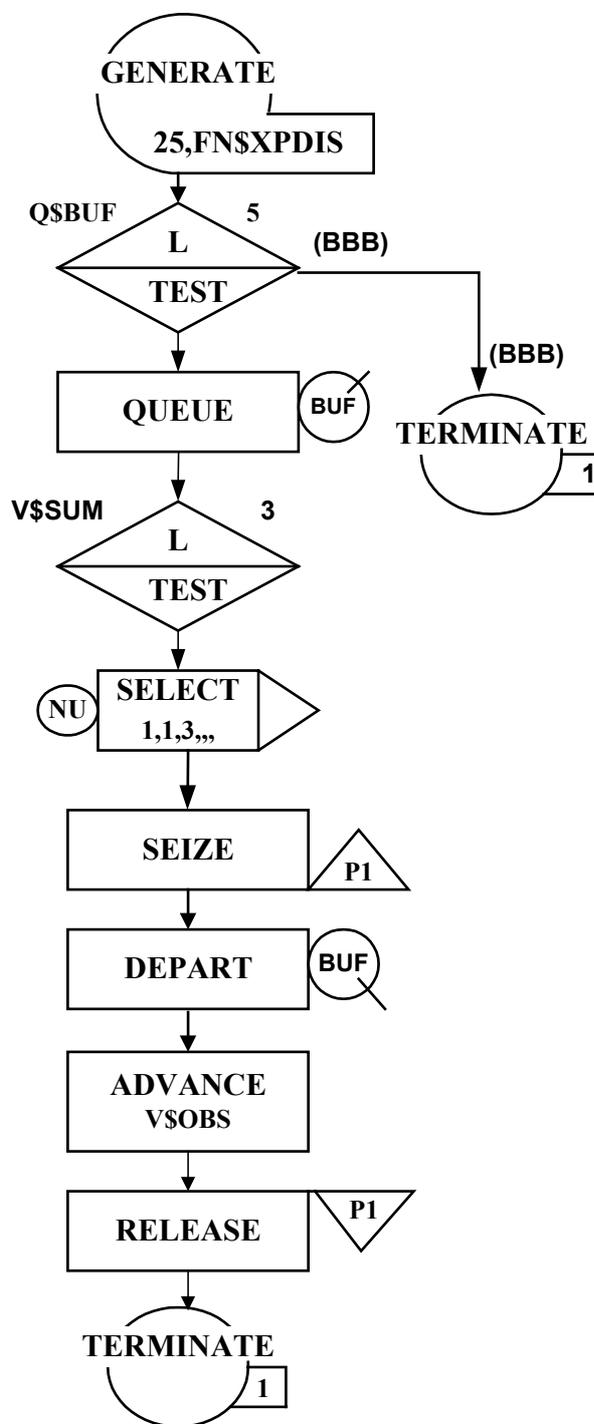


Рис. 5. Модель многоканальной СМО с общей очередью

На рис. 5 приведена блок-схема модели. Рассмотрим логику работы модели.

Транзакты, имитирующие заявки, вводятся в модель с помощью блока GENERATE. Блок TEST L Q\$BUF,5 проверяет условие заполнения очереди. Если очередь заполнена, то транзакт получает отказ (переходит к блоку TERMINATE). Блок QUEUE BUF фиксирует поступление транзактов в очередь. Если все устройства заняты, то транзакт не может пройти через блок TEST L V\$SUM, 3. Переменная SUM FVARIABLE F1+F2+F3 будет равна 3 только в случае занятости всех 3 устройств.

Блок SELECT NU 1,1,3 производит выбор первого свободного канала из трех (поля B и C определяют нижний и верхний пределы изменения номеров каналов) и заносит этот номер в параметр транзакта 1 (поле A).

Длительность обслуживания определяется переменной OBS FVARIABLE 80#FN\$XPDIS.

В блоках TERMINATE, куда входят обслуженные и получившие отказ транзакты, в поле A задана 1. Если задать карту START 1000, то это будет соответствовать поступлению в систему 1000 заявок.

Статистика по использованию прибора и очереди, накапливается процессе моделирования автоматически в файле REPORT.GPS.

3.3. Модель многоканальной СМО с отдельными очередями

Рассмотрим СМО с несколькими каналами, перед которыми образуются очереди. В СМО поступает ординарный поток заявок с распределением интервалов между последовательными заявками по экспоненциальному закону с интенсивностью $\lambda = 0,04$.

Длина каждой очереди ограничена 5 единицами. Поступившая заявка занимает первый свободный канал в порядке нумерации. Если все каналы заняты, заявка присоединяется к самой короткой очереди. Если очередь заполнена, то заявка получает отказ. Обслуживаются заявки в порядке поступ-

ления. Длительность обслуживания не зависит от номера канала и распределена по экспоненциальному закону с интенсивностью $\mu = 0,05$.

Нас интересуют следующие характеристики СМО:

1. Коэффициенты использования обслуживающих приборов.
2. Среднее время ожидания в очередях.
3. Средние длины очередей.
4. Вероятность отказа в обслуживании.

На рис. 6 приведена блок-схема модели. Рассмотрим логику работы модели.

Транзакты, имитирующие заявки, вводятся в модель с помощью блока GENERATE. Блок SELECT NU 1,1,3,,,AAA производит выбор первого свободного канала из трех (поля В и С определяют нижний и верхний пределы изменения номеров каналов) и заносит этот номер в параметр транзакта 1 (поле А). Если все каналы заняты, то транзакт переходит к блоку SELECT, работающему в режиме выбора минимума. В первый параметр транзакта заносится номер очереди с минимальной длиной. В поле Е указывается групповое имя СЧА Q – текущее содержимое очереди. Блок TEST проверяет условие заполнения очереди. При этом ис-

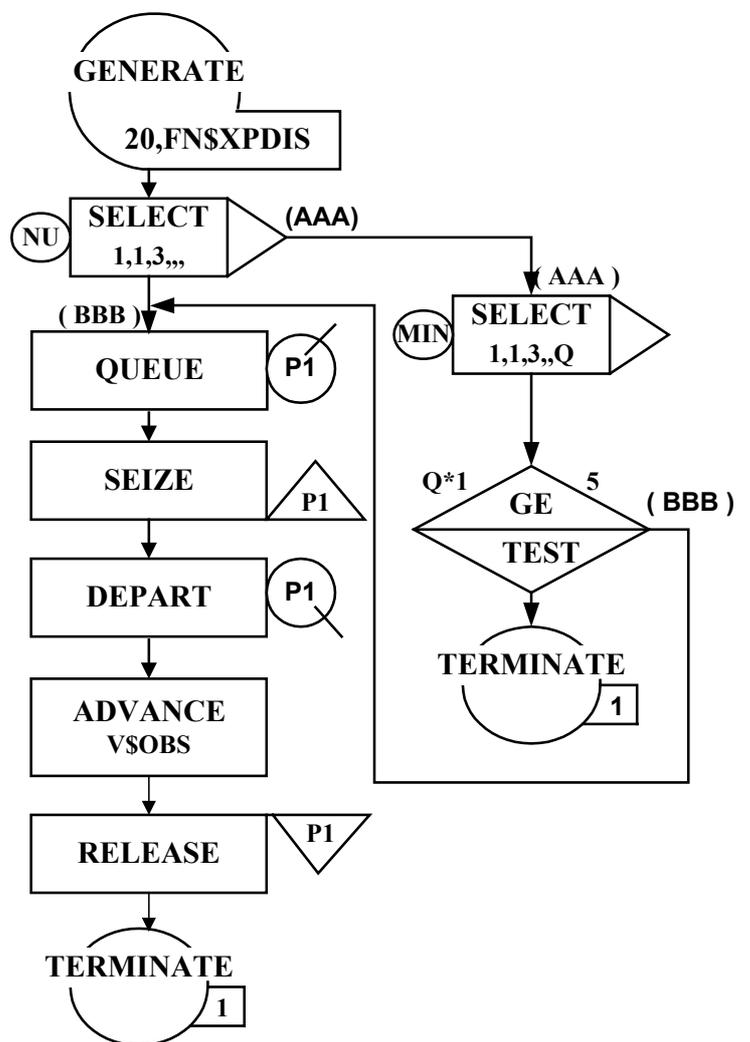


Рис. 6. Модель многоканальной СМО с отдельными очередями

пользуется косвенная адресация очереди Q*1 (проверяется текущее содержимое очереди, номер которой записан в первом параметре транзакта).

В блоках TERMINATE, куда входят обслуженные и получившие отказ транзакты, в поле A задана 1. Если задать карту START 1000, то это будет соответствовать поступлению в систему 1000 заявок.

3.4. Моделирование СМО с различными дисциплинами выбора заявок из очереди

В рассмотренных примерах выбор заявок из очереди осуществляется "в порядке поступления". При использовании рассмотренных ранее блоков изменить дисциплину обслуживания нельзя, поскольку порядок обработки транзактов в цепи текущих событий определяется моментом их поступления и приоритетом.

Для изменения последовательности обработки транзактов их нужно вывести из цепи текущих событий и в нужный момент - в момент начала обслуживания - ввести в цепь текущих событий. Таким элементом, куда могут быть выведены транзакты, является цепь пользователя.

Для ввода в цепь пользователя и вывода из нее используются соответственно блоки LINK (ВВЕСТИ В ЦЕПЬ) и UNLINK (ВЫВЕСТИ ИЗ ЦЕПИ) (приложение 2).

Блок LINK может быть использован в двух режимах: условном и безусловном. Рассмотрим работу блока в безусловном режиме.

В этом случае операнд С не используется. В поле А указывается имя цепи пользователя (числовое или символическое). Поле В определяет, в какое место списка пользователя следует поместить этот транзакт. Если в поле В записано ключевое слово FIFO, то транзакт помещается в конец списка, если LIFO – в начало списка. В других случаях транзакты упорядочиваются в соответствии с вычисленным значением поля В, где обычно записывается один из СЧА транзактов, таких, как PR, M1 или P. Если поле В содержит СЧА PR, то транзакты упорядочиваются по убыванию приоритета. В остальных случа-

ях производится упорядочение по возрастанию указанного СЧА.

Блок UNLINK выводит транзакт из цепи пользователя. Указатели отношения (G, GE, L, LE, E, NE : >, >=, <, <=, =, ≠), записываемые во вспомогательном поле операции (поле X), определяют, какое явное условие рассматривается. Если указатель отношения не задан, предполагается отношение равенства (E). При этом в полях A, B, C, D, E, F может задаваться следующая информация: A – имя цепи пользователя (символическое или числовое); B – имя блока, куда переходят транзакты цепи; C – число выводимых транзактов (константа, СЧА или символ ALL – все); D, E, – описание их см. в табл. 2; F – альтернативный выход; необязателен, по умолчанию транзакт-инициатор вывода переходит в следующий блок.

Таблица 2

Значения операндов D и E блока UNLINK

№ пп	Операнд D	Операнд E	Условие, необходимое для вывода из цепи
1	Пусто	Пусто	Вывод из НАЧАЛА ЦЕПИ
2	BACK	Пусто	Вывод из конца цепи
3	Любой СЧА	Пусто	Пусть в операнде D находится величина j ; транзакт удовлетворяет условию вывода из цепи, если величина его j -го параметра {G, GE, L, LE, E, NE} величины j-го параметра транзакта, инициирующего вывод
4	Любой СЧА	Любой СЧА	Пусть в операнде D находится величина j ; транзакт удовлетворяет условию вывода из цепи, если величина его j -го параметра {G, GE, L, LE, E, NE} величины, находящейся в операнде E
5	BV\$[имя]	Пусто	Транзакты удаляются из цепи в соответствии со значением счетчика (поле C) только при BV\$[имя]=1

3.5. Модель одноканальной СМО с выбором заявок в порядке поступления

Построим модель одноканальной СМО, рассмотренную в параграфе 3.1, используя цепь пользователя BUF (рис. 7). Рассмотрим логику работы модели. В начале моделирования прибор FAC1 свободен, а цепь пользователя

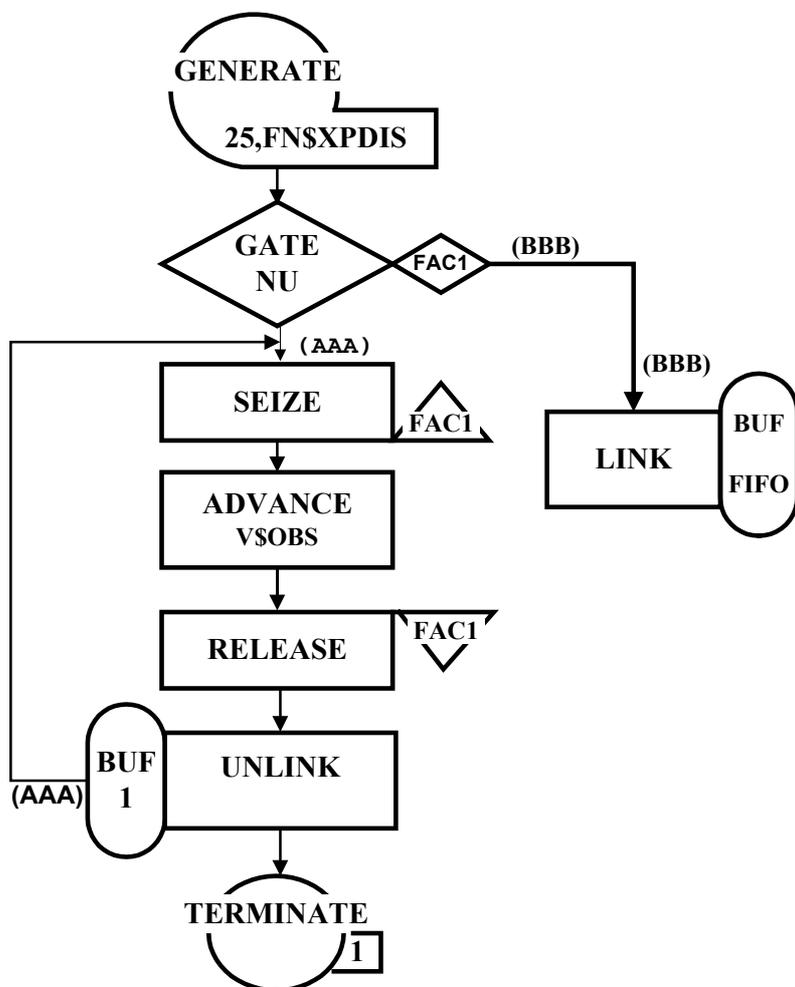


Рис. 7. Модель СМО с обслуживанием заявок в порядке поступления

BUF пуста.

При входе транзакта в блок GATE проверяется условие занятости прибора FAC1. Если прибор свободен, то транзакт переходит в блок SEIZE FAC1, т.е. занимает прибор. Если прибор занят, то транзакт отправляется в конец цепи пользователя.

Как только закончится обслуживание, и транзакт войдет в блок UNLINK, из цепи пользователя с именем BUF (поле A) к блоку

AAA (поле B) будет направлен 1 транзакт (поле C) из начала цепи (поле D и E пусты). Транзакт-инициатор перейдет к блоку TERMINATE и будет выведен из модели, а транзакт, выведенный из цепи, займет прибор. В процессе моделирования формируется статистика цепи пользователя. В рассмотренном примере транзакты помещались в конец цепи пользователя (FIFO), а выводились из начала цепи (пустые поля D и E). Этим обеспечивалась дисциплина "первый пришел – первый обслужен". Эту же дисциплину можно реализо-

вать, если транзакты помещать в начало цепи, а выводить из конца цепи, т.е. блоки изменить следующим образом:

LINK BUF, LIFO; UNLINK BUF, AAA, 1, BACK

3.6. Модель одноканальной СМО с выбором заявок по динамическому приоритету

Незначительная модификация блок-схемы (рис. 7) позволяет реализовать обслуживание по динамическому приоритету (первой обслуживается заявка, имеющая наименьшее время обслуживания)

Очевидно, время обслуживания необходимо определить до блока GATE. На рис. 8 приведена модель такой системы. После ввода транзакта в модель в блоке ASSIGN первому параметру транзакта присваивается длительность обслуживания.

Транзакт в цепи пользователя располагается таким образом: ближе к началу цепи будет находиться транзакт, имеющий меньшее значение первого параметра P1 (в поле В блока LINK задан СЧА – P1). Тогда, если транзакты выводить из начала цепи (пустые поля D, E в блоке UNLINK), то всегда будет выводиться транзакт, имеющий наименьшее время обслуживания. В блоке ADVANCE в поле А задано значение параметра P1, равное времени обслуживания.

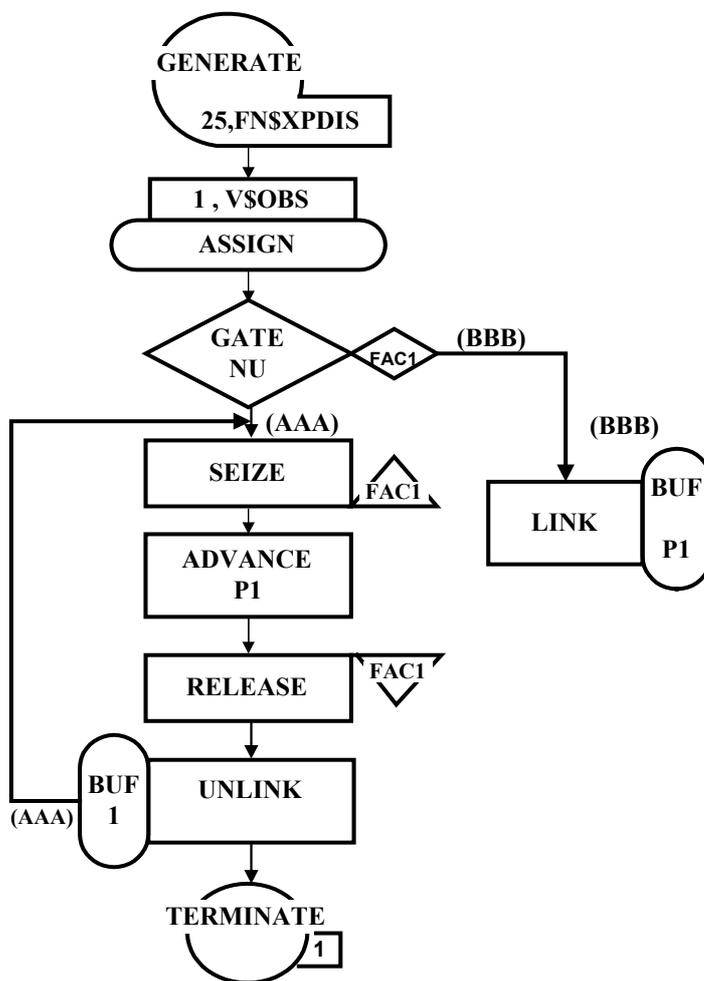


Рис. 8. Модель СМО с обслуживанием заявок по динамическому приоритету

3.7. Модель одноканальной СМО с обслуживанием заявок в случайном порядке

Рассмотрим модель одноканальной СМО, где выбор заявок на обслуживание производится в случайном порядке с равной вероятностью. Отличие от ранее рассмотренных примеров будет заключаться в последовательности вывода транзактов из цепи пользователя.

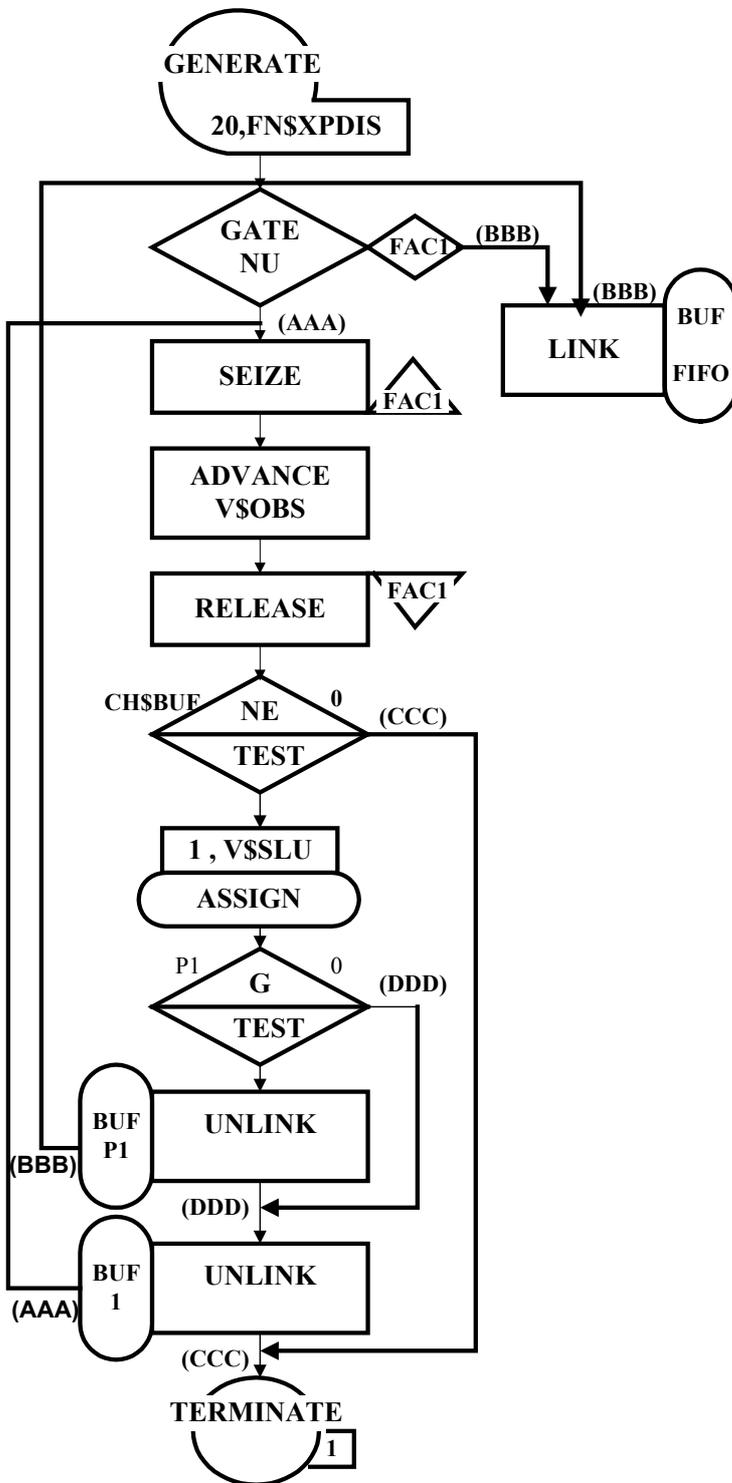


Рис. 9. Модель СМО с обслуживанием заявок в случайном порядке

живание производится в случайном порядке с равной вероятностью. Отличие от ранее рассмотренных примеров будет заключаться в последовательности вывода транзактов из цепи пользователя. Прежде всего необходимо выбрать номер транзакта, который должен обслуживаться. Количество транзактов в цепи пользователя определяет СЧА CH\$BUF (см. приложение 1). Тогда, если выполнить операцию деления по модулю RN1@CH\$BUF, мы получим случайное число, принимающее значения от 0 до (CH\$BUF-1) с равной вероятностью. Например, если в цепи находится 8 транзактов, то с равной вероятностью можем получить числа 0, 1, 2, 3, 4, 5, 6, 7, и вывести определенный транзакт из цепи пользователя в зависимости от их значений.

Определим арифметическую переменную

```
SLU VARIABLE RN1@CH$BUF
```

Рассмотрим логику работы модели (рис. 9). Часть блоков работает аналогично ранее рассмотренным. В блоке ASSIGN первому параметру транзакта присваивается случайное число, с равной вероятностью принимающее значение от 0 до CH\$BUF- 1. Блок TEST проверяет, равно ли это число 0. Если оно равно 0, то из цепи пользователя выводится и направляется на обслуживание транзакт, стоящий в начале цепи. Если это число не равно 0, то из начала цепи в конец цепи перемещается число транзактов, равное P1, и после этого следующий транзакт отправляется на обслуживание.

3.8. Модель одноканальной СМО с ограниченным временем ожидания

Рассмотрим одноканальную СМО, в которой ограничено допустимое время пребывания в очереди. Длительность ожидания – случайная величина, распределенная равномерно со средним 50 с. и разбросом 10 с.

Рассмотрим работу блока SPLIT (РАСЩЕПИТЬ). Этот блок позволяет ввести дополнительные транзакты в модель. Транзакт, входящий в блок SPLIT, называют родителем. Дополнительные транзакты, вводимые в модель, называют потомками. Потомки идентичны родителям: имеют тот же уровень приоритета, то же число параметров, отметка времени потомка совпадает с отметкой времени родителя, значения параметров также идентичны.

В рассматриваемом примере используются операнды А и В блока SPLIT:

А – число дополнительных транзактов, вводимых в модель;

В – имя блока, куда будут направлены эти дополнительные транзакты.

Модель системы при обслуживании заявок в порядке поступления приведена на рис. 10 .

Рассмотрим логику работы модели и подробно остановимся на процессе удаления транзакта из цепи пользователя, если время ожидания истекло.

Если прибор занят, то после ввода в модель транзакт переходит к блоку ASSIGN 2, C1. В этом блоке второму параметру транзакта присваивается значение C1- относительное время (см. приложение 1).

В рассматриваемой модели относительное время совпадает с абсолютным, т.е. в параметре 2 фиксируется момент поступления транзакта. В блоке SPLIT генерируется транзакт-потомок, который переходит в блок ADVANCE 50,10. В этом блоке он задерживается на время допустимого пребывания в очереди.

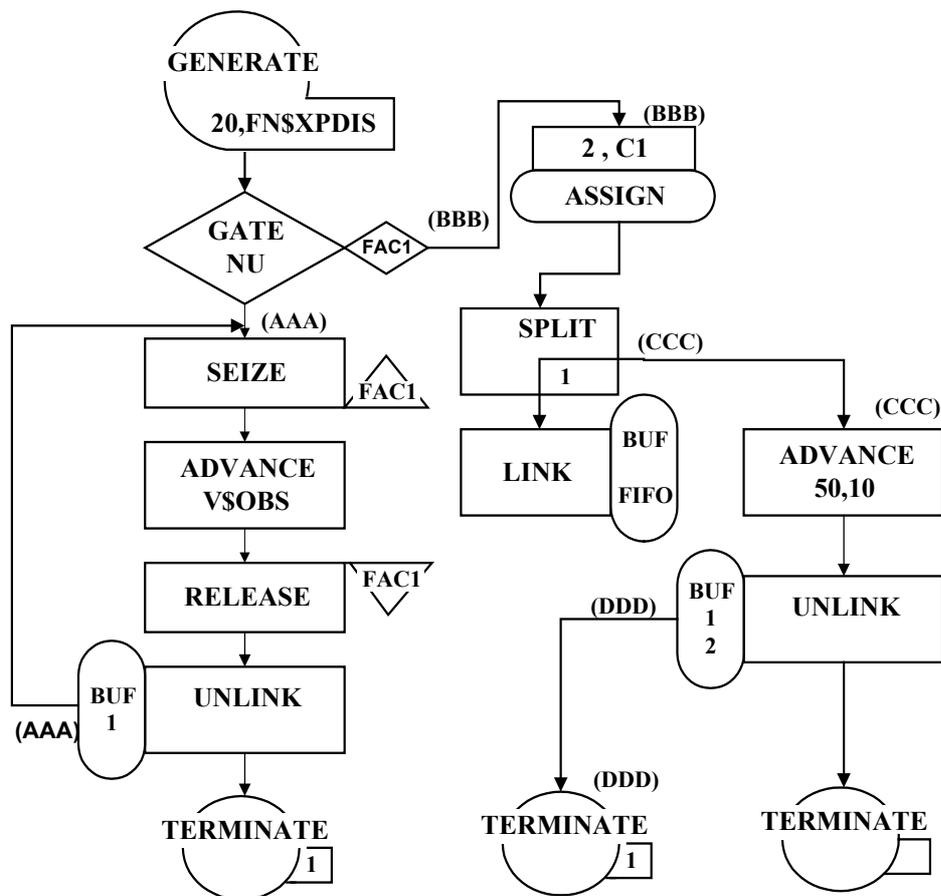


Рис. 10. Блок- схема модели СМО с ограниченным временем ожидания в очереди

Транзакт-родитель переходит в блок LINK. Причем второй параметр родителя и потомка имеет одно и то же значение. После выхода из блока ADVANCE транзакт-потомок входит в блок UNLINK. В поле D стоит величина 2 (см. таблица 1, поз. 3); следовательно, из цепи пользователя должен быть выведен транзакт, имеющий такое же значение второго параметра. А

это – транзакт-родитель. Если он до этого момента не был обслужен, то он будет выведен из цепи. Транзакт-потомок переходит в блок TERMINATE и выводится из модели. В модели на рис. 10 используется три блока TERMINATE. В данном случае это удобно, так как в конце моделирования распечатывается число транзактов, вошедших в каждый блок. И по числу входов в блоки TERMINATE можно подсчитать частоту отказов.

В блоках TERMINATE, куда входят обслуженные и получившие отказ транзакты, в поле А задана 1. Если задать карту START 1000, то это будет соответствовать поступлению в систему 1000 заявок.

3.9. Модель многоканальной СМО с отдельными очередями с использованием цепи пользователя

Построим модель многоканальной СМО с отдельными очередями и обслуживанием заявок в порядке поступления, рассмотренную в параграфе 3.3., используя цепь пользователя. На рис. 11 приведена блок-схема модели. Рассмотрим логику работы модели.

Транзакты, имитирующие заявки, вводятся в модель с помощью блока GENERATE. Блок SELECT NU 1,1,3,,,AAA производит выбор первого свободного канала (поля В и С определяют нижний и

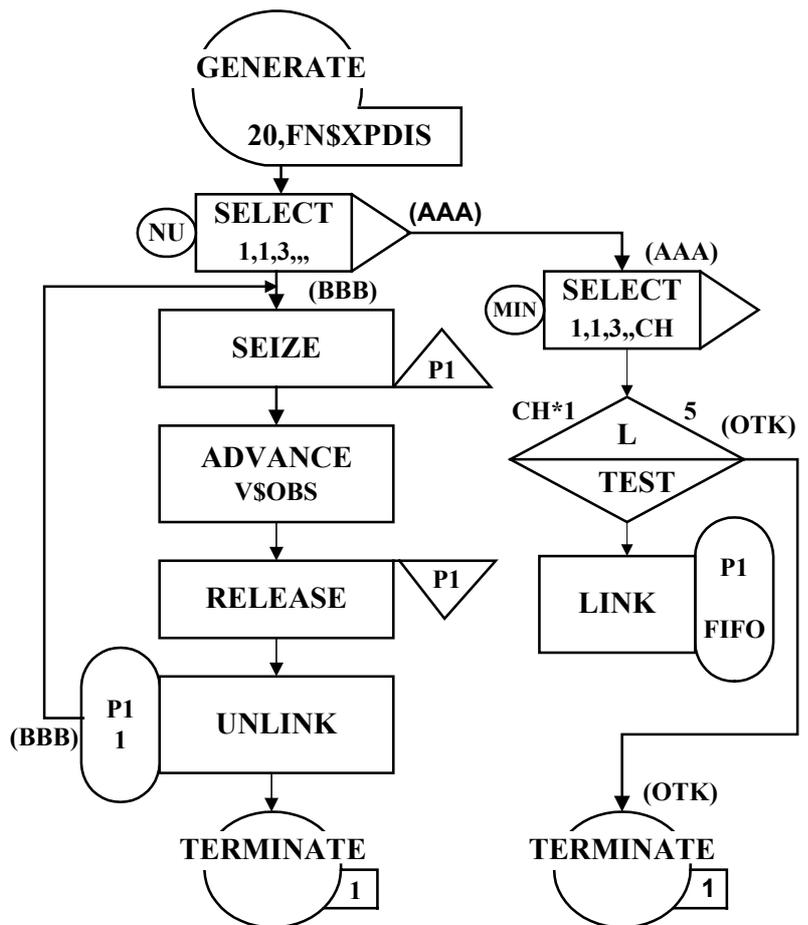


Рис. 11. Модель многоканальной СМО с отдельными очередями

верхний пределы изменения номеров каналов) и заносит этот номер в первый параметр транзакта (поле А). Если все каналы заняты, то транзакт переходит к блоку SELECT, работающему в режиме выбора минимума (см. приложение 2). В первый параметр транзакта заносится номер цепи пользователя с минимальной длиной. В поле Е указывается групповое имя СЧА СН (см. приложение 1) – текущее содержимое цепи пользователя. Блок TEST проверяет условие заполнения очереди. При этом используется косвенная адресация цепи пользователя СН*1 (проверяется текущее содержимое цепи, номер которой записан в первом параметре транзакта). В блоках TERMINATE, куда входят обслуженные и получившие отказ транзакты, в поле А задана 1. Если задать карту START 1000, то это будет соответствовать поступлению в систему 1000 заявок.

3.10. Модель многоканальной СМО с идентичными каналами и общей очередью

В качестве примера рассмотрим следующую задачу.

В СМО через случайные промежутки времени, распределенные по экспоненциальному закону с интенсивностью $\lambda=0,0025$, поступает поток заявок. С вероятностями $P_1=0,3$; $P_2=0,4$; $P_3=0,3$ заявка относится к одному из трех типов. Имеется три канала, которые обслуживают заявки. Длительность обслуживания распределена по нормальному закону с параметрами, зависящими от типа заявки:

1 тип: $M_{об}=50$, $\sigma_{об}=9$;

2 тип: $M_{об}=100$, $\sigma_{об}=15$;

3 тип: $M_{об}=150$, $\sigma_{об}=25$.

Любая заявка может обслуживаться любым каналом. Заявки, застающие все 3 канала занятыми, образуют очередь и обслуживаются в порядке поступления. Необходимо получить оценки процесса функционирования системы:

1. Среднее время ожидания в очереди заявок каждого типа – T_1 , T_2 , T_3 .

2. Среднее количество заявок каждого типа в очереди – N_i .

3. Коэффициент использования каналов – K .

Очевидно, при построении модели поступающие заявки будут соответствовать транзактам, а работу обслуживающих каналов можно моделировать с помощью "накопителя". На рис. 12 приведена блок-схема работы модели, а на рис.13 – листинг программы.

Рассмотрим работу модели. Блок GENERATE вводит транзакты в модель. Для моделирования случайного события (определения типа заявки) используется дискретная функция STAN, определяющая с заданной вероятностью тип заявки 1, 2 или 3. Для сбора статистики очередей используются блоки QUEUE, DEPART. Причем номер очереди определяется типом заявки, т.е. в результате мы получим статистику по трем очередям, хотя в модели используется одна пара блоков QUEUE, DEPART. Блоки ENTER, LEAVE моделируют занятие и освобождение единицы "накопителя", что соответствует занятию и освобождению канала.

Емкость накопителя должна соответствовать количеству каналов и задается с помощью специальной карты PRIV STORAGE 3. Обслуживание заявок моделирует блок ADVANCE. Причем, обслуживание зависит от типа заявки, который определен в первом параметре (в блоке ASSIGN). Для определения времени обслуживания используется арифметическая переменная

OBSL FVARIABLE FN\$SNORM#FN\$SKO+FN\$SRED.

Функция SNORM определяет нормированную случайную величину, а функции SKO и SRED соответственно среднеквадратичное отклонение и математическое ожидание, зависящие от первого параметра, т.е. от типа заявки.

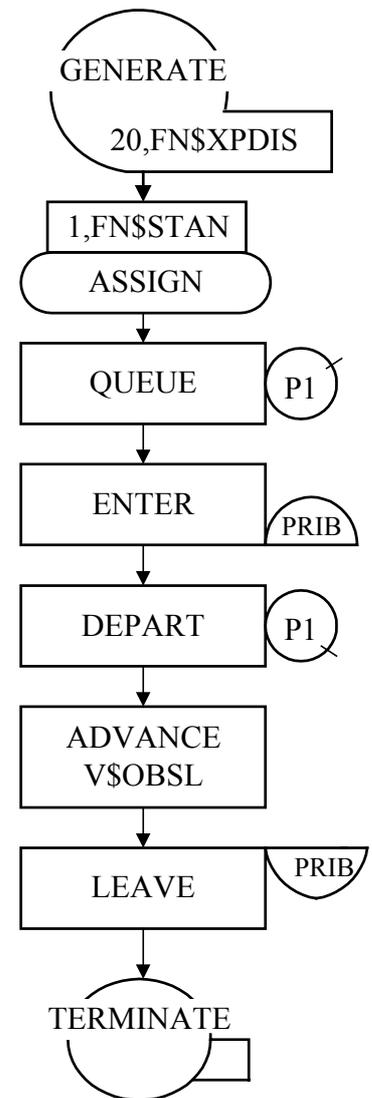


Рис. 12. Модель СМО с идентичными каналами

Длительность моделирования определяется картами

GENERATE 40000

TERMINATE 1

START 1.

Относительное и абсолютное время составляет 40000 временных единиц

```
GPSS/PC Program File MOD_PRI.GPS. (V 2, # 37349) 05-
19-1998 18:59:28
10 SNORM FUNCTION RN1,C25
0,-5/.00003,-4/.00135,-3/.00621,-2.5/.02275,-2
.06681,-1.5/.11507,-1.2/.15866,-1/.21186,-.8/.27425,-.6
.34458,-.4/.42074,-.2/.5,0/.57926,.2/.65542,.4
.72575,.6/.78814,.8/.84134,1/.88493,1.2/.93319,1.5
.97725,2/.99379,2.5/.99865,3/.99997,4/1,5
20 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
30 STAN FUNCTION RN1,D3
.3,1/.7,2/1,3
40 SKO FUNCTION P1,D3
1,9/2,15/3,25
50 SRED FUNCTION P1,D3
1,50/2,100/3,150
60 PRIB STORAGE 3
70 OBSL FVARIABLE FN$SNORM#FN$SKO+FN$SRED
80 GENERATE 40, FN$XPDIS
90 ASSIGN 1, FN$STAN
100 QUEUE P1
110 ENTER PRIB
120 DEPART P1
130 ADVANCE V$OBSL
140 LEAVE PRIB
150 TERMINATE
160 GENERATE 40000
170 TERMINATE 1
START 1
```

Рис. 13. Пример оформления программы модели рис. 12

Контрольные вопросы

3.1. Назовите основные структуры систем массового обслуживания (СМО)?

3.2. С помощью каких блоков обеспечивается моделирование СМО с различными дисциплинами выбора заявок из очереди?

3.3. Используя примеры моделей, рассмотренные в параграфах 3.1, 3.4, 3.5, 3.6, 3.7, 3.8, составить модели одноканальной СМО с различными дисциплинами заполнения очереди и обслуживания

Дисциплина выбора заявок на обслуживание	Дисциплина заполнения очереди			
	Очередь без ограничений	Ограничено время ожидания	Получает отказ поступившая заявка	Вытесняется самая старая заявка
В порядке поступления	A1	A2	A3	A4
С динамическим приоритетом	A5	A6	A7	A8
По минимуму времени допустимого ожидания		A9	A10	АП
В случайном порядке с равной вероятностью	A12	A13	A14	A15

3.4. Используя примеры моделей, рассмотренные в параграфах 3.1 – 3.10, составить модели многоканальной СМО с различными дисциплинами: заполнения очереди, обслуживания и выбора обслуживающего канала

Тип очереди	Дисциплина выбора каналов В1, В2, В3		
	Дисциплина выбора очереди В4, В5, В6		
Общая очередь	В порядке освобождения канала	В порядке нумерации 1,2,..первый свободный	В случайном порядке с равной вероятностью
	В1	В2	В3
Очередь раздельная к каждому каналу	По наименьшей длине очереди	По наименьшему коэффициенту использования канала	В случайном порядке
	В4	В5	В6

4. РЕАЛИЗАЦИЯ МОДЕЛИ НА ЭВМ

4.1. Команды GPSS/PC и технология работы с пакетом

Пакет GPSS/PC включает в себя два основных модуля: модуль GPSSPC.EXE, представляющий интегрированную среду, в которой производится ввод, редактирование, отладка и выполнение модели, и модуль GPSSREPT. EXE, предназначенный для получения стандартного отчета GPSS/PC. Загрузка обоих модулей производится обычным образом .

После загрузки интегрированной среды на экране появляется “заставка” с названием пакета: начинается так называемый сеанс работы с GPSS/PC. Затем заставка гасится, и появляется экран, разделенный на две части: большая верхняя часть содержит окно данных, меньшая нижняя часть – окно команд. Окно данных в начальный момент пусто, в окне команд в верхней командной строке высвечен символ “приглашения” >, сигнализирующий о готовности системы принимать команды.

Если исходная программа с моделью еще не введена и не записана на диске, то необходимо ввести ее с клавиатуры. Ввод производится в командную строку. Сначала вводится *номер строки* очередного оператора и нажимается клавиша *Пробел*. Курсор автоматически перемещается к началу следующего поля – *поля имени*, и в позиции курсора высвечивается символ L, сигнализирующий о том, что вы находитесь в поле имени (LABEL —метка). Если оператор имеет имя, необходимо ввести его и нажать клавишу *Пробел*, в противном случае – сразу нажать клавишу *Пробел*. В любом случае курсор переходит к началу следующего поля — *поля операции*, о чем сигнализирует символ V (VERB —глагол) в позиции курсора. Необходимо ввести название оператора и нажать клавишу *Пробел*. Очень удобным является то, что название оператора не обязательно вводить полностью: как только транслятор распознает оператор по нескольким первым буквам, он после нажатия клавиши *Пробел* сам дополнит его до полного названия.

При синтаксической ошибке в операторе под командной строкой появляется указатель на место ошибки, причем ошибочный символ не вводится. Необходимо в этом случае повторить ввод символа.

Аналогичным образом вводятся поля операндов, при этом в позиции курсора высвечивается обозначение текущего поля (A, B, ..., G). Для перехода к следующему полю операндов *вводится запятая*, для перехода к полю комментариев – *Пробел*. При переходе курсора в поле комментариев в позиции курсора высвечивается символ ;, сигнализирующий о возможности начать ввод комментария.

По окончании ввода последнего поля операндов или комментария следует нажать клавишу *Enter*, при этом введенный оператор транслируется и отображается в окне данных, а командная строка очищается, и в ее первой позиции снова появляется символ “приглашения”.

По мере ввода новых операторов окно данных заполняется, и по окончании ввода в нем находится вся исходная программа в последовательности ввода, необязательно совпадающей с последовательностью нумерации строк. Для отображения в окне данных исходной программы в последовательности нумерации строк необходимо ввести в командную строку команду DISPLAY (отобразить). Эта команда, как и все остальные команды GPSS/PC, вводится без номера строки. С помощью команды DISPLAY можно также вывести в окно данных отдельную строку, указав ее номер в поле A команды, или последовательность строк, указав начальный и конечный номера в полях A и B соответственно.

Удалить строки из исходной программы можно командой DELETE (удалить), указав в полях A и B начальный и конечный номера удаляемой последовательности. Для удаления одной строки достаточно ввести лишь поле A.

При необходимости вставить в текст новый оператор, поместив его между уже введенными операторами, достаточно ввести его с промежуточ-

ным номером строки. Вы можете перенумеровать строки, введя команду RENUMBER (перенумеровать), в поле **A** которой указывается номер первой строки, а в поле **B** – шаг перенумерации.

Отредактировать содержимое строки можно с помощью команды EDIT (редактировать), в поле **A** которой указывается номер редактируемой строки. При вводе такой команды в командной строке появляется редактируемая строка. Подводя курсор к нужным позициям строки, вы можете внести в нее необходимые изменения. По окончании редактирования следует нажать клавишу *Enter*, и отредактированная строка перенесется в окно данных, заменив в исходной программе первоначальную строку с этим номером. В этом можно убедиться, введя команду DISPLAY.

Если редактируемый оператор короткий, а изменений в нем много, то редактирование удобнее произвести, введя измененный оператор с тем же номером строки.

Если работа с моделью предполагается и по окончании данного сеанса, то после ввода и редактирования исходную программу имеет смысл записать на диск. Для этого необходимо ввести команду SAVE (сохранить), в поле **A** которой указывается имя файла, в который будет записана модель. Файл должен иметь расширение .GPS.

Записав модель в файл, вы сможете в следующем сеансе работы с GPSS/PC не вводить ее заново с клавиатуры, а считать с диска, введя команду @спецификация_файла, где спецификация_файла - полное имя файла, которое вы дали исходной программе в команде SAVE, включающее расширение .GPS. При выполнении команды @ операторы исходной программы по мере их считывания из файла транслируются и выводятся в окно данных.

4.2. Прогон модели и наблюдение за моделированием

После того, как исходная программа модели введена с клавиатуры или считана с диска и оттранслирована, в памяти ПК создалась текущая модель, и

теперь можно выполнить ее прогон. Для этого в командную строку необходимо ввести управляющий оператор START (см. 4.3.), указав в поле А соответствующее начальное значение счетчика завершений. После нажатия клавиши ENTER оператор START переносится в окно данных, и прогон модели начинается. Об этом сигнализирует сообщение Simulation in Progress, появляющееся в нижней строке командного окна – строке состояния, а также так называемый индикатор моделирования, мигающий в правой стороне нижней части окна данных.

Если прогон модели достаточно длинный, то можно наблюдать за процессом моделирования, открывая те или иные *графические окна*. Это производится путем нажатия клавиши Alt одновременно с символьной клавишей с первой буквой названия окна.

Например, после нажатия клавиш Alt + B в верхней части экрана на месте окна данных появляется окно блоков (BLOCKS), изображающее динамику продвижения транзактов через блок-схему модели. Рядом с каждым блоком выводится текущее число транзактов в нем, которое обновляется в процессе моделирования. Нажав клавиши Alt + N, можно заменить эту информацию на общее число транзактов, прошедших через каждый блок. Блок, в котором находится активный транзакт, выделен повышенной яркостью (на цветных мониторах – другим цветом).

Нажав клавиши Alt + F, вы можете наблюдать окно устройств (FACILITIES), в котором наглядно отображена информация о текущем состоянии каждого устройства модели: его использовании, занятости, очереди к нему.

Аналогичную информацию о многоканальных устройствах можно получить, нажав Alt + S и открыв окно памятей (STORAGES).

Если в модели используются статистические таблицы, то, нажав клавиши Alt + T, вы откроете окно таблиц (TABLES) с гистограммой распределения соответствующего атрибута модели, обновляющейся в процессе моде-

лирования. Над гистограммой выводятся также текущие значения среднего и среднеквадратического отклонения табулируемого атрибута.

Если в модели используются матрицы, то, нажав клавиши Alt + M, вы откроете окно матриц (MATRICES), в котором можно наблюдать обновляющиеся в процессе моделирования значения элементов матриц.

Находясь в любом из перечисленных окон, вы можете путем нажатия клавиш Alt + L включить трассировку активного транзакта. При этом в верхней части окна появляется строка, содержащая информацию о текущем модельном времени, номере активного транзакта и его продвижении через блок-схему модели. Отключить трассировку можно повторным нажатием этих же клавиш.

Перемещение внутри окна любого типа к тому или иному объекту этого типа осуществляется путем нажатия клавиш управления курсором PgUp, PgDn и End. Возвращение в окно данных производится путем нажатия клавиш Alt + D.

Следует заметить, что наблюдение графических окон и особенно строки трассировки существенно замедляет моделирование, и при длинных прогонах моделей этой возможностью не следует злоупотреблять.

Открытие того или иного окна может быть выполнено также с помощью команды WINDOW (окно), в поле A которой указывается имя окна, однако удобнее это делать так, как описано выше.

Кроме графических окон внутри любого из них, кроме окна данных, может быть открыто до четырех микроокон. Микроокна открываются и закрываются командой MICROWINDOW (микроокно), имеющей следующий формат: MICROWINDOW A, B, C; комментарий. В поле A указывается номер микроокна – константа 1, 2, 3, 4. Поле B содержит наблюдаемую величину – любой СЧА модели. Поле C определяет состояние микроокна в результате выполнения команды: ON – открыто, OFF – закрыто. Если поле C пусто, то по умолчанию команда открывает заданное микроокно. В поле коммента-

рия может быть задано название микроокна длиной до восьми символов. При открытии любого окна заданные микроокна с обновляющейся в процессе моделирования информацией появляются в правой части соответствующего окна. Микроокно имеет форму прямоугольника с названием над рамкой, если оно было задано в комментарии к команде MICROWINDOW.

В процессе моделирования можно также наблюдать одновременно до двух графиков зависимостей любых СЧА модели от модельного времени. Для этого необходимо до запуска модели ввести одну или две команды PLOT (начертить), имеющие следующий формат:

PLOT A, B, C, D; комментарий. В поле A указывается аргумент зависимости – любой СЧА модели. Поле B должно содержать максимальное значение этого СЧА, определяющее масштаб изображения по оси Y. Операнд B задается константой, значение которой должно быть не менее 13. Поля C и D определяют начальное и конечное значения модельного времени, определяющие масштаб изображения по оси X. Эти операнды также задаются константами. В поле комментария может быть задан заголовок графика длиной до 34 символов.

График обновляется при каждом изменении модельного времени, если оно попадает в диапазон, заданный операндами C и D. Указанный в поле A СЧА-аргумент вычисляется относительно первого транзакта, обрабатываемого после изменения модельного времени.

Процесс моделирования можно прервать, нажав одну из клавиш ESC или Home. При этом в строке состояния командного окна появляется сообщение о номере активного транзакта, обрабатываемого симулятором в момент прерывания. Вы можете узнать значения интересующих вас стандартных числовых атрибутов модели в момент прерывания, введя команду SHOW (показать), операндом которой служат отдельные СЧА или выражения из них. Значение заданного в команде СЧА или выражения выводится в окно данных или другое активное окно. Введя команду EVENTS (события), можно

увидеть в окне данных содержимое списков текущих и будущих событий. Команда USERCHAINS (списки пользователя) позволяет просматривать в окне данных содержимое списков пользователя. Обе последние команды не имеют операндов.

Инициировать прерывание моделирования можно также с помощью команды STOP (остановить), имеющей следующий формат: STOP A, B, C. В поле A указывается номер транзакта, вызывающего прерывание, задаваемый константой. Если это поле пусто, то прерывание вызывается любым транзактом. В поле B задается имя или номер блока, при входе в который происходит прерывание. Если этот операнд опущен, то прерывание происходит при входе в любой блок. В поле C указывается ON для установки условия прерывания и OFF для снятия этого условия (по умолчанию ON).

Например, команда STOP 100, MET1 устанавливает условие прерывания моделирования при входе транзакта с номером 100 в блок с именем MET1. Команда STOP 2 будет вызывать прерывание при каждом продвижении транзакта с номером 2, а команда STOP, CHAIR при каждом входе любого транзакта в блок с именем CHAIR. Наконец, команда STOP без операндов будет вызывать прерывание при каждом продвижении любого транзакта, а команда STOP,, OFF снимает все условия прерывания, установленные ранее другими командами STOP.

Прервав моделирование, можно также воспользоваться командой STEP (выполнить шаг) для пошагового выполнения модели с целью ее отладки. Операнд в поле A команды задает количество входов активного транзакта в блоки при каждом выполнении команды. Обычно этот операнд равен 1, и каждое выполнение команды STEP приводит к продвижению активного транзакта к следующему блоку. Отладку с использованием команды STEP удобно проводить, находясь в окне блоков.

Для продолжения моделирования после прерывания следует ввести в командную строку команду CONTINUE (продолжить).

Команды STEP и CONTINUE могут не только вводиться в командную строку с клавиатуры, но и выбираться из меню команд, появляющемся в командном окне при активизации любого графического окна. Выбор производится подводом крестообразного курсора в прямоугольную область нужной команды и нажатием клавиши Insert. В окне блоков меню команд предоставляет также некоторые дополнительные возможности.

Команды STEP, CONTINUE, а также любые другие часто используемые команды удобно загрузить на функциональные клавиши F1 – F10. Для этого после ввода загружаемой команды с клавиатуры необходимо нажать клавиши Ctrl + Fn, где n – номер выбранной функциональной клавиши. После загрузки команды на функциональную клавишу для ее выполнения достаточно нажатия этой клавиши.

4.3.Управляющие операторы GPSS/PC

Кроме строк, непосредственно кодирующих основные блоки модели, необходимы операторы, управляющие процессом моделирования (управляющие операторы GPSS/PC).

Оператор SIMULATE (моделировать) устанавливает предел реального времени, отводимого на прогон модели. Если прогон не завершится до истечения этого времени, то он будет прерван принудительно с выдачей накопленной статистики в отчет.

Оператор SIMULATE имеет единственный операнд A, содержащий предельное время моделирования в минутах, задаваемое константой. Оператор размещается перед оператором START, начинающим лимитированный прогон. Если длительность моделирования не ограничена, этот оператор может отсутствовать.

Оператор START используется для управления началом моделирования и его продолжительностью. Формат оператора

START A, B, C, D.

В поле А организуется счетчик числа прогонов модели. При каждом входе транзакта в блок TERMINATE из содержимого поля А карты START вычитается число единиц, указанное в поле А блока TERMINATE. В поле В может быть помещен указатель NP, блокирующий вывод результатов моделирования в файл REPORT.GPS. В этом случае необходимо сформировать файл с результатами моделирования с помощью команды REPORT имя файла, NOW. Поле С не используется. Запись 1 в поле D указывает на то, что при формировании файла результатов выводится содержимое цепей текущих и будущих событий.

Команда CLEAR (ОЧИСТИТЬ) используется при необходимости выполнить несколько прогонов модели, изменяя исходные данные. Карта устанавливает в нуль абсолютное и относительное время, убирает из модели все транзакты, сбрасывает статистики и приводит к исходному состоянию все объекты.

Предположим, необходимо исследовать работу системы (см. параграф 3.10, рис. 12, 13) при различном количестве обслуживающих каналов. Количество каналов определяется емкостью накопителя, картой

```
PRI STORAGE 3.
```

Если после карты START 1 (рис.13) добавить карты CLEAR , PRI STORAGE 2, START 1, то мы получим результаты моделирования при использовании двух каналов. В данном случае номер строки не указывается, и команда выполняется сразу после ввода.

Для исследования модели при изменении параметров целесообразно задавать их через сохраняемые величины. Например, при исследовании системы рис. 12, 13 при изменении интенсивности входного потока удобно задать строку:

```
80 GENERATE X$INT,FN$XPDIS.
```

Тогда при исследовании модели при изменении интенсивности входного потока используется следующая последовательность операторов:

- 1) INITIAL X\$INT,20; START 1,NP;
REPORT RES20,NOW;
- 2) CLEAR; INITIAL X\$INT,40; START 1,NP;
REPORT RES40,NOW;
- 3) CLEAR; INITIAL X\$INT,60; START 1,NP;
REPORT RES60,NOW.

В результате мы получим три прогона модели при среднем интервале между моментами поступления 20, 40, 60 ед. Результаты моделирования будут записаны соответственно в файлы RES20, RES40, RES60. Для обработки результатов моделирования и получения отчета в стандартной форме используется модуль GPSSREPT.EXE .

Оператор RMULT (установить значения генераторов) позволяет перед началом прогона установить начальные значения генераторов случайных чисел RN, определяющие генерируемые ими последовательности. Поля A - G оператора могут содержать начальные значения генераторов соответственно RN1– RN7, задаваемые константами.

Оператор RESET (сбросить) сбрасывает всю статистическую информацию, накопленную в процессе прогона модели. При этом состояние аппаратных, динамических и запоминающих объектов, а также генераторов случайных чисел сохраняется, и моделирование может быть возобновлено с повторным сбором статистики. Оператор не имеет операндов.

С оператором RESET связано различие между относительным (СЧА C1) и абсолютным (СЧА AC1) модельным временем. Таймер относительного времени C1 измеряет модельное время, прошедшее после последнего сброса статистики оператором RESET, а таймер абсолютного времени AC1 – модельное время, прошедшее после начала первого прогона модели. Если не использовалось ни одного оператора RESET, то значения этих таймеров совпадают. Оператор RESET устанавливает таймер C1 в ноль и не влияет на таймер AC1.

Оператор RESET используется обычно при моделировании стационарных процессов, когда требуется собрать статистику по отдельным интервалам стационарности или исключить влияние переходного периода на собираемую статистическую информацию.

4.4. Точность результатов моделирования

Точность статистического моделирования на GPSS/PC определяется несколькими факторами:

- 1) выбором единицы измерения времени;
- 2) количеством реализаций или длительностью моделирования для стационарных систем;
- 3) начальным состоянием системы.

4.4.1. Выбор масштаба времени

В GPSS/PC используются целочисленные значения моментов времени. Поэтому неправильный выбор масштаба времени может привести к значительной погрешности. Масштаб времени определяется разработчиком модели и зависит от значения тех временных интервалов, которые задаются в соответствующих блоках модели.

Можно привести некоторые рекомендации [3], позволяющие снизить этот вид погрешности. Если задаются случайные величины, распределенные по экспоненциальному закону, то математическое ожидание, равное $1/\lambda$, должно быть больше 50. Эти же рекомендации можно использовать при других видах распределений.

В системе моделирования GPSS World используются вещественные значения модельного времени, поэтому при ее использовании можно обойтись без масштабирования.

4.4.2. Переходный процесс и точность результатов

Обычно при моделировании стационарных процессов нам необходимо получить статистику в “типичном”, установившемся режиме. Однако начальное состояние часто не соответствует установившемуся. Поэтому при сборе статистики начальное “нетипичное” состояние, будет вносить существенную погрешность при подсчете характеристик системы.

Для получения достоверной статистики можно поступить следующим образом:

- а) моделировать до возникновения типичных условий работы системы;
- б) отбросить собранную до этого времени статистику без изменения состояния модели;
- в) продолжить моделирование, собирая статистику, на которую уже не влияют нетипичные ситуации.

Для определения длительности входа в стационарный режим используются карты RESET , START.

Карта RESET производит сброс накопленной статистики без изменения состояния модели, а карта START возобновляет моделирование.

Рассмотрим пример [3].

На трикотажной фабрике 50 швейных машин работают по 8 часов в день и по 5 дней в неделю. Любая из этих машин может выйти из строя в любой момент времени. В этом случае ее заменяют резервной машиной, причем либо сразу, либо по мере ее появления из ремонта. Тем временем сломанную машину отправляют в мастерскую, где ее ремонтируют и возвращают в цех, но уже в качестве резервной. Управляющий хочет знать, сколько рабочих следует нанять для работы в мастерской, сколько машин необходимо иметь в резерве, какую платить за это арендную плату, чтобы резервными машинами можно было подменить 50 собственных. Цель – минимизация стоимости производства. Оплата рабочих в мастерской 3,75 доллара в час. За машины,

находящиеся в резерве, надо платить 30 долларов в день. Почасовой убыток при использовании менее 50 машин в производстве оценивается в 20 долларов за машину. Этот убыток возникает из-за снижения производства.

Исследуем зависимость точности моделирования от длительности переходного процесса.

На рис. 14 приведена программа модели на GPSS/PC.

5 REZ	MATRIX	,75,2
10 MEN	STORAGE	4
20 NOWON	STORAGE	50
30	GENERATE	,,,54,10
40 COM	ENTER	NOWON
50	ADVANCE	157,25
60	LEAVE	NOWON
65	ENTER	MEN
67	ADVANCE	7,3
68	LEAVE	MEN
70	TRANSFER	,COM
80	GENERATE	,,,1,5
85 COM2	ADVANCE	40
90	ASSIGN	1+,1
100	MSAVEVALUE	REZ,P1,1,SR\$NOWON
105	TRANSFER	,COM2
107	GENERATE	40
110	TERMINATE	1

Рис. 14. Программа модели на GPSS/PC

Длительность прогона определяется блоками 107, 110 и составляет 1 неделю (40 часов). Тогда для определения “ входа” в стационарный режим будем моделировать процесс в течение 20 недель и сбрасывать статистику с помощью оператора RESET через каждую неделю. Анализ полученных результатов позволяет сделать вывод о длительности моделирования в стационарном режиме. Далее производится прогон модели в течение 30 недель с помощью оператора START 30.

Результаты моделирования приведены на рис. 15. Как видно из рисунка, за счет статистики переходного режима установившееся состояние не достигается и через 50 недель. Если произвести сброс статистики через 20 недель, то установившееся состояние достигается уже через несколько не-

дель.

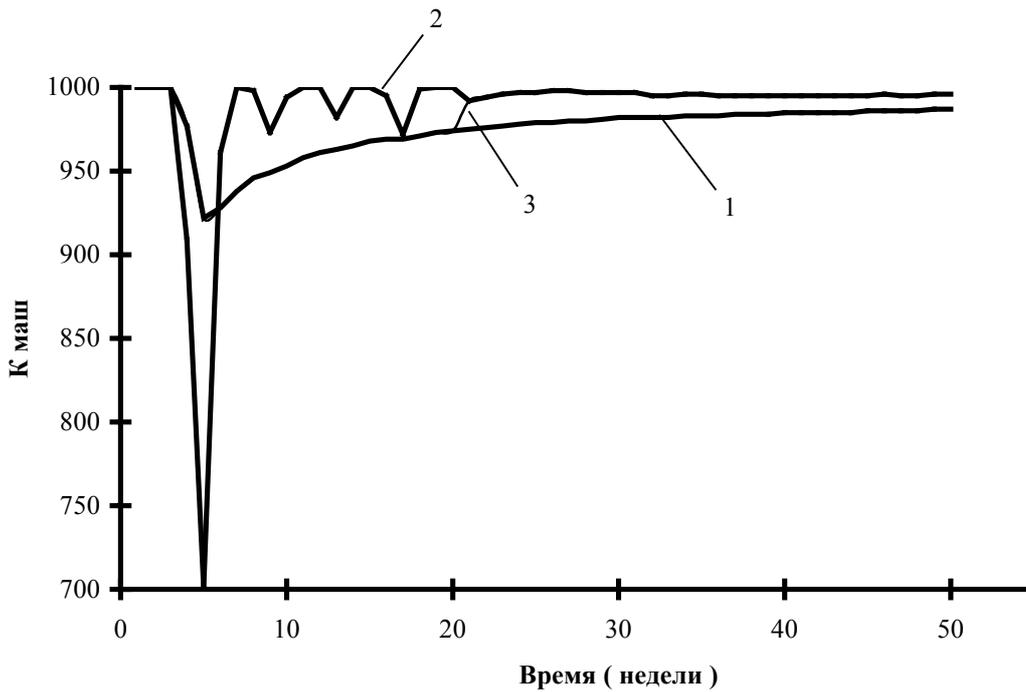


Рис. 15. Зависимость коэффициента загрузки (в долях тысячи) цеха при 4 резервных машинах и 4 рабочих:
1-без сброса статистики;
2 -сброс статистики через 1 неделю до 20, далее без сброса статистики;
3- сброс статистики через 20 недель, далее без сброса статистики

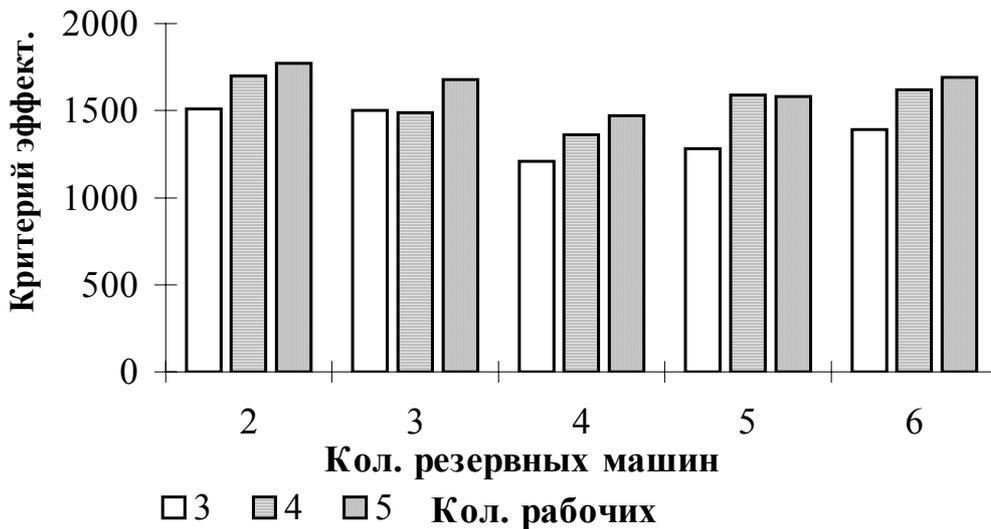


Рис. 16. Зависимость расходов от числа резервных машин и количества рабочих

На рис. 16 приведены результаты исследования модели при вариации

количества резервных машин и количества рабочих. Как видно из диаграмм, наименьшие расходы достигаются при 4 резервных машинах и 3 рабочих.

4.4.3. Обработка одновременных событий

Так как модельное время в GPSS/PC целочисленно, то оказывается

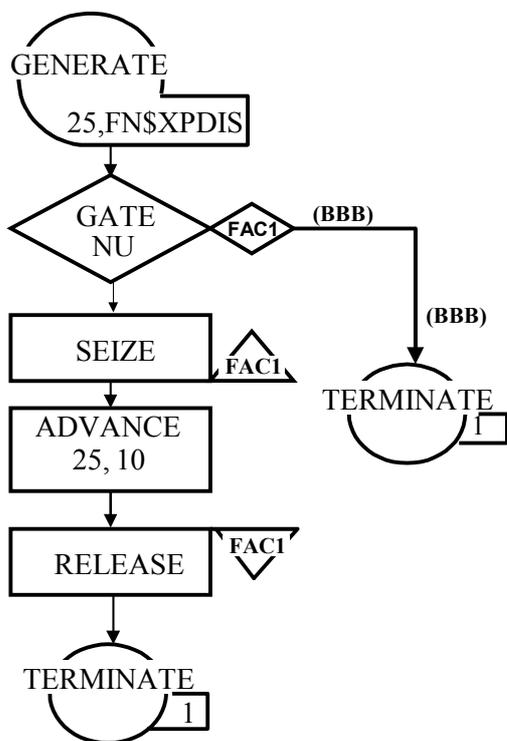


Рис. 17. Модель СМО с отказами

вполне вероятным одновременное наступление двух или более событий, причем вероятность этого тем больше, чем крупнее выбранная единица модельного времени. В некоторых случаях одновременное наступление нескольких событий, или так называемый временной узел, может существенно нарушить логику модели.

Рассмотрим модель системы массового обслуживания на рис. 17. Здесь может образоваться временной узел между событиями “поступление транзакта на вход модели” и “завершение обслуживания в устройстве FAC1”. Если эти два события

происходят одновременно, то обработка временного узла зависит от последовательности транзактов в списке текущих событий.

Предположим, что первым в списке расположен транзакт, освобождающий прибор FAC1. Тогда вначале будет обработан этот транзакт, т. е. событие “завершение обслуживания в FAC1”, причем условие “Прибор FAC1 не занят”, проверяемое в блоке GATE, станет истинным. Затем будет обработан транзакт, поступивший на вход модели в блок GATE из блока GENERATE. При этом транзакт будет впущен в блок SEIZE, и устройство FAC1 в тот же момент модельного времени снова окажется занятым. Такая ситуация при обработке временного узла представляется естественной.

Предположим теперь, что первым в списке текущих событий расположен транзакт, поступающий на вход модели. Так как условие “Прибор FAC1 не занят” ”ложно”, то блок GATE направит этот транзакт в блок с именем ВВВ. Таким образом, в модели будет зафиксирован отказ в обслуживании, хотя в этот же момент модельного времени устройство станет доступным.

Порядок расположения транзактов в списке текущих событий случаен, и в среднем в половине случаев временной узел будет обрабатываться не так, как нужно. В результате статистика, связанная с отказами, окажется искаженной. Для правильной обработки временного узла надо обеспечить такой порядок расположения транзактов в списке текущих событий, чтобы транзакт, освобождающий устройство, всегда располагался первым. Этого можно добиться, управляя приоритетами транзактов.

Для приведенного примера достаточно между блоками SEIZE FAC1 и ADVANCE 25, 10 вставить блок PRIORITY 2. Тогда при поступлении транзакта в блок задержки будет повышаться его приоритет. При очередном просмотре цепи текущих событий транзакт завершающий обслуживание всегда будет обрабатываться первым.

Контрольные вопросы

- 4.1. Из каких модулей состоит пакет GPSS/PC?
- 4.2. Как осуществляется прогон модели и наблюдение за процессом моделирования?
- 4.3. Назовите команды, управляющие процессом моделирования?
- 4.4. Какие факторы определяют точность моделирования на GPSS/PC?
- 4.5. Как влияет переходный процесс на точность статистических оценок?
- 4.6. Что такое временной узел? Какие блоки используются для правильной обработки временных узлов?

5. ПРИМЕРЫ МОДЕЛЕЙ

5.1. Модель справочной телефонной сети

Концептуальное описание системы. Пять операторов работают в справочной телефонной сети города, сообщая номера телефонов по запросам абонентов, которые обращаются по одному номеру 09. Автоматический коммутатор переключает абонента на того оператора, в очереди которого ожидается наименьшее количество абонентов. Причем наибольшая допустимая длина очереди перед оператором – 2 абонента. Если все очереди имеют максимальную длину, вновь поступивший вызов получает отказ. Обслуживание абонентов оператором длится 30 ± 20 с. Вызовы поступают в справочную через каждые 5 ± 3 с.

Исследуемые показатели. Смоделируйте обслуживание 1000 вызовов. Определите частоту отказов, коэффициенты загрузки операторов, среднюю длительность ожидания в очереди. Как изменятся характеристики системы при 6 операторах?

Построение модели. Поскольку в GPSS/PC используется целочисленное модельное время, необходимо ввести масштаб времени. Выберем единицу измерения времени 0,1 с.

В качестве модели выберем многоканальную СМО с отдельными очередями, приведенную в параграфе 3.9 на рис. 11.

Составим таблицу соответствий

Объект GPSS	Объект реальной системы
<i>Транзакт</i> Параметры P1	Запросы абонента Номер оператора
<i>Устройства:</i> 1, 2, 3, 4, 5	Операторы справочной службы
<i>Цепь пользователя:</i> 1, 2, 3, 4, 5	Очереди к операторам

Программа для 5 операторов приведена на рис.18.

```

; GPSS/PC Program File SPR5.GPS. (V 2, # 40550) 08-19-2002 14:27:22
10          GENERATE      50,30
20          SELECT NU     1,1,5,, ,AAA
30 BBB      SEIZE        P1
40          ADVANCE      300,200
50          RELEASE      P1
60          UNLINK       P1,BBB,1
70          TERMINATE    1
80 AAA      SELECT MIN    1,1,5,, ,CH
90          TEST L       CH*1,2,ОТК
95          PRIORITY     5
100         LINK         P1,FIFO
110 ОТК     TERMINATE    1

```

Рис. 18. Программа модели справочной телефонной сети

Исследование модели. На рис. 19 приведены результаты прогона модели при 5 операторах, а в таблице 3 результаты исследования модели при 5 и 6 операторах. При расчете среднего времени ожидания в очереди производилось усреднение по всем очередям.

Анализ результатов позволяет сделать следующие выводы.

При 5 операторах:

- коэффициенты загрузки операторов незначительно снижаются с увеличением номера оператора;
- длительность ожидания в очереди незначительно уменьшается с увеличением номера очереди.

При увеличении числа операторов с 5 до 6:

- незначительно уменьшается загрузка операторов;
- уменьшается частота отказов с 17,1% до 2,1%;
- уменьшается длительность ожидания в очереди с 52 сек. до 38 сек.

START_TIME	END_TIME	BLOCKS	FACILITIES	STORAGES	FREE_MEMORY
0	50127	12	5	0	16112

LINE	LOC	BLOCK_TYPE	ENTRY_COUNT	CURRENT_COUNT	RETRY
10	1	GENERATE	1014	0	0
20	2	SELECT	1014	0	0
30	BBB	SEIZE	833	0	0
40	4	ADVANCE	833	4	0
50	5	RELEASE	829	0	0
60	6	UNLINK	829	0	0
70	7	TERMINATE	829	0	0
80	AAA	SELECT	1008	0	0
90	9	TEST	1008	0	0
95	10	PRIORITY	837	0	0
100	11	LINK	837	10	0
110	OTK	TERMINATE	171	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAILABLE	OWNER	PEND	INTER	RETRY	DELAY
1	162	0.999	309.14	1	1003	0	0	0	0
2	164	0.997	304.98	1	997	0	0	0	0
3	167	0.996	299.09	1	1005	0	0	0	0
4	164	0.994	304.09	1	998	0	0	0	0
5	176	0.993	283.10	1	0	0	0	0	0

USER_CHAIN	CHAIN_SIZE	RETRY	AVE.CONT	ENTRIES	MAX	AVE.TIME
1	2	0	1.88	163	2	577.36
2	2	0	1.83	165	2	555.68
3	2	0	1.75	168	2	522.67
4	2	0	1.66	165	2	504.66
5	1	0	1.52	176	2	434.23

Рис. 19. Результаты прогона модели рис. 18

Таблица 3

Результаты исследования модели справочной системы

Количество операторов	Коэффициенты загрузки операторов						Среднее время ожидания в очереди (сек.)	Частота отказов
	1	2	3	4	5	6		
5	0,999	0,997	0,996	0,994	0,993		52	17,1%
6	0,998	0,996	0,99	0,984	0,981	0,969	38	2,1%

5.2. Модель информационно-поисковой библиографической системы

Концептуальное описание системы. Информационно-поисковая библиографическая система построена на базе двух ЭВМ и имеет один терминал для ввода и вывода информации. Первая ЭВМ обеспечивает поиск литературы по научно-техническим проблемам (вероятность обращения к ней – 0,7), а вторая – по медицинским (вероятность обращения – 0,3). Пользователи обращаются к услугам системы каждые 5 ± 2 мин. Если в очереди к терминалу ожидают 10 пользователей, то вновь прибывшие пользователи получают отказ в обслуживании. Поиск информации на первой ЭВМ продолжается 6 ± 4 мин., а на второй 3 ± 2 мин. Для установления связи с нужной ЭВМ и передачи текста запроса пользователи тратят 2 ± 1 мин. Вывод результатов поиска происходит за 1 мин.

Исследуемые показатели. Смоделируйте процесс работы системы за 80 часов работы. Определите среднюю и максимальную длину очереди к терминалу, коэффициенты загрузки технических средств системы. Как изменятся характеристики системы, если будет установлен еще один терминал?

Построение модели. Составим таблицу соответствий

Объект GPSS	Объект реальной системы
<i>Транзакты</i> Сегмент 1 Сегмент 2 Параметры P1 P2	Запросы пользователей Таймер Номер ЭВМ Номер терминала
<i>Устройства:</i> 1, 2 3, 4	ЭВМ Терминалы
<i>Цепь пользователя:</i> BUF <i>Функции:</i> OBS1 OBS2 <i>Переменная:</i> OBS	Очередь к терминалам Длительность поиска информации соответственно на первой и второй ЭВМ Длительность поиска информации

Введем масштаб времени. Выберем единицу измерения времени 1 с.
 Построим модель системы для двух терминалов.

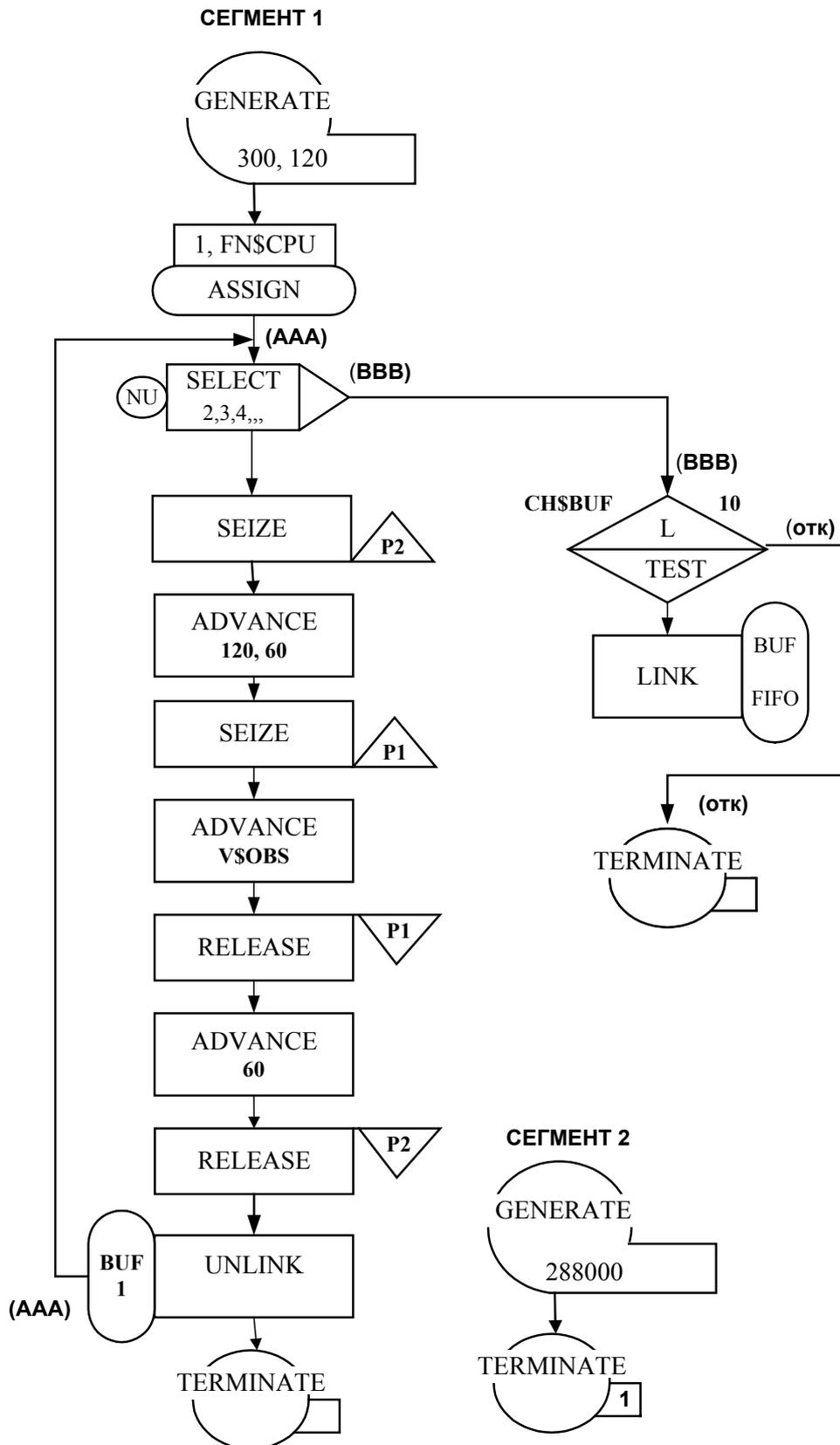


Рис. 20. Модель информационно-поисковой системы

Модель системы (рис. 20) состоит из двух сегментов. Первый сегмент является моделью системы. Вторым сегментом (таймер) определяется длительность прогона 80 час (288000 сек.). Для запуска модели необходимо использовать команду START 1.

На рис. 21 приведена программа модели для двух терминалов.

При моделировании длительности поиска информации можно задать две функции для моделирования случайных величин, распределенных равномерно:

```
OBS1 FUNCTION RN2, C2
```

```
0, 120/ 1,600
```

```
OBS2 FUNCTION RN2, C2
```

```
0, 60/1, 300
```

и присвоить именам функций численные значения с помощью оператора EQU:

```
OBS1 EQU 1
```

```
OBS2 EQU 2
```

Тогда с помощью переменной OBS FVARIABLE FN*1 можно определить длительность обслуживания в зависимости от номера ЭВМ.

Для выбора типа ЭВМ в зависимости от типа литературы используем дискретную функцию, которая моделирует случайное событие:

```
CPU FUNCTION RN1, D2
```

```
0.7,1/1,2
```

Для выбора свободного терминала используется блок

```
85 AAA SELECT NU 2,3,4,,BBB .
```

Номер свободного терминала заносится во второй параметр транзакта. Если используется один терминал, в полях В и С необходимо записать 3.

; GPSS/PC Program File INFBIB2TER.GPS.(V 2, # 40550) 08-19-2002 17:16:35

* *Сегмент 1*

```
10 OBS1      EQU          1
20 OBS2      EQU          2
30 OBS1      FUNCTION     RN2,C2
0,120/1,600
40 OBS2      FUNCTION     RN2,C2
0,60/1,300
50 CPU       FUNCTION     RN1,D2
0.7,1/1,2
60 OBS      FVARIABLE     FN*1
70          GENERATE     300,120
80          ASSIGN      1, FN$CPU
85 AAA      SELECT NU     2,3,4,, ,BBB
100         SEIZE       P2
110         ADVANCE     120,60
120         SEIZE       P1
130         ADVANCE     V$OBS
140         RELEASE     P1
150         ADVANCE     60
160         RELEASE     P2
170         UNLINK      BUF,AAA,1
180         TERMINATE
190 BBB      TEST L      CH$BUF,10,OTK
200         LINK        BUF,FIFO
210 OTK      TERMINATE
* Сегмент 2
220         GENERATE     288000
230         TERMINATE     1
```

Рис. 21. Программа модели информационно-поисковой системы

Исследование модели. Результаты исследования модели приведены в таблице 4

Таблица 4

Результаты исследования информационно-поисковой системы

Количество терминалов	Коэффициент использования				Максимальная длина очереди	Средняя длина очереди	Среднее время ожидания в очереди (мин.)	Частота отказов	Количество обслуженных пользователей
	Терм 1	Терм 2	ЭВМ1	ЭВМ2					
1	0,998	–	0,525	0,102	10	9,5	75.	38%	594
2	0,954	0,942	0,882	0,186	10	2,3	14	0	964

Анализ результатов показывает, что при одном терминале система не обеспечивает удовлетворительное обслуживание:

- имеет большое количество отказов (38%);
- имеет большое среднее время ожидания в очереди (75 мин.).

При добавлении еще одного терминала качество обслуживания улучшается:

- отказов в обслуживании нет;
- среднее время ожидания уменьшается до 14 мин;
- коэффициенты использования терминалов достаточно высоки (0,954 и 0,942).

5.3. Модель информационной системы реального времени

Концептуальное описание системы. Информационная система реального времени состоит из центрального процессора (ЦП), основной памяти (ОП) емкостью 10000 байтов и накопителя на магнитных дисках (МД). Запросы от удаленных терминалов поступают в среднем каждые 120 мс и обрабатываются на ЦП за время 1мс. После этого запрос помещается в ОП или

получает отказ, если ОП заполнена (каждый запрос занимает 200 байтов памяти). Для обслуживания запросов производится поиск информации на МД за время в среднем 120 мс. и ее считывание за время в среднем 10 мс. Работа с МД не требует вмешательства ЦП. Для подготовки ответа необходима работа ЦП в течение 5 мс. После этого запрос освобождает место в ОП. Закон распределения значений, заданных средними, экспоненциальный.

Исследуемые показатели. Смоделируйте процесс обслуживания 1000 запросов. Определите частоту отказов, среднее содержимое ОП, коэффициенты загрузки процессора и МД.

Построение модели. Выберем единицу измерения времени 0,1мс. Тогда все временные интервалы увеличатся в 10 раз

Составим таблицу соответствий

Объект GPSS	Объект реальной системы
<i>Транзакты</i> Сегмент 1 Сегмент 2	Запросы пользователей Загрузчик
<i>Устройства:</i> CPU DSK	Центральный процессор (ЦП) Магнитный диск (МД)
<i>Цепь пользователя:</i> MEM	Оперативная память
<i>Функции:</i> XPDIS	Функция экспоненциального распределения с единичным значением среднего

На рис. 22 приведена блок-схема модели, а на рис. 23 программа модели.

Модель состоит из двух сегментов. Первый сегмент имитирует поступление запросов в ОП. Транзакты (запросы) вводятся в модель с помощью блока GENERATE. Далее моделируется процесс обработки запроса процессором. Блок TEST проверяет условие заполнения цепи пользователя MEM (ОП). Если цепь пользователя заполнена, то транзакт переходит в блок TERMINATE (получает отказ). Если цепь не заполнена, то транзакт поступает в цепь пользователя.

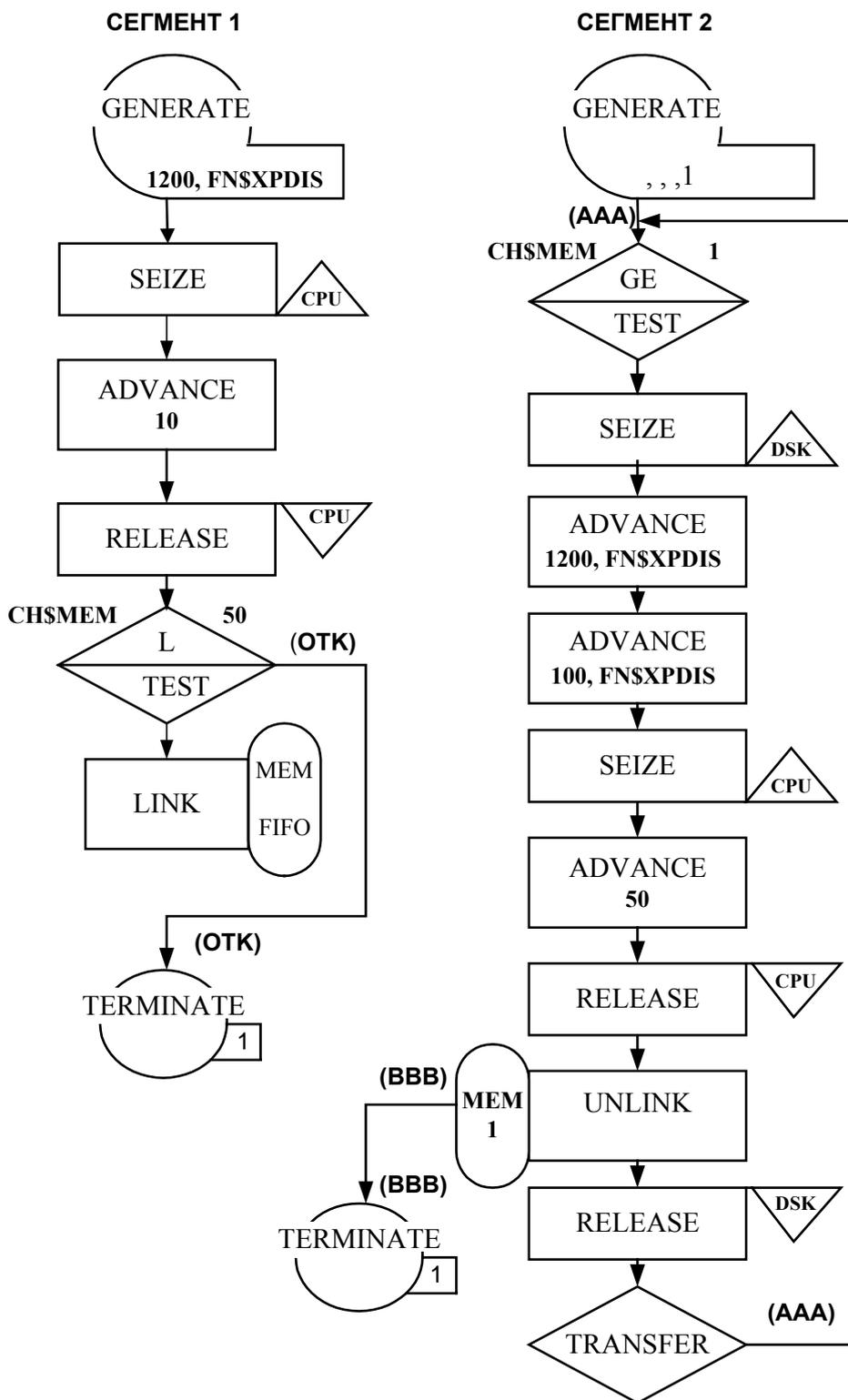


Рис.22. Модель информационной системы реального времени

Второй сегмент моделирует процесс обслуживания запросов. В модель вводится с помощью блока **GENERATE** „, 1 один транзакт. Этот транзакт исполняет роль загрузчика, т. е. он в порядке поступления направляет на обслуживание транзакты из цепи пользователя.

Блок AAA TEST GE CH\$MEM, 1 определяет наличие в цепи пользователя транзактов (запросов). Если транзакты в цепи пользователя (ОП) есть, то производится обработка транзакта с помощью блоков 100 – 170. Блок UNLINK MEM, BBB, 1 выводит обслуженный транзакт из цепи пользователя в блок BBB TERMINATE 1. Транзакт–загрузчик с помощью блока TRANSFER , AAA возвращается к блоку TEST и цикл повторяется. Для обслуживания 1000 запросов необходимо задать команду START 1000.

```
; GPSS/PC Program File INFSIST.GPS. (V 2, # 40550) 08-20-2002 13:41:50
```

```
10 XPDIS FUNCTION RN3,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
```

* *Сегмент 1*

```
20 GENERATE 1200, FN$XPDIS
30 SEIZE CPU
40 ADVANCE 10
50 RELEASE CPU
60 TEST L CH$MEM, 50, OTK
70 LINK MEM, FIFO
```

* *Сегмент 2*

```
80 GENERATE , , , 1
90 AAA TEST GE CH$MEM,1
100 SEIZE DSK
110 ADVANCE 1200, FN$XPDIS
120 ADVANCE 100, FN$XPDIS
130 SEIZE CPU
140 ADVANCE 50
150 RELEASE CPU
160 UNLINK MEM, BBB, 1
170 RELEASE DSK
180 TRANSFER , AAA
190 BBB TERMINATE 1
200 OTK TERMINATE 1
```

Рис. 23. Программа модели информационной системы реального времени

Исследование модели. На рис. 24 приведены результаты прогона модели.

GPSS/PC Report file REPORT.GPS. (V 2, # 40550) 08-20-2002 13:44:24 page 1

START_TIME	END_TIME	BLOCKS	FACILITIES	STORAGES	FREE_MEMORY
0	1284073	19	2	0	12048

LINE	LOC	BLOCK_TYPE	ENTRY_COUNT	CURRENT_COUNT	RETRY
20	1	GENERATE	1044	0	0
30	2	SEIZE	1044	0	0
40	3	ADVANCE	1044	0	0
50	4	RELEASE	1044	0	0
60	5	TEST	1044	0	0
70	6	LINK	987	44	0
80	7	GENERATE	1	0	0
90	AAA	TEST	944	0	0
100	9	SEIZE	944	0	0
110	10	ADVANCE	944	1	0
120	11	ADVANCE	943	0	0
130	12	SEIZE	943	0	0
140	13	ADVANCE	943	0	0
150	14	RELEASE	943	0	0
160	15	UNLINK	943	0	0
170	16	RELEASE	943	0	0
180	17	TRANSFER	943	0	0
190	BBB	TERMINATE	943	0	0
200	OTK	TERMINATE	57	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAILABLE	OWNER	PEND	INTER	RETRY	DELAY
CPU	1987	0.044	28.98	1	0	0	0	0	0
DSK	944	0.999	1359.59	1	2	0	0	0	0

USER_CHAIN	CHAIN_SIZE	RETRY	AVE.CONT	ENTRIES	MAX	AVE.TIME
MEM	44	0	34.50	987	50	44882.59

Рис. 24. Результаты прогона модели рис. 23

Анализ результатов позволяет определить следующие характеристики:

- частота отказов – 5,7%;
- среднее содержимое ОП – 34,5;
- среднее время пребывания запроса – 4,5 сек.;
- коэффициент загрузки процессора – 0,044;
- коэффициент загрузки МД – 0,999.

Заключение

В настоящее время метод имитационного моделирования стал эффективным средством решения сложных задач анализа и проектирования информационных и вычислительных систем. Быстрое развитие информационных технологий и средств компьютерной техники вызвало появление ряда прикладных программ имитационного моделирования систем как дискретного, так и непрерывного типа. Широкое распространение для моделирования дискретных систем получил язык моделирования GPSS. В настоящее время в учебном процессе и инженерной практике используются системы моделирования GPSS/PC для операционной системы MSDOS и GPSS World для операционной системы Windows фирмы Minuteman Software.

Освоить методологию и инструментальные средства имитационного моделирования можно только при решении задач моделирования с использованием конкретных инструментальных средств. Поэтому в данном учебном пособии основное внимание уделено моделированию процессов функционирования систем с использованием пакета моделирования GPSS/PC.

Вхождение пользователя в мировую сеть Internet позволяет получать огромные потоки информации, в том числе и по различным версиям языка моделирования GPSS. Например, есть портал WWW.GPSS.RU, на котором находится информация о работах в области моделирования на GPSS. Использование этой информации позволяет найти как методический материал, так и примеры построения конкретных моделей.

В данном учебном пособии недостаточно внимания уделено статистической оценке достоверности результатов моделирования, планированию эксперимента. В системе GPSS/PC средства такой оценки ограничены. Однако в системе GPSS World есть специальные процедуры для решения этих задач. Этому вопросу при имитационном моделировании необходимо уделить особое внимание.

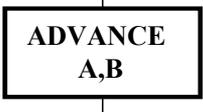
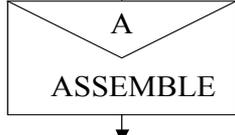
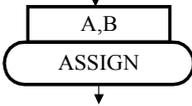
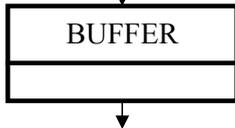
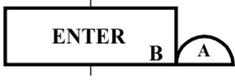
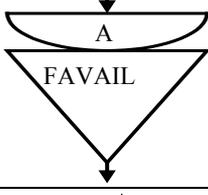
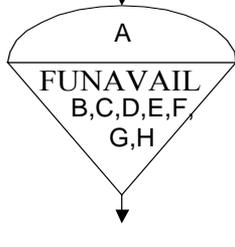
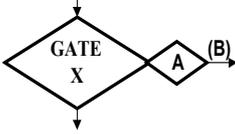
Приложение 1

Список стандартных числовых атрибутов

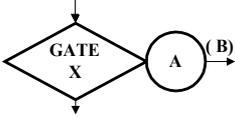
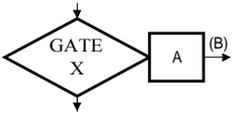
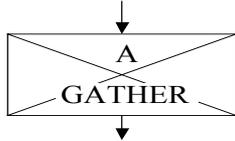
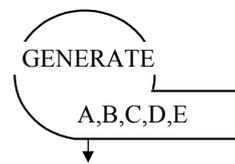
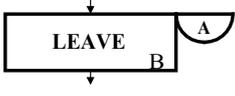
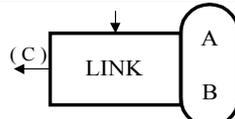
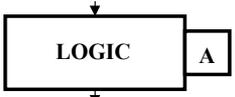
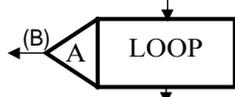
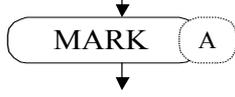
Элемент	СЧА	Краткое содержание
Блоки	N	Счетчик входов
	W	Счетчик текущего содержимого
Системные атрибуты	C1	Значение относительного времени
	AC1	Значение абсолютного времени
	RN	Случайное число. При использовании в качестве аргументов функции 0.000000-0.999999; в остальных случаях 000-999
	TG	Значение счетчика завершенный
Накопители	S	Текущее содержимое
	SA	Среднее содержимое
	SC	Счетчик числа входов
	SR	Коэффициент использования (в долях 1000)
	SM	Максимальное содержимое
	ST	Среднее время задержки на 1 емкости
Очереди.	Q	Текущее содержимое
	QA	Среднее содержимое
	QC	Счетчик числа входов
	QM	Максимальное содержимое
	QT	Среднее время пребывания (без QC)
	QX	Среднее время пребывания (без QZ)
	QZ	Счетчик числа нулевых входов
Переменные	BV	Значение булевой переменной
	V	Значение арифметической переменной
Приборы	F	Состояние прибора (1- занят, 0- свободен)
	FC	Счетчик числа занятий
	FR	Коэффициент использования (в долях 1000)
	FT	Среднее время задержки на одно занятие
	FI	Состояние прибора (1-захвачен, 0-не захвачен)
	FV	Состояние прибора (1-не доступен, 0-доступен)
Сохраняемые величины	X	Значение сохраняемой величины
	MX (a,b)	Текущее содержимое ячейки матрицы сохраняемых величин (строка a, столбец b)
Таблицы	TB	Средняя величина невзвешенных входов
	TC	Количество невзвешенных входов
	TD	Стандартное отклонение невзвешенных входов
Транзакты	P	Величина параметра
	PR	Приоритет
	M1	Время пребывания в модели
	MP	Время с момента входа в блок MARK
Цепь пользователя	CA	Среднее содержимое
	CC	Общее число входов
	CH	Текущее содержимое
	CM	Максимальное содержимое
	CT	Среднее время пребывания на 1 вход
Функция	FN	Значение функции

Приложение 2

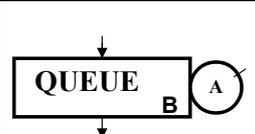
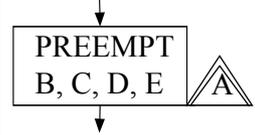
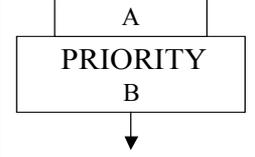
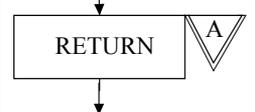
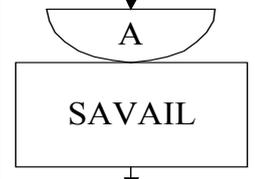
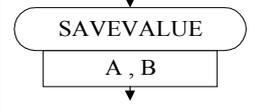
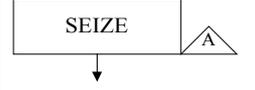
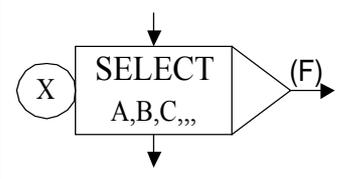
Основные блоки языка моделирования GPSS

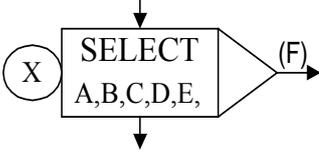
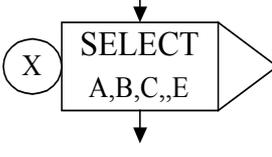
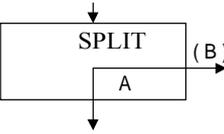
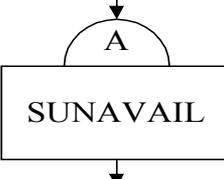
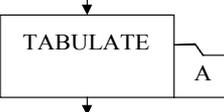
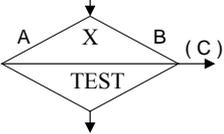
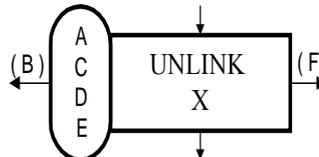
БЛОК	ОПЕРАЦИЯ	A	B	C
	ADVANCE (Задержать)	Среднее время	Модификатор разброса или функции	
	ASSEMBLE (Соединить)	Счетчик соединения		
	ASSIGN (Назначить)	№ параметра	Назначаемая величина	
	BUFFER (Возобновить просмотр)			
	DEPART (Покинуть очередь)	Имя очереди	Число освобождаемых единиц	
	ENTER (Войти)	Имя накопителя	Число занимаемых единиц	
	FAVAIL (Сделать прибор доступным)	Имя прибора		
	FUNAVAIL (Сделать прибор недоступным)	<p style="text-align: center;"><u>Поле операндов</u></p> <p>A- Имя прибора; B- режим {RE, CO }; C- номер альтернативного блока; D- номер параметра; E- режим {RE,CO}; F- номер альтернативного блока; G- режим {RE, CO }; H- номер альтернативного блока</p>		
	GATE {NI, I, NU, U,FV,FNV} (Впустить)	Имя прибора	Следующий блок при невыполнении условия	

Продолжение прил. 2

БЛОК	ОПЕРАЦИЯ	А	В	С
	GATE {SE,SF, SNE, SNF,SV,SNV} (Впустить)	Имя накопителя	Следующий блок при невыполнении условия	
	GATE {LS, LR } (Впустить)	Имя логического переключателя	Следующий блок при невыполнении условия	
	GATHER (Собрать)	Счетчик сборки		
	GENERATE (Генерировать)	Среднее время	Модификатор разброса или модификатор функции	Интервал смещения
			D -Ограничитель	E -Уровень приоритета
	LEAVE (Выйти)	Имя накопителя	Число единиц	
	LINK (Ввести в цепь)	Имя цепи пользователя	Критерий присоединения	Альтернативный блок
	LOGIC { I, R, S }	Имя логического переключателя	R- выключить; S- включить; I -переключить	
	LOOP (Организовать цикл)	№ параметра	Следующий блок, если параметр не равен 0	
	MARK (Отметить)	№ параметра		
	MATCH (Синхронизировать)	Адрес сопряженного блока MATCH		

Продолжение прил. 2

БЛОК	ОПЕРАЦИЯ	А	В	С
	QUEUE (Встать в очередь)	Имя очереди	Число единиц	
	PREEMPT (Захватить)	А- Имя прибора; В- Возможность захвата по приоритету {PR}; С- № блока для захваченного транзакта; D- № параметра у захваченного транзакта; Е- Возможность снятия с обслуживания {RE}		
	PRIORITY (Назначить приоритет)	Новое значение приоритета	Возможность возобновления просмотра [BUFFER]	
	RELEASE (освободить прибор)	Имя прибора		
	RETURN (Вернуть)	Имя прибора		
	SAVAIL (Сделать накопитель доступным)	Имя накопителя		
	SAVEVALUE (Сохранить значение)	Имя сохраняемой величины	Величина, которую следует сохранить	
	SEIZE (Занять прибор)	Имя прибора		
	SELECT {NI,NU,U,FV,FNV,SE,SF,SNE,SNF,SV,SNV,LS, LR} (логический режим)	<p><u>Поле операндов</u></p> <p>А-параметр, в который заносит номер элемента; В-нижний предел; С- верхний предел; D, Е- не используются; F- альтернативный выход</p>		

БЛОК	ОПЕРАЦИЯ	А	В	С
	SELECT {L, LE, G, GE, E, NE} (режим от- ношения)	<u>Поле операндов</u> А-параметр, в который заносят номер элемента; В-нижний предел; С- верхний предел; D, Е- сравниваемые СЧА; F- альтернативный выход		
	SELECT {MIN, MAX} режим поиска минимума, мак- симума)	<u>Поле операндов</u> А-параметр, в который заносят номер элемента; В-нижний предел; С- верхний предел; Е- групповое имя СЧА; D, F- не используются		
	SPLIT (Расщепить)	Число по- томков	Следующий блок для по- томков	Параметр, в который заносят по- рядковый номер
	SUNAVAIL (Сделать на- копитель не доступным)	Имя накопи- теля		
	TABULATE (Табулиро- вать)	Имя таблицы		
	TERMINATE (Завершить)	Счетчик числа завер- шений		
	TEST {G, GE, L, LE, E, NE} (Проверить)	Первая ве- личина	Вторая вели- чина	Следующий блок при невыполне- нии условия
	UNLINK {L, LE, G, GE, NE, E} (Вывести из цепи)	<u>Поле операндов</u> А- имя цепи пользователя; В-блок, в который входят выводимые транзак- ты; С -счетчик выводимых транзактов; D- № параметра; Е – аргумент парно- сти; F- альтернативный выход		

Приложение 3

Процедуры формирования непрерывных случайных величин в GPSS World

№	Вид распределения, Синтаксис	Плотность распределения $f(x)$	Математическое ожидание	Дисперсия
1	Экспоненциальное Синтаксис: Real = EXPONENTIAL(N, λ , β) N – номер датчика случайных чисел RNj	$f(x) = \begin{cases} \frac{1}{\beta} e^{-\frac{(x-\lambda)}{\beta}}, & \beta \geq 0 \\ 0, & \beta < 0 \end{cases}$	$\beta + \lambda$	β^2
2	Нормальное Синтаксис Real = NORMAL(N, μ , σ)	$f(x) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}$	μ	σ^2
3	Логарифмически-нормальное Синтаксис Real = LOGNORMAL(N, λ , μ , σ)	$f(x) = \begin{cases} \frac{e^{-\frac{(\ln(x-\lambda)-\mu)^2}{2\sigma^2}}}{(x-\lambda)\sqrt{2\pi\sigma^2}}, & x > \lambda \\ 0, & \text{вне диапазона} \end{cases}$	$e^{\mu + \frac{\sigma^2}{2}} + \lambda$	$e^{2\mu + \sigma^2} (e^{\sigma^2} - 1)$
4	Треугольное (a=Min; b=Max; c=Mode) Синтаксис Real = TRIANGULAR(N, a, b, c)	$f(x) = \begin{cases} \frac{2(x-a)}{(b-a)(c-a)}, & a \leq x \leq c \\ \frac{2(b-x)}{(b-a)(b-c)}, & c < x \leq b \\ 0, & \text{вне диапазона} \end{cases}$	$\frac{a+b+c}{3}$	$\frac{a^2 + b^2 + c^2 - ab - ac - bc}{18}$

Окончание прил. 3

№	Вид распределения, Синтаксис	Плотность распределения $f(x)$	Математическое ожидание	Дисперсия
5	Равномерное на интервале a, b Синтаксис Real = UNIFORM(N, a, b)	$f(x) = \begin{cases} \frac{1}{(b-a)}, & a \leq x \leq b \\ 0, & \text{вне интервала} \end{cases}$	$\frac{(a+b)}{2}$	$\frac{(b-a)^2}{12}$
6	Бета – распределение Синтаксис Real = БЕТА(N, $\lambda_1, \lambda_2, \alpha_1, \alpha_2$) $\lambda_1 = Min$ $\lambda_2 = Max$ α_1, α_2 – параметры.	$f(x) = \begin{cases} \frac{(x-\lambda_1)^{\alpha_1-1}(\lambda_2-x)^{\alpha_2-1}}{(\lambda_2-\lambda_1)^{\alpha_1+\alpha_2-1} B(\alpha_1, \alpha_2)}, \\ 0, & \text{вне интервала} \end{cases}$ $B(\alpha_1, \alpha_2) = \int_0^1 t^{\alpha_1-1} (1-t)^{\alpha_2-1} dt.$	$\frac{\alpha_1(\lambda_2-\lambda_1)}{\alpha_1+\alpha_2} + \lambda_1$	$\frac{(\lambda_2-\lambda_1)^2(\alpha_1\alpha_2)}{(\alpha_1+\alpha_2)^2(\alpha_1+\alpha_2+1)}$
7	Вейбулла Синтаксис Real = WEIBULL(N, λ, β, α)	$f(x) = \begin{cases} \frac{\alpha(x-\lambda)^{(\alpha-1)}}{\beta^\alpha} e^{-\left(\frac{x-\lambda}{\beta}\right)^\alpha}, & x > \lambda \\ 0, & \text{вне интервала} \end{cases}$	$\frac{\beta}{\alpha} \Gamma\left(\frac{1}{\alpha}\right) + \lambda$	$\frac{\beta^2}{\alpha} \left(2\Gamma\left(\frac{2}{\alpha}\right) - \frac{1}{\alpha} \left(\Gamma\left(\frac{1}{\alpha}\right) \right)^2 \right)$ $\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} e^{-t} dt$
8	Гамма Синтаксис Real = GAMMA(N, λ, β, α)	$f(x) = \begin{cases} \frac{\beta^{-\alpha} (x-\lambda)^{\alpha-1} e^{-\frac{(x-\lambda)}{\beta}}}{\Gamma(\alpha)}, & x > \lambda \\ 0, & \text{вне диапазона} \end{cases}$ $\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} e^{-t} dt$	$\alpha \beta$	$\alpha \beta^2$

Приложение 4

Процедуры формирования дискретных случайных величин в GPSS World

№	Вид распределения, Синтаксис	Вероятность распределения $f(x)$	Математическое ожидание	Дисперсия
1	Биномиальное Синтаксис Integer = BINOMIAL(N, t, P)	$f(x) = \begin{cases} \frac{t!}{x!(t-x)!} P^x (1-P)^{t-x}, & x \in (0, 1, 2, \dots, t) \\ 0 & \end{cases}$	$t P$	$t P (1 - P)$
2	Дискретное равномерное Синтаксис Integer = DUNIFORM(N, Min, Max) i=Min: j=Max	$f(x) = \begin{cases} \frac{1}{j-i+1}, & x \in (i, i+1, \dots, t) \\ 0, & \text{вне интервала} \end{cases}$	$\frac{i+j}{2}$	$\frac{(j-i+1)^2}{12}$
3	Геометрическое Синтаксис Integer = GEOMETRIC(N, P)	$f(x) = \begin{cases} P (1-P)^x, & x \in (0, 1, \dots) \\ 0, & \text{вне интервала} \end{cases}$	$\frac{1-P}{P}$	$\frac{1-P}{P^2}$
4	Отрицательное биномиальное Синтаксис Integer = NEGBINOM(N, S, P)	$f(x) = \begin{cases} \frac{(S+x-1)!}{x!(S-1)!} P^S (1-P)^x, & x \in (0, 1, \dots) \\ 0, & \text{вне интервала} \end{cases}$	$\frac{S(1-P)}{P}$	$\frac{S(1-P)}{P^2}$
5	Пуассона Синтаксис Integer = POISSON(N, Mean) $\lambda = Mean$	$f(x) = \frac{\lambda^x e^{-\lambda}}{x!}, x \in (0, 1, \dots)$	λ	λ

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Советов Б.Я., Яковлев С.А. Моделирование систем: Учебник. -М.: Высш. шк.,2001. – 343 с.
2. Советов Б.Я., Яковлев С.А. Моделирование систем. Практикум: Учеб. пособие. – М.: Высш. шк., 1999. – 224 с.
3. Шрайбер Т. Дж. Моделирование на GPSS.- М.: Машиностроение,1980. – 592 с.
4. Васильев А. И. Имитационное моделирование информационных и вычислительных систем с использованием языка моделирования GPSS: Метод. указания к лабораторным работам. - Владивосток: ДВГТУ, 1998.
5. Марков А. А. Моделирование информационно-вычислительных процессов: Учеб. пособие. – М.: Изд-во МГТУ им. Н. Э. Баумана, 1999.
6. GPSS/PC general purpose simulation. Reference Manual.–Minuteman software. P. O. Box 171. Stow, Massachusetts 01775, 1986.
7. Толуев Ю. И., Грундспенькис Я. А. Статистическое моделирование на ЭВМ: Конспект лекций. – Рига: Рижский политехн. ин-т, 1986.
8. Томашевский В. Н., Жданова Е. Г. Имитационное моделирование в среде GPSS. – М.: Бестселлер, 2003. – 416 с.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. ОСНОВЫ СТАТИСТИЧЕСКОГО ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ	4
1.1. ОСНОВНЫЕ ЭТАПЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ.....	4
1.2. МОДЕЛИРОВАНИЕ СЛУЧАЙНЫХ ОБЪЕКТОВ	6
1.3. СБОР И СТАТИСТИЧЕСКАЯ ОБРАБОТКА ДАННЫХ МОДЕЛИРОВАНИЯ	11
1.4. ОПРЕДЕЛЕНИЕ ДЛИТЕЛЬНОСТИ МОДЕЛИРОВАНИЯ И КОЛИЧЕСТВА РЕАЛИЗАЦИЙ.....	15
1.5. ИМИТАЦИОННЫЕ ЭКСПЕРИМЕНТЫ ДЛЯ ИССЛЕДОВАНИЯ И ОПТИМИЗАЦИИ МОДЕЛИ.....	19
2. ПОСТРОЕНИЕ АЛГОРИТМИЧЕСКОЙ МОДЕЛИ НА GPSS/PC	23
2.1. КРАТКИЕ СВЕДЕНИЯ О ЯЗЫКЕ МОДЕЛИРОВАНИЯ	23
2.2. ФОРМИРОВАНИЕ РЕЗУЛЬТАТОВ МОДЕЛИРОВАНИЯ В GPSS/PC	35
3. ПОСТРОЕНИЕ МОДЕЛЕЙ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ НА GPSS	40
3.1. МОДЕЛЬ ОДНОКАНАЛЬНОЙ СМО С ОЖИДАНИЕМ	41
3.2. МОДЕЛЬ МНОГОКАНАЛЬНОЙ СМО С ОБЩЕЙ ОЧЕРЕДЬЮ.....	43
3.3. МОДЕЛЬ МНОГОКАНАЛЬНОЙ СМО С РАЗДЕЛЬНЫМИ ОЧЕРЕДЯМИ	44
3.4. МОДЕЛИРОВАНИЕ СМО С РАЗЛИЧНЫМИ ДИСЦИПЛИНАМИ ВЫБОРА ЗАЯВОК ИЗ ОЧЕРЕДИ	46
3.5. МОДЕЛЬ ОДНОКАНАЛЬНОЙ СМО С ВЫБОРОМ ЗАЯВОК В ПОРЯДКЕ ПОСТУПЛЕНИЯ.....	48
3.6. МОДЕЛЬ ОДНОКАНАЛЬНОЙ СМО С ВЫБОРОМ ЗАЯВОК ПО ДИНАМИЧЕСКОМУ ПРИОРИТЕТУ	49
3.7. МОДЕЛЬ ОДНОКАНАЛЬНОЙ СМО С ОБСЛУЖИВАНИЕМ ЗАЯВОК В СЛУЧАЙНОМ ПОРЯДКЕ.....	50
3.8. МОДЕЛЬ ОДНОКАНАЛЬНОЙ СМО С ОГРАНИЧЕННЫМ ВРЕМЕНЕМ ОЖИДАНИЯ	51
3.9. МОДЕЛЬ МНОГОКАНАЛЬНОЙ СМО С РАЗДЕЛЬНЫМИ ОЧЕРЕДЯМИ С ИСПОЛЬЗОВАНИЕМ ЦЕПИ ПОЛЬЗОВАТЕЛЯ.....	53
4. РЕАЛИЗАЦИЯ МОДЕЛИ НА ЭВМ	58
4.1. КОМАНДЫ GPSS/PC И ТЕХНОЛОГИЯ РАБОТЫ С ПАКЕТОМ.....	58
4.2. ПРОГОН МОДЕЛИ И НАБЛЮДЕНИЕ ЗА МОДЕЛИРОВАНИЕМ	60
4.3. УПРАВЛЯЮЩИЕ ОПЕРАТОРЫ GPSS/PC	65
4.4. ТОЧНОСТЬ РЕЗУЛЬТАТОВ МОДЕЛИРОВАНИЯ	68
5. ПРИМЕРЫ МОДЕЛЕЙ	74
5.1. МОДЕЛЬ СПРАВОЧНОЙ ТЕЛЕФОННОЙ СЕТИ.....	74
5.2. МОДЕЛЬ ИНФОРМАЦИОННО-ПОИСКОВОЙ БИБЛИОГРАФИЧЕСКОЙ СИСТЕМЫ	77
5.3. МОДЕЛЬ ИНФОРМАЦИОННОЙ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ	81
ЗАКЛЮЧЕНИЕ	86
ПРИЛОЖЕНИЕ 1. СПИСОК СТАНДАРТНЫХ ЧИСЛОВЫХ АТТРИБУТОВ	87
ПРИЛОЖЕНИЕ 2. ОСНОВНЫЕ БЛОКИ ЯЗЫКА МОДЕЛИРОВАНИЯ GPSS	88
ПРИЛОЖЕНИЕ 3. ПРОЦЕДУРЫ ФОРМИРОВАНИЯ НЕПРЕРЫВНЫХ СЛУЧАЙНЫХ ВЕЛИЧИН В GPSS WORLD	92
ПРИЛОЖЕНИЕ 4. ПРОЦЕДУРЫ ФОРМИРОВАНИЯ ДИСКРЕТНЫХ СЛУЧАЙНЫХ ВЕЛИЧИН В GPSS WORLD	94
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	95