

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное  
учреждение высшего профессионального образования  
«Оренбургский государственный университет»

Кафедра программного обеспечения вычислительной  
техники и автоматизированных систем

В. В. Паничев

# ОЦЕНКА ЭФФЕКТИВНОСТИ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ В СРЕДЕ GPSS WORLD

Рекомендовано к изданию Редакционно-издательским советом федерального государственного бюджетного образовательного учреждения высшего профессионального образования «Оренбургский государственный университет» в качестве методических указаний для студентов, обучающихся по программам высшего профессионального образования по направлению подготовки 231000.68 Программная инженерия

Оренбург  
2013

УДК 519.876.5(076.5)  
ББК 22.18я7  
П16

Рецензент – профессор, доктор экономических наук В. Н. Шепель

Паничев, В.В.

П16 Оценка эффективности систем массового обслуживания в среде GPSS World: методические указания к выполнению лабораторных работ / В.В. Паничев; Оренбургский гос. ун-т. – Оренбург: ОГУ, 2013. – 83 с.

Методические указания к выполнению лабораторных работ по дисциплинам «Моделирование» и «Теория эксперимента» предназначены для оказания методической помощи при выполнении лабораторных работ. Дисциплина «Моделирование» входит в состав базовой части, а дисциплина «Теория эксперимента» в состав вариативной части общенаучного цикла дисциплин магистрантов очной формы обучения направления 231000.68 Программная инженерия.

В методических указаниях изложены теоретические сведения по системам массового обслуживания и по системе GPSS World, применяемой для моделирования процессов функционирования СМО, цели, задания и методика выполнения лабораторных работ по дисциплинам: «Моделирование» и «Теория эксперимента».

УДК 519.876.5(076.5)  
ББК 22.18я7

© Паничев В.В., 2013  
© ОГУ, 2013

## Содержание

Введение .....	4
1 Теоретические сведения о системах массового обслуживания.....	5
1.1 Общие положения о СМО .....	5
1.2 Аналитические модели систем массового обслуживания .....	9
2 Теоретические сведения о среде моделирования <i>GPSS World</i> .....	20
2.1 Принципы организации системы <i>GPSS World</i> .....	20
2.2 Среда и функциональная структура языка моделирования.....	22
2.2.1 Введение в систему <i>GPSS</i> .....	22
2.2.2 Описание объектов <i>GPSS World</i> .....	30
2.2.3. Форматы описания блоков и операторов <i>GPSS</i> .....	36
2.2.4. Стандартные числовые атрибуты.....	38
2.3 Методика применения языка <i>GPSS</i> для имитации СМО.....	40
2.3.1 Имитация функционирования одноканальной СМО .....	41
3 Лабораторный практикум по дисциплине «Моделирование».....	46
3.1 Лабораторная работа № 1 .....	46
3.2 Задание на лабораторную работу № 2 .....	49
3.3 Задание на лабораторную работу № 3 .....	51
3.4 Задание на лабораторную работу № 4 .....	54
4 Лабораторный практикум по дисциплине «Теория эксперимента».....	58
4.1 Лабораторная работа № 5 .....	58
4.2 Лабораторная работа № 6 .....	69
Список использованных источников .....	83

## Введение

Методические указания предназначены для выполнения лабораторных работ по дисциплинам «Моделирование» и «Теория эксперимента».

Задачи, которые решаются при выполнении работ соответствуют рабочим программам дисциплин:

- обучение технологии построения и использования моделей при исследовании различных явлений;

- обучение методам количественного и качественного анализа результатов исследований сложных систем.

Выполнение лабораторных работ позволяет формировать элементы следующих компетенций в соответствии с ФГОС ВПО и ООП ВПО по данному направлению подготовки:

а) общекультурных (ОК):

- способность к самостоятельному обучению новым методам исследования, к изменению научного и научно – производственного профиля своей профессиональной деятельности (ОК-2);

- способность самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях, непосредственно не связанных со сферой деятельности (ОК-6);

б) профессиональных (ПК):

- способность к проектной деятельности в профессиональной сфере на основе системного подхода, умения строить и использовать модели для описания и прогнозирования различных явлений, осуществлять их качественный и количественный анализ (ПК-6).

# 1 Теоретические сведения о системах массового обслуживания

## 1.1 Общие положения о СМО

*Системой массового обслуживания* (СМО) называют любую систему, в которой обслуживаются какие-либо заявки или требования, поступающие в случайные моменты времени. Примеры СМО: телефонная станция; бюро ремонта; билетная касса; парикмахерская; ЭВМ. Изучением случайных процессов в системах массового обслуживания занимается теория массового обслуживания.

Любое устройство, непосредственно занимающееся обслуживанием заявок, называется *каналом обслуживания*. СМО бывают как одноканальными, так и многоканальными. Пример одноканальной СМО - билетная касса с одним кассиром; пример многоканальной – та же касса с несколькими кассирами.

Различают СМО *с отказами* и СМО *с очередью*. В СМО с отказами заявка, пришедшая в момент, когда все каналы заняты, получает отказ, покидает СМО и в процессе ее дальнейшей работы не участвует. В СМО с очередью заявка, пришедшая в момент занятости всех каналов, не покидает СМО, а становится в очередь и ждет, пока не освободится какой-нибудь канал. Число мест в очереди  $m$  может быть как ограниченным, так и неограниченным. При  $m=0$  СМО с очередью превращается в СМО с отказами. Очередь может иметь ограничения не только по количеству стоящих в ней заявок (по длине очереди), но и по времени ожидания (такие СМО называются «системами с нетерпеливыми клиентами»).

СМО с очередью различаются не только по ограничениям очереди, но и по *дисциплине обслуживания*: обслуживаются ли заявки в порядке поступления, или в случайном порядке, или же некоторые заявки обслуживаются вне очереди (так называемые «СМО с приоритетом»). Приоритет может иметь несколько градаций или рангов.

Аналитическое исследование СМО проще в случае, если все потоки событий являются простейшими (стационарными, пуассоновскими). Это значит, что интервалы времени между событиями в потоках имеют показательное

распределение с параметром, равным интенсивности соответствующего потока. Для СМО это допущение означает, что как поток заявок, так и поток обслуживаний - простейшие. Под потоком обслуживания понимают поток заявок, обслуживаемых одна за другой одним непрерывно занятым каналом. Этот поток оказывается простейшим, только, если время обслуживания заявки  $T_{обсл}$  представляет собой случайную величину, имеющую показательное распределение. Параметр этого распределения  $\mu$  есть величина, обратная среднему времени обслуживания:  $\mu = 1 / t_{обс}$ , где  $t_{обс} = M [T_{обсл}]$ . Условимся в дальнейшем всякую СМО, в которой все потоки простейшие, называть *простейшей* СМО. Если все потоки событий простейшие, то процесс, протекающий в СМО, представляет собой случайный марковский процесс с дискретными состояниями и непрерывным временем. При выполнении некоторых условий для этого процесса существует финальный стационарный режим, при котором как вероятности состояний, так и другие характеристики процесса не зависят от времени [1].

Задачи теории массового обслуживания — нахождение вероятностей различных состояний СМО, а также установление зависимости между заданными параметрами (числом каналов  $n$ , интенсивностью потока заявок  $\lambda$ , распределением времени обслуживания и т.д.) и *характеристиками эффективности* работы СМО. В качестве таких характеристик могут рассматриваться:

среднее число заявок  $A$ , обслуживаемое СМО в единицу времени, или *абсолютная пропускная способность* СМО;

вероятность обслуживания поступившей заявки  $Q$  или *относительная пропускная способность* СМО  $Q = A / \lambda$ ;

вероятность отказа  $P_{отк}$ , т.е. вероятность того, что поступившая заявка не будет обслужена, получит отказ;  $P_{отк} = 1 - Q$ ;

среднее число заявок в СМО (обслуживаемых или ожидающих в очереди)  $z$ ;

среднее число заявок в очереди  $\check{r}$ ;

среднее время пребывания заявки в СМО (в очереди или под обслуживанием)

$t_{сист}$ ;

среднее время пребывания заявки в очереди  $t_{оч}$ ;

среднее число занятых каналов  $k$ .

В общем случае все эти характеристики зависят от времени. Но многие СМО работают в неизменных условиях достаточно долгое время, и поэтому для них успевает установиться режим, близкий к стационарному.

Для любой открытой СМО в предельном стационарном режиме среднее время пребывания заявки в системе  $t_{cuct}$  выражается через среднее число заявок в системе с помощью формулы Литтла:

$$\bar{t}_{cuct} = \bar{z} / \lambda, \quad (1.1)$$

где  $\lambda$  — интенсивность потока заявок.

СМО называется открытой, если интенсивность поступающего на нее потока заявок не зависит от состояния самой СМО.

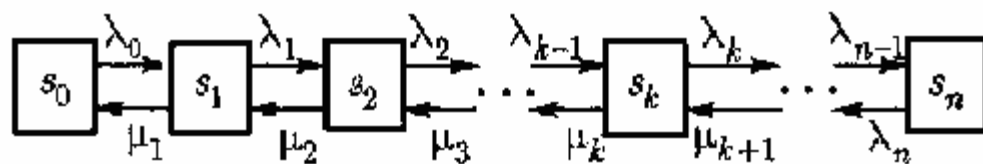


Рисунок 1.1 – Граф состояний СМО

Аналогичная формула (называемая также формулой Литтла) связывает среднее время пребывания заявки в очереди  $t_{оч}$  и среднее число  $r$  заявок в очереди:

$$\bar{t}_{оч} = r / \lambda. \quad (1.2)$$

Формулы Литтла позволяют вычислять не обе характеристики эффективности (среднее время пребывания и среднее число заявок), а только какую-нибудь одну из них.

Специально подчеркнем, что формулы (1.1) и (1.2) справедливы для любой открытой СМО (одноканальной, многоканальной, при любых видах потоков заявок и обслуживаний); единственное требование к потокам заявок и обслуживаний - чтобы они были стационарными.

Аналогично универсальное значение для открытых СМО имеет формула, выражающая среднее число занятых каналов  $k$  через абсолютную пропускную способность  $\bar{k} = A/\mu$ , где  $\mu$  - интенсивность потока обслуживаний.

Очень многие задачи теории массового обслуживания, касающиеся простейших СМО, решаются с помощью схемы гибели и размножения. Если граф состояний СМО может быть представлен в виде, показанном на рисунке 1.1, то финальные вероятности состояний выражаются формулами [1]:

$$\begin{aligned}
 p_0 &= \left\{ 1 + \frac{\lambda_0}{\mu_1} + \frac{\lambda_0 \lambda_1}{\mu_1 \mu_2} + \dots + \frac{\lambda_0 \lambda_1 \dots \lambda_{k-1}}{\mu_1 \mu_2 \dots \mu_k} + \dots + \frac{\lambda_0 \lambda_1 \dots \lambda_{n-1}}{\mu_1 \mu_2 \dots \mu_n} \right\}^{-1} \\
 p_1 &= \frac{\lambda_0}{\mu_1} p_0; p_2 = \frac{\lambda_0 \lambda_1}{\mu_1 \mu_2} p_0; \dots; p_k = \frac{\lambda_0 \lambda_1 \dots \lambda_{k-1}}{\mu_1 \mu_2 \dots \mu_k} p_0; \dots; \\
 p_n &= \frac{\lambda_0 \lambda_1 \dots \lambda_{n-1}}{\mu_1 \mu_2 \dots \mu_n} p_0.
 \end{aligned} \tag{1.3}$$

При выводе формул для среднего числа заявок (в очереди или в системе) широко используется прием дифференцирования рядов, состоящий в следующем. Если  $x < 1$ , то

$$\sum_{k=1}^{\infty} kx^k = x \sum_{k=1}^{\infty} \frac{d}{dx} x^k = x \frac{d}{dx} \sum_{k=1}^{\infty} x^k = \frac{d}{dx} \frac{x}{1-x} = \frac{x}{(1-x)^2}$$

и окончательно

$$\sum_{k=1}^{\infty} kx^k = \frac{x}{(1-x)^2}. \tag{1.4}$$

Ниже приведем без вывода ряд формул, выражающих финальные вероятности состояний и характеристики эффективности для некоторых часто встречающихся типов СМО.



## 1.2 Аналитические модели систем массового обслуживания

**Простейшая СМО с отказами** (задача Эрланга). На  $n$ -канальную СМО с отказами (рисунке 1.2) поступает простейший поток заявок с интенсивностью  $\lambda$ , время обслуживания — показательное с параметром  $\mu$ . За время наблюдения  $T$  в систему может поступить  $N$  заявок, из них будет обслужено  $N_0$  и потеряно  $N_1$  [3].

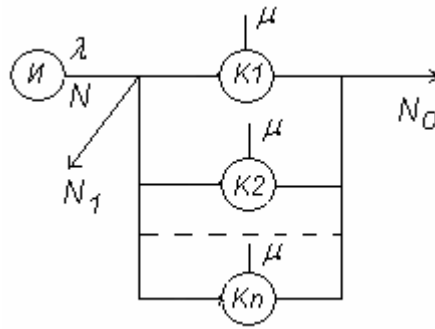


Рисунок 1.2 –  $n$ -канальная СМО с отказами

Состояния СМО определяются числом заявок, находящихся в СМО (в силу отсутствия очереди, оно совпадает с числом занятых каналов):

$S_0$  – все каналы СМО свободны;

$S_1$  – занят один канал, остальные свободны; ...;

$S_k$  – занято  $k$  каналов, остальные свободны ( $1 \leq k \leq n$ ); ...;

$S_n$  – заняты все  $n$  каналов.

Финальные вероятности состояний выражаются формулами Эрланга:

$$p = \left\{ 1 + \frac{\rho}{1!} + \frac{\rho^2}{2!} + \dots + \frac{\rho^n}{n!} \right\}^{-1}; \quad p_k = \frac{\rho^k}{k!} p_0 \quad (k = 1, 2, \dots, n), \quad (1.5)$$

где  $\rho = \lambda / \mu$ .

Характеристики эффективности:

$$A = \lambda(1 - p_n); \quad Q = 1 - p_n; \quad P_{отк} = p_n; \quad \bar{k} = \rho(1 - p_n). \quad (1.6)$$

При больших значениях  $n$  вероятности состояний (5.33) удобно вычислять через табулированные функции:

$$P(m, a) = \frac{a^m}{m!} e^{-a} \quad (\text{распределение Пуассона}) \quad (1.7)$$

и

$$R(m, a) = \sum_{k=0}^m \frac{a^k}{k!} e^{-a}, \quad (1.8)$$

с помощью которых можно найти вероятность

$$P(m, a) = R(m, a) - R(m-1, a). \quad (1.9)$$

Пользуясь этими функциями, формулы Эрланга (1.5) можно переписать в виде

$$p_k = P(k, \rho) / R(n, \rho) \quad (k = 0, 1, \dots, n), \quad (k = 0, 1, \dots, n). \quad (1.10)$$

**Простейшая одноканальная СМО с неограниченной очередью.** На одноканальную СМО (рисунке 1.3) поступает простейший поток заявок с интенсивностью  $\lambda$ . Время обслуживания — показательное с параметром  $\mu = 1/t_{\text{обсл}}$ . Длина очереди не ограничена [3].

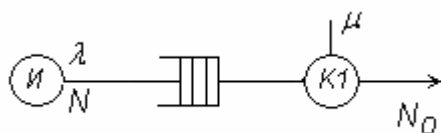


Рисунок 1.3 – Одноканальная СМО с очередью

Финальные вероятности существуют только при  $\rho = \lambda / \mu < 1$  (при  $\rho \geq 1$  очередь растет неограниченно). Состояния СМО нумеруются по числу заявок в СМО, находящихся в очереди и на обслуживании:

$S_0$  — СМО свободна;

$S_1$  — канал занят, очереди нет;

$S_2$  — канал занят, заявка стоит в очереди; ...;

$S_k$  — канал занят,  $k-1$  заявок стоят в очереди; ....

Финальные вероятности состояний выражаются формулами [1]:

$$P_0 = 1 - \rho, \quad P_k = \rho^k (1 - \rho) \quad (k = 1, 2, \dots), \quad (1.11)$$

где  $\rho = \lambda / \mu < 1$ .

Характеристики эффективности СМО:

$$A = \lambda; \quad Q = 1; \quad P_{отк} = 0;$$

$$\bar{Z} = \frac{\mu}{1 - \rho}; \quad \bar{r} = \frac{\rho^2}{1 - \rho}; \quad \bar{t}_{сум} = \frac{\rho}{\lambda(1 - \rho)}; \quad \bar{t}_{оч} = \frac{\rho^2}{\lambda(1 - \rho)}.$$

Здесь среднее число занятых каналов (или вероятность того, что канал занят)

$$\bar{k} = \lambda / \mu = \rho.$$

**Простейшая одноканальная СМО с ограниченной очередью.** На одноканальную СМО поступает простейший поток заявок с интенсивностью  $\lambda$ ; время обслуживания — показательное с параметром  $\mu = 1 / \bar{t}_{обсл}$ . В очереди  $m$  мест. Если заявка приходит в момент, когда все места в очереди заняты, она получает отказ и покидает СМО. Состояния СМО:

$s_0$  — СМО свободна;

$s_1$  — канал занят, очереди нет;

$s_2$  — канал занят, одна заявка стоит в очереди; ...;

$s_k$  — канал занят,  $k-1$  заявок стоят в очереди; ...;

$s_{m+1}$  — канал занят,  $m$  заявок стоят в очереди.

Финальные вероятности состояний существуют при любом  $\rho = \lambda / \mu$  и равны:

$$p_0 = \frac{1 - \rho}{1 - \rho^{m+2}}; \quad P_k = \rho^k p_0 \quad (k = 1, \dots, m + 1). \quad (1.12)$$

Характеристики эффективности СМО:

$$A = \lambda(1 - p_{m+1}); \quad Q = 1 - p_{m+1}; \quad P_{отк} = p_{m+1}.$$

Среднее число занятых каналов (вероятность того, что канал занят)

$$\bar{k} = 1 - p_0. \quad (1.13)$$

Среднее число заявок в очереди

$$\bar{r} = \frac{\rho^2 [1 - \rho^m (m + 1 - m\rho)]}{(1 - \rho^{m+2})(1 - \rho)}. \quad (1.14)$$

Среднее число заявок в СМО

$$\bar{z} = \bar{r} + \bar{k}.$$

По формуле Литтла среднее время пребывания заявки в СМО равно

$$\bar{t}_{сист} = \bar{z} / \lambda; \quad \bar{t}_{оч} = \bar{r} / \lambda.$$

### **Простейшая многоканальная СМО с неограниченной очередью.**

На  $n$ -канальную СМО (рисунок 1.4) поступает простейший поток заявок с интенсивностью  $\lambda$ ; время обслуживания одной заявки - показательное с параметром  $\mu$ .

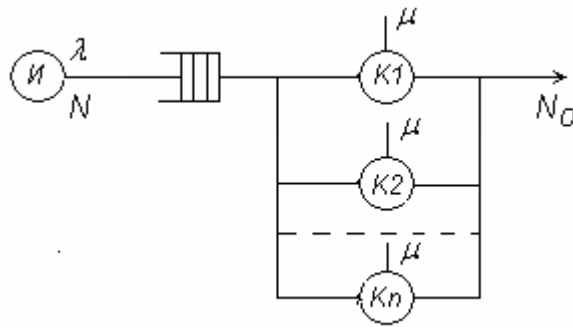


Рисунок 1.4 – Многоканальная СМО с очередью [3]

Финальные вероятности существуют только при  $\rho/n = \chi < 1$ , где  $\rho = \lambda/\mu$ .

Состояния СМО нумеруются по числу заявок в СМО:

$s_0$  — СМО свободна;

$s_1$  — занят один канал;

$s_k$  — занято  $k$  каналов ( $1 \leq k \leq n$ );

$s_n$  — заняты все  $n$  каналов;

$s_{n+1}$  — заняты все  $n$  каналов, одна заявка стоит в очереди;

$s_{n+r}$  — заняты все  $n$  каналов,  $r$  заявок стоят в очереди.

Финальные вероятности состояний выражаются формулами:

$$P_0 = \left\{ 1 + \frac{\rho}{1!} + \dots + \frac{\rho^n}{n!} + \frac{\rho^{n+1}}{n * n!} \frac{1}{1 - \chi} \right\}^{-1};$$

$$P_k = \frac{\rho^k}{k!} p_0 (1 \leq k \leq n); \quad p_{n+r} = \frac{\rho^{n+r}}{n^r * n!} p_0 (r \geq 1).$$

С помощью функций  $P(m, a)$  и  $R(m, a)$  и этой формулы могут быть приведены к виду

$$p_k = \frac{P(k, \rho)}{R(n, \rho) + P(n, \rho) \frac{\chi}{1 + \chi}} (k = 0, \dots, n);$$

$$P_{n+r} = \chi^r p_n (r = 1, 2, \dots).$$

Характеристики эффективности СМО:

$$\bar{r} = \rho^{n+1} p_0 / [n n! (1 - \chi)^2] = \chi * p_n / (1 - \chi)^2;$$

$$\bar{z} = \bar{r} + \bar{k} = \bar{r} + \rho;$$

$$\bar{t}_{\text{сум}} = \bar{z} / \lambda; \quad \bar{t}_{\text{оч}} = \bar{r} / \lambda.$$

**Простейшая многоканальная СМО с ограниченной очередью.** Условия и нумерация состояний те же, с той разницей, что число  $m$  мест в очереди ограничено. Финальные вероятности состояний существуют при любых  $\lambda$  и  $\mu$  и выражаются формулами:

$$p_0 = \left\{ 1 + \frac{\rho}{1!} + \dots + \frac{\rho^n}{n!} + \frac{\rho^{n+1}}{n * n!} \frac{1 - \chi^m}{1 - \chi} \right\}^{-1};$$

$$p_k = \frac{\rho^k}{k!} p_0 \quad (1 \leq k \leq n);$$

$$p_{n+r} = \frac{\rho^{n+r}}{n^r * n!} p_0 \quad (1 \leq r \leq m),$$

$$\chi = \rho / n = \lambda / (n\mu).$$

**Простейшая одноканальная замкнутая СМО с ожиданием.** Если в разомкнутой системе источник обладает неограниченным числом требований (заявок), то в отличие от нее источник замкнутой системы имеет ограниченное число требований. В замкнутой системе требование, покинувшее источник после обслуживания возвращается в источник, и в дальнейшем может неоднократно циркулировать по цепи: источник, очередь, канал обслуживания и опять источник (рисунок 1.5).

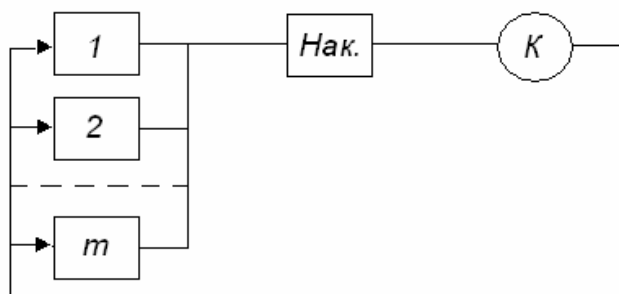


Рисунок 1.5 –Одноканальная замкнутая СМО с ожиданием

**Пример замкнутой СМО с ожиданием.** Один наладчик обслуживает  $m$  компьютеров, у каждого из которых периодически возникает сбой (отказ) в работе, интервалы между отказами распределены по экспоненциальному закону с интенсивностью  $\lambda$ . Если в момент отказа наладчик свободен, он сразу приступает к наладке ПК, если – нет, то ПК становится в очередь на восстановление. Длительность восстановления ПК распределена также по экспоненциальному закону с интенсивностью обслуживания  $\mu$ . Потенциальное число заявок на обслуживание постоянное и равно  $m$ . Интенсивность входного потока требований зависит от числа работающих объектов (ПК) в источнике. Требуется найти характеристики эффективности замкнутой системы:

пропускную способность ( $A$ ) системы (среднее число объектов восстанавливаемых наладчиком в единицу времени);

среднее число неисправных объектов ( $\bar{w}$ );

среднее число объектов в очереди ( $\bar{r}$ );

вероятность занятости наладчика ( $P_{зан}$ ).

Состояния СМО нумеруются по числу заявок (неисправных объектов) в СМО:

$s_0$  — все объекты (ПК) исправны, наладчик свободен;

$s_1$  — 1 объект неисправен, наладчик занят;

$s_2$  — 2 объекта неисправны, из них 1 объект налаживается;

$s_k$  —  $k$  объектов неисправны, из них 1 объект налаживается,  $k-1$  объектов

стоят в очереди;

$s_m$  — все  $m$  объектов неисправны, из них 1 объект налаживается,  $m-1$  неисправных объектов стоят в очереди.

Для определения финальных вероятностей состояний системы составим граф состояний (рисунок 1.6).

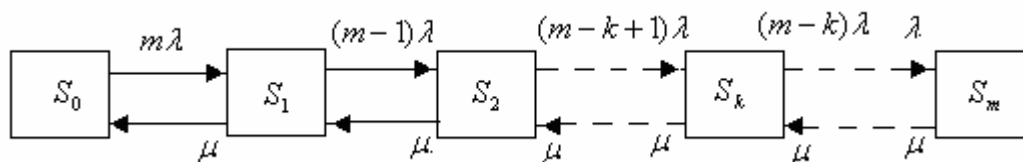


Рисунок 1.6 – Граф состояний одноканальной замкнутой СМО с ожиданием

По графу состояний СМО с использованием формул (1.3) можно определить выражения для финальных вероятностей состояний системы:

$$P_0 = [1 + m\rho + m(m+1)\rho^2 + \dots + m(m+1)\dots(m-k+1)\rho^k + \dots + m!\rho^{m+1}]^{-1};$$

$$P_1 = m\rho P_0;$$

$$P_2 = m(m-1)\rho^2 P_0; \dots$$

$$P_k = m(m-1)\dots(m-k+1)\rho^k P_0, 1 \leq k \leq m; \dots$$

$$P_m = m!\rho^m P_0.$$

Чтобы определить абсолютную пропускную способность  $A$  сначала найдем вероятность того, что наладчик будет занят

$$P_{зан} = 1 - P_0.$$

Если наладчик занят, то он налаживает  $\mu$  объектов в единицу времени, значит, абсолютная пропускная способность будет равна

$$A = (1 - P_0)\mu.$$



Среднее число неисправных объектов  $\bar{w}$  можно выразить через  $A$  с помощью следующего рассуждения: *каждый работающий объект порождает поток отказов с интенсивностью  $\lambda$ , в среднем работает  $m - \bar{w}$  объектов, порождаемый ими поток отказов имеет интенсивность  $(m - \bar{w})\lambda$ , все эти неисправности ликвидируются наладчиком.* Тогда  $(1 - P_0)\mu = (m - \bar{w})\lambda$ , откуда

$$\bar{w} = m - (1 - P_0) / \rho .$$

Среднее число объектов в очереди можно найти как сумму

$$\bar{w} = \bar{r} + \bar{k} ,$$

где  $\bar{k}$  - среднее число занятых каналов обслуживания.

В данном случае число занятых каналов равно нулю, если наладчик свободен, и единице, если наладчик занят, т. е.  $\bar{k} = 0 * P_0 + 1(1 - P_0) = 1 - P_0$ . Следовательно

$$\bar{r} = m - (1 - P_0) / \rho - (1 - P_0), \text{ или } \bar{r} = m - (1 - P_0) * (1 + (1 / \rho)) .$$

**Простейшая многоканальная замкнутая СМО с ожиданием.** Бригада из  $n$  рабочих обслуживает  $m$  станков ( $n < m$ ). Поток отказов одного станка равен  $\lambda$ . Среднее время наладки  $\bar{t}_{\text{обсл}} = 1 / \mu$ . Все потоки простейшие. Требуется найти финальные вероятности состояний системы и характеристики эффективности: абсолютную пропускную способность системы  $A$  и среднее число неисправных станков  $\bar{w}$  (рисунок 1.7).

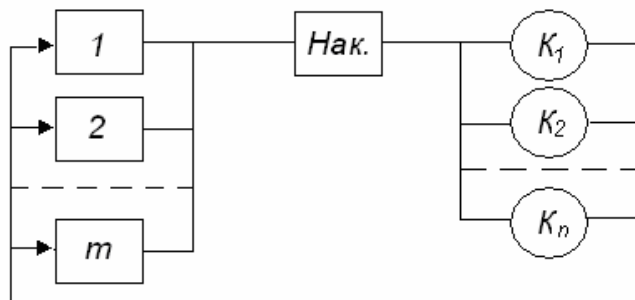


Рисунок 1.7 – Концептуальная модель многоканальной замкнутой СМО с ожиданием

Состояния СМО также нумеруются по числу заявок (неисправных объектов) в СМО:

$s_0$  — все станки исправны, рабочие свободны;

$s_1$  — 1 станок неисправен, 1 рабочий занят, остальные свободны;

$s_2$  — 2 станка неисправны, 2 рабочих заняты, остальные свободны;

$s_k$  —  $k$  станков неисправны,  $k$  рабочих заняты, остальные свободны;...

$s_n$  —  $n$  станков неисправны, все рабочие заняты;

$s_{n+1}$  —  $n+1$  станок неисправен, из них  $n$  — наладывается, 1 станок в очереди;...

$s_{n+r}$  —  $n+r$  станков неисправны, из них  $n$  — наладывается,  $r$  станков в очереди;...

$s_m$  — все  $m$  станков неисправны, из них  $n$  станков наладываются,  $m-n$  неисправных станков стоят в очереди.

Граф состояний многоканальной СМО будет иметь вид (рисунке 1.8).

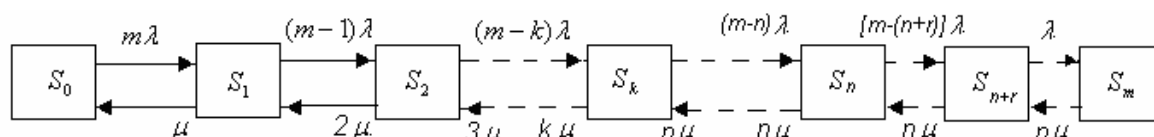


Рисунок 1.8 – Граф состояний многоканальной замкнутой СМО с ожиданием

По этому графу состояний СМО с использованием формул (1.3) определим выражения для финальных вероятностей состояний многоканальной замкнутой СМО с ожиданием [1]:

$$P_0 = \left( 1 + m\rho + \frac{m(m-1)}{2!} \rho^2 + \frac{m(m-1)\dots(m-k+1)}{k!} \rho^k + \dots + \frac{m(m-1)\dots(m-n)}{n!n} \rho^n + \dots + \frac{m(m-1)\dots[m-(n+r-1)]}{n!n^r} \rho^{n+r} + \dots + \frac{m!}{n!n^{m-n}} \rho^m \right)^{-1};$$

$$P_1 = m\rho P_0; \quad P_2 = m(m-1)\rho^2 P_0 / 2!; \quad P_k = m(m-1)\dots(m-k+1)\rho^k P_0 / k!, 1 \leq k \leq m;$$

$$P_n = m(m-1)\dots(m-n+1)\rho^n P_0;$$

$$P_{n+r} = m(m-1)\dots[m-(n+r-1)]\rho^{n+r} P_0 / n!n^{m-n}, \text{ где } 1 \leq r \leq m-1;$$

$$P_m = m! \rho^m P_0 / n!n^{m-n}.$$

Через эти вероятности определяется среднее число занятых рабочих:  
 $\bar{k} = P_1 + 2P_2 + \dots + (n-1)P_{n-1} + n(1 - P_0 - P_1 - \dots - P_{n-1})$ , абсолютная пропускная способность системы  $A = \bar{k}\mu$  и среднее число неисправных станков -  $\bar{w} = m - \bar{k}\mu / \lambda = m - \bar{k} / \rho$ .

## 2 Теоретические сведения о среде моделирования *GPSS World*

### 2.1 Принципы организации системы *GPSS World*

Пакет моделирования дискретных систем (ПМДС) *GPSS* разработан фирмой *IBM* в начале 70-х годов XX в. и является одним из самых распространенных в мире специализированных языков программирования. Система моделирования *GPSS World* является торговой маркой фирмы *MINUTEMAN Software*. Язык *GPSS* был специально разработан для построения имитационных моделей сложных дискретных систем и относится к интерпретирующим языковым системам, применяемым, в основном, для имитации пространственно-временного движения объектов различной природы при фиксированной структуре блочной схемы. Формальными моделями, для имитации которых она используется, являются системы массового обслуживания, конечные и вероятностные автоматы, сети Петри, агрегаты [2].

Система *GPSS World* предназначена для имитационного моделирования сложных дискретных систем. Имитационное моделирование обеспечивает возможность испытания, оценки и проведения экспериментов с предлагаемой системой без каких-либо непосредственных воздействий на нее. При имитационном моделировании проводится эксперимент с программой, которая является моделью системы. Несколько часов, недель или лет работы исследуемой системы могут быть промоделированы на ЭВМ за несколько секунд или минут. В большинстве случаев модель является не точным аналогом системы, а скорее ее символическим изображением. Однако такая модель позволяет проводить измерения, которые невозможно провести каким-либо другим способом.

Первым шагом при анализе любой конкретной системы является разбиение системы на элементы, и формулирование логических правил, управляющих взаимодействием этих элементов. Полученное в результате описание называется моделью системы. Модель обычно включает в себя те аспекты системы, которые представляют интерес или их необходимо исследовать. Исследование

разнообразных дискретных систем показало, что любую систему можно описать с помощью необходимого набора абстрактных элементов, называемых *объектами*. Логические правила поведения таких систем описываются аналогичным в каждом случае набором операций. В системе *GPSS World* языком моделирования является язык *GPSS*.

Язык *GPSS* можно отнести к языкам высокого уровня. В силу этого он имеет довольно слабые алгоритмические возможности. Для устранения этого недостатка в систему *GPSS World* добавлен *PLUS* – язык низкого уровня. Выражения, процедуры и эксперименты *PLUS* можно использовать в *GPSS*-моделях.

Наиболее распространенным методом описания систем является составление блок-диаграмм. *Блок-диаграмма* — графическое представление операций, происходящих внутри системы. Другими словами, блок-диаграмма описывает взаимодействие событий внутри системы. Линии, соединяющие блоки, указывают маршруты потоков сообщений или указывают последовательность выполняемых событий. В случае нескольких вариантов действий от блока отходят несколько линий. Если же к блоку подходят несколько линий, то это означает, что выполняемая операция является общей для двух или более последовательностей блоков. Выбор логических путей может основываться на статистических или логических условиях, действующих в момент выбора.

При составлении блок-диаграммы рекомендуется сначала начертить общий график функционирования системы. Затем составить отдельную блок-диаграмму для соответствия основных функций с блоками *GPSS/World*, после чего внести детали. Чтобы выявить эти дополнительные подробности, некоторые части системы нужно проанализировать более тщательно. Степень детализации зависит от моделируемой системы и от целей моделирования. Начинающие программисты составляют слишком подробную блок-диаграмму. Часто в этом нет необходимости. Блок-диаграммы получили широкое применение при описании систем, но форма представления обычно зависит и от самой системы, и от специалиста, описывающего эту систему. Поэтому при построении блок-диаграмм следует соблюдать определенные условия, являющиеся основой создания программы на

языке моделирования [2].

В *GPSS World* имеется определенное количество типов блоков для задания объектов и операций над ними. Каждому блоку соответствует графическое изображение на блок-диаграмме. Стрелки между блоками указывают маршруты потоков сообщений. Чтобы применить язык моделирования *GPSS World*, каждый блок блок-диаграммы заменяется соответствующим оператором *GPSS World*.

## **2.2 Среда и функциональная структура языка моделирования**

### **2.2.1 Введение в систему *GPSS***

В состав системы *GPSS World* входят следующие типы объектов *транзакты*, *блоки*, *списки*, *устройства памяти*, *логические ключи*, *очереди*, *таблицы*, *ячейки*, *функции*, *переменные* [4].

*Объекты* в моделируемой системе предназначены для различных целей. Выбор объектов в конкретной моделируемой системе зависит от характеристик модели и в некоторых случаях от специалиста, составляющего модель. Совершенно не обязательно, чтобы в одной модели участвовали все типы объектов. Обязательным является лишь то, что в каждой модели должны быть блоки и сообщения, иначе просчитать ее будет невозможно. Какие объекты, помимо сообщений и блоков, будут включены в модель, зависит от того, какие именно используют блоки и операнды блоков (которые могут повлечь появление операторов описания других объектов).

*Транзакты (сообщения)* — это динамические объекты *GPSS World*. Они создаются в определенных точках модели, продвигаются *интерпретатором* через блоки, а затем уничтожаются. Сообщения являются аналогами единиц-потоков в реальной системе. Сообщения могут представлять собой различные элементы даже в одной системе. Например, в модели ЭВМ одни сообщения могут являться прообразами программ пользователя, решаемых на данной ЭВМ, а другие представляют поток отказов в аппаратных средствах компьютера. С каждым

сообщением связаны *параметры*. Параметры могут использоваться для связи конкретных числовых данных с этим сообщением. В приведенном выше примере параметрами сообщений 1-го типа могут быть: время обслуживания программы центральным процессором; число обращений к дискам; идентификация пользователя и т. д. Сообщения движутся от блока к блоку так, как движутся элементы, которые они представляют (программы в примере с ЭВМ). Каждое продвижение считается событием, которое должно происходить в конкретный момент времени. Интерпретатор автоматически определяет моменты наступления событий. В тех случаях, когда событие не может произойти, хотя момент его наступления подошел (например, при попытке занять устройство, когда оно уже занято), сообщение (транзакт) прекращает продвижение до снятия блокирующего условия. Сообщения нумеруются последовательно, начиная с единицы. Параметры сообщений принимают значения из множества целых чисел. Каждое сообщение имеет один или более параметров, параметры нумеруются. Номера параметров используются для ссылок на значения, присвоенные параметрам. Сообщению может присваиваться приоритет, используемый системой в ходе моделирования.

С транзактами связаны следующие понятия:

- отметка времени (MARK TIME) – значение модельного времени, когда транзакт впервые появился в системе;
- семейство транзактов (ASSEMBLY SET) – совокупность транзактов, объединенных в группу;
- индикатор задержки (DELAY INDIKATOR) – указывает номер блока, перед которым находится транзакт, не имея возможности войти в него;
- индикатор трассировки (TRACE INDIKATOR) – указывает номера блоков, которые проходятся активным транзактом;
- текущий блок (CURRENT BLOCK) – номер блока, в котором находится активный транзакт;
- следующий блок (NEXT BLOCK) – номер блока, следующего за текущим блоком на пути движения активного транзакта;
- списки (CHAINS) – упорядоченные массивы информации, отображающие

состояние транзактов в моделируемой системе.

Транзакты (сообщения) в процессе имитации могут быть в следующих состояниях:

— активное (ACTIVE) – транзакт, имеющий наивысший приоритет в списке текущих событий;

— приостановленное (SUSPENDED) – транзакт находится в ожидании активизации в списках будущих или текущих событий;

— пассивное (PASSIVE) – транзакт временно переводится в список пользователя, список задержки или список ожидания;

— завершившееся (TERMINATE) – транзакт уничтожен и больше не выполняется в текущей модели;

— прерванное (PREEMTED) – обработка транзакта некоторым устройством прервана и транзакт переведен в один или более списков прерывания.

После того как система описана, исходя из операций, которые она выполняет, ее нужно описать на языке *GPSS World*, используя блоки, которые выполняют соответствующие операции в модели. Ниже перечислены некоторые свойства этих блоков. В блоках могут происходить события четырех основных типов:

- 1) создание или уничтожение транзактов (сообщений);
- 2) изменение числового атрибута объекта;
- 3) задержка сообщения на определенный период времени;
- 4) изменение маршрута транзакта (сообщения) в модели.

Аналогами обслуживающих приборов в реальных системах *GPSS World* являются объекты типа «оборудование». К объектам этого типа относятся одноканальные устройства, многоканальные устройства и логические ключи.

Одноканальные устройства, в дальнейшем их будем называть *устройствами*, представляют собой оборудование, которое в любой момент времени может быть занято только одним сообщением. Интерпретатор записывает информацию о том, какое сообщение в настоящий момент занимает устройство. Если другое сообщение попытается захватить устройство, то это сообщение задерживается до тех пор, пока устройство не освободится. Программа также автоматически подсчитывает общее



время занятости устройства, что позволяет определить коэффициент использования каждого устройства. Подсчитывается также общее число сообщений, занимавших устройство, что позволяет вычислить среднее время занятости устройства одним сообщением. В реальных системах объекты типа «устройство» могут иногда прерывать обслуживание одних элементов и начинать обслуживание других.

Например, в порт может прибыть судно с грузом высокого приоритета и потребовать немедленного обслуживания у одного из причалов. Прибывшее судно прервет, таким образом, разгрузку любого судна, находящегося у причала (впоследствии разгрузка этого судна возобновится). Такое прерывание можно легко представить средствами *GPSS World*, поскольку устройство *GPSS World* тоже может быть временно прервано другим сообщением. Само устройство в логических и статистических целях продолжает считаться занятым. После окончания прерывания автоматически возобновляется обслуживание прерванного сообщения.

В *GPSS World* многоканальные устройства представляют собой объекты типа «оборудование» для параллельной обработки, они могут быть использованы несколькими транзактами (сообщениями) одновременно. Многоканальные устройства используются для представления физического оборудования, например, зрительного зала театра, стоянки автомобилей и, в некоторых случаях, основной памяти в системах для обработки данных. Пользователь определяет емкость каждого многоканального устройства, используемого в модели, а интерпретатор ведет учет числа единиц многоканальных устройств, занятых в каждый момент времени. Если сообщение пытается занять больше единиц многоканального устройства, чем свободно в данный момент, обработка этого сообщения задерживается до того момента, пока в многоканальном устройстве освободится достаточный объем. Программа автоматически ведет подсчет числа транзактов, входящих в многоканальное устройство, а также определяет среднее число единиц многоканальных устройств, занятых одним сообщением, и среднее время пребывания транзакта (сообщения) в многоканальном устройстве. Эти статистические данные выдаются в конце счета и позволяют определить, насколько эффективно используются в системе объекты параллельной обработки и достаточна

ли их емкость.

Ранее происходившие в моделируемой системе события могут заблокировать и изменить движение последующих событий. Например, кассир кинотеатра, уходя на обед, ставит табличку «К следующему окну», и все последующие клиенты обращаются в другую кассу. Для моделирования таких ситуаций введены *логические ключи*. Транзакт может устанавливать эти ключи в положение «включено» или «выключено». Впоследствии состояние ключа может быть проверено другими транзактами для определения пути их следования. Транзакт может установить ключ (установка ключа соответствует состоянию «включено»), сбросить его (состояние «выключено») или инвертировать, т. е. изменить на противоположное значение состояния ключа. Состояние ключа может быть изменено любым другим транзактом, и любое сообщение может использовать состояние ключа для выбора одного из двух возможных путей или ожидать момента изменения состояния ключа.

*Арифметические переменные* позволяют вычислять арифметические выражения, состоящие из *стандартных числовых атрибутов* (СЧА), которые описаны ниже. Запись выражения переменной аналогична записи соответствующих операторов языка *Фортран*. В выражении переменной используют операторы, арифметические действия и вызовы библиотечных функций.

*Булевские переменные* позволяют пользователю проверять в одном блоке *GPSS World* одновременно несколько условий, исходя из состояния или значения объектов и их атрибутов, т. е. в данном блоке проводится обращение к булевской переменной, выражение которой содержит в себе проверку нескольких условий. Булевская переменная имеет значение 1, если выражение переменной истинно, и 0, если выражение переменной ложно.

Выражениями, используемыми в арифметических и булевских переменных, являются комбинации математических операторов, библиотечных функций, СЧА и констант, составленные по правилам элементарной алгебры. Выражения вычисляются слева направо с учетом иерархии операций. Оценка арифметических выражений производится в формате с плавающей точкой с двойной точностью (64 бит). Результат оценки усекается и хранится как целое число.

Используя *функции*, пользователь может проводить вычисления непрерывных или дискретных функциональных зависимостей между аргументом функции (независимая величина) и зависимым значением функции. Функции широко применяются, например, для задания случайного интервала времени между генерацией двух сообщений. Все функции в *GPSS World* задаются табличным способом с помощью специальных операторов описания Функции.

Ячейки сохраняемых величин и матрицы ячеек сохраняемых величин также относятся к объектам *GPSS World*. Эти ячейки используются для хранения некоторой числовой информации. Например, значение, занесенное в ячейку, может представлять собой длину очереди в какой-то конкретной точке модели. Любое сообщение может произвести запись информации в эти объекты. Впоследствии записанную в эти объекты информацию может считывать любое сообщение. Например, в случае моделирования работы порта для построения зависимости ожидания судов в очереди определенной длины от времени достаточно через каждые  $X$  единиц времени заносить значение длины очереди в ячейки (каждое значение в отдельную ячейку). В конце счета значения этих ячеек будут распечатаны, и по ним можно построить искомую зависимость.

В любой системе движение потока транзактов (сообщений) может быть задержано из-за недоступности оборудования. Например, требуемые устройства могут быть уже заняты или многоканальные устройства, в которые нужно войти, уже заполнены. В этом случае задержанные сообщения ставятся в *очередь* — еще один тип объектов *GPSS World*. Учет этих очередей составляет одну из основных функций интерпретатора *GPSS World*.

Пользователь может определить специальные точки в модели, в которых нужно собирать статистику об очередях. Тогда интерпретатор автоматически будет собирать статистику об очередях (длину очереди, среднее время пребывания в очереди и т. д.). Число задержанных сообщений и продолжительность этих задержек определяется только в этих заданных точках. Интерпретатор также автоматически подсчитывает в этих точках общее число сообщений, поступающих в очередь. Это делается примерно так же, как для устройства и памятей. В определенных счетчиках

подсчитывается число сообщений, задерживающихся в каждой очереди, так как может представлять интерес число сообщений, прошедших какую-либо точку модели без задержки. Интерпретатор подсчитывает среднее время пребывания транзакта в очереди (для каждой очереди) и максимальное число транзактов (сообщений) в очереди.

Как сказано выше, интерпретатор автоматически накапливает определенную статистику относительно устройств, памятей и очередей. Пользователь может также собрать дополнительную статистическую информацию. Например, необходимо собрать статистику о времени пребывания зрителя в кассе кинотеатра (время в очереди, время на приобретение билета). Поскольку в данной модели участвуют разные переменные (время, затрачиваемое на приобретение билета, длина очереди к окошку и т. д.), общее время пребывания сообщений в модели не может быть одинаковым. Сначала определяется диапазон значений, в которых может колебаться время пребывания сообщений в системе. Для удобства обработки информации этот диапазон следует как-то упорядочить (например, по числовым значениям). Самый простой способ достижения такого упорядочивания — устанавливать диапазон значений, скажем, от 5 до 6 мин и затем табулировать те значения, которые попадают в этот диапазон. Если задается ряд последовательных диапазонов (например, 5—6, 6—7, 7—8 и т. д.), можно учесть все значения, которые попадают в каждый из заданных диапазонов. В результате табулируются частоты попадания значений времени пребывания сообщений в каждый из диапазонов (например, в диапазон 6...7 мин попало 568 значений). Для упрощения табулирования в *GPSS World* предусмотрен специальный объект — *таблица*.

Таблица состоит из частотных классов, куда заносится число попаданий конкретного числового атрибута в каждый конкретный частотный класс. Для каждой таблицы вычисляется математическое ожидание и среднеквадратичное отклонение. Эта статистика является стандартной для всех таблиц. При задании таблиц могут понадобиться различные режимы работы таблиц. Режим работы зависит от того, какая статистика нужна.

Описанные выше объекты в модели системы можно считать

присутствующими неявно, т.е. их существование предполагается. Пользователю нет необходимости определять каждый объект и описывать, как его использовать. Он только описывает те действия, которые производятся над этими объектами. Анализируя выходную статистику, пользователь может установить, какое влияние эти действия оказали на объект. Нет особой трудности в описании последовательности действий, выполняемых над одним или несколькими типами отдельных сообщений. Трудность заключается в определении взаимосвязи нескольких типов сообщений. И в этом плане применение *GPSS World* особенно удобно. Описав условия, которые воздействуют на отдельные сообщения, и задав интерпретатору такой режим, по которому он будет продвигать сообщения заданным образом, можно по выходной статистике установить взаимосвязь различных типов сообщений.

Для того чтобы обеспечить правильную временную последовательность событий в модели системы, организованы часы, хранящие значения текущего момента в модели. Все отрезки времени моделируемой системы измеряются целыми числами. В отличие от обычных часов, измеряющих время в определенных единицах, обычно в секундах, часы в *GPSS World* меняют значение только для того, чтобы указать время наступления ближайшего события. Например, если текущее значение часов модели равно 2, а очередное событие должно наступить в момент времени 7, то значение часов увеличивается сразу на 5 единиц. Отметим, что единицы времени в модели не обязательно должны быть конкретными единицами времени, такими, как секунда или час. Основной единицей времени в модели можно выбрать любую единицу, которая позволит получить необходимую точность моделирования. Важно помнить, единицы времени выбирают исходя из требований пользователя к точности моделирования [5].

Поскольку целью построения любой модели является исследование моделируемой системы, интерпретатор *GPSS World* автоматически собирает стандартную статистику по каждому типу объектов, занятых в модели. В зависимости от того, какие объекты и как используются в модели, пользователь имеет возможность получать дополнительную статистику, как в процессе счета

модели, так и по окончании счета. Эта дополнительная статистика помогает оценить результаты счета модели. Стандартная статистика используется и при отладке модели, и при оценке самой системы.

### 2.2.2 Описание объектов *GPSS World*

**Транзакты.** Любую модель рассматриваемой системы в терминах языка *GPSS* можно представить в виде комбинации компонентов, которые составляют интерпретатор *GPSS World*. *Транзакт* — это некоторое сообщение (заявка, требование на обслуживание), которое поступает извне на вход системы и подлежит обработке [2, 9, 19]. Транзакт — обязательный элемент каждой модели на *GPSS*. Транзакты генерируются и уничтожаются, производя при этом воздействия на объекты, образующие модель системы. В терминах систем массового обслуживания (*Q-схем*) транзактам можно сопоставить требования, сообщения или заявки, поступающие в систему на обслуживание [5].

В то же время транзакт это самостоятельный объект системы *GPSS*. Отличие одних транзактов от других состоит в назначаемом им наборе параметров. Структурно транзакт в моделируемой системе *S* не определяется, его существование предполагается после генерации и прохождения по заданным компонентам системы. Работа системы происходит под действием транзактов, которые определяют собой пространственно-временное продвижение реальных объектов в реальной системе. Само моделирование заключается в продвижении сообщений (транзактов) от блока к блоку аналогично функционированию реальной системы. Генерирование транзактов осуществляется блоком *GENERATE*.

**Блоки.** Структура программы на *GPSS* базируется на блок-диаграмме, согласно которой осуществляется продвижение транзактов. С реальными объектами в моделирующей системе *GPSS* связана совокупность блоков, имеющих специальное обозначение и формат описания (записи).



Рисунок 2.1 - Структура GPSS-модели

Продвижение транзактов в модели осуществляет блок *GENERATE*, задержку транзактов во времени может произвести блок *ADVANCE*, вывод транзактов из модели осуществляет блок *TERMINATE*. Каждый блок системы *GPSS* имеет входы и выходы, с помощью которых осуществляется их связь в модели через транзакты.

В системе *GPSS* блоки *GENERATE* и *TERMINATE* имеют особый статус: блок *GENERATE* имеет только выход, блок *TERMINATE* имеет только вход. Блок *GENERATE* генерирует транзакты, блок *TERMINATE* завершает процесс продвижения транзактов.

Структура *GPSS*-модели имеет вид, показанный на рисунке 2.1.

В системе *GPSS World* имеется 53 блока, каждый из которых имеет по несколько параметров с определенными функциональными назначениями.

**Списки.** В *GPSS World* транзакты или сообщения хранятся в списках. *Списки* представляют собой структуры данных, в которых размещена информация о транзактах. Списки относятся к элементам внутренней организации *GPSS*, они обеспечивают внутреннюю логику работы моделирующей системы. Сообщение в любой момент времени находится в одном из пяти списков: текущих событий, будущих событий, прерываний, пользователя, синхронизации.

В *списке текущих событий* транзакты расположены в порядке убывания приоритета. Внутри класса транзакты с одинаковыми приоритетами располагаются в порядке поступления их в список. В списке текущих событий транзакт находится либо в активном состоянии (в способности к продвижению к следующему блоку),

либо в состоянии задержки (когда движение транзакта блокируется какими-либо объектами моделирующей системы).

*Список будущих событий* содержит транзакты, которые могут начать движение в модели в будущие моменты времени. Такие транзакты размещаются в списке строго в порядке возрастания времени начала движения по отношению к значению текущего времени. Приоритеты не влияют на порядок транзактов в рассматриваемом списке.

*Список прерываний* содержит транзакты, обслуживание которых (дальнейшее продвижение по модели) прервано на одном или более устройствах. В списке прерываний транзакты располагаются в порядке поступления. Транзакты находятся в списке прерываний до тех пор, пока не будут сняты все условия прерываний.

*Список пользователя* представляет собой список транзактов, удаленных из списка текущих событий с помощью блока *LINK* и помещенных в список пользователя как временно неактивных.

*Список синхронизации* или список транзактов, которые в данный момент находятся в состоянии сравнения и при этом:

— ожидают в блоках *ASSEMBLE* или *GATHER* прихода заданного числа транзактов из той же группы или из того же семейства;

— находятся в блоках *MATH* в ожидании поступления транзакта из того же семейства на сопряженный блок *MATCH*.

Списки призваны оптимизировать работу моделирующей системы, представляющей собой программу-интерпретатор. В задачу интерпретатора входит, прежде всего, определение ближайшего следующего события, которое должно произойти. После того как определено следующее событие (события), интерпретатор должен выполнить ряд проверок, чтобы определить, существуют ли в системе условия для наступления этого события, если условия существуют, то интерпретатор переходит к обработке события. При этом интерпретатор может собирать некоторую статистическую информацию. Эта статистика впоследствии является частью выходных данных, по которым можно судить о работе моделируемой системы. Основной цикл программы выявления ближайшего



следующего события и его обработка повторяется столько раз, сколько нужно для завершения моделирования. Таким образом, основной задачей, которую выполняет интерпретатор, является просмотр возможных событий. При этом просмотр должен быть организован так, чтобы затрачиваемое на него время было минимальным. Также должен учитываться вопрос об изменении условного времени при организации просмотра возможных событий. Общую внутреннюю последовательность событий можно разбить на три фазы: изменение значения условного времени, просмотр списка текущих событий, движение транзактов. Структура списка *GPSS World* показана на рисунке 2.2.

**Устройства.** Устройства в системе *GPSS World* формируются с помощью блоков *SEIZE*, *RELEASE*, *PREEMPT*, *RETURN*. Наличие в модели устройств позволяет автоматически регистрировать статистическую информацию. Устройства моделируют реальные объекты, в которых может происходить обработка транзактов (сообщений). В *GPSS* предусмотрена возможность прерывания устройств. Имеются средства логической проверки состояния устройств (свободно, занято, прервано, не прервано). Захват (на обработку) и освобождение устройства моделируются соответственно блоками *SEIZE* и *RELEASE*. Прерывание устройства моделирует блок *PREEMPT*, снятие прерывания — блок *RETURN*. Проверка состояния устройства осуществляется блоком *GATE*.

Номер транзакта	Номер текущего блока	Номер следующего блока	Время выхода из списка	Приоритет	Время помещения в список
-----------------	----------------------	------------------------	------------------------	-----------	--------------------------

Рисунок 2.2 - Структура списка *GPSS World*

**Памяти.** Память, или емкость накопителя, служит для моделирования реальных объектов, обладающих конечной величиной числа сообщений (транзактов), ожидающих своей очереди на обработку. Память моделируется блоками *ENTER* и *LEAVE*, а также оператором *STORAGE*, который задает емкость памяти (накопителя). Проверка состояния памяти осуществляется блоком *GATE*.

**Логические ключи.** Логические ключи служат для моделирования объектов с двумя логическими состояниями «включено», «выключено». Управление ключами осуществляется транзактами, продвигающимися по модели. Транзакт может установить ключ в состояние «включено», сбросить его — перевести в состояние «выключено» или инвертировать, т. е. изменить состояние ключа на противоположное значение. Логические ключи моделируются блоками *LOGIC* и *GATE*.

**Очереди.** Если в системе может происходить задержка в продвижении сообщения или задержка в его обработке, то тогда возникает очередь. Для сбора и регистрации статистики об очередях в моделирующей системе ставят блоки *QUEUE* и *DEPART*. При входе сообщения (транзакта) в блок *QUEUE* текущая длина очереди получает приращение. Уход сообщения из очереди отображается блоком *DEPART*.

**Таблицы.** Для сбора различного рода статистических данных и их регистрации используют таблицы, которые моделируются блоком *TABULATE* и оператором *TABLE*. Занесение интересующей информации в таблицу осуществляется блоком *TABULATE* в момент входа очередного транзакта в этот блок. Описание структуры таблицы и типа заносимых данных (стандартизированных самой системой *GPSS*) осуществляется оператором *TABLE*.

**Ячейки.** В системе *GPSS World* для сохранения некоторой числовой информации используются ячейки. Бывают ячейки сохраняемых величин и матрицы ячеек сохраняемых величин. Начальные значения ячеек и матриц можно задавать при помощи оператора описания *INITIAL*. Размер матрицы задается оператором описания *MATRIX*. Изменение содержимого ячеек осуществляется блоками *SAVEVALUE* и *MSAVEVALUE*. Блок *SAVEVALUE* используется для замены, увеличения или уменьшения текущего содержимого заданной ячейки, блок *MSAVEVALUE* — для записи значений в матрицы, а также для увеличения или уменьшения значений записанных в матрицах.

**Функции и переменные.** Функции и переменные в *GPSS World* относятся к вычислительным объектам. В *GPSS World* имеется три типа переменных: арифметические переменные; арифметические переменные с плавающей точкой;

булевыи переменные.

В арифметических переменных, как уже отмечалось, используются следующие арифметические операции:

+ — алгебраическое сложение;

- - алгебраическое вычитание;

# — алгебраическое умножение;

^ — возведение в степень;

/ - алгебраическое деление, остаток отбрасывается; при этом сохраняется только частное (прямой слэш);

\ — деление нацело (обратный слэш);

@ — деление по модулю.

Все функции в *GPSS World* задаются табличным способом с помощью специальных операторов описания функций (непрерывного или дискретного типа).

Арифметические переменные с фиксированной точкой и с плавающей точкой описываются соответственно операторами *VARIABLE* и *FVARIABLE*. Только при описании переменных с плавающей точкой (*FVARIABLE*) допускается применение дробных констант.

Булевыи переменные представляют собой логические выражения. Булевыи переменные описываются оператором *BVARIABLE* с соответствующими операторами отношений или условий. В булевских переменных используются три типа операторов: логические (для определения состояния объектов), булевыи (*OR* — оператор логического сложения и *AND* — оператор логического умножения) и операторы отношения (*G* — больше, *L* — меньше, *E* — равно, *NE* — не равно, *LE* — меньше или равно, *GE* — больше или равно). Булевыи переменные дают возможность принимать решения в зависимости от состояния и значения многих объектов, входящих в моделирующую систему *GPSS/PC*.

В арифметических и булевских переменных могут также использоваться библиотечные функции и выражения следующего вида:

**SIN(...)** – синус,

**COS(...)** — косинус,

**TAN(...)** — тангенс,  
**ATN(...)** — арктангенс,  
**EXP(...)** — экспонента,  
**LOG(...)** — натуральный логарифм,  
**SQR(...)** — квадратный корень,  
**ABC(...)** — абсолютное значение,  
**INT(...)** — целая часть,  
**NOT** — логическое отрицание.

Функции *GPSS World* описываются оператором *FUNCTION*, через который задается тип функции (непрерывный или дискретный) и количество пар точек, задаваемых в следующих строках после строки с оператором *FUNCTION*.

Операторы, применяемые в программах *GPSS World*, обычно располагаются за телом основной программы, т. е. либо до блока *GENERATE*, либо после блока *TERMINATE* (см. рисунок 2.1). При этом для них не обязательна нумерация строк, которая применяется в теле основной программы.

### 2.2.3. Форматы описания блоков и операторов *GPSS*

Блоки (рисунок 2.1) составляют основное тело программы *GPSS*. Блоки должны нумероваться по мере их появления в программе, т. е. номер каждого последующего блока должен быть больше предыдущего. Нумерация блока подразумевает номер строки, в которой осуществляется формат записи блока. Нумерация строк, в которых записываются служебные слова того или иного оператора не обязательна. Формат записи:

**<№>\_<метка>\_БЛОК\_<А>,<В>,<С>,[<D>]; Комментарии**

Описание начинается с номера строки, отсчет которых начинается сверху вниз, после пробела (указан символ \_) помещается по необходимости метка блока, которая должна начинаться с буквы и содержать не более пяти алфавитно-цифровых

символов. После метки или, если она отсутствует, после номера строки следуют два пробела, после которых указывается имя блока (*БЛОК*). Далее после одного пробела располагаются операнды, разделенные запятыми. В литературе в описаниях программ обязательные операнды помещают в символы: больше  $>$ , меньше  $<$  (операнды  $A, B, C$ ), необязательные операнды помещают дополнительно еще в квадратные скобки, как, например, для операнда  $D$ . Количество операндов определяется типом блока и его функциональным назначением в данной программе. Если рабочая строка заканчивается точкой с запятой, то далее в этой строке можно поместить комментарий. Комментарий может помещаться в строке, в первой позиции (колонке) которой будет находиться звездочка (\*). Звездочки часто используют для логического разделения фрагментов программы, удобства чтения и восприятия. Какую-то строку можно, начиная с первой позиции, полностью заполнить звездочками. Формат записи операторов имеет следующий вид:

**[метка]\_ОПЕРАТОР\_<A>,<B>,<C>,<D>,...; Комментарии**

Операторы *GPSS* подразделяются на *операторы* описания данных и контроля управления. В отдельную группу операторов выделяют команды *GPSS*. Описание начинается с метки (алфавитно-цифровой не более 5 знаков), после которой достаточно проставить один пробел и затем имя оператора. После имени оператора через один пробел указываются операнды. Комментарии пишутся после точки с запятой. Количество операндов различно для разных операторов. Для некоторых операторов метка не ставится. В выходной статистике *GPSS* строки с операторами не фиксируются и не указываются. Для записи операторов можно использовать нумерованные строки, но они не будут присутствовать в выходной статистике. Формат записи операторов с нумерованными строками будет приведен позднее при рассмотрении конкретных программ на *GPSS*.

#### 2.2.4. Стандартные числовые атрибуты

Все элементы, используемые в *GPSS*, имеют специальные числовые характеристики, которые называются стандартными числовыми атрибутами (СЧА). Например, для различия одного сообщения от другого можно применить такой числовой атрибут, как приоритет. Каждый транзакт (сообщение) может иметь несколько параметров, по которым они отличаются друг от друга. Транзакты имеют четыре СЧА:

*PR* — приоритет по умолчанию равен нулю;

*Pj* или *\*j*, или *\*<имя>*, или *\*\$<имя>* — значение параметра *j* текущего сообщения или значение параметра с именем *<имя>* текущего транзакта. Этот СЧА задает номер (*j*) или имя параметра транзакта;

*MPj* — значение времени, равное разности абсолютного модельного времени и величины модельного времени, записанного в *j*-том параметре текущего транзакта. Разница может быть как положительной, так и отрицательной;

*MBj* — флаг синхронизации: присваивается 1, если транзакт в блоке *j* принадлежит тому же семейству, что и текущий транзакт; в противном случае присваивается 0. Синхронизация отмечается между сопряженными блоками *MATCH*.

Поэтому транзакты могут идентифицироваться по заданным СЧА. Значения СЧА для транзактов задаются разработчиком *GPSS*-программы.

Другие объекты *GPSS*, как и транзакты, имеют соответствующие наборы СЧА: блоки, устройства, многоканальные устройства, очереди, таблицы, ячейки и матрицы сохраняемых величин, списки и группы, вычислительные объекты. Для изменения значений некоторых атрибутов к ним можно обращаться программным путем согласно логике рассматриваемой модели. По мере изучения системы *GPSS* будут приводиться СЧА соответствующих объектов.

В *GPSS* еще существуют так называемые системные СЧА, к которым пользователь может обращаться в модели, но не может изменять их значение. К системным числовым атрибутам относятся следующие величины:

*CI* — текущее значение условного времени. Автоматически изменяется программой и устанавливается в ноль управляющими операторами *CLEAR* и *RESET*;

*ACI* — текущее значение абсолютного времени. Автоматически изменяется программой. Эта величина не меняется под действием управляющего оператора *RESET* и устанавливается в 0 лишь под действием оператора *CLEAR*;

*MI* — время пребывания в модели сообщения, обрабатываемого программой в данный момент. Эта величина может изменяться блоком *MARK*. Величина *MI* вычисляется как разница текущего значения абсолютного времени и отметки времени обрабатываемого транзакта;

*ZI* — возвращает размер свободной оперативной памяти в байтах;

*XNI* — возвращает номер активного транзакта;

*TGI* — число равное текущему значению счетчика завершений. Транзакты, вошедшие в блоки *TERMINATE* с нулевым операндом  $\langle A \rangle$ , уменьшают значение этого счетчика на число, равное значению операнда  $\langle A \rangle$ ;

*PR* — приоритет транзакта обрабатываемого в данный момент. По умолчанию приоритет равен нулю. Эта величина может изменяться (назначаться) блоком *PRIORITY*;

*RN<sub>j</sub>* — число, вычисляемое *j* генератором (датчиком) случайных чисел ( $j = 1, \dots, 7$ , т. е. *RN1*, *RN2*, ..., *RN7*). Все датчики генерируют последовательность равномерно распределенных чисел от 0 до 999 включительно. Если число используется в качестве аргумента функции или элемента в переменной, то тогда осуществляется генерация случайных равномерно распределенных чисел в пределах от 0 до 0.999999.

Следует отметить, что СЧА (включая системные числовые атрибуты), т. е. их символьные обозначения, нельзя использовать в качестве меток операторов и блоков программы *GPSS*.

С помощью СЧА осуществляется дополнительное специальное кодирование операторов и блоков *GPSS*. Практически всегда СЧА можно использовать в качестве операндов блоков и большинства операторов.

В *GPSS* стандартные числовые атрибуты имеют следующие объекты: блоки,

устройства, многоканальные устройства, очереди, таблицы, ячейки и матрицы ячеек сохраняемых величин, списки и группы, вычислительные объекты (СЧА вычисленного значения функции, СЧА вычисленного значения переменной). С помощью СЧА пользователь получает доступ к характеристикам состояния системы в процессе моделирования.

### **2.3 Методика применения языка GPSS для имитации СМО**

Данные для ЭВМ подготавливаются в виде программы, представляющей собой последовательность операторов блоков и операторов команд или процедур, которая составляется по схеме модели, набранной из стандартных символов. Созданная GPSS-программа, работая в режиме интерпретации, генерирует и передает транзакты из блока в блок в соответствии с правилами, устанавливаемыми блоками. Каждый переход транзакта приписывается к определенному моменту системного времени.

В процессе прогона имитационной модели обеспечивается возможность постоянного наблюдения в шести интерактивных графических окнах (машинная мультипликация, работа одноканальных и многоканальных устройств, функционирование блоков, отображение таблиц и матриц). Система выдает отчет о результатах моделирования, содержащий широкий спектр параметров блоков, устройств, очередей, таблиц, пользовательских списков.

Для пакета GPSS были введены некоторые условности и общие представления о моделируемых системах. Введение таких обобщений позволило создать язык блок-диаграмм, в котором блоки соответствуют некоторым действиям, встречающимся в системах исследуемого класса. Вводимый набор блоков для блок-диаграмм однозначно определяет наборы операторов языка, осуществляющих описание структуры моделируемой системы, и логических правил, определяющих ее функционирование.

В блок-диаграммах GPSS блоки представляют собой выполняемые над динамическими объектами операции, а стрелки между блоками отражают маршруты



передвижения данных объектов по системе. Альтернативные ситуации отражаются более чем одной стрелкой, выходящей из блока.

Таким образом, процесс создания модели на языке блок-диаграмм *GPSS* сводится к:

*декомпозиции исходной системы* до уровня элементарных операций, используемых в пакете *GPSS*,

*формированию фиксированной схемы*, отражающей последовательность элементарных операций, выполняемых над динамическими объектами,

*определению набора логико-вероятностных правил продвижения потоков объектов* по имеющейся схеме.

Построение блок-диаграмм *GPSS* предполагает знакомство программиста с набором операторов пакета *GPSS*. Набор операторов языка однозначно соответствует набору блоков для описания блок-диаграмм, поэтому построение блок-диаграммы не является самоцелью, а лишь промежуточным этапом при построении имитационной модели исследуемой системы с использованием операторов пакета *GPSS*.

### **2.3.1 Имитация функционирования одноканальной СМО**

Процесс моделирования в системе *GPSS* заключается в продвижении сообщений от блока к блоку с указанием команд и действий с помощью операторов. Продвижение сообщений по модели начинается с блока *GENERATE* и заканчивается в блоке *TERMINATE*. Задержка сообщений во времени по заданному закону происходит в блоке *ADVANCE*. Ввод в систему сообщений может быть также задан по какому-либо статистическому закону. Перечисленные три блока относятся к блокам динамической категории. Сами сообщения или транзакты являются динамическими объектами системы *GPSS*. Следует помнить, что транзакты не могут входить в блок *GENERATE* и выходить из блока *TERMINATE*, в противном случае происходит ошибка выполнения программы и ее останов. Моделирующая программа начинается с блока *GENERATE* и заканчивается блоком *TERMINATE* [2].

К аппаратной категории относятся блоки первой группы устройств *SEIZE*, *RELEASE*. Введение в моделирующую программу устройств (с помощью соответствующих блоков) позволяет автоматически регистрировать статистическую информацию. Рассмотрим более подробно наиболее характерные примеры программ с указанными блоками и операторами.

**Пример 1.** Заявки поступают в систему массового обслуживания (*Q*-схему) через фиксированное время (7 единиц времени). Обработка (обслуживание) каждой заявки занимает также некоторое фиксированное время (5 единиц времени). После обработки заявки покидают *Q*-схему. Провести обработку 100 заявок. Единицу времени назначает пользователь.

За единицу времени принята 1 мин, а программе решения примера 1 присвоено имя *SAMPLE1.GPS*. В этой программе транзакты (заявки, сообщения) генерируются блоком *GENERATE* каждые 7 мин, проходят через устройство под номером 1, задерживаются на 5 мин и покидают (освобождают) устройство 1. Захват и освобождение устройства осуществляют блоки *SEIZE* и *RELEASE*. Задержка транзактов во времени происходит в блоке *ADVANCE*. Вывод транзактов производит блок *TERMINATE*. Оператор *START* начинает процесс моделирования для указанного максимального числа счетчика завершений.

Программа (имя файла *SAMPLE01.GPS*) в рабочем окне *GPSS World* для рассматриваемого примера имеет вид, показанный на рисунке 1.

```
GENERATE 7          ; Генерировать транзакт каждые 7 мин .
SEIZE     Canal    ; Занять прибор .
ADVANCE  5          ; Задержать транзакт на 5 мин .
RELEASE  Canal    ; Освободить прибор .
TERMINATE 1        ; Уничтожить транзакт .
```

Рисунок 2.3 – Программа имитации СМО в рабочем окне *GPSS World*

Приведем некоторые общие правила оформления программ.

1 После написания программы необходимо перед запуском на выполнение сохранить ее как файл с расширением *GPS* и именем, состоящим из букв и цифр

длиной до 20 символов. Допускается применять символы только латинского алфавита. Во-вторых, обязательным является нумерация блоков в порядке возрастания, операторы можно не нумеровать.

2 Как правило, после номера устанавливается не менее двух пробелов и записывается имя блока. В рабочем окне можно использовать не более 255 позиций, начиная с левого края.

3 Комментарии пишутся после знака точки с запятой (;) в любом месте программы.

Для запуска модели необходимо:

командой CREATE SIMULATION меню COMMAND создать *процесс моделирования*, появляется окно - журнал Sample1.1.sim – JOURNAL;

вызвать окно команды START, в поле команды записать число прогонов (START100) и клавишей ОК запустить модель.

После выполнения заданного числа прогонов появляется стандартный отчет *Sample1.1.1 – REPORT* с результатами моделирования, который автоматически формируется системой *GPSS/World*.

В приведенном на рисунке 2.4 файле стандартного отчета сначала идет информационная строка системы, строка с указанием версии, даты, времени;

затем строка стандартного сообщения о начале моделирования *START TIME*, его окончании *END TIME*, количестве используемых в программе блоков *BLOCKS*, количестве установленных устройств *FACILITIES*, количестве многоканальных устройств, для которых определяется заданная емкость накопителя *STORAGES*.

Далее приводятся:

номера блоков, которые определяются системой (*LOC*);

названия блоков *BLOCK\_TIPE*;

количество транзактов, проходящих через соответствующий блок программы *ENTRY\_COUNT*;

текущее количество транзактов, задержанных в блоке на момент окончания моделирования *CURRENT\_COUNT*;

количество транзактов, ожидающих специальных условий для прохождения

через данный блок *RETRY*.

GPSS World Simulation Report - Sample1.1.1

	START TIME	END TIME	BLOCKS	FACILITIES	STORAGES				
	0.000	705.000	5	1	0				
	NAME		VALUE						
	CANAL		10000.000						
LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY				
	1	GENERATE	100 ( <b>N</b> )	0	0				
	2	SEIZE	100	0	0				
	3	ADVANCE	100	0	0				
	4	RELEASE	100 ( <b>No</b> )	0	0				
	5	TERMINATE	100	0	0				
FACILITY	ENTRIES	UTIL. ( <b>Кл</b> )	AVE.TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
CANAL	100	0.709	5.000	1	0	0	0	0	0
FEC XN	PRI	BDT	ASSEM	CURRENT	NEXT	PARAMETER	VALUE		
101	0	707.000	101	0	1				

Рисунок 2.4 – Стандартный отчет *Sample1.1.1 – REPORT*

Как видно, для данной программы количество транзактов, проходящих через блоки равно 100. Все 100 транзактов, которые были сгенерированы блоком GENERATE, прошли через все блоки программы.

Работа устройства *FACILITY* под номером 1 оценивается следующими стандартными атрибутами:

количество транзактов, прошедших через устройство, указывается в поле *ENTRIES* (в нашем случае это число 100);

коэффициент использования в относительных единицах указывается в поле *UTIL* (в нашем случае число 0.709, которое показывает ту часть периода моделирования, в течение которого устройство было занято);

среднее время обработки или задержки одного транзакта в устройстве указывается в поле *AVE.\_TIME* (в нашем случае равно 5, которое так же можно определять, как среднее время занятости устройства);

поле *AVAILABLE* определяет состояние готовности устройства в конце

периода моделирования (оно равно  $1$ , если устройство готово, и  $0$ , — если не готово);

поле *OWNER* (прямой перевод — владелец) определяет номер последнего транзакта, занимавшего устройство (если устройство не занималось, то устанавливается  $0$ );

поле *PEND* (от англ. «pendent» — ожидающий решения) определяет количество транзактов, ожидающих освобождения устройства, находящегося в режиме прерывания;

поле *INTER* определяет количество транзактов, прерывающих работу устройства в данный момент;

поле *RETRY* определяет количество транзактов, ожидающих специальных условий, зависящих от состояния объекта типа «устройство»;

поле *DELAY* (задержка) определяет количество транзактов, ожидающих занятия или освобождения устройства.

В рассматриваемом примере последние пять полей равны нулю, так как система работает без каких-либо специальных условий и прерываний.

Содержание информации, которая выводится для транзактов из СБС следующее:

поле *XN* — номер транзакта;

поле *PRI* — приоритет транзакта;

поле *BDT* — таблица модельных событий — абсолютное модельное время выхода транзакта из СБС (и перехода транзакта в СТС);

поле *ASSEM* — номер семейства транзактов;

поле *CURRENT* — номер блока, где находится транзакт ( $0$  — если транзакт еще не вошел ни в один блок модели);

поле *NEXT* — номер блока, в который должен войти транзакт;

поле *PARAMETER* — номер или имя параметра транзакта;

поле *VALUE* — значение параметра.

### 3 Лабораторный практикум по дисциплине «Моделирование»

#### 3.1 Лабораторная работа № 1

##### Оценка показателей эффективности простейшей СМО в среде «GPSS/World»

**Цель работы:** освоение приемов по разработке программных моделей СМО и определению основных показателей эффективности одноканальной СМО с потерями.

**Задание на лабораторную работу** В СМО поступают заявки, распределенные по экспоненциальному закону с интенсивностью  $\lambda$ . Обработка заявок осуществляется также по экспоненциальному закону с интенсивностью обслуживания  $\mu$ . Если поступившая заявка застает канал занятым, то она теряется. Необходимо смоделировать работу СМО в течение 8 часов.

Определить основные параметры одноканальной СМО с потерями:  $Q; P_{отк}; K_u; \bar{T}_o$ , при  $\mu = 2\lambda$ . Выявить вид зависимости  $Q = f(\mu)$ , при  $\lambda = const$ .

Таблица 1 – Варианты заданий

№ вар	1	2	3	4	5	6	7	8	9	10	11
$\lambda$ 1/мин	0,2	0,4	0,5	0,6	0,7	0,8	0,9	2	2,5	3	5

#### Порядок выполнения работы

##### 1 Исследование одноканальной СМО с потерями в среде GPSS

1.1 Аналитически определить основные параметры одноканальной СМО с потерями:  $Q; P_{отк}; K_u; \bar{T}_o$ .

1.2 Составить программу имитационной модели в среде GPSS/W, пользуясь методикой, изложенной в учебном пособии «Практикум по программированию процессов функционирования типовых СМО в среде GPSS/W».

Для этого:

1.2.1 Составьте объект «Модель» в виде последовательности операторов модели. При этом необходимо стремиться к тому, чтобы поведение модели как можно точнее отражало поведение системы. Для устранения очереди заявок перед каналом между операторами GENERATE и SEIZE необходимо включить оператор «GATE NU Canal,Fakil», отправляющий заявки в случае занятости канала в блок с меткой «Fakil».

```
GENERATE 480 ;Генерировать транзакт внешнего цикла.
TERMINATE 1 ;Уничтожить транзакт внешнего цикла.
GENERATE (Exponential(1,0,1/0.2)) ; Генерация транзактов (заявок) по
;экспоненциальному ЗР интервалов между заявками с  $\lambda=0.2$  1/мин.
GATE NU Canal,Fakil ; Проверка занятости канала.
SEIZE Canal ; Занятие канала обслуживания.
ADVANCE (Exponential(2,0,1/0.4)) ; Обслуживание транзактов (заявок).
RELEASE Canal ; Освобождение канала обслуживания.
TERMINATE 0 ; Уничтожение транзактов.
Fakil LINK Poteri,FIFO ;Потерянные заявки.
```

Рисунок 3 – Программа модели простейшей одноканальной СМО

1.2.2 Создайте объект «Процесс моделирования» и по результатам трансляции в окне «Журнал» оцените качество трансляции. При наличии ошибок в наборе операторов и их полей необходимо их устранить в объекте «Модель». После чего снова создать объект «Процесс моделирования».

1.3 Запустив «Процесс моделирования» командой «Старт», проведите анализ появившегося отчета, из которого определите результаты имитации (число поступивших  $N$ , число обслуженных  $N_0$ , число потерянных  $N_1$  заявок и коэффициент использования) и занесите их в таблицу 2.

1.4 Используя команды CLEAR и START, еще 4 раза запустите «Процесс моделирования», и проведите аналогичный п. 1.3 анализ появившихся отчетов.

Таблица 2 – Статистические данные из отчетов

	$N$	$N_o$	$Q$	$P_{отк}$	$K_u$
1					
2					
3					
4					
5					
$M(X)$					

1.5 По данным таблицы 2 определите вероятность отказа заявке в обслуживании и вероятность обслуживания поступившей заявки по каждому прогону модели.

1.6 Рассчитайте средне-арифметические оценки показателей эффективности системы и сравните их с теоретическими.

1.7 Снимите зависимость  $Q$  от интенсивности обслуживаний  $\mu$  при неизменной интенсивности потока заявок  $\lambda$ .

1.8 Постройте график зависимости. Сделайте выводы.

### Содержание отчета

1 Тема, цель задания.

2 Q- схема одноканальной СМО.

3 Результаты теоретических расчетов:  $Q; P_{отк}; K_u; \bar{T}_o$ ;

4 Результаты исследований по п.п. 1.2...1.8 (Программа модели, стандартный отчет, таблицы, графики, выводы).



### 3.2 Задание на лабораторную работу № 2

#### Оценка показателей эффективности трехканальной СМО с потерями в среде «GPSS/World»

**Цель работы:** экспериментальная проверка теоретических показателей эффективности простейшей трехканальной СМО с потерями.

**Задание.** В трехканальную систему массового обслуживания с потерями поступают заявки, распределенные по экспоненциальному закону с интенсивностью  $\lambda$ . Обработка поступивших заявок осуществляется также по экспоненциальному закону с интенсивностью обслуживания  $\mu$ . Если поступившая заявка застает все каналы занятыми, то она теряется. Необходимо смоделировать работу системы обслуживания в течение 8 часов.

Определить основные параметры трехканальной СМО с потерями:  $Q; P_{отк}; A; \bar{K}$ , при  $\mu = 0.6\lambda$ . Выявить вид зависимости  $Q = f(n)$ , где  $n$  – число каналов в системе при  $\lambda = const$ .

Таблица 1 – Варианты заданий

№ вар	1	2	3	4	5	6	7	8	9	10	11
$\lambda$ 1/ мин	0,2	0,4	0,5	0,6	0,7	0,8	0,9	2	2,5	3	5

#### Порядок выполнения работы

##### 1 Исследование трехканальной СМО с потерями в среде GPSS

1.1 Аналитически определить основные характеристики эффективности трехканальной СМО с потерями  $Q; P_{отк}; A; \bar{K}$ .

1.2 Составить программу имитационной модели в среде GPSS/W, пользуясь методикой, изложенной в учебном пособии «Практикум по программированию процессов функционирования типовых СМО в среде GPSS/W». Для этого:

1.2.1 Составьте объект «Модель» в виде последовательности операторов модели. При этом необходимо стремиться к тому, чтобы поведение модели как можно точнее отражало поведение системы. Для исключения очереди заявок после блока GENERATE необходимо включить блок «*GATE NU Canal,Fakil*», отправляющий заявки в случае занятости всех каналов в блок с меткой «*Fakil*».

1.2.2 Создайте объект «Процесс моделирования» и по результатам трансляции в окне «Журнал» оцените качество трансляции. При наличии ошибок в наборе операторов и их полей необходимо их устранить в объекте «Модель». После чего снова создать объект «Процесс моделирования».

1.3 Запустив «Процесс моделирования» командой «Старт», проведите анализ появившегося отчета, из которого определите результаты имитации (число поступивших  $N$ , число обслуженных  $N_o$ , число потерянных  $N_l$  заявок и среднее число занятых каналов) и занесите их в таблицу 2.

1.4 Используя команды CLEAR и START, еще 4 раза запустите «Процесс моделирования», и проведите аналогичный п. 1.3 анализ появившихся отчетов.

Таблица 2 – Статистические данные из отчетов

	$N$	$N_o$	$Q$	$Pomk$	$K$
1					
2					
3					
4					
5					
$M(X)$					

1.5 По данным таблицы 2 определите вероятность отказа заявке в обслуживании и вероятность обслуживания поступившей заявки по каждому прогону модели.

1.6 Рассчитайте средне-арифметические оценки показателей эффективности системы и сравните их с теоретическими.

1.7 Снимите зависимость  $Q$  от числа каналов в системе  $n$  при неизменной интенсивности потока заявок  $\lambda$ .

1.8 Постройте графики теоретической и экспериментальной зависимостей вероятности обслуживания системой как функции числа каналов. Сделайте выводы.

### **Содержание отчета**

1 Тема, цель задания.

2 Концептуальная модель СМО.

3 Аналитический расчет характеристик эффективности СМО.

4 Результаты экспериментальных исследований: программа, стандартный отчет, таблицы, графики, выводы.

### **3.3 Задание на лабораторную работу № 3**

**Оценка показателей эффективности простейшей разомкнутой СМО с ожиданием в среде «GPSS/World»**

**Цель работы:** экспериментальная проверка теоретических показателей эффективности простейшей разомкнутой СМО с ожиданием.

**Задание.** *В одноканальную систему массового обслуживания с ограниченной очередью поступают заявки, распределенные по экспоненциальному закону с интенсивностью  $\lambda$ . Обработка поступивших заявок осуществляется также по экспоненциальному закону с интенсивностью обслуживания  $\mu$ . Емкость накопителя  $L=3$ . Если поступившая заявка застаёт канал занятыми, то она*

становится в очередь и теряется только в случае отсутствия свободных мест в очереди. Необходимо смоделировать работу системы обслуживания в течение 8 часов.

Определить основные параметры одноканальной СМО с ограниченной очередью:  $Q; \bar{r}; \bar{t}_{ож}; K_u$ , при  $\mu = 0.6\lambda$ . Выявить вид зависимости  $Q = f(n)$ , где  $n$  – число каналов в системе при  $\lambda = const$ .

Таблица 1 – варианты заданий

№ вар	1	2	3	4	5	6	7	8	9	10	11
$\lambda$ 1/ мин	0,2	0,4	0,5	0,6	0,7	0,8	0,9	2	2,5	3	5

## Порядок выполнения работы

### 1 Исследование одноканальной СМО с ограниченной очередью в среде GPSS

1.1 Аналитически определить основные параметры одноканальной СМО с ограниченной очередью  $Q; \bar{r}; \bar{t}_{ож}; K_u$ .

1.2 Составить программу имитационной модели в среде GPSS/W, пользуясь методикой, изложенной в учебном пособии «Практикум по программированию процессов функционирования типовых СМО в среде GPSS/W». Для этого:

1.2.1 Составьте объект «Модель» в виде последовательности операторов модели. При этом необходимо стремиться к тому, чтобы поведение модели как можно точнее отражало поведение системы. Для исключения очереди заявок после блока *GENERATE* необходимо включить блок «*GATE SNF Nak,Fakil*», отправляющий заявки в случае занятости всех каналов в блок с меткой «*Fakil*».

1.2.2 Создайте объект «Процесс моделирования» и по результатам трансляции в окне «Журнал» оцените качество трансляции. При наличии ошибок в наборе операторов и их полей необходимо их устранить в объекте «Модель». После чего снова создать объект «Процесс моделирования».

1.3 Запустив «Процесс моделирования» командой «Старт», проведите анализ появившегося отчета, из которого определите результаты имитации (число поступивших  $N$ , число обслуженных  $N_o$ , число потерянных  $N_l$  заявок и среднее число заявок в очереди, коэффициенты использования канала и накопителя) и занесите их в таблицу 2.

1.4 Используя команды *CLEAR* и *START*, еще 4 раза запустите «Процесс моделирования», и проведите аналогичный п. 1.3 анализ появившихся отчетов.

Таблица 2 – Статистические данные из отчетов

	$N$	$N_o$	$Q$	$r$	$Ku$	$тожс=r/\lambda$
1						
2						
3						
4						
5						
$M(X)$						

1.5 По данным таблицы 2 определите вероятность отказа заявке в обслуживании и вероятность обслуживания поступившей заявки по каждому прогону модели.

1.6 Рассчитайте средне-арифметические оценки показателей эффективности системы и сравните их с теоретическими.

1.7 Снимите зависимость  $Q$  от емкости накопителя  $L$  при неизменных интенсивности потока заявок  $\lambda$  и интенсивности обслуживания  $\mu$ .

1.8 Постройте графики теоретической и экспериментальной зависимостей вероятности обслуживания системой как функции емкости накопителя. Сделайте выводы.

### Содержание отчета

1 Тема, цель задания.

2 Q- схема одноканальной СМО с очередью.

3 Результаты теоретических расчетов:  $Q; \bar{r}; \bar{t}_{ож}; K_u$ ;

4 Результаты исследований по п.п. 1.2...1.8 (Программа модели, стандартный отчет, таблицы, графики, выводы).

### 3.4 Задание на лабораторную работу № 4

#### Оценка показателей эффективности замкнутой СМО с ожиданием

##### Цель работы:

закрепление знаний основных положений теории массового обслуживания и выработка навыков в исследовании систем;

экспериментальная проверка теоретических показателей эффективности простейшей разомкнутой СМО с ожиданием.

**Задание.** *Один наладчик обслуживает  $m$  компьютеров, у которых периодически возникает сбой (отказ) в работе, интервалы между отказами, распределены по экспоненциальному закону с интенсивностью  $\lambda$ . Если в момент отказа наладчик свободен, он сразу приступает к наладке ПК, если – нет, то ПК становится в очередь на восстановление. Длительность восстановления ПК распределена также по экспоненциальному закону с интенсивностью обслуживания  $\mu$ . Необходимо смоделировать работу системы обслуживания в течение 8 часов, при  $\mu = 0.3\lambda$  и  $m=5$ .*

Определить основные параметры одноканальной замкнутой СМО с ожиданием:  $A; P_{зан}; \bar{w}; \bar{r}$ , где  $A$  – интенсивность восстановления ПК;  $P_{зан}$  - вероятность занятости наладчика;  $\bar{w}$  - среднее число неисправных ПК;  $\bar{r}$  - среднее число ПК, ожидающих восстановления.

Выявить вид зависимости  $P_{зан} = f(n)$ , если  $\lambda = const$  и  $\mu = const$ , а  $n$  – число наладчиков.

Таблица 1 – Варианты заданий

№ вар	1	2	3	4	5	6	7	8	9	10	11
$\lambda$ 1/мин	0,2	0,4	0,5	0,6	0,7	0,8	0,9	2	2,5	3	5

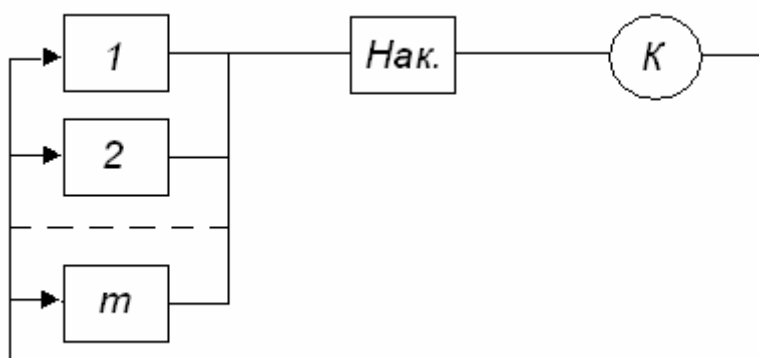


Рисунок 1 – Концептуальная модель одноканальной замкнутой СМО с ожиданием

## Порядок выполнения работы

### 1 Исследование одноканальной замкнутой СМО с ожиданием в среде GPSS

1.1 Аналитически определить основные параметры одноканальной замкнутой СМО с ожиданием:  $A; P_{зан}; \bar{w}; \bar{r}$ .

1.2 Составить программу имитационной модели в среде GPSS/W, пользуясь методикой, изложенной в учебном пособии «Практикум по программированию процессов функционирования типовых СМО в среде GPSS/W».

Для этого:

1.2.1 Составьте объект «Модель» в виде последовательности операторов модели (программу ИМ системы). При этом необходимо стремиться к тому, чтобы поведение модели как можно точнее отражало поведение системы.

1.2.2 Создайте объект «Процесс моделирования» и по результатам трансляции в окне «Журнал» оцените качество трансляции. При наличии ошибок в наборе операторов и их полей необходимо их устранить в объекте «Модель». После чего снова создать объект «Процесс моделирования».

1.3 Запустив «Процесс моделирования» командой «Старт», проведите анализ появившегося отчета, из которого определите результаты имитации (число поступивших в ремонт ПК  $N$ , число восстановленных ПК  $N_0$ ,  $P_{зан}$  - вероятность занятости наладчика,  $\bar{w}$  - среднее число неисправных ПК,  $\bar{r}$  - среднее число ПК, ожидающих восстановления) и занесите их в таблицу 2.

1.4 Используя команды CLEAR и START, еще 4 раза запустите «Процесс моделирования», и проведите аналогичный п. 1.3 анализ появившихся отчетов.

Таблица 2 – Статистические данные из стандартных отчетов

	$N$	$N_0$	$A$	$P_{зан}$	$\bar{w}$	$\bar{r}$
1						
2						
3						
4						
5						
$M(X)$						

1.5 По данным таблицы 2 определите интенсивность восстановления ПК по каждому прогону модели.

1.6 Рассчитайте средне-арифметические оценки показателей эффективности системы и сравните их с теоретическими.

1.7 Снимите зависимость вероятности занятости наладчиков от их количества  $P_{зан} = f(n)$ , если  $\lambda = const$  и  $\mu = const$ , а  $n$  – число наладчиков.

1.8 Постройте графики аналитической и экспериментальной зависимостей. Сделайте выводы.



## **Содержание отчета**

- 1 Тема, цель задания.
- 2 Схема концептуальной модели исследуемой системы.
- 3 Программная модель системы.
- 4 Результаты исследований (аналитические расчеты показателей эффективности системы, один из стандартных отчетов, таблицы, графики, выводы).

## **4 Лабораторный практикум по дисциплине «Теория эксперимента»**

### **4.1 Лабораторная работа № 5**

#### **Исследование элементов системы моделирования GPSS на имитационных моделях процессов массового обслуживания**

##### **Цель работы:**

изучение особенностей использования языка моделирования GPSS/W для проведения статистических экспериментов с моделями систем;

освоение навыков практического применения объектов статистической категории при решении конкретных задач моделирования систем, формализуемых в виде схем массового обслуживания (Q-схем).

Краткие сведения по применению языка моделирования GPSS/W.

Язык моделирования дискретных систем GPSS/W построен в предположении, что моделью сложной дискретной системы является описание ее элементов и логических правил их взаимодействия в процессе функционирования моделируемой системы. Предполагается также, что для определенного класса моделируемых систем можно выделить небольшой набор абстрактных элементов — объектов. Причем набор логических правил также ограничен и может быть описан небольшим числом стандартных операций. Комплекс программ, описывающих функционирование объектов и выполняющих логические операции, является основой для создания программной модели системы данного класса.

Объекты GPSS/W подразделяются на 7 категорий и 14 типов. Объекты GPSS/W позволяют описать их взаимодействие сравнительно несложными наборами операций и достаточно просто и наглядно представить процесс функционирования исследуемой системы, формализуемой в виде Q-схемы, с помощью этих объектов языка. Кроме того, для облегчения пользователю процесса построения модели в GPSS/W разработан язык блок-диаграмм, позволяющий

упростить переход от алгоритма к программе модели системы. Каждый блок GPSS/W имеет свой графический аналог. С помощью набора графических аналогов отображается пространственная конструкция модели, упрощая дальнейшую линеаризацию программы модели.

Построение блок-диаграмм знакомит программиста с набором операторов языка. Набор операторов языка однозначно соответствует набору блоков для описания блок-диаграммы. Вследствие этого очевидно, что построение блок-диаграммы является не самоцелью, а лишь промежуточным этапом при написании имитационной модели исследуемой системы с использованием операторов языка GPSS/W. При этом процесс создания модели включает [2]:

- исходное описание моделируемой системы;
- формализацию системы в терминах Q-схем;
- декомпозицию системы к виду блок-диаграммы;
- перевод блок-диаграммы в программу на GPSS/W.

Основой модели являются программы, описывающие функционирование выделенного конечного набора объектов, и программа управления моделированием — симулятор, основные функции которого следующие:

- обеспечение заданных программистом маршрутов продвижения динамических объектов, называемых далее транзактами (сообщениями);

- планирование событий, происходящих в модели, путем регистрации времени наступления каждого события и выполнения их в нарастающей временной последовательности;

- регистрация статистической информации о функционировании модели;
- продвижение модельного времени в процессе моделирования системы.

Основные сокращения и обозначения, используемые симулятором GPSS/W этапе системного ввода, приведены в разделах 2,3.

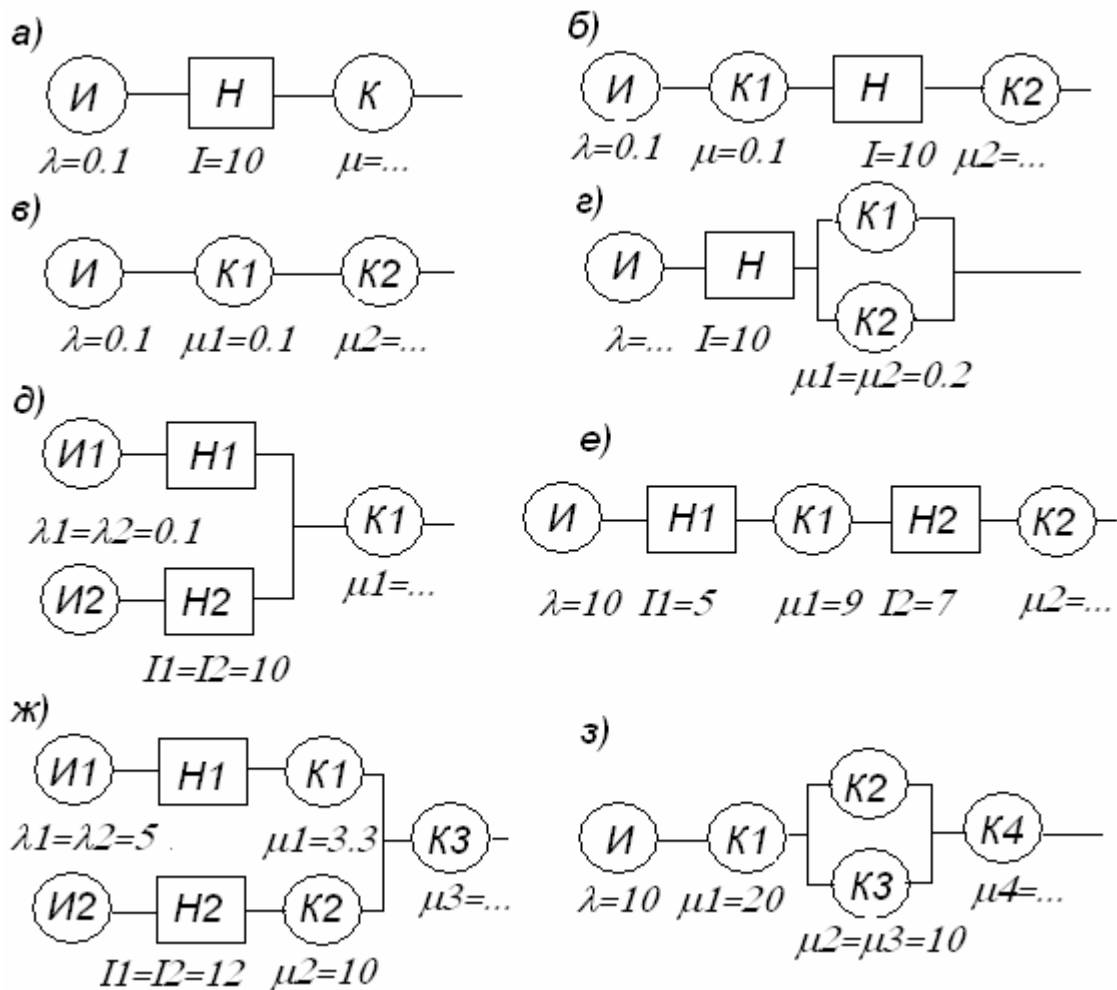


Рисунок 1 – Структуры вариантов Q-схем

На рисунке 1 приведены различные варианты одно - (а, б, в, г, д) и - двухфазных (е, ж, з) Q-схем (систем массового обслуживания).

Здесь использованы следующие обозначения:

И — источник заявок;

Н — накопитель заявок;

К — канал обслуживания;

$\lambda_i$  — интенсивность  $i$ -го потока заявок (потоки заявок в данной лабораторной работе считаются распределенными по пуассоновскому закону);

$\mu_{ij}$  — интенсивность обслуживания в  $ij$ -м канале,

$L_k$  — емкость  $k$ -того накопителя.

Для расчета недостающих параметров системы необходимо воспользоваться

аппаратом теории массового обслуживания.

Ограничением является необходимость установления стационарного режима загрузки системы  $p < 1$ .

### Задания для домашней подготовки

1 Изучите основные элементы Q-схемы и особенности их структурного представления.

2 Ознакомьтесь с возможностями перехода от концептуальных моделей систем к типовым математическим схемам.

3 Изучите основные операторы языка GPSS/W, необходимые для выполнения лабораторной работы.

4 Изучите особенности работы в среде в интерактивном режиме при выполнении лабораторной работы.

### Задания к работе

*Задание 1* Постройте модель варианта Q-схемы и обеспечьте сбор статистических данных о процессе функционирования исследуемого в работе варианта системы при условиях, задаваемых преподавателем.

Смоделируйте процесс функционирования Q-схемы на интервале времени  $(0, T)$ , где  $T = 1000$  единиц времени (ед. вр.).

Исходная GPSS-программа для выполнения п. 1 задания приведена на рисунок 2.

GIST TABLE AC1,500,400,21	;Создать таблицу Гистограмма.
GENERATE 100000	;Внешний цикл.
TERMINATE 1	;Уничтожить транзакт ВЦ.
GENERATE (Exponential(1,0,100))	;Генерация транзактов.
QUEUE 1	;Стать в очередь.
SEIZE 1	;Занять прибор 1.
DEPART 1	;Освободить очередь.

```

ADVANCE (Exponential(2,0,50)) ;Задержать транзакт.
RELEASE 1 ;Освободить прибор 1.
TRANSFER BOTH,MET1,MET2 ;Передать в пр.2 или в пр.3.
MET2 SEIZE 2 ;Занять прибор 2.
ADVANCE (Exponential(4,0,200)) ;Задержать транзакт.
RELEASE 2 ;Освободить прибор 2.
TRANSFER ,MET3 ;Передать в пр.4.
MET1 SEIZE 3 ;Занять прибор 3.
ADVANCE (Exponential(4,0,200)) ;Задержать транзакт.
RELEASE 3 ;Освободить прибор 3.
MET3 QUEUE 4 ;Стать в очередь.
SEIZE 4 ;Занять прибор 11.
DEPART 4 ;Освободить очередь.
ADVANCE (Exponential(2,0,100)) ;Задержать транзакт.
RELEASE 4 ;Освободить прибор 4.
TABULATE GIST ;Заполнить таблицу.
TERMINATE 0 ;Уничтожить транзакт.

```

Рисунок 2 – Текст исходной программы к заданию 1

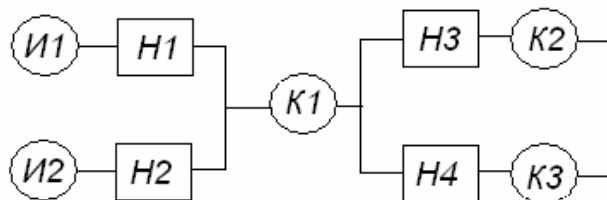


Рисунок 3 – Структура Q-схемы к заданию 2

```

SVE1 EQU 1
SVE2 EQU 2
SVE3 EQU 3
SVE4 EQU 4
SVE1 STORAGE 10
SVE2 STORAGE 10
SVE3 STORAGE 10
SVE4 STORAGE 10
GENERATE 10000 ;Внешний цикл.
TERMINATE 1 ;Уничтожить транзакт ВЦ.
GENERATE (Exponential(1,0,50)) ;Источник1.

```

```

ASSIGN      1,1          ;Разметка транзактов от ист.1.
ENTER      1,1          ;Вход в накопитель 1.
TRANSFER   ,WAY
GENERATE    (Exponential(1,0,25)) ;Источник2.
ASSIGN      1,2          ;Разметка транзактов от ист.1.
ENTER      2,1          ;Вход в накопитель 1.
WAY SEIZE   1           ;Занять прибор 1.
LEAVE      P1           ;Освободить исх. накопитель.
ADVANCE    17           ;Обработка в приборе 1.
RELEASE    1           ;Освободить прибор 1.
TEST E     P1,1,KAN3    ;Переход к фазе2.
ENTER      3,1          ;Занятие накопителя 3.
SEIZE      2           ;Занять прибор 2.
LEAVE      3,1          ;Освободить накопитель 3.
ADVANCE    50           ;Обработка в приборе 2.
RELEASE    2           ;Освободить прибор 2.
TERMINATE  0
KAN3 ENTER  4,1          ;Занятие накопителя 4.
SEIZE      3           ;Занять прибор 3.
LEAVE      4,1          ;Освободить накопитель 4.
ADVANCE    25           ;Обработка в приборе 3.
RELEASE    3           ;Освободить прибор 3.
TERMINATE  0
GENERATE    200
SAVEVALUE  1+,1
SAVEVALUE  X1,FR1
TERMINATE  0

```

Рисунок 4 - Текст исходной программы к заданию 2

*Задание 2.* Необходимо исследовать на машинной модели *Q-схему*, структура которой приведена на рис 3. Исследуемая система представляет собой двухфазную *Q-схему* с отказами. В качестве выходных потоков системы рассматриваются: потоки обслуженных заявок *NO* и потерянных заявок двух типов *N1* и *N2*. Предполагается, что интервалы времени между моментами поступления заявок из источников *I1* и *I2* имеют экспоненциальное распределение, а время обслуживания в каналах *K1* *K2* и *K3* постоянно. Интенсивности поступления заявок из источников *I1* и *I2* соответственно равны  $\lambda_1 = 0,021$  1/ед вр. и  $\lambda_2 = 0,041$  1/ед. вр., а емкости

накопителей  $H_1, H_2, H_3$  и  $H_4$  —  $L_1 = L_3 = L_4 = 10$  и  $L_2 = 20$ . Время обслуживания заявок в каналах  $K_1, K_2$  и  $K_3$  задано в исходной программе. Заявки от источников  $I_1$  и  $I_2$  обслуживаются каналами  $K_2$  и  $K_3$  соответственно.

Необходимо провести моделирование процесса функционирования данной  $Q$ -схемы на интервале времени  $(0, T)$ ,  $T = 1000$  ед. вр.

При выполнении конкретного варианта п. 2 задания необходимо (по указанию преподавателя) в исходную программу, текст которой приведен на рис. 4, внести изменения, обеспечивающие:

1) обработку в канале  $K_1$  заявок от источника  $I_1$  в течение 15 ед. вр., а от источника  $I_2$  — в течение 20 ед. вр.;

2) получение функции распределения времени обслуживания в первой фазе данной  $Q$ -схемы по каждому типу заявок отдельно;

3) получение функции распределения времени обслуживания во второй фазе данной  $Q$ -схемы по каждому типу заявок отдельно;

4) получение функции распределения полного времени обслуживания по каждому типу заявок отдельно;

5) определение вероятностей переполнения накопителей  $Q$ -схемы;

6) определение вероятностей отказа в обслуживании при переполнении накопителей по каждому типу заявок отдельно;

7) определение функций распределения длин очередей заявок в накопителях  $Q$ -схемы.

8) построение и вывод на печать графика изменения загрузки канала  $K_1$  на интервале времени моделирования  $(0, T)$ .



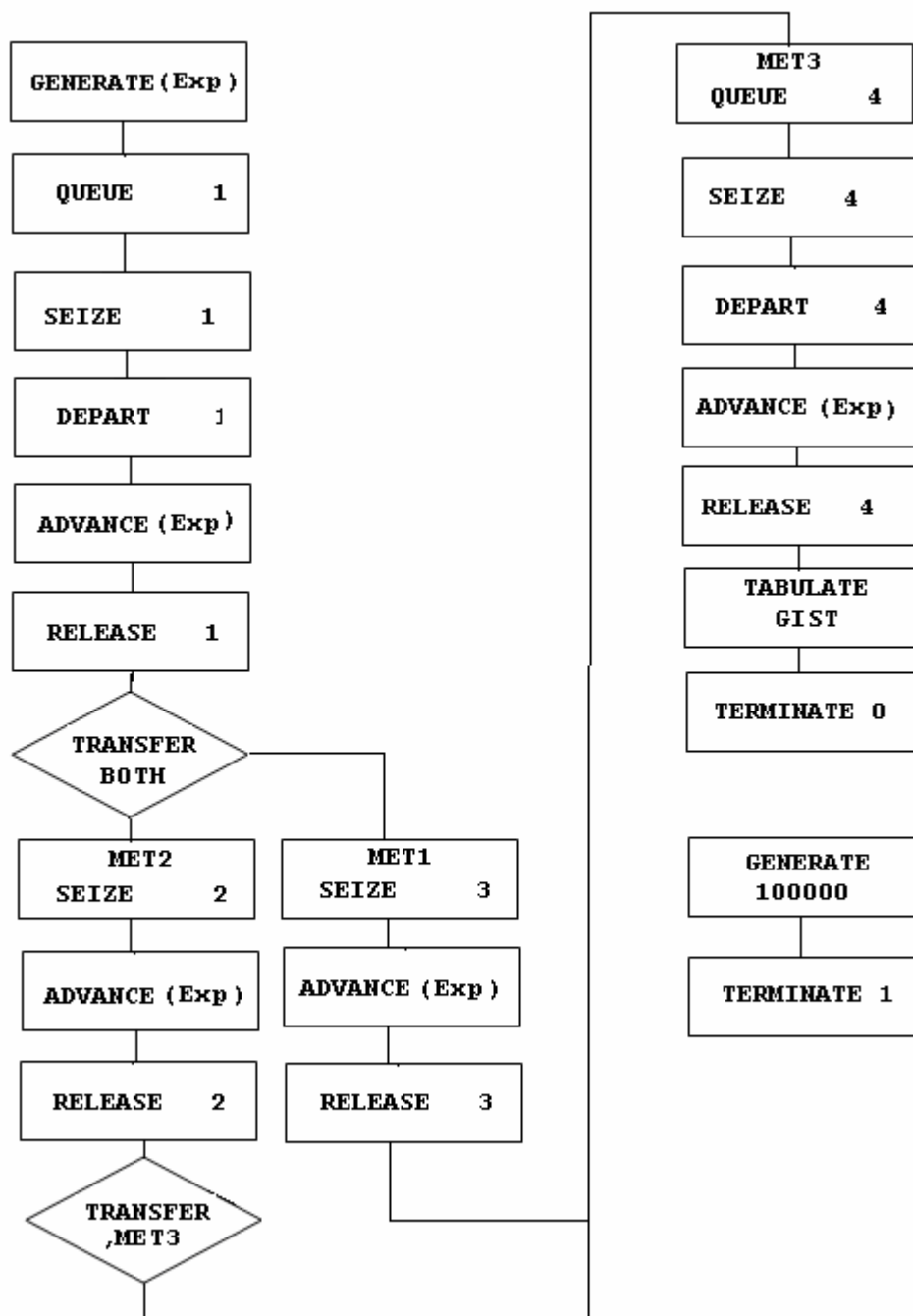


Рисунок 5 - Блок-диаграмма GPSS-модели Q-схемы

### Порядок выполнения работы

1 Ознакомиться с методическими указаниями по выполнению данной лабораторной работы.

2 Получить варианты заданий 1 и 2 и провести необходимые предварительные

аналитические расчеты.

3 Составить по заданному варианту блок-диаграмму *GPSS* и программу на языке *GPSS/W*.

4 Провести имитационный эксперимент на ПЭВМ, получить результаты прогона модели и сравнить их с расчетными.

5 Разработать мероприятия по ликвидации «узких мест» исследуемой системы, представленной в виде *Q-схемы*.

### Пример выполнения работы

Рассмотрим пример выполнения лабораторной работы по двум пунктам задания, рассмотренным ранее.

*Задание 1.* Пусть в качестве заданного варианта объекта моделирования выступает *Q-схема*, структура и параметры которой показаны на рис. 1з.

Исходя из заданных условий и необходимости обеспечения ограничений по загрузке системы  $p = 1$ , получаем

$$t_4 = \min\{1/\lambda_1; t_1; t_{2,3}\},$$

где  $t_{2,3} = 1/(1/t_2 + 1/t_3) = t_2 t_3 / (t_2 + t_3)$ , единиц времени.

Таким образом, для данной *Q-схемы* после подстановки исходных данных имеем  $t_4 = 0.05$  единиц времени.

Блок-диаграмма *GPSS* модели процесса функционирования исследуемой *Q-схемы* с использованием символики, показана на рис. 5. Текст соответствующей *GPSS*-программы приведен на рисунке 2. Результаты моделирования, выведенные на печать, показаны на рисунке 6.

По результатам использования таких статистических категорий *GPSS* как очереди и таблицы можно сделать следующие выводы: в блоках 7 и 15 существуют устойчивые очереди заявок, что является следствием выбора загрузки  $p = 1$ .

При помощи статистической таблицы строится функция распределения времени пребывания заявок в *Q-схеме*, гистограмма которой выведена на печать (рисунок 8).

GPSS World Simulation Report - Sample04.46.1

Wednesday, January 25, 2012 07:41:39

START TIME		END TIME		BLOCKS	FACILITIES	STORAGES			
0.000		100000.000		23	4	0			
LABEL	LOC	BLOCK	TYPE	ENTRY	COUNT	CURRENT	COUNT	RETRY	
	1	GENERATE		1		0		0	
	2	TERMINATE		1		0		0	
	3	GENERATE		983		0		0	
	4	QUEUE		983		0		0	
FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
1	983	0.483	49.112	1	984	0	0	0	0
2	485	0.957	197.240	1	952	0	0	31	0
3	466	0.973	208.889	1	950	0	0	31	0
4	894	0.977	109.275	1	894	0	0	0	55
QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY	
1	8	0	983	489	0.438	44.520	88.589	0	
4	88	55	949	20	43.638	4598.361	4697.357	0	
TABLE	MEAN	STD.DEV.	RANGE		RETRY		FREQUENCY	CUM.%	
GIST	6401.389	2945.539			0				
			—	-	500.000		4	0.45	
			500.000	-	900.000		14	2.02	
			900.000	-	1300.000		33	5.71	
			1300.000	-	1700.000		35	9.63	
			1700.000	-	2100.000		20	11.87	
			2100.000	-	2500.000		10	12.99	
			2500.000	-	2900.000		41	17.58	
			2900.000	-	3300.000		46	22.73	
			3300.000	-	3700.000		48	28.11	
			3700.000	-	4100.000		21	30.46	
			4100.000	-	4500.000		17	32.36	
			4500.000	-	4900.000		17	34.27	
			4900.000	-	5300.000		20	36.51	
			5300.000	-	5700.000		16	38.30	
			5700.000	-	6100.000		17	40.20	

6100.000	-	6500.000	20	42.44
6500.000	-	6900.000	14	44.01
6900.000	-	7300.000	8	44.90
7300.000	-	7700.000	7	45.69
7700.000	-	8100.000	64	52.86
8100.000	-	8500.000	89	62.82
8500.000	-	8900.000	115	75.70
8900.000	-	9300.000	117	88.80
9300.000	-	9700.000	61	95.63
9700.000	-	10100.000	35	99.55
10100.000	-	10500.000	4	100.00

Рисунок 6 – Результаты моделирования  $Q$ -схемы

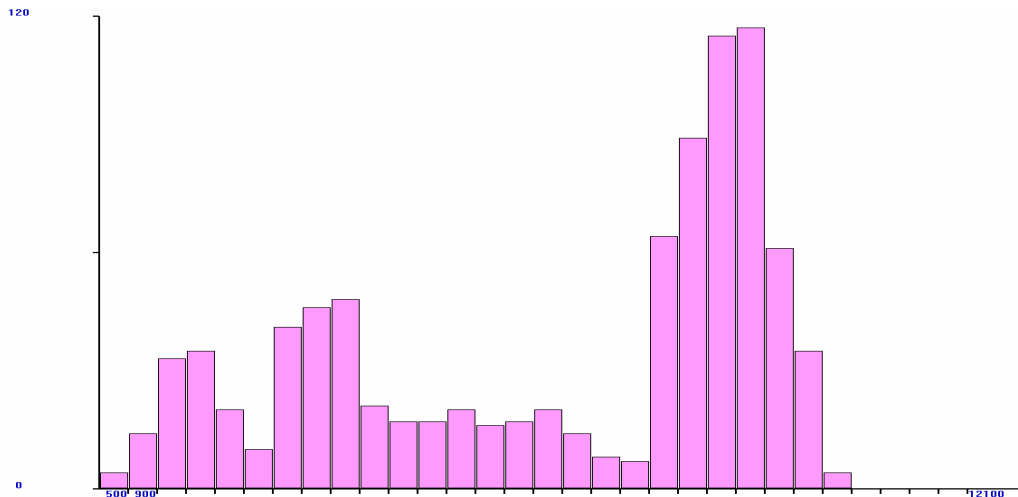


Рисунок 7 – Гистограмма распределения времени пребывания заявок в  $Q$ -схеме

*Задание 2.* Пусть необходимо в данном случае выполнить п. 8 этого задания, т. е. построить и вывести на печать график изменения загрузки канала К1 на интервале  $(O, T)$ . Для выполнения этого задания выберем количество точек в требуемом графике, равным 50. Тогда интервал времени между моментами снятия отчетов будет равен 200 усл. ед. вр. Формирование графика осуществляется дополнительной специализированной частью *GPSS*-программы, выполняемой параллельно с основной программой.

## **Отчет о работе**

### **Отчет должен содержать:**

- 1) пункты 1 и 2 задания и исходные данные по вариантам;
- 2) краткое описание используемых для моделирования программных и технических средств;
- 3) обоснование выбора конкретных статистических объектов *GPSS* для реализации данного варианта задания;
- 4) обоснование принятых решений при выполнении заданий 1 и 2;
- 5) текст результирующих программ с внесенными изменениями и окончательные результаты моделирования системы.

### **Задания для домашней подготовки**

- 1 Приведите элементы *Q-схем*, используемые для структурного представления процессов массового обслуживания
- 2 Укажите параметры и характеристики модели системы, подлежащие масштабированию в случае нецелочисленных значений.
- 3 Перечислите блоки *GPSS*, используемые для определения характеристик очередей и потоков заявок в системе.
- 4 Опишите, какие меры можно использовать для выявления при моделировании «узких мест» в исследуемой системе.

## **4.2 Лабораторная работа № 6**

### **Планирование экспериментов с имитационными моделями систем массового обслуживания**

**Цель работы:** изучение методов планирования машинных экспериментов с моделями систем, формализуемых в виде схем массового обслуживания (*Q-схем*), приобретение навыков решения конкретных задач указанного типа в режиме

диалога с компьютером, проведение имитационных экспериментов в соответствии с построенным планом в среде моделирования *GPSS/W*.

Напомним краткие сведения из теории планирования эксперимента. Эффективность машинных экспериментов с имитационными моделями систем, формализуемых в виде Q-схем (схем массового обслуживания), существенно зависит от выбора плана эксперимента, так как именно план определяет объем и порядок проведения вычислений на ПЭВМ, приемы накопления и статистической обработки результатов моделирования системы и в целом влияет на эффективность использования ресурсов компьютера при моделировании.

Математические методы планирования экспериментов основаны на кибернетическом представлении процесса проведения эксперимента, наиболее подходящей моделью которого является абстрактная схема типа «черного ящика» вида  $Y = \varphi(X)$ , где  $Y = (y_1, y_2, \dots, y_k)$  — множество векторов зависимых выходных переменных, называемых реакциями;  $X = (x_1, x_2, \dots, x_m)$  — множество векторов входных независимых переменных, называемых факторами.

Функция, связывающая реакцию с факторами, называется функцией реакции.

При проведении машинного эксперимента с моделью для оценки характеристик процесса 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  $x_i, i = 1, \dots, m$ , влияющие на искомую характеристику, и описать функциональную зависимость; 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 число реализаций и их порядок в эксперименте [2].

Свойства объекта исследования, т. е. процесса машинного моделирования системы, можно описывать с помощью различных методов (моделей планирования).

Получение модели, описывающей реакции изучаемой системы на многофакторное возмущение — одна из задач планирования эксперимента. Наиболее распространенными и полно отвечающими задачам статистического

моделирования являются полиномиальные модели. Задача нахождения полиномиальной модели, описывающей систему или отдельные ее характеристики, состоит в оценке вида и параметров некоторой функции  $\{x_1, x_2, \dots, x_m\}$ .

Рассмотрим влияние  $m$  количественных факторов  $x_i, i=1, \dots, m$  на некоторую реакцию  $\eta$  в отведенной для экспериментирования локальной области  $[x_{i \min} \dots x_{i \max}, i=1, \dots, m]$  факторного пространства  $G$ . Функцию реакции  $\varphi(x_1, x_2, \dots, x_m)$  представим в виде полинома степени  $d$  от  $m$  переменных, который содержит  $C_{m+d}^d$  коэффициентов:

$$\eta = \tilde{b}_0 + \sum_{1 \leq i \leq m} \tilde{b}_i x_i + \sum_{1 \leq j < i \leq m} \tilde{b}_{ij} x_j x_i + \sum_{i_1, i_2, \dots, i_m} \tilde{b}_{i_1 i_2 \dots i_m} x_1^{i_1} x_2^{i_2} \dots x_m^{i_m}, \sum ij = d.$$

Для оценки коэффициентов данного уравнения используем методы линейной регрессии.

После выбора модели планирования следующей задачей является планирование и проведение эксперимента для оценки числовых значений коэффициентов используемого уравнения. Так как полином содержит  $C_{m+d}^d$  коэффициентов, подлежащих определению, то план эксперимента  $D$  должен содержать по крайней мере  $N \geq C_{m+d}^d$  различных экспериментальных точек

$$D = \begin{vmatrix} x_{11} x_{21} \dots x_{m1} \\ x_{12} x_{22} \dots x_{m2} \\ \dots \dots \dots \\ x_{1N} x_{2N} \dots x_{mN} \end{vmatrix},$$

где  $x_{iu}$  — значения, которые принимает  $i$ -я переменная в  $u$ -м испытании,  $i=1, 2, \dots, m, u=1, 2, \dots, N$ .

Реализовав испытания в  $N$  точках области факторного пространства, отведенной для экспериментирования, получим вектор наблюдений, имеющий следующий вид:





Для оценки свободного члена  $b_0$  и определения эффектов взаимодействия  $b_{12}, b_{13}, \dots, b_{123}$  план эксперимента  $D$  расширяют до матрицы планирования  $X$  путем добавления соответствующей «фиктивной переменной»: единичного столбца  $x_0$  и столбцов произведений  $\tilde{x}_1\tilde{x}_2, \tilde{x}_1\tilde{x}_3, \dots, \tilde{x}_1\tilde{x}_2\tilde{x}_3, \dots$ . Но при этом количество испытаний в ПФЭ значительно превосходит число определяемых коэффициентов линейной модели плана эксперимента, т. е. ПФЭ обладает большой избыточностью и поэтому возникает проблема сокращения их количества. В связи с этим эффективно применение дробных факторных экспериментов (ДФЭ). Правило проведения ДФЭ следующее: *для сокращения числа испытаний новому фактору присваивается значение вектор-столбца матрицы, принадлежащего взаимодействию, которым можно пренебречь*. Использование ДФЭ позволяет в два и более раз уменьшить число испытаний, что значительно сокращает затраты ресурсов ПЭВМ на проведение машинных экспериментов с моделями систем, а также позволяет сохранить все основные свойства ПФЭ (ортогональность, симметричность, условие нормировки).

### **Задания для домашней подготовки**

- 1 Изучите методы планирования машинных экспериментов с моделями систем
- 2 Ознакомьтесь с особенностями проведения и интерпретации результатов имитационных экспериментов с моделями массового обслуживания (*Q-схемами*).
- 3 Изучите особенности работы среды моделирования *GPSS/W*.

### **Задание к работе**

Провести анализ зависимости влияния независимых переменных модели системы, формализованной в виде *Q-схемы*, на зависимые переменные с построением плана эксперимента при использовании метода наименьших квадратов и линейного регрессионного анализа.

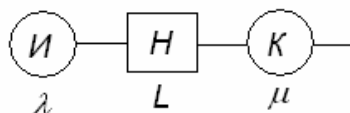


Рисунок 1 - Q-схема исследуемой СМО

В качестве объекта моделирования используется Q-схема, структура которой показана на рисунке 1. Здесь обозначено: И - источник заявок, Н - накопитель для хранения заявок, К - канал обслуживания,  $\lambda$  - интенсивность входящего потока заявок,  $\mu$  - интенсивность обслуживания,  $L$  - емкость накопителя.

Варианты лабораторной работы показаны в таблице 1, где ПФЭ и ДФЭ - полный и дробный факторные эксперименты соответственно;  $T$  - время задержки заявки,  $\bar{n}$  - средняя длина очереди заявок,  $K(\rho)$  - коэффициент загрузки.

Таблица 1- Варианты лабораторной работы

Вариант	Тип плана	Переменные					
		независимые				зависимые	
		$\lambda$	$\mu$	$L$	$T$	$\bar{n}$	$K(\rho)$
1	ДФЭ	+	+	+	+		
2	ДФЭ	+	+			+	
3	ПФЭ	+	+	+			+
4	ДФЭ	+	+		+		
5	ПФЭ	+	+	+		+	
6	ДФЭ	+	+	+	+		+
7	ДФЭ	+	+				
8	ДФЭ	+	+			+	

## Порядок выполнения работы

- 1 Ознакомиться с методическими указаниями по выполнению данной лабораторной работы.
- 2 Получить у преподавателя вариант задания на составление плана машинного эксперимента с моделью  $Q$ -схемы.
- 3 Записать формальную постановку задачи в соответствии с полученным заданием.
- 4 Подготовить исходные данные, необходимые для решения поставленной задачи планирования машинного эксперимента с использованием ПЭВМ.
- 5 Провести необходимые расчеты по построению плана эксперимента на ПЭВМ.
- 6 В соответствии с полученным планом провести машинный эксперимент и сравнить результаты с неуправляемым экспериментом.

## Пример выполнения работы

Пусть необходимо провести машинный эксперимент по исследованию характеристик  $Q$ -схемы (вариант 3 из таблицы 1), построить план эксперимента, описать модель планирования машинного эксперимента, получить оценки коэффициентов модели и провести планируемый имитационный эксперимент с моделью  $Q$ -схемы. Исследуется однофазная одноканальная  $Q$ -схема со следующими параметрами: интенсивность поступления заявок  $\lambda = 13$  1/с; интенсивность обслуживания  $\mu = 10$  1/с; емкость накопителя заявок  $L = 10$ . В имитационном эксперименте необходимо оценить среднее время задержки в системе  $T$  при минимальных затратах машинных ресурсов.

При проведении машинного эксперимента для оценки среднего времени задержки заявки  $T$  в  $Q$ -схеме, являющегося характеристикой (реакцией) системы, необходимо определить влияние факторов, находящихся в функциональной связи с искомой характеристикой. Для этого отберем факторы, влияющие на искомую

характеристику, и опишем функциональную зависимость; определим уровни выбранных факторов, фиксированный набор которых определяет одно из возможных состояний рассматриваемой *Q-схемы*. Одновременно этот набор является условием проведения машинного эксперимента. Каждому фиксированному набору уровней факторов соответствует определенная точка в многомерном пространстве, называемом факторным. Реализация этих задач является неформализованной в планировании эксперимента.

Для составления плана эксперимента выделим следующие факторы: интенсивность поступления заявок  $x_1 = \lambda$ ; интенсивность обслуживания заявок  $x_2 = \mu$ ; емкость буфера  $x_3 = L$ .

Определяем локальную подобласть планирования эксперимента путем выбора основного (нулевого) уровня  $x_{i0}$  и интервалов варьирования для каждого выбранного фактора. Сведем полученные данные в таблицу 2.

Таблица 2 – Интервалы варьирования

Факторы	Уровни факторов			Интервалы варьирования, $l/c$
	-1	0	+1	
$x_1$	10	13	20	5
$x_2$	6	10	14	4
$x_3$	4	7	10	3

Существует вполне определенная зависимость между уровнями факторов и реакцией системы, которую представим в виде соотношения

$$y = \varphi(x_1, x_2, x_3).$$

Для определения этой зависимости строим математическую (аналитическую) модель планирования в виде полинома первого порядка

$$y = b_0 + b_1\tilde{x}_1 + b_2\tilde{x}_2 + b_3\tilde{x}_3 + b_{12}\tilde{x}_1\tilde{x}_2 + b_{13}\tilde{x}_1\tilde{x}_3 + b_{23}\tilde{x}_2\tilde{x}_3 + b_{123}\tilde{x}_1\tilde{x}_2\tilde{x}_3. \quad (1)$$

Выбранная модель включает линейные члены полинома и их произведения. Для оценки коэффициентов модели используем план эксперимента типа  $2^k$  с варьированием всех  $k$  факторов на двух уровнях, т. е.  $q = 2$ .

Полный факторный эксперимент дает возможность определить коэффициенты регрессии, соответствующие не только линейным эффектам, но и всем эффектам взаимодействий. План ПФЭ представим в виде матрицы планирования, помещенной в таблице 3.

Таблица 3 - ПФЭ типа  $2^3$

Номер опыта	План ПФЭ								Реакция $y$
	$x_0$	$x_1$	$x_2$	$x_3$	$x_1x_2$	$x_1x_3$	$x_2x_3$	$x_1x_2x_3$	
1	+1	-1	-1	-1	+1	+1	+1	-1	$y_1$
2	+1	-1	-1	+1	+1	-1	-1	+1	$y_2$
3	+1	-1	+1	-1	-1	+1	-1	+1	$y_3$
4	+1	-1	+1	+1	-1	-1	+1	-1	$y_4$
5	+1	+1	-1	-1	-1	-1	+1	+1	$y_5$
6	+1	+1	-1	+1	-1	+1	-1	-1	$y_6$
7	+1	+1	+1	-1	+1	-1	-1	-1	$y_7$
8	+1	+1	+1	+1	+1	+1	+1	+1	$y_8$

В то же время количество испытаний в ПФЭ значительно превосходит число определяемых коэффициентов, т. е. обладает большой избыточностью. Одним из способов уменьшения этой избыточности является применение ДФЭ, позволяющего значительно сократить количество испытаний и сэкономить ресурсы ЭВМ. Используем ДФЭ типа  $2^{(3-1)}$ , что позволит сократить количество опытов в 2 раза. Матрица планирования представлена в таблице 4.

Таблица 4 - ДФЭ типа  $2^{(3-1)}$

Номер опыта $N$	$x_3=x_1x_2, x_0=x_1x_2x_3$				$y$	$T$
1	+ 1	-	-	+	$y_1$	6.527
2	+ 1	+	+	+	$y_2$	15.197
3	+ 1	+	-	-	$y_3$	5.684
4	+ 1	-	+	-	$y_4$	16.294

Проведение опыта в  $i$ -й точке факторного пространства сводится к следующей процедуре:

по нормированным значениям факторов  $i$ -й строки плана эксперимента определяем натуральные значения координат

$$x_N = \|x_{1N}, x_{2N}, \dots, x_{mN}\|;$$

факторы устанавливаем на уровни, соответствующие координатам точки  $x_N$ ;  
измеряем отклик (реакцию) системы  $y_{Ni}$ .

Определим реакцию  $Q$ -схемы при выбранных уровнях факторов

$$T_i = 1/(\mu_i C_i - \lambda_i),$$

и занесем данные в таблицу 4.

Используя метод наименьших квадратов, определим коэффициенты регрессионной модели:

$$b_j = \sum_{i=1}^N x_{ji} y_i / N, j = \overline{0, k}, \quad (2)$$

где  $N$  — число опытов;  $j$  — число факторов.

Определение коэффициентов выбранной модели планирования эксперимента состоит в следующем:

по нормированным значениям факторов  $i$ -й строки плана эксперимента суммируем значения реакций, полученные при выбранных уровнях факторов;

полученный результат делим на число опытов в эксперименте  $N$ .

В соответствии с приведенной процедурой определяем коэффициенты модели планирования эксперимента по формуле (2) с учетом знаков в столбцах таблицы 4

$$b_0 = (y_1 + y_2 + y_3 + y_4) / 4 = (6.527 + 15.197 + 5.684 + 16.294) / 4 = 10.9255;$$

$$b_1 = (-y_1 + y_2 + y_3 - y_4) / 4 = (-6.527 + 15.197 + 5.684 - 16.294) / 4 = -0.485;$$

$$b_2 = (-y_1 + y_2 + y_3 - y_4) / 4 = (-6.527 + 15.197 - 5.684 + 16.294) / 4 = 4.82;$$

$$b_3 = (-y_1 + y_2 + y_3 - y_4) / 4 = (+6.527 + 15.197 - 5.684 - 16.294) / 4 = -0.0635.$$

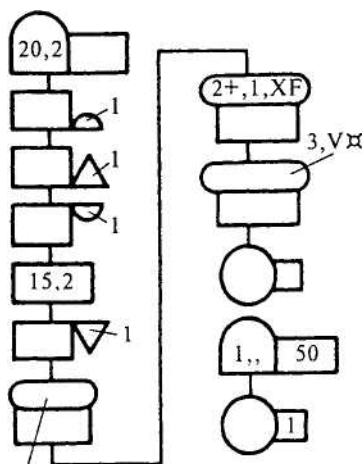


Рисунок 3 - Блок-диаграмма GPSS-модели Q-схемы

Определим значения коэффициентов и, подставив их в выбранную нами модель планирования эксперимента (1), получим

$$Y = 10.9255 - 0.485X_1 + 4.82X_2 - 0.0635X_3. \quad (3)$$

Следующим (заключительным) этапом планирования эксперимента является интерпретация полученных результатов и принятия решения, т. е. определение

## оценок величины и направления влияния факторов на реакцию системы.

```

SVE EQU 1
SVE STORAGE 10
AAA1 VARIABLE X$SVR/X$SUM
VREM1 MATRIX ,1,1
    GENERATE (Exponential(1,0,8)) ;Генерация транзактов.
    ENTER 1 ;Стать в очередь.
    SEIZE 1 ;Занять прибор 1.
    LEAVE 1 ;Освободить очередь.
    ADVANCE (Exponential(1,0,6)) ;Задержать транзакт.
    RELEASE 1 ;Освободить прибор 1.
    SAVEVALUE SVR+,M1
    SAVEVALUE SUM+,1
    MSAVEVALUE VREM1,1,1,V$AAA1
    TERMINATE

    SVE EQU 2
SVE STORAGE 10
AAA2 VARIABLE X$SVR/X$SUM
VREM2 MATRIX ,1,1
    GENERATE (Exponential(1,0,18)) ;Генерация транзактов.
    ENTER 2 ;Стать в очередь.
    SEIZE 2 ;Занять прибор 2.
    LEAVE 2 ;Освободить очередь.
    ADVANCE (Exponential(1,0,14)) ;Задержать транзакт.
    RELEASE 2 ;Освободить прибор 2.
    SAVEVALUE SVR+,M1
    SAVEVALUE SUM+,1
    MSAVEVALUE VREM2,1,1,V$AAA2
    TERMINATE

    SVE EQU 3
SVE STORAGE 4
AAA3 VARIABLE X$SVR/X$SUM
VREM3 MATRIX ,1,1
    GENERATE (Exponential(1,0,18)) ;Генерация транзактов.
    ENTER 3 ;Стать в очередь.
    SEIZE 3 ;Занять прибор 3.
    LEAVE 3 ;Освободить очередь.
    ADVANCE (Exponential(1,0,6)) ;Задержать транзакт.
    RELEASE 3 ;Освободить прибор 3.

```



```

SAVEVALUE SVR+,M1
SAVEVALUE SUM+,1
MSAVEVALUE VREM3,1,1,V$AAA3
TERMINATE

SVE EQU 4
SVE STORAGE 4
AAA4 VARIABLE X$SVR/X$SUM
VREM4 MATRIX ,1,1
GENERATE (Exponential(1,0,8)) ;Генерация транзактов.
ENTER 4 ;Стать в очередь.
SEIZE 4 ;Занять прибор 4.
LEAVE 4 ;Освободить очередь.
ADVANCE (Exponential(1,0,14)) ;Задержать транзакт.
RELEASE 4 ;Освободить прибор 4.
SAVEVALUE SVR+,M1
SAVEVALUE SUM+,1
MSAVEVALUE VREM4,1,1,V$AAA4
TERMINATE 1

```

Рисунок 4 - Текст программы моделирования *Q-схемы*

Анализируя выражение (3) и используя правила интерпретации, делаем вывод, что на реакцию системы (время задержки заявки) наиболее сильно влияет фактор  $X_2$  (интенсивность обслуживания заявки), а влияние первого и третьего факторов (интенсивности поступления заявок и пропускной способности обслуживающего канала) неодинаково. Знаки перед коэффициентами  $X_1, X_3$  указывают, что при увеличении значений этих факторов реакция системы убывает, а при увеличении значения фактора  $X_2$  реакция системы увеличивается.

Результаты проведенного анализа показывают, что можно варьировать значения факторов с целью оптимизации реакции системы, т. е. решать следующую оптимизационную задачу  $T_{зад} \rightarrow \min$ , либо окончить эксперимент.

Блок-диаграмма *GPSS*-модели процесса функционирования исследуемой *Q-схемы* показана на рис. 3. Текст *GPSS*-программы приведен на рисунке 4.

Анализ проведенного планируемого эксперимента показывает, что для получения одних и тех же результатов (среднего времени задержки заявки в *Q-схеме*

Т) без использования метода планирования эксперимента необходимо провести восемь опытов, т.е. реализовать восемь прогонов модели на ПЭВМ, а с учетом использования построенного плана ДФЭ для получения того же результата достаточно в 2 раза меньше опытов, т.е. можно обойтись четырьмя прогонами модели на ПЭВМ. Так решается задача получения максимальной информации об объекте моделирования с минимальными затратами машинных ресурсов на проведение эксперимента.

### **Отчет о работе**

Отчет должен содержать:

- 1) задание и исходные данные для выполнения работы;
- 2) текст программы для определения коэффициентов модели планирования эксперимента;
- 3) блок-диаграмму и *GPSS*-программу имитационного эксперимента с моделью исследуемого варианта *Q-схемы*;
- 4) результаты обработки экспериментальных данных, анализ полученных результатов и выводы по работе.

### **Задания для самопроверки**

- 1 Приведите критерии целесообразности использования при моделировании на ЭВМ метода планирования экспериментов.
- 2 Объясните процедуру формирования плана эксперимента.
- 3 Назовите особенности планирования машинных экспериментов.
- 4 Сформулируйте требования к модели планирования машинного эксперимента.

## Список использованных источников

1 Вентцель, Е.С. Задачи и упражнения по теории вероятностей: учебное пособие для студ. Втузов / Е.С. Вентцель, Л.А. Овчаров. – М.: Издательский центр «Академия», 2003. – 448 с.

2 Советов, Б.Я. Моделирование систем: учебное пособие-практикум / Б.Я. Советов, С.А. Яковлев. – М.: Высшая школа, 2003. – 343 с.

3 Паничев, В.В. Компьютерное моделирование: Электронное учебное пособие / В.В. Паничев, Н.А. Соловьев. – Оренбург: ОГУ, 2009. – 132 с.

4 Руководство пользователя по GPSS World; Перевод с английского - Казань: Изд-во «Мастер Лайн», 2002. – 384 с.