

на содержание документа

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ГРАЖДАНСКОЙ АВИАЦИИ

Е.Г. Мурачев

МОДЕЛИРОВАНИЕ СИСТЕМ

ПОСОБИЕ

по выполнению лабораторных работ

*для студентов III курса
специальности 220100
дневного отделения*

Москва – 2006

МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ГРАЖДАНСКОЙ АВИАЦИИ

**Кафедра вычислительных машин, комплексов,
систем и сетей**
Е.Г.Мурачев

МОДЕЛИРОВАНИЕ

ПОСОБИЕ
по выполнению лабораторных работ

*для студентов IV курса
специальности 230101
дневного обучения*

Москва – 2006

Лабораторная работа №1

Основы имитационного моделирования с помощью языка GPSS

Цель работы

1. Знакомство со структурой языка GPSS
2. Создание программы в среде программирования GPSS
3. Принципы построения модели на основе теории очередей
4. Составление программного кода для модели с параллельной структурой
5. Изучение объекта исследования с помощью составленной модели
6. Анализ результатов моделирования

ОБЩИЕ СВЕДЕНИЯ

МАТЕМАТИЧЕСКИЕ СХЕМЫ МОДЕЛИРОВАНИЯ СИСТЕМ

Наибольшие затруднения и наиболее серьезные ошибки при моделировании возникают при переходе от содержательного к формальному описанию объектов исследования. Эффективным является язык математических схем, позволяющий во главу угла поставить вопрос об адекватности перехода от содержательного описания системы к ее математической схеме, а лишь затем решать вопрос о конкретном методе получения результатов с использованием ЭВМ: аналитическом или имитационном, а возможно, и комбинированном, т. е. аналитико-имитационном. Применительно к конкретному объекту моделирования, т. е. к сложной системе, разработчику модели должны помочь конкретные, уже прошедшие апробацию для данного класса систем математические схемы, показавшие свою эффективность в прикладных исследованиях на ЭВМ и получившие название типовых математических схем.

2.1. ОСНОВНЫЕ ПОДХОДЫ К ПОСТРОЕНИЮ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ СИСТЕМ

Исходной информацией при построении математических моделей процессов функционирования систем служат данные о назначении и условиях работы исследуемой (проектируемой) системы S . Эта информация определяет основную цель моделирования системы S и позволяет сформулировать требования к разрабатываемой математической модели M . Причем уровень абстрагирования зависит от круга тех вопросов, на которые исследователь системы хочет получить ответ с помощью модели, и в какой-то степени определяет выбор математической схемы.

Математические схемы. Введение понятия «математическая схема» позволяет рассматривать математику не как метод расчета, а как метод мышления, как средство формулирования понятий, что является наиболее важным при переходе от словесного описания системы к формальному представлению процесса ее функционирования в виде некоторой математической модели (аналитической или имитационной). При пользовании математической схемой исследователя системы S в первую очередь должен интересовать вопрос об адекватности отображения в виде конкретных схем реальных процессов в исследуемой системе, а не возможность

получения ответа (результата решения) на конкретный вопрос исследования. Например, представление процесса функционирования информационно-вычислительной системы коллективного пользования в виде сети схем массового обслуживания дает возможность хорошо описать процессы, происходящие в системе, но при сложных законах распределения входящих потоков и потоков обслуживания не дает возможности получения результатов в явном виде

Математическую схему можно определить как звено при переходе от содержательного к формальному описанию процесса функционирования системы с учетом воздействия внешней среды, т. е. имеет место цепочка «описательная модель — математическая схема — математическая [аналитическая или (и) имитационная] модель».

Каждая конкретная система S характеризуется набором свойств, под которыми понимаются величины, отражающие поведение моделируемого объекта (реальной системы) и учитывающие условия ее функционирования во взаимодействии с внешней средой (системой) E . При построении математической модели системы необходимо решить вопрос об ее полноте. Полнота модели регулируется в основном выбором границы «система S — среда E ». Также должна быть решена задача упрощения модели, которая помогает выделить основные свойства системы, отбросив второстепенные. Причем отнесение свойств системы к основным или второстепенным существенно зависит от цели моделирования системы (например, анализ вероятностно-временных характеристик процесса функционирования системы, синтез структуры системы и т. д.).

Формальная модель объекта. Модель объекта моделирования, т. е. системы S , можно представить в виде множества величин, описывающих процесс функционирования реальной системы и образующих в общем случае следующие подмножества:

совокупность *входных воздействий* на систему

$$x_i \in X, \quad i=1,2,\dots,n;$$

совокупность *воздействий внешней среды*

$$v_l \in V, l=1,2,\dots,n;$$

совокупность *внутренних (собственных) параметров* системы

$$h_k \in H, k=1,2,\dots,n;$$

совокупность *выходных характеристик* системы

$$y_j \in Y, j=1,2,\dots,n;$$

При этом в перечисленных подмножествах можно выделить управляемые и неуправляемые переменные. В общем случае x , v , h , y являются элементами непересекающихся подмножеств и содержат как детерминированные, так и стохастические составляющие.

При моделировании системы S входные воздействия, воздействия внешней среды E и внутренние параметры системы являются *независимыми (экзогенными) переменными*, а выходные характеристики системы являются *зависимыми (эндогенными) переменными*

Процесс функционирования системы S описывается во времени оператором F_s , который в общем случае преобразует экзогенные переменные в эндогенные в соответствии с соотношениями вида

$$y(t) = F_s(x, v, h, t) \quad (1.1)$$

Совокупность зависимостей выходных характеристик системы от времени для всех видов y называется *выходной траекторией* $y(t)$. Зависимость (1.1) называется *законом*

функционирования системы S и обозначается F_s . В общем случае закон функционирования системы F_s может быть задан в виде функции, функционала, логических условий, в алгоритмической и табличной формах или в виде словесного правила соответствия.

Весьма важным для описания и исследования системы S является понятие *алгоритма функционирования* A_s , под которым понимается метод получения выходных характеристик с учетом входных воздействий $x(t)$, воздействий внешней среды $v(t)$ и собственных параметров системы $h(t)$. Очевидно, что один и тот же закон функционирования F_s системы S может быть реализован различными способами, т. е. с помощью множества различных алгоритмов функционирования A_s .

Соотношение (1.1) является математическим описанием поведения объекта (системы) моделирования во времени t , т. е. отражает его динамические свойства. Поэтому математические модели такого вида принято называть *динамическими моделями (системами)*.

Для статических моделей математическая модель (1.1) представляет собой отображение между двумя подмножествами свойств моделируемого объекта Y и $\{X, V, H\}$, что в векторной форме может быть записано как

$$y=f(x, v, h). \quad (1.2)$$

Соотношения (1.1) и (1.2) могут быть заданы различными способами: аналитически (с помощью формул), графически, таблично и т. д. Такие соотношения в ряде случаев могут быть получены через свойства системы S в конкретные моменты времени, называемые *состояниями*.

Если рассматривать процесс функционирования системы S как последовательную смену состояний, то они могут быть интерпретированы как координаты точки в n -мерном фазовом пространстве, причем каждой реализации процесса будет соответствовать некоторая фазовая траектория. Совокупность всех возможных значений состояний называется *пространством состояний* объекта моделирования Z .

Состояния системы S в момент времени t_0 полностью определяются начальными условиями Z^0 , входными воздействиями $x(t)$, внутренними параметрами $h(t)$ и воздействиями внешней среды $v(t)$, которые имели место за промежуток времени $t^* - t_0$, с помощью двух векторных уравнений

$$Z(t)=\Phi(z^0, x, v, h, t) \quad (1.3)$$

$$y(t)=F(z, t) \quad (1.4)$$

Первое уравнение по начальному состоянию z^0 и экзогенным переменным x, v, h определяет вектор-функцию $z(t)$, а второе по полученному значению состояний $z(t)$ — эндогенные переменные на выходе системы $y(t)$. Таким образом, цепочка уравнений объекта «вход — состояния — выход» позволяет определить характеристики системы

$$y(t)=F\{\Phi(z^0, x, v, h, t)\} \quad (1.5)$$

В общем случае время в модели системы S может рассматриваться на интервале моделирования $(0, T)$ как непрерывное, так и дискретное.

Таким образом, под *математической моделью объекта* (реальной системы) понимают конечное подмножество переменных $\{x(t), v(t), h(t)\}$ вместе с математическими связями между ними и характеристиками $y(t)$.

Если математическое описание объекта моделирования не содержит элементов случайности или они не учитываются, т. е. если можно считать, что в этом случае

стохастические воздействия внешней среды $v(t)$ и стохастические внутренние параметры $h(t)$ отсутствуют, то модель называется *детерминированной* в том смысле, что характеристики однозначно определяются детерминированными входными воздействиями

$$y(t) = f(x, t) \quad (1.6)$$

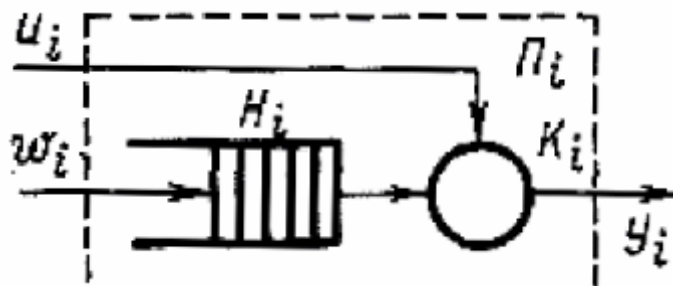
Очевидно, что детерминированная модель является частным случаем стохастической модели.

НЕПРЕРЫВНО-СТОХАСТИЧЕСКИЕ МОДЕЛИ (Q-СХЕМЫ)

Особенности непрерывно-стохастического подхода рассмотрим на примере использования в качестве типовых математических схем *систем массового обслуживания*, которые будем называть *Q-схемами*. Системы массового обслуживания представляют собой класс математических схем, разработанных в теории массового обслуживания и различных приложениях для формализации процессов функционирования систем, которые по своей сути являются процессами обслуживания.

Основные соотношения. В качестве процесса обслуживания могут быть представлены различные по своей физической природе процессы функционирования экономических, производственных, технических и других систем, например потоки поставок продукции некоторому предприятию, потоки деталей и комплектующих изделий на сборочном конвейере цеха, заявки на обработку информации ЭВМ от удаленных терминалов и т. д. При этом характерным для работы таких объектов является случайное появление заявок (требований) на обслуживание и завершение обслуживания в случайные моменты времени, т. е. стохастический характер процесса их функционирования.

В любом элементарном акте обслуживания можно выделить две основные составляющие: ожидание обслуживания заявкой и собственно обслуживание заявки.



Прибор обслуживания заявок

Это можно изобразить в виде некоторого i -го прибора обслуживания Π_i (рисунок), состоящего из накопителя заявок H_i , в котором может одновременно находиться L^H заявок, где L^H — емкость i -го накопителя и канала обслуживания заявок (или просто канала) K_i . На каждый элемент прибора обслуживания Π_i поступают потоки событий: в накопитель H_i — поток заявок w_i , на канал K_i — поток обслуживаний U_i .

Потоком событий называется последовательность событий, происходящих одно за другим в какие-то случайные моменты времени. Различают потоки однородных и неоднородных событий. Поток событий называется *однородным*, если он характеризуется только моментами поступления этих событий (вызывающими моментами)

и задается. Момент наступления i -го события — неотрицательное вещественное число. Однородный поток событий также может быть задан в виде последовательности промежутков времени между i -м и $(i-1)$ -м событиями, которая однозначно связана с последовательностью вызывающих моментов.

При моделировании различных систем применительно к элементарному каналу обслуживания K_i можно считать, что поток заявок w_i , т. е. интервалы времени между моментами появления заявок (вызывающие моменты) на входе K_i образует подмножество неуправляемых переменных, а поток обслуживания U , т. е. интервалы времени между началом и окончанием обслуживания заявки, образует подмножество управляемых переменных.

Заявки, обслуженные каналом K_i и заявки, покинувшие прибор H_i по различным причинам не обслуженными (например, из-за переполнения накопителя H_i , образуют выходной поток y_i , т. е. интервалы времени между моментами выхода заявок образуют подмножество выходных переменных.

Процесс функционирования прибора обслуживания Π можно представить как процесс изменения состояний его элементов во времени $z(t)$. Переход в новое состояние означает изменение количества заявок, которые в нем находятся (в канале K_i и в накопителе H).

В практике моделирования систем, имеющих более сложные структурные связи и алгоритмы поведения, для формализации используются не отдельные приборы обслуживания, а Q -схемы, образуемые композицией многих элементарных приборов обслуживания (сети массового обслуживания).

Для того, чтобы осуществить процесс моделирования полученной Q -схемы используют различные языки имитационного моделирования. Одним из таких языков является язык имитационного моделирования GPSS (см. приложение 1)

Порядок выполнения работы:

1. Ознакомиться с методическими указаниями по выполнению данной лабораторной работы.
2. Получить варианты заданий и провести необходимые предварительные аналитические расчеты.
3. Составить по заданному варианту блок-диаграмму и программу на языке GPSS.
4. Провести имитационный эксперимент на ЭВМ, варьируя значения задаваемых параметров, получить результаты прогона модели и сравнить их расчетными.

Отчет по работе:

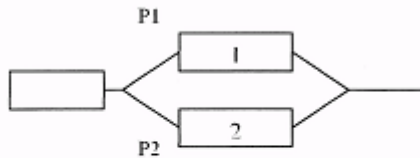
Отчет должен содержать:

1. Задание и исходные данные для выполнения работы.
2. Блок-диаграмму и GPSS-программу имитационного эксперимента с моделью исследуемого варианта системы массового обслуживания.
3. Результаты обработки экспериментальных данных, анализ полученных результатов и выводы по работе.

Моделирование Q -схемы с параллельной структурой

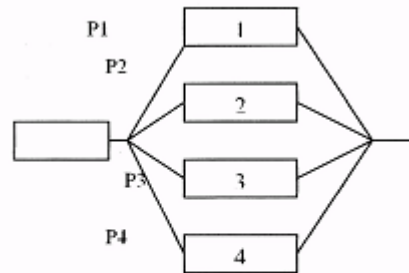
В качестве примера приведем двухканальную и четырехканальную Q -схемы

Блок диаграмма для 2-х



и

4-х канальной схемы



Пример программы для 2-х

```
simulate
generate 13,3
transfer 0.36,sim1,sim2
sim1 seize 1
advance 9,4.5
release 1
transfer ,T
sim2 seize 2
advance 3,1.5
release 2
T terminate 1
start 100
```

и 4-х канальной схемы

```
simulate
generate 13,3
transfer pick,3,6
transfer ,sim1
transfer ,sim2
transfer ,sim3
transfer ,sim4
sim1 seize 1
advance 9,4.5
release 1
transfer ,T
sim2 seize 2
advance 9,4.5
release 2
transfer ,T
sim3 seize 3
advance 3,1.5
release 3
transfer ,T
sim4 seize 4
advance 3,1.5
release 4
T terminate 1
start 100
```

Задание:

В систему массового обслуживания поступают заявки по равномерному закону распределения через $A \pm B$ минут. Обработка заявок осуществляется двумя обслуживающими каналами. Поступление заявок в тот или иной канал происходит с вероятностью P_1 и P_2 . Обслуживание заявок каждым каналом происходит по равномерному закону со временем $T_1 \pm T_2$. Провести моделирование системы с параметрами A, B, P_1, P_2, T_1, T_2 . Провести моделирование системы с параметрами $A=A+A/2, B=B+B/2, P_1, P_2, T_1=T_1+T_1/2, T_2=T_2+T_2/2$. Необходимо осуществить обработку 100 заявок при одном и двух прогонах программы.

№ варианта	A	B	P_1	P_2	T_1	T_2
1	6	2	0.2	0.8	3	1
2	5	1	0.25	0.75	2	1
3	7	1	0.31	0.69	4	2
4	15	3	0.43	0.57	11	3
5	8	2	0.74	0.26	2	1
6	18	3	0.84	0.16	14	4
7	27	4	0.56	0.44	20	3
8	22	2	0.34	0.66	15	4
9	12	4	0.82	0.18	6	2
10	11	2	0.23	0.77	5	1
11	9	1	0.11	0.89	4	1
12	13	3	0.36	0.64	9	3
13	10	2	0.73	0.27	7	2
14	17	3	0.93	0.07	10	3
15	15	4	0.45	0.55	11	4
16	8	2	0.24	0.76	4	1
17	16	3	0.33	0.67	12	3
18	7	2	0.14	0.86	16	5
•Ж	22	1	0.42	0.58	14	3
20	13	1	0.64	0.36	8	2
21	24	3	0.55	0.45	18	4
22	19	4	0.25	0.75	13	3
23	26	2	0.28	0.72	20	5
24	4	1	0.87	0.13	2	1
25	29	4	0.26	0.74	22	5
26	22	3	0.82	0.18	17	4
27	17	2	0.25	0.75	4	3
28	14	4	0.49	0.51	8	2
29	19	2	0.66	0.34	9	2
30	25	1	0.28	0.72	22	5
31	6	3	0.97	0.03	4	1

Контрольные вопросы

1. Что лежит в основе теории очередей
2. Что собой представляет имитационное моделирование
3. Что такое модель объекта и модель процесса
4. то собой представляет язык имитационного моделирования GPSS

5. Как осуществляется построение программы на языке GPSS
6. Как осуществляется исследование объекта с помощью полученной модели
7. В чем заключается анализ результатов исследования
8. Что такое критерий оптимизации
9. В чем состоит конечная цель процесса моделирования

Отчет по работе

Отчет должен содержать:

4. Задание и исходные данные для выполнения работы.
5. Блок-диаграмму и GPSS-программу имитационного эксперимента с моделью исследуемого варианта системы массового обслуживания.
6. Результаты обработки экспериментальных данных, анализ полученных результатов и выводы по работе.

Моделирование процесса многофазной обработки заявок с помощью языка GPSS

Цель работы

1. Освоение методов моделей сложных объектов с фазовой структурой
2. Более глубокое освоение языка имитационного моделирования GPSS
3. Отработка методики исследования объекта моделирования
4. Поиск оптимального решения

ОБЩИЕ СВЕДЕНИЯ

Моделирование Q -схем с фазовой структурой

Если приборы массового обслуживания и их параллельные композиции соединены последовательно, то имеет место многофазное обслуживание (многофазная Q -схема). Таким образом, для задания Q -схемы необходимо использовать оператор сопряжения R , отражающий взаимосвязь элементов структуры (каналов и накопителей) между собой. Связи между элементами Q -схемы изображают в виде стрелок (линий потока, отражающих направление движения заявок). Различают разомкнутые и замкнутые Q -схемы. В разомкнутой Q -схеме выходной поток обслуженных заявок не может снова поступить на какой-либо элемент, т. е. обратная связь отсутствует, а в замкнутых Q -схемах имеются обратные связи, по которым заявки двигаются в направлении, обратном движению вход-выход.

Собственными (внутренними) параметрами Q -схемы будут являться количество фаз, количество каналов в каждой фазе, количество накопителей каждой фазы, емкость i -го накопителя. Следует отметить, что в теории массового обслуживания в зависимости от емкости накопителя применяют следующую терминологию для систем массового обслуживания: системы с потерями, т. е. накопитель в приборе отсутствует, а имеется только канал обслуживания системы с ожиданием, (т. е. накопитель имеет бесконечную емкость и очередь заявок не ограничивается) и системы смешанного типа (с ограниченной емкостью накопителя). Всю совокупность собственных параметров Q -схемы обозначим как подмножество H .

Для задания Q -схемы также необходимо описать алгоритмы ее функционирования, которые определяют набор правил поведения заявок в системе в различных неоднозначных ситуациях. В зависимости от места возникновения таких ситуаций различают алгоритмы (дисциплины) ожидания заявок в накопителе H , и обслуживания заявок каналом K_i каждого элементарного обслуживающего прибора Π_i Q -схемы. Неоднородность заявок, отражающая процесс в той или иной реальной системе, учитывается с помощью введения классов приоритетов.

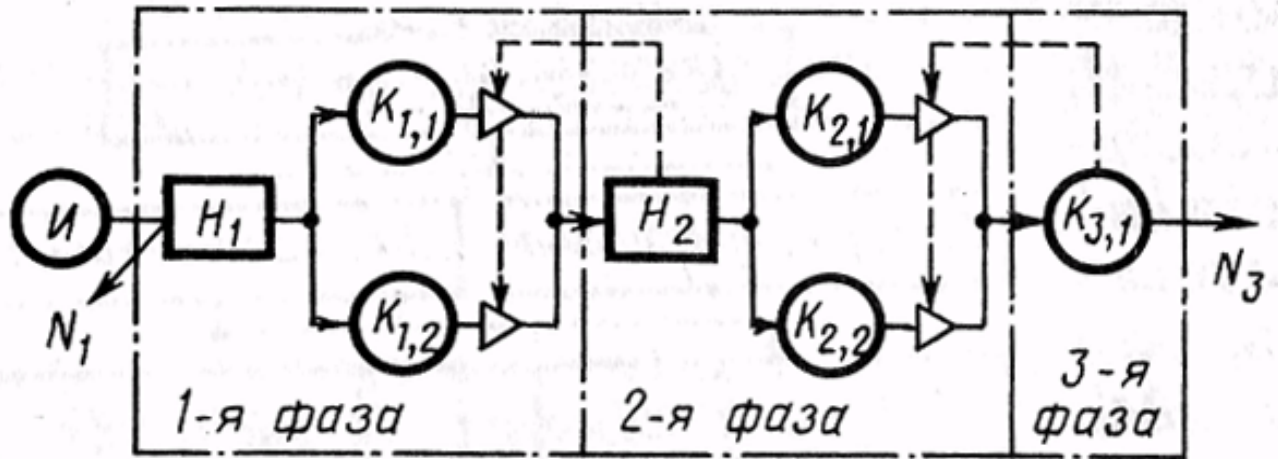
В зависимости от динамики приоритетов в Q -схемах различают статические и динамические приоритеты. Статические приоритеты назначаются заранее и не зависят от состояний Q -схемы, т. е. они являются фиксированными в пределах решения конкретной задачи моделирования. Динамические приоритеты возникают при моделировании в зависимости от возникающих ситуаций. Исходя из правил выбора заявок из накопителя H_i на обслуживание каналом K_h можно выделить относительные и абсолютные

приоритеты. *Относительный приоритет* означает, что заявка с более высоким приоритетом, поступившая в накопитель H_i ожидает окончания обслуживания предшествующей заявки каналом K_i и только после этого занимает канал. *Абсолютный приоритет* означает, что заявка с более высоким приоритетом, поступившая в накопитель H_i прерывает обслуживание каналом K_i заявки с более низким приоритетом и сама занимает канал (при этом вытесненная из K_i заявка может либо покинуть систему, либо может быть снова записана на какое-то место в H_i).

При рассмотрении алгоритмов функционирования приборов обслуживания Π_i (каналов K_i и накопителей H_i) необходимо также задать набор правил, по которым заявки покидают H_i и K_i для H_i — либо правила переполнения, по которым заявки в зависимости от заполнения H_i покидают систему, либо правила ухода, связанные с истечением времени ожидания заявки в H_i для K_i — правила выбора маршрутов или направлений ухода. Кроме того, для заявок необходимо задать правила, по которым они остаются в канале K_i или не допускаются до обслуживания каналом K_h , т. е. правила блокировок канала. При этом различают блокировки K_i по выходу и по входу. Такие блокировки отражают наличие управляющих связей в Q -схеме, регулирующих поток заявок в зависимости от состояний Q -схемы. Весь набор возможных алгоритмов поведения заявок в Q -схеме можно представить в виде некоторого оператора алгоритмов поведения заявок.

Таким образом, Q -схема, описывающая процесс функционирования системы массового обслуживания любой сложности, однозначно задается в виде $Q = (W, U, H, Z, R, A)$. При ряде упрощающих предположений относительно подмножеств входящих потоков W , потоков обслуживания U (выполнение условий стационарности, ординарности и ограниченного последствия) оператора сопряжения элементов структуры R (однофазное одноканальное обслуживание в разомкнутой системе), подмножества собственных параметров H (обслуживание с бесконечной емкостью накопителя), оператора алгоритмов обслуживания заявок A (бесприоритетное обслуживание без прерываний и блокировок) для оценки вероятностно-временных характеристик можно использовать аналитический аппарат, разработанный в теории массового обслуживания.

Математическое обеспечение и ресурсные возможности современных ЭВМ позволяют достаточно эффективно провести моделирование различных систем, формализуемых в виде Q -схем, используя либо пакеты прикладных программ, созданные на базе алгоритмических языков общего назначения, либо специализированные языки имитационного моделирования.



Пример Q-схемы общего вида

На рисунке представлена трехфазная Q-схема ($L=3$) с блокировкой каналов по выходу в 1-й и 2-й фазах обслуживания (пунктирные линии на рисунке). В качестве выходящих потоков такой Q-схемы могут быть рассмотрены поток потерянных заявок из H_j и поток обслуженных заявок из $K_{3,1}$ (N_3 на рисунке).

Для имитационной модели рассматриваемой Q-схемы можно записать следующие переменные и уравнения: эндогенная переменная P — вероятность потери заявок; экзогенные переменные: t_m — время появления очередной заявки из N ; $t_{k,j}$ — время окончания обслуживания каналом $K_{k,j}$ очередной заявки, $k=1, 2, 3$; $j=1, 2$; вспомогательные переменные: z_i и $Z_{k,j}$ — состояния H ; параметры: L — емкость, L^* — число каналов в i -й фазе.

При имитации процесса функционирования Q-схемы на ЭВМ, требуется организовать массив состояний. В этом массиве должны быть выделены: подмассив K для запоминания текущих значений $Z_{k,j}$, соответствующих каналов K_j^* и времени окончания обслуживания очередной заявки, подмассив H для записи текущего значения z , соответствующих накопителей H_i , $i=1, 2$; подмассив H , в который записывается время поступления очередной заявки из источника (H).

Процедура моделирования процесса обслуживания каждым элементарным каналом сводится к следующему. Путем обращения к генератору случайных чисел с законом распределения, соответствующим обслуживанию данных, получается длительность времени обслуживания и вычисляется время окончания обслуживания, а затем фиксируется состояние $Z_{k,j}$, при освобождении $Z_{k,j}=0$; в случае блокировки K_{kj} записывается $Z_{k,j}=2$. При поступлении заявки в H , к его содержимому добавляется единица, т. е. $z_i = z_i + 1$, а при уходе заявки из H , на обслуживание вычитается единица, т. е. $z_i = z_i - 1$, $i=1, 2$.

Возможности модификации моделирующих алгоритмов Q-схемы. В плане усложнения машинных моделей при исследовании вариантов системы S можно рассмотреть следующие модификации: наличие потоков заявок нескольких типов. В этом случае необходимо иметь несколько источников (генераторов) заявок и фиксировать признак принадлежности заявки к тому или иному потоку тогда, когда накопители и каналы рассматриваемой Q-схемы критичны к этому признаку или требуется определить характеристики обслуживания заявок каждого из потоков в отдельности.

Наличие приоритетов при постановке заявок в очередь в накопитель. В зависимости от класса приоритета заявок может быть рассмотрен случай, когда заявки одного класса имеют приоритет по записи в накопитель (при отсутствии свободных мест вытесняют из накопителя заявки с более низким классом приоритета, которые при этом считаются потерянными). Этот фактор может быть учтен в моделирующем алгоритме соответствующей *Q-схемы* путем фиксации для каждого накопителя признаков заявок, которые в нем находятся (путем организации соответствующего массива признаков).

1. Наличие приоритетов при выборе заявок на обслуживание каналов. По отношению к каналу могут быть рассмотрены заявки с абсолютным и относительным приоритетами. Заявки с абсолютным приоритетом при выборе из очереди в накопитель вытесняют из канала заявки с более низким классом приоритета, которые при этом снова поступают в накопитель (в начало или конец очереди) или считаются потерянными, а заявки с относительным приоритетом дожидаются окончания обслуживания каналом предыдущей заявки. Эти особенности учитываются в моделирующих алгоритмах приоритетных *Q-схем*, при определении времени освобождения канала и выборе претендентов на его занятие. Если наличие абсолютных приоритетов приводит к потере заявок, то необходимо организовать фиксацию потерянных заявок.

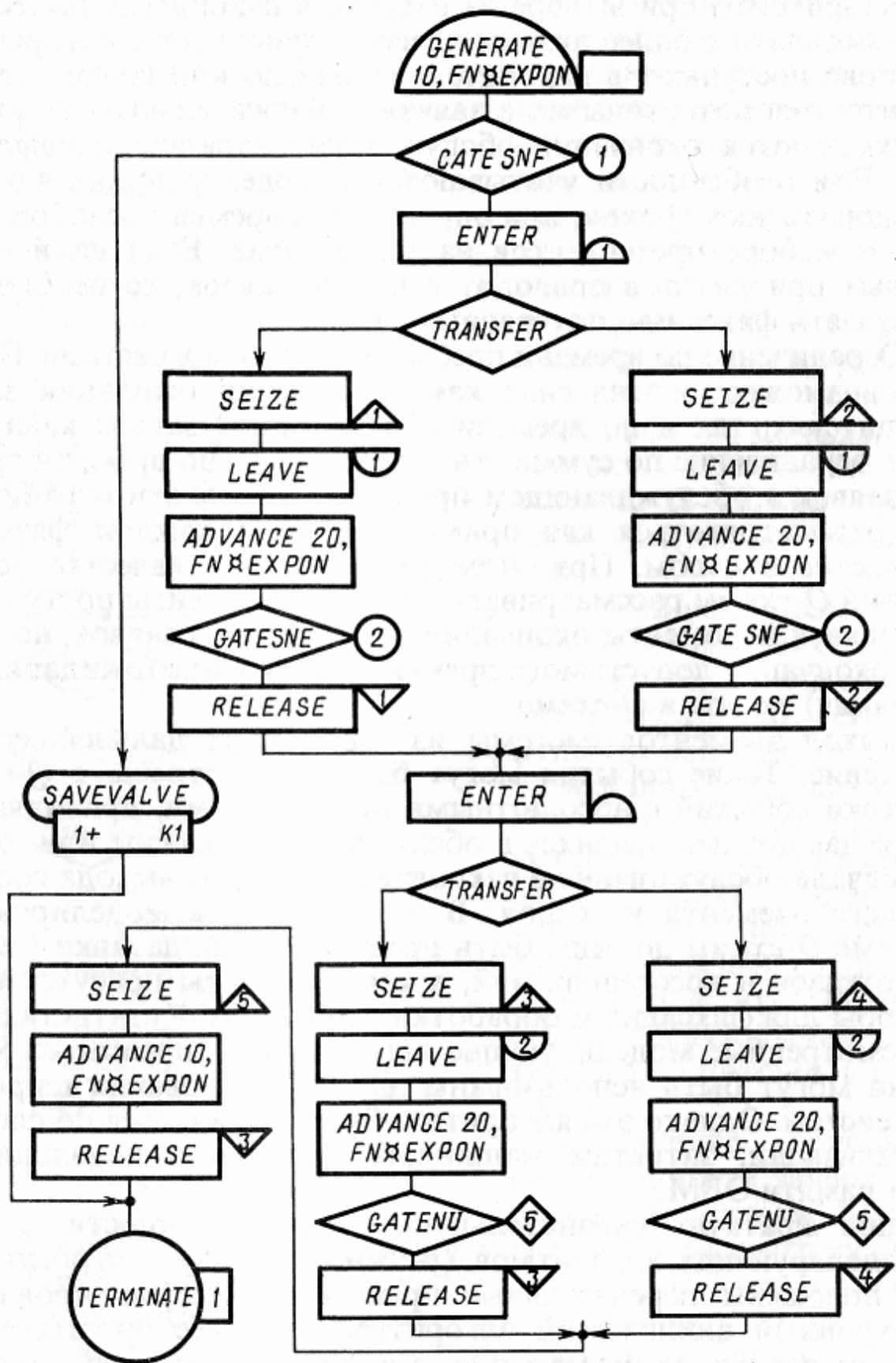


Рис. 8.14. Блок-диаграмма моделирующего алгоритма в символическом языке GPSS

2. Ограничение по времени пребывания заявок в системе. В этом случае возможно ограничение как по времени ожидания заявок в накопителях, так и по времени обслуживания заявок каналами, а также ограничение по сумме этих времен, т. е. по времени пребывания заявок в обслуживающем приборе. Причем эти ограничения могут рассматриваться как применительно к каждой фазе, так и к *Q-схеме* в целом. При этом необходимо в качестве особых состояний *Q-схемы* рассматривать не только моменты поступления новых заявок и моменты окончания обслуживания заявок, но и моменты окончания допустимого времени пребывания (ожидания, обслуживания) заявок в *Q-схеме*.

3. Выход элементов системы из строя и их дальнейшее восстановление. Такие события могут быть рассмотрены в *Q-схеме*, как потоки событий с абсолютными приоритетами, приводящими к потере заявок, находящихся в обслуживании в канале или ожидающих начала обслуживания в накопителе в момент выхода соответствующего элемента из строя. В этом случае в моделирующем алгоритме *Q-схемы* должны быть предусмотрены датчики (генераторы) отказов и восстановлений, а также должны присутствовать операторы для фиксации и обработки необходимой статистики.

Рассмотренные моделирующие алгоритмы и способы их модификации могут быть использованы для моделирования широкого класса систем. Однако эти алгоритмы будут отличаться по сложности реализации, затратам машинного времени и необходимого объема памяти ЭВМ.

Детерминированный и асинхронный циклический алгоритмы наиболее просты с точки зрения логики их построения, так как при этом используется перебор всех элементов *Q-схемы* на каждом шаге. Трудности возникают с машинной реализацией этих алгоритмов вследствие увеличения затрат машинного времени на моделирование, так как просматриваются все состояния элементов *Q-схемы*. Затраты машинного времени на моделирование существенно увеличиваются при построении детерминированных моделирующих алгоритмов *Q-схем*, элементы которых функционируют в различных масштабах времени, например когда длительности обслуживания заявок каналами многоканальной *Q-схемы* значительно отличаются друг от друга.

В стохастическом синхронном алгоритме рассматриваются прошлые изменения состояний элементов *Q-схемы*, которые произошли с момента предыдущего просмотра состояний, что несколько усложняет логику этих алгоритмов.

Асинхронный спорадический алгоритм позволяет просматривать при моделировании только те элементы *Q-схемы*, изменения состояний которых могли иметь место на данном интервале системного времени, что приводит к некоторому упрощению этих моделирующих алгоритмов по сравнению с синхронными алгоритмами и существенному уменьшению затрат машинного времени по сравнению с детерминированными и циклическими алгоритмами.

Затраты необходимой оперативной памяти ЭВМ на проведение имитации могут быть значительно уменьшены при построении блочных моделей, когда отдельные блоки (модули) *Q-схемы* реализуются в виде процедур (подпрограмм).

К настоящему времени накоплен значительный опыт моделирования *Q-схем* (при их классическом рассмотрении или в различных приложениях). Рассмотренные моделирующие алгоритмы позволяют практически отразить всевозможные варианты многофазных и многоканальных *Q-схем*, а также провести исследование всего спектра их вероятностно-временных характеристик, различных выходных характеристик, интересующих исследователя или разработчика системы 5.


```

SIMULATE      Программа имитации многофазной Q-схемы
1 STORAGE    10
2 STORAGE    10
EXPON FUNCTION RN1.C24
0 0 .1 .104 .2 .222 .3 .355 .4 .509 .5 .69
6 .915 .7 .12 .75 1.38 .8 1.6 .84 .83 .88 2.12
.9 2.3 .92 2.52 .94 2.81 .95 2.99 .96 3.2 .97 3.5
.98 3.9 .99 4.6 .995 5.3 .998 6.2 .999 7.0 .9997 8.0
GENERATE      10.FN#EXPON
3ATE SNF      1.OTK
ENTER         1
TRANSFER      BOTH.KAN11.KAN12
KAN 11 SEIZE  1
LEAVE         1
ADVANCE       20. FN#EXPON
GATE SNF      2
RELEASE       1
TRANSFER      .NAK2
KAN12 SEIZE   2
LEAVE         1
ADVANCE       20.FN#EXPON
GATE SNF      2
RELEASE       2
NAK2 ENTER   2
TRANSFER      BOTH.KAN21.KAN22
KAN21 SEIZE   3
LEAVE         2
ADVANCE       20. FN#EXPON
GATE NU       5
RELEASE      3
TRANSFER      .KAN31
KAN22 SEIZE   4
LEAVE         2
ADVANCE       20. FN#EXPON
GATE NU       5
RELEASE       4
KAN31 SEIZE   5
ADVANCE       10.FN#EXPON
RELEASE       5
TRANSFER      .END
OTK SAVEVALVE 1+.K1
END TERMINATE 1

```

Программа реализации моделирующего алгоритма на языке GPSS

Рассмотрим особенности моделирования систем, формализуемых в виде *Q-схем*, с использованием языка имитационного моделирования *GPSS*. В этом случае отпадает необходимость выбора принципа построения моделирующего алгоритма, так как механизм системного времени и просмотра состояний уже заложен в систему имитации дискретных систем, т. е. в язык *GPSS*.

№ карты	№ блока	Назначение оператора (карты)
1	—	Сообщает, что после ассемблирования необходимо начать счет по программе
2	—	Задаёт ёмкость накопителя 1
3	—	Задаёт ёмкость накопителя 2
4 — 8	—	Описывают функцию экспоненциального распределения с именем EXPON, номером FN1 и значениями в интервале (0, 1)
9	1	Генерирует транзакты с интервалами, распределёнными по экспоненциальному закону и средним значением 10 условных единиц
10	2	Проверяет, есть ли свободные места в накопителе 1
11	3	Занимает одно место в накопителе 1
12	4	Направляет транзакт в один из свободных каналов 1 и 2
13	5	Занимает канал 1, 1
14	6	Освобождает одно место в накопителе 1
15	7	Задерживает транзакт на случайный интервал времени в соответствии с экспоненциальным законом со средним значением 20
16	8	Блокирует продвижение транзактов при занятости накопителя 2
17	9	Освобождает канал 1, 1
18	10	Направляет транзакты к блоку с меткой NAK2
19 — 23	11 — 15	Выполняют функции, аналогичные блокам 5 — 9 по отношению к каналу 1, 2
24	16	Занимает одно место в накопителе 2
25	17	Направляет транзакт в один из свободных каналов 2,1 и 2,2
26 — 30	18 — 22	Выполняет функции, аналогичные блокам 5 — 9 по отношению к каналу 2,1
31	23	Направляет транзакты к блоку с меткой KAN31
32 — 36	24 — 28	Выполняет функции, аналогичные блокам 5 — 9 по отношению к каналу 2,2
37	29	Занимает канал 3,1
38	30	Задерживает транзакт на случайный интервал времени в соответствии с экспоненциальным законом со средним значением 10 условных единиц
39	31	Освобождает канал 3,1
40	32	Направляет транзакты к блоку с меткой END
41	33	Подсчитывает число транзактов, получивших отказ в обслуживании
42	34	Уничтожает транзакт
43	—	Означает конец набора входных карт модели и устанавливает начальное значение счётчика числа завершений, равное 1000
44	—	Передаёт управление операционной системе

Выполнение лабораторной работы

1. С помощью языка имитационного моделирования GPSS составить трехфазную модель обработки заявок с параллельным дублированием согласно схемы 1 и 2.
2. Произвести процесс исследования модели путем варьирования основных параметров модели в диапазоне -15% - +15% от заданных значений.
3. Определить оптимальный режим функционирования из расчета максимального выполнения заявок и минимальных затрат времени.
4. Исходные данные взять из лабораторной работы №1

Контрольные вопросы

1. Какие *Q*-схемы называются многофазными
2. Как классифицируются многофазные *Q*-схемы
3. Какова технология имитации процесса функционирования *Q*-схемы
4. Каковы возможности модификации моделирующих алгоритмов *Q*-схем

Лабораторная работа №3

Планирование и проведение машинного эксперимента Многофазной Q-схемы

Цель работы

1. Изучение методов планирования эксперимента
2. Составление многофакторного плана второго порядка
3. Проведение машинного эксперимента

ОБЩИЕ СВЕДЕНИЯ

ПЛАНИРОВАНИЕ МАШИННЫХ ЭКСПЕРИМЕНТОВ С МОДЕЛЯМИ СИСТЕМ

Имитационное моделирование является по своей сути машинным экспериментом с моделью исследуемой или проектируемой системы. План имитационного эксперимента на ЭВМ представляет собой метод получения с помощью эксперимента необходимой пользователю информации. Эффективность использования экспериментальных ресурсов существенным образом зависит от выбора плана эксперимента. Основная цель экспериментальных исследований с помощью имитационных моделей состоит в наиболее глубоком изучении поведения моделируемой системы. Для этого необходимо планировать и проектировать не только саму модель, но и процесс ее использования, т. е. проведение с ней экспериментов на ЭВМ.

МЕТОДЫ ТЕОРИИ ПЛАНИРОВАНИЯ ЭКСПЕРИМЕНТОВ

Машинный эксперимент с моделью системы S при ее исследовании и проектировании проводится с целью получения информации о характеристиках процесса функционирования рассматриваемого объекта. Эта информация может быть получена как для анализа характеристик, так и для их оптимизации при заданных ограничениях, т. е. для синтеза структуры, алгоритмов и параметров системы S . В зависимости от поставленных целей моделирования системы S на ЭВМ имеются различные подходы к организации имитационного эксперимента с машинной моделью M_M . Основная задача планирования машинных экспериментов — получение необходимой информации об исследуемой системе S при ограничениях на ресурсы (затраты машинного времени, памяти и т. п.). К числу частных задач, решаемых при планировании машинных экспериментов, относятся задачи уменьшения затрат машинного времени на моделирование, увеличения точности и достоверности результатов моделирования, проверки адекватности модели и т. д.

Машинный эксперимент. Эффективность машинных экспериментов с моделями M_M существенно зависит от выбора плана эксперимента, так как именно план определяет объем и порядок проведения вычислений на ЭВМ, приемы накопления и статистической обработки результатов моделирования системы S . Поэтому основная задача планирования машинных экспериментов с моделью M_M формулируется следующим образом: необходимо получить информацию об объекте моделирования, заданном в

виде моделирующего алгоритма (программы), при минимальных или ограниченных затратах машинных ресурсов на реализацию процесса моделирования.

Таким образом, при машинном моделировании рационально планировать и проектировать не только саму модель M_M системы S , но и процесс ее использования, т. е. проведение с ней экспериментов с использованием инструментальной ЭВМ.

Для планирования эксперимента наиболее важное значение имеет следующее:

- 1) простота повторения условий эксперимента на ЭВМ с моделью M_M системы S ;
- 2) возможность управления экспериментом с моделью M_M , включая его прерывание и возобновление;
- 3) легкость варьирования условий проведения эксперимента (воздействию внешней среды E);
- 4) наличие корреляции между последовательностью точек в процессе моделирования;
- 5) трудности, связанные с определением интервала моделирования.

Преимуществом машинных экспериментов является возможность полного воспроизведения условий эксперимента с моделью исследуемой системы S . Сравнить две альтернативы возможно при одинаковых условиях, что достигается, например, выбором одной и той же последовательности случайных чисел для каждой из альтернатив. Существенным достоинством является простота прерывания и возобновления машинных экспериментов, что позволяет применять последовательные и эвристические приемы планирования, которые могут оказаться нереализуемыми в экспериментах с реальными объектами. При работе с машинной моделью M_M всегда возможно прерывание эксперимента на время, необходимое для анализа результатов и принятия решений об его дальнейшем ходе (например, о необходимости изменения значений параметров модели M_M).

Недостатком машинных экспериментов является то, что часто возникают трудности, связанные с наличием корреляции в выходных последовательностях, т. е. результаты одних наблюдений зависят от результатов одного или нескольких предыдущих, и поэтому в них содержится меньше информации, чем в независимых наблюдениях. Так как в большинстве существующих методов планирования экспериментов предполагается независимость наблюдений, то многие из этих методов нельзя непосредственно применять для машинных экспериментов при наличии корреляции.

Основные понятия планирования экспериментов. В связи с тем что математические методы планирования экспериментов основаны на кибернетическом представлении процесса проведения эксперимента, наиболее подходящей моделью последнего является абстрактная схема, называемая «черным ящиком». При таком кибернетическом подходе различают входные и выходные переменные: $x_1, x_2, \dots, x_k; y_1, y_2, \dots, y_i$. В зависимости от того, какую роль играет каждая переменная в проводимом эксперименте, она может являться либо фактором, либо реакцией. Пусть, например, имеют место только две переменные: x и y . Тогда если цель эксперимента — изучение влияния переменной x на переменную y , то x — фактор, а y — реакция. В экспериментах с машинными моделями M_M системы S фактор является экзогенной или управляемой (входной) переменной, а реакция — эндогенной (выходной) переменной.

Каждый фактор x_i , $i=1, 2, \dots, k$, может принимать в эксперименте одно из нескольких значений, называемых *уровнями*. Фиксированный набор уровней факторов определяет одно из возможных состояний рассматриваемой системы. Одновременно этот набор представляет собой условия проведения одного из возможных экспериментов.

Каждому фиксированному набору уровней факторов соответствует определенная точка в многомерном пространстве, называемом *факторным пространством*. Эксперименты не могут быть реализованы во всех точках факторного пространства, а лишь в принадлежащих допустимой области, как, например, это показано для случая двух факторов x_1 и x_2 на рисунке (плоскость $x_1, 0, x_2$).

Существует вполне определенная связь между уровнями факторов и реакцией (откликом) системы, которую можно представить в виде соотношения

$$y_i = \Psi(x_1, x_2, \dots, x_k), i = 1, 2, \dots, m$$

Функцию Ψ_i связывающую реакцию с факторами, называют *функцией реакции*, а геометрический образ, соответствующий функции реакции, — *поверхностью реакции*. Исследователю заранее не известен вид зависимостей $\Psi_i, i = 1, 2, \dots, m$, поэтому используют приближенные соотношения:

$$y_i = \varphi(x_1, x_2, \dots, x_k), i = 1, 2, \dots, m$$

Зависимости φ_i находятся по данным эксперимента. Последний необходимо поставить так, чтобы при минимальных затратах ресурсов (например, минимальном числе испытаний), варьируя по специально сформулированным правилам значения входных переменных, построить математическую модель системы и оценить ее характеристики.

При планировании экспериментов необходимо определить основные свойства факторов. Факторы при проведении экспериментов могут быть управляемыми и неуправляемыми, наблюдаемыми и ненаблюдаемыми, изучаемыми и не изучаемыми, количественными и качественными, фиксированными и случайными.

Фактор называется *управляемым*, если его уровни целенаправленно выбираются исследователем в процессе эксперимента. При машинной реализации модели M_M исследователь принимает решения, управляя изменением в допустимых пределах различных факторов.

Фактор называется *наблюдаемым*, если его значения наблюдаются и регистрируются. Обычно в машинном эксперименте с моделью M_M наблюдаемые факторы совпадают с управляемыми, так как нерационально управлять фактором, не наблюдая его. Но неуправляемый фактор также можно наблюдать. Например, на этапе проектирования конкретной системы S нельзя управлять заданными воздействиями внешней среды E , но можно наблюдать их в машинном эксперименте. *Наблюдаемые неуправляемые факторы* получили название *сопутствующих*. Обычно при машинном эксперименте с моделью M_M число сопутствующих факторов велико, поэтому рационально учитывать влияние лишь тех из них, которые наиболее существенно воздействуют на интересующую исследователя реакцию.

Фактор относится к *изучаемым*, если он включен в модель M_M для изучения свойств системы S , а не для вспомогательных целей, например для увеличения точности эксперимента.

Фактор будет *количественным*, если его значения — числовые величины, влияющие на реакцию, а в противном случае *фактор* называется *качественным*. Например, в модели системы, формализуемой в виде схемы массового обслуживания (*Q-схемы*), количественными факторами являются интенсивности входящих потоков заявок, интенсивности потоков обслуживания, емкости накопителей, количество обслуживающих каналов и т. д., а качественными факторами — дисциплины постановки в очередь, выбора из очереди, обслуживания заявок каналами и т. д. Качественным факторам в отличие от количественных не соответствует числовая шкала. Однако и для них можно построить условную порядковую шкалу, с помощью которой производится кодирование,

устанавливая соответствие между условиями качественного фактора и числами натурального ряда.

Фактор называется *фиксированным*, если в эксперименте исследуются все интересующие экспериментатора значения фактора, а если экспериментатор исследует только некоторую случайную выборку из совокупности интересующих значений факторов, то *фактор* называется *случайным*. На основании случайных факторов могут быть сделаны вероятностные выводы и о тех значениях факторов, которые в эксперименте не исследовались.

В машинных экспериментах с моделями M_M не бывает неуправляемых или ненаблюдаемых факторов применительно к исследуемой системе S . В качестве воздействий внешней среды E , т. е. неуправляемых и ненаблюдаемых факторов, в машинной имитационной модели выступают стохастические экзогенные переменные. Если имитационная модель сформулирована, то все факторы определены и нельзя во время проведения данного эксперимента (испытания) с моделью M_M вводить дополнительные факторы.

Каждый фактор может принимать в испытании одно или несколько значений, называемых *уровнями*, причем фактор будет управляемым, если его уровни целенаправленно выбираются экспериментатором. Для полного определения фактора необходимо указать последовательность операций, с помощью которых устанавливаются его конкретные уровни. Такое определение фактора называется *операциональным* и обеспечивает однозначность понимания фактора.

Основными требованиями, предъявляемыми к факторам, являются требование управляемости фактора и требование непосредственного воздействия на объект. Под управляемостью фактора понимается возможность установки и поддержания выбранного нужного уровня фактора постоянным в течение всего испытания или изменяющимся в соответствии с заданной программой. Требование непосредственного воздействия на объект имеет большое значение в связи с тем, что трудно управлять фактором, если он является функцией других факторов.

При планировании эксперимента обычно одновременно изменяются несколько факторов. Определим требования, которые предъявляются к совокупности факторов. Основные из них — совместимость и независимость. Совместимость факторов означает, что все их комбинации осуществимы, а независимость соответствует возможности установления фактора на любом уровне независимо от уровней других.

При проведении машинного эксперимента с моделью M_M для оценки некоторых характеристик процесса функционирования исследуемой системы S экспериментатор стремится создать такие условия, которые способствуют выявлению влияния факторов, находящихся в функциональной связи с искомой характеристикой.

Для этого необходимо: отобрать факторы $x_i, i = 1, 2, \dots, k$, влияющие на искомую характеристику, и описать функциональную зависимость; установить диапазон изменения факторов $x_{i \min} \div x_{i \max}$; определить координаты точек факторного пространства $\{x_1, x_2, \dots, x_k\}$, в которых следует проводить эксперимент; оценить необходимое число реализаций и их порядок в эксперименте.

Свойства объекта исследования, т. е. процесса машинного моделирования системы S , можно описывать с помощью различных методов (моделей планирования). Для выбора конкретной модели необходимо сформулировать такие ее особенности, как адекватность, содержательность, простота и т. д. Под содержательностью модели планирования понимается ее способность объяснять множество уже известных фактов, выявлять новые и предсказывать их дальнейшее развитие. Простота — одно из главных

достоинств модели планирования, выражающееся в реализуемости эксперимента на ЭВМ, но при этом имеет место противоречие с требованиями адекватности и содержательности.



Для экстремального планирования экспериментов наибольшее применение нашли модели в виде алгебраических полиномов. Предполагаем, что изучается влияние k количественных факторов $x_i, i = \overline{1, k}$, на некоторую реакцию η в отведенной для экспериментирования локальной области факторного пространства G , ограниченной $x_{i \min} \div x_{i \max}, i = \overline{1, k}$ (рисунок для случая $k=2$). Допустим, что функцию реакции $\Psi(x_1, x_2, \dots, x_k)$ можно с некоторой степенью точности представить в виде полинома степени d от k переменных

$$\eta = \tilde{b}_0 + \sum_{1 \leq i \leq k} \tilde{b}_i x_i + \sum_{1 \leq i < j \leq k} \tilde{b}_{ij} x_i x_j + \dots + \sum_{i_1, i_2, \dots, i_k} \tilde{b}_{i_1, i_2, \dots, i_k} x_1^{i_1} x_2^{i_2} \dots x_k^{i_k}, \sum i_j = d,$$

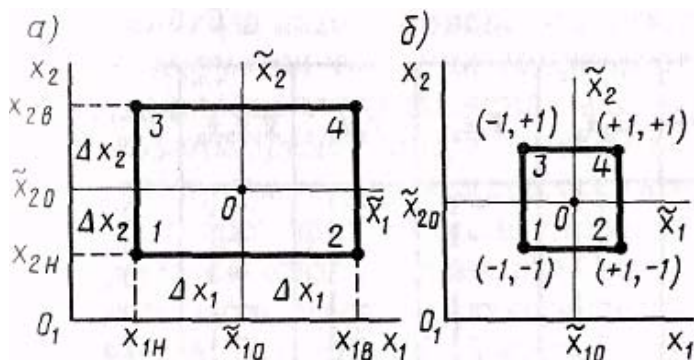
который содержит C_{k+d}^d коэффициентов.

В рамках выбранной модели планирования в виде алгебраических полиномов строится план эксперимента путем варьирования каждого из факторов $x_i, i = \overline{1, k}$ на нескольких уровнях q относительно исходной точки x_{i_0} , представляющей центр эксперимента.

Виды планов экспериментов. Эксперимент, в котором реализуются все возможные сочетания уровней факторов, называется *полным факторным экспериментом* (ПФЭ). Если выбранная модель планирования включает в себя только линейные члены полинома и их произведения, то для оценки коэффициентов модели используется план

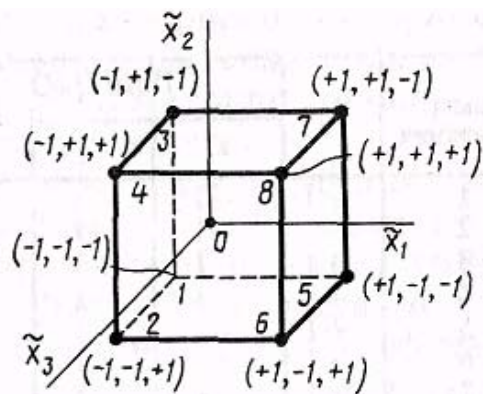
эксперимента с варьированием всех k факторов на двух уровнях, т. е. $\delta=2$. Такие планы называются планами типа 2^k , где $N=2^k$ — число всех возможных испытаний.

Начальный этап планирования эксперимента для получения коэффициентов линейной модели основан на варьировании факторов на двух уровнях: нижнем x_{i_n} и верхнем x_{i_v} — симметрично расположенных относительно основного уровня \bar{x}_i , $i = \overline{1, k}$. Геометрическая интерпретация показана на рисунке *a*.



**Геометрическая интерпретация
полного факторного эксперимента
типа 2^2 :**

a — без масштабирования; *b* — при масштабировании по осям



**Геометрическая интерпретация
полного факторного
эксперимента типа 2^3**

Так как каждый фактор принимает лишь два значения то для стандартизации и упрощения записи условий каждого испытания и обработки выборочных данных эксперимента масштабы по осям факторов выбираются так, чтобы нижний уровень соответствовал -1, верхний +1, а основной — нулю. Это легко достигается с помощью преобразования вида

$$\tilde{X}_i = (x_i - x_{i_0}) / \Delta x_i, i = 1, 2, \dots, k, \text{ где } \Delta x_i = (x_{i_{\min}} + x_{i_{\max}}) / 2,$$

где \tilde{X}_i — кодированное значение i -го фактора; x_i — натуральное значение фактора; x_{i_0} — нулевой уровень; $\Delta x_i = (x_{i_v} - x_{i_n}) / 2$ — интервал варьирования фактора.

Расположение точек для ПФЭ типа 2^2 показано на рисунке *b*. Выписывая комбинации уровней факторов для каждой экспериментальной точки квадрата, получим план D полного факторного эксперимента типа 2^2 .

При этом планы можно записывать сокращенно с помощью условных буквенных обозначений строк. Для этого порядковый номер фактора ставится в соответствие строчной букве латинского алфавита: $x_1 - a, x_2 - b$ и т. д.

Таблица 3.1

Номер испытания	1	2	3	4
X_1	-1	+1	-1	+1
X_2	-1	-1	+1	+1
Обозначения строк	(1)	a	b	ab

Затем для каждой строки плана выписываются латинские буквы только для факторов, находящихся на верхних уровнях; испытание со всеми факторами на нижних уровнях обозначается как (1). Запись плана в буквенных обозначениях показана в последней строчке.

Геометрическая интерпретация ПФЭ 2^3 приведена на рисунке, а его план ниже:

Таблица 3.2

Номер испытания	1	2	3	4	5	6	7	8
X_1	-1	-1	-1	-1	+1	+1	+1	+1
X_2	-1	-1	+1	+1	-1	-1	+1	+1
X_3	-1	+1	-1	+1	-1	+1	-1	+1
Обозначения строк	(1)	c	b	bc	a	ac	ab	abc

Полный факторный эксперимент дает возможность определить не только коэффициенты регрессии, соответствующие линейным эффектам, но и коэффициенты регрессии, соответствующие всем эффектам взаимодействия. Эффект взаимодействия двух (или более) факторов появляется при одновременном варьировании этих факторов, когда действие каждого из них на выход зависит от уровня, на которых находятся другие факторы.

Для оценки свободного члена b_0 и определения эффектов взаимодействия $b_{12}, b_{13}, \dots, b_{123}, \dots$ план эксперимента D расширяют до матрицы планирования X путем добавления соответствующей «фиктивной переменной»: единичного столбца x_0 и столбцов произведений, как показано, например, для ПФЭ типа 2^3 в таблице 3.2

Таблица 3.3

Номер испытания	X_0	План ПФЭ			$X_1 * X_2$	$X_1 * X_3$	$X_2 * X_3$	$X_1 * X_2 * X_3$	Реакция у
		X_1	X_2	X_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

Как видно из рассмотренных планов экспериментов типов 2^2 и 2^3 , 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.

Рассмотрим построение планов так называемого дробного факторного эксперимента. Пусть имеется простейший полный факторный эксперимент типа 2^2 . Используя матрицу планирования, приведенную в табл. 3.4, можно вычислить коэффициенты и представить результаты в виде уравнения

$$Y = b_0 + b_1x_1 + b_2x_2 + b_{12}x_1x_2$$

Таблица 3.4

Номер испытания	X_0	План ПФЭ		$(X_3)X_1 * X_2$	Реакция Y
		X_1	X_2		
1	+1	-1	-1	+1	y_1
2	+1	-1	-1	+1	y_2
3	+1	-1	+1	-1	y_3
4	+1	-1	+1	-1	y_4

Если в выбранных интервалах варьирования уровня процесс можно описать линейной моделью, то достаточно определить три коэффициента: b_0, b_1 и b_3 . Таким образом, остается одна степень свободы, которую можно использовать для минимизации числа испытаний. При линейном приближении $b_{12} = 0$ и вектор-столбец x_1x_2 (табл. 3.4) можно использовать для нового фактора x_3 . Поставим в табл. 3.4 этот фактор в скобках над взаимодействием x_1x_2 . В этом случае отдельных оценок, которые имели место в ПФЭ типа 2, уже не будет и оценки смещаются следующим образом: при постулировании линейной модели все парные взаимодействия не учитывают. Таким образом, вместо восьми испытаний в полном факторном эксперименте типа 2^3 необходимо провести только четыре. Правило проведения дробного факторного эксперимента формулируется так: для сокращения числа испытаний новому фактору присваивается значение вектор-столбца матрицы, принадлежащего взаимодействию, которым можно пренебречь.

При проведении эксперимента из четырех испытаний для оценки влияния трех факторов пользуются половиной ПФЭ типа 2^3 , так называемой «полуреplikой». Если приравнять x_3 и x_1x_2 , то можно получить вторую «полуреplikу». Для обозначения дробных реплик, в которых d линейных эффектов приравнены к эффектам взаимодействия, пользуются условным обозначением 2^{k-d} . Например, «полуреплика» от 2^6 записывается в виде 2^{6-1} , а «четвертьреплика» — 2^{6-2} .

Когда модель планирования анализируется методами дисперсионного анализа, применяют планы дисперсионного анализа. Если при постановке эксперимента реализуются все возможные совокупности условий, то говорят о полных классификациях дисперсионного анализа. Если проводится сокращение перебора вариантов — это неполная классификация дисперсионного анализа. Сокращение перебора может проводиться случайным образом (без ограничения на рандомизацию) или в соответствии с некоторыми правилами (с ограничениями на рандомизацию). Чаще всего в качестве таких планов используют блочные планы и планы типа латинского квадрата.

Выполнение лабораторной работы

1. Составление матрицы планирования полного факторного эксперимента для Четырех факторов с эффектами взаимодействия.

2. Объект исследования выбирается из лабораторной работы №2
3. Осуществить расчет имитационной модели с использованием исходных данных определенных на основании составленного плана ПФЭ
4. Результаты зафиксировать в таблице и использовать как исходные данные в лабораторной работе №4

Контрольные вопросы

1. Как определяется объем выборки ПФЭ
2. Как нормируются значения факторов ПФЭ
3. Как составляется матрица планирования ПФЭ без эффектов взаимодействия
4. Как составляется матрица планирования ПФЭ с эффектами взаимодействия
5. Как составляется дробная реплика ПФЭ
6. Как организовывается машинный эксперимент на основе ПФЭ

Лабораторная работа № 4

Обработка результатов машинного эксперимента и определение Оптимальных режимов функционирования системы

Цель работы

1. Изучение методов обработки результатов эксперимента
2. Определение модели методом планирования
3. Поиск оптимальных режимов функционирования системы

ОБЩИЕ СВЕДЕНИЯ

ОБРАБОТКА МАШИННЫХ ЭКСПЕРИМЕНТОВ С МОДЕЛЯМИ СИСТЕМ

Для экстремального планирования экспериментов наибольшее применение нашли модели в виде алгебраических полиномов. Предполагаем, что изучается влияние k количественных факторов $x_i, i = \overline{1, k}$, на некоторую реакцию η в отведенной для экспериментирования локальной области факторного пространства G , ограниченной $x_{i \min} \div x_{i \max}, i = \overline{1, k}$. Допустим, что функцию реакции $\Psi(x_1, x_2, \dots, x_k)$ можно с некоторой степенью точности представить в виде полинома степени d от k переменных

$$\eta = \tilde{b}_0 + \sum_{1 \leq i \leq k} \tilde{b}_i x_i + \sum_{1 \leq i < j \leq k} \tilde{b}_{ij} x_i x_j + \dots + \sum_{i_1, i_2, \dots, i_k} \tilde{b}_{i_1, i_2, \dots, i_k} x_1^{i_1} x_2^{i_2} \dots x_k^{i_k}, \sum i_j = d, \quad (4.1)$$

который содержит C_{k+d}^d коэффициентов.

Соотношение (4.1) может быть представлено как

$$\eta = \vec{f}(\vec{x}) \vec{B}, \quad (4.2)$$

где $\vec{f}(\vec{x})$ — вектор с элементами $f_\alpha(\vec{x}), \alpha = \overline{0, k^d}$, входящими в исходный полином;

\vec{B} — вектор коэффициентов, которые соответственно имеют такой вид:

$$\vec{f}(\vec{x}) = \|f_0(\vec{x}), f_1(\vec{x}), \dots, f_{k^d}(\vec{x})\|, k^d = C_{k+d-1}^d,$$

$$\vec{x} = \|1, \dots, 1; x_1, \dots, x_k; x_1^2, \dots, x_k^2; x_1 x_2, \dots, x_{k-1} x_k; \dots; x_1^j, \dots, x_k^j; x_1^d, \dots, x_k^d; x_1^{d-1} x_2, \dots, x_{k-(d-1)} \dots x_{k-1} x_k\|;$$

$$\vec{B} = \left\| \tilde{b}_0; \tilde{b}_1, \dots, \tilde{b}_k; \tilde{b}_{11}, \dots, \tilde{b}_{kk}; \tilde{b}_{12}, \dots, \tilde{b}_{(k-1)k}; \dots; \tilde{b}_{1 \dots 1}, \dots, \tilde{b}_{k \dots k}; \tilde{b}_{1 \dots 12}, \dots, \tilde{b}_{[k-(d-1)] \dots (k-1)k} \right\|.$$

Введем фиктивную переменную $x_0 = 1$, а также переменные

$$x_{k+1} = x_1^2, x_{k+2} = x_2^2, \dots, x_{2k+1} = x_k^2,$$

$$x_{2k+2} = x_1 x_2, \dots, x_{k'} = x_{k-(d-1)} \dots x_{k-1} x_k.$$

Тогда (4.1) запишется как однородное линейное уравнение вида

$$\eta = \bar{f}(\bar{x})\bar{B} = \sum_{\alpha=0}^k \tilde{b}_{\alpha} f_{\alpha}(\bar{x}) = \sum_{\alpha=0}^k \tilde{b}'_{\alpha} x_{\alpha}, \quad (4.3)$$

где $\tilde{b}'_0 = \tilde{b}_0; \tilde{b}'_1 = \tilde{b}_1, \dots, \tilde{b}'_k = \tilde{b}_k; \tilde{b}'_{k+1} = \tilde{b}_{11}, \dots, \tilde{b}'_{2k+1} = \tilde{b}_{kk}; \tilde{b}'_{2k+2} = \tilde{b}_{12}, \dots, \tilde{b}'_k = \tilde{b}_{[k-(d-1)] \dots (k-1)k}$.

Для оценки коэффициентов в (4.3) можно применить методы линейной регрессии.

Аппроксимация полиномов второго порядка функции реакции в однофакторной модели планирования может быть представлена в виде:

$$\eta = \tilde{b}_0 + \tilde{b}_1 x_1 + \tilde{b} x_1^2.$$

Более сложные объекты требуют применения полиномиальных моделей планирования большего порядка. Так, модель второго порядка в k -факторном эксперименте будет иметь вид:

$$\eta = \tilde{b}_0 + \sum_{i=1}^k \tilde{b}_i x_i + \sum_{i=1}^k \tilde{b}_{ij} x_i^2 + \sum_{i \neq j}^k \tilde{b}_{ij} x_i x_j.$$

На практике часто стремятся к использованию линейной модели планирования, преобразуя исходные полиномиальные модели. Например, модель второго порядка

$$\eta = \sum_{i=0}^k \tilde{b}_i x_i + \sum_{i=1}^k \sum_{j=1}^k \tilde{b}_{ij} x_i x_j$$

может быть преобразована к линейному виду путем введения фиктивных переменных $x_{ij} = x_i x_j$. Тогда в результате получается модель множественной линейной регрессии вида

$$\eta = \sum_{i=0}^k \tilde{b}'_i x_i + \sum_{i=1}^k \sum_{j=1}^k \tilde{b}'_{ij} x_{ij}.$$

Функция реакции может иметь и более сложную зависимость от факторов. В этом случае некоторые из них удастся привести к линейному виду. Такими моделями являются мультипликативная, регрессионная, экспоненциальная и др.

Если выбрана модель планирования, т. е. выбран вид функции $\eta = \varphi(x_1, x_2, \dots, x_k)$ и записано ее уравнение, то остается в отведенной для исследования области факторного пространства G спланировать и провести эксперимент для оценки числовых значений констант (коэффициентов) этого уравнения.

Так как полином (4.2) или (4.3) содержит C_{k+d}^d коэффициентов, подлежащих определению, то план эксперимента D должен содержать по крайней мере $N \geq C_{k+d}^d$ различных экспериментальных точек:

$$D = \left\| \begin{array}{c} x_{11} x_{21} \dots x_{k1} \\ x_{12} x_{22} \dots x_{k2} \\ \dots \dots \dots \\ x_{1N} x_{2N} \dots x_{kN} \end{array} \right\|,$$

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

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

$$\bar{y} = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{pmatrix}$$

где y_u — реакция, соответствующая u -й точке плана $\bar{x}_u = \|x_{1u}, x_{2u}, \dots, x_{ku}\|$, $u = \overline{1, N}$.

При незначительном влиянии неуправляемых входных переменных и параметров по сравнению с вводимыми возмущениями управляемых переменных в планировании эксперимента предполагается верной следующая модель:

$$y_u = \eta_u + c_u = \varphi_u(\bar{x}) + e_u = b_0 x_{0u} + b_1 x_{1u} + \dots + b_k x_{ku} + b_{k+1} x_{k+1u} + \dots + b_{k'} x_{k'u} + e_u, k' = C_{k+d-1}^d,$$

$$u = \overline{1, N},$$

где e_u — ошибка (шум, флуктуация) испытания, которая предполагается независимой нормально распределенной случайной величиной с математическим ожиданием $M[e_u] = 0$ и постоянной дисперсией $D[e_u] = \sigma_e^2 = const$.

Выписав аналогичные соотношения для всех точек плана $u = \overline{1, N}$ получим матрицу планирования

$$X = \begin{pmatrix} x_{01} x_{11} x_{21} \dots x_{k1} x_{k+11} \dots x_{k'1} \\ x_{02} x_{12} x_{22} \dots x_{k2} x_{k+12} \dots x_{k'2} \\ \dots \\ x_{0N} x_{1N} x_{2N} \dots x_{kN} x_{k+1N} \dots x_{k'N} \end{pmatrix}$$

размерностью $N \times (k' + 1)$.

Рассмотрим особенности планирования эксперимента для линейного приближения поверхности реакции, причем построению плана предшествует проведение ряда неформализованных действий (принятие решений), направленных на выбор локальной области факторного пространства G (рис. 4.1).

Вначале следует выбрать границы $x_{i \min}$ и $x_{i \max}$ области определения факторов, задаваемые исходя из свойств исследуемого объекта, т. е. на основе анализа априорной информации о системе S и внешней среде E . Например, такая переменная, как температура, при термобарических экспериментах принципиально не может быть ниже абсолютного нуля и выше температуры плавления материала, из которого изготовлена термобарокамера.

После определения области G необходимо найти локальную подобласть для планирования эксперимента путем выбора основного (нулевого) уровня x_{i0} и интервалов варьирования $\Delta x_i, i = \overline{1, k}$.

В качестве исходной точки x_{i0} выбирают такую, которая соответствует наилучшим условиям, определенным на основе анализа априорной информации о системе S , причем эта точка не должна лежать близко к границам области определения факторов $x_{i \min}$ и $x_{i \max}$. На выбор интервала варьирования Δx_i , накладываются естественные ограничения снизу (интервал не может быть меньше ошибки фиксирования уровня фактора, так как в противном случае верхний и нижний уровни окажутся неразличимыми) и сверху (верхний и нижний уровни не должны выходить за область определения G).

В рамках выбранной модели планирования в виде алгебраических полиномов строится план эксперимента путем варьирования каждого из факторов $x_i, i = \overline{1, k}$ на нескольких уровнях q относительно исходной точки x_{i0} , представляющей центр эксперимента.

После проведения машинного эксперимента необходимо определить коэффициенты алгебраического полинома, представляющего собой модель планирования. Для достижения этой цели можно воспользоваться методами матричной алгебры. Матрица-столбец коэффициентов полинома определяется по формуле:

$$B = (X^T X)^{-1} (X^T Y),$$

где B – матрица-столбец коэффициентов полинома; X – матрица планирования эксперимента; X^T – транспонированная матрица планирования эксперимента; Y – матрица-столбец реакции исследуемой системы.

Если принять во внимание, что матрица планирования двухуровневого плана первого порядка с эффектами взаимодействия является ортогональной, что говорит о независимости всех факторов, то коэффициенты полинома могут быть определены по формуле:

$$b_j = \sum_{i=1}^N \frac{x_{ij} y_i}{N}.$$

Полученный полином позволяет определить оптимальные области значений факторов исследуемого объекта. Для этой цели могут быть использованы как аналитические так и поисковые методы. Из аналитических методов можно воспользоваться определением частных производных по всем факторам и приравнивания их к нулю. Из решения полученной системы уравнений определяются оптимальные значения факторов.

Из поисковых методов можно воспользоваться одним из градиентных методов, позволяющих с помощью ЭВМ достаточно быстро определить область оптимальных значений факторов

Выполнение лабораторной работы

1. По результатам машинного эксперимента, проведенного в лабораторной работе №3, записать матрицу планирования первого порядка с эффектами взаимодействия и определить значения коэффициентов полинома, выбранного в качестве модели планирования
2. Определить оптимальные области значений факторов процесса функционирования системы

Контрольные вопросы

1. Как выбирается вид полинома в качестве модели планирования
2. Как составляется матрица планирования
3. Какие бывают планы проведения эксперимента.
4. Какими свойствами обладает матрица планирования
5. Как осуществляется расчет коэффициентов полинома
6. Как осуществляется поиск оптимальных значений факторов

ЯЗЫК ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ GPSS

1. ФУНКЦИОНАЛЬНАЯ СТРУКТУРА GPSS

Система GPSS ориентирована на класс объектов, процесс функционирования которых можно представить в виде множества состояний и правил перехода из одного состояния в другое, определяемых в дискретной пространственно-временной области. Примерами таких объектов являются вычислительные системы, сети ЭВМ, системы передачи сообщений, транспортные объекты, склады, магазины, предприятия и т.п. В качестве формальных моделей таких объектов используют системы массового обслуживания, автоматы, стохастические сети, сети Петри и макросети, агрегаты и т.п.

В состав GPSS входят следующие типы объектов: транзакты, блоки, списки, устройства, памяти, логические ключи, очереди, таблицы, ячейки, функции, переменные. Любую модель на языке GPSS можно представить в виде комбинации компонентов, взятых из числа названных объектов. Модель имеет три уровня представления:

- верхний уровень, определяемый комбинацией основных функциональных объектов: устройств, памяти, ключей, очередей;
- средний уровень, представляемый схемой из типовых блоков, между которыми перемещаются транзакты;
- нижний уровень - уровень физической реализации языка GPSS в виде программ и наборов данных, составляющих основу моделирующей системы.

1.1. БЛОКИ

Разработчик конструирует модель из блоков, прибегая, как правило, к наглядной форме ее отображения в виде блок-схемы. Для удобства графического представления модели каждый блок GPSS имеет принятое стандартное обозначение. Построенная схема является одновременно программой на языке GPSS. Для ее ввода в ЭВМ необходимо последовательность блоков представить в виде списка операций, добавив к названиям блоков требуемые операнды.

Каждый блок GPSS имеет входы и выходы, с помощью которых осуществляется их связь в модели. Существуют два особых блока: GENERATE, имеющий только выход, и TERMINATE, имеющий только вход. Через блок GENERATE транзакты вводятся в модель. Блок TERMINATE удаляет транзакты из модели. Любую модель на языке GPSS можно представить в виде совокупности блоков (рис. 1.1).

Ниже дано описание основных функциональных объектов GPSS.

GENERATE

Блоки модели

TERMINATE

1.2. ТРАНЗАКТЫ

Функционирование объекта отображается в модели в виде перемещения транзактов от блока GENERATE в блок TERMINATE через промежуточные блоки. Транзакты, или сообщения являются абстрактными подвижными элементами, которые могут моделировать различные объекты реального мира: сообщения, программы, транспортные средства, людей и т.п. Перемещаясь между блоками модели, транзакты вызывают (и испытывают) различные действия. Возможны их задержки в некоторых точках модели, изменения маршрутов и направлений движения, расщепление транзактов на несколько копий и т.п. С каждым транзактом связан упорядоченный набор данных. Он включает номер транзакта; номер блока, в котором в данный момент находится транзакт; номер следующего блока; время перехода в следующий блок; приоритет, характеризующий очередность обработки транзактов в определенных случаях; а также набор параметров, с помощью которых каждому транзакту можно присвоить числовые значения, выражающие желаемые свойства или характеристики моделируемых объектов: вес, скорость, объем, цвет, время обработки и т.п.

Все объекты GPSS имеют числовые характеристики, называемые стандартными числовыми атрибутами (СЧА). Транзакты имеют четыре СЧА:

PR1 - приоритет;

P\$*j* - параметр с номером *j*;

M\$1 - время прохождения транзактом участка модели;

MP\$*j* - промежуточное время, записываемое в параметре P_j ;

X\$*j* - номер транзакта.

Блоки, влияющие на атрибуты транзактов и на их движение в модели:

В GPSS/PC существует ряд блоков, с помощью которых можно изменить значение параметров транзактов. Блок ASSIGN модифицирует значение параметра транзакта, входящего в этот блок. Блок INDEX изменяет значение параметра номер 1. Блок INCREMENT увеличивает значение параметра, блок DECREMENT уменьшает величину параметра.

Блок MARK записывает в указанный им параметр текущее значение таймера абсолютного времени или, если параметр не указан, заменяет значение отметки времени на текущее значение таймера.

С помощью блока USING осуществляется доступ текущего транзакта к параметрам другого транзакта модели. Блок LOCATE определяет и записывает в параметр текущего транзакта номер блока, в который должен войти определенный транзакт. Пересылка транзакта в требуемую точку модели осуществляется блоком TRANSFER. Блок TEST, проверяя выполнение определенного условия, может также направлять транзакты в заданную точку модели, если условие не выполняется. В этом его действия аналогичны блоку GATE. Для организации в модели циклов используют блок LOOP, с помощью которого можно предусмотреть многократное прохождение транзактом заданной цепочки блоков.

Изменение приоритета транзакта осуществляет блок PRIORITY. Блок BUFFER возобновляет просмотр списка текущих событий и используется, как правило, в качестве режима блока PRIORITY.

С помощью блоков TRACE и UNTRACE осуществляют трассировку движения транзактов в модели.

При этом по каждому транзакту выводятся данные следующего вида:

ТРАНЗ 1 ИЗ 2 В 3 ВРЕМЯ 54 TERMINATIONS TO GO 1 , которые означают: транзакт номер 1 выходит из блока 2 в блок 3 в момент времени 54 единиц, значение счетчика завершений равно 1 (см. 2.1.32, 2.3.6)

1.3. СПИСКИ

Списки относятся к элементам внутренней организации системы GPSS. Они представляют собой структуры данных, в которых размещается полная информация о транзактах. С помощью списков обеспечивается внутренняя логика работы моделирующей системы.

Список будущих событий содержит транзакты, которые смогут начать движение в модели в будущие моменты времени. Это те транзакты, для которых моменты начала движения определены в блоках GENERATE и ADVANCE.

В список текущих событий входят транзакты, которые должны перемещаться в модели в текущий момент модельного времени. Если при этом транзакт входит в блок ADVANCE с ненулевым временем задержки, то он перемещается в список будущих событий.

В список прерываний помещаются транзакты, обслуживание которых прервано блоком PREEMPT. После снятия прерывания в блоке RETURN транзакты вновь возвращаются в список будущих событий. На рис. 1.2 представлена структура списка GPSS/PC.

В список синхронизации помещаются транзакты, ожидающие объединения с другими транзактами в блоках GATHER и ASSEMBLE или находящиеся в блоках MATCH. После выполнения условий синхронизации транзакты возвращаются в список текущих событий. Кроме рассмотренных списков, обработка которых происходит без участия программиста, в GPSS/PC существуют списки пользователя, управление которыми осуществляется с помощью блоков LINK и UNLINK. Списки пользователя имеют стандартные числовые атрибуты, к которым можно обращаться внутри модели:

- CH\$j - текущее число транзактов в списке j;
- CA\$j - среднее число транзактов в списке j;
- CM\$j - максимальное число транзактов в списке j;
- CC\$j - общее число входов транзактов в список j;
- CT\$j - среднее время пребывания транзакта в списке j.

Процедура просмотра списка текущих событий:

Процесс движения транзактов в модели сопровождается просмотром содержимого списков, внесением в них изменений, а также перемещением транзактов из одного списка в другой.

Транзакты, находящиеся в списке текущих событий, расположены в порядке убывания приоритетов. Когда транзакт вводится в список текущих событий, он становится последним среди транзактов, имеющих тот же приоритет. Процедура просмотра списка текущих событий выполняет три основные действия:

2. 1. Изменение момента условного времени. При этом момент времени устанавливается равным наименьшему времени выхода из блока для транзактов, находящихся в списке будущих событий. Все транзакты, имеющие такое время выхода, переводятся из списка будущих событий в список текущих событий;

3. Последовательный просмотр транзактов от начала списка к его концу с целью определения возможностей движения их в модели;

4. Попытка продвижения транзакта в последующие блоки. Если транзакт может войти в следующий блок, выполняется программа, соответствующая данному блоку. Если дальнейшее движение транзакта невозможно в силу тех или иных условий, процедура переходит к обработке следующего транзакта. Так до тех пор, пока дальнейшее продвижение любого транзакта станет невозможным. Тогда осуществляется изменение момента условного времени, и процедура повторяется сначала.

1.4. УСТРОЙСТВА

Устройства моделируют объекты, в которых может происходить обработка транзактов. Как правило, она связана с затратами времени. Особенность устройств состоит в том, что каждое из них в данный момент времени может быть занято лишь одним транзактом. Существует аналогия между устройствами GPSS и каналами систем массового обслуживания. В GPSS имеется возможность моделировать прерывания устройств. Существуют средства логической проверки состояния устройств. Каждое из действий с устройством отображается в модели определенным блоком.

Захват и освобождение устройства моделируются соответственно блоками SEIZE и RELEASE. Для проверки состояния устройств используют GATE. Прерывание моделирует блок PREEMPT, снятие прерывания - блок RETURN.

С устройствами связаны следующие СЧА:

F\$j - состояние устройства с номером j: 0 - если устройство свободно, и 1 - если устройство занято;

FR\$j - коэффициент использования устройства];

FC\$j - число входов в устройство j;

FT\$j - среднее время использования устройства] одним транзактом.

Устройства имеют также стандартные логические атрибуты (СЛА), каждый из которых может принимать одно из двух значений: "ИСТИНА" или "ЛОЖЬ".

U - устройство занято;

NU - устройство свободно;

I - устройство прервано;

N1 - не прервано.

Проверка состояния устройства осуществляется блоком GATE, который в зависимости от значения проверяемого СЛА либо беспрепятственно пропускает транзакты (если СЛА = "истина"), либо задерживает их или направляет по другому маршруту (если СЛА = "ложь").

1.5. ПАМЯТИ

Памяти служат для моделирования объектов, обладающих определенной емкостью. Памяти моделируются блоками ENTER и LEAVE. Входящий в блок ENTER транзакт занимает определенную часть памяти. При входе транзакта в блок LEAVE память освобождается. Емкость памяти задают с помощью оператора STORAGE, который не является блоком GPSS и относится к числу служебных карт.

Памяти имеют следующие стандартные числовые атрибуты:

S\$j - емкость памяти j;

R\$j - свободный объем памяти j;
SR\$j - коэффициент использования памяти];
SM\$j - максимальное заполнение памяти];
SA\$j - среднее заполнение памяти j];
SC\$j - число входов в память j];
ST\$j - среднее время пребывания транзакта в памяти j].

Памяти имеют также стандартные логические атрибуты, которые используются для проверки состояния памяти:

SE\$j - память j пуста;
SNE\$j - память j не пуста;
SF\$j - память j заполнена;
SNF\$j - память j не заполнена.

Проверка состояния памяти осуществляется блоком GATE.

1.6. ЛОГИЧЕСКИЕ КЛЮЧИ

Для представления в модели коммутлируемых объектов с двумя состояниями ("ВКЛЮЧЕНО" - "ВЫКЛЮЧЕНО") используют логические ключи, моделируемые блоками LOGIC и GATE. При входе транзакта в блок LOGIC происходит срабатывание соответствующего ключа. Ключ может быть включен (S), выключен (R) или переключен (I). Ключи не имеют СЧА. Их состояние определяется стандартными логическими атрибутами:

LSS\$j - ключ j включен;
LR\$j - ключ j выключен.

Проверка состояния ключа осуществляется блоком GATE.

1.7. ОЧЕРЕДИ

Транзакты в процессе движения могут задерживаться перед блоками, вход в которые в данных условиях невозможен. Примерами таких блоков из числа рассмотренных выше являются SEIZE (если ранее вошедший в этот блок транзакт не вошел в блок RELEASE), ENTER (если текущий объем памяти равен первоначально заданному), GATE (если в этом блоке не указан альтернативный выход и проверяемое условие не выполняется). При поступлении транзактов на вход задерживающих блоков образуются очереди. Для сбора статистики об очередях в местах задержки ставят блоки QUEUE. Эти блоки сами по себе не создают очередь, а лишь являются средством ее регистрации. При входе транзакта в блок QUEUE текущая длина очереди получает приращение. Уход из очереди отображается блоком DEPART.

Очереди имеют следующие стандартные числовые атрибуты:

Q\$j - текущая длина очереди j];
QM\$j - максимальная длина очереди j];
QA\$j - средняя длина очереди j];
QC\$j - число входов в очередь j];

QZ\$j - число входов в очередь с нулевым временем пребывания (транзакт прошел через блок QUEUE, не задерживаясь в очереди);
QT\$j - среднее время пребывания в очереди], включая нулевые входы;
QX\$j - среднее время пребывания в очереди], без нулевых входов.

1.8. ТАБЛИЦЫ

Для сбора статистических данных о различных отчетах модели и их представления в стандартной табличной форме используют таблицы. Занесение информации в таблицу осуществляется блоком TABULATE в момент входа очередного транзакта в этот блок. Описание структуры таблицы и типа заносимых данных (СЧА) осуществляется картой TABLE.

Стандартные числовые атрибуты таблиц:

TB\$j - среднее значение фиксируемой в таблице переменной;

TC\$j - число входов в таблицу];

TD\$j - стандартное среднеквадратическое отклонение табулируемой переменной.

1.9. ЯЧЕЙКИ

Для записи в процессе моделирования текущих значений СЧА используют ячейки. Занесение информации в ячейку осуществляет блок SAVEVALUE. При входе транзакта в этот блок требуемое значение СЧА фиксируется в ячейке, номер которой определяется операндом блока SAVEVALUE. Ячейки имеют различные форматы (слово, полуслово, с плавающей точкой), которым соответствуют следующие СЧА:

X\$j] - текущее значение, записанное в ячейке j формата слово;

XH\$j] - текущее значение, записанное в ячейке j формата полуслова;

XL\$j] - содержимое ячейки с плавающей точкой.

Изменение содержимого ячеек может осуществляться блоками SDECREMENT и SINCREMENT. Блок SDECREMENT вычитает заданное значение из величины, содержащейся в ячейке. Блок SINCREMENT добавляет требуемое приращение к содержимому ячейки.

1.10. ФУНКЦИИ И ПЕРЕМЕННЫЕ

Функции служат для отображения зависимостей между двумя (СЧА. В GPSS/PC имеется два типа функций: непрерывные (C) и дискретные (D). Функцию задают набором пар точек - координат. Непрерывная функция воспроизводится в виде ломаной кривой, отрезки которой соединяют соседние точечные значения.

Дискретная функция имеет вид ступенчатой кривой. На рис. 1.3 представлены непрерывная (а) и дискретная (б) функции, соответствующие одному и тому же набору точек $X_1, Y_1; X_2, Y_2; X_3, Y_3; X_4, Y_4$, где X означает аргумент, а Y - значение функции.

Функцию описывают картой FUNCTION. За ней помещают одну или несколько карт, содержащих координатные точки.

Карта описания функции задает ее тип, количество пар точек (X_i, Y_i) и СЧА, используемый в качестве аргумента. Формат этой карты имеет следующий вид:

НОМЕР FW[FUNCTION] A,B

Где: НОМЕР - номер функции, задаваемый программистом;

A - аргумент (любой СЧА);

B - тип функции.

Тип функции указывают как C_n - для непрерывной и D_n - для дискретной функции, где n - число пар точек (X_i, Y_i) .

Карты, содержащие точки (X_i, Y_i) , имеют следующий вид:

$X_1, Y_1 / X_2, Y_2 / \dots / X_i, Y_i / \dots / X_n, Y_n$ причем обязательно $X_1 < X_2 < \dots < X_i < \dots < X_n$.

В GPSS/PC существуют также операторы для описания переменных, составляемых из стандартных числовых атрибутов. Целочисленные переменные, а также булевы переменные описываются оператором VARIABLE. Переменные с плавающей точкой описываются оператором FVARIABLE.

Операторы переменной имеют следующий формат:

ИМЯ VARIABLE SNA(oper)SNA(oper)... (FVARIABLE)

где:

SNA - допустимые стандартные числовые атрибуты;

(oper) - арифметические и логические операции:

"+" - сложить;

"-" - вычесть;

"/" - разделить;

"@" - разделить по модулю;

"()" - скобки;

"1" - логическое "И";

" " - логическое "ИЛИ";

"=" - равенство;

"o" - не равно;

"<" - меньше;

">=" - больше чем или равно;

"<=" - меньше чем или равно;

">" - больше.

2. ФОРМАТЫ ОПЕРАТОРОВ GPSS/PC

Операторы GPSS/PC имеют следующий формат:

Метка _____ Операция __ Операнды __ ;Комментарии

Знак "_" указывает пробел, знак ";" объявляет начало поля комментариев.

Метка, если она имеется, должна начинаться с первой колонки и содержать не более пяти алфавитно-цифровых символов, начинающихся с буквы.

Звездочка (*) в первой колонке означает строку комментариев.

Поле операции содержит название блока или служебного оператора (карты). Это поле может начинаться со второй колонки и должно быть отделено от метки пробелом. Длина поля операции не менее четырех символов (начальные символы блоков или карт).

Поле операндов отделяют от поля операции пробелом. Между операндами должны стоять запяты.

Пример:

```
^FACILITY DESCRIPTION
FAC SEIZE 1 ;CPU
    ADVAN 10,5 ;MSEC
    RELEASE 1
MAC TRANSFER 3,FAC
```

Карты описания таблиц, функций, переменных и памяти должны иметь в поле метки число от 1 до 32767 ($2^{15} - 1$) или предварительно определенный символ.

Примеры:

```

^ ENTITIES DESCRIPTION
1 STORAGE      1280 ;MAIN
4 TABL        M1Д10ДО

```

В полях операндов могут быть использованы следующие обозначения.

Константа - целое число от 1 до 32767 или предварительно определенная последовательность символов.

СЧА \$ const - стандартный числовой атрибут, номер которого определяется константой const.

* const - значение параметра, номер которого определяется константой const

СЧА1*СЧА2 \$ const - косвенная адресация с использованием стандартного -числового атрибута 1 (СЧА1), номер которого задан значением стандартного числового атрибута 2 (СЧА2), определяемого константой const.

СЧА%СЧА \$ const - косвенная адресация через стандартные числовые атрибуты.

Примеры:

XF*V\$2

Содержимое полнословной ячейки, номер которой определяется значением переменной 2.

FN*P\$1 Значение функции, номер которой определяется содержимым параметра 1.

XЩ2 Содержимое полусловной ячейки номер два.

*TERM

Значение параметра, номер которого определен константой TERM. В описательной части программы значение константы TERM должно быть определено оператором EQU, например: TERM EQU 10 при этом *TERM означает содержимое десятого параметра. Ниже дано полное описание форматов блоков, а также служебных и управляющих операторов GPSS/PC.

2.1. БЛОКИ

2.1.1. ADVANCE (ЗАДЕРЖАТЬ)

ADVANCE - задерживает транзакт.

Формат: ADVA[NCE] A,B

A - среднее время задержки (константа, если B не задано);

B - разброс относительно среднего значения, должен быть меньше или равен A.

Блок ADVANCE моделирует временную задержку транзакта в течение определенного интервала. Значение задержки по умолчанию равно нулю. Если поле B не является FN\$, то время задержки является случайным числом, распределенным равномерно на интервале от (A+B) до (A-B). Если поле B является функцией FN\$, то время задержки определяется произведением поля A на значение функции в поле B. Функция может быть использована для задания времени задержки с определенным средним значением и отклонением, зависящими от A.

Примеры:

ADVANCE 75

Транзакт будет оставаться в блоке ADVANCE в течение 75 единиц модельного времени.

ADVANCE 12,5

Транзакт будет оставаться в блоке от 7 до 17 единиц времени.

ADVANCE 5, FN\$1

Время задержки равно произведению значения функции FN\$1 на число пять.

См. также:

GENERATE

2.1.2. ASSEMBLE (СОЕДИНИТЬ)

ASSEMBLE - объединение транзактов, принадлежащих одному семейству (или ансамблю).

Формат: ASSE[MBLE] A

A - число объединяемых транзактов.

Первый транзакт семейства, достигнув блока ASSEMBLE, задерживается в нем до тех пор, пока остальные члены семейства не поступят в этот блок. Когда транзакты, число которых указано в поле A, поступят в этот блок, они будут удалены из модели, а первый прибывший транзакт продолжит движение.

Пример:

ASSEMBLE 3

После того, как 3 транзакта одного семейства войдут в блок, один (первый) выйдет из блока и продолжит движение, остальные будут уничтожены.

См. также:

GATHER

MATCH

SPLIT

2.1.3. ASSIGN (ПРИСВОИТЬ)

ASSIGN - изменяет значение параметра транзакта.

Формат: ASSI[GN] A,B

A - номер изменяемого параметра (+, -);

B - новое значение параметра.

Если за полем A следует знак "+" или "-", то значение поля B соответственно добавляется или вычитается из A. Если знаки "-" или "+" не указаны, то значение поля B становится текущим значением параметра.

Примеры:

ASSIGN 2,8

Присваивает параметру 2 значение 8.

ASSIGN 3+, V\$5

Добавляет значение переменной 5 к параметру 3.

См. также:

DECREMENT INCREMENT

2.1.4. BUFFER (ВОЗОБНОВИТЬ ПРОСМОТР СПИСКА)

BUFFER - возобновляет просмотр списка текущих событий.
Формат: BUFF[ER]
Не имеет операндов.

Останавливает процесс движения транзакта и заново начинает просмотр списка текущих событий. Не связан с временем задержки.

Пример:

BUFFER Возобновляет просмотр списка текущих событий.

См.также:

PRIORITY

2.1.5. DECREMENT (УМЕНЬШИТЬ ЗНАЧЕНИЕ)

DECREMENT - уменьшает значение параметра транзакта.
Формат: DECREMENT] A,B

A - номер изменяемого параметра;

B - значение для вычитания.

Величина, указанная в поле B, вычитается из значения параметра, номер которого указан в поле A.

Примеры:

DECREMENT 3,7

Вычитается число 7 из значения третьего параметра текущего транзакта.

DECREMENT XH\$6,V\$4

Значение переменной 4 вычитается из параметра, номер которого задан ячейкой 6 формата полуслово.

См.также:

ASSIGN

INCREMENT

2.1.6. DEPART (ПОКИНУТЬ ОЧЕРЕДЬ)

DEPART - удаляет транзакт из очереди.

Формат: DEPA[RT] A,B

A - номер (имя) очереди;

B - число удаляемых из очереди элементов.

Удаляет текущий транзакт из очереди, указанной в поле A, и уменьшает содержимое очереди на значение поля B. Транзакт может находиться одновременно в двух различных очередях.

Примеры:

DEPART 5

Текущий транзакт удаляется из очереди 5, а длина очереди уменьшается на единицу.

DEPART P\$3,7

Транзакт удаляется из очереди, номер которой определен параметром 3, длина очереди уменьшается на 7 элементов.

См.также:

QUEUE

2.1.7. ENTER (ВОЙТИ В ПАМЯТЬ) ENTER

- помещает транзакт в память.

Формат: ENTE[R] A,B

A - имя памяти символическое или числовое;

B - число занимаемых единиц памяти (по умолчанию единица).

Проверяется наличие свободного объема памяти, номер которой определен в поле A. Если имеется свободная память, то транзакт входит в блок ENTER. При этом занятый объем памяти увеличивается на значение поля B. Если транзакт не может войти в блок ENTER, он задерживается в предыдущем блоке.

Примеры:

ENTER 1

Войти в память 1, занимая единицу ее объема.

ENTER PS1DO

Войти в память, номер которой определяется параметром 1, занимая 10 единиц памяти.

См.также:

LEAVE STORAGE

2.1.8. GATE (ВПУСТИТЬ)

GATE - вспомогательный блок, проверяющий состояния устройств, памяти, логических ключей.

Формат: GATE_R A,B

Внутренний операнд R определяет проверяемое условие в виде стандартного логического атрибута и может принимать следующие значения:

U - устройство занято;

NU - устройство не занято;

I - устройство прервано;

N1 - устройство не прервано;

SF - память заполнена;

SNF - память не заполнена;

SE - память пустая;

SNE - память не пустая;

LR - ключ выключен;

LS - ключ включен;

M - транзакт находится в состоянии синхронизации;

MN - транзакт не находится в состоянии синхронизации.

Если проверяемое условие для объекта, номер которого определяется полем A, выполняется (СЛА "ИСТИНА"), то транзакт входит в блок GATE. Если условие "ЛОЖЬ", то возможны два случая:

- если поле B задано, то транзакт идет в блок, номер которого указан в поле B;

- если в поле B пробел, то транзакт ждет в предыдущем блоке, пока не выполнится условие.

Примеры:

GATE_SF 1

Если память номер 1 полна, то транзакт входит в блок GATE, в противном случае ждет ее заполнения.

GATE_NU P\$2,МЕТ

Если устройство, номер которого определен параметром P\$2, не занято, то транзакт входит в блок GATE, в противном случае он идет в блок с именем МЕТ.

2.1.9. GATHER (СОБИРАТЬ)

GATHER - накапливает транзакты, являющиеся членами семейства.

Формат: GATHER_A

A - счетчик транзактов, которые должны быть накоплены.

Транзакты одного семейства задерживаются в блоке GATHER до тех пор, пока их число не станет равным значению поля A. Когда последний транзакт войдет в блок GATHER, все они одновременно выходят из него в том порядке, в котором поступили. Состояние блока GATHER может быть проверено блоком GATE.

Пример:

GATHER 6

Транзакты накапливаются в этом блоке до тех пор, пока в нем не соберутся шесть транзактов из одного семейства, после чего все они смогут продолжать движение.

См. также:

ASSEMBLE

MATCH

SPLIT

2.1.10. GENERATE (ГЕНЕРИРОВАТЬ)

GENERATE - вводит транзакты в модель.

Формат: GENE[RATE] A,B,C,D,E,F,G

A - среднее значение интервала времени;

B - разброс или модификатор среднего значения (по умолчанию ноль);

C - время появления первого транзакта;

D - общее число генерируемых транзактов;

E - уровень приоритета каждого транзакта; (от 0 до 127, знач. по умолчанию 0);

F - число параметров (по умолчанию 12);

G - тип параметра (F -полнословный,

H -полусловный -по умолчанию).

Вводит транзакты в модель, посылая их в следующий по порядку блок. Если в поле B не указана функция, то интервал между поступлением транзактов определяется случайным числом, равномерно распределенным в диапазоне от (A-B) до (A+B). Если поле B является функцией (FN\$), то этот интервал определяется произведением поля A на значение функции, заданной в поле B.

Примеры:

GENERATE 15,3,25

Генерируются транзакты с интервалом прихода от 12 до 18 единиц времени, первый из которых поступает в момент времени 25 единиц.

GENERATE 5,FN\$2,,15

Генерируются транзакты с приоритетом 15, интервалы времени между ними определяются произведением числа 5 на значение функции FN\$2.

См. также:

ADVANCE
TERMINATE

2.1.11. PREEMPT (ПРЕРВАТЬ)

PREEMPT - переводит устройство в прерванное состояние.

Формат: PREE[MPT] A

A - номер прерываемого устройства.

Транзакт получает в пользование устройство, указанное в поле A, если это устройство не было прервано другим транзактом. Если предыдущий транзакт захватил устройство через блок PREEMPT, текущий транзакт блокируется.

Пример:

PREEMPT P\$1

Если устройство, номер которого задан параметром P1, не было переведено в состояние прерывания, то транзакт, входящий в этот блок, захватывает его.

См. также:

SEIZE
RELEASE
RETURN

2.1.12. PRINT (НАПЕЧАТАТЬ)

PRINT - печатать до семи значений СЧА.

Формат: PRIN[T] SNA1,SNA2,...,SNA7

SNA - любые допустимые стандартные числовые атрибуты.

Печатается список от 1 до 7 СЧА при каждом входе транзакта в этот блок. Значения СЧА не изменяются.

Примеры:

PRINT P\$1,P\$2,P\$3,P\$4

Печать значений параметров 1,2,3,4.

PRINT XH\$3,XH\$4,V\$2,FR\$8,P\$9,P\$10

Печать значений ячеек 3 и 4, переменной 2, коэффициента использования прибора 8 и содержимого параметров 9 и 10.

2.1.22. PRIORITY (НАЗНАЧИТЬ ПРИОРИТЕТ)

PRIORITY - изменяет уровень приоритета транзакта.

Формат: PRIORIT[ITY] A,BUFFER

A - новый приоритет (целое число в диапазоне от 0 до 127); [BUFFER] - указывает возобновление процедуры просмотра списка текущих событий.

Устанавливает приоритет входящему транзакту в соответствии со значением поля A. Если операнд BUFFER отсутствует, продолжается обработка текущего транзакта. Если операнд BUFFER указан, то текущий транзакт помещается в список текущих событий и возобновляется просмотр списка.

Примеры:

PRIORITY 4

Назначить транзакту уровень приоритета 4, попытка продвинуть транзакт в следующий блок.

PRIORITY XF\$1,BUFFER

Назначить транзакту приоритет, равный значению ячейки 1, поместить транзакт в список текущих событий и возобновить его просмотр.

См.также:

BUFFER GENERATE

2.1.13. QUEUE (СТАТЬ В ОЧЕРЕДЬ) QUEUE -

помещает транзакт в конец очереди.

Формат: QUEU[E] A,B

A - номер очереди (числовое или символьное имя очереди); B - число добавляемых к очереди элементов (по умолчанию 1).

Увеличивает текущее содержимое очереди, указанной в поле A, на значение в поле B. Если поле B не определено, увеличивает содержимое очереди на единицу. Транзакт может находиться в двух различных очередях одновременно.

Примеры:

QUEUE 5

Присоединить транзакт к очереди 5,увеличив ее длину на единицу.

QUEUE P\$1,3

Стать в очередь, указанную в параметре 1, и увеличить ее длину на 3.

См.также:

DEPART

2.1.14 RELEASE (ОСВОБОДИТЬ)

RELEASE - освобождает устройство.

Формат: RELE[ASE] A

A - номер устройства (числовое или символьное имя освобождаемого устройства).

Устройство, указанное в поле A, освобождается и становится доступным для других транзактов. Освободить устройство должен тот же транзакт, который его занимал.

Примеры:

RELEASE 10

Освободить устройство 10.

RELEASE P\$2

Освободить устройство, указанное в параметре 2.

См.также:

PREEMPT

RETURN

SEIZE

2.1.15. TABULATE (ЗАНЕСТИ В ТАБЛИЦУ)

TABULATE - заносит значение в таблицу.

Формат: TABULATE] A,B A - номер таблицы;

В - вес, указывающий сколько раз значение должно быть занесено в таблицу (по умолчанию 1).

Транзакты, входящие в блок TABULATE, осуществляют занесение данных в таблицу, указанную в поле А. Поле В определяет весовой фактор, который используется, если данные заносятся во взвешенную таблицу. Описание структуры таблицы осуществляется картой TABLE.

Примеры:

TABULATE 4 Занести значение в таблицу 4 с весом 1 (т.е один раз).

TABULATE P\$3,5

Занести наблюдаемую величину с весом 5, т.е 5 раз, во взвешенную таблицу, указанную в параметре.

2.1.16. TERMINATE (ЗАВЕРШИТЬ)

TERMINATE - удаляет транзакт.

Формат: TERMINATE] А

А - величина, вычитаемая из содержимого счетчика завершений (поле А карты Транзакт удаляется из модели и поступает в пассивный буфер. Если в поле А пробел, воздействия на счетчик завершений не происходит, в противном случае его значение уменьшается на величину, указанную в поле А.

Примеры:

TERMINATE Транзакт удален, но значение счетчика завершений не изменяется.

TERMINATE 2 Значение счетчика завершений уменьшается на 2.

См.также:

GENERATE

2.1.17. TEST (ПРОВЕРИТЬ) TEST - сравнивает два

стандартных числовых атрибута.

Формат: TESTj А,В,С

г - внутренний операнд, принимающий значения:

Е - равно;

NE - не равно;

L - меньше чем;

LE - меньше чем или равно;

G - больше чем;

GE - больше чем или равно;

А - стандартный числовой атрибут;

В - стандартный числовой атрибут;

С - номер альтернативного блока.

Значения стандартных числовых атрибутов, указанных в поле А и В, сравниваются отношением, определяемым операндом г. Если условие выполняется, транзакт вводится в блок. Если условие не выполняется и определено поле С, транзакт переходит в указанный блок, если же С не задано, транзакт задерживается в предыдущем блоке.

Примеры:

TEST_E P\$1,2,LABEL

Если значение первого параметра равно двум, транзакт войдет в блок TEST и продолжит движение в следующий блок. Если значения не равны, транзакт перейдет в блок с меткой LABEL.

TEST_L FN\$2,P\$2

Если функция 2 меньше, чем параметр 2, транзакт входит в блок TEST, в противном случае он становится заблокированным.

2.1.18. TRANSFER (ПЕРЕДАТЬ)

TRANSFER - изменяет движение транзакта в модели.

Формат: TRANSFER] A,B,C,D

A - режим передачи (пробел, ДЬЗОТН,Ж,Р,РІСК,8ВК,81М);

B - следующий блок;

C - следующий блок;

D - значение индекса, используемое в режиме ALL.

Транзакт направляется в блок, определяемый в соответствии с режимом передачи, указанным в поле A.

Режимы передачи поля A:

1. Пробел - транзакт передается в блок, определяемый полем B.
2. "." - статистический режим: в поле A указано десятичное число, выражающее вероятность перехода в блок C; его дополнение до единицы указывает вероятность перехода в блок B.
3. ALL - транзакт последовательно пытается перейти в блоки, определяемые значениями B, B+D, B+2D,...,C.
4. BOTH - транзакт последовательно пытается войти в блок B, затем в блок C, до тех пор, пока один из них станут доступным.
5. FN - функциональный режим: поле B является номером функции; следующий блок определяется суммой значения этой функции поля C.
6. P - параметрический режим: поле B является номером параметра; следующий блок определяется суммой значения этого параметра и поля C.
7. PICK - выборочный режим: блок выбирается с равной вероятностью из блоков с номерами: B, B+1,..., C.
8. SBR - режим перехода к подпрограмме: номер текущего блока помещается в параметр, указанный в поле C, а транзакт передается в блок, номер которого указан в поле B.
9. SIM - одновременный режим: проверяется одновременное выполнение условий беспрепятственного движения транзактов в задерживающих блоках. Если условие выполняется, транзакт передается в следующий блок, в противном случае транзакт переходит на блок C.

Примеры:

TRANSFER ,NEXT Безусловная передача в блок с меткой NEXT.

TRANSFER .400,FAC1,FAC2

С вероятностью 0.6 транзакты будут переданы в блок FAC1 и с вероятностью 0.4 в блок FAC2,

[на начало пособия](#)

[на содержание документа](#)