

Серия «Проектирование»

Кудрявцев Е. М.

GPSS World

Основы имитационного моделирования
различных систем



Москва, 2004

УДК 004.94
ББК 32.973.26-018.2
К88

Кудрявцев Е. М.

К88 GPSS World. Основы имитационного моделирования различных систем. – М.: ДМК Пресс, 2004. – 320 с.: ил. (Серия «Проектирование»).

ISBN 5-94074-219-X

GPSS World – мощная универсальная среда моделирования как дискретных, так и непрерывных процессов, предназначенная для профессионального моделирования самых разнообразных процессов и систем. Книга содержит описание структуры среды GPSS World. Представлены основные этапы программирования. Даются общие понятия и определения языка имитационного моделирования GPSS. Особое внимание уделено вопросам разработки программ на этом языке. Представлены различные виды экспериментов: пользовательские, сгенерированные системой и оптимизационные. Книга содержит большое количество рисунков, иллюстрирующих интегрированную среду моделирования GPSS World, программирование на языке GPSS и решения конкретных задач. Автор приводит многочисленные примеры моделирования как производственных, так и непроизводственных систем, например систем массового обслуживания и др.

Книга предназначена для широкого круга читателей: учащихся, студентов, инженеров, менеджеров и других специалистов, занимающихся моделированием функционирования всевозможных систем.

ББК 32.973.26-018.2

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

Материал, изложенный в данной книге, многократно проверен. Но, поскольку вероятность технических ошибок все равно существует, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 5-94074-219-X

© Кудрявцев Е. М., 2003
© ДМК Пресс, 2004

Содержание

Предисловие	10
-------------------	----

Глава 1

Интегрированная среда GPSS World

Структура и запуск GPSS World	14
-------------------------------------	----

Главное меню GPSS	15
-------------------------	----

Меню File	16
-----------------	----

Состав и структура меню	16
-------------------------------	----

Создание нового файла	17
-----------------------------	----

Открытие существующего файла	20
------------------------------------	----

Закрытие файла	20
----------------------	----

Сохранение файла в текущем каталоге	20
---	----

Сохранение файла в другом каталоге	21
--	----

Установка параметров принтера и печати	21
--	----

Настройка принтера и установка параметров бумаги	21
--	----

Связь с Internet	21
------------------------	----

Выход из системы GPSSW	22
------------------------------	----

Меню Edit	22
-----------------	----

Вставка блоков GPSS в модель	24
------------------------------------	----

Меню Search	24
-------------------	----

Меню View	25
-----------------	----

Меню Command	25
--------------------	----

Меню Window	26
-------------------	----

Меню Help	27
-----------------	----

Панель инструментов GPSSW	28
---------------------------------	----

Окно исходной модели	29
----------------------------	----

Просмотр значений выражений в динамике	33
--	----

Ввод и редактирование текста	35
Выделение фрагмента текста	35
Удаление фрагмента текста	36
Помещение фрагмента текста в буфер обмена данными	37
Вставка фрагмента текста из буфера обмена данными	37
Специальные комбинации клавиш	37
Изменение шрифта в тексте	38
Поиск и замена фрагмента текста	39
Настройка установок	40
Установка параметров моделирования	40
Установка содержания отчета	41
Установка параметров генераторов случайных чисел	42
Назначение функциональных клавиш	43
Установка просмотра выражений	44
Установка символа умножения	46

Глава 2

Основы моделирования в системе GPSSW

Базовые понятия и определения	48
Система массового обслуживания	48
Имитационная модель	48
Представление времени в процессе имитации	49
Динамические элементы системы	50
Именованные величины	52
Типы данных	53
Элементы выражений	54
Арифметические целые переменные	54
Арифметические переменные с фиксированной точкой	55
Булевы переменные	55
Вычислительные выражения	55
Основные операторы языка GPSS	58
Цепи событий	61
Основные составляющие системы GPSSW	64
Типы объектов	64
Операторы языка PLUS	65

Строковые процедуры	65
Математические процедуры	66
Сервисные процедуры	66
Функции типовых распределений вероятностей	67
Основные этапы моделирования в системе GPSSW	67
Постановка задачи	68
Выявление основных особенностей	68
Создание имитационной модели процесса	69
Моделирование системы	70
Модернизация исходной модели	71
Отладка модели	71
Снимки и динамические окна	72
Инициализация элементов	75
Определение матрицы	75
Моделирование в интерактивном режиме	76
Постановка задачи	77
Создание имитационной модели	77
Транслирование модели	77
Открытие диалогового окна BLOCK ENTITIES	78
Моделирование системы	78
Просмотр сообщений	79
Отладка модели	84

Глава 3

Моделирование непроизводственных систем 91

Моделирование работы магазина	92
Постановка задачи	92
Выявление основных особенностей	92
Создание имитационной модели процесса	92
Представление имитационной модели	94
Подготовка к моделированию системы	94
Моделирование системы	96
Графическое представление результатов моделирования	96
Визуализация процесса функционирования системы	100
Просмотр отдельных результатов моделирования	104

Моделирование движения на пешеходном переходе	109
Постановка задачи	109
Выявление основных особенностей	109
Построение имитационной модели	110
Моделирование системы	114
Визуализация процесса функционирования моделируемой системы	118
Моделирование работы переговорного пункта	121
Постановка задачи	121
Выявление основных особенностей	121
Создание имитационной модели процесса	122
Представление имитационной модели	124
Подготовка к моделированию системы	124
Моделирование системы	126
Вывод гистограммы	128
Моделирование работы супермаркета	129
Постановка задачи	129
Выявление основных особенностей	129
Создание имитационной модели процесса	130
Представление имитационной модели	135
Подготовка к моделированию системы	135
Моделирование системы	138
Моделирование системы «Хищник–добыча»	141
Постановка задачи	141
Выявление основных особенностей	142
Создание имитационной модели процесса	142
Представление имитационной модели	143
Моделирование системы	144
Вывод результатов в графическом виде	145
Анализ полученных результатов	146
Моделирование распространения эпидемии	148
Постановка задачи	148
Выявление основных особенностей	149
Создание имитационной модели процесса	149
Представление имитационной модели	151
Моделирование системы	152
Анализ полученных результатов	154

Глава 4

Моделирование производственных систем	157
Моделирование работы транспортного конвейера	158
Постановка задачи	158
Выявление основных особенностей	158
Создание имитационной модели процесса	159
Модификация моделируемой системы	163
Выявление основных особенностей	165
Модель модифицированной системы	165
Оценка надежности работы системы	168
Постановка задачи	168
Выявление основных особенностей	168
Построение имитационной модели	170
Моделирование системы	171
Моделирование работы участка цеха	174
Постановка задачи	174
Выявление основных особенностей	174
Создание имитационной модели	175
Проведение имитационного моделирования	178
Анализ результатов имитационного моделирования	182
Моделирование работы автозаправочной станции	183
Постановка задачи	183
Выявление основных особенностей	183
Построение имитационной модели процесса	183
Представление имитационной модели	186
Моделирование системы	187
Графическое представление результатов моделирования	188
Визуализация процесса функционирования системы	193
Вывод отдельных результатов моделирования	196
Моделирование работы инструментальной кладовой	197
Постановка задачи	197
Выявление основных особенностей	197
Создание имитационной модели процесса	198
Представление имитационной модели	202
Моделирование системы	202
Подготовка к моделированию	203

Моделирование системы управления качеством	205
Постановка задачи	205
Выявление основных особенностей	206
Создание имитационной модели процесса	206
Представление имитационной модели	210
Моделирование системы	210
Построение гистограммы	212
Моделирование системы управления запасами	213
Постановка задачи	213
Выявление основных особенностей	214
Создание имитационной модели процесса	214
Представление имитационной модели	217
Моделирование системы	218
Построение графиков отдельных параметров функционирования системы	221
Анализ результатов	223
Оценка эффективности работы системы с учетом отказов	229
Постановка задачи	229
Выявление основных особенностей	229
Создание имитационной модели	230
Моделирование системы	233
Подготовка к моделированию	234

Глава 5

Моделирование систем

массового обслуживания

Классификация систем массового обслуживания	238
Анализ работы разомкнутых СМО	240
Одноканальная разомкнутая СМО с простейшими потоками	240
Решение задачи аналитическим методом	241
Решение задачи имитационным методом	245
Одноканальная разомкнутая СМО с равномерными потоками	250
Многоканальная разомкнутая СМО с простейшими потоками	257
Аналитический метод решения задачи	258
Имитационный метод решения задачи	260
Многоканальная разомкнутая СМО со смешанными потоками	266

Анализ работы замкнутых СМО	274
Одноканальная замкнутая СМО с простейшими потоками	274
Решение задачи имитационным методом	278
Одноканальная замкнутая СМО с равномерными потоками	284
Многоканальная замкнутая СМО с простейшими потоками	290
Аналитический метод решения	292
Решение задачи имитационным методом	294
Анализ работы многофазных СМО	300
Многофазная замкнутая СМО с равномерными потоками	300
Многофазная разомкнутая СМО с равномерными потоками	306
Предметный указатель	314

Предисловие

В мире информационных технологий имитационное моделирование переживает второе рождение. И это в первую очередь связано с появлением в 2000 году мощного программного продукта фирмы Minuteman Software – GPSS World (GPSSW, General Purpose System Simulation World – Мировая общецелевая система моделирования), разработанного для ОС Windows. Этот программный продукт вобрал в себя весь арсенал новейших информационных технологий. Он включает развитые графические оболочки для создания моделей и интерпретации выходных результатов моделирования, мультимедийные средства и видео, объектно-ориентированное программирование и др. В основу системы GPSS World положен язык имитационного моделирования GPSS (General Purpose System Simulation – Общецелевая система моделирования), разработанный профессором Гордоном чуть более 40 лет назад. В такой бурно развивающейся области, как программное обеспечение, только небольшое количество языков программирования достигло подобного почтенного возраста. Долгожительством языка GPSS объясняется многими причинами:

- он прост в изучении и использовании;
- многие пользователи достигли больших успехов при решении реальных проблем с использованием именно языка GPSS;
- наиболее важные классы объектов (требования (транзакты), каналы, накопители, логические переключатели и др.) и их свойства широко используются в реальных вычислительных сетях, производственных и коммерческих системах и т.д.;
- диапазон использования языка достаточно широк;
- язык постоянно совершенствуется;
- расширение создаваемых моделей легко осуществимо;
- доступно широкое использование анимации;
- пользователи способны легко понять внутреннюю логику и алгоритмы GPSS;
- интерфейс прост и удобен;
- при построении модели язык позволяет оперировать непосредственно понятиями имитируемой системы.

Система GPSS World – мощная универсальная среда моделирования как дискретных, так и непрерывных процессов, предназначенная для профессионального моделирования самых разнообразных процессов и систем. Эта система явилась следующим шагом развития системы GPSS/PC (1984 год), ориентированной на DOS. Обе системы разработаны специалистами фирмы Minuteman Software (основана в 1982 году) под руководством Спрингера Кокса. Сначала система GPSS World появилась в 1994 году с ориентацией на OS/2 фирмы IBM, и только в 2000 году она была реализована под ОС Windows фирмы Microsoft.

В GPSS World появились дополнительные возможности, вот основные из них:

- по всем классам объектов и переменных реализованы динамические графические окна, в которых представляется в реальном времени промежуточная и выходная статистика;
- гибкий процедурный язык PLUS может быть использован для построения моделей и в процедурах проведения эксперимента;
- введены средства поддержки факторного анализа, традиционного дисперсионного (ANOVA) и регрессионного анализа, оптимизация на основе методологии оптимального планирования эксперимента;
- стали доступны элементы непрерывного моделирования;
- решены проблемы с целочисленным модельным временем.

С помощью этой системы, например, можно эффективно моделировать как производственные, так и непроизводственные процессы: функционирование торговых и увеселительных заведений, портов, уличное движение, проведение военных действий, работу редакций, учреждений и сети Internet, различных систем массового обслуживания и т.д. Система имеет большой набор команд для управления процессом моделирования, которые можно как использовать в интерактивном режиме, так и включать в модель. Обеспечена возможность проведения экспериментов, сгенерированных системой, пользовательских и оптимизационных. В системе GPSSW реализована процедура визуализации процесса функционирования модели с использованием методов мультипликации.

Система GPSSW имеет новый высокоскоростной транслятор, работающий в сотни раз быстрее его предшественников. Для быстрого исправления ошибок используется полноэкранный текстовый редактор.

Системы GPSSW и GPSS/PC совместимы и обычно выдают результаты, которые являются статистически неразличимыми. Подобный уровень совместимости доступен простым устранением нескольких различий. Однако с точки зрения функционирования система GPSSW имеет существенное отличие от GPSS/PC. Если система GPSS/PC работает в режиме интерпретатора – построчного выполнения программы, – то GPSSW функционирует в режиме транслятора. Любые ошибки, обнаруженные в течение трансляции, сохраняются в очереди сообщений об ошибках, так что их легко найти и исправить. Курсор автоматически перемещается в местоположение очередной ошибки, и сообщение об ошибке появляется в строке состояния в нижней части главного окна системы.

Другое существенное отличие GPSSW от GPSS/PC заключается в том, что в новой системе нет необходимости нумеровать строки программы. Если же в модели есть нумерация, то система GPSSW ее просто игнорирует.

Система включает большое число различных типов окон, упрощающих просмотр и анализ объектов модели. Дружественный интерфейс облегчает работу с системой. В ней имеется библиотека распределений вероятностей, которые можно широко использовать в процессе моделирования. Кроме того, есть библиотека процедур, обеспечивающая манипуляции со строковыми данными и позволяющая выполнять расчеты и широко использовать распределения вероятности.

Система имеет встроенные средства обслуживания, которые поддерживают режим захвата и печати графических окон для выполнения моментальных снимков (Snapshot). Мощный встроенный текстовый редактор предназначен для создания, оперативного изменения, редактирования самых разнообразных имитационных моделей, и имеется большой набор команд для управления процессом моделирования.

Использование системы моделирования GPSS World не только значительно ускоряет процесс моделирования и исследования самых разнообразных систем массового обслуживания и непрерывных процессов, но и позволяет проводить оптимизационные эксперименты.

Книга состоит из пяти глав.

Глава 1 знакомит читателей с интегрированной средой программирования GPSSW. В ней рассмотрена структура среды и правила запуска программы, приведено описание системы меню, панели инструментов и других элементов системы GPSSW.

В главе 2 даются основные понятия и определения языка имитационного моделирования – GPSS, рассказывается об этапах программирования. Здесь же описан процесс разработки программ на языке GPSS в среде GPSSW.

В главе 3 проводится моделирование работы непроизводственных систем: магазина, переговорного пункта, пешеходного перехода, супермаркета, а также систем «Хищник–добыча» и развития эпидемии.

В главе 4 описывается моделирование работы производственных систем: участка цеха, автозаправочной станции, инструментальной кладовой, систем управления качеством и запасом.

Глава 5 посвящена моделированию, пожалуй, самых распространенных систем – систем массового обслуживания. Рассматриваются одноканальные и многоканальные системы, замкнутые и разомкнутые, многофазные с различными потоками требований, функционирующих в системе.

Книга содержит большое количество рисунков, иллюстрирующих интегрированную среду моделирования GPSS World, программирование на языке GPSS и решение конкретных задач.

Автор далек от мысли, что данное издание лишено недостатков, не совсем уверен в абсолютной правильности перевода некоторых действий и с благодарностью примет все замечания и предложения по улучшению книги, которые просит направлять по адресам: editor-in-chief@dmkpress.ru, kem1@mail.ru.

Интегрированная среда GPSS World

Структура и запуск GPSS World	14
Главное меню GPSS	15
Меню File	16
Меню Edit	22
Меню Search	24
Меню View	25
Меню Command	25
Меню Window	26
Меню Help	27
Панель инструментов GPSSW	28
Окно исходной модели	29
Ввод и редактирование текста	35
Настройка установок	40

Глава посвящена описанию интегрированной среды программирования GPSS World. Для краткости мы будем называть ее GPSSW. В главе рассматривается структура среды, правила запуска системы GPSSW. Приводятся сведения о системе меню, панели инструментов, текстовом редакторе и др.

Структура и запуск GPSS World

GPSS World – это общецелевая система моделирования, значительно облегчающая процесс создания и выполнения программ имитационного моделирования.

В книге рассматривается студенческая версия системы GPSS World, которую можно бесплатно скачать с сайта фирмы Minuteman Software по адресу: www.minutemansoftware/download.

Студенческая версия системы GPSS World так же эффективна, как и коммерческая, и на сегодняшних персональных компьютерах выполняется в тысячу раз быстрее, чем работала оригинальная версия GPSS/PC в 1984 году.

В процессе инсталляции система GPSSW по умолчанию устанавливается в каталог C:\Program Files\Minuteman Software\GPSS World Student Version.

Для системы GPSSW требуется IBM-совместимый компьютер с ОС Windows 95, 98 и выше. Целесообразно использовать процессоры типа Pentium II и выше. Желательно иметь мышь. Требуется, по меньшей мере, 32 Мбайт оперативной памяти и 10 Мбайт свободного пространства на жестком диске (винчестере).

Запуск системы GPSSW можно выполнить несколькими способами.

Первый способ: дважды щелкните мышью по файлу GPSSW.exe в каталоге, в котором была установлена система. Появится главное окно системы GPSS World.

Второй способ:

- щелкните мышью по кнопке **Пуск** ОС Windows. Появится всплывающее меню;
- щелкните по пункту **Программы** во всплывающем меню. Появится следующее всплывающее меню;
- щелкните по пункту **GPSS World Student Version**. Откроется главное окно системы GPSSW (рис. 1.1).

В первой строке (строке заголовка) главного окна указано название окна – **GPSS World**. Во второй строке располагаются пункты главного меню, в третьей – стандартная панель инструментов. Нижняя строка главного окна – строка состояния системы, в которой дается краткое описание выделенной команды.

Система GPSSW имеет иерархическую систему меню, состоящую из главного меню, систем выпадающих и всплывающих меню (подменю).

Главное меню служит для вызова выпадающих меню. Набор пунктов главного меню располагается во второй строке главного окна системы GPSS.

Выпадающее меню предназначено для вызова всплывающего меню, диалогового окна или соответствующей команды. Выпадающее меню располагается под соответствующим пунктом главного меню.

Всплывающее меню позволяет вызывать другое всплывающее меню, диалоговое окно или соответствующую команду. Всплывающее меню располагается справа от выбранного пункта выпадающего меню. Для выбора пункта всплывающего меню достаточно щелкнуть по нему левой кнопкой мыши.

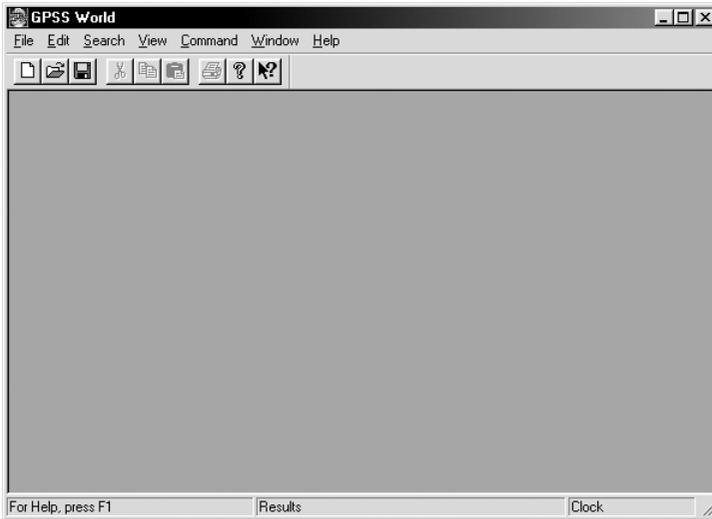


Рис. 1.1. Главное окно системы GPSS World

Главное меню GPSS

Главное меню обеспечивает доступ ко всем средствам системы GPSSW. По своей сути главное меню является основным управляющим центром этой системы. Роль дополнительного центра играет стандартная панель инструментов, включающая ряд кнопок, за которыми закреплены наиболее часто используемые команды.

Система меню GPSSW – это хорошо скоординированная совокупность выпадающих и всплывающих меню. После щелчка мышью по любому пункту главного меню или нажатия на «горячие» клавиши (HotKeys) на экране появляется соответствующее выпадающее меню. «Горячая» клавиша выделяется в названии пункта меню путем подчеркивания одной буквы. Чтобы с помощью клавиатуры получить быстрый доступ к пункту главного меню, а значит и к нужному выпадающему меню, необходимо нажать на клавишу **Alt** и, удерживая ее, на ту алфавитную клавишу, название которой подчеркнуто. Например, для быстрого обращения к пункту **View** (Вид) достаточно нажать комбинацию клавиш **Alt+V**. Вызов соседнего выпадающего меню осуществляется с помощью клавиши со стрелкой влево или вправо. Выделить любой пункт выпадающего или всплывающего меню (подменю) можно путем наведения на него указателя мыши или с помощью клавиш перемещения со стрелкой вниз либо вверх.

Для выполнения выделенного пункта меню существует несколько способов:

- нажать на клавишу **Enter**;
- нажать на «горячую» клавишу, название которой подчеркнуто;
- щелкнуть мышью по выделенному пункту.

Многие пункты главного меню заблокированы и обесцвечены серым тоном до тех пор, пока вы не откроете и не создадите соответствующие объекты системы GPSSW.

Чтобы закрыть выпадающее или всплывающее меню, достаточно нажать на клавишу **Esc** или щелкнуть мышью по полю окна вне меню.

Выпадающее меню – это меню, появляющееся после щелчка по любому пункту главного меню. Оно, в свою очередь, состоит из пунктов подменю, или команд.

Пункт меню, после названия которого стоит многоточие, – диалоговая команда, и при ее выборе появляется диалоговое окно.

Если в правой части пункта имеется треугольная стрелка, то при выборе такого пункта открывается подменю (всплывающее меню).

В том случае, если яркость пункта понижена, этот пункт в данный момент не доступен для использования.

Флажок (галочка), предшествующий пункту или появляющийся при его выборе, свидетельствует о том, что пункт может находиться во включенном или выключенном состоянии: он считается включенным при наличии флажка слева и выключенным – при отсутствии.

Справа от названия пункта меню часто указываются клавиши-акселераторы (shortcut keys, acceleration keys), предназначенные для оперативного доступа к этому пункту или команде.

Далее рассмотрим выпадающие меню для каждого пункта главного меню.

Меню File

Состав и структура меню

Пункт **File** (Файл) главного меню служит для работы с файлами документов.

Файл – это именованная упорядоченная совокупность данных или кодов программ на диске, имеющая имя и расширение. Файлы имитационных моделей в системе GPSSW записываются в окне **Model** (Модель) и сохраняются с расширением **.gps**, которое указывается сразу после имени файла. Текстовые файлы системы GPSSW записываются в окне **Text File** (Текстовый файл) и сохраняются с расширением **.txt**. Они имеют текстовый формат, и их легко прочитать и модифицировать при помощи любого текстового редактора.

Файлы могут содержать и результаты проведенного моделирования. Эти файлы создаются после сохранения содержимого окна **REPORT** (Отчет). При этом файл будет иметь расширение **.grg**. Кроме того, можно сохранить сообщения, появляющиеся в процессе моделирования систем. Эти сообщения, выводимые в окне **JOURNAL** (Журнал), можно сохранить в файле с расширением **.sim**.

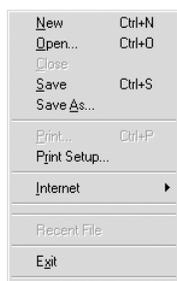


Рис. 1.2. Выпадающее меню пункта **File** главного меню

Выбор пункта **File** главного меню или нажатие комбинации клавиш **Alt+F** вызывает выпадающее меню работы с файлами, показанное на рис. 1.2.

Выпадающее меню пункта **File** главного меню включает большой набор пунктов и соответствующих «горячих» клавиш:

- **New** (Создать) – **Ctrl+N**;
- **Open...** (Открыть) – **Ctrl+O**;
- **Close** (Заккрыть);
- **Save** (Сохранить) – **Ctrl+S**;
- **Save As...** (Сохранить как);
- **Print...** (Печать) – **Ctrl+P**;
- **Internet**;
- **Recent File** (Последний файл);
- **Exit** (Выйти).

Создание нового файла

При выборе пункта **New** или нажатии комбинации клавиш **Ctrl+N** обеспечивается вызов диалогового окна **Новый документ**. У пункта **New** есть на стандартной панели инструментов дублирующая пиктограмма с изображением чистого листа с загнутым правым верхним уголком. Используя диалоговое окно **Новый документ**, можно создать новый файл для моделирования с помощью пункта **Model** с расширением .gpr и текстовый файл с помощью пункта **Text File** с расширением .txt. После двойного щелчка мышью по требуемому типу файла или щелчка по кнопке **OK** при выделенном нужном файле появится соответствующее окно: для ввода моделируемой системы – **Untitled Model 1** (Без названия модель 1) – или для создания текстового файла – **Untitled Text File 1** (Без названия текстовый файл 1).

В качестве примера введем в окне **Untitled Model 1** модель одноканальной разомкнутой системы массового обслуживания (СМО). На рис. 1.3 представлена модель простейшей системы массового обслуживания, в которой выделены основные события.

Охарактеризуем каждое событие, возникшее в СМО:

1. Появление требования в системе (**GENERATE** – Генерировать).
2. Вход требования в очередь (**QUEUE** – Очередь).
3. Определение занятости канала обслуживания (**SEIZE** – Занять). Если канал занят, то требование остается в очереди, если свободен – то входит в канал обслуживания.
4. Выход требования из очереди (**DEPART** – Выйти).
5. Обслуживание требования в канале обслуживания (**ADVANCE** – Задержать).
6. Освобождение канала обслуживания (**RELEASE** – Освободить).
7. Выход требования из системы (**TERMINATE** – Завершить).

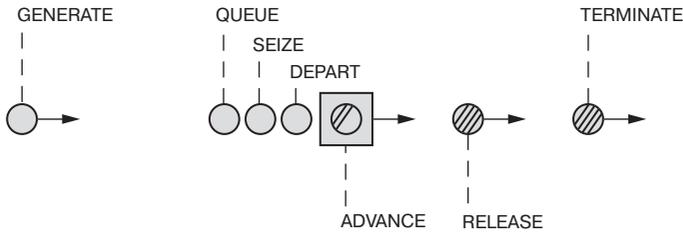


Рис. 1.3. Простейшая система массового обслуживания

Эта программа в системе GPSSW будет выглядеть так, как показано на рис. 1.4. Используемые в программе операторы полностью соответствуют содержанию и логике моделируемой системы. Справа от операторов пишутся параметры (признаки, переменные), которые характеризуют данное событие. Так, в операторе **GENERATE** первая цифра – 7 – определяет средний интервал времени между поступлениями требований в систему на обслуживание, а вторая – 2 – максимально допустимое отклонение от среднего времени. В операторах **QUEUE** и **DEPART** цифра 1 определяет номер очереди, в которую вошло и из которой собирается выйти требование. В операторах **SEIZE** и **RELEASE** символы KAN определяют символическое имя канала обслуживания, в который собирается войти требование, если он освободился, и выйти – если требование уже в нем обслужилось. В операторе **ADVANCE** первая цифра – 6 – определяет среднее время обслуживания требования, а вторая – 3 – максимально допустимое отклонение от этого времени. Оператор **TERMINATE** выполняет удаление одного требования из системы. Цифра 200 в операторе **START** означает число требований, которые необходимо пропустить через систему.

Untitled Model 1		
GENERATE	7,2	; Генерирует вход требования в систему с интервалом [5 - 9] ед. времени.
QUEUE	1	; Увеличивает содержимое очереди под номером 1 на одно требование.
SEIZE	KAN	; Определяет занятость канала по имени KAN.
DEPART	1	; Уменьшает содержимое очереди под номером 1 на одно требование.
ADVANCE	6,3	; Обслуживает требование в канале обслуживания - [3 - 9] ед. времени.
RELEASE	KAN	; Освобождает канал обслуживания по имени KAN.
TERMINATE	1	; Выход требования из системы.
START	200	; Начало моделирования с числом требований - 200.

Рис. 1.4. Окно **Untitled Model 1** с введенной в него моделью одноканальной разомкнутой СМО

В нашем простейшем примере требования поступают на обслуживание в систему случайно в интервале [5–9] единиц времени с равномерным распределением. А время обслуживания колеблется в интервале [3–9] единиц времени, также

с равномерным распределением. При этом моделируется прохождение через систему 200 требований.

В системе моделирования GPSSW предусмотрен стандартный отчет, в котором выводятся результаты моделирования.

Стандартный отчет включает в себя результаты моделирования системы, например такие, как:

- коэффициент использования канала обслуживания;
- средняя длина очереди;
- среднее время пребывания требования в очереди и ряд других.

Чтобы получить стандартный отчет для модели при наличии в ней управляющего оператора **START**, необходимо:

- щелкнуть по кнопке **Command** (Команда) главного меню. Появится выпадающее меню;
- щелкнуть по кнопке **Create Simulation** (Создать выполняемую модель). Появится окно **JOURNAL**, а затем – **REPORT** с результатами моделирования; фрагмент последнего показан на рис. 1.5.

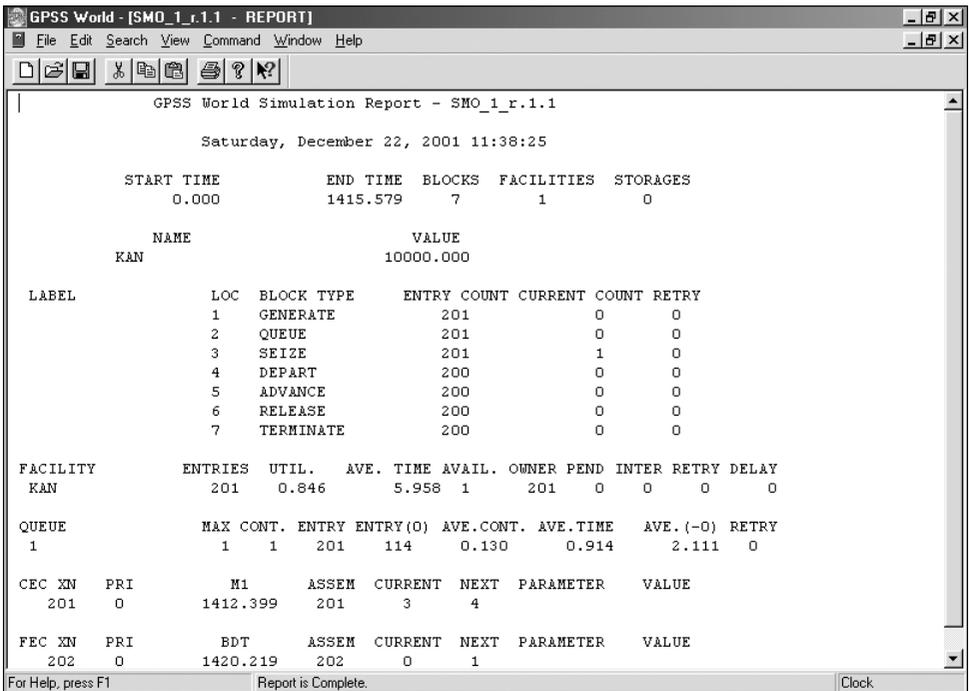


Рис. 1.5. Стандартный вывод результатов моделирования

В стандартный отчет включаются следующие основные показатели моделирования системы:

- время моделирования системы – **END TIME** (Время окончания);
- число обслуженных требований в канале обслуживания – **ENTRIES** (Число входов);
- коэффициент использования канала обслуживания – **UTIL.** (Использование);
- среднее время обслуживания требования в канале – **AVE. TIME** (Среднее время);
- максимальная длина очереди – **MAX** (Максимальная);
- средняя длина (содержимое) очереди – **AVE.CONT.** (Среднее содержимое);
- среднее время пребывания требования в очереди – **AVE. TIME** – и ряд других показателей.

Для разработки модели функционирования этой, казалось бы, очень простой системы массового обслуживания с использованием наиболее распространенных языков программирования BASIC, Pascal, C или FORTRAN потребовалось бы на порядок большее число операторов и высокий уровень программиста. Решение же такого типа задач в системе GPSSW доступно каждому пользователю после небольшой подготовки.

Открытие существующего файла

Выбор пункта выпадающего меню **Open...** или нажатие комбинации клавиш **Ctrl+O** вызывает стандартное диалоговое окно открытия файла **Open**. Достаточно найти нужный каталог (папку), файл, а затем, щелкнув дважды левой кнопкой мыши по нужному файлу, вы откроете его содержимое. У пункта **Open...** есть на стандартной панели инструментов дублирующая пиктограмма с изображением открытой папки.

Заккрытие файла

Пункт выпадающего меню **Close** закрывает активное окно, то есть окно текущей программы (файла), и система переходит к предыдущему окну (либо к пустому первому, если другие окна не выводились). Если какая-либо из моделей в закрываемых окнах подвергалась модификации, то при закрытии файла появится диалоговое окно сообщений GPSS World. В этом диалоговом окне система спрашивает, надо ли сохранять изменения в файле. Если документ не модифицировался, соответствующее окно просто закрывается.

Сохранение файла в текущем каталоге

Выбор пункта выпадающего меню **Save** или нажатие комбинации клавиш **Ctrl+S** сохраняет файл под тем же именем. Допустим, мы хотим сохранить только что созданный текстовый файл. Тогда появится стандартное диалоговое окно **Save As**.

Поскольку содержимое окна **Text File** сохраняется в файле с расширением .txt, то в диалоговом окне **Save As** появится список текстовых файлов, если они есть, имеющих расширение .txt. То есть тип файла определяется системой автоматически. Если сохраняется файл, который уже имеет имя, то соответствующее диалоговое окно не вызывается.

Пунктом **Save** рекомендуется пользоваться периодически при подготовке сложных моделей. Это позволяет сохранить все изменения в документах и избежать потери данных, например, при случайном выключении компьютера до записи файла. У пункта **Save** есть на стандартной панели инструментов дублирующая пиктограмма с изображением дискеты.

Сохранение файла в другом каталоге

Пункт выпадающего меню **Save As...** вызывает диалоговое окно **Save As**, с помощью которого можно сохранить файл под новым именем и/или в новом месте, в другом формате. Допустим, мы хотим сохранить файл результатов моделирования – содержимое окна **REPORT**. Появится диалоговое окно **Save As**. Поскольку содержимое окна **REPORT** сохраняется в файле с расширением .gpr, то в диалоговом окне **Save As** появится список ранее сохраненных файлов с расширением .gpr.

Установка параметров принтера и печати

Выбор пункта выпадающего меню **Print...** или нажатие комбинации клавиш **Ctrl+P** открывает диалоговое окно **Print** для ввода имени принтера, числа копий и ряда других установок.

У пункта **Print...** на стандартной панели инструментов имеется дублирующая пиктограмма с изображением принтера. Диалоговое окно **Print** представляет собой стандартное окно печати Windows. Оно позволяет устанавливать параметры принтера.

Настройка принтера и установка параметров бумаги

Пункт выпадающего меню **Print Setup...** (Настройка принтера) вызывает диалоговое окно настройки принтера **Print Setup** для указания размера бумаги и ряда других установок.

Связь с Internet

Пункт выпадающего меню **Internet** вызывает всплывающее меню, показанное на рис. 1.6.

Всплывающее меню включает два пункта:

- **Download Notices** (Загрузка объявлений);
- **GPSS Web Page...** (Web-страница GPSS) вызывает диалоговое окно с общей информацией от фирмы по системе GPSS.

Если в процессе использования системы GPSSW открывалось несколько файлов, то они будут представлены в выпадающем меню пункта **File** главного меню после пункта **Internet**.

Download Notices
GPSS Web Page...

Рис. 1.6. Всплывающее меню пункта **Internet** выпадающего меню пункта **File**

Выход из системы GPSSW

Пункт выпадающего меню **Exit** обеспечивает выход из системы GPSS. Перед выходом из системы необходимо, чтобы все загруженные ранее окна, содержимое которых подверглось редактированию и модификации, были сохранены на диске с помощью команды **Save As...** или **Save**. Тогда при выборе пункта **Exit** можно наблюдать последовательное исчезновение окон. Если пользователь забыл сохранить содержимое окна на диск, система сообщит об этом, выдав запрос. Нужно ответить **Yes** (Да), если документ нужно сохранить, **No** (Нет) – если сохранения не требуется, и **Cancel** (Отменить) – для отмены закрытия окна.

Меню Edit

Выбор пункта **Edit** (Правка) главного меню или нажатие комбинации клавиш **Alt+E** вызывает выпадающее меню редактирования (рис. 1.7).

Для выполнения того или иного действия можно воспользоваться следующими командами из этого меню:

- **Undo** (Отменить) или нажатие комбинации клавиш **Ctrl+Z** отменяет последнюю из выполненных команд;
- **Cut** (Вырезать) или нажатие комбинации клавиш **Ctrl+X** удаляет выделенный фрагмент текста и помещает его в буфер обмена данными;
- **Copy** (Копировать) или нажатие комбинации клавиш **Ctrl+C** копирует выделенный фрагмент текста и помещает его в буфер обмена данными;
- **Paste** (Вставить) или нажатие комбинации клавиш **Ctrl+V** вставляет содержимое буфера обмена данными в текущую позицию курсора в окне текстового редактора;
- **Insert Line** (Вставить строку) или нажатие комбинации клавиш **Ctrl+I** вставляет строку в программу;
- **Delete Line** (Удалить строку) или нажатие комбинации клавиш **Ctrl+D** удаляет выделенную строку программы;
- **Font ...** (Шрифт) вызывает диалоговое окно для выбора шрифта;
- **Expression Window ...** (Окно выражения) вызывает диалоговое окно **Edit Expression Window** (Окно редактирования выражения);
- **Plot Window ...** (Окно графика) вызывает диалоговое окно **Edit Plot Window** (Окно редактирования графика);

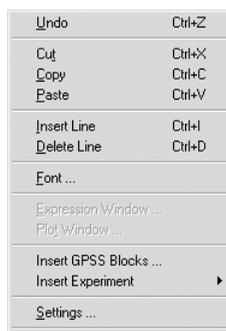


Рис. 1.7
Выпадающее меню пункта **Edit** главного меню

- **Insert GPSS Blocks ...** (Вставить блоки GPSS) вызывает диалоговое окно, в котором можно выбрать щелчком мыши нужный блок GPSS (рис. 1.8);
- **Insert Experiment** (Вставить эксперимент) вызывает всплывающее меню для выбора соответствующего эксперимента (рис. 1.9).



Рис. 1.8. Диалоговое окно **Insert GPSS Block into Model Object**



Рис. 1.9. Всплывающее меню пункта **Insert Experiment**

Всплывающее меню включает два пункта:

- **Screening ...** (Экранный) вызывает диалоговое окно **Screening Experiment Generator**;
- **Optimizing ...** (Оптимизация) вызывает диалоговое окно **Optimizing Experiment Generator**;
- **Settings ...** (Установки) вызывает диалоговое окно **SETTINGS**, в котором можно определить те или иные установки системы (рис. 1.10).

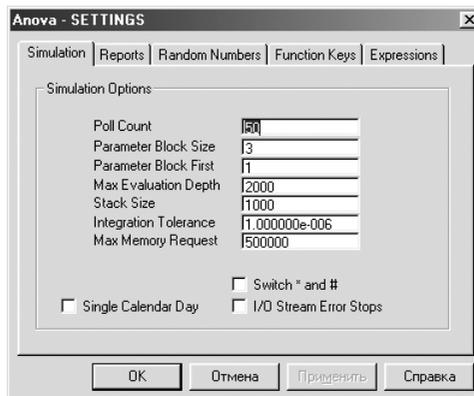


Рис. 1.10. Диалоговое окно **SETTINGS** с открытой вкладкой **Simulation** для файла под именем **Анова**

Диалоговое окно **SETTINGS**, в свою очередь, включает пять вкладок для обеспечения соответствующих установок:

- **Simulation** (Моделирование);
- **Reports** (Отчеты);
- **Random Numbers** (Случайные числа);
- **Function Keys** (Функциональные клавиши);
- **Expressions** (Выражения).

Вставка блоков GPSS в модель

В диалоговое окно **Insert GPSS Block into Model Object** (Вставить блок GPSS в модель) входит пятьдесят три блока. Если щелкнуть по любому из них, появится соответствующее диалоговое окно. Допустим, что вы щелкнули по блоку **GENERATE**. Шаблон блока **GENERATE** появится в диалоговом окне **Enter Block Information** (Информация для ввода в блок), показанном на рис. 1.11, для ввода необходимой информации.

Enter Block Information

GENERATE

GENERATE - Create XN for future entry.

A: 9 Intergeneration time.

B: 1 Halfrange or Function Modifier.

C: Start delay time.

D: Creation limit.

E: Priority.

F:

G:

H:

Label:

Comment: Вход покупателей в магазин

OK Cancel Help

Рис. 1.11. Диалоговое окно **Enter Block Information** с шаблоном блока **GENERATE**

Меню Search

Выбор пункта **Search** (Поиск) главного меню или нажатие комбинации клавиш **Alt+S** открывает выпадающее меню, представленное на рис. 1.12.

Меню содержит следующие пункты:

- **Find / Replace** (Найти / Заменить) или нажатие комбинации клавиш **Ctrl+Alt+F** выводит диалоговое окно **Find / Replace**;
- **Go to Line ...** (Перейти к строке) или нажатие комбинации клавиш **Ctrl+Alt+G** раскрывает диалоговое окно **Enter Line Number** (Введите номер строки) для перехода к указанной строке;
- **Next Bookmark** (Следующая закладка) или нажатие комбинации клавиш **Ctrl+Alt+B** обеспечивает переход к следующей закладке;
- **Mark** (Установить метку) устанавливает невидимую метку у объекта;
- **UnMark** (Удалить метку) удаляет невидимую метку у объекта;
- **UnMark All** (Удалить все метки) удаляет все метки;
- **Select to Bookmark** (Выделить до отметки) выделяет текст от положения курсора до ближайшей закладки;
- **Next Error** (Следующая ошибка) или нажатие комбинации клавиш **Ctrl+Alt+N** перемещает курсор на следующую ошибку;
- **Previous Error** (Предыдущая ошибка) или нажатие комбинации клавиш **Ctrl+Alt+P** перемещает курсор на предыдущую ошибку.

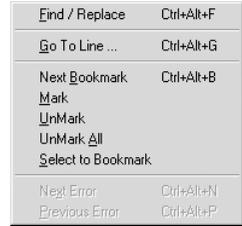


Рис. 1.12
Выпадающее меню пункта **Search** главного меню

Меню View

Выбор пункта **View** главного меню или нажатие комбинации клавиш **Alt+V** вызывает выпадающее меню, представленное на рис. 1.13.

Меню содержит следующие пункты, позволяющие открывать различные окна, панели или меню:

- **Notices** (Сообщения) вызывает окно **Notices**;
- **Toolbar** (Панель инструментов) устанавливает или удаляет в главном окне системы стандартную панель инструментов;
- **Entity Details** (Детальное представление элемента) представляет подробности;
- **Simulation Clock** (Часы моделирования).

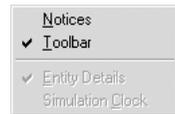


Рис. 1.13
Выпадающее меню пункта **View** главного меню

Меню Command

Выбор пункта **Command** главного меню или нажатие комбинации клавиш **Alt+C** вызывает выпадающее меню команд, изображенное на рис. 1.14.

Меню предлагает следующие пункты:

- **Create Simulation** (Создать выполняемую модель) или нажатие комбинации клавиш **Ctrl+Alt+S** дает команду на вызов транслятора и выполнение трансляции исходной модели с фиксацией даты и времени начала и окончания трансляции;
- **Retranslate** (Перетранслировать) или нажатие комбинации клавиш **Ctrl+Alt+R** обеспечивает перетранслирование модели;
- **Repeat Last Command** (Повторить последнюю команду) или нажатие комбинации клавиш **Ctrl+Alt+L** обеспечивает повторение выполнения последней команды;
- **CONDUCT** (Управление) дает возможность проведения эксперимента;
- **START** (Пуск) обеспечивает запуск оттранслированной программы на выполнение с помощью диалогового окна **Start Command** (Выполнить команду);
- **STEP1** (Шаг 1) или нажатие комбинации клавиш **Ctrl+Alt+1** обеспечивает пошаговое выполнение оттранслированной программы;
- **HALT** (Останов) прерывает процесс моделирования;
- **CONTINUE** (Продолжить) обеспечивает продолжение процесса моделирования;
- **CLEAR** (Очистить) – возвращение моделирования в первоначальное состояние;
- **RESET** (Сброс) – сброс статистики в начальное состояние;
- **SHOW ...** (Показать) вызывает диалоговое окно **Show Command** (Показать команду), которое обеспечивает возможность просмотра искомых параметров в окне **JOURNAL**;
- **Custom ...** (Пользователь) вызывает диалоговое окно **Simulation Command** (Команда моделирования) для ввода команд управления пользователем во время моделирования.

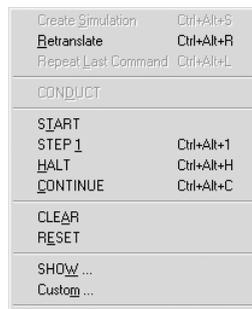


Рис. 1.14. Выпадающее меню пункта **Command** главного меню

Меню Window

Система GPSSW позволяет эффективно работать с несколькими моделями. Под каждую модель отводится отдельное окно. Одно из них обычно является активным – то, с которым пользователь работает в настоящий момент. Состояние других окон «замораживается», но их можно активизировать в любое время без новой загрузки и без исполнения связанной с окном программы.

Система GPSS позволяет организовать работу с несколькими окнами одновременно. Выбор пункта **Window** (Окно) главного меню или нажатие комбинации

клавиш **Alt+W** вызывает выпадающее меню управления работой с несколькими окнами, представленное на рис. 1.15.

Меню предлагает следующие пункты:

- **Cascade** (Каскад) позволяет расположить окна одно за другим каскадом так, что видны их заголовки; активное окно оказывается поверх остальных;
- **Tile** (Мозаика) располагает окна одно под другим сверху вниз. Всем окнам отводятся равные части Рабочего стола;
- **Simulation Window** (Окно моделирования) вызывает всплывающее меню (рис. 1.16), которое включает ряд пунктов:
 - **Blocks Window** (Окно блоков);
 - **Expressions Window** (Окно выражений);
 - **Facilities Window** (Окно каналов обслуживания);
 - **Logicswitches Window** (Окно логических переключателей);
 - **Matrix Window ...** (Окно матрицы);
 - **Plot Window ...** (Окно гистограммы);
 - **Queues Window** (Окно очередей);
 - **Savevalues Window** (Окно сохраняемых величин);
 - **Storages Window** (Окно накопителей);
 - **Table Window** (Окно таблицы);
- **Simulation Snapshot** (Снимок моделирования) вызывает всплывающее меню со списком окон различных снимков моделирования и окон, открытых в данный момент.



Рис. 1.15. Выпадающее меню пункта **Window** главного меню

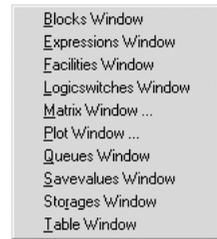


Рис. 1.16. Всплывающее меню пункта **Simulation Window** выпадающего меню

Меню Help

Выбор пункта **Help** (Справка) главного меню или нажатие комбинации клавиш **Alt+N** открывает выпадающее меню справочной системы (рис. 1.17).

Меню содержит следующие пункты:

- **Help Topics** (Вызов справки) или нажатие клавиши **F1** вызывает справочную систему GPSSW Application Help;
- **About GPSS World ...** (О системе GPSS World) обеспечивает дополнительной информацией по GPSSW.

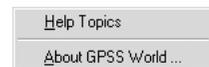


Рис. 1.17. Выпадающее меню пункта **Help** главного меню

Панель инструментов GPSSW

Для эффективной работы со средой GPSSW удобно иметь на экране панель инструментов. Открыть или закрыть стандартную панель инструментов (рис. 1.18) можно с помощью пункта **Toolbar** (Панель инструментов) выпадающего меню **View**.



Рис. 1.18. Стандартная панель инструментов

При перемещении указателя мыши по значкам (пиктограммам) панели инструментов появляется подсказка о назначении каждой кнопки панели, а в строке состояния в нижней части главного окна GPSSW представлено более подробное пояснение. С любой из кнопок связан определенный пункт меню, а рисунок на кнопке помогает ее идентифицировать.

На стандартной панели инструментов размещены следующие девять кнопок:

- **New file** (Новый файл) с изображением листа бумаги – открывает очередное новое окно текстового редактора с номером на единицу больше предыдущего (например, <Untitled-1>);
- **Open file** (Открыть файл) с изображением открытой папки – вызывает диалоговое окно **Open file to edit/view** (Открыть файл для редактирования и просмотра), позволяющее вызывать существующий файл;
- **Save file** (Сохранить файл) с изображением дискеты – открывает диалоговое окно **Save As** для сохранения файла под определенным именем на нужном диске и в нужном каталоге, если выполняется первое сохранение. При повторных щелчках по этой кнопке (свою работу стоит периодически сохранять во избежание потерь информации) открывается диалоговое окно **Save**;
- **Cut** с изображением ножниц – удаляет выделенный фрагмент текста в буфер обмена данными;
- **Copy** с изображением двух листов бумаги – копирует выделенный фрагмент текста или программы в буфер обмена данными;
- **Paste** с изображением портфеля – вставляет фрагмент текста из буфера обмена данными;
- **Print** с изображением принтера – выводит диалоговое окно **Print**, в котором указываются имя принтера, режим печати, число копий и другие параметры печати;
- **About** (О системе) с изображением знака вопроса – вызывает окно с информацией о системе GPSS World;
- **Help** с изображением стрелки и знака вопроса – изменяет вид указателя мыши на знак вопроса, которым можно выделить объект для получения по нему справки.

Окно исходной модели

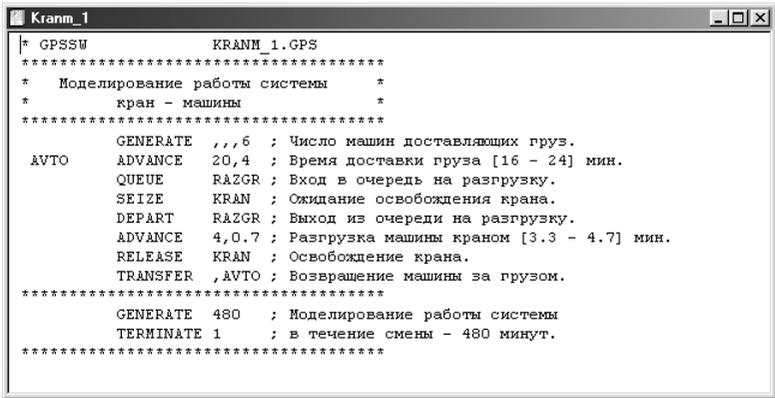
Окно исходной модели системы GPSSW предназначено для эффективной разработки, проверки и отладки программ в GPSSW. Это окно вызывается автоматически при открытии файла с программой на языке GPSS в диалоговом окне **Open**.

Вывести на экран диалоговое окно открытия файла можно тремя способами:

- выбрать пункт **Open...** выпадающего меню пункта **File**;
- нажать комбинацию клавиш **Ctrl+O**;
- щелкнуть по кнопке с изображением открытой папки на стандартной панели инструментов.

В диалоговом окне открытия файла необходимо найти соответствующую папку (каталог) и дважды щелкнуть по ней мышью. В раскрытой папке (каталоге) аналогичным способом следует отметить нужный файл, который будет помещен в текстовое окно системы GPSS.

В качестве примера рассмотрим файл KRAMM_1.GPS, который содержит программу, написанную на языке имитационного моделирования GPSS в системе GPSSW (рис. 1.19).



```

Kranm_1
|# GPSSW          KRANM_1.GPS
*****
*   Моделирование работы системы   *
*   кран - машины                   *
*****
      GENERATE   ,, , 6 ; Число машин доставляющих груз.
AVTO  ADVANCE   20,4 ; Время доставки груза [16 - 24] мин.
      QUEUE     RAZGR ; Вход в очередь на разгрузку.
      SEIZE     KRAN ; Ожидание освобождения крана.
      DEPART    RAZGR ; Выход из очереди на разгрузку.
      ADVANCE   4,0,7 ; Разгрузка машины краном [3.3 - 4.7] мин.
      RELEASE   KRAN ; Освобождение крана.
      TRANSFER  , AVTO ; Возвращение машины за грузом.
*****
      GENERATE  480 ; Моделирование работы системы
      TERMINATE 1 ; в течение смены - 480 минут.
*****

```

Рис. 1.19. Окно исходной модели

Следует помнить, что в каждом окне модели можно работать только с одним файлом – одной программой. Каждый раз при открытии файла система GPSSW отображает его в новом окне модели.

Программу, находящуюся в окне модели, можно оттранслировать и получить искомый результат. Для этого важно, чтобы окно модели было активным. Если вы

не уверены в этом, щелкните мышью в любом месте окна и сделайте его активным. Чтобы загрузить программу на транслирование и выполнение:

- щелкните мышью по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните мышью по пункту **Create Simulation ...** (Создать выполняемую модель) выпадающего меню. Если программа без ошибок и в ней присутствует управляющая команда (оператор) **START**, которую не сопровождают символы **NP** (Not Print – Не печатать), то результаты моделирования появятся в окне **REPORT**.

Можно также нажать комбинацию клавиш **Ctrl+Alt+S**, которая дает команду на вызов транслятора и выполнение процесса трансляции исходной модели. Имитационное моделирование в программе **KRANM_1.GPS** не выполняется, так как в ней нет управляющей команды **START**. Но в результате вышеуказанных действий появится окно **JOURNAL** с сообщением даты и времени начала и окончания проведения трансляции, например такое, как на рис. 1.20.

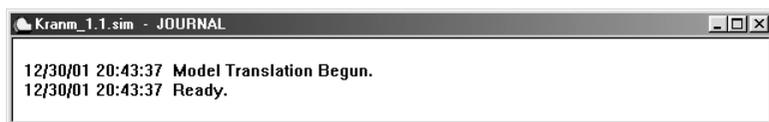


Рис. 1.20. Окно **JOURNAL**

Из сообщения, приведенного на рис. 1.20, следует, что началась трансляция исходной модели (Model Translation Begun – Начало трансляции модели) 30 декабря 2001 года в 20 ч 43 мин 37 с и закончилась (Ready – Готово) практически в это же время.

Затем оттранслированная программа может быть запущена на выполнение. Для этого:

- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **Start** выпадающего меню. Появится диалоговое окно **Start Command**, показанное на рис. 1.21.

После щелчка по кнопке **OK** появятся результаты моделирования.

На рис. 1.22 показан фрагмент результатов моделирования задачи **KRANM_1.GPS** в окне **REPORT**.

Название программы сразу определяет, что моделируется, – кран и машины, с которыми он работает. Окно **JOURNAL** служит также и для отображения диагностических сообщений об ошибках в программах.



Рис. 1.21. Диалоговое окно **Start Command**

GPSS World Simulation Report - Kranm_1.1.1

Sunday, December 30, 2001 20:52:49

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	480.000	10	1	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
KRAN	107	0.885	3.970	1	1	0	0	0	0

QUEUE	MAX CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY
RAZGR	4	0	107	22	0.490	2.197	2.765

Рис. 1.22. Фрагмент результатов моделирования задачи *KRANM_1.GPS*

Допустим, вы сделали ошибку в операторе **ADVANCE**, написав его имя неверно, например **ADANCE**. Сообщение об этом появится в окне **JOURNAL** (рис. 1.23).

```

12/30/01 21:00:26 Model Translation Begun.
12/30/01 21:00:26 Line 8, Col 6. Invalid identifier. Expecting a GPSS Verb.
12/30/01 21:00:26 AVTO ADANCE 20,4 ; Время доставки груза [16 - 24] мин.
12/30/01 21:00:26 **** Model Translation Aborted ****

```

Рис. 1.23. Окно **JOURNAL** с указанием ошибки

В окне **JOURNAL** указываются номер строки (Line 8) и позиция в строке (Col 6), где сделана ошибка. Ниже приводится некорректная строка программы:

```
AVTO ADANCE 20,4 ; Время доставки груза [16 - 24] мин.
```

Еще ниже дается сообщение:

```
**** Model Translation Aborted **** (Трансляция исходной модели прервана)
```

Чтобы быстро выйти на строку, где сделана ошибка, можно использовать пункт главного меню **Search** (Поиск). Для этого:

- щелкните по пункту **Search** главного меню. Появится выпадающее меню;
- щелкните по пункту **Go to Line** (Перейти к строке) выпадающего меню. Появится диалоговое окно **Enter Line Number** (Введите номер строки), показанное на рис. 1.24;
- введите в текстовом поле номер строки, в которой сделана ошибка.

Каждый этап моделирования сопровождается указанием даты и времени выполнения, включая и этапы диагностики ошибок.

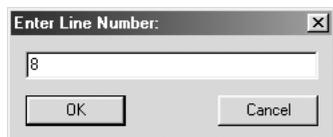


Рис. 1.24. Диалоговое окно
Enter Line Number

а затем – по пункту **Copy** (Копировать) выпадающего меню или по одноименной кнопке на стандартной панели инструментов либо нажмите комбинацию клавиш **Ctrl+C**.

Следует иметь в виду, что система GPSSW имеет широкий набор разнообразных окон:

- **Model** (Модель) – окно полноэкранный текстового редактора модели;
- **JOURNAL** (Журнал) – окно журнала для записи различных сообщений;
- **Blocks Window** (Окно блоков) – окно для интерактивного представления динамики перемещения требований по блокам;
- **Expressions Window** (Окно выражений) – окно для интерактивного представления значений выражений;
- **Facilities Window** (Окно каналов обслуживания) – окно для интерактивного представления динамики значений параметров каналов обслуживания;
- **Logicswitches Window** (Окно логических переключателей) – окно для интерактивного представления динамики логических переключателей;
- **Matrix Window** (Окно матрицы) – окно для интерактивного представления динамики значений элементов матрицы;
- **Plot Window** (Окно графика) – окно для интерактивного представления до 8 графиков выражений;
- **Queues Window** (Окно очередей) – окно для интерактивного представления динамики изменения очереди;
- **Savevalues Window** (Окно сохраняемых величин) – окно для интерактивного представления динамики изменения значений сохраняемых величин в процессе моделирования;
- **Storages Window** (Окно накопителей) – окно для интерактивного представления динамики изменения значений параметров накопителя;
- **Table** (Таблица) – окно для интерактивного представления динамики изменения значений таблицы;
- **Transaction Snapshot** (Снимок требования) – изображение состояния требования в процессе моделирования;
- **CEC Snapshot** (Снимок цепи текущих событий) – изображение состояния цепи текущих событий в процессе моделирования;
- **FEC Snapshot** (Снимок цепи будущих событий) – изображение состояния цепи будущих событий в процессе моделирования;

Система GPSSW обеспечивает возможность копирования и передачи текста программы внутри окна и между окнами, а также между окнами и любым приложением, используя буфер обмена данными.

Для этого выделите нужный текст (фрагмент программы, программу и т.д.) в окне, щелкните по пункту **Edit** (Правка) главного меню системы GPSSW,

- **Numeric Groups Snapshot** (Снимок числовых групп) – изображение состояния числовых групп в процессе моделирования;
- **Userchains Snapshot** (Снимок цепи пользователя) – изображение состояния цепи пользователя в моделировании;
- **Transaction Groups Snapshot** (Снимок групп требований) – изображение состояния групп требований в процессе моделирования.

Просмотр значений выражений в динамике

Для представления значений выражений в динамике следует использовать окно **Expressions Window**, которое можно открыть в процессе моделирования. Рассмотрим возможность представления значений выражений для созданной нами модели (см. рис. 1.19).

Предварительно модель должна быть оттранслирована. Для этого:

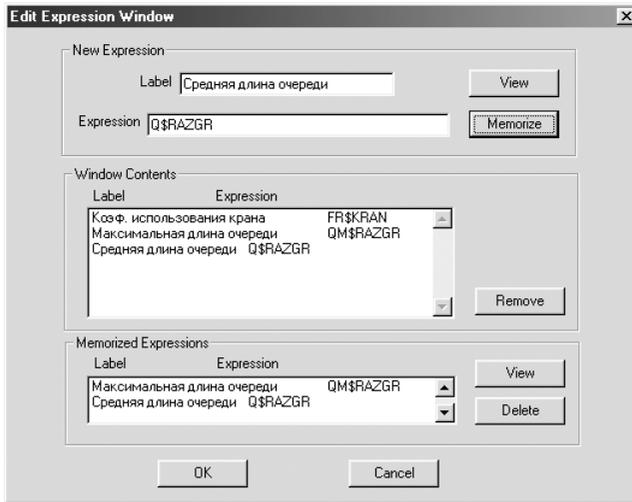
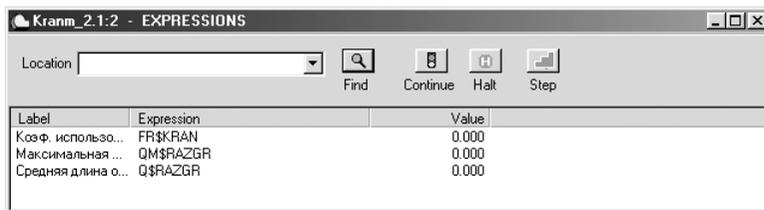
- нажмите комбинацию клавиш **Ctrl+Alt+S** или щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) в выпадающем меню. Появится окно **JOURNAL** с сообщениями о результатах трансляции.

Если нет ошибок, то активизируются пункты выпадающего меню пункта **Window** главного меню. Затем:

- щелкните по пункту **Window** главного меню или нажмите комбинацию клавиш **Alt+W**. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** (Окно моделирования) выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Expressions Window** всплывающего меню. Появятся два окна: **Edit Expression Window** (Окно редактирования выражения) и **EXPRESSIONS** (Выражения). При этом первое окно будет активизировано и наложено на второе. Введите в соответствующие текстовые поля окна **Edit Expression Window** последовательно выражения, которые вы хотели бы просмотреть в динамике. Допустим, вы хотите посмотреть выражения, используя соответствующие стандартные числовые атрибуты, например:
 - коэффициент использования крана $FR\$\$K\$RAN$;
 - максимальную длину очереди машин на разгрузку $Q\$\$R\$AZGR$;
 - среднюю длину очереди машин на разгрузку $Q\$\$R\$AZGR$.

Для этого в полях **Label** (Метка) и **Expression** (Выражение) последовательно введите нужные выражения. После ввода соответствующего названия и выражения щелкните по кнопкам **View** (Просмотр) и **Memorize** (Запомнить). Окончательно окно **Edit Expression Window** будет выглядеть так, как показано на рис. 1.25.

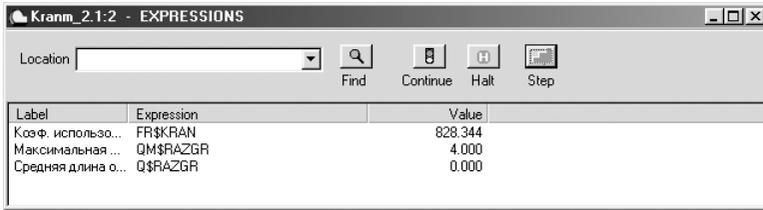
После этого щелкните по кнопке **OK**. Появится окно **EXPRESSIONS**, которое будет выглядеть так, как показано на рис. 1.26.

Рис. 1.25. Окно **Edit Expression Window** для модели *Kranm_2*Рис. 1.26. Окно **EXPRESSIONS** для модели *Kranm_2*

Теперь можно перейти к выполнению процесса моделирования. Для этого:

- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **START** в выпадающем меню. Появится диалоговое окно **Start Command**, в котором щелкните по кнопке **OK**. И вы увидите в окне **EXPRESSIONS** изменяющиеся значения заданных выражений. В любой момент вы можете прервать процесс моделирования, если щелкнете мышью по кнопке **Halt** (Остановить) – второй справа в окне **EXPRESSIONS**. Щелкнув по кнопке **Continue** (Продолжить) – второй слева в окне **EXPRESSIONS**, – можно продолжить выполнение прерванного процесса моделирования. Если же щелкнуть по кнопке **Step** (Шагнуть) после прерывания процесса моделирования, то будет выполнен только один шаг моделирования системы. В это время появятся соответствующие значения просматриваемых в динамике выражений, например такие, которые показаны на рис. 1.27.

Следует иметь в виду, что значение коэффициента использования, выдаваемое в окне **EXPRESSIONS**, нужно разделить на 1000. Таким образом, в нашей задаче



Label	Expression	Value
Коеф. использо...	FR\$K\$RAN	828.344
Максимальная ...	Q\$R\$AZGR	4.000
Средняя длина о...	Q\$R\$AZGR	0.000

Рис. 1.27. Окно **EXPRESSIONS** для модели *Kranm_2* после выполнения одного шага моделирования

коэффициент использования крана будет равен 0,828, а максимальная длина очереди машин на разгрузку – 4.

Ввод и редактирование текста

В систему GPSSW встроен текстовый редактор. Ввод или замена символов текста выполняется в месте расположения курсора, который периодически мигает, как, например, в текстовом редакторе Word. Для вставки или замены символов используется клавиша **Insert** (Вставить). С помощью клавиш перемещения – со стрелками вверх, вниз, влево, вправо, – расположенных в правой части клавиатуры, и мыши курсор можно перемещать в любое место текстового окна. Клавиша **Delete** (Удалить) удаляет символ, расположенный сразу за курсором. Клавиша **Backspace** удаляет символ, расположенный перед курсором. Клавиша **Home** (В начало строки) перемещает курсор в начало строки, в которой находится курсор, а **End** (В конец строки) – в конец текстовой строки.

Для вставки пустой строки можно использовать два способа. Нажать на клавишу **Enter**, когда курсор находится в начале строки, на месте которой должна быть введена пустая строка, или в конце строки, после которой вводится пустая строка. Вставку пустой строки можно осуществить и с помощью пунктов меню. Для этого выполните следующие действия:

- щелкните по пункту **Edit** главного меню. Появится выпадающее меню;
- щелкните по пункту **Insert Line** (Вставить строку) выпадающего меню. Будет вставлена пустая строка, и курсор переместится в ее начало.

Можно удалить из текста целый фрагмент. Для этого необходимо предварительно его выделить.

Выделение фрагмента текста

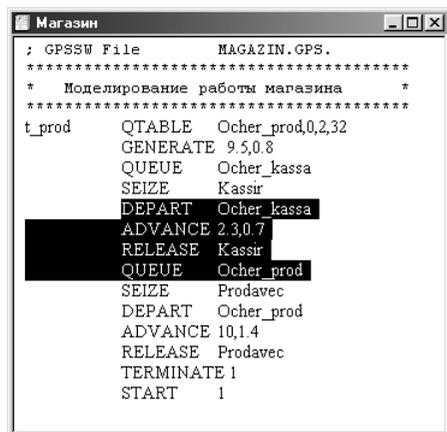
Выделить символ, слово, строку или фрагмент текста можно с помощью клавиатуры или мыши. Для выделения с помощью клавиатуры нажмите на клавишу **Shift** и, удерживая ее, нажимайте нужное число раз на те или иные клавиши перемещения со стрелками вверх, вниз, влево, вправо.

Для выделения слова достаточно щелкнуть по нему левой кнопкой мыши.

Для выделения строки следует щелкнуть перед строкой левой кнопкой мыши.

Для выделения фрагмента текста с помощью мыши нажмите левую кнопку мыши и, удерживая ее, протяните курсор по фрагменту выделяемого текста. После этого отпустите кнопку мыши.

Выделение фрагмента текста можно выполнить и с помощью выпадающего меню пункта **Search** (Поиск) главного меню. Допустим, нам надо выделить фрагмент, показанный на рис. 1.28.



```

Магазин
; GPSSW File      MAGAZIN.GPS.
*****
*   Моделирование работы магазина   *
*****
t_prod    QTABLE    Ocher_prod,0,2,32
          GENERATE  9.5,0.8
          QUEUE     Ocher_kassa
          SEIZE     Kassir
          DEPART    Ocher_kassa
          ADVANCE   2.3,0.7
          RELEASE   Kassir
          QUEUE     Ocher_prod
          SEIZE     Prodavec
          DEPART    Ocher_prod
          ADVANCE   10,1.4
          RELEASE   Prodavec
          TERMINATE 1
          START     1
  
```

Рис. 1.28. Выделение фрагмента программы с помощью меню

Для этого:

- установите курсор мыши перед оператором, до которого должен быть выделен фрагмент, В нашем примере это будет оператор

```
SEIZE     Prodavec
```

- щелкните по пункту **Search** главного меню. Появится выпадающее меню;
- щелкните по пункту **Mark** (Метка) выпадающего меню;
- установите курсор мыши перед оператором, от которого должен быть выделен фрагмент. В нашем примере это будет оператор

```
DEPART    Ocher_kassa
```

- щелкните по пункту **Select to Bookmark** (Выделить до отметки) выпадающего меню. Фрагмент будет выделен.

Удаление фрагмента текста

Для удаления выделенного фрагмента текста существует несколько способов:

- нажать на клавишу **Delete** – выделенный фрагмент будет удален;
- нажать комбинацию клавиш **Shift+Delete** или **Ctrl+X** – выделенный фрагмент будет удален, но сохранен в буфере обмена данными;

- щелкнуть по пункту **Edit** (Правка) главного меню, а затем – по пункту **Cut** (Вырезать) выпадающего меню – выделенный фрагмент будет удален, но сохранен в буфере обмена данными.

Помещение фрагмента текста в буфер обмена данными

Выделенный фрагмент текста можно поместить в буфер обмена данными. Для этого существует несколько способов:

- нажать комбинацию клавиш **Shift+Delete** – выделенный фрагмент будет удален, но сохранен в буфере обмена данными;
- нажать комбинацию клавиш **Ctrl+Insert** – выделенный фрагмент будет сохранен в буфере обмена данными;
- щелкнуть по пункту **Edit** главного меню, а затем – по пункту **Copy** выпадающего меню, – выделенный фрагмент будет скопирован и сохранен в буфере обмена данными.

Вставка фрагмента текста из буфера обмена данными

Содержимое буфера обмена данными можно вставить в любое место в текстовом окне или даже в другом окне – там, где расположен курсор. Для этого сначала помещают курсор в место вставки фрагмента.

Для вставки содержимого буфера обмена данными можно использовать несколько способов:

- нажать комбинацию клавиш **Shift+Insert**;
- щелкнуть по пункту **Edit** главного меню, а затем – по пункту **Paste** (Вставить) выпадающего меню;
- нажать комбинацию клавиш **Ctrl+V**.

Для отмены этой операции можно использовать несколько способов:

- нажать комбинацию клавиш **Alt+Backspace**;
- щелкнуть по пункту **Edit** главного меню, а затем – по пункту **Undo** (Отменить) выпадающего меню;
- нажать комбинацию клавиш **Ctrl+Z**.

Специальные комбинации клавиш

Существуют специальные комбинации «горячих» клавиш для быстрого перемещения по текстовому объекту:

- **Ctrl+Alt+B** – следующая метка;
- **Ctrl+Alt+G** – следующая строка;

- **Ctrl+Alt+N** – следующая ошибка;
- **Ctrl+Alt+P** – предыдущая ошибка.

Для редактирования текста могут быть использованы следующие комбинации клавиш:

- **Ctrl+Alt+F** – найти и заменить;
- **Ctrl+I** – вставить строку;
- **Ctrl+D** – удалить строку;
- **Ctrl+Z** – отменить последнюю команду;
- **Ctrl+C** – скопировать в буфер обмена данными;
- **Ctrl+X** – вырезать и поместить в буфер обмена данными;
- **Ctrl+V** – вставить из буфера обмена данными.

Имеются также комбинации клавиш для выполнения следующих операций:

- **Ctrl+O** – открыть файл;
- **Ctrl+S** – сохранить файл;
- **Ctrl+P** – распечатать файл.

Изменение шрифта в тексте

Для изменения шрифта в фрагменте текста:

- выделите фрагмент текста, в котором необходимо изменить шрифт, начертание и/или размер;
- щелкните по пункту **Edit** главного меню. Появится выпадающее меню;
- щелкните по пункту **Font (Шрифт)** выпадающего меню. Появится диалоговое окно **Выбор шрифта**, показанное на рис. 1.29.

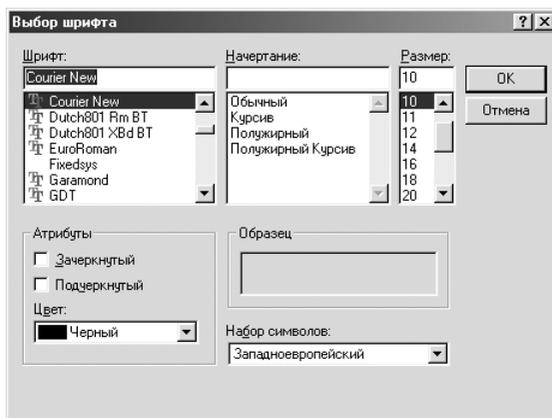


Рис. 1.29. Диалоговое окно **Выбор шрифта**

В разделе **Шрифт**: выберите один из нужных шрифтов. В разделе **Начертание**: выделите одно из четырех возможных начертаний (обычный, курсив, полужирный и полужирный курсив). В разделе **Размер**: выберите требуемый размер шрифта. В разделе **Атрибуты** можно установить написание текста в обычном виде, зачеркнутом или подчеркнутом, поставив мышью флажок в соответствующем окошке. В этом же разделе можно установить цвет шрифта, используя раскрывающийся список цветов. В разделе **Образец** сразу же после выбора типа шрифта, начертания, размера, атрибутов можно увидеть, как будет выглядеть соответствующий шрифт.

Для программирования целесообразно использовать шрифт Courier New – это шрифт с одинаковым размером всех символов. Он позволяет легко выравнивать колонки в модели, так как все знаки одинаковы по ширине.

Поиск и замена фрагмента текста

Для поиска и замены одного фрагмента текста другим существует два способа: с помощью пунктов меню или комбинации клавиш **Ctrl+Alt+F**.

Для поиска и замены с помощью пунктов меню:

- щелкните по пункту **Search** (Поиск) главного меню. Появится выпадающее меню;
- щелкните по пункту **Find / Replace** (Найти / Заменить) выпадающего меню. Появится диалоговое окно **Find / Replace**, показанное на рис. 1.30.

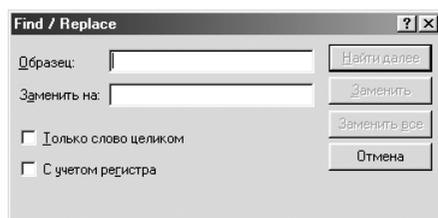


Рис. 1.30. Диалоговое окно **Find / Replace**

В приведенном выше диалоговом окне **Find / Replace** указываются фрагменты текста для замены. При этом возможна автоматическая замена по всему тексту, когда курсор стоит в начале текста, или полуавтоматическая. При использовании окна **Find / Replace** следует обратить внимание на то, что процедура поиска работает в одном направлении и заканчивается тогда, когда достигнут конец файла. Лучше поместить курсор в начало файла при выполнении поиска.

Все окна в системе GPSS можно для удобства перемещать по экрану. Для этого в пределах строки заголовка окна нажмите кнопку мыши и, удерживая ее, переместите окно в нужное место. Затем отпустите кнопку мыши для фиксации нового местоположения перемещенного окна.

Настройка установок

Для управления моделированием, настройки внешнего вида сообщений, содержания окон и определения функциональных клавиш используются соответствующие установки. Установки – это совокупность значений, которые формируют определенное представление большинства GPSSW-объектов. Все создаваемые объекты наследуют начальные установки уже созданной исходной модели. Так, если вы хотите изменить значения для всех моделей в своем проекте, вы должны сделать это в исходном объекте модели. Только текстовые объекты не содержат установок системы GPSSW. Только эти объекты системы GPSSW могут быть открыты для редактирования во внешних текстовых редакторах. Для введения тех или иных установок:

- щелкните по пункту **Edit** (Правка) главного меню. Появится выпадающее меню;
- щелкните по пункту **Settings** (Установки) выпадающего меню. Появится диалоговое окно **SETTINGS** с пятью вкладками.

Установка параметров моделирования

Диалоговое окно **SETTINGS** на рис. 1.31 показывает установки моделирования для модели под именем **Kranm_1.gps**.

Вкладка **Simulation** (Моделирование) содержит несколько значений, которые можно изменять и которые воздействуют на пределы и поведение процесса моделирования.

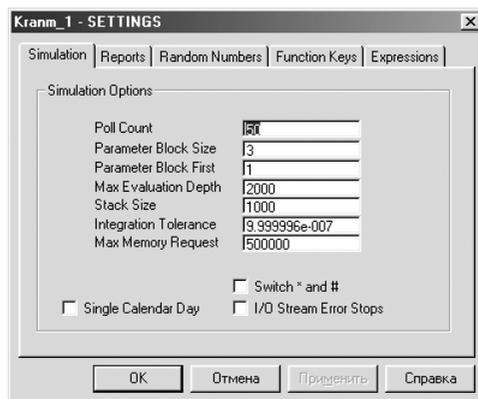


Рис. 1.31. Диалоговое окно **SETTINGS** с открытой вкладкой **Simulation** для модели **Kranm_1**

В поле **Poll Count** (Индекс опроса) устанавливается число блочных вхождений, которые будут предприняты перед проверкой любого вида прерывания при введении пользователем команд пункта **Command** главного меню.

Если используется много параметров в требовании, то можно для ускорения выполнения программы поместить их в блок параметров.

В поле **Parameter Block Size** (Размер блока параметров) устанавливается размер блока параметров. В поле **Parameter Block First** (Первый блок параметров) указывается число параметров в блоке. Поле **Max Evaluation Depth** (Максимальная глубина вычисления) используется, чтобы защитить против циклических зависимостей в PLUS-процедуре. В поле **Stack Size** (Размер стека) указывается размер стека для вложенных PLUS-процедуры и библиотечных процедур. Можно сэкономить пространство, уменьшив это значение, но если вы получите сообщение «Stack Overflow Error Stop» (Остановка по ошибке из-за переполнения стека), то придется увеличить размер стека.

Поле **Integration Tolerance** (Общий допуск) определяет общую степень допуска. Если допуск широкий, то интегрирование будет проходить быстрее, но может теряться некоторая точность в конечных результатах.

Поле **Max Memory Request** (Максимальная требуемая память) определяет максимальную требуемую оперативную память для структур типа таблиц и матриц.

Три флажка в нижней части вкладки управляют степенью совместимости с системой GPSS/PC, которая доступна только в коммерческой версии.

Флажок **Single Calendar Day** используется для того, чтобы при моделировании использовать один список будущих событий (FEC) вместо нескольких.

Флажок **I/O Stream Error Stops** (Ввод/вывод потока остановок при ошибках) обеспечивает ввод/вывод потока остановок при возникновении ошибок.

Установка содержания отчета

Вкладка **Reports**, показанная на рис. 1.32, прежде всего, связана с представлением тех или иных отчетов – результатов моделирования.

Флажок **Create Standard Reports** (Создать стандартные сообщения) обеспечивает автоматическое создание стандартных сообщений после окончания процесса моделирования.

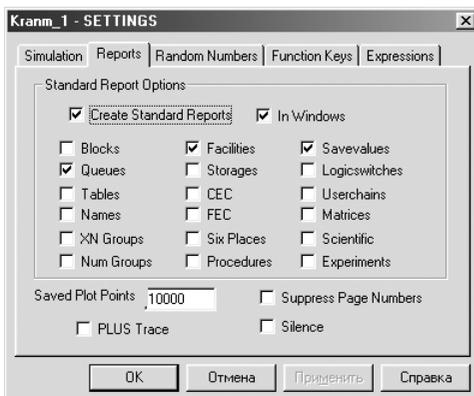


Рис. 1.32. Диалоговое окно **SETTINGS** с открытой вкладкой **Reports** для модели *Kranm_1*

Флажок **In Windows** (В окнах) обеспечивает текстовое представление каждого объекта сообщения вместо сохранения их в файле.

Ниже приводится группа флажков, обеспечивающих вывод результатов функционирования того или иного элемента моделируемой системы:

- **Blocks** (Блоки);
- **Queues** (Очереди);
- **Tables** (Таблицы);
- **Names** (Имена);
- **XN Groups** (XN-группы);
- **Num Groups** (Числовые группы);
- **Facilities** (Каналы обслуживания);
- **Storages** (Накопители);
- **SEC** (Цепь текущих событий);
- **FEC** (Цепь будущих событий);
- **Six Places** (Шесть знаков);
- **Procedures** (Процедуры);
- **Savevalues** (Сохраняемые величины);
- **Logicswitches** (Логические переключатели);
- **Userchain** (Цепь пользователя);
- **Matrices** (Матрицы);
- **Scientific** (Научный);
- **Experiments** (Эксперименты).

Наличие галочки в том или ином окошке обеспечивает вывод в стандартном отчете результатов моделирования выделенных элементов.

Флажок **Six Places** устанавливает точность действительных значений с шестью десятичными знаками вместо трех.

Флажок **Scientific** используется для представления мантиссы чисел в виде десятичной экспоненты в сообщениях и потоках данных.

Поле **Saved Plot Points** (Сохраненное значение точек) сообщает системе GPSSW, сколько пространства выделить круговому буферу для точек.

Флажок **Suppress Page Numbers** (Не вставлять номера страниц) служит для удаления номеров страниц при печати текстовых объектов.

Флажок **PLUS Trace** (Трассировка PLUS) обеспечивает трассировку языка программирования PLUS.

Флажок **Silence** (Молчание) обеспечивает выдачу ошибок без звукового сопровождения.

Установка параметров генераторов случайных чисел

Вкладка **Random Numbers**, показанная на рис. 1.33, позволяет управлять внутренними потоками генераторов случайных чисел, а также внешней нумерацией дочерних объектов.

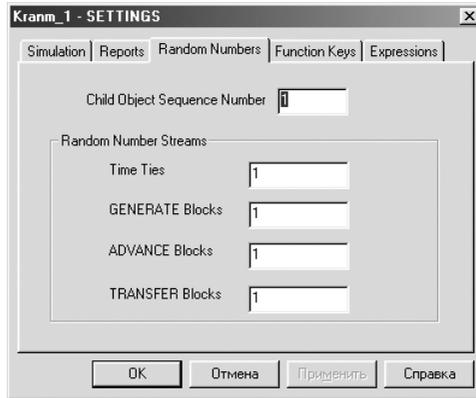


Рис. 1.33. Диалоговое окно **SETTINGS** с открытой вкладкой **Random Numbers** для модели *Kranm_1*

Поле **Child Object Sequence Number** (Порядковый номер дочернего объекта) определяет нумерацию дочерних объектов.

Для сохранения уникальных названий объектов система GPSSW прибавляет порядковый номер к названиям недавно созданных дочерних объектов. Число, которое нужно использовать для следующего порядкового номера, сохраняется.

В разделе **Random Number Streams** (Номера потоков случайных чисел) есть четыре текстовых поля:

- **Time Ties** (Временные связи) позволяет определять, когда те или иные события происходят одновременно. Если используется значение 0, то временные связи не будут рандомизированы;
- **GENERATE Blocks** (Блоки **GENERATE**) позволяет задать номер генератору случайных чисел. Блоки **GENERATE** вычисляют время между двумя входами – время для операндов А и В. Если вы определяете неположительный номер, то используется генератор случайных чисел под номером 1;
- **ADVANCE Blocks** (Блоки **ADVANCE**) позволяет выбрать номер генератора случайных чисел – времени задержки для операндов А и В. Если вы определяете неположительный номер, то используется генератор случайных чисел под номером 1;
- **TRANSFER Blocks** (Блоки **TRANSFER**) позволяет определить, какой генератор случайных чисел должен быть использован, когда блок передачи выбирает вероятностного адресата блока. Если вы определяете неположительный номер, то используется генератор случайных чисел под номером 1.

Назначение функциональных клавиш

Вкладка **Function Keys** (Функциональные клавиши), показанная на рис. 1.34, используется для назначения тех или иных функциональных клавиш командам управления процессом моделирования.

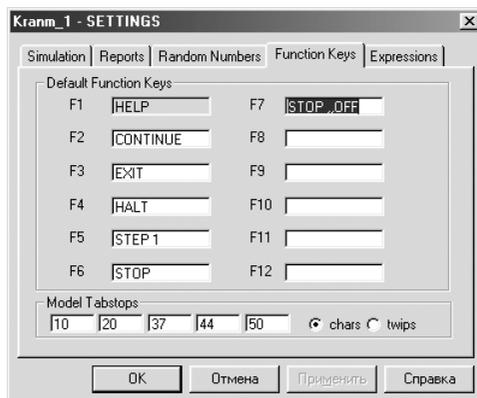


Рис. 1.34. Диалоговое окно **SETTINGS** с открытой вкладкой **Function Keys** для модели *Kranm_1*

В разделе **Default Function Keys** (Функциональные клавиши по умолчанию) приведены установки по умолчанию для функциональных клавиш **F1–F12**. Так, нажатие функциональных клавиш **F1–F7** соответствует выбору следующих команд:

- **HELP** (Справка) – вызывает справочную систему GPSSW Application Help;
- **CONTINUE** (Продолжить) – обеспечивает продолжение выполнения прерванного процесса моделирования системы;
- **EXIT** (Выйти) – обеспечивает выход из процесса моделирования;
- **HALT** (Остановить) – обеспечивает прерывание процесса моделирования;
- **STEP 1** – обеспечивает выполнение одного шага моделирования;
- **STOP** (Остановить) – обеспечивает определение условий остановки процесса моделирования;
- **STOP,,OFF** (Отменить остановку) – удаляет все условия остановки процесса моделирования.

В разделе **Model Tabstops** (Табличные остановки курсора в модели) устанавливаются положения курсора при использовании клавиши **Tab** (Табуляция).

Установка просмотра выражений

Вкладка **Expressions** (Выражения), показанная на рис. 1.35, предназначена для просмотра выражений, включенных для контроля.

На этой вкладке вы можете определить выражение, которое должно иметь такое же имя, которое использовалось в качестве метки. Эта вкладка позволяет вам сохранять самые различные выражения, связанные с объектом моделирования. При необходимости вы можете выбрать любое выражение из списка. Вкладка **Expressions** обеспечивает возможность запоминания наиболее часто используемых выражений при моделировании той или иной системы. Для этого:

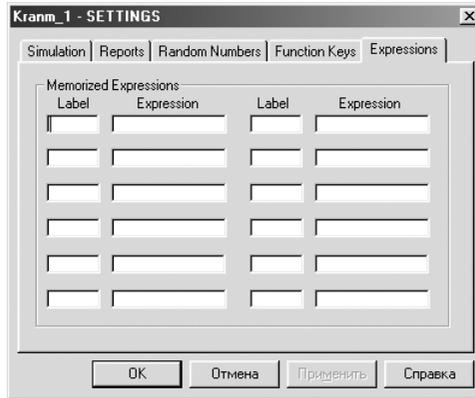


Рис. 1.35. Диалоговое окно **SETTINGS** с открытой вкладкой **Expressions** для модели *Kranm_1*

- щелкните по пункту **Edit** главного меню. Появится выпадающее меню;
- щелкните по пункту **Expressions Window** (Окно выражений);
- введите нужные выражения и щелкните по кнопке **Memorize** (Запомнить).

После этого очень просто вызывать окно **Edit Expression Window** (Окно редактирования выражения) для просмотра и редактирования его содержимого. Окно **Edit Expression Window** может выглядеть так, как показано на рис. 1.36.

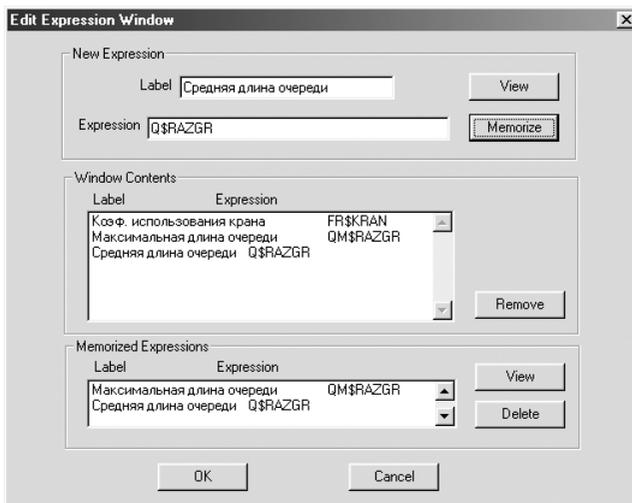


Рис. 1.36. Диалоговое окно **Edit Expression Window** для модели *Kranm_1*

Установка символа умножения

По умолчанию система GPSSW использует в качестве символа умножения значок #, что иногда затрудняет восприятие выражений. Можно использовать вместо символа # традиционный знак *. Для этого:

- щелкните по пункту **Edit** (Правка) главного меню. Появится выпадающее меню;
- щелкните по пункту **Settings** (Установки) выпадающего меню. Появится диалоговое окно **SETTINGS**;
- щелкните по вкладке **Simulation** (Моделирование), а затем включите флажок **Switch * and #** (Включить * вместо #);
- щелкните по кнопке **Применить**, а затем – по кнопке **ОК**.

Основы моделирования в системе GPSSW

Базовые понятия и определения	48
Основные составляющие системы GPSSW	64
Основные этапы моделирования в системе GPSSW	67
Моделирование в интерактивном режиме	76
Отладка модели	84

В главе изложены базовые понятия и определения языка GPSS. Рассказывается об этапах программирования, операторах и командах языка GPSS.

Базовые понятия и определения

Система GPSSW, как вам уже известно, предназначена для моделирования дискретных (в основном, систем массового обслуживания) и непрерывных систем.

Система массового обслуживания

Система массового обслуживания – это совокупность последовательно связанных между собой входящих потоков требований на обслуживание (машин, самолетов, пользователей и т.д.), накопителей, очередей, каналов обслуживания (станций техобслуживания, аэродромов, ЭВМ и т.д.) и выходящих потоков требований после обслуживания.

Входящий поток требований – это последовательность входящих требований, нуждающихся в обслуживании в системе и подчиняющихся определенному закону.

Выходящий поток требований – это последовательность выходящих требований, обслуженных в системе и подчиняющихся определенному закону.

Требование (транзакт) – это объект, поступающий в систему и нуждающийся в определенном обслуживании в данной системе. Требование является активным элементом в моделируемой системе. Под требованием можно понимать сообщение, машину, изделие, информацию, пользователя и т.д. Требование в системе GPSSW – это объект с набором определенных признаков, который определяется уникальным номером. Объекты нумеруются последовательно, начиная с номера 1.

Канал обслуживания – устройство, в котором выполняется обслуживание требования. Основным параметром канала обслуживания является время обслуживания, которое, как правило, является случайной величиной.

Накопитель (буфер) – это место временного расположения требований, нуждающихся в обслуживании. Накопители характеризуются не временем обслуживания требований, а емкостью – максимально возможным количеством одновременно находящихся в накопителе требований.

Имитационная модель

Имитационная модель – это формальное описание логики функционирования исследуемой системы во времени, учитывающее наиболее существенные взаимодействия ее элементов и обеспечивающее возможность проведения статистических экспериментов.

Имитационная модель СМО – это модель, отражающая поведение системы и изменения ее состояния во времени при заданных потоках требований, поступающих на входы системы. Параметры входных потоков требований – внешние параметры СМО. Выходными параметрами являются величины, характеризующие свойства системы – качество ее функционирования, – например такие, как:

- коэффициенты использования каналов обслуживания;
- максимальная и средняя длина очередей в системе;
- время нахождения требований в очередях и каналах обслуживания и т.д.

Имитационное моделирование – это численный метод определения параметров функционирования самых различных систем по многочисленным реализациям с учетом вероятностного характера протекания процесса.

Основой имитационного моделирования является метод статистических испытаний – метод Монте-Карло. Этот метод наиболее эффективен при исследовании сложных систем, на функционирование которых оказывают существенное влияние случайные факторы.

Имитационное моделирование позволяет исследовать СМО при различных типах входных потоков и разной интенсивности поступления требований в систему, а также различных дисциплинах обслуживания требований.

Модель в системе GPSSW – это последовательность операторов моделирования. Оператор моделирования может быть или оператором GPSS, или процедурой языка PLUS (Programming Language Under Simulation – Язык программирования для моделирования).

Представление времени в процессе имитации

При имитационном моделировании выделяют три представления времени: реальное, модельное (системное) и машинное время.

Реальное время – это время, в котором происходит функционирование моделируемой системы в реальной жизни, например час, смена, год.

Модельное (системное) время – это время, в котором происходит функционирование моделируемой системы при проведении имитационного моделирования на ЭВМ. В системном времени выполняются следующие действия:

- осуществляется переход моделируемой системы из одного состояния в другое;
- выполняется синхронизация работы всех компонент имитационной модели;
- обеспечивается управление ходом имитационных экспериментов;
- обеспечивается параллельная реализация событий в моделируемой системе.

Машинное время – это время, отражающее затраты времени ЭВМ на проведение имитационного моделирования.

Система моделирования GPSSW основана на переходе требований (транзактов) от блока к блоку (от оператора к оператору) в определенные моменты времени, что называется событием.

Событие – это нечто, меняющее статус связанных с ним состояний системы. События соответствуют конкретным изменениям в реальной системе: требование появилось, требование вошло в очередь, требование обслуживается и т.д. При этом прослеживается процесс моделирования с использованием как абсолютного времени (текущего), так и относительного (с начала прогона модели, с момента последней модификации или последнего сброса и т.д.). События характеризуются условиями (или законом) возникновения и типом, который определяет порядок их обработки (дисциплину обслуживания). Событие по существу представляет собой мгновенное изменение некоторого элемента системы или состояния системы в целом.

Очень важная особенность GPSSW заключается в том, что продолжительность моделирования определяется не длительностью интервалов между событиями, а числом событий, возникающих в системе. Поэтому переход на более мелкие единицы времени не увеличивает время моделирования.

Динамические элементы системы

Динамическими объектами системы являются требования (транзакты), которые создаются и уничтожаются в определенные моменты времени моделирования той или иной системы. Требования перемещаются от блока к блоку, которые имитируют реальные элементы моделируемой системы.

Поведение требования определяется его несколькими фазовыми переменными (атрибутами, параметрами).

Параметры требования – это набор значений, связанных с требованием. Каждое требование может иметь любое число параметров. Каждый параметр имеет номер – положительное целое число. Значение любого параметра активного требования может быть возвращено через стандартный числовой атрибут (Standard Number Attribute, SNA) P<номер требования> или P\$<имя требования>. Необходимо создать параметры требования и присвоить им значения до того, как они будут использоваться. Параметрами могут быть:

- *приоритет*, определяющий предпочтение, которое требование получает, когда оно вместе с другими требованиями ожидает один и тот же ресурс. Требования с более высокими приоритетными значениями получают предпочтение. Наиболее важные приоритетные очереди в моделировании – цепь текущих событий (Current Events Chain), цепи задержек каналов обслуживания (Facility Delay Chains) и цепи задержек накопителей (Storage Delay Chains). Эффект приоритета заключается в том, что требование с более высоким приоритетом будет выбрано на обслуживание раньше требований с низким приоритетом;
- *метка времени* – абсолютное время – с начала моделирования или с момента входа требования в блок **MARK** (Метка) без операнда A;
- *номер ансамбля* – положительное целое число, сохраняемое внутри каждого требования. Номера для ансамбля используются, чтобы синхронизировать требования в блоках **ASSEMBLE** (Объединить), **GATHER** (Собрать) и **MATCH** (Согласовать). Когда требование создается блоком **GENERATE**, его номер для ансамбля устанавливается равным номеру требования. Когда требование создается блоком **SPLIT** (Разделить), номер для ансамбля устанавливается равным номеру родительского требования. Можно изменить номер требования для ансамбля, используя блок **ADOPT** (Принять);
- *индикатор задержки* – флажок, сохраняемый в каждом требовании, который устанавливается при входе в любой блок и сбрасывается при входе в блок **TRANSFER SIM**. Он используется блоками **TRANSFER SIM**, чтобы переадресовать требования;

- *индикатор трассировки* – флажок, сохраняемый в требовании, который заставляет генерировать сообщение трассы каждый раз, когда требование, входит в любой блок. Индикатор трассировки устанавливается блоком **TRACE** (Трассировать) и сбрасывается блоком **UNTRACE** (Не трассировать);
- *текущий блок* – номер блока, который содержит требование;
- *следующий блок* – номер блока, в который требование будет пытаться войти далее;
- *цепи состояния требования* – в них требование постоянно находится. Считается, что требование может находиться в одном из нескольких состояний:
 - *активное требование* – самое приоритетное требование в цепи текущих событий;
 - *приостановленное требование* – оно ожидает в цепи будущих или текущих событий, чтобы стать активным;
 - *пассивное требование* – оно при моделировании временно находится в цепи пользователя (User Chain), задержки (Delay Chain) или ожидания (Pending Chain);
 - *удаленное требование* – оно было удалено и больше не существует в системе моделирования.

Кроме того, имеются другие состояния, которые не являются взаимоисключающими с вышеуказанными:

- *выгруженное требование*, которое выгрузилось из канала обслуживания и находится в одной или большем числе цепей прерываний;
- в любой момент во время дискретной стадии моделирования одно какое-либо требование пытается войти в новый блок GPSS. Это требование является активным. Вообще, активное требование перемещается настолько далеко, насколько это возможно в процессе моделирования. Когда оно не может перемещаться дальше, активизируется другое требование. При этом может быть только одно активное требование. Активное требование – самое высокое приоритетное требование в цепи текущих событий.

Можно выделить стандартные числовые атрибуты, связанные с требованиями:

- A1 – определяет объединенный набор активных требований;
- MB\$<имя> или MB<номер> – возвращает 1, если имеется требование в блоке, которое находится в том же самом наборе, что и активное требование. Иначе возвращается 0;
- MP\$<имя> или MP<номер> – определяет время прохождения требования с заданным параметром, то есть текущее абсолютное время моделирования системы;
- M1 – время прохождения требования. M1 возвращает абсолютное время моделирования системы минус метка времени требования;
- P\$<имя>, P<номер> или *<имя> – возвращает значение параметра активного требования;
- PR – приоритет активного требования;
- XN1 – возвращает номер активного требования.

Именованные величины

В системе GPSSW используют символы обозначения и специальные символы.

Символы обозначения включают прописные буквы A–Z, строчные буквы a–z, цифры 0–9 и символ (подчеркивание).

Специальные символы – это символы, используемые для обозначения операторов и пунктуации: # (решетка), * (звездочка), + (плюс), - (минус), / (косая черта направо), \ (косая черта налево) и , (запятая). Символ [^] также является оператором.

Для идентификации объектов, переменных и местоположения программы используются имена – последовательность символов. Существует несколько правил для создания имени:

- можно использовать от 1 до 250 символов;
- имя должно начинаться с символа;
- имя не должно быть ключевым словом системы GPSSW.

Система GPSSW не различает в обозначениях верхний и нижний регистры (прописные или строчные буквы). Только символы в строковых константах и комментариях сохраняют нижний регистр. Все другие символы строчных букв преобразуются в прописные.

Нельзя присваивать имена операторов, команд, а также стандартных числовых атрибутов, используемых в системе. Чтобы избежать риска совпадения имени с ключевыми словами системы GPSSW, включайте в имя знак подчеркивания где-нибудь после первого символа, в середине или в конце. Используйте, по крайней мере, 3 символа сначала, а затем цифры, так как имена в стандартных числовых атрибутах имеют 1–2 буквы и цифру.

Именованные величины – это имена, размещаемые в поле меток или в операторе присваивания языка PLUS. Если имена используются в качестве метки оператора GPSS, они называются метками. Если они используются в команде EQU или операторе присваивания PLUS, они называются переменными пользователя. Именованные величины обычно имеют глобальную область использования, и к ним можно обращаться в любом месте модели.

Метка – имя, которое используется в команде создания объекта. В отличие от переменных пользователя, меткам автоматически присваивается уникальное целое число больше 9999. Когда вы обращаетесь к помеченному объекту, система GPSSW сначала восстанавливает номер объекта, сохраненный как значение метки. Номер объекта – строго положительное целое число, которое используется объектом моделирования, чтобы найти или создать любой объект GPSS. Номер объекта устанавливается системой GPSSW во время его создания. Однако если вы хотите назначить другой номер объекту, используйте команду EQU, помещая ее перед оператором определения объекта.

Переменная пользователя – именованная величина, не используемая как метка объекта, которой можно давать числовое или строковое значение и по которой можно интегрировать, как по непрерывной переменной. Переменные пользователя нужно инициализировать, прежде чем их можно будет применять.

Большую роль в процессе моделирования играют величины, которые сохраняют свое начальное значение на протяжении всего процесса моделирования.

Сохраняемые величины – это величины, начальные значения которых задаются перед началом моделирования и к которым можно обратиться из любого места программы.

Матрица – это многомерный массив элементов. Массив в общем случае характеризуется размерностью и размером – числом элементов в каждой размерности. Массив размерностью 1 представляет собой строку (столбец), а размер – это число элементов в строке (столбце). Массив размерностью 2 представляет собой прямоугольную таблицу, а размер – это число элементов в строке (столбце), умноженное на число строк (столбцов) в таблице. Матрицу размерностью 3 можно представить в виде нескольких таблиц данных или результатов моделирования.

Матричный элемент – это любой элемент матрицы. Матричный элемент представляет собой индексированную величину.

Логические переключатели – объекты, которые могут находиться только в одном из двух состояний: установки или сброса.

Типы данных

Все переменные пользователя, матричные элементы, сохраняемые величины и параметры требования могут иметь значение любого типа данных. В системе выделяют три главных типа данных: целые, вещественные и строковые константы. Первые два относятся к числовому типу данных.

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

Вещественные константы имеют числа с двойной точностью с плавающей запятой. Они имеют точность 15 десятичных цифр и диапазон экспоненты от –306 до 306.

Строковая константа – последовательность символов ASCII, взятая в двойные кавычки. Строковая константа может иметь любой размер, насколько позволит память. Для создания и управления строковыми константами в системе имеется множество строковых процедур, которые находятся в библиотеке процедур. Строковые константы используются тогда, когда вы выводите результаты моделирования в файл результата. Вы можете также использовать строковые константы, чтобы формировать свои собственные специальные сообщения.

Элементы выражений

Элементы выражений – это основные стандартные блоки выражений, которые, в свою очередь, могут использоваться в полях операндов операторов GPSS и процедурах PLUS.

Элементами выражений могут быть:

- строковые константы, например "Go to metka";
- вещественные константы, например 201.6;
- целые константы, например 17;
- имена, например Kanal;
- элементы матрицы, например $Massiv[P\$Part, Q2+20]$;
- система числовых атрибутов, например AC1, F\$My_Kran, MX\$Mat1(2,1) и SR*MY_PARM.

Арифметические целые переменные

Арифметические целые переменные определяются с помощью оператора **VARIABLE** (Переменная). Перед оператором **VARIABLE** в поле меток ставится символьное или числовое имя переменной (идентификатор), а в поле переменных пишется арифметическое выражение, определяющее данную переменную, например:

```
19 VARIABLE Q2 + 3
```

Такая запись означает, что арифметическая переменная под номером 19, которую далее именуют V19, равна сумме числа требований в очереди под номером 2 (Q2) и константы 3.

```
SUM VARIABLE (P3 + P4)/5
```

Такая запись означает, что арифметическая переменная с символьным именем SUM, которую далее именуют V\$SUM, равна сумме значений 3-го и 4-го параметров требования, деленных на 5.

Каждый раз при обращении к арифметическим переменным V19 или V\$SUM их значения будут рассчитываться по приведенным выше выражениям, составляющие которых в процессе моделирования могут менять свои значения.

Ссылки из любого места программы на арифметическую переменную вводятся с помощью стандартного обозначения. Обращение к переменной с числовым или с символьным именем в нашей задаче будет выполняться соответственно так: V19 и V\$SUM.

Значение арифметической переменной может использоваться как:

- элемент другой арифметической переменной, булевой переменной;
- аргумент функции, таблицы;
- зависимое значение функции, задаваемое признаком;
- операнд операторов **ASSIGN**, **INDEX**, **LOOP**, **SPLIT** и др.

В арифметических выражениях допускаются следующие арифметические операции: + (сложение); - (вычитание); * (умножение); / (деление).

Переменные вычисляются слева направо, причем операции умножения и деления обладают старшинством над операциями сложения и вычитания.

Арифметическое выражение образуется из целых констант, других арифметических переменных, стандартных числовых атрибутов, знаков арифметических операций и круглых скобок.

Арифметические переменные с фиксированной точкой

Арифметические переменные с фиксированной точкой определяются с помощью оператора **FVARIABLE**. Перед оператором **FVARIABLE** в поле меток ставится символьное или числовое имя переменной (идентификатор). После оператора в поле переменных пишется арифметическое выражение, определяющее данную переменную, например

```
PER    FVARIABLE (S1-S$CAN)/5 + 3.6
```

Эта запись означает арифметическую переменную с фиксированной точкой под символьным именем PER, которую далее именуют $V\$PER$ и которая равна сумме разности текущих вместимостей накопителей под номером 1 и символьным именем CAN, деленной на 5, и десятичной константы с фиксированной точкой, равной 3,6.

Ссылка на арифметическую переменную с фиксированной точкой выполняется так же, как и на арифметическую переменную. В нашем примере это $V\$PER$.

Арифметические выражения арифметической переменной с фиксированной точкой образуются аналогично арифметической целой переменной.

Булевы переменные

Булевы переменные определяются с помощью оператора **BVARIABLE**, в поле меток которого ставится символьное или числовое имя (идентификатор) переменной, а в поле переменных пишется булево выражение. Булево выражение образуется из стандартных числовых атрибутов булевых переменных, знаков булевых операций и условных операторов. Булево выражение принимает одно из двух значений: 1 или 0.

Вычислительные выражения

Вычислительные выражения представляют собой комбинацию математических операторов, библиотечных функций, стандартных числовых атрибутов и констант, которые удовлетворяют правилам элементарной алгебры. Они вычисляются согласно иерархии операторов, перечисленных выше, и в направлении слева

направо. Порядок вычисления можно изменить с помощью круглых скобок, как это делается в любом алгебраическом выражении.

Ниже представлены вычислительные и логические операторы, используемые в системе GPSSW (табл. 2.1).

Таблица 2.1

Оператор	Пример использования	Описание
^	A ^ B	Возведение в степень
#	A # B	Умножение
/	A / B	Деление
\	A \ B	Целочисленное деление. Возвращает результат целочисленного деления A на B
@	A @ B	Целочисленный остаток. Возвращает целочисленный остаток от деления A на B
-	A - B	Вычитание
+	A + B	Сложение
>= или 'GE'	A >= B	Возвращается 1, если A в цифровой форме больше или равно B, в противном случае возвращается 0
<= или '@'	A <= B	Возвращается 1, если в цифровой форме A меньше или равно B, в противном случае возвращается 0
> или 'G'	A > B	Возвращается 1, если в цифровой форме A больше B, в противном случае возвращается 0
< или 'L'	A < B	Возвращается 1, если в цифровой форме A меньше B, в противном случае возвращается 0
= или 'E'	A = B	Возвращается 1, если в цифровой форме A равно B, в противном случае возвращается 0
!= или 'NE'	A != B	Возвращается 1, если в цифровой форме A отличается от B, в противном случае возвращается 0
& или 'AND'	A & B	Возвращается 1, если A и B не равны нулю, в противном случае возвращается 0
или 'OR'	A B	Возвращается 1, если A, или B, или оба не равны нулю, в противном случае возвращается 0

Вычисления выражений выполняются в такой последовательности:

- ^ – возведение в степень;
- #, /, \ – умножение, деление, целочисленное деление;
- @ – целочисленный остаток;

- -, + – вычитание, сложение;
- >=, <=, >, < – операторы сравнения;
- =, != – равный, не равный;
- & – логическое И;
- | – логическое ИЛИ.

Например, выражение

$$2 \# 5 \wedge 2 + 34$$

вычисляется так: сначала число 5 возводится в степень 2, затем полученный результат, 25, умножается на смежный множитель 2, после чего новый результат, 50, прибавляется к слагаемому 34. Окончательный результат – 84. Это выражение можно представить и в таком виде:

$$((2 \# (5 \wedge 2)) + 34).$$

Для изменения установленного порядка выполнения арифметических операций используются круглые скобки. Например, выражение

$$(2 \# 5) \wedge 2 + 34$$

вычисляется в такой последовательности: сначала выполняется умножение числа 2 на 5, затем полученный результат, 10, возводится в степень 2, и новый результат, 100, прибавляется к слагаемому 34. Окончательный результат – 134. Это выражение можно представить и в таком виде:

$$(((2 \# 5) \wedge 2) + 34).$$

Операторы системы GPSSW определяют тип данных непосредственно перед тем, как операция применяется. Поэтому нет необходимости беспокоиться о типах данных при создании PLUS-выражений. Выражения могут оцениваться в числовой или строковой формах. Когда выражение оценивается в числовой форме, строковый результат преобразуется к его числовому эквиваленту, основанному на числах, с которых начинается строка. Строка, начинающаяся не с цифры, преобразуется к числовому нулю. Точно так же, когда выражение оценивается как строка, любой числовой результат преобразуется к строковому эквиваленту.

В сообщениях и потоках данных представлением больших чисел можно управлять, подавляя экспоненциальный формат. Для этого:

- щелкните по пункту **Edit** главного меню или нажмите комбинацию клавиш **Alt+E**. Появится выпадающее меню;
- щелкните по пункту **Settings** (Установки) выпадающего меню. Появится диалоговое окно **SETTINGS**;
- выберите вкладку **Report** (Отчет) и установите галочку напротив элемента **Scientific** (Научный).

После выполнения этих действий число 1100, например, будет представлено как 1.1e3 в экспоненциальном формате.

Основные операторы языка GPSS

Оператор – это специальное имя (обозначение) для определенного действия (операции) над операндами (данными). В табл. 2.2 приведен перечень наиболее часто используемых операторов языка имитационного моделирования GPSS с краткими пояснениями их назначения.

Таблица 2.2

Оператор		Описание
GENERATE	A,B,C,D,E,F,G,H	Сгенерировать требование
QUEUE	A,B	Увеличить содержимое очереди
DEPART	A,B	Уменьшить содержимое очереди
SEIZE	A	Занять канал обслуживания
RELEASE	A	Освободить канал обслуживания
ENTER	A,B	Увеличить вместимость накопителя
LEAVE	A,B	Уменьшить вместимость накопителя
ADVANCE	A,B	Задержать перемещение требования
ASSIGN	A,B	Модифицировать параметр требования
PREEMPT	A,B,C,D,E	Отстранить требование от обслуживания
RETURN	A	Вернуть канал обслуживания A
BUFFER		Идти в хвост цепи текущих событий
GATE	A,B	Переместить в зависимости от состояния
GATHER	A	Подождать родственные требования
LINK	A,B,C	Ввести в цепь пользователя
UNLINK	A,B,C,D,E,F,G	Вывести из цепи пользователя
<Имя> STORAGE	A	Определить вместимость накопителя
MARK	A	Создать временную метку
MATCH	A	Подождать другое требование
SELECT	A,B,C,D,E,F,G	Выбрать элемент
SAVEVALUE	A,B,C	Модифицировать сохраняемое значение
MSAVEVALUE	A,B,C,D	Модифицировать значение матрицы
PRIORITY	A,B	Изменить приоритет
SPLIT	A,B,C,D	Разделить требование
TABULATE	A,B	Собрать табличные данные элемента
TERMINATE	A	Уничтожить требование
TEST	A,B,C	Переместить согласно сравнению
TRANSFER	A,B,C,D	Передать в новое местоположение

Таблица 2.2 (окончание)

Оператор		Описание
ASSEMBLE	A	Собрать родственные требования
EQU	A,B	Определить эквивалентность
MATRIX	A,B,C	Задать матрицу
INITIAL	A,B	Инициализировать
RMULT	A,B,C,D,E,F,G	Установить значение генератора
<Имя> VARIABLE	A	Определить целую переменную
<Имя> FVARIABLE	A	Определить вещественную переменную
<Имя> BVARIABLE	A	Определить булеву переменную
<Имя> FUNCTION	A,B	Определить функцию
<Имя> TABLE	A,B,C,D,E	Определить таблицу
QTABLE	A,B,C,D	Определить таблицу для очереди
LOGIC	A	Изменить логический переключатель
LOOP	A,B	Изменить параметр требования и зациклить
REPORT		Сообщить
COUNT	A,B,C,D,E	Поместить индекс объекта внутрь параметра
TRACE		Начать трассировку
UNTRACE		Завершить трассировку
INDEX	A,B	Увеличить параметр требования
EXECUTE	A	Выполнить блочную операцию элемента
WRITE		Разместить текстовую строку в потоке данных
READ		Считать текстовую строку из потока данных
RESET		Сбросить
ADOPT		Изменить номер ансамбля
CLOSE		Закрыть поток данных
JOIN		Увеличить группу членов
ALTER		Изменить атрибуты требования
DISPLACE		Переместить требование к блоку
EXAMINE		Исследовать член группы
FAVAL		Сделать канал обслуживания доступным
FUNAVAl		Сделать канал обслуживания недоступным
INTEGRATION		Включить или выключить интегрирование
PLUS		Оценить PLUS-выражение
SAVAL		Сделать накопитель доступным
SUNAVAl		Сделать накопитель недоступным
REMOVE		Уменьшить группу членов
SEEK		Установить позицию в потоке данных
SCAN		Восстановить информацию группы членов

Каждый оператор языка GPSS должен находиться на одной строке текста и включать в совокупности с другими элементами не более 250 символов. Он структурно состоит из нескольких полей. Выделяют следующие поля оператора:

- номер строки (необязательный);
- метка (необязательный);
- собственно оператор;
- операнды (зависят от введенного оператора);
- комментарий (необязательный).

Поле – переменное число печатаемых символов, за которыми следует незаполненное пространство или разделитель.

Номера строк, если они используются, должны вводиться с первой позиции (столбца). Однако номера строк в системе GPSS World игнорируются транслятором. Номер же строки, данный в сообщении об ошибках, – это число, обозначающее место строки в программе, а не указанный пользователем номер строки.

Поле метки – это поле, в котором может быть введено имя данного оператора – числовое или символьное – для обращения к нему из другого места программы.

Поле оператора – это поле, в котором записывается один из операторов языка имитационного моделирования GPSS.

Поле операндов – это поле ввода требуемых для данного процесса моделирования операндов (аргументов оператора). Большинство операторов имеют один или несколько операндов. Поля операндов часто называют полями переменных. Традиционно операнды разделяются запятыми. Допускается пропускать в операторах некоторые операнды – так называемые необязательные операнды. В поле переменных выделяют *подполя*, разделяемые при записи запятыми и служащие для указания числовых значений переменных, стандартных числовых атрибутов, числовых или символьных меток и т.п. Подполя могут быть пустыми. Стандартные числовые атрибуты служат для сокращенного указания различных величин, фигурирующих в модели, например:

- K15 – константа, равная 15;
- V12 – переменная под номером 12;
- Q\$kanal – средняя длина очереди под символьным именем kanal;
- X17 – сохраняемая величина под номером 17;
- FN\$ECSP – функция с именем ECSP;
- P2 – значение параметра требования под номером 2;
- *4 – содержимое параметра требования под номером 4;
- S*5 – вместимость накопителя, определенная в параметре требования под номером 5.

Поле комментария – это поле размещения комментария, относящегося к используемому оператору. Существует два основных вида комментария: входные и промежуточные. Первый вид комментария начинается с символов ; (точка с запятой) или * (звездочка), а далее идет любой текст.

Во входном комментарии можно указать, что делает программа, какую систему моделирует, какие требуются входные данные, что получается в результате моделирования. Входные комментарии оформляются по определенным правилам. В первой позиции (колонке) ставится знак * или ;. Далее с любой позиции (колонки) пишется необходимая поясняющая информация.

Промежуточные комментарии пишутся в поле переменных (операндов), отделяясь знаком ;. Промежуточные комментарии целесообразно писать с одной и той же позиции (колонки), например 30 или 35. Они должны указывать цель действия или объяснять логику моделирования при использовании того или иного оператора.

Цепи событий

Событие, связанное с конкретным требованием, находящимся в данный момент в системе, может сохраняться в одном из списков – цепей.

Цепь текущих событий (Current Events Chain – СЕС) – это цепь, включающая события для требований, которые планируются к продвижению в одном или нескольких операторах (блоках) в течение текущего или ближайшего модельного времени. В этой цепи находятся события, время наступления которых меньше или равно текущему модельному времени. События с меньшим временем связаны с перемещением тех требований, которые должны двигаться, но были заблокированы. В цепи текущих событий требования расположены в порядке убывания приоритета соответствующих событий, при равных приоритетах – в порядке поступления в цепь. Каждое событие требования в списке текущих событий может находиться либо в активном состоянии, либо в состоянии задержки. Если событие активно, то соответствующее требование может быть перемещено по системе. Если перемещение требования невозможно, например, из-за занятости канала обслуживания, то событие требования переводится в состояние задержки.

Как только завершается продвижение активного требования, начинается просмотр задержанных требований, и некоторые из них переводятся в разряд активных.

Эта процедура повторяется до тех пор, пока в цепи текущих событий не будут перемещены (обработаны) все активные требования. Модельному времени присваивается значение, равное времени наступления ближайшего из этих событий. Данное событие заносится в цепь текущих событий. Те из них, время которых равно текущему модельному времени, также переписываются в цепь текущих событий. Просмотр заканчивается, когда в цепи остаются события, время наступления которых больше текущего модельного времени.

Цепь пользователя (User Chain) – это цепь, включающая требования, которые не планируются к продвижению. Это требования, которые пользователь удалил из цепи текущих событий и поместил в цепь в качестве временно пассивных требований.

Цепь прерываний (Interrupt Chain) – это цепь выгружаемых требований. Когда требование входит в блочный оператор **PREEMPT** и канал обслуживания в настоящее время принадлежит другому требованию, монопольное использование

дается новому требованию. Старое требование помещается в цепь прерываний так, чтобы его монопольное использование могло быть восстановлено позже. Требования в одной или большем числе цепей прерывания могут все еще двигаться в процессе моделирования, однако их движение ограничено.

Цепь будущих событий (Future Events Chain, FEC) – это цепь, включающая события требований (транзактов), которые находятся в состоянии ожидания событий в будущем. В этой цепи находятся события, время наступления которых больше текущего модельного времени, то есть они должны произойти в будущем, но условия их наступления уже определены. Например, известно, что требование будет обслуживаться определенным каналом обслуживания 27 единиц времени.

Значение системного времени всегда равно намеченному времени последнего требования, которое будет принято из цепи будущих событий.

Блочные операторы **ADVANCE** и **GENERATE** служат для размещения требования в цепи будущих событий.

Эти блочные операторы берут приращение времени как операнд и вычисляют абсолютное время перед размещением требования в цепи будущих событий. Когда системное время достигает этого абсолютного времени, требование перемещается в цепь текущих событий так, чтобы его движение возобновилось в процессе моделирования.

Блочные операторы **PREEMPT** и **DISPLACE** можно использовать для удаления требования из цепи будущих событий. Такие требования можно переупорядочить, используя блочный оператор **ADVANCE**.

В случае, если планировщик должен выбрать новое активное требование и не может найти его в цепи текущих событий, ему нужно взять требование из цепи будущих событий. Удаление одного или большего числа требований всегда заставляет системное время продвигаться вперед. Связанные требования перемещаются из цепи будущих событий в цепь текущих событий в порядке их приоритетности. Требование в цепи текущих событий с самым высоким приоритетом тогда становится активным.

Цепь повторений (Retry Chain) – список требований, которые ожидают изменения состояния канала обслуживания.

Требования, которые подвергаются испытаниям, предшествующим входу в блок, помещаются в цепь повторений. Эти испытания происходят тогда, когда требование пытается войти в блочные операторы **GATE**, **TEST**, **TRANSFER ALL** или **TRANSFER BOTH**. Любое требование в цепи повторений ожидает изменения значения стандартного числового атрибута. Когда значение SNA изменяется, любое требование в цепи повторений объекта активизируется. Это приводит к изменению в цепи текущих событий. Когда требование становится активным, определенное условие испытания повторяется. Так как этот процесс занимает определенное компьютерное время без продвижения требования в модели, то неразумный выбор условий может привести к неэффективному моделированию. Если при повторении испытание прошло успешно, то требование вводится в следующий блок. Когда требование входит в блок, оно удаляется из всех цепей повторений автоматически.

Цепь задержки (Delay Chain) – приоритетная цепь требований, ожидающих монопольного использования канала обслуживания.

Если требованию, которое пытается войти в блочный оператор **SEIZE**, соответствующий определенному каналу обслуживания, отказано в использовании, оно помещается в цепь задержки того же канала обслуживания в приоритетном порядке. Точно так же требование, которое пытается в приоритетном режиме войти в блочный оператор **PREEMPT**, соответствующий определенному каналу обслуживания, и которому отказано в использовании, блокируется и помещается в цепь задержки канала обслуживания в приоритетном порядке.

Рассмотрим на простейшем примере процесс формирования цепей событий. Допустим, в моделируемой системе используются требования (транзакты) i типов, каждый из которых обрабатывается отдельным каналом обслуживания. Известны законы поступления требований в систему и длительность их обслуживания. Таким образом, в системе существует i параллельных независимых процессов – P_i . События, относящиеся к рассматриваемым процессам, соответственно обозначим как C_{ij} . Временная диаграмма работы моделируемой системы при обслуживании одного требования каждого типа показана на рис. 2.1.

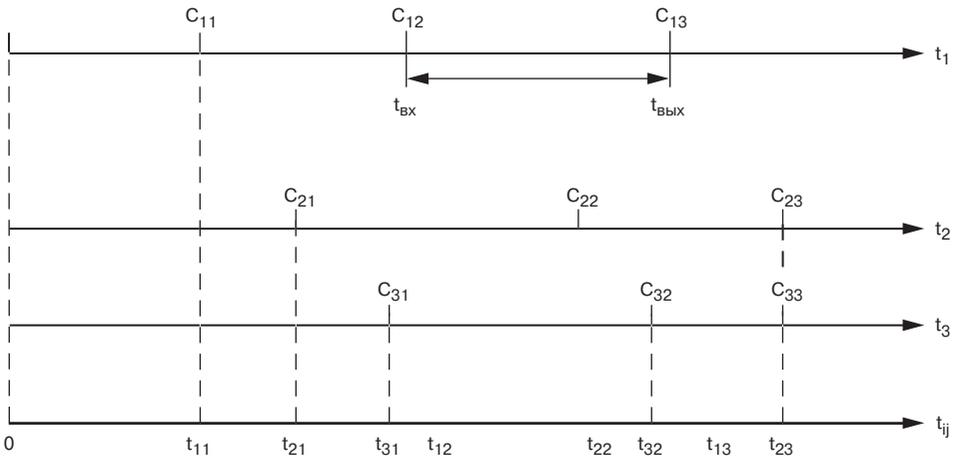


Рис. 2.1. Временная диаграмма работы системы

Для каждого процесса строится своя цепь событий, однако эти цепи являются общими для всей модели. Формирование цепи начинается с заполнения списка (цепи) будущих событий. Как указывалось выше, в эту цепь помещаются события, время которых превышает текущее значение модельного времени. Очевидно, что на момент заполнения списка время наступления прогнозируемых событий должно быть известно. На первом шаге $t_M = 0$, и в список будущих событий заносятся события C_{i1} : C_{11} , C_{21} , C_{31} ... Затем событие с наименьшим временем наступления, C_{11} , переносится в список текущих событий. Если одновременных с ним событий нет, то оно обрабатывается и исключается из списка текущих событий.

После этого снова корректируется список будущих событий и т.д., пока не закончится заданный интервал времени моделирования.

Динамика изменения содержания цепей будущих и текущих событий для нашего примера приведена в табл. 2.3.

Таблица 2.3

Время наступления события, t_{ij}	Список текущих событий, C_{ij}	Список будущих событий, C_{ij}
0	0	$C_{11'}$ $C_{21'}$ C_{31}
t_{11}	C_{11}	$C_{21'}$ $C_{31'}$ C_{12}
t_{21}	C_{21}	$C_{31'}$ $C_{12'}$ C_{22}
t_{31}	C_{31}	$C_{12'}$ $C_{22'}$ C_{32}
t_{12}	C_{12}	$C_{22'}$ $C_{32'}$ C_{13}
t_{22}	C_{22}	$C_{32'}$ $C_{13'}$ C_{23}
t_{32}	C_{32}	$C_{13'}$ $C_{23'}$ C_{33}
t_{13}	C_{13}	$C_{23'}$ C_{33}
t_{23}	$C_{23'}$ C_{33}	

Знание механизма формирования различных цепей событий существенно облегчает процесс отладки и модификации модели.

Основные составляющие системы GPSSW

Система GPSSW включает:

- различные типы объектов языка имитационного моделирования GPSS;
- язык программирования высокого уровня PLUS – небольшой, но мощный процедурный язык программирования;
- много блочных инструкций (свыше 50) и команд (свыше 25) и более 35 системных числовых атрибутов;
- строковые, математические, сервисные и другие процедуры;
- функции типовых распределений.

Программа в системе GPSSW представляет собой последовательность операторов языка программирования GPSS, операторов процедуры и эксперимента языка PLUS.

Типы объектов

В системе GPSSW имеются различные типы объектов, которые можно использовать при создании моделей:

- транзакты (Transactions) – обслуживаемые требования (заявки);
- блоки (Blocks);

- каналы (аппараты) обслуживания (Facilities);
- функции GPSS (GPSS Functions);
- логические переключатели (Logicswitches);
- матрицы (Matrixes);
- очереди (Queues);
- накопители (Storages);
- сохраняемые величины (Savevalues);
- табличные цепи пользователя (Tables User Chains);
- переменные (Variables);
- числовые группы (Numeric Groups);
- группы требований (Transaction Groups);
- генераторы случайных чисел (Random Number Generators).

За исключением требований (транзактов), объекты никогда не удаляются из моделирования. Однако некоторые типы объектов могут быть переопределены в интерактивном режиме: **STORAGE**, **TABLE**, **QTABLE**, **MATRIX** или **VARIABLE**.

PLUS-процедуры можно также переопределить в интерактивном режиме. Блоки не могут быть переопределены в системе GPSSW. Однако, используя блок **EXECUTE**, можно переопределить значение операнда A.

Операторы языка PLUS

Ниже перечислены операторы языка PLUS:

- **BEGIN** (Начать);
- **DO...WHILE** (Выполнять...до тех пор, пока);
- **END** (Закончить);
- **EXPERIMENT** (Эксперимент);
- **GOTO** (Перейти к);
- **IF...THEN...ELSE...** (Если...то...иначе...);
- **CALL** (Вызвать процедуру);
- **RETURN** (Вернуть);
- **TEMPORARY** (Временно).

Строковые процедуры

Система GPSSW имеет богатую встроенную библиотеку процедур для создания и изменения строк:

- **Align** (Выворнуть) – вернуть копию одной строки, помещенной в другую и выровненной по правому краю;
- **Catenate** (Связать) – вернуть копию двух строк, объединенных в одну;
- **Copies** (Копии) – создать строку из нескольких копий строки;
- **Datatype** (Строка данных) – вернуть строку, обозначающую тип данных аргумента;
- **Find** (Найти) – вернуть смещение одной строки в другой;

- **Left** (Левый) – вернуть копию подстроки, начинающейся слева;
- **Length** (Длина) – вернуть указанное число символов в строке;
- **Lowercase** (Нижний регистр) – вернуть представление строки в нижнем регистре;
- **Place** (Разместить) – разместить одну строку в другой с выравниванием по левому краю;
- **Polycatenate** (Связать) – вернуть копию одной или более строк, объединенных в одну;
- **Right** (Вправо) – вернуть копию подстроки, начинающейся справа;
- **String** (Строка) – преобразовать элемент данных в строковый эквивалент;
- **StringCompare** (Сравнение строки) – вернуть результат целого числа после сравнения строки;
- **Substring** (Подстрока) – вернуть копию подстроки аргумента строки;
- **Trim** (Вырезать) – удалить передние и остаточные пробелы;
- **Uppercase** (Верхний регистр) – вернуть строку в верхнем регистре;
- **Value** (Значение) – вернуть числовой эквивалент строки;
- **Word** (Слово) – вернуть копию одного из слов в строке.

Математические процедуры

Система GPSSW включает небольшую библиотеку математических процедур:

- **ABS** – абсолютное значение;
- **ATN** – арктангенс;
- **COS** – косинус;
- **INT** – целое;
- **EXP** – экспонента;
- **LOG** – натуральный логарифм;
- **SIN** – синус;
- **SQR** – квадратный корень;
- **TAN** – тангенс.

Сервисные процедуры

Система GPSSW имеет дополнительные процедуры для требований, находящихся в очереди:

- **QueryXNExist** – определяет существование требования;
- **QueryXNParameter** – восстанавливает значение параметра требования;
- **QueryXNAssemblySet** – восстанавливает набор ансамбля требования;
- **QueryXNPriority** – восстанавливает приоритет требования;
- **QueryXNM1** – восстанавливает время метки требования.

В систему GPSSW включены также новые сервисные процедуры для поддержки экспериментов:

- **DoCommand** – транслирует командную строку в глобальном контексте и посылает объекту моделирования;
- **ANOVA** – выполняет дисперсионный анализ.

Функции типовых распределений вероятностей

Система GPSSW включает встроенные функции многочисленных типовых распределений вероятностей:

- Beta – бета-распределение;
- Binomial – биномиальное распределение;
- Discrete Uniform – дискретное равномерное распределение;
- Exponential – экспоненциальное распределение;
- Extreme Value A;
- Extreme Value B;
- Gamma – гамма-распределение;
- Geometric – геометрическое распределение;
- Inverse Gaussian – обратное гауссово распределение;
- Inverse Weibull – обратное распределение Вейбула;
- Laplace – распределение Лапласа;
- Logistic – логистическое распределение;
- LogLaplace – логлапласово распределение;
- LogLogistic – логлогистическое распределение;
- LogNormal – логнормальное распределение;
- Negative Binomial – отрицательное биномиальное распределение;
- Normal – нормальное распределение;
- Pareto – распределение Парето;
- Pearson Type V – распределение Пирсона типа V;
- Pearson Type VI – распределение Пирсона типа VI;
- Poisson – пуассоновское распределение;
- Triangular – треугольное распределение;
- Uniform – равномерное распределение;
- Weibull – распределение Вейбула.

Основные этапы моделирования в системе GPSSW

Система GPSSW достаточно проста в изучении и универсальна в применении. Эффективное использование системы предусматривает выполнение ряда этапов:

1. Постановка задачи.
2. Выявление основных особенностей.

3. Создание имитационной модели процесса.
4. Представление имитационной модели в системе GPSSW.
5. Моделирование системы.

Рассмотрим подробнее каждый из этапов.

Постановка задачи

Постановка задачи – это этап содержательного описания процесса моделирования с указанием числовых значений работы тех или иных составляющих исследуемого процесса. На этом этапе указывается, что необходимо определить. Исследование самых разнообразных систем и процессов методом имитационного моделирования заключается в определении происходящих в системе событий. Чтобы облегчить это определение, целесообразно первоначально графически изобразить процесс функционирования системы и выделить в нем характерные события. Поведение требования в моделируемой системе не является независимым, оно обуславливается событиями, в которых принимают участие и другие требования. Сам же процесс имитации должен отображать хронологию событий в последовательности, имеющей место в реальном процессе.

Основными элементами системы моделирования GPSSW являются стандартные компоненты реальных систем и процессов: каналы обслуживания (приборы), очереди, накопители, переключатели, требования и др. Достаточный набор подобных компонентов и программная реализация их функционирования позволяют моделировать самые разнообразные системы и процессы. Использование же элементами системы количественных параметров в виде констант, переменных, функций, сохраняемых величин позволяет исследовать большое многообразие систем.

Выявление основных особенностей

На этом этапе определяют:

- характер функционирования системы (непрерывная или дискретная система);
- потоки поступления требований в систему (регулярные, случайные или смешанные);
- число требований, поступающих в один момент времени (ординарный или неординарный поток);
- характер взаимодействия смежных требований в потоке (с последствием или нет);
- характер поведения требований, поступающих в систему на обслуживание (с отказами, с ограниченным ожиданием или с ожиданием без ограничения);
- способ выбора требований на обслуживание (с приоритетом, по мере поступления, случайно, последний обслуживается первым). Иногда в таком случае говорят о дисциплине обслуживания;
- время обслуживания требований (детерминированное или случайное);

- число каналов обслуживания (одноканальная или многоканальная система);
- число фаз обслуживания (однофазная или многофазная система);
- однородность требований, поступающих на обслуживание (однородные и неоднородные);
- связь выходных и входных потоков (замкнутая или разомкнутая система).

На этом этапе следует определить продолжительность моделирования системы, выбрать способы представления тех или иных данных. Здесь же осуществляется декомпозиция сложного моделируемого процесса на ряд простых сегментов, чтобы их описание могло быть выполнено достаточно просто и наглядно. При этом следует максимально использовать встроенные процедуры, имеющиеся в системе. Это значительно упрощает процесс построения и исследования модели.

Создание имитационной модели процесса

На этом этапе необходимо подробно изложить все действия, связанные с функционированием той или иной системы, возможно, с помощью циклических процедур. Надо указать, какая входная информация требуется и как она будет использована соответствующими операторами системы.

Процесс моделирования начинается с создания исходной модели на языке имитационного моделирования GPSS. Самый простой способ начать моделирование – это использование, а затем и модернизация существующей модели. В папке C:\Program Files\Minuteman Software\GPSS World Student Version\Samples Models имеется большой набор типовых примеров для обучения моделированию. Инструкции к моделям могут находиться в нескольких текстовых объектах. Процесс моделирования начинается с открытия текстового объекта (файла), если он присутствует в модели. Для обращения к текстовому объекту в модели используется оператор **INCLUDE**, далее следует имя текстового файла с расширением .txt, записываемое в двойных кавычках.

Исходная модель – это определенный набор (список) операторов модели. Оператором модели может быть оператор языка имитационного моделирования GPSS, процедура языка программирования PLUS или определение PLUS-эксперимента.

На втором этапе создается имитационная модель с помощью транслятора системы GPSSW. Результатом трансляции программы является объект моделирования блочной структуры.

Трансляцию исходной модели можно выполнить двумя способами.

Первый способ:

- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** или **Retranslate**. Появится окно **JOURNAL**, в котором указываются дата и время начала и окончания трансляции исходной модели.

Второй способ: нажмите комбинацию клавиш **Ctrl+Alt+S** для трансляции или **Ctrl+Alt+R** для перетрансляции программы.

Если в процессе трансляции в программе обнаруживаются синтаксические ошибки, то их можно исправить. Для этого:

- щелкните по пункту **Search** (Поиск) главного меню. Появится выпадающее меню;
- щелкните по пункту **Next Error** (Следующая ошибка).

После выполнения этих действий курсор мыши каждый раз устанавливается на очередной ошибке. Эти действия проводятся до тех пор, пока не будут устранены все выявленные ошибки. При этом курсор мыши циклически проходит список всех выявленных транслятором ошибок.

Моделирование системы

Как только все синтаксические ошибки будут устранены, можно послать оттранслированную модель на выполнение. Для этого:

- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **START**. Появится диалоговое окно **Start Command**, в котором можно определить режим моделирования. После определения режима моделирования щелкните по кнопке **ОК**. Появится окно **JOURNAL**, в котором указываются дата и время начала и окончания процесса моделирования оттранслированной модели, а затем – окно **REPORT** с результатами имитационного моделирования.

Можно послать инструкции для моделирования тремя способами:

- используя наиболее общие команды, перечисленные в выпадающем меню пункта **Command** главного меню;
- щелкнув по пункту **Custom** в выпадающем меню пункта **Command** главного меню. Появится диалоговое окно **Simulation Command**, где можно напечатать любую инструкцию для моделирования, даже PLUS-процедуру, и послать ее существующему объекту моделирования;
- используя набор функциональных клавиш с собственным набором команд. Это можно сделать через пункты главного меню **Edit** и выпадающего меню **Settings** в главном окне системы. После нажатия соответствующей функциональной клавиши объекту моделирования посылается определенная команда.

Комплексные процедуры и длинные списки команд могут также быть посланы с помощью команды **INCLUDE** (Включить), связанной с предварительно созданным текстовым объектом.

Команды управления моделированием могут быть вставлены в имитационную модель, или их можно интерактивно ввести в процессе моделирования. Для этого:

- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **Custom**. Появится диалоговое окно **Simulation Custom**;
- введите в поле диалогового окна нужную команду управления и щелкните по кнопке **ОК**.

Пункты (команды) выпадающего меню пункта **Command** главного меню становятся активными после транслирования исходной модели.

Команда **START** используется для запуска процесса моделирования. Эта команда может быть введена в модели или в интерактивном режиме.

Многократное моделирование может быть выполнено с использованием последовательности управляющих операторов **RESET**, **CLEAR** и **START**.

Остановить процесс моделирования можно несколькими способами:

- щелкните по пункту **Command** главного меню, а затем – по пункту **Halt** (Остановить) выпадающего меню;
- нажмите комбинацию клавиш **Ctrl+Alt+H**;
- нажмите на функциональную клавишу **F4**;
- щелкните по пункту **Command** главного меню, а затем – по пункту **Custom** выпадающего меню. Появится диалоговое окно **Simulation Custom**. Введите в поле диалогового окна команду **HALT** и щелкните по кнопке **OK**;
- используйте командный файл.

Вы можете возобновить моделирование, используя команду **CONTINUE** или **STEP**.

Интерактивные команды **HALT** и **SHOW** выполняются в момент их ввода, а другие команды ставятся в очередь. Они помещаются в конце списка команд, которые еще не были закончены к моменту ввода. Когда оператор процедуры языка PLUS посылается в интерактивном режиме, процедура регистрируется в пределах моделирования. После этого процедура может быть вызвана из любого PLUS-выражения. Если процедура с тем же именем уже существует в пределах моделирования, она переопределяется.

Модернизация исходной модели

Если возникает необходимость в модернизации исходной модели, то необходимо активизировать окно с исходной моделью. Для этого:

- щелкните по пункту **Window** главного меню. Появится выпадающее меню;
- щелкните по имени файла, содержащего исходную модель.

Можно вернуться к окну с исходной моделью и другим способом. Если часть этого окна видна в главном окне системы, то щелкните по нему мышью – и оно всплывет на передний план.

Отладка модели

Во время отладки модели можно использовать многочисленные средства визуализации, имеющиеся в системе GPSSW. Среди 10 динамических окон есть окна **PLOT** и **EXPRESSIONS**, которые позволяют визуализировать оценку любого выражения PLUS, поскольку оно изменяется динамически. Окно **TABLE** дает возможность визуализировать сходимость гистограмм. Кроме того, имеется 6 дополнительных

окон для представления снимков. Они предназначены для профессионалов моделирования, нуждающихся в детальной информации о состоянии моделирования изнутри. Система GPSSW имеет большой набор стандартных сообщений. Они описывают конечное состояние и/или результаты моделирования. Можно изменять содержание стандартных сообщений, редактируя установки во время моделирования объекта. Промежуточные сообщения часто используются во время отладки модели.

Новый высокоэффективный транслятор системы работает как минимум на два порядка быстрее, чем его предшественник. Когда транслятор обнаруживает одну или большее число ошибок в модели, то создается круговой список сообщений об ошибках, к которым можно обращаться из моделируемой системы. Для этого используйте команды **Next Error** (Следующая ошибка) или **Previous Error** (Предыдущая ошибка) в меню, выпадающем при щелчке по пункту **Search** (Поиск) главного меню. После каждой остановки в списке на той или иной ошибке выдается сообщение о ней в строке состояния в нижней части главного окна системы. При этом курсор размещается перед синтаксическим элементом, который вызвал ошибку. Это значительно облегчает устранение ошибок. Для устранения ошибок используется текстовый редактор.

Если ошибка произошла в текстовом объекте, вводимом оператором **INCLUDE** (Включить), можно вставить исправленные операторы в новый временный модельный объект и снова его оттранслировать.

Снимки и динамические окна

Система GPSSW имеет многочисленные окна, которые можно разделить на две категории: окна снимков моделирования и динамические окна моделирования. Диалоговое окно – это динамическое окно, вид которого зависит от тех или иных изменений в модели или моделировании.

Все окна открываются с помощью пунктов **Simulation Snapshot** (Снимок моделирования) и **Simulation Window** (Окно моделирования) в выпадающем меню пункта **Window** главного меню. При этом появляются всплывающие меню, в которых можно выбрать те или иные окна моделирования.

Снимок – это отдельное изображение, которое делается в заданный момент времени в процессе моделирования системы.

Можно получить снимок следующих объектов:

- любого требования в системе;
- цепи текущих событий;
- цепи будущих событий;
- числовых групп;
- цепей пользователя;
- группы требований.

Динамические окна модифицируются в процессе моделирования. Они могут быть обновлены, когда вы транслируете модель или вызываете управляющую команду **RESET** (Сброс).

Вы можете открыть интерактивное представление:

- блочной структуры модели;
- произвольного списка выражений;
- каналов обслуживания;
- логических переключателей;
- любого перекрестного раздела матричного объекта;
- графиков выражений;
- объектов очереди;
- объектов сохраняемых величин;
- объектов накопителей;
- любого объекта таблицы.

Рекомендуется использовать одновременно только несколько динамических окон в процессе моделирования. Но их можно открыть значительно больше, когда моделирование приостановлено.

Для вывода содержания любого окна на печать щелкните по пункту **File** главного меню, а затем – по пункту **Print** в выпадающем меню. Появится диалоговое окно **Print**, в котором надо ввести соответствующие установки.

Большинство динамических окон моделирования имеют дополнительную панель, называемую инструментальной панелью отладки, которая позволяет отлаживать модель, используя соответствующие кнопки (рис. 2.2).

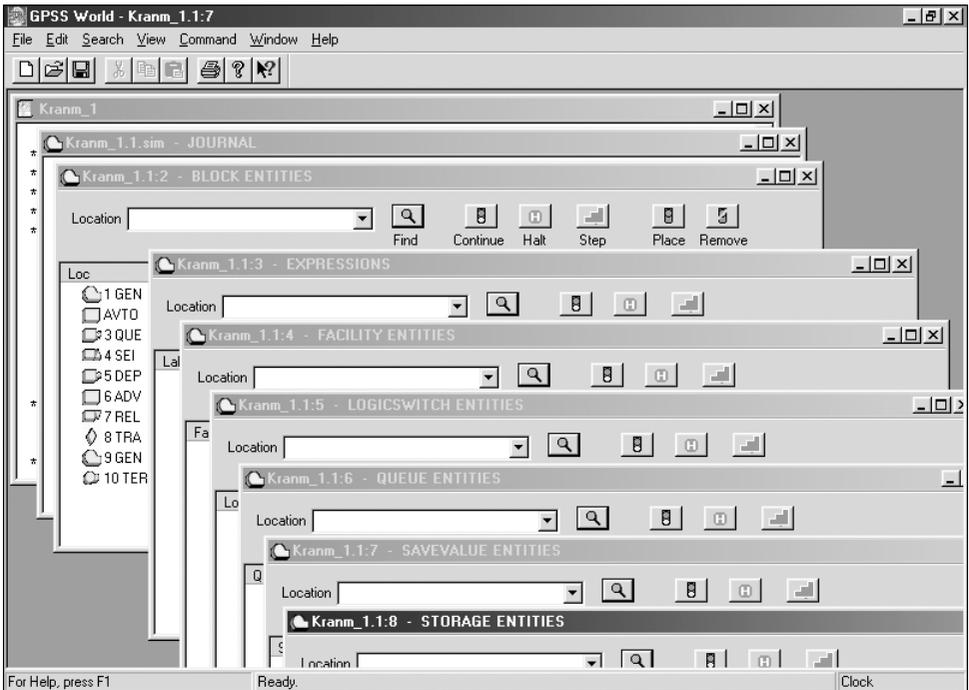


Рис. 2.2. Панели отладки в динамических окнах системы GPSSW

Динамическое окно **BLOCK ENTITIES** (Блочные элементы) имеет следующие кнопки на панели отладки:

- **Find** (Найти);
- **Continue** (Продолжить) – продолжить процесс моделирования после его остановки;
- **Halt** (Остановить) – прервать процесс моделирования;
- **Step** (Шагнуть) – сделать один шаг и остановить процесс моделирования;
- **Place** (Разместить) – разместить условие остановки в указанном блоке;
- **Remove** (Удалить) – удалить условие остановки в указанном блоке.

Щелчком мыши выделите блок в динамическом окне **BLOCK ENTITIES**. Затем щелкните по кнопке **Place**, чтобы разместить условие остановки, которое прерывает моделирование, когда требование пытается войти в помеченный блок. Если блок выбран и вы щелкаете по кнопке **Remove**, то первоначальное условие остановки удаляется. Используйте кнопку **Halt**, чтобы послать команду остановки объекту моделирования. Допустим, мы выделили блок **QUEUE** щелчком мыши, а затем щелкнули по кнопке **Place**. Далее выделим блок **ADVANCE** щелчком мыши и снова щелкнем по кнопке **Place**. Таким образом, мы разместили две остановки в модели. Теперь можно просмотреть набор условий остановки в окне снимка **Stop Conditions** (Условия остановки). Для этого:

- щелкните по пункту **Window** главного меню. Появится выпадающее меню;
- щелкните по пункту **Simulation Snapshot** (Снимок моделирования) выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **User Stops** (Остановки, установленные пользователем) во всплывающем меню. Появится диалоговое окно **Stop Conditions**. В нашем примере это окно снимка будет выглядеть так, как показано на рис. 2.3.

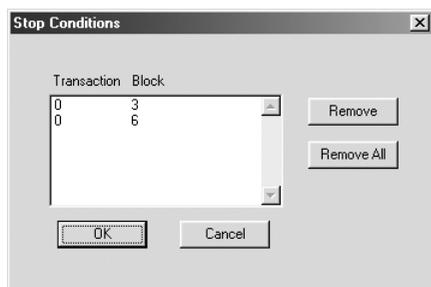


Рис. 2.3. Окно снимка **Stop Conditions**

Окно снимка **Stop Conditions** дает снимок набора всех активных условий остановки требования. Условие остановки определено номерами требования и блочного элемента. 0 означает любое требование. Остановки добавляются к списку с помощью команды **Place** на инструментальной панели отладки. Можно удалить выбранную остановку или все, используя соответственно кнопки **Remove** или **Remove All** (Удалить все) в окне снимка **Stop Conditions**. Или можно использовать кнопку **Remove** на инструментальной панели отладки.

Инициализация элементов

Команда **INITIAL** (Инициализировать) предназначена для инициализации элементов матрицы **MX**<имя>, логического переключателя **LS**<имя> и сохраняемой величины **X**<имя>. Команда **INITIAL** записывается в таком виде:

```
INITIAL A, B
```

Поле операнда **A** содержит инициализируемый объект, а поле операнда **B** – значение, которое будет присвоено указанному объекту. Это значение может быть следующим: пустое, число, строка, имя или **UNSPECIFIED** (Неопределенное).

Если операнд **B** не используется, то присваивается значение 1. Если операнд **B** имеет ключевое слово **UNSPECIFIED**, тогда сохраняемая величина (**Savevalue**), матрица (**Matrix**) или матричный элемент помещаются в состояние неопределенности, чтобы указать отсутствие данных в матрице результатов.

Если операнд **A** определяет логический переключатель (**Logicswitch**), то могут быть присвоены только два значения: 0 или 1.

Если операнд **B** явно определен как 0, присваивается значение 0, в противном случае – 1. Опция **UNSPECIFIED** не может использоваться с **LS**-классом **SNA**, так как логический переключатель не имеет неопределенного состояния.

Если операнд **A** определяет имя матрицы, то все элементы в матрице помещаются в состояние, обозначенное операндом **B**. Ниже даны примеры использования команды **INITIAL**.

```
INITIAL X$How_you, "How are you. "
```

В этом примере команда **INITIAL** присваивает сохраняемой величине под именем **How_you** строковую константу "How are you. ".

```
INITIAL MX$Zapas (Month_6, Day_21), 56
```

В этом примере команда **INITIAL** присваивает значение 56 элементу матрицы под именем **Zapas** с номером строки **Month_6** и номером столбца **Day_21**. Именам **Month_6** и **Day_21** должны быть предварительно присвоены соответствующие целые числа с помощью команды **EQU**.

```
INITIAL Matrix_1, UNSPECIFIED
```

Эта команда готовит предварительно определенный матричный объект по имени **Matrix_1** к использованию в эксперименте, в котором могут отсутствовать данные.

Определение матрицы

Команда **MATRIX** определяет матрицу. Она записывается в таком виде:

```
<имя> MATRIX A, B, C, D, E, F, G
```

В поле метки указывается имя матрицы, и далее:

- в поле операнда **A** – неиспользованное поле (для совместимости с ранними версиями GPSS);
- в поле операнда **B** – максимальный индекс элементов в первой размерности;

- в поле операнда C – максимальный индекс элементов во второй размерности;
- в поле операнда D (необязательно) – максимальный индекс элементов в третьей размерности и т.д.

Например:

```
Matr1 MATRIX ,10,5
```

В этом примере мы определяем матрицу по имени Matr1. Поле операнда A не используется. В операнде B указано число строк, а в операнде C – число столбцов в матрице. Матрица в системе GPSSW может быть шестиразмерной. Однако только к первым двум размерностям можно обращаться в блоке **MSAVEVALUE**. С помощью процедур языка программирования PLUS можно обращаться ко всем элементам любой матрицы.

Матрица никогда не удаляется из моделирования, однако она может быть переопределена другой командой **MATRIX**. С начала создания матрицы или при использовании команды **CLEAR ON** всем элементам матрицы присваивается 0. Однако можно использовать команду **INITIAL**, чтобы присвоить элементам состояние UNSPECIFIED.

Для обращения к элементу матрицы используется следующий стандартный числовой атрибут:

```
MX<имя> (m, n)
```

где m – номер строки; n – номер столбца матрицы.

Матрицу можно просматривать в интерактивном окне **Matrix Window**. Это окно показывает двуразмерный раздел любой матрицы.

Моделирование в интерактивном режиме

Рассмотрим теперь одну из наиболее интересных диалоговых особенностей системы GPSSW – возможность включения в модель любого оператора в интерактивном режиме в процессе моделирования, кроме оператора **GENERATE**, который является единственным исключением.

Для моделирования в интерактивном режиме, прежде всего, мы должны иметь активное требование, которое можно получить несколькими способами:

- использовать команду **STOP** для прерывания процесса моделирования;
- использовать функциональную клавишу, обеспечивающую вызов команды **STOP**;
- подождать до тех пор, пока процесс моделирования не прекратится.

Воспользуемся командой **STOP**. Эта команда регистрирует состояние моделируемой системы, когда любое требование пытается войти в любой блок.

Рассмотрим использование интерактивного режима моделирования на конкретном примере.

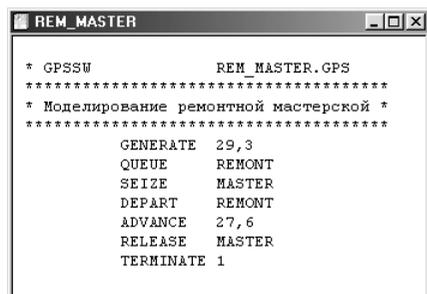
Постановка задачи

Допустим, что нам надо исследовать работу некоей мастерской. В мастерскую поступают для ремонта те или иные изделия с интервалом 29 ± 3 единицы времени с равномерным законом распределения, то есть в интервале [26–32] единицы времени. Известно время ремонта поступающих в систему изделий. Оно колеблется в пределах 27 ± 6 единиц времени с равномерным законом распределения, то есть в интервале [21–33] единицы времени. Требуется определить параметры функционирования системы.

Создание имитационной модели

Допустим, что имитационная модель выглядит так, как показано на рис. 2.4.

Первые четыре строки модели представляют собой комментарии к нашей задаче. Вид этой части модели зависит от вкуса разработчика. Однако желательно представить здесь информацию о том, с помощью какой системы проводится моделирование, в каком файле будет храниться модель и что она выполняет.



```

REM_MASTER
* GPSSW          REM_MASTER.GPS
*****
* Моделирование ремонтной мастерской *
*****
GENERATE 29,3
QUEUE   REMONT
SEIZE   MASTER
DEPART  REMONT
ADVANCE 27,6
RELEASE MASTER
TERMINATE 1
    
```

Рис. 2.4. Имитационная модель ремонтной мастерской REM_MASTER

Транслирование модели

Для транслирования модели выполните следующие действия:

- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню. Появится окно **JOURNAL** с сообщениями о результатах трансляции. Сообщения будут выглядеть, например, так, как показано на рис. 2.5.



```

REM_MASTER.1.sim - JOURNAL
12/20/01 08:30:35 Model Translation Begun.
12/20/01 08:30:36 Ready.
    
```

Рис. 2.5. Окно **JOURNAL** для имитационной модели REM_MASTER

Поскольку на данный момент активным становится окно **JOURNAL**, то оно перекрывает окно модели.

Открытие диалогового окна **BLOCK ENTITIES**

Для открытия диалогового окна **BLOCK ENTITIES** выполните следующие действия:

- щелкните по пункту **Window** главного меню. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** (Окно моделирования) выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Blocks Window** (Окно блоков) всплывающего меню. Появится диалоговое окно **BLOCK ENTITIES** (Блочные элементы), показанное на рис. 2.6;

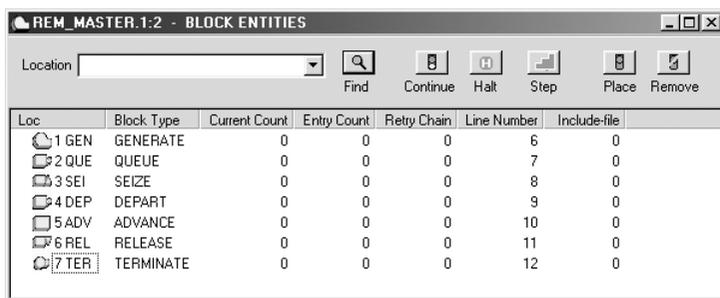


Рис. 2.6. Диалоговое окно **BLOCK ENTITIES**

- щелкните дважды по первому блоку – **GENERATE**;
- щелкните по кнопке **Place** (Разместить) на панели инструментов в верхней части диалогового окна **BLOCK ENTITIES**.

Моделирование системы

Для моделирования системы выполните следующие действия:

- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **START**. Появится диалоговое окно **Start Command**, показанное на рис. 2.7;



Рис. 2.7. Диалоговое окно **Start Command**

- вместо 1 введите 1000 – это число изделий, которые предполагается отремонтировать в мастерской;
- щелкните по кнопке **ОК**.

Просмотр сообщений

Чтобы просмотреть сообщения в окне **JOURNAL**, выполните следующие действия:

- щелкните по пункту **Window** главного меню. Появится выпадающее меню, показанное на рис. 2.8.

В нижней части выпадающего меню сообщается, что на данный момент в системе GPSSW открыты три окна:

- **1 REM_MASTER** – окно модели под именем REM_MASTER;
- **REM_MASTER.1.sim – JOURNAL** для модели под именем REM_MASTER. Следует обратить внимание, что сообщения, появляющиеся в окне **JOURNAL**, имеют расширение .sim;
- **REM_MASTER.1:2 – BLOCK ENTITIES** для модели REM_MASTER.

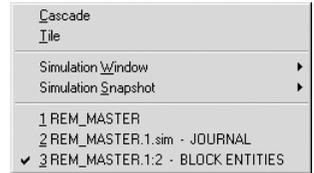


Рис. 2.8. Выпадающее меню пункта **Window** главного меню

Перед названием последнего окна стоит галочка. Значит, в данный момент оно является активным. Щелчком левой кнопки мыши можно активизировать любое открытое окно;

- щелкните по названию второго окна – **REM_MASTER.1.sim – JOURNAL**. Появится окно с сообщением об останове, показанное на рис. 2.9.

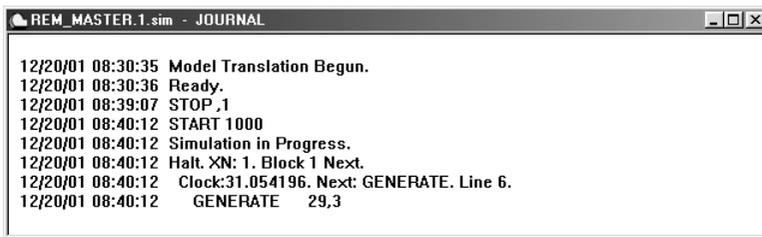


Рис. 2.9. Окно **JOURNAL** с сообщением об останове в модели REM_MASTER

Вернемся к диалоговому окну **BLOCK ENTITIES**, щелкнув по нему на видимой его части. Оно выйдет на передний план. Затем для создания активного требования:

- щелкните по первому блоку – **GENERATE** (Генерировать);
- щелкните по кнопке **Remove** (Удалить) на панели инструментов в верхней части диалогового окна.

Таким образом, мы создали активное требование. Теперь можно присвоить ему имя и значение. Для этого:

- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **Custom** (Заказать) выпадающего меню. Появится диалоговое окно **Simulation Command**, в котором введите оператор **ASSIGN** (Присвоить), например, в виде, представленном на рис. 2.10;
- щелкните по кнопке **OK**.

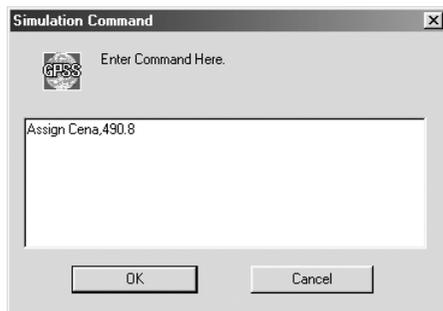


Рис. 2.10. Диалоговое окно **Simulation Command**

Активное требование теперь имеет параметр требования по имени *Cena*, содержащий заданное значение.

Для просмотра всех параметров этого требования, в том числе и вновь введенного параметра под именем *Cena*, выполните следующие действия:

- щелкните по пункту **Window** главного меню. Появится выпадающее меню;
- щелкните по пункту **Simulation Snapshot** (Снимок моделирования) выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **CEC Snapshot**. Появится окно **CURRENT EVENTS CHAIN SNAPSHOT** (Снимок цепи текущих событий);
- щелкните в нем по пиктограммам, расположенным слева, – по всем пиктограммам со знаком +. Тогда окно может выглядеть так, как показано на рис. 2.11.



Рис. 2.11. Снимок цепи текущих событий в модели **REM_MASTER**

Теперь закройте окно снимка СЕС в модели **REM_MASTER**, щелкнув мышью по кнопке закрытия (с крестиком), расположенной в верхнем правом углу окна.

Можно также посмотреть значение параметра активного требования с помощью команды **SHOW** (Показать). Для этого:

- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **SHOW**. Появится диалоговое окно **Show Command** (Показать команду), в котором введите имя активного требования – P\$Cena. Это будет выглядеть так, как изображено на рис. 2.12;

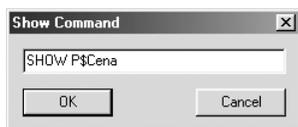


Рис. 2.12. Диалоговое окно **Show Command**

- щелкните по кнопке **OK**. Значение параметра P\$Cena появится в окне **JOURNAL** и в строке состояния в нижней части главного окна системы GPSSW;
- активизируйте окно **JOURNAL** с помощью выпадающего меню пункта **Window** главного меню – об этом говорилось выше. Появится окно с фиксацией введенного значения нового параметра требования, показанное на рис. 2.13.

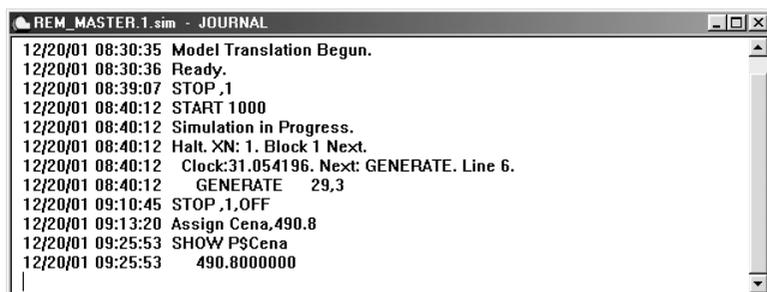


Рис. 2.13. Окно **JOURNAL** для модели **REM_MASTER**

Теперь попробуем использовать оператор **TRACE** (Трассировать). Для этого:

- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **Custom** выпадающего меню. Появится диалоговое окно **Simulation Command**, в котором введите оператор **TRACE**;
- щелкните по кнопке **OK**. Появится дополнительное сообщение в окне **JOURNAL** (Журнал) с фиксацией времени и номера требования.

Это действие имеет тот же самый эффект, как если бы активное требование пропустили через блок **TRACE**. Индикатор следа для активного требования теперь включен. Чтобы продолжить моделирование:

- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **Continue** (Продолжить) выпадающего меню. Процесс моделирования продолжится. Появится измененное окно **JOURNAL**, показанное на рис. 2.14.

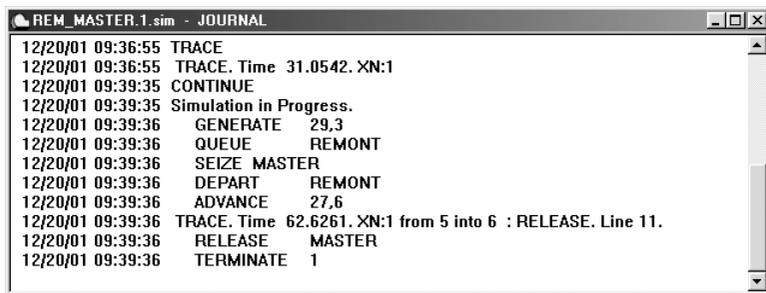


Рис. 2.14. Измененное окно **JOURNAL** для модели **REM_MASTER**

Заметьте, что в окне **JOURNAL** каждый вход в блок **TRACE** кончается следом, потому что индикатор следа требования включен.

Система GPSSW может также вызывать командные файлы, состоящие из инструкций **INITIAL** (Инициализировать) или других команд. Это можно делать или в интерактивном режиме, вводя команду **INCLUDE** (Включить) в диалоговом окне **Custom Command** (Заказные команды), или помещая инструкции в модель.

Допустим, наша программа под названием **REM_TEL2.GPS** будет выглядеть так, как изображено на рис. 2.15.

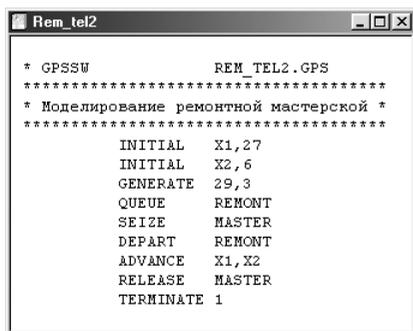


Рис. 2.15. Программа моделирования **Rem_tel2**

Но мы хотим изменить значения двух сохраняемых величин: **X1** и **X2**. Для этого мы должны создать текстовый файл, в котором будут находиться их новые значения:

- щелкните по кнопке **New** (Создать) – первой кнопке на стандартной панели инструментов с изображением чистого листа. Появится диалоговое окно **Новый документ**, показанное на рис. 2.16;
- щелкните по опции **Text File** (Текстовый файл), а затем – по кнопке **ОК**. Появится текстовое окно под названием **Untitled Text File 1** (Неименованный текстовый файл 1);
- введите в текстовом окне необходимые начальные значения так, как показано на рис. 2.17;



Рис. 2.16. Диалоговое окно **Новый документ**

- щелкните по кнопке **Save** (Сохранить) – третьей кнопке с изображением дискеты на стандартной панели инструментов. Появится диалоговое окно **Сохранить как**, показанное на рис. 2.18;
- введите в текстовом поле **Имя файла:** имя файла, например Rem_tel2;
- выберите в раскрывающемся списке **Тип файла:** – Text File (*.txt);
- щелкните по кнопке **Сохранить**.

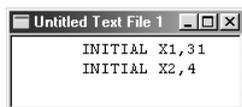


Рис. 2.17
Текстовое окно
Untitled Text File 1

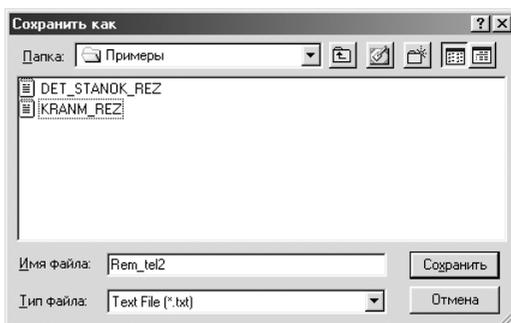


Рис. 2.18. Диалоговое окно **Сохранить как**

Теперь можно переходить к использованию текстового файла. Для этого:

- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню. Появится окно **JOURNAL** с сообщениями о результатах трансляции;
- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **Custom** выпадающего меню. Появится диалоговое окно **Simulation Command**, показанное на рис. 2.19;
- щелкните по кнопке **ОК**. Появится окно **JOURNAL** с сообщением о результатах включения текстового файла. Сообщение будет выглядеть так:

```
INCLUDE "Rem_tel2.txt"
```

- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **Retranslate** (Перетранслировать).

После этого можно переходить к моделированию процесса с уже новыми начальными значениями исходных величин. Для этого:

- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **START** выпадающего меню. Появится диалоговое **Start Command**;



Рис. 2.19. Диалоговое окно *Simulation Command*

- введите нужное число заказчиков – пользователей мастерской, – например 100, и щелкните по кнопке **ОК**. Появятся результаты моделирования с новыми начальными данными.

Можно текстовый файл ввести непосредственно в исходную модель. Это будет выглядеть так, как показано на рис. 2.20.

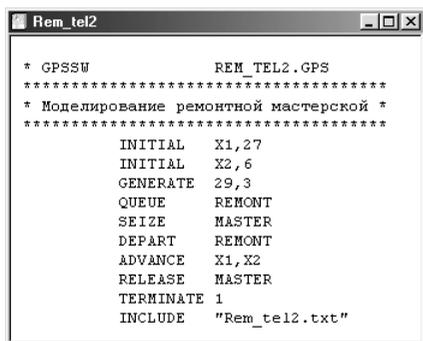


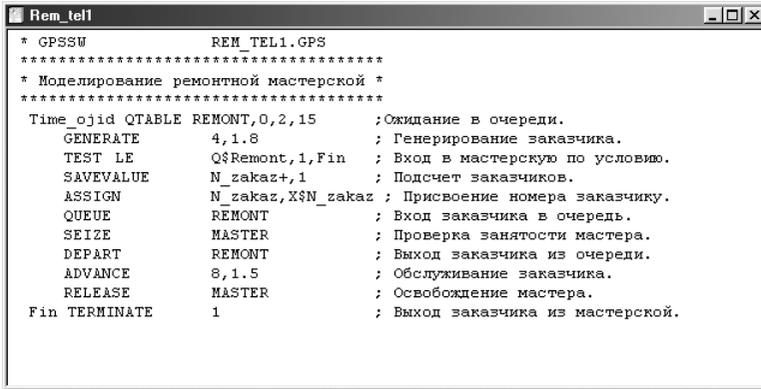
Рис. 2.20. Имитационная модель *Rem_tel2* с новыми начальными значениями

Отладка модели

Рассмотрим на конкретном примере отладку модели под названием REM_TEL1.GPS. Для этого:

- щелкните по пункту **File** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **New** выпадающего меню. Появится диалоговое окно **Новый документ**;
- выделите пункт **Model** и щелкните по кнопке **ОК**. Появится окно модели, в котором введите программу, показанную на рис. 2.21.

Во введенной программе в самом начале добавлен оператор **QTABLE**, например, с меткой *time_ojid* (Время ожидания) для сбора информации и построения соответствующей гистограммы функционирования очереди под именем, скажем, *Remont*.



```

* GPSSW          REM_TEL1.GPS
*****
* Моделирование ремонтной мастерской *
*****
Time_ojid QTABLE REMONT,0,2,15 ;Ожидание в очереди.
GENERATE 4,1.8 ; Генерирование заказчика.
TEST LE Q$Remont,1,Fin ; Вход в мастерскую по условию.
SAVEVALUE N_zakaz+,1 ; Подсчет заказчиков.
ASSIGN N_zakaz,X$N_zakaz ; Присвоение номера заказчику.
QUEUE REMONT ; Вход заказчика в очередь.
SEIZE MASTER ; Проверка занятости мастера.
DEPART REMONT ; Выход заказчика из очереди.
ADVANCE 8,1.5 ; Обслуживание заказчика.
RELEASE MASTER ; Освобождение мастера.
Fin TERMINATE 1 ; Выход заказчика из мастерской.

```

Рис. 2.21. Окно имитационной модели ремонтной мастерской *Rem_tel1*

Чтобы вызвать окно для представления имитационной модели в системе GPSSW, можно также нажать комбинацию клавиш **Ctrl+Alt+S**.

Предварительно назначим чаще всего используемым командам соответствующие функциональные клавиши для данной модели. Для этого:

- щелкните по пункту **Edit** (Правка) главного меню системы. Появится выпадающее меню;
- щелкните по пункту **Settings** (Установки) выпадающего меню. Появится соответствующее диалоговое окно для данной программы;
- щелкните по вкладке **Function Keys** (Функциональные клавиши), а затем – по кнопке **OK**. Появится соответствующая вкладка;
- введите в текстовые поля **F8** и **F9** соответственно команды `SHOW P$N_zakaz` и `SHOW X$N_zakaz`. Это будет выглядеть так, как показано на рис. 2.22;
- щелкните по кнопке **OK**.



Рис. 2.22. Окно **SETTINGS** для модели ремонтной мастерской *Rem_tel1*

Эти действия обеспечили возможность вызова двух команд – `SHOW P$N_zakaz` и `SHOW X$N_zakaz` – с помощью соответственно клавиш **F8** и **F9**. Это поможет сэкономить много времени при моделировании и использовании различных исходных данных.

Теперь перейдем к этапу трансляции модели. Для этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню.

Затем откроем окно **BLOCK ENTITIES** (Блочные элементы). Для этого:

- щелкните по пункту **Window** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Blocks Window** во всплывающем меню. Появится диалоговое окно **BLOCK ENTITIES**. В нашей задаче оно выглядит так, как показано на рис. 2.23.

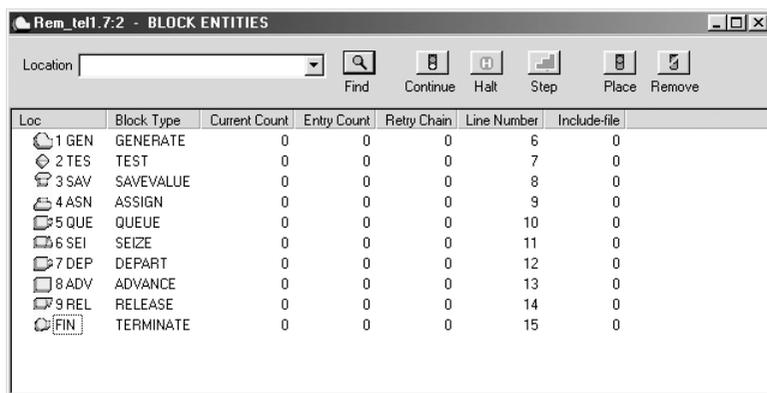


Рис. 2.23. Диалоговое окно **BLOCK ENTITIES** для модели *Rem_tel1*

Теперь давайте поместим условие остановки требования (заказчика) на блоке 5. Для этого:

- щелкните по пункту **Command** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **Custom** (Заказать) выпадающего меню. Появится диалоговое окно **Simulation Command**;
- введите в текстовом поле `STOP 5` (Остановить на блоке 5). Это будет выглядеть так, как изображено на рис. 2.24;
- щелкните по кнопке **ОК**. Появится окно **JOURNAL**.



Рис. 2.24. Диалоговое окно *Simulation Command*

Расположите окна **JOURNAL** и **BLOCK ENTITIES** так, чтобы можно было видеть каждое из них, а остальные пока скройте, то есть щелкните по кнопке со знаком подчеркивания (Свернуть) – первой в верхнем правом углу окна.

Для перемещения окна наведите курсор мыши на заголовок окна – верхнее синее поле, – нажмите левую кнопку мыши и, удерживая ее нажатой, переместите окно в нужное место, затем отпустите кнопку. Для уменьшения или увеличения размеров окна подведите курсор к любой его границе. Указатель мыши преобразуется в двустороннюю стрелку; нажав левую кнопку, перемещайте курсор, а вместе с ним и границу в нужном направлении.

Для показа окон:

- щелкните по пункту **Window** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **Tile** (Мозаика) выпадающего меню. Окна разместятся друг под другом так, как показано на рис. 2.25.

Теперь перейдем к моделированию системы:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **START** выпадающего меню. Появится диалоговое окно **Start Command**, в котором укажите число прогонов модели, например 100, и щелкните по кнопке **OK**. В ранее открытые окна добавится информация, и эти окна будут выглядеть так, как изображено на рис. 2.26.

В окне **JOURNAL** вы видите сообщение о том, что требование остановилось на блоке 5. В окне **BLOCK ENTITIES** перед блоком 5 (5 QUE QUEUE ...) остановилось требование (транзакт) в виде маленького прямоугольника.

Теперь мы сделаем один шаг в моделировании с использованием функциональной клавиши **F5**, которая по умолчанию соответствует команде **STEP 1** (см. рис. 2.22). В окна **JOURNAL** и **BLOCK ENTITIES** снова добавится информация.

Поскольку мы назначили функциональные клавиши **F8** и **F9** соответственно командам **SHOW P\$N_zakaz** и **SHOW X\$N_zakaz**, то в любой момент, нажав на эти клавиши, можем в окне **JOURNAL** получить численные значения стандартных числовых атрибутов **P\$N_zakaz** и **X\$N_zakaz**.

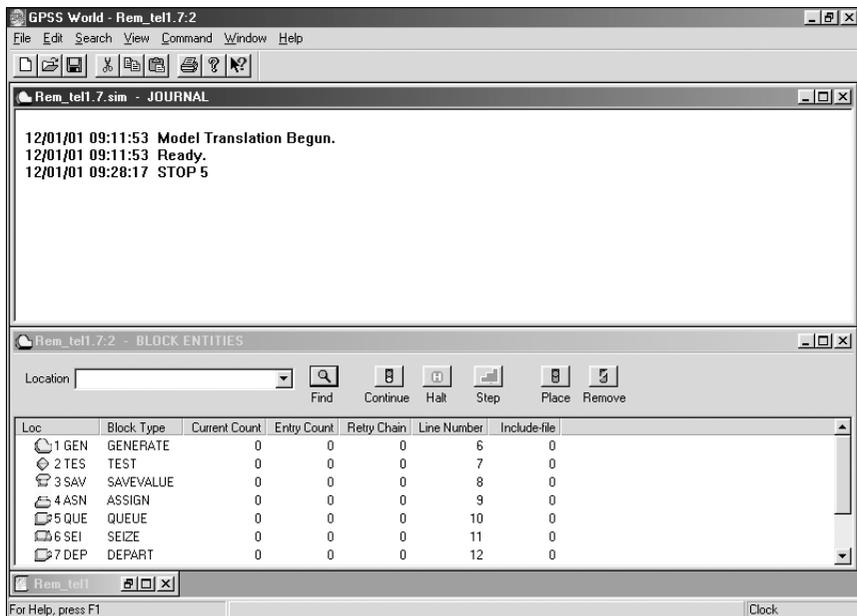


Рис. 2.25. Главное окно системы с двумя окнами – **JOURNAL** и **BLOCK ENTITIES** – до начала моделирования для модели *Rem_tel1*

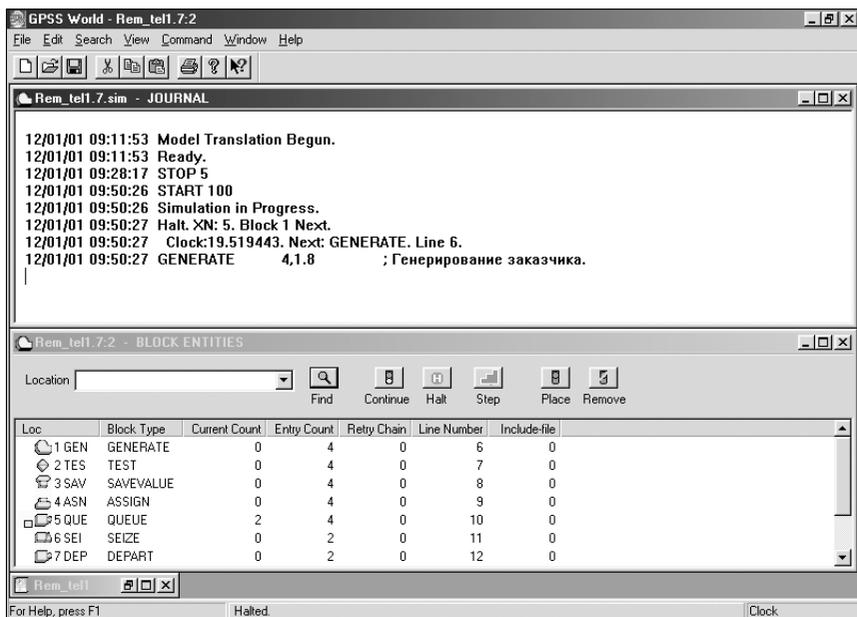


Рис. 2.26. Главное окно системы с двумя окнами – **JOURNAL** и **BLOCK ENTITIES** – после начала моделирования для модели *Rem_tel1*

Теперь, когда мы умеем управлять активным требованием (заказчиком), давайте снимем условие остановки. Для этого:

- щелкните по пункту **Window** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **Simulation Snapshot** (Снимок моделирования) выпадающего меню. Появится диалоговое окно **Stop Conditions** (Условия остановки). Оно будет выглядеть так, как показано на рис. 2.27;

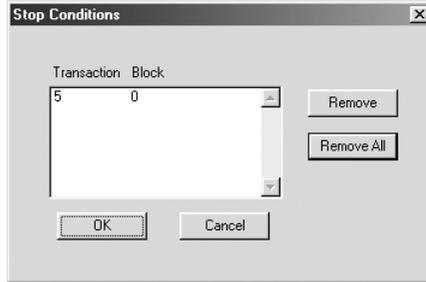


Рис. 2.27. Диалоговое окно **Stop Conditions**

- щелкните по числу 5 для выделения строки;
- щелкните по кнопке **Remove** – выделенная строка исчезнет;
- щелкните по кнопке **OK**. Условие остановки в примере будет снято. Об этом появится следующее сообщение в окне **JOURNAL**:

```
STOP 5, 0, OFF
```

Можно посмотреть некоторые другие показатели работы моделируемой системы, например текущее содержание очереди, которое выводится с помощью соответствующего стандартного числового атрибута Q\$Remont. Для этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **SHOW** (Показать) выпадающего меню. Появится диалоговое окно **Show Command**, в котором укажите искомый показатель – Q\$Remont. Это будет выглядеть так, как представлено на рис. 2.28;



Рис. 2.28. Диалоговое окно **Show Command**

- щелкните по кнопке **OK**. Искомый показатель появится в окне **JOURNAL** и в строке состояния в нижней части главного окна системы GPSSW.

Нажимая на функциональную клавишу **F5**, мы на каждом шаге можем посмотреть, что же происходит во время работы системы.

Можно просмотреть требования и в другом окне – **TABLE WINDOW**. Для этого:

- щелкните по пункту **Window** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Table Window** (Окно гистограммы) во всплывающем меню. Появится диалоговое окно **Open Table Window** (Открыть окно гистограммы). Для нашей задачи оно выглядит так, как показано на рис. 2.29;

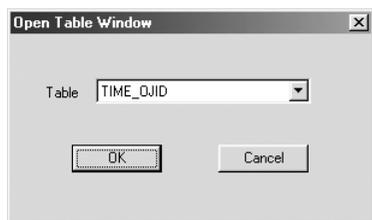


Рис. 2.29. Диалоговое окно **Open Table Window**

- щелкните по кнопке **OK**. Появится окно **TABLE WINDOW** с соответствующей гистограммой – оно показано на рис. 2.30.

На этой гистограмме можно увидеть распределение времени ожидания заказчиков ремонта. Среднее время ожидания составляет 13,070 мин, стандартное отклонение – 2,912 мин.

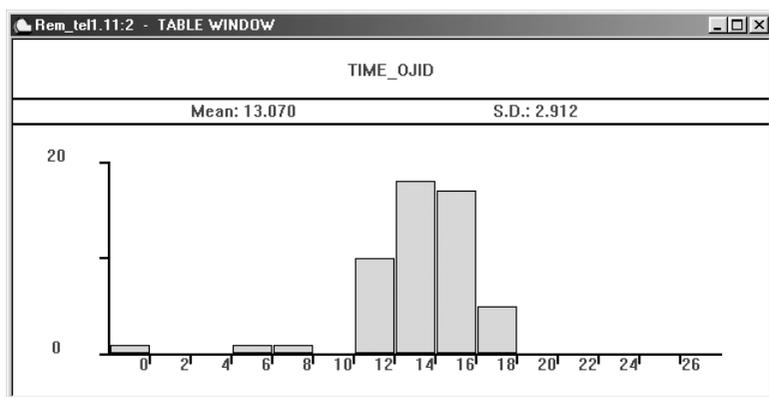


Рис. 2.30. Диалоговое окно **TABLE WINDOW** для модели *Rem_tel1*

Моделирование непроизводственных систем

Моделирование работы магазина	92
Моделирование движения на пешеходном переходе ...	109
Моделирование работы переговорного пункта	121
Моделирование работы супермаркета	129
Моделирование системы «Хищник–добыча»	141
Моделирование распространения эпидемии	148

В этой главе приведены примеры использования системы GPSSW для моделирования различных непроизводственных процессов.

Моделирование работы магазина

Постановка задачи

Допустим, нам надо промоделировать работу небольшого магазина, который имеет один кассовый аппарат и одного продавца. Известны следующие параметры функционирования магазина:

- поток покупателей (требований), приходящих в магазин за покупками, равномерный;
- интервал времени прибытия покупателей колеблется в пределах от 8,7 до 10,3 мин включительно, или $9,5 \pm 0,8$ мин;
- время пребывания покупателей у кассового аппарата составляет $2,3 \pm 0,7$ мин. После этого покупатели подходят к продавцу для получения товара;
- время, потраченное на обслуживание покупателей продавцом, составляет $10 \pm 1,4$ мин.

Требуется определить параметры функционирования магазина:

- коэффициент загрузки кассира;
- коэффициент загрузки продавца;
- максимальное, среднее и текущее число покупателей в каждой очереди;
- среднее время обслуживания в каждом канале обслуживания;
- среднее время нахождения покупателя в каждой очереди и др.

Выявление основных особенностей

Для моделирования работы магазина необходимо сформировать входной поток покупателей (требований) и временной интервал моделирования работы магазина. Но перед этим необходимо выбрать единицу измерения времени. Для моделирования работы магазина можно взять в качестве единицы измерения минуту.

Создание имитационной модели процесса

Создание имитационной модели начнем с построения заголовка модели, который может быть представлен, например, в таком виде:

```
; GPSSW File      MAGAZIN.GPS
*****
*   Моделирование работы магазина   *
*****
```

Моделирование потока покупателей будем выполнять с помощью оператора **GENERATE** (Генерировать). В нашем примере он будет выглядеть так:

```
t_prod QTABLE      Ocher_prod,0,2,32
GENERATE          9.5,0.8
```

В поле операнда А указывается средний интервал времени между прибытием в магазин двух идущих один за другим покупателей (требований, транзактов). В нашем примере он составляет 9,5 мин.

В поле операнда В дано отклонение времени прихода покупателей от среднего. В нашем примере это отклонение составляет 0,8 мин.

Покупатель, пришедший в магазин, сначала встает в очередь к кассиру, если она есть. Это можно промоделировать оператором **QUEUE** (Очередь), который только в совокупности с соответствующим оператором **DEPART** (Выйти) собирает статистическую информацию о работе моделируемой очереди.

В нашем примере оператор **QUEUE** будет выглядеть так:

```
QUEUE Ocher_kassa
```

В поле операнда А дается символьное или числовое имя очереди. Таких очередей в сложных системах может быть очень много. В нашей задаче дадим очереди имя `Ocher_kassa` (Очередь в кассу). Желательно, чтобы присваиваемое имя отражало суть описываемого элемента системы.

Следуя логике, покупатель может выйти из очереди только тогда, когда освободится кассир (канал обслуживания). Для этого вводится оператор **SEIZE**, который определяет занятость канала обслуживания, и при его освобождении очередное требование выходит из очереди и идет в канал на обслуживание. Это может выглядеть так:

```
SEIZE Kassir
```

В поле операнда А дается символьное или числовое имя канала обслуживания. Таких каналов обслуживания в системе может быть очень много. В нашей задаче каналу дано имя `Kassir` (Кассир). Здесь также имя должно отражать суть описываемого элемента системы.

Выход покупателя из очереди в кассу фиксируется оператором **DEPART** с соответствующим названием очереди. В нашем примере это будет выглядеть так:

```
DEPART Ocher_kassa
```

Далее должно быть промоделировано время пребывания покупателя, непосредственно обслуживаемого кассиром. Это время в нашем примере составляет $2,3 \pm 0,7$ мин. Для моделирования этого процесса используется оператор **ADVANCE** (Задержать), который в нашей задаче будет выглядеть так:

```
ADVANCE 2.3,0.7
```

После обслуживания кассиром покупатель отправляется к продавцу за получением оплаченного товара. Однако перед этим системе должно быть послано сообщение об освобождении канала обслуживания. Это делается с помощью оператора **RELEASE**, который в нашей задаче записывается так:

```
RELEASE Kassir
```

Следует особо подчеркнуть, что парные операторы **QUEUE** и **DEPART** для каждой очереди должны иметь одно и то же, но свое уникальное имя. Это же относится и к операторам **SEIZE** и **RELEASE**.

После обслуживания в кассе покупатель направляется к продавцу – следующему каналу обслуживания. Процесс моделирования этой цепи аналогичен только что описанному. И в нашем примере он может быть представлен, например, в таком виде:

```
QUEUE    Ocher_prod
SEIZE    Prodavec
DEPART   Ocher_prod
ADVANCE  10,1.4
RELEASE  Prodavec
```

После обслуживания продавцом (каналом обслуживания) покупатель (требование) покидает систему. Это действие может быть представлено оператором **TERMINATE** (Завершить):

```
TERMINATE 1
```

В поле операнда A стоит число 1. Это означает, что систему обслуживания – магазин – покупатели покидают по одному. Завершающим оператором в нашей задаче является управляющая команда **START** (Начать), позволяющая начать моделирование:

```
START 100
```

В поле операнда A стоит число 100, показывающее, с каким числом покупателей будет моделироваться система работы магазина.

Представление имитационной модели

Для представления имитационной модели выполните следующие действия:

- щелкните по пункту **File** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **New** выпадающего меню. Появится диалоговое окно **Новый документ**;
- выделите пункт **Model** и щелкните по кнопке **ОК**. Появится окно модели, в котором введите данную программу. Это будет выглядеть так, как показано на рис. 3.1.

Во введенной программе в самом начале добавлен оператор **QTABLE** с меткой `t_prod` для сбора информации и построения соответствующей гистограммы функционирования очереди под именем `Ocher_prod`.

Чтобы вызвать окно для представления имитационной модели в системе GPSSW, можно также нажать комбинацию клавиш **Ctrl+Alt+S**.

Подготовка к моделированию системы

Перед началом моделирования можно установить вывод тех параметров моделирования, которые нужны пользователю. Для этого:

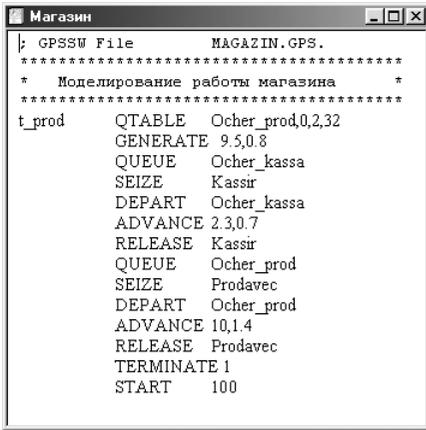


Рис. 3.1. Окно имитационной модели «Магазин»

- щелкните по пункту **Edit** (Правка) главного меню системы или нажмите комбинацию клавиш **Alt+E**. Появится выпадающее меню;
- щелкните по пункту **Settings** (Установки) выпадающего меню. Появится диалоговое окно **SETTINGS** для данной модели, в котором можно установить нужные выходные данные, которые отмечаются флажком (галочкой). Для нашего примера это может выглядеть так, как представлено на рис. 3.2.

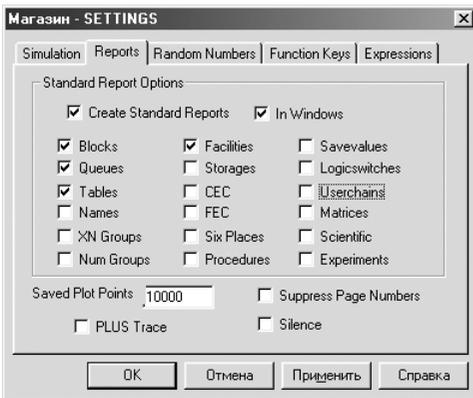


Рис. 3.2. Окно **SETTINGS** с установками для имитационной модели магазина

Наличие галочки в окошках говорит о том, что эта информация будет выведена в окне результатов моделирования. В нашем примере будет выведена информация для следующих объектов:

- **Blocks** (Блоки);
- **Queues** (Очереди);
- **Tables** (Таблицы/гистограммы);
- **Facilities** (Каналы обслуживания).

Моделирование системы

После создания имитационную (выполняемую) модель необходимо оттранслировать и запустить на выполнение. Для этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню.

Если управляющая команда **START** есть в модели, то исходная имитационная модель после трансляции, если в ней нет ошибок, начнет выполняться. Будет выполняться то число прогонов, которое указано в поле операнда A команды **START**. Затем появится окно **JOURNAL**.

Если управляющей команды **START** в модели нет, то исходная имитационная модель будет транслироваться, и если в ней нет ошибок, то будет получена оттранслированная модель – в машинных кодах, готовая к выполнению моделирования, так называемая выполняемая модель.

Перед началом моделирования, а точнее после появления окна **JOURNAL**, можно настроить графики вывода некоторых параметров функционирования системы. Для этого:

- щелкните по пункту **Window** главного меню системы или нажмите комбинацию клавиш **Alt+W**. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Plot Window** (Окно графика) всплывающего меню. Появится диалоговое окно **Edit Plot Window** (Окно редактирования графика), которое необходимо соответствующим образом заполнить.

Графическое представление результатов моделирования

Допустим, мы хотим на всем периоде моделирования видеть график того, как меняется длина очереди к продавцу. Для нашей задачи окно **Edit Plot Window** может быть заполнено так, как показано на рис. 3.3.

После заполнения диалогового окна **Edit Plot Window** щелкните по кнопкам **Plot** (График), **Memorize** (Запомнить), а затем – по кнопке **OK**. Появится заготовка графика (рис. 3.4).

После этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **START**. Появится диалоговое окно **Start Command**;
- введите в диалоговом окне **Start Command** число посетителей магазина, например 1000, и щелкните по кнопке **OK**. Появится окно **REPORT** с результатами моделирования. На заднем плане будет размещаться график;

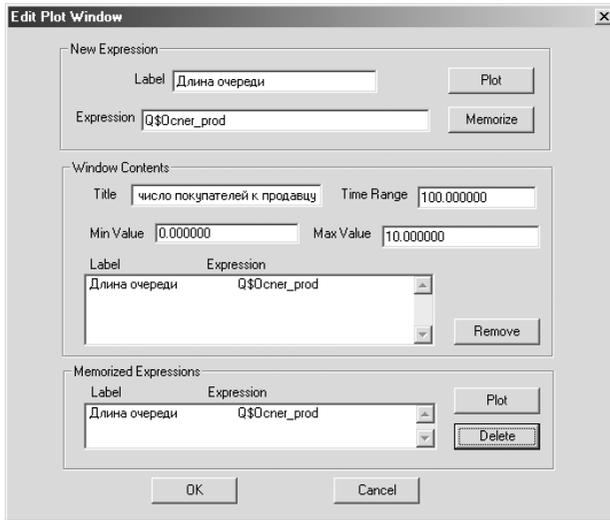


Рис. 3.3. Диалоговое окно **Edit Plot Window** для имитационной модели магазина

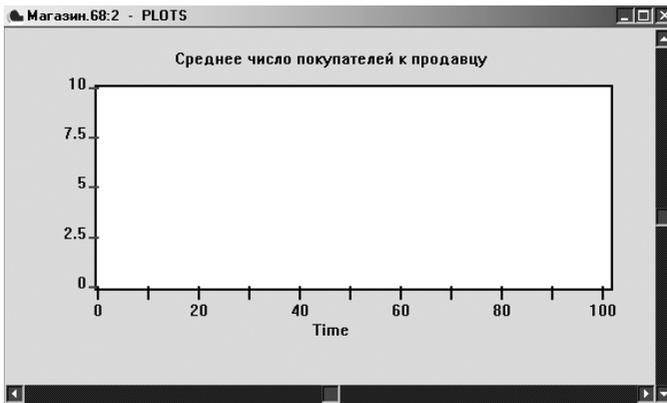


Рис. 3.4. Заготовка графического представления длины очереди к продавцу для имитационной модели магазина

- щелкните по графику, расположенному на заднем плане, – он выйдет на первый план;
- используя горизонтальную и вертикальную полосы прокрутки, вы можете просмотреть построенный график.

Фрагмент графика для нашего примера представлен на рис. 3.5.

При выводе графика на передний экран **REPORT** с результатами моделирования переместится на задний план. Для просмотра окна **REPORT** щелкните по нему мышью. Оно вновь перейдет на передний план и будет выглядеть так, как показано на рис. 3.6.

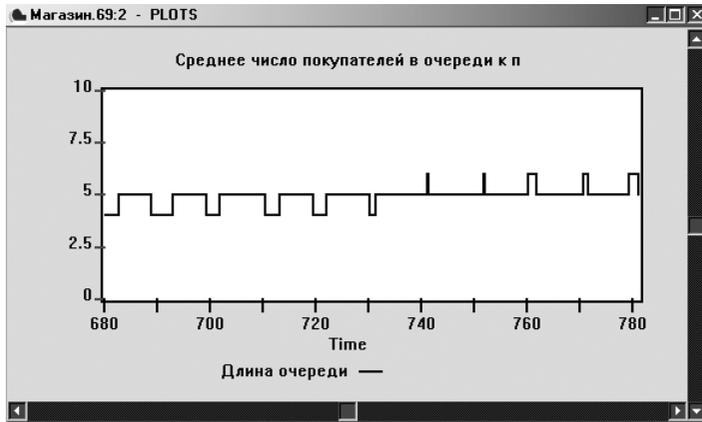


Рис. 3.5. Фрагмент изображения длины очереди к продавцу для имитационной модели магазина

GPSS World - [Магазин.57.1 - REPORT]

File Edit Search View Command Window Help

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	1016.324	12	2	0

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT	COUNT	RETRY
	1	GENERATE	107	0	0	0
	2	QUEUE	107	0	0	0
	3	SEIZE	107	0	0	0
	4	DEPART	107	0	0	0
	5	ADVANCE	107	0	0	0
	6	RELEASE	107	0	0	0
	7	QUEUE	107	6	0	0
	8	SEIZE	101	1	0	0
	9	DEPART	100	0	0	0
	10	ADVANCE	100	0	0	0
	11	RELEASE	100	0	0	0
	12	TERMINATE	100	0	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
KASSIR	107	0.244	2.318	1	0	0	0	0	0
PRODAVEC	101	0.987	9.929	1	101	0	0	0	6

QUEUE	MAX CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY
OCHER_PROD	7	7	107	2	3.607	34.260	34.913
OCHER_KASSA	1	0	107	107	0.000	0.000	0.000

TABLE	MEAN	STD.DEV.	RANGE	RETRY FREQUENCY	CUM.%
T_PROD	34.338	17.466		0	

For Help, press F1 Report is Complete. Clock

Рис. 3.6. Окно **REPORT** с фрагментом результатов моделирования для имитационной модели магазина

В верхней строке окна **REPORT** (Отчет) указываются:

- **START TIME** (Начальное время) – 0.000;
- **END TIME** (Время окончания) – 1016.324;
- **BLOCKS** (Число блоков) – 12;
- **FACILITIES** (Число каналов обслуживания) – 2;
- **STORAGES** (Число накопителей) – 0.

Ниже перечисляются блоки модели и количество входов в них требований (покупателей). При этом каждый блок имеет свой числовой номер.

Еще ниже указываются результаты моделирования каналов обслуживания под назначенными нами именами **KASSIR** и **PRODAVEС** соответственно:

- **ENTRIES** (Число входов) – 107, 101;
- **UTIL.** (Коэффициент использования) – 0.244, 0.987;
- **AVE. TIME** (Среднее время обслуживания) – 2.318, 9.929;
- **AVAIL.** (Доступность) – 1, 1;
- **OWNER** (Возможное число входов) – 0, 101;
- **PEND** – 0, 0;
- **INTER** – 0, 0;
- **RETRY** (Повтор) – 0, 0;
- **DELAY** (Отказано) – 0, 6.

Еще ниже указываются результаты моделирования каждой очереди под присвоенными нами именами **OSHER_PROD** и **OSHER_KASSA** соответственно:

- **MAX** (Максимальное содержание) – 7 и 1;
- **CONT.** (Текущее содержание) – 7 и 0;
- **ENTRY** (Число входов) – 107 и 107;
- **ENTRY(0)** (Число нулевых входов) – 2 и 107;
- **AVE.CONT.** (Среднее число входов) – 3.607 и 0.000;
- **AVE.TIME** (Среднее время) – 34.260 и 0.000;
- **AVE.(–0)** – 34.913 и 0.000;
- **RETRY** – 0 и 0.

Еще ниже указываются результаты моделирования для построения по табличным данным гистограммы **T_PROD** функционирования очереди под именем **OSHER_PROD**:

- **MEAN** (Средняя) – 34.338;
- **STD.DEV.** (Среднее квадратическое отклонение) – 17.466;
- **RANGE** (Область);
- **RETRY** – 0;
- **FREQUENCY** (Частота);
- **CUM.%** (Суммарный процент).

При наличии оператора

```
t_prod QTABLE Ocher_prod, 0, 2, 32
```

можно вывести соответствующую гистограмму. Для этого:

- щелкните по пункту **Window** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** (Окно моделирования) выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Table Window** (Окно гистограммы) во всплывающем меню. Появится диалоговое окно **Open Table Window** (Открыть окно гистограммы). В раскрывающемся списке **Table** щелкните по нужной гистограмме. Для нашей задачи окно будет выглядеть так, как показано на рис. 3.7;
- щелкните по кнопке **OK**. Появится соответствующая гистограмма. Для нашей задачи она выглядит так, как показано на рис. 3.8.

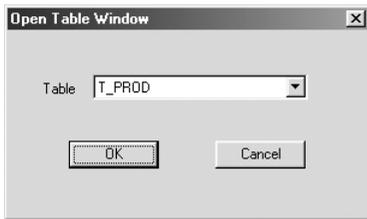


Рис. 3.7. Диалоговое окно **Open Table Window** для выбора нужной гистограммы

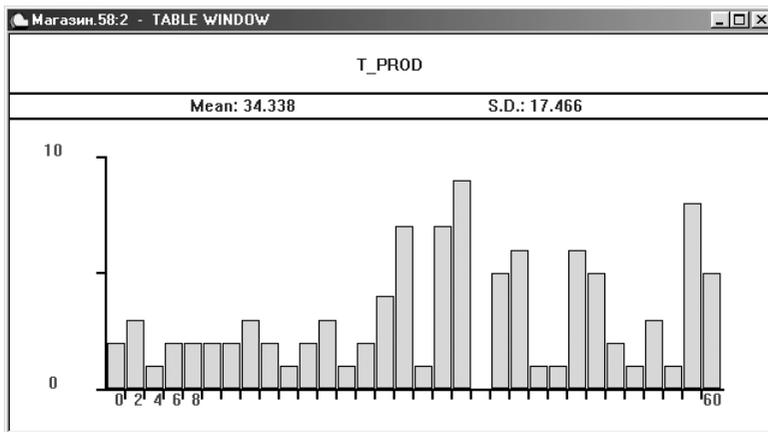


Рис. 3.8. Окно гистограммы очереди к продавцу в модели «Магазин»

Визуализация процесса функционирования системы

После трансляции модели система GPSSW обеспечивает возможность визуального наблюдения перемещения покупателей (активных требований) в процессе

моделирования. Если в модели есть команда управления **START**, она должна быть заблокирована, то есть переведена в комментарии. Для этого в позиции 1 поставьте звездочку. Для включения режима визуального наблюдения перемещения покупателей в процессе моделирования выполните следующие действия:

- щелкните по пункту **Window** главного меню системы или нажмите комбинацию клавиш **Alt+W**. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Block Entities** (Блочные элементы) всплывающего меню. Появится диалоговое окно **BLOCK ENTITIES** (рис. 3.9).

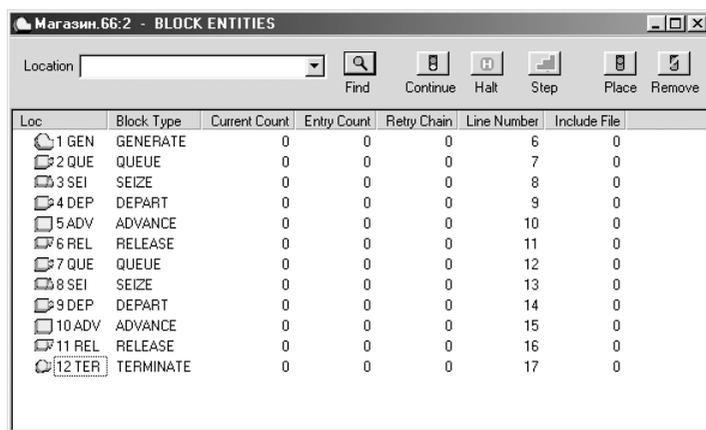


Рис. 3.9. Блок-схема модели «Магазин»

Для визуализации перемещения активных требований (транзактов) в процессе моделирования:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **START**. Появится диалоговое окно **Start Command** (рис. 3.10);
- введите число покупателей, которые собираются посетить магазин, например 100;
- щелкните по кнопке **OK**. Начнется процесс поступления и перемещения активных требований (покупателей) в магазине. Каждое перемещение требования по блокам системы фиксируется в правой части окна **BLOCK ENTITIES**;
- щелкните по кнопке **Halt** (Остановить), расположенной на панели кнопок управления в верхней правой части окна **BLOCK ENTITIES**. Одно из состояний моделирования системы магазина представлено на рис. 3.11;

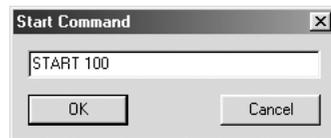
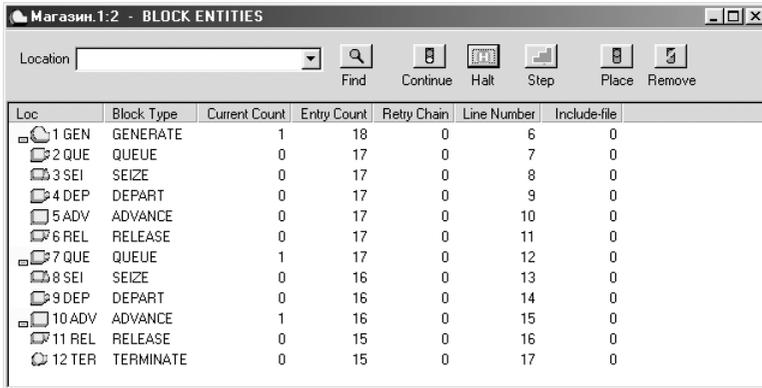


Рис. 3.10. Диалоговое окно **Start Command**



Loc	Block Type	Current Count	Entry Count	Retry Chain	Line Number	Include-file
1 GEN	GENERATE	1	18	0	6	0
2 QUE	QUEUE	0	17	0	7	0
3 SEI	SEIZE	0	17	0	8	0
4 DEP	DEPART	0	17	0	9	0
5 ADV	ADVANCE	0	17	0	10	0
6 REL	RELEASE	0	17	0	11	0
7 QUE	QUEUE	1	17	0	12	0
8 SEI	SEIZE	0	16	0	13	0
9 DEP	DEPART	0	16	0	14	0
10 ADV	ADVANCE	1	16	0	15	0
11 REL	RELEASE	0	15	0	16	0
12 TER	TERMINATE	0	15	0	17	0

Рис. 3.11. Одно из состояний моделирования системы магазина в детальном представлении

- щелкните по кнопке **Continue** (Продолжить) для продолжения моделирования или по кнопке **Step** (Шагнуть), чтобы промоделировать и просмотреть изменения в системе в течение одного шага. По кнопке **Step** можно щелкать многократно для подробного просмотра последовательных шагов моделирования системы.

Можно визуально наблюдать перемещения активных требований (транзактов) в процессе моделирования в том же окне **BLOCK ENTITIES**, но только без излишних деталей. Для этого:

- щелкните по пункту **View** (Вид) главного меню системы или нажмите комбинацию клавиш **Alt+V**. Появится выпадающее меню;
- щелкните по пункту **Entity Details** (Детальное представление) выпадающего меню. По умолчанию перед пунктом **Entity Details** стоит галочка, что означает установленный режим детального представления моделируемой системы. После щелчка по этому пункту мышью лишняя информация исчезнет, и окно **BLOCK ENTITIES** будет для нашей задачи выглядеть так, как показано на рис. 3.12.

Каждое требование (покупатель) связано с набором параметров, которые характеризуют его в любой момент времени моделирования системы. Эти параметры имеют либо числовое, либо символьное обозначение. Их можно просмотреть в любом из возможных состояний. Для этого:

- щелкните по пункту **Window** главного меню системы или нажмите комбинацию клавиш **Alt+W**. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Block Entities** всплывающего меню. Появится диалоговое окно **BLOCK ENTITIES** (рис. 3.13).

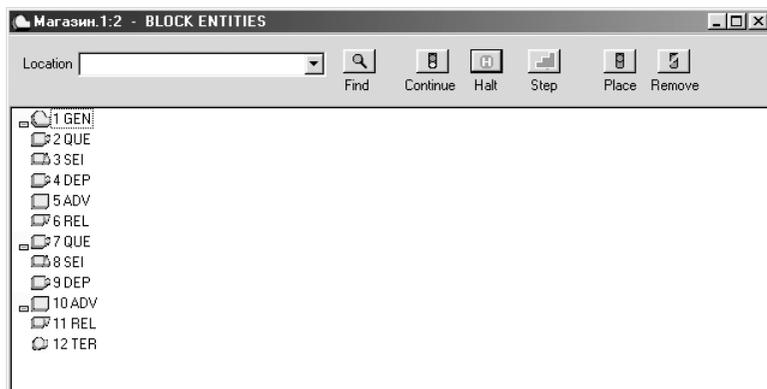


Рис. 3.12. Одно из состояний моделирования системы магазина в упрощенном представлении

The screenshot shows the same window as Figure 3.12, but with a table view of the block entities. The table has the following columns: Loc, Block Type, Current Count, Entry Count, Retry Chain, Line Number, and Include-file.

Loc	Block Type	Current Count	Entry Count	Retry Chain	Line Number	Include-file
1 GEN	GENERATE	0	0	0	6	0
2 QUE	QUEUE	0	0	0	7	0
3 SEI	SEIZE	0	0	0	8	0
4 DEP	DEPART	0	0	0	9	0
5 ADV	ADVANCE	0	0	0	10	0
6 REL	RELEASE	0	0	0	11	0
7 QUE	QUEUE	0	0	0	12	0
8 SEI	SEIZE	0	0	0	13	0
9 DEP	DEPART	0	0	0	14	0
10 ADV	ADVANCE	0	0	0	15	0
11 REL	RELEASE	0	0	0	16	0
12 TER	TERMINATE	0	0	0	17	0

Рис. 3.13. Блок-схема моделируемой системы магазина

Следует помнить, что включение окна просмотра **BLOCK ENTITIES** значительно замедляет процесс моделирования, так как тратится время на визуализацию процесса моделирования. Для ускорения процесса моделирования и быстрого получения конечного результата целесообразно закрыть окно просмотра **BLOCK ENTITIES**. Это можно сделать несколькими способами:

- щелкнуть по кнопке с крестиком, расположенной в правом верхнем углу окна;
- щелкнуть по пиктограмме блока, расположенной в левом верхнем углу окна. Появится всплывающее меню, в котором щелкните по пункту **Close**;
- нажать комбинацию клавиш **Ctrl+F4**;
- дважды щелкнуть по пиктограмме блока, расположенной в левом верхнем углу окна.

Просмотр отдельных результатов моделирования

Можно просматривать искомые результаты моделирования без использования окна **REPORT**. Для этого предварительно в управляющем операторе **START** в поле операнда В после запятой ставятся символы NP (Not Print – Не печатать), например, так:

```
START 100,NP
```

Затем:

- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню. Появится окно **JOURNAL**, показанное на рис. 3.14.

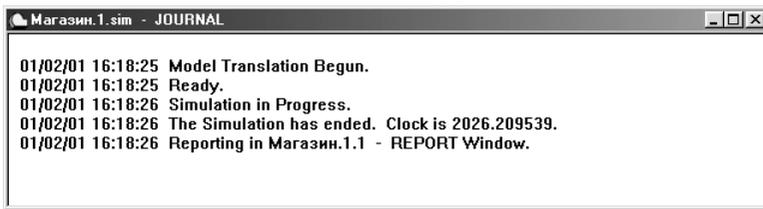


Рис. 3.14. Окно **JOURNAL** для модели «Магазин»

Можно в этом же окне посмотреть некоторые результаты моделирования с использованием команды **SHOW**. Для этого выполните следующие действия:

- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **SHOW** (Показать) выпадающего меню. Появится диалоговое окно **Show Command** (рис. 3.15);



Рис. 3.15. Диалоговое окно **Show Command**

- введите в строке диалогового окна **Show Command** стандартный числовой атрибут C1 (Время моделирования системы) и щелкните по кнопке **OK**. В окне **JOURNAL** появится результат работы команды **SHOW** – значение введенного стандартного числового атрибута (рис. 3.16).

Эта команда обеспечивает вывод относительного времени моделирования системы как в строке состояния, так и в окне **JOURNAL**.

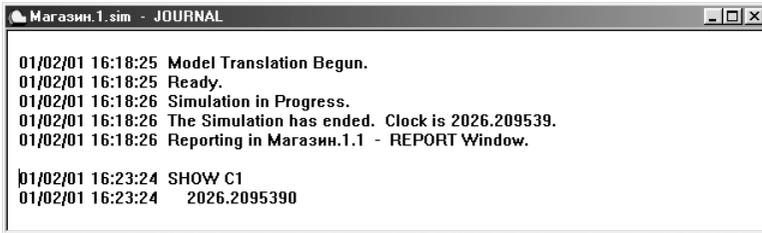


Рис. 3.16. Просмотр в окне **JOURNAL** стандартного числового атрибута *C1*

Можно выводить и другие значения стандартных числовых атрибутов. Например, для нашей задачи мы хотели бы знать максимальную длину очереди. Для этого:

- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **SHOW** выпадающего меню. Появится диалоговое окно **Show Command**;
- введите в строке диалогового окна **Show Command**, например, `QM$Ocher_prod` и щелкните по кнопке **OK**. Будет показано максимальное содержимое указанной очереди – ее максимальная длина. Окно **JOURNAL** будет выглядеть примерно так, как показано на рис. 3.17.

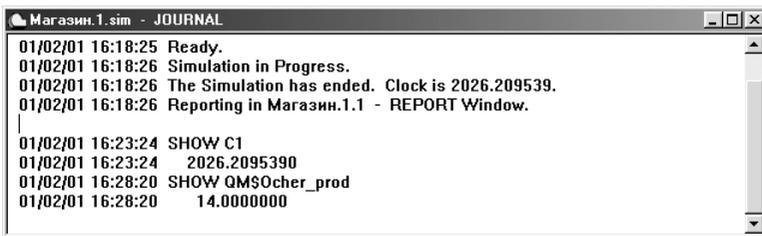


Рис. 3.17. Просмотр в окне **JOURNAL** максимальной длины очереди

Система GPSS обеспечивает возможность просмотра в динамике изменения одного или нескольких числовых значений тех или иных элементов системы в процессе моделирования. Для этого следует использовать окно **Edit Expression Window** (Окно редактирования выражения), которое может содержать список нужных выражений, или **FACILITY ENTITIES** (Элементы канала обслуживания).

Рассмотрим сначала изменения значений выражений в окне **Edit Expression Window**. Наиболее простыми выражениями являются переменные или стандартные числовые атрибуты.

Оттранслируйте созданную модель и сделайте один шаг в моделировании. Допустим, вам надо промоделировать работу магазина при посещении его 100 покупателями. Но вы сначала промоделируете работу магазина с одним покупателем,

то есть управляющая команда **START** в поле операнда В будет иметь значение 1. После этого:

- щелкните по пункту **Window** главного меню или нажмите комбинацию клавиш **Alt+W**. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** выпадающего меню или нажмите на клавишу **W**. Появится всплывающее меню;
- щелкните по пункту **Expressions Window** (Окно выражений) во всплывающем меню или нажмите на клавишу **E**. Появится диалоговое окно **Edit Expression Window**;
- введите в текстовом поле **Label** (Метка) диалогового окна **Edit Expression Window** исходное название просматриваемого элемента системы, например:

Коэффициент использования кассира

- введите в следующем текстовом поле **Expression** (Выражение) имя рассматриваемого выражения – переменной, стандартного числового атрибута и т.д. Например, стандартный числовой атрибут `FR$kassir` – это коэффициент использования канала обслуживания по имени `kassir`;
- щелкните по кнопкам **Memorize** (Запомнить) и/или **View** (Просмотреть). Кнопка **View** позволит вам видеть выражение для этого прогона, в то время как кнопка **Memorize** обеспечивает его запоминание для использования в будущем. После того как информация для какого-либо одного элемента системы введена в оба текстовых поля и выполнены щелчки по кнопкам **View** и **Memorize**, можно снова вводить аналогичную информацию в те же текстовые поля.

На рис. 3.18 показан возможный вариант ввода исходных выражений в диалоговом окне **Edit Expression Window**.

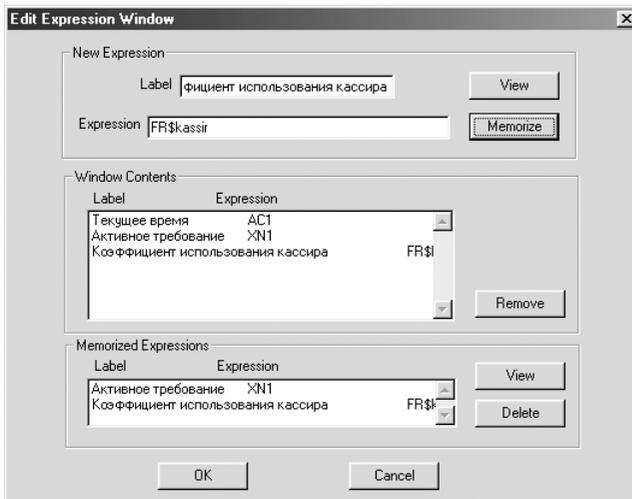


Рис. 3.18. Диалоговое окно **Edit Expression Window** для имитационной модели магазина

После ввода очередных искомых параметров в диалоговом окне **Edit Expression Window** щелкните по кнопкам **View** и **Memorize**, а затем – по кнопке **OK**. Появится окно **EXPRESSIONS** (рис. 3.19).

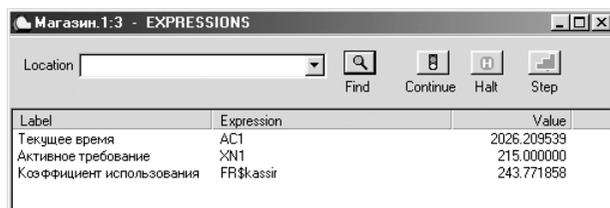


Рис. 3.19. Окно **EXPRESSIONS** для имитационной модели магазина

После этого:

- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **START** выпадающего меню. Появится диалоговое окно **Start Command** (см. рис. 3.10);
- введите в строке диалогового окна **Start Command** после команды **START** значение 100 и щелкните по кнопке **OK**.

Теперь вы можете наблюдать динамику изменения значений, введенных в окне **EXPRESSIONS**. Можно в любой момент остановить процесс моделирования, щелкнув по кнопке **Halt**, расположенной в правой верхней части окна. Одно из таких промежуточных состояний окна **EXPRESSIONS** показано на рис. 3.20.

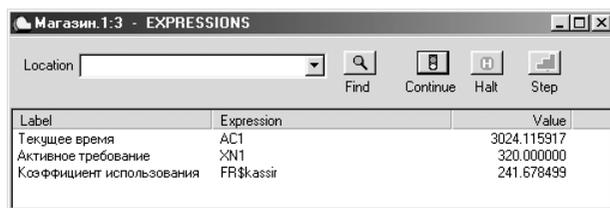


Рис. 3.20. Одно из состояний окна **EXPRESSIONS** для имитационной модели магазина

Следует заметить, что для вычисления истинного значения коэффициента использования канала обслуживания необходимо полученные результаты разделить на 1000. Стандартный числовой атрибут FR\$<символьное имя> выдает значение от 0 до 999.

После прерывания моделирование может быть продолжено, если щелкнуть по кнопке **Continue**. Моделирование можно выполнять и по одному шагу, если использовать кнопку **Step**.

Рассмотрим теперь изменения значений показателей (параметров) функционирования каналов обслуживания. Это мы будем делать с помощью окна **FACILITY**

ENTITIES (Показатели канала обслуживания). Сначала, как и ранее, оттранслируйте созданную модель и сделайте один шаг в моделировании. Допустим, вам надо промоделировать работу магазина при посещении его 100 покупателями. Но вы сначала промоделируете работу магазина с одним покупателем, то есть управляющая команда **START** в поле операнда **B** будет иметь значение 1. После этого:

- щелкните по пункту **Window** главного меню или нажмите комбинацию клавиш **Alt+W**. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** (Окно моделирования) выпадающего меню или нажмите на клавишу **W**. Появится всплывающее меню;
- щелкните по пункту **Facilities Window** (Окно каналов обслуживания) во всплывающем меню или нажмите на клавишу **F**. Появится окно показателей функционирования каналов обслуживания **FACILITY ENTITIES**.

На рис. 3.21 показан возможный вариант значений показателей работы каналов обслуживания в нашей задаче в окне **FACILITY ENTITIES**.

Facility	Utilization	Delay Chain	Acquisitions	Available	Ave. Time	Owner XN	Retry Chain	Pending Chain	Interrupt Chain
KASSIR	0.244	0	107	+	2.318	0	0	0	0
PRODAVEC	0.987	6	101	+	9.929	101	0	0	0

Рис. 3.21. Окно **FACILITY ENTITIES** для имитационной модели «Магазин»

После этого:

- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **START** выпадающего меню. Появится диалоговое окно **Start Command**;
- введите в строке диалогового окна **Start Command** после команды **START** значение 150 и щелкните по кнопке **OK**.

Теперь вы можете наблюдать динамику изменения значений в окне **FACILITY ENTITIES**. Можно в любой момент времени остановить процесс моделирования, щелкнув по кнопке **Halt** (Остановить), расположенной в правой верхней части окна.

Следует заметить, что в окне **FACILITY ENTITIES** значения коэффициента использования каналов обслуживания даются в реальном представлении. После прерывания моделирование может быть продолжено, если щелкнуть по кнопке **Continue** (Продолжить). Моделирование можно выполнять и по одному шагу, если использовать кнопку **Step** (Шаг).

Моделирование движения на пешеходном переходе

Постановка задачи

Нужно промоделировать движение на пешеходном переходе, оборудованном светофором. Допустим, что в одном направлении автомобили подъезжают к переходу с равномерным распределением с интервалом времени 20 ± 10 с, а пешеходы, желающие пересечь улицу по переходу, прибывают к нему с интервалом 30 ± 10 с также с равномерным распределением. Если переход занят пешеходами, то дорожное движение останавливается, и создается очередь из ожидающих автомобилей. Если сигнал светофора зеленый и пешеходный переход не занят пешеходами, то автомобили проезжают. Если сигнал светофора красный или пешеходный переход занят пешеходами, то автомобили не могут проезжать. Время проезда через пешеходный переход составляет 10 ± 2 с и распределяется в замкнутом интервале [8–12] с согласно равномерно распределенному закону. Требуется:

- определить основные параметры функционирования пешеходного перехода, накопив статистику после проезда через него в одном направлении 1000 автомобилей;
- определить среднюю загрузку перехода.

Выявление основных особенностей

Для моделирования заданного процесса необходимо сформировать входные потоки требований и временной интервал моделирования всего процесса. Но перед этим нужно выбрать единицу измерения времени. Для данного процесса можно взять в качестве единицы измерения времени секунду. Для моделирования заданного процесса необходимо сформировать два входных потока: поток автомобилей, подъезжающих к переходу, и поток пешеходов, использующих переход. В качестве ограничителя периода имитации можно использовать число автомобилей, подъезжающих к переходу с одной стороны, например 1000.

Для решения нашей задачи потребуется создать три сегмента модели для моделирования:

- поток автомобилей;
- поток пешеходов;
- режим работы светофора.

Построение имитационной модели

Построение имитационной модели следует начинать с создания заголовка модели, который может быть представлен, например, в таком виде:

```
; GPSSW                Peshehod.GPS
*****
* Моделирование пешеходного перехода *
*****
```

Сначала сформируем поток машин, пересекающих переход, – первый сегмент программы:

```
*****
*      Моделирование потока машин      *
*****
```

Это можно сделать с помощью оператора **GENERATE**, который в нашей задаче запишем в таком виде:

```
GENERATE 20,10
```

В поле операнда А указывается среднее время поступления (обязательно) требования (машины) на обслуживание (проезд через переход), в поле операнда В – отклонение от среднего времени поступления требования (не обязательно). Далее введем пару операторов: сначала **QUEUE**, а затем через несколько операторов и **DEPART** с одним и тем же именем (числовым или символьным) в поле операнда А для сбора статистической информации по очереди машин перед переходом. Эта очередь образуется во время занятости пешеходного перехода или из-за красного сигнала на светофоре. Условно назовем эту очередь именем *Ocher1*. Сначала зафиксируем вход в очередь автомобиля с помощью оператора **QUEUE**. Это будет выглядеть так:

```
QUEUE Ocher1
```

Когда сигнал светофора зеленый (значение 0) для автомобилей и пешеходный переход не занят, тогда значение стандартного числового атрибута, оценивающего возможность проезда автомобиля через пешеходный переход, принимается равным 0. Это означает, что автомобиль может пересекать пешеходный переход. Эти условия должны быть протестированы блочным оператором **TEST**. Когда сигнал светофора красный (значение 100) или переход не свободен от пешеходов, тогда значению стандартного числового атрибута, оценивающего возможность проезда, возвращается значение, равное 1. Это можно записать так:

```
TEST E X$Svet_Avtom,F$Perehod
```

В операторе **TEST** использован логический оператор E (Equal – Равно). Как можно заметить, поле операнда С пусто, следовательно, требование не может войти в блок **TEST**, пока заданное условие числовых атрибутов, приведенных в полях операндов А и В, не выполнится.

В поле операнда А находится сохраняемая величина по имени *Svet_Avtom*, а в поле В – стандартный числовой атрибут, характеризующий содержимое канала

обслуживания – наличие пешеходов на переходе. Когда сигнал светофора зеленый – сохраняемая величина по имени `Svet_Avtom` равна 0 – и пешеходный переход не занят – стандартный числовой атрибут по имени `FSPerehod` равен 0, – тогда машина (требование) может пересекать переход.

Далее введем пару операторов: сначала **SEIZE**, а затем через несколько операторов и **RELEASE** с одним и тем же именем (числовым или символьным) в поле операнда A для сбора статистической информации по каналу обслуживания – переходу:

```
SEIZE Perehod
```

Если канал обслуживания освобожден, то требование может покинуть очередь, используя оператор **DEPART**:

```
DEPART Ocher1
```

Далее автомобиль пересекает пешеходный переход, затрачивая на это время, равное 10 ± 2 с. Это действие моделируется оператором **ADVANCE**:

```
ADVANCE 10,2
```

В поле операнда A указывается среднее время пересечения перехода автомобилем (обязательно), в поле операнда B – отклонение от среднего времени пересечения перехода автомобилем (не обязательно). Далее вводится оператор **RELEASE**, сообщающий об освобождении канала обслуживания:

```
RELEASE Perehod
```

Далее используется оператор **TERMINATE** для завершения моделирования в первом сегменте программы:

```
TERMINATE 1
```

После этого создаем другой сегмент программы, в котором формируется второй поток – поток пешеходов:

```
*****
* Моделирование потока пешеходов *
*****
```

Он очень похож на первый сегмент программы. Оператор **GENERATE** генерирует поток пешеходов с временем подхода пешеходов к переходу, равным 30 ± 10 с:

```
GENERATE 30,10
```

Далее введем пару операторов: сначала **QUEUE**, а затем через несколько операторов и **DEPART** с одним и тем же именем (числовым или символьным) в поле операнда A для сбора статистической информации по очереди пешеходов. Эта очередь создается во время занятости пешеходного перехода машинами или из-за красного сигнала на светофоре. Условно назовем эту очередь `Ocher2`. Сначала

зафиксируем вход в очередь автомобиля с помощью оператора **QUEUE**. Это будет выглядеть так:

```
QUEUE Ocher2
```

Когда сигнал светофора зеленый (значение 0) для пешехода, тогда значение стандартного числового атрибута, оценивающего возможность прохода пешеходом перехода, принимается равным 0. Это означает, что пешеход может пересекать улицу. Эти условия должны быть протестированы блочным оператором **TEST**. Когда сигнал светофора красный (значение 100) или переход занят автомобилем, тогда стандартному числовому атрибуту, оценивающему возможность перехода, возвращается значение, равное 1. Это можно записать так:

```
TEST E X$Svet_Pesheh, F$Perehod
```

В поле операнда А находится сохраняемая величина по имени Svet_Pesheh, а в поле В – стандартный числовой атрибут, характеризующий содержимое канала обслуживания – наличие автомобилей на пешеходном переходе. Когда сигнал светофора зеленый – сохраняемая величина по имени Svet_Pesheh равна 0 – и пешеходный переход не занят – стандартный числовой атрибут по имени F\$Perehod равен 0, – тогда пешеход (требование) может пойти по переходу.

Далее введем пару операторов: сначала **SEIZE**, а затем через несколько операторов и **RELEASE** с одним и тем же именем (числовым или символьным) в поле операнда А – Perehod – для сбора статистической информации по каналу обслуживания:

```
SEIZE Perehod
```

Если канал обслуживания – переход – разрешен и свободен от автомобилей, то требование может покинуть очередь, используя оператор **DEPART**:

```
DEPART Ocher2
```

Далее пешеход пересекает переход, затрачивая на это время, равное 10 ± 2 с. Это действие моделируется оператором **ADVANCE**:

```
ADVANCE 10,2
```

В поле операнда А указывается среднее время прохода перехода пешеходом (обязательно), в поле операнда В – отклонение от среднего времени прохода пешеходом (не обязательно). Далее вводится оператор **RELEASE**, сообщающий об освобождении канала обслуживания (перехода):

```
RELEASE Perehod
```

Затем используется оператор **TERMINATE** для завершения моделирования во втором сегменте программы:

```
TERMINATE 1
```

Переходим к созданию третьего сектора программы – моделирования работы светофора:

```
*****
*   Моделирование работы светофора   *
*****
GENERATE ,,,1
```

Оператор **GENERATE** генерирует одно требование (транзакт), обеспечивающее работу светофора. Для этого в поле операнда D вводится значение 1.

Далее используется сохраняемая величина под именем Svet_Pesheh для указания цвета сигнала светофора со стороны пешеходов – Krasn (Красный):

```
Begin1 SAVEVALUE Svet_Pesheh,Krasn
```

Затем используется сохраняемая величина под именем Svet_Avtom для указания цвета сигнала светофора со стороны автомобилей – Zelen (Зеленый):

```
SAVEVALUE Svet_Avtom,Zelen
```

Далее моделируется время включенного состояния зеленого сигнала с использованием оператора **ADVANCE**. В поле A оператора **ADVANCE** вводится имя переменной Zelen_time, в которой хранится информация о продолжительности зеленого сигнала светофора:

```
ADVANCE Zelen_time
```

Затем используется сохраняемая величина под именем Svet_Pesheh для указания в ней цвета сигнала светофора со стороны пешеходов – Zelen:

```
SAVEVALUE Svet_Pesheh,Zelen
```

Потом используется сохраняемая величина под именем Svet_Avtom для указания в ней цвета сигнала светофора со стороны автомобилей – Krasn:

```
SAVEVALUE Svet_Avtom,Krasn
```

Далее моделируется время включенного состояния зеленого сигнала с использованием оператора **ADVANCE**. В поле A оператора **ADVANCE** вводится имя переменной Krasn_time, в которой хранится информация о продолжительности зеленого сигнала светофора:

```
ADVANCE Krasn_time
```

Затем все повторяется сначала. Для этого используется оператор **TRANSFER** в режиме безусловного перехода к оператору с символической меткой Begin1:

```
TRANSFER ,Begin1
```

И наконец, используемым ранее переменным присваиваются конкретные числовые значения. Это можно записать так:

```
Zelen_time EQU 200
Zelen      EQU 0
Krasn     EQU 100
Krasn_time EQU 300
```

В окончательном виде модель, имитирующая движение на пешеходном переходе, будет выглядеть так, как показано на рис. 3.22 и 3.23.

```

Peshehod_1
; GPSSW      Peshehod.GPS
*****
* Моделирование пешеходного перехода *
*****
* Моделирование потока машин *
*****
GENERATE 20,10 ; формирование потока машин у перехода.
QUEUE   Ocher1 ; Вход в очередь машин перед переходом.
TEST E  X$Svet_Avtom,F$Perehod ; Запрет движения машин до появления зеленого
SEIZE   Perehod ; цвета светофора и освобождения перехода от пешеходов.
DEPART  Ocher1 ; Выход из очереди машин перед переходом.
ADVANCE 10,2 ; Пересечение машины перехода.
RELEASE Perehod ; Освобождение перехода от машины.
TERMINATE 1 ; Удаление машины из системы.
*****
* Моделирование потока пешеходов *
*****
GENERATE 30,10 ; формирование потока пешеходов у перехода.
QUEUE   Ocher2 ; Вход в очередь пешеходов перед переходом.
TEST E  X$Svet_Pesheh,F$Perehod ; Запрет входа на переход до появления зеленого
SEIZE   Perehod ; цвета светофора и освобождения перехода от машин.
DEPART  Ocher2 ; Выход из очереди пешеходов перед переходом.
ADVANCE 10,2 ; Пересечение пешеходов перехода.
RELEASE Perehod ; Освобождение перехода от пешехода.
TERMINATE 1 ; Удаление пешехода из системы.
*****

```

Рис. 3.22. Первая часть имитационной модели пешеходного перехода

```

Peshehod_1
*****
* Моделирование работы светофора *
*****
GENERATE ,,,1 ; Генерирование состояния светофора.
Begin1  SAVEVALUE Svet_Pesheh,Krasn ; Сохранение красного цвета светофора со стороны пешехода
        SAVEVALUE Svet_Avtom,Zelen ; Сохранение зеленого цвета светофора со стороны машин.
        ADVANCE Zelen_time ; Время горения зеленого цвета светофора.
        SAVEVALUE Svet_Pesheh,Zelen ; Сохранение зеленого цвета светофора со стороны пешехода
        SAVEVALUE Svet_Avtom,Krasn ; Сохранение красного цвета светофорасо стороны машин.
        ADVANCE Krasn_time ; Время горения красного цвета светофора.
        TRANSFER ,Begin1 ; Переход коператору с меткой Begin1.
Zelen_time EQU 200 ; Время горения зеленого цвета светофора.
Zelen EQU 0
Krasn EQU 100
Krasn_time EQU 300 ; Время горения красного цвета светофора.
START 1000

```

Рис. 3.23. Вторая часть имитационной модели пешеходного перехода

Моделирование системы

Так как в имитационной модели имеется управляющая команда **START**, то исходная имитационная модель будет транслироваться, и начнется процесс моделирования системы.

Однако перед началом моделирования желательно установить вывод тех параметров моделирования, которые нужны пользователю. Для этого:

- щелкните по пункту **Edit** (Правка) главного меню системы или нажмите комбинацию клавиш **Alt+E**. Появится выпадающее меню;
- щелкните по пункту **Settings** (Установки) выпадающего меню. Появится диалоговое окно **SETTINGS** для данной модели. В нашем примере появится окно под именем **Peshhod – REPORT – SETTINGS**, в котором устанавливаются нужные выходные данные; оно может выглядеть так, как показано на рис. 3.24.

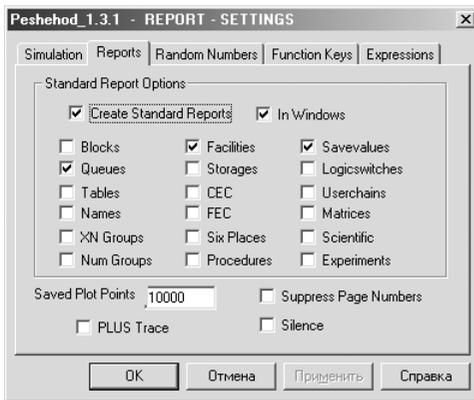


Рис. 3.24. Окно **SETTINGS** с установками для имитационной модели пешеходного перехода

Наличие галочек в окошках говорит о том, что эта информация будет выведена в окне результатов моделирования **REPORT** (Отчет). В нашем примере будет выведена информация для трех типов объектов:

- **Queues** (Очереди);
- **Savevalues** (Сохраняемые величины);
- **Facilities** (Каналы обслуживания).

Окно **REPORT** представлено на рис. 3.25.

В верхней строке указываются:

- **START TIME** (Начальное время) – 0.000;
- **END TIME** (Время окончания) – 13672.492;
- **BLOCKS** (Число блоков) – 24;
- **FACILITIES** (Число каналов обслуживания) – 1;
- **STORAGES** (Число накопителей) – 0.

Ниже указываются результаты моделирования канала обслуживания под именем **PEREHOD**:

- **ENTRIES** (Число входов) – 1000;
- **UTIL.** (Коэффициент использования) – 0.730;

```

GPSS World Simulation Report - Peshehod_1.1.1

Saturday, January 05, 2002 20:39:33

START TIME          END TIME  BLOCKS  FACILITIES  STORAGES
0.000              13672.492    24      1           0

FACILITY           ENTRIES  UTIL.   AVE. TIME AVAIL.  OWNER  PEND  INTER  RETRY  DELAY
PEREHOD            1000    0.730   9.977  1          0     0     0     135   0

QUEUE              MAX CONT. ENTRY ENTRY(0) AVE.CONT. AVE.TIME  AVE.(-0)  RETRY
OCHER1             137  129   681    8    69.218  1389.704  1406.223  0
OCHER2              8    6   454   179   1.988   59.856   98.817   0

SAVEVALUE          RETRY    VALUE
SVET_PESHEH        6        100.000
SVET_AVTOM         129      0

```

Рис. 3.25. Результаты первого варианта моделирования пешеходного перехода

- **AVE. TIME** (Среднее время обслуживания) – 9.977;
- **AVAIL.** (Доступность) – 1;
- **OWNER** – 0;
- **PEND** – 0;
- **INTER** – 0;
- **RETRY** (Повтор) – 135;
- **DELAY** (Отказ) – 0.

Еще ниже указываются результаты моделирования для очередей под именами OCHER1 и OCHER2 соответственно:

- **MAX** (Максимальное содержание) – 137, 8;
- **CONT.** (Текущее содержание) – 129, 6;
- **ENTRY** (Число входов) – 681, 454;
- **ENTRY(0)** (Число нулевых входов) – 8, 179;
- **AVE.CONT.** (Среднее число входов) – 69.218, 1.988;
- **AVE.TIME** (Среднее время) – 1389.704, 59.856;
- **AVE.(-0)** – 1406.223, 98.817;
- **RETRY** – 0, 0.

Затем указываются значения сохраняемых величин под именами SVET_AVTOM и SVET_PESHEH соответственно:

- **RETRY** – 129, 6;
- **VALUE** (Значение) – 0, 100.000.

Попробуем изменить длительность включенного состояния зеленого сигнала светофора. Установим его равным 400 с. Это будет выглядеть так:

```
Zelen_time EQU 400
```

Затем снова запустим программу на выполнение:

```
START 1000
```

Получим другие результаты моделирования. Фрагмент полученных результатов моделирования представлен на рис. 3.26.

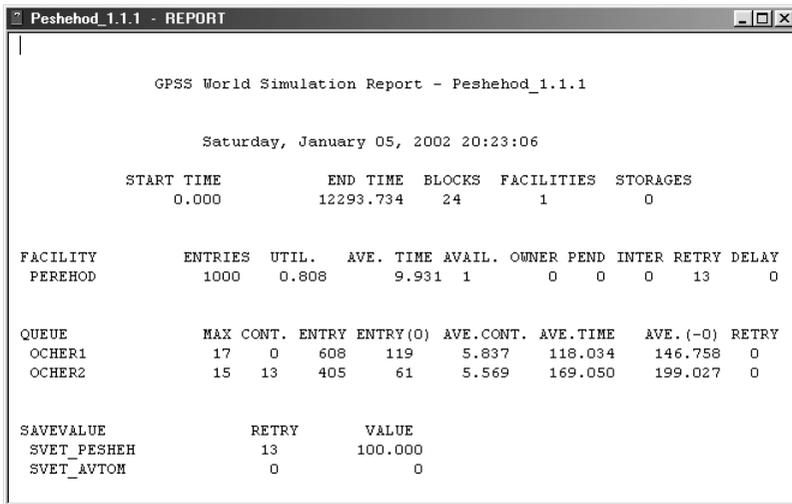


Рис. 3.26. Результаты второго варианта моделирования пешеходного перехода

В верхней строке указывается:

- **START TIME** – 0.000;
- **END TIME** – 12293.734;
- **BLOCKS** – 24;
- **FACILITIES** – 1;
- **STORAGEES** – 0.

Ниже указываются результаты моделирования канала обслуживания под именем PEREHOD:

- **ENTRIES** – 1000;
- **UTIL.** – 0.808;
- **AVE. TIME** – 9.931;
- **AVAIL.** – 1;

- **OWNER** – 0;
- **PEND** – 0;
- **INTER** – 0;
- **RETRY** – 13;
- **DELAY** – 0.

Еще ниже указываются результаты моделирования для очередей под именами OSHER1 и OSHER2 соответственно:

- **MAX** – 17, 15;
- **CONT.** – 0, 13;
- **ENTRY** – 608, 405;
- **ENTRY(0)** – 119, 61;
- **AVE.CONT.** – 5.837, 5.569;
- **AVE.TIME** – 118.034, 169.050;
- **AVE.(–0)** – 146.758, 199.027;
- **RETRY** – 0, 0.

Еще ниже указываются значения сохраняемых величин под именами SVET_AVTOM и SVET_PESHEN соответственно:

- **RETRY** – 0, 13;
- **VALUE** – 0, 100.000.

Визуализация процесса функционирования моделируемой системы

Перед началом моделирования, а точнее после трансляции модели, система GPSSW обеспечивает возможность визуального наблюдения перемещения активных требований (транзактов) в процессе моделирования. Если в модели есть команда управления **START**, она должна быть заблокирована, то есть переведена в комментарии. Для этого в позиции 1 поставьте звездочку. Процесс визуального наблюдения перемещения активных требований в процессе моделирования включает следующие этапы:

- щелкните по пункту **Window** главного меню системы или нажмите комбинацию клавиш **Alt+W**. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** (Окно моделирования) выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Block Entities** (Блочные элементы) всплывающего меню. Появится диалоговое окно **BLOCK ENTITIES** (рис. 3.27).

Для визуализации перемещения активных требований в процессе моделирования:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **START**. Появится диалоговое окно **Start Command**;

Loc	Block Type	Current Count	Entry Count	Retry Chain	Line Number	Include File
1 GEN	GENERATE	0	0	0	5	0
2 QUE	QUEUE	0	0	0	6	0
3 TES	TEST	0	0	0	7	0
4 SEI	SEIZE	0	0	0	8	0
5 DEP	DEPART	0	0	0	9	0
6 ADV	ADVANCE	0	0	0	10	0
7 REL	RELEASE	0	0	0	11	0
8 TER	TERMINATE	0	0	0	12	0
9 GEN	GENERATE	0	0	0	13	0
10 QUE	QUEUE	0	0	0	14	0
11 TES	TEST	0	0	0	15	0
12 SEI	SEIZE	0	0	0	16	0
13 DEP	DEPART	0	0	0	17	0
14 ADV	ADVANCE	0	0	0	18	0
15 REL	RELEASE	0	0	0	19	0
16 TER	TERMINATE	0	0	0	20	0
17 GEN	GENERATE	0	0	0	24	0
18 BEGIN1	SAVEVALUE	0	0	0	25	0
19 SAV	SAVEVALUE	0	0	0	26	0
20 ADV	ADVANCE	0	0	0	27	0
21 SAV	SAVEVALUE	0	0	0	28	0
22 SAV	SAVEVALUE	0	0	0	29	0
23 ADV	ADVANCE	0	0	0	30	0

Рис. 3.27. Блок-схема моделируемой системы пешеходного перехода

- щелкните по кнопке **OK**. Начнется процесс поступления и перемещения активных требований (автомашин) в моделируемой системе (на пешеходном переходе). Каждое перемещение требования по блокам системы фиксируется в правой части окна;
- щелкните по кнопке **Halt** (Остановить), расположенной на панели кнопок управления в верхней правой части окна **BLOCK ENTITIES**. Одно из состояний моделирования системы, представленное в окне **BLOCK ENTITIES**, показано на рис. 3.28;
- щелкните по кнопке **Continue** (Продолжить) для продолжения моделирования или по кнопке **Step** (Шаг), чтобы промоделировать и просмотреть изменения в системе в течение одного шага. Кнопку **Step** можно нажимать многократно для подробного просмотра последовательных шагов моделирования системы.

Можно визуально наблюдать перемещения активных требований в процессе моделирования в том же окне **BLOCK ENTITIES**, но только без излишних деталей. Для этого:

- щелкните по пункту **View** (Вид) главного меню системы или нажмите комбинацию клавиш **Alt+V**. Появится выпадающее меню;
- щелкните по пункту **Entity Details** (Детальное представление) выпадающего меню. По умолчанию перед пунктом **Entity Details** стоит галочка, что означает установленный режим детального представления моделируемой системы. После щелчка по нему мышью лишняя информация исчезнет, и окно **BLOCK ENTITIES** будет для нашей задачи выглядеть так, как показано на рис. 3.29.

Loc	Block Type	Current Count	Entry Count	Retry Chain	Line Number	Include File
1 GEN	GENERATE	0	4	0	5	0
2 QUE	QUEUE	0	4	0	6	0
3 TES	TEST	0	4	0	7	0
4 SEI	SEIZE	0	4	0	8	0
5 DEP	DEPART	0	4	0	9	0
6 ADV	ADVANCE	1	4	0	10	0
7 REL	RELEASE	0	3	0	11	0
8 TER	TERMINATE	0	3	0	12	0
9 GEN	GENERATE	0	2	0	13	0
10 QUE	QUEUE	2	2	0	14	0
11 TES	TEST	0	0	0	15	0
12 SEI	SEIZE	0	0	0	16	0
13 DEP	DEPART	0	0	0	17	0
14 ADV	ADVANCE	0	0	0	18	0
15 REL	RELEASE	0	0	0	19	0
16 TER	TERMINATE	0	0	0	20	0
17 GEN	GENERATE	0	1	0	24	0
BEGIN1	SAVEVALUE	0	1	0	25	0
19 SAV	SAVEVALUE	0	1	0	26	0
20 ADV	ADVANCE	1	1	0	27	0
21 SAV	SAVEVALUE	0	0	0	28	0
22 SAV	SAVEVALUE	0	0	0	29	0
23 ADV	ADVANCE	0	0	0	30	0

Рис. 3.28. Одно из состояний моделирования системы пешеходного перехода в детальном представлении

1 GEN	24 TRA
2 QUE	
3 TES	
4 SEI	
5 DEP	
6 ADV	
7 REL	
8 TER	
9 GEN	
10 QUE	
11 TES	
12 SEI	
13 DEP	
14 ADV	
15 REL	
16 TER	
17 GEN	
BEGIN1	
19 SAV	
20 ADV	
21 SAV	
22 SAV	
23 ADV	

Рис. 3.29. Одно из состояний моделирования системы пешеходного перехода в упрощенном представлении

Следует помнить, что включение окна просмотра **BLOCK ENTITIES** значительно замедляет процесс моделирования, так как тратится время на визуализацию процесса моделирования. Для ускорения процесса моделирования и быстрого получения конечного результата, целесообразно закрыть окно **BLOCK ENTITIES**. Это можно сделать несколькими способами:

- щелкнуть по кнопке с крестиком, расположенной в правом верхнем углу окна;
- щелкнуть по пиктограмме блока, расположенной в левом верхнем углу окна. Появится всплывающее меню, в котором щелкните по пункту **Close** (Закрыть);
- нажать комбинацию клавиш **Ctrl+F4**;
- дважды щелкнуть по пиктограмме блока, расположенной в левом верхнем углу окна.

Моделирование работы переговорного пункта

Постановка задачи

Допустим, нам надо промоделировать работу переговорного пункта, который имеет одно помещение для трех посетителей. Известны следующие параметры функционирования переговорного пункта. Поток посетителей (требований), приходящих на переговорный пункт, равномерный. Интервал между прибытиями посетителей колеблется в пределах от 0,85 до 2,85 мин включительно, или $1,85 \pm 1$ мин. Время оплаты каждого переговора составляет $1,5 \pm 0,4$ мин, а время разговора посетителей по телефону – $4,4 \pm 1,35$ мин. Время ожидания вызова абонента составляет $3,5 \pm 1,1$ мин. Время разговора посетителей, оплаты разговора и ожидания вызова абонента подчиняется равномерному распределению вероятностей. Если все телефоны переговорного пункта заняты, то посетитель ожидает освобождения одного из них.

Требуется определить параметры функционирования переговорного пункта:

- коэффициент загрузки переговорного пункта;
- максимальное, среднее и текущее число посетителей в переговорном пункте;
- среднее время обслуживания в переговорном пункте и др.

Выявление основных особенностей

Для моделирования работы переговорного пункта необходимо сформировать входной поток посетителей (требований) и временной интервал моделирования работы переговорного пункта. Но перед этим необходимо выбрать единицу измерения времени. Для моделирования работы переговорного пункта можно взять в качестве единицы измерения времени минуту.

Создание имитационной модели процесса

Построение имитационной модели начнем с создания заголовка модели, который может быть представлен, например, в таком виде:

```
;GPSS World - PEREGOV_PUNKT.GPS
*****
*   Моделирование переговорного пункта   *
*****
```

Программу работы переговорного пункта можно представить в виде трех секторов.

В первом секторе указывается вместимость переговорного пункта. Это можно выполнить с помощью оператора **STORAGE** (Накопитель), который в нашем примере будет выглядеть так:

```
Punkt STORAGE 4
```

В этом же секторе используем оператор **TABLE** для формирования таблицы с информацией об использовании посетителями переговорного пункта:

```
Transit TABLE M1, .5, 1, 3
```

В операторе **TABLE** в поле операнда А используется стандартный числовой атрибут M1, который фиксирует время прохождения требования, находящегося в стадии обслуживания.

Во втором секторе будем моделировать поток посетителей и работу переговорного пункта.

Моделирование потока посетителей будем выполнять с помощью оператора **GENERATE** (Генерировать). В нашем примере он будет выглядеть так:

```
GENERATE 1.85, 1
```

В поле операнда А указывается средний интервал времени между прибытием в переговорный пункт двух идущих один за другим посетителей (требований, транзактов). В нашем примере среднее время прибытия требований составляет 1,85 мин. В поле операнда В дано отклонение времени поступления посетителей от среднего. В нашем примере оно составляет 1 мин.

Посетитель, пришедший на переговорный пункт, сначала определяет количество посетителей переговорного пункта. Если их в пункте уже три, то пункт занят, и новый посетитель ожидает его освобождения. Это действие можно промоделировать с помощью оператора **GATE SNF** (Storage Not Full – Накопитель не полон):

```
Povtor GATE SNF Punkt, Zanyt
```

Если накопитель под символьным именем Punkt не полон, то посетитель входит в переговорный пункт. Это моделируется оператором **ENTER** (Войти):

```
ENTER Punkt
```

Далее посетитель встает в очередь, если она есть, для оформления телефонного разговора. Это можно промоделировать оператором **QUEUE** (Очередь), который только в совокупности с соответствующим оператором **DEPART** (Выйти) собирает статистическую информацию о работе моделируемой очереди. В нашем примере оператор **QUEUE** будет выглядеть так:

```
QUEUE Ocher_kassir
```

В поле операнда А дается символьное или числовое имя очереди. Таких очередей в системе может быть очень много. В нашей задаче очереди дано имя *Ocher_kassir*. Желательно, чтобы присваиваемое имя отражало суть описываемого элемента системы. Посетитель может выйти из очереди только тогда, когда освободится кассир (канал обслуживания). Для этого вводится оператор **SEIZE** (Занять), который определяет занятость канала обслуживания, и при его освобождении очередное требование выходит из очереди и идет в канал на обслуживание. Это может выглядеть так:

```
SEIZE Kassir
```

В поле операнда А дается символьное или числовое имя канала обслуживания. Таких каналов обслуживания в системе может быть также много. В нашей задаче каналу дано имя *Kassir*. Желательно, чтобы присваиваемое имя отражало суть описываемого элемента системы.

Выход покупателя из очереди в кассу фиксируется оператором **DEPART** с соответствующим названием очереди. В нашем примере это будет выглядеть так:

```
DEPART Ocher_kassir
```

Далее должно быть промоделировано время оплаты разговора. Это время в нашем примере составляет $1,5 \pm 0,4$ мин. Для моделирования этого процесса используется оператор **ADVANCE** (Задержать), который в нашей задаче будет выглядеть так:

```
ADVANCE 1.5,0.4
```

После обслуживания кассиром посетитель идет к телефону для ведения переговоров. Однако перед этим системе должно быть послано сообщение об освобождении канала обслуживания (кассира). Это делается с помощью оператора **RELEASE** (Освободить), который в нашей задаче записывается так:

```
RELEASE Kassir
```

Следует особо подчеркнуть, что парные операторы **QUEUE** и **DEPART** для каждой очереди должны иметь одно и то же, но свое уникальное имя. Это же относится и к операторам **SEIZE** и **RELEASE**.

После обслуживания в кассе посетитель направляется к телефонному аппарату. Это может быть промоделировано оператором **ADVANCE**:

```
ADVANCE 4.4,1.35
```

После проведения переговоров посетитель (требование) освобождает телефонный аппарат (покидает систему). Это действие может быть представлено оператором **LEAVE** (Оставить):

```
LEAVE Punkt
```

После этого посетитель покидает систему – переговорный пункт. Это действие может быть промоделировано оператором **TERMINATE**:

```
TERMINATE
```

В третьем секторе определяется время моделирования системы. Этот процесс может быть представлен такой совокупностью операторов:

```
GENERATE    480
TERMINATE   1
START       1
```

Первый оператор – **GENERATE** – в третьем секторе определяет время моделирования системы – 480 мин. Второй – **TERMINATE** – определяет уменьшение общего времени моделирования на 1 мин. Завершающим оператором в нашей задаче является управляющая команда **START**, позволяющая начать моделирование.

Представление имитационной модели

Для представления имитационной модели выполните следующие действия:

- щелкните по пункту **File** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **New** (Создать) выпадающего меню. Появится диалоговое окно **Новый документ**;
- выделите пункт **Model** и щелкните по кнопке **ОК**. Появится окно модели, в котором введите данную программу. Это показано на рис. 3.30.

Чтобы вызвать окно для представления имитационной модели в системе GPSSW, можно также нажать комбинацию клавиш **Ctrl+Alt+S**.

Подготовка к моделированию системы

Перед началом моделирования можно установить вывод тех параметров моделирования, которые нужны пользователю. Для этого:

- щелкните по пункту **Edit** (Правка) главного меню системы или нажмите комбинацию клавиш **Alt+E**. Появится выпадающее меню;
- щелкните по пункту **Settings** (Установки) выпадающего меню. Появится диалоговое окно **SETTINGS** для данной модели, в котором можно установить нужные выходные данные. Для нашего примера это может выглядеть так, как показано на рис. 3.31.

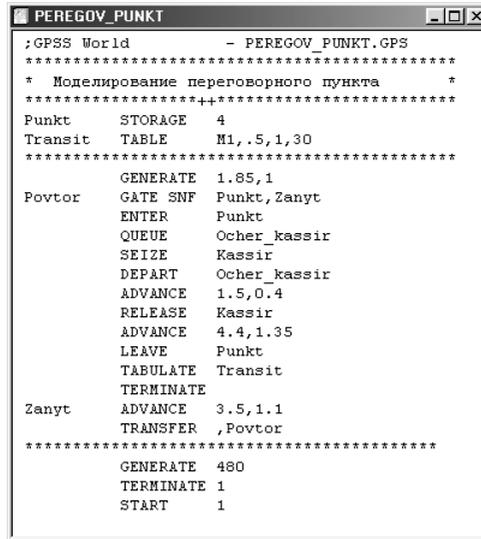


Рис. 3.30. Окно имитационной модели переговорного пункта

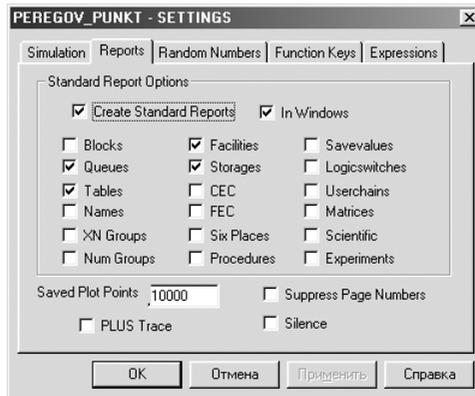


Рис. 3.31. Окно **SETTINGS** с установками для имитационной модели переговорного пункта

Наличие галочки в окошках говорит о том, что эта информация будет выведена в окне результатов моделирования. В нашем примере будет выведена информация для следующих объектов:

- **Queues** (Очереди);
- **Tables** (Таблицы/гистограммы);
- **Facilities** (Каналы обслуживания);
- **Storages** (Накопители).

Моделирование системы

Для начала моделирования:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню. Начнется трансляция исходной модели, а затем и ее выполнение, так как в программе имеется управляющая команда **START**, обеспечивающая автоматическое выполнение оттранслированной программы. После завершения программы появится информация о трансляции и выполнении в окне **JOURNAL** и результаты работы программы в окне **REPORT** (рис. 3.32).

GPSS World Simulation Report - PEREGOV_PUNKT.1.1
 Wednesday, January 02, 2002 19:41:14

	START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
	0.000	480.000	16	1	1

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
KASSIR	260	0.802	1.481	1	261	0	0	0	0

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY
OCHER_KASSIR	2	0	260	116	0.240	0.442	0.799	0

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE. C.	UTIL.	RETRY	DELAY
PUNKT	4	1	0	4	260	1	3.413	0.853	0	0

TABLE	MEAN	STD. DEV.	RANGE	RETRY FREQUENCY	CUM. %	
TRANSIT	10.098	7.012		0		
			3.500 -	4.500	2	0.78
			4.500 -	5.500	31	12.84
			5.500 -	6.500	49	31.91
			6.500 -	7.500	51	51.75
			7.500 -	8.500	27	62.26
			8.500 -	9.500	10	66.15
			9.500 -	10.500	15	71.98

For Help, press F1 Report is Complete. Clock

Рис. 3.32. Окно **REPORT** с фрагментом результатов моделирования работы переговорного пункта

В верхней строке указываются:

- **START TIME** (Начальное время) – 0.000;
- **END TIME** (Время окончания) – 480.000;
- **BLOCKS** (Число блоков) – 16;
- **FACILITIES** (Число каналов обслуживания) – 1;
- **STORAGES** (Число накопителей) – 1.

Ниже указываются результаты моделирования канала обслуживания под именем KASSIR:

- **ENTRIES** (Число входов) – 260;
- **UTIL.** (Коэффициент использования) – 0.802;
- **AVE. TIME** (Среднее время обслуживания) – 1.481;
- **AVAIL.** (Доступность) – 1;
- **OWNER** – 261;
- **PEND** – 0;
- **INTER** – 0;
- **RETRY** (Повторные входы) – 0;
- **DELAY** (Отказано) – 0.

Еще ниже указываются результаты моделирования очереди по имени ОСНЕР_KASSIR:

- **MAX** (Максимальное содержание) – 2;
- **CONT.** (Текущее содержание) – 0;
- **ENTRY** (Число входов) – 260;
- **ENTRY(0)** (Число нулевых входов) – 116;
- **AVE.CONT.** (Среднее содержимое) – 0.240;
- **AVE.TIME** (Среднее время) – 0.442;
- **AVE.(–0)** – 0.799;
- **RETRY** – 0.

Ниже указываются результаты функционирования накопителя (STORAGE) под именем Punkt:

- **CAP.** (Capacity – Вместимость) – 4;
- **REM.** (Remove – Удален) – 1;
- **MIN.** (Минимальное содержимое) – 0;
- **MAX.** (Максимальное содержимое) – 4;
- **ENTRIES** (Число входов) – 260;
- **AVL.** (Доступность) – 1;
- **AVE.C.** (Средняя вместимость) – 3.413;
- **UTIL.** (Коэффициент использования) – 0.853;
- **RETRY** – 0;
- **DELAY** – 0.

Еще ниже указываются результаты моделирования для построения по табличным данным соответствующей гистограммы **TABLE** функционирования накопителя под именем TRANSIT:

- **MEAN** (Средняя) – 10.098;
- **STD.DEV.** (Среднее квадратическое отклонение) – 7.012;
- **RANGE** (Область);
- **RETRY** – 0;
- **FREQUENCY** (Частота);
- **CUM.%** (Суммарный процент).

Вывод гистограммы

При наличии оператора **TABLE** собирается соответствующая информация для построения гистограммы, иллюстрирующей распределение времени прохождения требования в стадии обслуживания. Этот оператор выглядит так:

```
Transit TABLE M1, .5, 1, 3
```

Можно вывести соответствующую гистограмму. Для этого:

- щелкните по пункту **Window** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** (Окно моделирования) выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Table Window** (Окно гистограммы) во всплывающем меню. Появится диалоговое окно **Open Table Window** (Открыть окно гистограммы). В раскрывающемся списке **Table** щелкните по нужной гистограмме, если их в программе несколько. Для нашей задачи диалоговое окно **Open Table Window** выглядит так, как показано на рис. 3.33;
- щелкните по кнопке **OK**. Появится соответствующая гистограмма. Для нашей задачи она выглядит так, как показано на рис. 3.34.

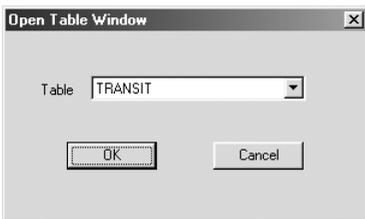


Рис. 3.33. Диалоговое окно выбора нужной гистограммы имитационной модели переговорного пункта

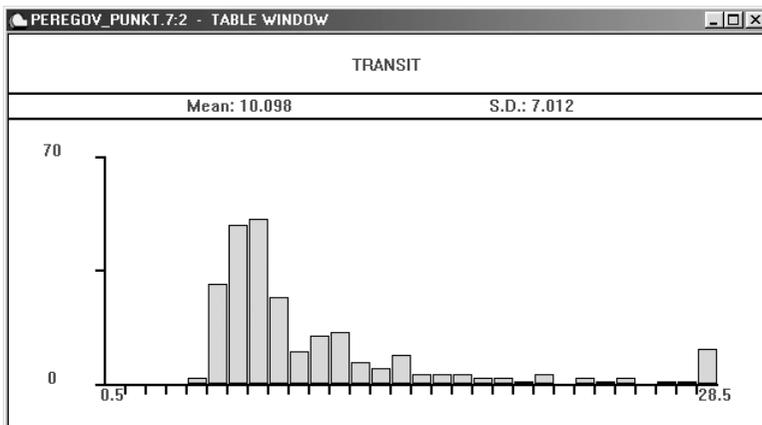


Рис. 3.34. Окно гистограммы имитационной модели переговорного пункта

Моделирование работы супермаркета

Постановка задачи

Допустим, нам надо промоделировать работу супермаркета, который имеет место для парковки 100 автомобилей. Если все места парковки автомобилей заняты, то автомобиль покидает супермаркет. Время прихода покупателей с места парковки автомобиля в магазин подчиняется равномерному закону и составляет 60 ± 40 с. Моделирование будем производить в течение рабочей смены, равной 8 ч. Магазин имеет 100 ручных тележек и 50 корзин для транспортировки купленных товаров. Магазин имеет пять кассовых аппаратов (кассиров), но первый из них предназначен для быстрого обслуживания покупателей с единичными покупками. Поток покупателей (требований), приходящих в магазин за покупками, экспоненциальный.

Экспоненциальное распределение вероятности представляется в таком виде:

$$f(x) = \frac{1}{\beta} e^{-\frac{(x-\lambda)}{\beta}}, \quad \beta \geq 0.$$

В противном случае оно равно 0.

При этом известно, что в интервале времени:

- от 0 до 30 мин параметры λ и β соответственно равны 0 и 60;
- от 30 до 90 мин параметры λ и β соответственно равны 0 и 40;
- от 90 до 150 мин параметры λ и β соответственно равны 0 и 80;
- свыше 150 мин параметры λ и β соответственно равны 0 и 120.

Если производится покупка более 10 видов товаров, то необходима тележка, в противном случае используется корзина. После этого покупатели подходят к нужным стеллажам для взятия товара. Число взятых товаров определяется с помощью датчика случайных чисел.

Требуется определить параметры функционирования супермаркета:

- коэффициент загрузки всех касс;
- максимальное, среднее и текущее число покупателей в каждой очереди;
- среднее время обслуживания в каждом канале обслуживания;
- среднее время нахождения покупателя в каждой очереди и др.

Также нужно построить следующие гистограммы:

- времени нахождения покупателей в системе;
- числа покупок, сделанных в течение смены;
- числа покупателей, посетивших супермаркет в течение смены.

Выявление основных особенностей

Для моделирования работы супермаркета необходимо сформировать входной поток покупателей (требований) и временной интервал моделирования работы

супермаркета. Но перед этим необходимо выбрать единицу измерения времени. Для моделирования работы супермаркета можно взять в качестве единицы измерения времени секунду.

Тогда общее время моделирования составит $8 \times 60 \times 60$ с.

Создание имитационной модели процесса

Построение имитационной модели начнем с создания заголовка модели, который может быть представлен, например, в таком виде:

```
; GPSSW File           - SUPER_MAGAZIN.GPS
*****
*   Моделирование работы супермаркета   *
*****
```

Далее разобьем всю модель на несколько секторов.

В первом секторе введем информацию, необходимую для функционирования супермаркета. Этот сектор может выглядеть так:

```
RMULT      1187
Kassa_2    EQU      2
Kassa_N    EQU      5
Time_work  VARIABLE 8#60#60
N_Pokupok  VARIABLE (RN1@96+5)
Finance    VARIABLE (RN1@3+1)#40+150
Time_system TABLE  M1,1000,1000,7
Pokupki    TABLE  P$Kol_pokupok,10,10,10
N_Pokupatel TABLE  X$Pokupatel,100,50,12
Park       STORAGE  100
Telejka    STORAGE  100
Korzina    STORAGE  50
Kassir     VARIABLE (P$Kol_pokupok)#2+P$Oplata
Time_mag   VARIABLE P$Kol_pokupok#100
INITIAL    X$Pokupatel,0
```

Оператор **RMULT** определяет набор начальных чисел первых семи генераторов случайных чисел.

Оператор **EQU** указывает номер объекта. В нашем примере определяются номера каналов обслуживания со 2 по 5 и соответствующие очереди к ним.

Переменная под именем *Time_work* определяет продолжительность моделирования супермаркета в течение рабочей смены – 8 ч – в секундах.

Переменная под именем *N_Pokupok* определяет число покупок, сделанных одним покупателем. Переменная под именем *Finance* определяет размер оплаты покупок. Операторы **TABLE** создают таблицы, представляющие собой набор целых чисел для построения соответствующих гистограмм. Каждое целое число представляет класс частоты в гистограмме.

Оператор **STORAGE** (Накопитель) определяет вместимость стоянки автомобилей в количестве 100 единиц, наличие ручных тележек в количестве 100 штук и корзин в количестве 50 единиц.

Переменная *Kassir* определяет время, затраченное кассиром на обслуживание покупателя. Переменная *Time_mag* определяет время нахождения покупателя в магазине.

Оператор **INITIAL** определяет начальное значение сохраняемой величины по имени *Pokupatel*. Ее значение в начале смены равно 0.

Во втором секторе моделируется поступление покупателей в супермаркет, определяется число покупок, их стоимость и направление покупателей к местоположению корзинок или ручных тележек. Этот сектор может выглядеть так:

```
Parking  TRANSFER  Both,,Lost
          ENTER    Park
          ADVANCE  60,40
          SAVEVALUE  Pokupatel+,1
          ASSIGN   Kol_pokupok,V$N_Pokupok
          ASSIGN   Oplata,V$Finance
          TEST LE  P$Kol_pokupok,10,QTelejka
          GATE SNF  Korzina,QTelejka
```

Когда требование входит в оператор **TRANSFER**, оно сначала направляется к следующему блоку; если тот не допускает требование – то к блоку с меткой *Lost*. А это означает, что автомобиль покидает систему – супермаркет. Если есть место, автомобиль въезжает на стоянку, что моделируется оператором **ENTER**. Затем покупатель идет в супермаркет. Это моделируется оператором **ADVANCE 60,40**. Время прихода покупателей с места парковки автомобиля в супермаркет подчиняется равномерному закону и составляет 60 ± 40 с.

В операторе **SAVEVALUE** в сохраняемой величине по имени *Pokupatel* происходит подсчет поступивших в супермаркет покупателей с начала открытия магазина до окончания смены.

Оператор **ASSIGN** (Присвоить) выполняет присваивание параметру требования (покупателя) по имени *Kol_pokupok* с помощью переменной *V\$N_Pokupok* (Число покупок).

Следующий оператор **ASSIGN** выполняет присваивание параметру требования (покупателя) по имени *Oplata* с помощью переменной *V\$Finance* (Размер оплаты покупок).

Оператор **TEST LE** определяет дальнейший путь следования покупателя. Если величина параметра требования по имени *P\$Kol_pokupok* (Число покупаемых товаров) меньше или равна 10, то требование перемещается к следующему оператору; если это условие не выполняется, то требование перемещается к оператору с символической меткой *Qtelejka*.

Далее оператор **GATE SNF** проверяет условие *SNF* (*Storage Not Full* – Накопитель не полон). Если есть в наличии корзины, то требование переходит к следующему

оператору, если нет – перемещается к оператору с символической меткой Qtelejka.

В третьем секторе моделируется сбор статистической информации по очереди, связанной с получением корзинок. Этот сектор может выглядеть так:

```

QUEUE      Korzina_Q
ENTER      Korzina
DEPART     Korzina_Q
ASSIGN     Tara,Korzina
TRANSFER  ,Magazin

```

Операторы **QUEUE** и **DEPART** собирают статистическую информацию по очереди, связанной с получением корзин, с именем Korzina_Q.

С помощью операторов **ENTER** и **LEAVE** система собирает информацию об использовании корзин покупателями.

Оператор **ASSIGN** выполняет присваивание параметру требования (покупателя) по имени Tara имя Korzina. Далее с помощью оператора **TRANSFER** происходит переход требования к оператору с символической меткой Magazin. То есть покупатель непосредственно входит в магазин для выбора необходимых покупок.

В четвертом секторе моделируется сбор статистической информации по очереди, связанной с получением тележек. Этот сектор может выглядеть так:

```

QTelejka QUEUE  Telejka_Q
              ENTER  Telejka
              DEPART Telejka_Q
              ASSIGN Tara,Telejka

```

Операторы **QUEUE** и **DEPART** собирают статистическую информацию по очереди, связанной с получением тележек, с именем Telejka_Q.

С помощью операторов **DEPART** и **LEAVE** система собирает информацию об использовании тележек покупателями.

Оператор **ASSIGN** выполняет присваивание параметру требования (покупателя) по имени Tara имя Telejka.

В пятом секторе моделируется перемещение требования в системе в зависимости от состояния системы. Этот сектор может выглядеть так:

```

Magazin ADVANCE V$Time_mag
        TEST LE  P$Kol_pokupok,10,Min_och
        COUNT L  Kassir_0,Kassa_2,Kassa_N,1,Q
        TEST E   P$Kassir_0,0,Min_och

```

Оператор **ADVANCE** моделирует время нахождения покупателя в магазине, используя переменную по имени Time_mag.

Оператор **TEST LE** определяет дальнейший путь следования покупателя. Если значение параметра требования по имени P\$Kol_pokupok меньше или равно 10, то требование перемещается к следующему оператору; если это условие не выполняется, то требование перемещается к оператору с символической меткой Min_och.

Оператор **COUNT L** определяет число объектов из определенного диапазона объектов, удовлетворяющих заданному условию. При этом счетчик объектов, удовлетворяющих заданному условию, помещается в параметр активного требования по имени *Kassir_0* (в поле операнда *A*). В поле операнда *B* указано имя объекта, находящегося в начале диапазона объектов. Его имя *Kassa_2*. В поле операнда *C* указывается имя объекта, находящегося в конце диапазона объектов.

В поле операнда *D* указывается число, которое сравнивается со значением атрибута объекта, заданного в операнде *E*, то есть со значением очереди.

Оператор **TEST E** определяет дальнейший путь следования покупателя. Если значение параметра требования по имени *P\$Kassir_0* (Число покупателей в очереди) равно 0, то требование перемещается к следующему оператору, если это условие не выполняется, то требование перемещается к оператору с символической меткой *Min_och*.

В шестом секторе собирается статистическая информация по очереди быстрого обслуживания по имени *Bistro_Q* и по каналу обслуживания по имени *Bistro*. Моделируется время обслуживания, выход из области взятия тары для покупок, а также переход к другому сектору.

Этот сектор может выглядеть так:

```

QUEUE      Bistro_Q
SEIZE      Bistro
DEPART     Bistro_Q
ADVANCE    V$Kassir
RELEASE    Bistro
LEAVE      P$Tara
TRANSFER  ,Fin
    
```

Операторы **QUEUE** и **DEPART** собирают статистическую информацию по очереди, связанной с быстрым обслуживанием, с именем *Bistro_Q*.

Операторы **SEIZE** и **RELEASE** собирают статистическую информацию по работе канала обслуживания по имени *Bistro*.

Оператор **ADVANCE** моделирует время быстрого обслуживания покупателей, которое определяется переменной по имени *Kassir*.

Оператор **LEAVE** завершает сбор статистической информации по работе накопителей, связанных с использованием тары.

Оператор **TRANSFER** направляет требования (покупателей) к оператору с символической меткой *Fin*.

В седьмом секторе собирается статистическая информация по очереди быстрого обслуживания по имени *Min_ochered* и по каналу обслуживания с тем же именем. Моделируется время обслуживания, выход из области взятия тары для покупок. Этот сектор может выглядеть так:

```

Min_och  SELECT MIN      Min_ochered,Kassa_2,Kassa_N,,Q
         QUEUE          P$Min_ochered
         SEIZE          P$Min_ochered
    
```

DEPART	P\$Min_ochered
ADVANCE	V\$Kassir
RELEASE	P\$Min_ochered
LEAVE	P\$Tara

Оператор **SELECT** (Выбрать) выбирает объект, удовлетворяющий заданному условию – MIN, – и записывает номер этого объекта в параметр активного требования. В поле операнда A указывается символьное имя параметра выбираемого объекта (обязательно) – Min_ochered. В поле операнда B задается нижняя граница диапазона имен очередей, для которых будет проверяться заданное условие (обязательно). В поле операнда C задается верхняя граница диапазона имен очередей, для которых будет проверяться заданное условие (обязательно). Этот операнд может представлять собой имя, целое положительное число, выражение в круглых скобках, стандартный числовой атрибут или SNA*параметр. Поле операнда D не используется с операторами **MIN** и **MAX**. В поле операнда E указывается имя класса стандартного числового атрибута. В нашем примере это Q (Queue).

Операторы **QUEUE** и **DEPART** собирают статистическую информацию по очередям, находящимся в параметре требования по имени Min_ochered.

Операторы **SEIZE** и **RELEASE** собирают статистическую информацию по работе каналов обслуживания, находящимся в параметре требования по имени Min_ochered.

Оператор **ADVANCE** моделирует время быстрого обслуживания покупателей, которое определяется переменной по имени Kassir.

Оператор **LEAVE** завершает сбор статистической информации по работе накопителей, связанных с использованием корзин и тележек.

В восьмом секторе собирается информация в табличном виде по времени работы системы и числу покупок. Моделируется время возвращения покупателя к автомобилю и удаление последнего из системы. Этот сектор может выглядеть так:

```

Fin TABULATE      Time_system
  TABULATE        Pokupki
  SAVEVALUE       Pokupatel-,1
  ADVANCE         60,50
  LEAVE           Park
  TERMINATE
Lost TERMINATE

```

Операторы **TABULATE** (Табулировать) выполняют сбор данных соответственно по времени обслуживания в системе и по числу сделанных покупок.

Оператор **SAVEVALUE** уменьшает число обслуженных покупателей.

Оператор **LEAVE** моделирует покидание автомобилем места стоянки. Предпоследний оператор **TERMINATE** последовательно уменьшает число требований – покупателей, находящихся в системе. Последний оператор **TERMINATE** имитирует потерю покупателя из-за нехватки мест на стоянке автомобилей.

В девятом секторе моделируются различные потоки покупателей в супермаркет. Этот сектор может выглядеть так:

```
GENERATE (Exponential(1,0,60)),,200
TRANSFER ,Parking
GENERATE (Exponential(1,0,40)),,1800,400
TRANSFER ,Parking
GENERATE (Exponential(1,0,80)),,5400,300
TRANSFER ,Parking
GENERATE (Exponential(1,0,120)),,9000
TRANSFER ,Parking
```

В операторе **GENERATE** в поле операнда A используется встроенное экспоненциальное распределение вероятностей для моделирования поступления покупателей в супермаркет.

В десятом секторе моделируется время работы системы. Этот сектор может выглядеть так:

```
GENERATE V$Time_work
TABULATE N_Pokupatel
TERMINATE 1
START 1
```

В операторе **GENERATE** в поле операнда A определяется время работы системы. Оператор **TABULATE** собирает информацию о числе покупателей, посетивших супермаркет. Оператор **TERMINATE** уменьшает на единицу время функционирования системы, а **START 1** запускает после компилирования программу на выполнение.

Представление имитационной модели

Для представления имитационной модели выполните следующие действия:

- щелкните по пункту **File** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **New** выпадающего меню. Появится диалоговое окно **Новый документ**;
- выделите пункт **Model** и щелкните по кнопке **ОК**. Появится окно модели, в котором введите данную программу. Эта модель представлена на рис. 3.35а–в.

Подготовка к моделированию системы

Перед началом моделирования можно установить вывод тех параметров моделирования, которые нужны пользователю. Для этого:

- щелкните по пункту **Edit** главного меню системы или нажмите комбинацию клавиш **Alt+E**. Появится выпадающее меню;

a)

```

Супер_Магазин
; GPSSW File                               | SUPER_MAGAZIN.GPS
*****
*      Моделирование работы супермаркета      *
*****
          RHULT      1187
Kassa_2   EQU        2
Kassa_N   EQU        5
Time_work VARIABLE  8#60#60
N_Pokupok VARIABLE  (RN1896+5)
Finance   VARIABLE  (RN183+1)#40+150
Time_system TABLE  M1,1000,1000,7
Pokupki   TABLE    P$Kol_pokupok,10,10,10
N_Pokupatel TABLE  X$Pokupatel,100,50,12
Park      STORAGE   100
Telejka   STORAGE   100
Korzina   STORAGE   50
Kassir    VARIABLE  (P$Kol_pokupok)#2+P$Oplata
Time_mag  VARIABLE  P$Kol_pokupok#100
          INITIAL    X$Pokupatel,0
*****
Parking   TRANSFER  Both,,Lost
          ENTER      Park
          ADVANCE    60,40
          SAVEVALUE  Pokupatel+,1
          ASSIGN     Kol_pokupok,V$N_Pokupok
          ASSIGN     Oplata,V$Finance
          TEST LE    P$Kol_pokupok,10,QTelejka
          GATE SNF   Korzina,QTelejka

```

б)

```

Супер_Магазин
*****
          QUEUE      Korzina_Q
          ENTER      Korzina
          DEPART     Korzina_Q
          ASSIGN     Tara,Korzina
          TRANSFER   ,Magazin
*****
QTelejka QUEUE      Telejka_Q
          ENTER      Telejka
          DEPART     Telejka_Q
          ASSIGN     Tara,Telejka
*****
Magazin  ADVANCE    V$Time_mag
          TEST LE    P$Kol_pokupok,10,Min_och
          COUNT L    Kassir_0,Kassa_2,Kassa_N,1,Q
          TEST E     P$Kassir_0,0,Min_och
*****
          QUEUE      Bistro_Q
          SEIZE      Bistro
          DEPART     Bistro_Q
          ADVANCE    V$Kassir
          RELEASE    Bistro
          LEAVE      P$Tara
          TRANSFER   ,Fin
*****
Min_och  SELECT MIN Min_ochered,Kassa_2,Kassa_N,,Q
          QUEUE      P$Min_ochered
          SEIZE      P$Min_ochered

```

Рис. 3.35а,б. Фрагменты имитационной модели супермаркета

- щелкните по пункту **Settings** выпадающего меню. Появится диалоговое окно **SETTINGS** для данной модели, в котором можно установить нужные выходные данные. Для нашего примера это окно может выглядеть так, как показано на рис. 3.36.

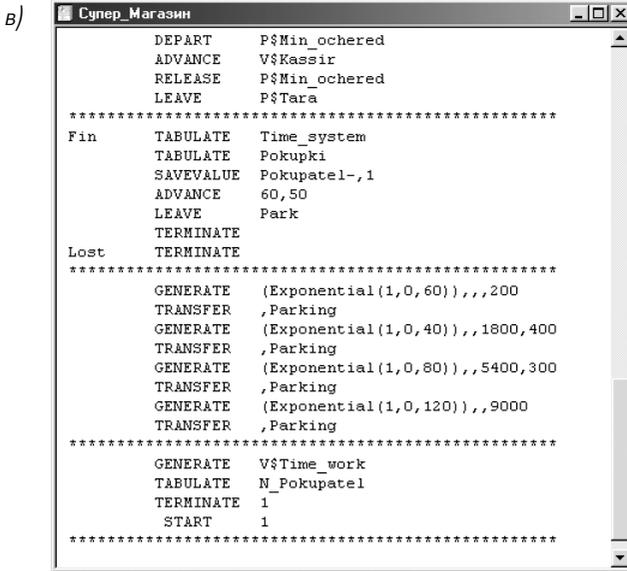


Рис. 3.35в. Фрагменты имитационной модели супермаркета



Рис. 3.36. Окно **SETTINGS** с установками для имитационной модели супермаркета

Наличие галочек в окошках говорит о том, что эта информация будет выведена в окне результатов моделирования. В нашем примере будет выведена информация по следующим объектам:

- **Queues** (Очереди);
- **Tables** (Таблицы/гистограммы);
- **Facilities** (Каналы обслуживания);
- **Storages** (Накопители);
- **Savevalues** (Сохраняемые величины).

Моделирование системы

После создания имитационную модель необходимо оттранслировать и запустить на выполнение. Для этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню.

Если управляющая команда **START** есть в модели, то исходная имитационная модель будет транслироваться, и если в ней нет ошибок, то начнется процесс моделирования системы. Будет выполняться то число прогонов, которое указано в поле A операнда команды **START**. Затем появится окно **JOURNAL**.

После завершения моделирования появится окно **REPORT** с результатами моделирования, представленными на рис. 3.37а,б.

Если управляющей команды **START** в модели нет, то исходная имитационная модель будет транслироваться, и если в ней нет ошибок, то будет получена оттранслированная модель в машинных кодах, готовая к выполнению моделирования.

В третьей строке сверху указывается:

- **START TIME** (Начальное время) – 0.000;
- **END TIME** (Время окончания) – 28800.000;
- **BLOCKS** (Число блоков) – 53;
- **FACILITIES** (Число каналов обслуживания) – 5;
- **STORAGES** (Число накопителей) – 3.

Ниже указываются результаты моделирования каналов обслуживания соответственно под номерами 2, 3, 4, 5 и BISTRO:

- **ENTRIES** (Число входов) – 83, 78, 72, 73, 18;
- **UTIL.** (Коэффициент использования) – 0.944, 0.879, 0.851, 0.819, 0.151;
- **AVE. TIME** (Среднее время обслуживания) – 327.635, 324.638, 340.205, 323.258, 241.444;
- **AVAIL.** – 1, 1, 1, 1, 1;
- **OWNER** – 937, 926, 830, 888, 0;
- **PEND** – 0, 0, 0, 0, 0;
- **INTER** – 0, 0, 0, 0, 0;
- **RETRY** – 0, 0, 0, 0, 0;
- **DELAY** – 9, 9, 9, 9, 0.

Еще ниже указываются результаты моделирования очередей (**QUEUE**) соответственно под номерами 2, 3, 4, 5 и с именами **TELEJKA_Q**, **KORZINA_Q** и **BISTRO_Q**:

- **MAX** (Максимальное содержание);
- **CONT.** (Текущее содержание);
- **ENTRY** (Число входов);
- **ENTRY(0)** (Число нулевых входов);

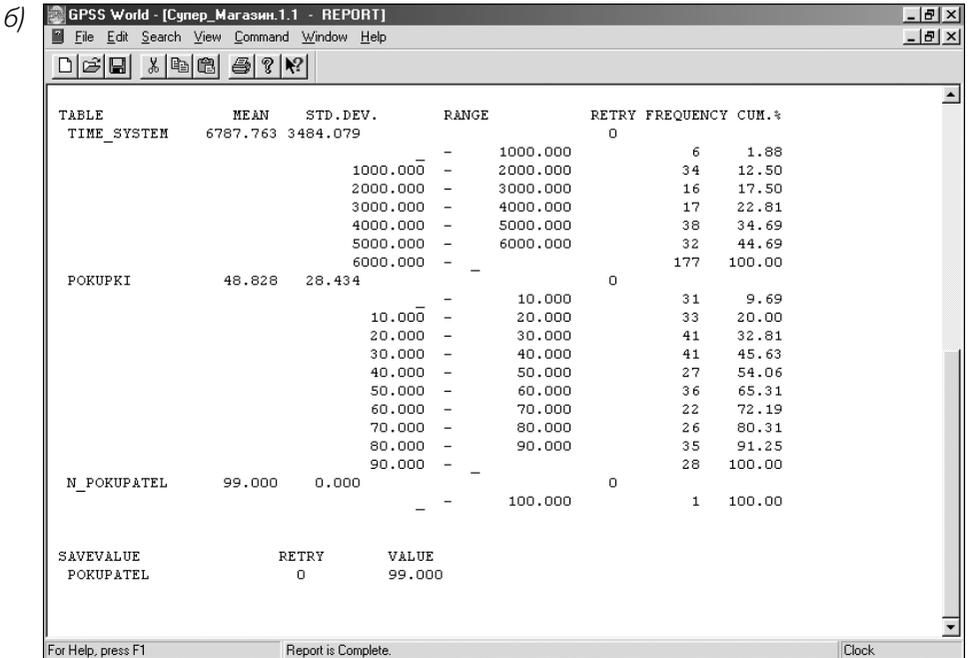
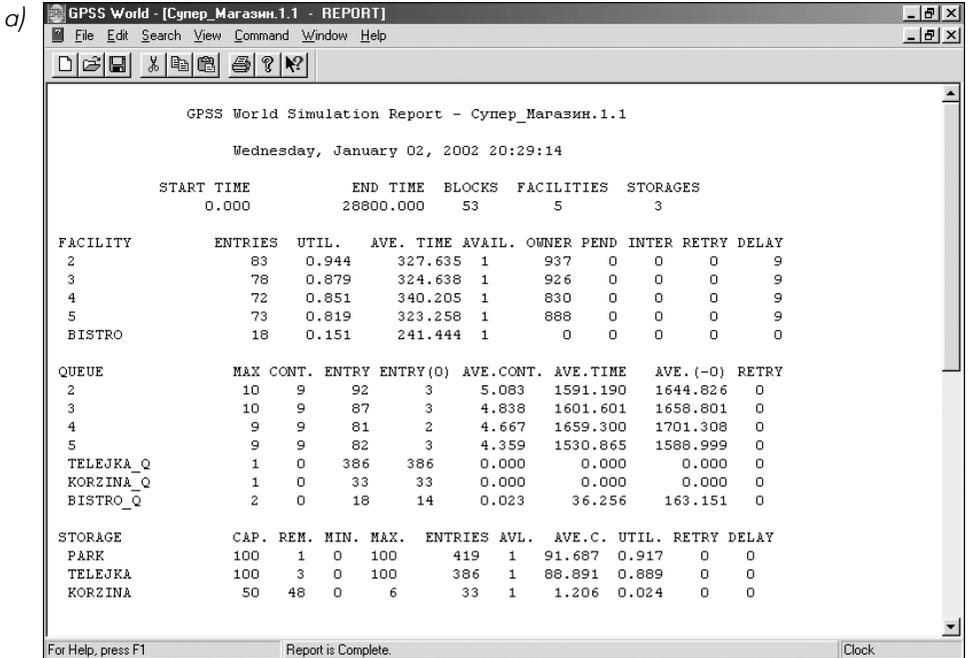


Рис. 3.37. Фрагменты результатов моделирования имитационной модели супермаркета

- **AVE.CONT.** (Среднее число входов);
- **AVE.TIME** (Среднее время);
- **AVE.(–0)**;
- **RETRY.**

Следующий фрагмент результатов моделирования содержит табличные данные для построения в дальнейшем соответствующих гистограмм **TIME_SYSTEM**, **POKUPKI**, **N_POKUPATEL**:

- **MEAN** (Средняя) – 6787.763, 48.828, 99.000;
- **STD.DEV.** (Среднее квадратическое отклонение) – 3484.079, 28.434, 0.000;
- **RANGE** (Область) – (0.000 – 6000.000), (0.000 – 90.000), (0.000 – 100.000);
- **RETRY** (Повтор) – 0, 0, 0;
- **FREQUENCY** (Частота);
- **CUM.%** (Суммарный процент).

Можно представить табличные данные в виде гистограммы. Для этого:

- щелкните по пункту **Window** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Table Window** во всплывающем меню. Появится диалоговое окно **Open Table Window**. В раскрывающемся списке **Table** щелкните по нужной гистограмме. Для нашей задачи это окно выглядит так, как показано на рис. 3.38;

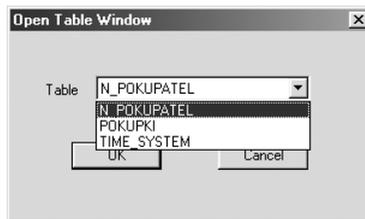


Рис. 3.38. Диалоговое окно выбора нужной гистограммы имитационной модели супермаркета

- щелкните по пункту **POKUPKI**;
- щелкните по кнопке **OK**. Появится соответствующая гистограмма. Для нашей задачи она выглядит так, как показано на рис. 3.39.

Затем все повторите сначала, но:

- щелкните по пункту **TIME_SYSTEM**;
- щелкните по кнопке **OK**. Появится соответствующая гистограмма. Для нашей задачи она выглядит так, как представлено на рис. 3.40.

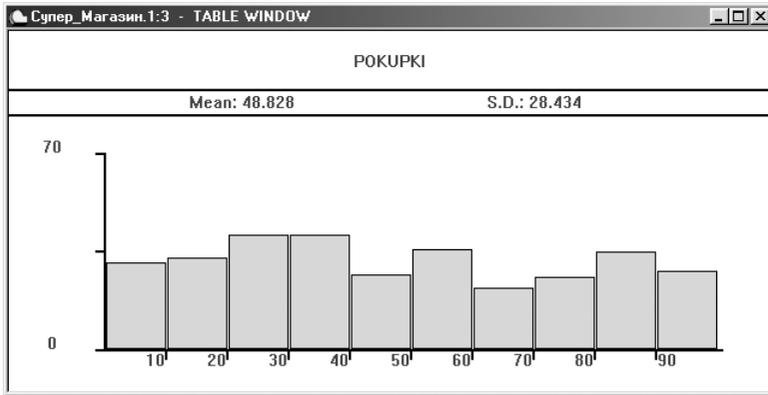


Рис. 3.39. Гистограмма покупок имитационной модели супермаркета

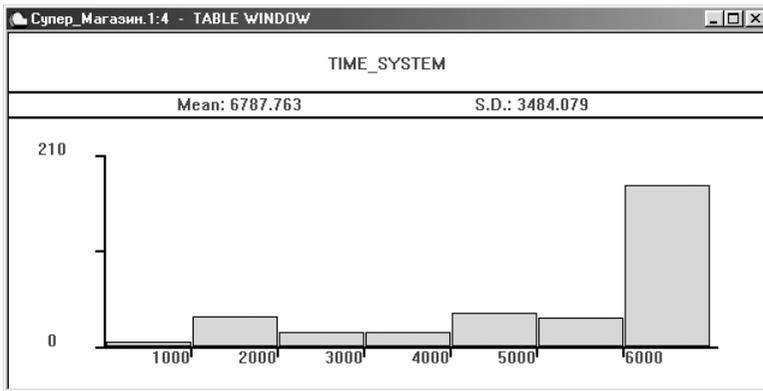


Рис. 3.40. Гистограмма системного времени имитационной модели супермаркета

Моделирование системы «Хищник–добыча»

Постановка задачи

Допустим, что на некоем острове неудержимо растет колония вредителей сельского хозяйства. Проблема уже настолько серьезна, что местные фермеры стали предпринимать серьезные меры для борьбы с вредителями. Они хотят везти на остров хищников, чтобы взять ситуацию под контроль. Требуется определить, сколько хищников необходимо завезти на остров, чтобы уничтожить всех вредителей. Это может быть система, например, «Фазаны–коларадский жук».

Выявление основных особенностей

Существует *модель Лотка–Вольтерра* (Lotka–Volterra) – модель «Хищник–добыча», – которая представляет собой систему обыкновенных дифференциальных уравнений. Для моделирования работы системы «Хищник–добыча» необходимо выделить две части процесса: моделируемые как непрерывный и как дискретный процесс.

Результаты моделирования представим в виде графика, в котором по оси абсцисс откладывается время моделирования системы, а по оси ординат – наличие хищников и добычи в каждый момент времени моделирования системы.

Но перед этим необходимо выбрать единицу измерения времени.

Создание имитационной модели процесса

Создание имитационной модели начнем с построения заголовка модели, который может быть представлен, например, в таком виде:

```
; GPSSW - Integrate.GPS
*****
*           Model Lotka-Volterra           *
*****
```

Модель будем формировать из трех секторов.

В первом секторе выполним часть процесса, моделируемую как непрерывный процесс. В этой части представим систему дифференциальных уравнений и начальные исходные данные.

Во втором секторе представим часть процесса, моделируемую как дискретный процесс.

В третьем секторе представим правую часть системы обыкновенных дифференциальных уравнений с помощью внешних процедур встроенного в систему GPSSW языка программирования PLUS.

Первый сектор модели включает систему обыкновенных дифференциальных уравнений, состоящую из двух уравнений первого порядка:

```
Hishnik  INTEGRATE  (Hishnik_prirost())
Dobicha  INTEGRATE  (Dobicha_prirost())
```

Кроме того, вводятся начальные условия и необходимые исходные данные:

```
Hishnik  EQU        40
Dobicha  EQU        600
K_       EQU        0.2000 ;Коеф. рождаемости хищника.
A_       EQU        0.0080 ;Коеф. смерти хищника.
B_       EQU        0.0002 ;Эффективность набега.
C_       EQU        0.0400 ;Коеф. рождаемости добычи.
```

Обратите внимание, что поскольку в записи уравнений используются однобуквенные или двухбуквенные обозначения, то применяются обозначения со знаком подчеркивания. Этот знак следует всегда вставлять, чтобы введенные обозначения случайно не совпали со стандартными числовыми атрибутами, встроенными в систему и несущими определенную информацию.

В скобках в правой части уравнений указан вызов процедур, представляющих собой правые части соответствующих уравнений системы, написанные на языке PLUS. Эти процедуры описываются в третьем секторе модели. Начальные условия и другая информация вводится с помощью оператора **EQU**. Второй сектор модели может быть представлен в таком виде:

```
GENERATE    2000
TERMINATE  1
```

Третий сектор модели может быть представлен в таком виде:

```
PROCEDURE Hishnik_prirost() BEGIN
  TEMPORARY Din_rojdeniy, Din_smerti, Din_prirosta;
  IF (Hishnik < 0) THEN Hishnik = 0;
  IF (Hishnik > 10e50) THEN Hishnik = 10e50 ;
  Din_rojdeniy = K_ # B_ # Hishnik # Dobicha;
  Din_smerti = A_ # Hishnik;
  Din_prirosta = Din_rojdeniy - Din_smerti;
  RETURN Din_prirosta;
END;
PROCEDURE Dobicha_prirost() BEGIN
  TEMPORARY Din_rojdeniy, Din_smerti, Din_prirosta;
  IF (Dobicha < 0) THEN Dobicha = 0;
  IF (Dobicha > 10e50) THEN Dobicha = 10e50 ;
  Din_rojdeniy = C_ # Dobicha;
  Din_smerti = B_ # Hishnik # Dobicha ;
  Din_prirosta = Din_rojdeniy - Din_smerti;
  RETURN Din_prirosta;
END;
```

Представление имитационной модели

Для представления имитационной модели выполните следующие действия:

- щелкните по пункту **File** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **New** выпадающего меню. Появится диалоговое окно **Новый документ**;
- выделите пункт **Model** и щелкните по кнопке **OK**. Появится окно модели, в котором введите данную программу. Это фрагмент модели будет выглядеть так, как показано на рис. 3.41.

```

GPSS World - [Integrate]
File Edit Search View Command Window Help
[Icons]
Hishnik INTEGRATE (Hishnik_prirost())
Dobicha INTEGRATE (Dobicha_prirost())
Hishnik EQU 40
Dobicha EQU 600
K_ EQU 0.2000 ;Коеф. рождаемости хищника.
A_ EQU 0.0080 ;Коеф. смерти хищника.
B_ EQU 0.0002 ;Эффективность набега.
C_ EQU 0.0400 ;Коеф. рождаемости добычи.
*****
GENERATE 2000
TERMINATE 1
*****
PROCEDURE Hishnik_prirost() BEGIN
TEMPORARY Din_rojdeniy, Din_smerti, Din_prirosta;
IF (Hishnik < 0) THEN Hishnik = 0;
IF (Hishnik > 10e50) THEN Hishnik = 10e50 ;
Din_rojdeniy = K_ # B_ # Hishnik # Dobicha;
Din_smerti = A_ # Hishnik;
Din_prirosta = Din_rojdeniy - Din_smerti;
RETURN Din_prirosta;
END;
PROCEDURE Dobicha_prirost() BEGIN
TEMPORARY Din_rojdeniy, Din_smerti, Din_prirosta;
IF (Dobicha < 0) THEN Dobicha = 0;
IF (Dobicha > 10e50) THEN Dobicha = 10e50 ;
Din_rojdeniy = C_ # Dobicha;
Din_smerti = B_ # Hishnik # Dobicha ;
Din_prirosta = Din_rojdeniy - Din_smerti;
RETURN Din_prirosta;
END;
For Help, press F1 Report is Complete. Clock

```

Рис. 3.41. Окно имитационной модели «Хищник–добыча»

Моделирование системы

После создания имитационную модель необходимо оттранслировать и запустить на выполнение. Для этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню.

Далее переходим к представлению искомого графика. Для этого:

- щелкните по пункту **Window** главного меню системы или нажмите комбинацию клавиш **Alt+W**. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** (Окно моделирования) выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Plot Window** (Окно графика) всплывающего меню. Появится диалоговое окно **Edit Plot Window** (Окно редактирования графика), в котором введите информацию, как показано на рис. 3.42.

Поскольку мы хотим в графическом окне расположить два графика, то заполнение диалогового окна **Edit Plot Window** необходимо выполнить в два этапа. На

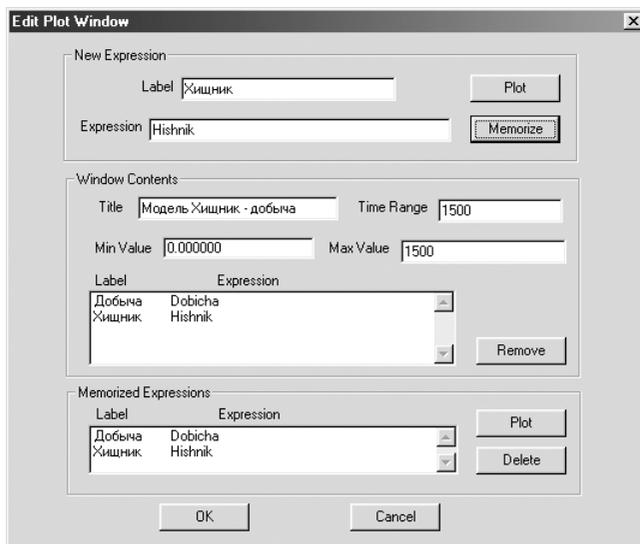


Рис. 3.42. Диалоговое окно *Edit Plot Window* для модели «Хищник–добыча»

первом этапе введите следующую информацию в текстовые поля, перемещаясь между ними с помощью клавиши **Tab**:

- в поле **Label** (Метка), например, слово «Добыча»;
- в поле **Expression** имя искомой переменной *Dobicha*;
- в поле **Title** (Заголовок), например, такой текст: «Модель Хищник–добыча»;
- в поле **Time Range** (Временная область) значение 1500;
- в поле **Min Value** (Минимальная величина) оставим приведенное значение без изменений;
- в поле **Max Value** (Максимальная величина) значение 1500.

Щелкните по кнопкам **Plot** (График) и **Memorise** (Запомнить). Информация, введенная в первых двух текстовых полях, появится в двух ниже расположенных многострочных списках.

Таким образом, мы указали всю необходимую информацию для вывода в графическом виде результатов моделирования, касающихся искомой переменной по имени *Dobicha*.

Вывод результатов в графическом виде

Теперь мы укажем все необходимые данные для вывода в графическом виде результатов моделирования, касающихся искомой переменной по имени *Hishnik*. Для этого введите следующую информацию в текстовые поля, перемещаясь между ними с помощью клавиши **Tab**:

- в верхнем поле **Label** вместо слова «Добыча», например, «Хищник»;
- в поле **Expression** вместо имени искомой переменной *Dobicha* имя другой искомой переменной *Hishnik*.

Затем:

- щелкните по кнопкам **Plot** и **Memorize**. Информация из первых двух текстовых полей добавится к ранее введенным данным в двух многострочных списках, расположенных ниже;
- щелкните по кнопке **OK**. Появится окно **PLOTS** (Графики), показанное на рис. 3.43, с заготовкой (шаблоном) графика.

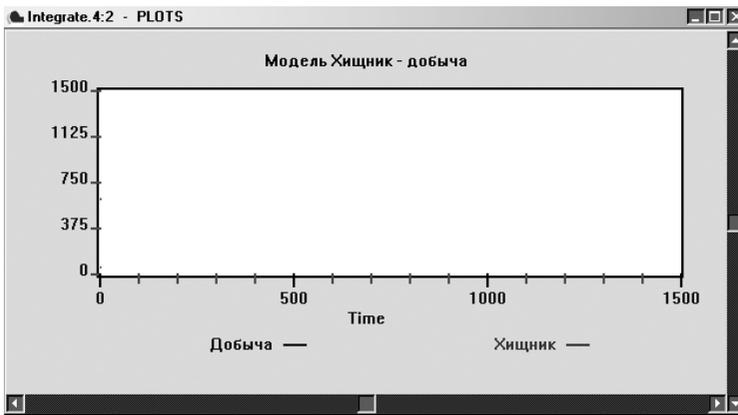


Рис. 3.43. Окно **PLOTS** с заготовкой графика

Если заготовка графика выглядит иначе, щелкните по горизонтальной полосе прокрутки графика слева от бегунка. Появится нужная заготовка. Теперь можно запустить систему на моделирование. Для этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **START** выпадающего меню. Появится диалоговое окно **Start Command**;
- щелкните по кнопке **OK**. Система дифференциальных уравнений, приведенная в модели, решается методом Рунге–Кутты пятого порядка, – эта функция встроена в систему. После решения в окне **PLOTS** появится искомый график, который может выглядеть так, как показано на рис. 3.44.

Анализ полученных результатов

Анализируя графические данные, можно отметить, что при заданных начальных значениях хищников и добычи соответственно 40 и 600 максимальное число хищников и добычи не превышает соответственно 600 и 1500.

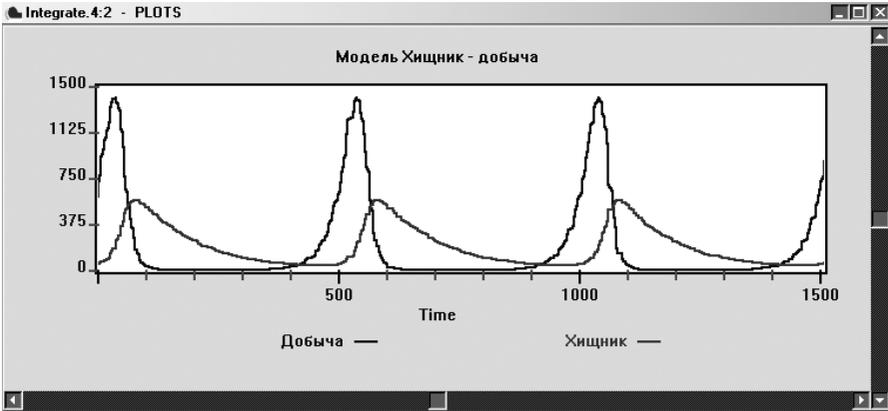


Рис. 3.44. Окно **PLOTS** с графиком для модели «Хищник–добыча»

Давайте теперь определим, что будет, если начальное число хищников взять равным 100, а значение добычи оставить прежним.

Чтобы увидеть результат решения:

- в окне **Model Window** введите значение хищников 100 вместо 40:

```
Hishnik EQU 100
```

- закройте окно **PLOTS**, щелкнув по кнопке с изображением крестика в верхней правой части окна;
- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **Retranslate** (Перетранслировать модель) выпадающего меню;
- щелкните по пункту **Window** главного меню системы или нажмите комбинацию клавиш **Alt+W**. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** (Окно моделирования) выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Plot Window** (Окно графика) всплывающего меню. Появится диалоговое окно **Edit Plot Window** (Окно редактирования графика). Теперь можно установить параметры для графика, которые были запомнены в прошлый раз с помощью кнопки **Memorize** (Запомнить);
- выберите в диалоговом окне **Edit Plot Window** в разделе **Memorized Expressions** (Запомненные выражения) строку с искомой переменной:

```
Добыча Dobicha
```

и щелкните по кнопке **Plot**;

- выберите в этом же разделе другую строку с искомой переменной:

```
Хищник Hishnik
```

и щелкните по кнопке **Plot**;

- введите в текстовом поле **Time Range** прежнее значение 1500;
- введите в текстовом поле **Max Value** (Максимальная величина) прежнее значение, равное 1500. Для перехода к другому текстовому полю можно использовать мышь;
- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **START** выпадающего меню. Появится диалоговое окно **Start Command**;
- щелкните по кнопке **OK**. Появится окно **PLOTS** с искомым графиком, который будет выглядеть так, как показано на рис. 3.45.

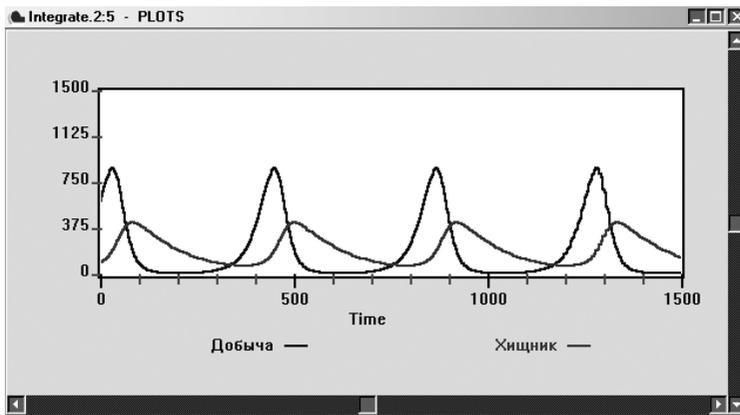


Рис. 3.45. График для модели «Хищник–добыча» после изменения некоторых параметров

Таким образом, при заданных начальных значениях хищников и добычи соответственно 100 и 600 максимальное число добычи и хищников не превысит соответственно 800 и 400.

Можете поэкспериментировать и с другими значениями хищников.

Моделирование распространения эпидемии

Постановка задачи

Допустим, что в микрорайоне, население которого составляет 50000 человек, возникла эпидемия, и появилось 80 инфекционных больных. Предположим, что прирост больных за день пропорционален произведению числа здоровых, еще не

переболевших и не приобретших иммунитет людей, на число больных. Коэффициент пропорциональности K_i (Коэффициент распространения инфекции) включает разного рода профилактические меры, принимаемые во время эпидемии. Требуется определить, как развивается эпидемия и каково будет максимальное число больных.

Выявление основных особенностей

Известно, что процесс развития эпидемии может быть представлен системой двух дифференциальных уравнений:

$$\begin{aligned} X'_t &= X_t \times (K_i \times Y_t - 1); \\ Y'_t &= -K_i \times X_t \times Y_t, \end{aligned}$$

где: X_t – больные;
 Y_t – здоровые.

Обратите внимание, что поскольку в записи уравнений используются однобуквенные или двухбуквенные обозначения, то применяются обозначения со знаком подчеркивания. Этот знак всегда следует вставлять, чтобы введенные обозначения случайно не совпали со стандартными числовыми атрибутами, встроенными в систему и несущими определенную информацию. Для моделирования эпидемии необходимо выделить две части процесса: моделируемые как непрерывный и как дискретный процесс.

Результаты моделирования представим в виде графика, в котором по оси абсцисс откладывается время моделирования системы, а по оси ординат – число здоровых и больных людей в каждый момент времени моделирования системы.

Но перед этим необходимо выбрать единицу измерения времени. Такой единицей может быть день.

Создание имитационной модели процесса

Построение имитационной модели начнем с создания заголовка модели, который может быть представлен, например, в таком виде:

```
; GPSSW - Эпидемия.GPS
*****
*           Моделирование эпидемии           *
*****
```

Модель будем формировать из трех секторов.

В первом секторе представим часть процесса, моделируемую как непрерывный процесс. В этой части введем систему дифференциальных уравнений и начальные данные.

Во втором секторе представим часть процесса, моделируемую как дискретный процесс.

В третьем секторе представим правую часть системы обыкновенных дифференциальных уравнений с помощью внешних процедур встроенного в систему GPSSW языка программирования PLUS.

Первый сектор модели включает систему обыкновенных дифференциальных уравнений, состоящую из двух уравнений первого порядка. Кроме того, введем начальные условия и необходимые исходные данные. Система дифференциальных уравнений может быть представлена в таком виде:

```
X_ INTEGRATE (Bolnoi())
Y_ INTEGRATE (Zdorov())
```

В скобках в правой части уравнений указан вызов процедур, представляющих собой правые части соответствующих уравнений системы. Эти процедуры описываются в третьем секторе модели. Ниже указаны начальные условия и другая информация, которая вводится с помощью оператора **EQU**. Параметр K_i представляет собой коэффициент распространения инфекции, который зависит от множества факторов. В нашей задаче он принят равным 0,00003.

```
Ki_ EQU 0.0001
Y_ EQU 50000
X_ EQU 80
```

Параметр $Y_$ – это число здоровых людей в самом начале процесса моделирования.

Параметр $X_$ – это число больных людей в самом начале процесса моделирования.

Второй сектор моделирует временной интервал процесса моделирования. Он принят в нашей задаче равным 30 дням и может быть представлен в таком виде:

```
GENERATE 30
TERMINATE 1
```

Третий сектор модели определяет правые части дифференциальных уравнений системы с помощью процедур встроенного языка программирования для моделирования PLUS. Поскольку в процессе интегрирования искомые параметры могут выйти за допустимые границы, то вводятся соответствующие ограничения с помощью условного оператора. Эта часть модели может быть представлена так:

```
PROCEDURE Bolnoi() BEGIN ; Динамика роста числа больных.
    TEMPORARY A_ ;
    IF (X_ < 0) THEN X_ = 0 ;
    IF (X_ > 10e50) THEN X_ = 10e50 ;
    A_ = X_ # (Ki_ # Y_ - 1) ;
    RETURN A_ ;
END ;
PROCEDURE Zdorov() BEGIN ; Динамика уменьшения числа здоровых.
```

```

TEMPORARY B_;
IF (Y_ < 0) THEN Y_ = 0;
IF (Y_ > 10e50) THEN Y_ = 10e50;
B_=(0 - Ki_) # X_ # Y_;
RETURN B_;
END;

```

Представление имитационной модели

Для представления имитационной модели выполните следующие действия:

- щелкните по пункту **File** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **New** выпадающего меню. Появится диалоговое окно **Новый документ**;
- выделите пункт **Model** и щелкните по кнопке **ОК**. Появится окно модели, в котором введете данную программу. Этот фрагмент модели будет выглядеть так, как показано на рис. 3.46.

```

Эпидемия
; GPSSW - Эпидемия.GPS
*****
* Моделирование эпидемии *
*****
X_ INTEGRATE (Bolnoi()) ; Динамика роста числа больных.
Y_ INTEGRATE (Zdorov()) ; Динамика уменьшения числа здоровых.
Ki_ EQU 0.0001 ; Коэффициент распространения инфекции.
Y_ EQU 50000 ; Начальное число здоровых людей.
X_ EQU 80 ; Начальное число больных людей.
*****
GENERATE 30 ; Моделирование в течение 30 дней.
TERMINATE 1
*****
PROCEDURE Bolnoi() BEGIN
TEMPORARY A_;
IF (X_ < 0) THEN X_ = 0;
IF (X_ > 10e50) THEN X_ = 10e50;
A_ = X_ # (Ki_ # Y_ - 1);
RETURN A_;
END;
PROCEDURE Zdorov() BEGIN
TEMPORARY B_;
IF (Y_ < 0) THEN Y_ = 0;
IF (Y_ > 10e50) THEN Y_ = 10e50;
B_=(0 - Ki_) # X_ # Y_;
RETURN B_;
END;

```

Рис. 3.46. Окно имитационной модели распространения эпидемии

Вызвать окно для представления имитационной модели в системе GPSSW можно также, нажав комбинацию клавиш **Ctrl+Alt+S**.

Моделирование системы

После создания имитационную модель необходимо оттранслировать и запустить на выполнение. Для этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню.

Далее переходим к представлению результатов решения системы дифференциальных уравнений в графическом виде. Для этого:

- щелкните по пункту **Window** главного меню системы или нажмите комбинацию клавиш **Alt+W**. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Plot Window** всплывающего меню. Появится диалоговое окно **Edit Plot Window**. Введите в это окно информацию, как показано на рис. 3.47.

Поскольку мы хотим в графическом окне расположить два графика, то заполнение диалогового окна **Edit Plot Window** необходимо выполнить в два этапа. На

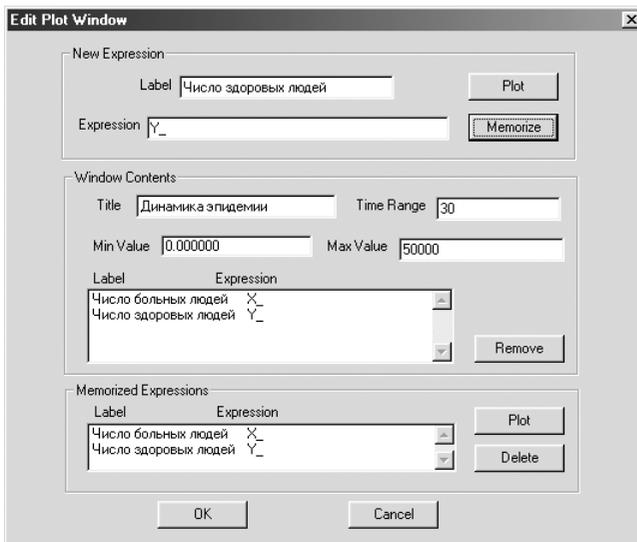


Рис. 3.47. Диалоговое окно **Edit Plot Window** для модели распространения эпидемии

первом этапе введите следующую информацию в текстовые поля, перемещаясь между ними с помощью клавиши **Tab**:

- в поле **Label**, например, фразу «Число больных людей»;
- в поле **Expression** имя искомой переменной X_1 ;
- в поле **Title**, например, такой текст: «Динамика эпидемии»;
- в поле **Time Range** значение 30 – это число дней моделирования;
- в поле **Min Value** оставим приведенное значение без изменений;
- в поле **Max Value** значение 50000 – это начальное количество здоровых людей.

Щелкните по кнопкам **Plot** и **Memorize**. Информация, введенная в первых двух текстовых полях, появится в двух многострочных списках, расположенных ниже.

Таким образом, мы указали всю необходимую информацию для вывода в графическом виде результатов моделирования, касающихся искомой переменной X_1 – числа больных людей.

Теперь мы дадим необходимую информацию для вывода в графическом виде результатов моделирования, касающихся искомой переменной Y_1 – числа здоровых людей. Для этого введите следующую информацию в текстовые поля, используя клавишу **Tab** для перемещения между ними:

- в поле **Label** вместо «Число больных людей», например, «Число здоровых людей»;
- в поле **Expression** вместо имени искомой переменной X_1 имя другой искомой переменной – Y_1 .

Далее:

- щелкните по кнопкам **Plot** и **Memorize**. Данные, указанные в первых двух текстовых полях, добавятся к ранее введенной информации в двух многострочных списках, расположенных ниже;
- щелкните по кнопке **OK**. Появится окно **PLOTS**, показанное на рис. 3.48, с заготовкой (шаблоном) графика.

Если заготовка графика выглядит иначе, щелкните по горизонтальной полосе прокрутки графика слева от бегунка. Появится нужная заготовка. Теперь можно запустить систему на моделирование. Для этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **START** выпадающего меню. Появится диалоговое окно **Start Command**;
- щелкните по кнопке **OK**. Начнется решение системы дифференциальных уравнений, приведенной в модели. Она будет решаться методом Рунге–Кутты пятого порядка. Затем появится окно **PLOTS**, представленное на рис. 3.49, с искомым графиком.

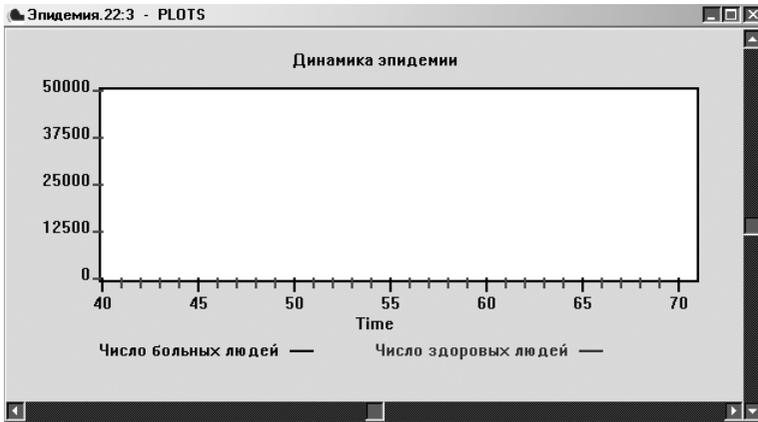


Рис. 3.48. Окно **PLOTS** с заготовкой графика для модели распространения эпидемии

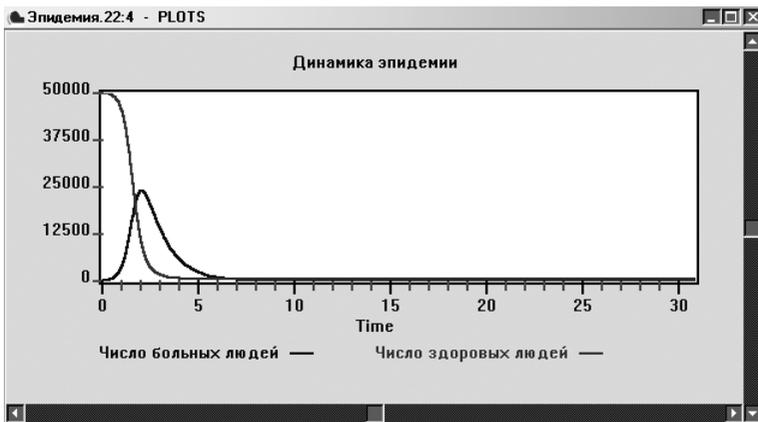


Рис. 3.49. Окно **PLOTS** с графиком для модели распространения эпидемии

Анализ полученных результатов

Анализируя графические данные, можно отметить, что при заданных начальных значениях – количестве здоровых людей 50000, а больных 801 – число больных достигнет максимума через 2,5 дня и составит около 25000 человек.

Давайте теперь посмотрим, что будет, если, проведя профилактические мероприятия, довести коэффициент распространения эпидемии до значения 0,00003. Чтобы увидеть результат этого решения:

- ведите новое значение коэффициента распространения эпидемии в первом секторе модели, заменив значение 0,0001 на 0,00003. Это будет выглядеть так:

```
Ki_ EQU 0.00003
```

- закройте окно **PLOTS**, щелкнув по кнопке с изображением крестика в верхней правой части окна;
- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **Retranslate** (Перетранслировать) выпадающего меню;
- щелкните по пункту **Window** главного меню системы или нажмите комбинацию клавиш **Alt+W**. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Plot Window** всплывающего меню. Появится диалоговое окно **Edit Plot Window**. Теперь можно установить параметры для графика, которые были запомнены в прошлый раз с помощью кнопки **Memorize**;
- выберите в диалоговом окне **Edit Plot Window** в разделе **Memorized Expressions** строку с искомой переменной:

Число больных людей $X_{_}$

и щелкните по кнопке **Plot**;

- выберите в этом же разделе другую строку с искомой переменной:

Число здоровых людей $Y_{_}$

и щелкните по кнопке **Plot**;

- введите в текстовом поле **Time Range** (Временная область) прежнее значение 30;
- введите в текстовом поле **Max Value** (Максимальная величина) прежнее значение, равное 50000. Для перехода к другому текстовому полю можно использовать мышь;
- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;

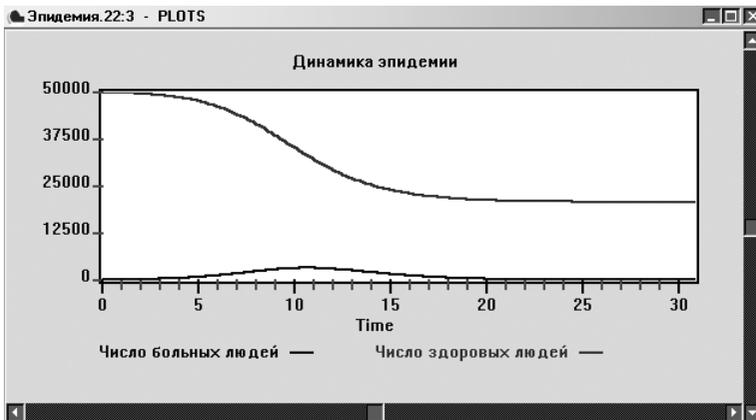


Рис. 3.50. График для модели распространения эпидемии после изменения начальных параметров

- щелкните по пункту **START** выпадающего меню. Появится диалоговое окно **Start Command**;
- щелкните по кнопке **OK**. Появится окно **PLOTS**, которое показано на рис. 3.50, с искомым графиком.

Таким образом, при заданных начальных значениях здоровых и больных людей соответственно 50000 и 80 число больных достигнет максимума на 11 день и будет составлять 4000. Это в шесть раз меньше, чем в первом случае.

Можете поэкспериментировать и с другими значениями коэффициента распространения эпидемии.

Моделирование производственных систем

Моделирование работы транспортного конвейера ...	158
Оценка надежности работы системы	168
Моделирование работы участка цеха	174
Моделирование работы автозаправочной станции ...	183
Моделирование работы инструментальной кладовой	197
Моделирование системы управления качеством	205
Моделирование системы управления запасами	213
Оценка эффективности работы системы с учетом отказов	229

В этой главе приведены примеры использования системы GPSSW для моделирования различных производственных процессов.

Моделирование работы транспортного конвейера

Постановка задачи

Допустим, к рабочим поступают на изготовление детали с транспортного конвейера. Интервал между поступлениями двух идущих одна за другой деталей равен 9 ± 1 мин. Время изготовления детали первым рабочим составляет 12 ± 1 мин, а вторым – 13 ± 2 мин. Если рабочий занят, он не берет деталь с конвейера, и она перемещается к другому рабочему. Требуется смоделировать работу первого и второго рабочих в течение смены. Необходимо определить коэффициент использования первого и второго рабочих (первого и второго каналов обслуживания) и число деталей, изготовленных каждым из них.

Выявление основных особенностей

Особенности моделирования данной системы заключаются в следующем:

- первая деталь (требование) поступает на изготовление (обслуживание) через 15 мин после начала смены;
- детали не накапливаются у первого рабочего. Если он занят, то детали поступают на обработку ко второму рабочему;
- время моделирования – рабочая смена – 480 мин.

Для реализации первой особенности используем оператор **GENERATE** (Генерировать):

```
GENERATE 5, , 15
```

В поле операнда А указан интервал времени, через который детали поступают в систему, – 5 мин, а в поле операнда С – время поступления первой детали на изготовление – 15 мин.

Для реализации второй особенности используем оператор **TRANSFER** (Передать):

```
TRANSFER BOTH, RABO
```

В поле операнда А указан режим работы оператора – BOTH (Оба), а в поле операнда С – имя метки оператора, к которому переходит деталь в случае занятости первого рабочего. Детали могут образовать очередь в связи с занятостью второго рабочего. При этом детали обрабатываются в порядке их поступления к рабочему. В поле операнда В имя оператора не указано (поле пусто). Это означает, что деталь может перейти к следующему по порядку оператору модели, если этот оператор не занят, в противном случае – к оператору с именем RABO.

Изобразим графически процесс функционирования системы «Транспортный конвейер–рабочий» (рис. 4.1).

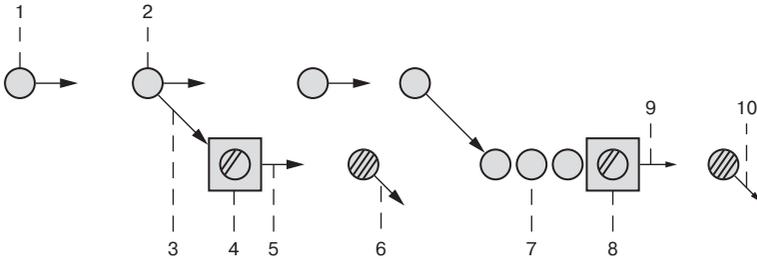


Рис. 4.1. Функционирование системы «Транспортный конвейер–рабочий»

На рис. 4.1 представлены основные события, которые возникают в процессе работы системы «Транспортный конвейер–рабочий»

Кратко охарактеризуем каждое событие в моделируемой системе:

1. Поступление деталей (требований) на обработку (**GENERATE**).
2. Передача детали первому или второму рабочему (**TRANSFER**).
3. Передача детали первому рабочему, если он свободен (**SEIZE**).
4. Время выполнения операции первым рабочим (**ADVANCE**).
5. Освобождение первого рабочего, то есть канала обслуживания (**RELEASE**).
6. Завершение выполнения операции – деталь покидает систему (**TERMINATE**).
7. Передача детали второму рабочему, если он свободен, а первый рабочий занят (**SEIZE**).
8. Выполнение операции вторым рабочим (**ADVANCE**).
9. Освобождение второго рабочего (**RELEASE**).
10. Завершение выполнения операции – деталь покидает систему (**TERMINATE**).

Создание имитационной модели процесса

Построение имитационной модели начнем с создания заголовка модели, который может быть представлен, например, в таком виде:

```
GPSSW                               Transp_konv.GPS
*****
*   Моделирование работы системы   *
*   "Транспортный конвейер–рабочий" *
*****
```

Далее разобьем всю модель на несколько секторов.

В первом секторе введем информацию, необходимую для функционирования системы. Этот сектор может выглядеть так:

```
INITIAL X1,9
INITIAL X2,1
INITIAL X3,12
INITIAL X4,1
INITIAL X5,13
INITIAL X6,2
```

В этом секторе с помощью оператора **INITIAL** (Инициализировать) задаются значения сохраняемых величин.

Во втором секторе моделируется процесс поступления деталей на обработку, передача деталей не занятому рабочему и обработка детали первым рабочим. Эта часть модели может быть сформирована следующим образом.

С помощью оператора **GENERATE** генерируется поток деталей, поступающих на обработку. Среднее время поступления деталей определяется сохраняемой величиной под номером 1, значение которой должно быть определено заранее. Отклонение от среднего времени определяется сохраняемой величиной под номером 2, значение которой также должно быть определено заранее. Это можно записать так:

```
GENERATE X1, X2
```

В поле операнда А указывается средний интервал времени между поступлениями требований в систему – X_1 . В задаче эта величина равна 9. Поле операнда В показывает величину отклонения от среднего интервала времени, равную ± 1 единице времени. Таким образом, интервал времени между поступлениями деталей в систему может составлять [8–10] единиц времени включительно. Определение конкретной величины интервала времени выполняется следующим образом. Встроенная подпрограмма случайных чисел (датчик случайных чисел) в интервале [0–1], используя равномерное распределение, определяет случайное число, например 0,25, затем система моделирования вычисляет искомую величину, используя формулу

$$X = A + (B - A) \times Y,$$

где А и В – соответственно левая и правая границы интервала поступления требований в систему.

Искомое случайное число в нашей задаче будет следующим:

$$X = 8 + (10 - 8) \times 0,25 = 8,5.$$

Передача деталей не занятому рабочему может быть осуществлена с использованием оператора **TRANSFER** в режиме BOTH (Оба). Если установлен режим передачи требований в один из двух операторов (BOTH), то требованию никогда не запрещается входить в оператор **TRANSFER**. Если оператор, указанный в поле В, занят, то требование переходит к оператору, указанному в поле С; если и этот оператор занят, то требование остается в операторе **TRANSFER** и в цепи текущих событий до освобождения занятого оператора в поле С. Это можно записать так:

```
TRANSFER BOTH, , RABO
```

Если первый рабочий занят, то деталь поступает на обработку ко второму рабочему – оператору с меткой RABO – при условии, что он не занят. Если и он занят, то деталь будет ожидать освобождения любого рабочего. Если первый рабочий освободился – а это определяется с помощью пары операторов **SEIZE** (Занять) и **RELEASE** (Освободить), – то деталь поступает к первому рабочему, где и производится ее обработка. Это можно записать так:

```

SEIZE      RAB1
ADVANCE    X3, X4
RELEASE    RAB1
TERMINATE

```

Операторы **SEIZE** и **RELEASE** с символьным именем RAB1 обеспечивают сбор статистической информации о канале обслуживания – первом рабочем. Оператор **ADVANCE** моделирует время выполнения операции первым рабочим (задерживает требование в канале обслуживания). Информация о времени обслуживания (задержки) требования указывается в полях операндов А и В. В поле операнда А представлено среднее время выполнения операции (обслуживания требования), а в поле операнда В – отклонение от среднего времени. Действительное время обслуживания требования определяется так же, как и время поступления деталей (требований) в операторе **GENERATE**. Так, если датчик случайных чисел выработал число 0,940, то время обслуживания требования будет следующим:

$$X = 11 + (13 - 11) \times 0,940 = 12,88.$$

В третьем секторе моделируется процесс обработки деталей вторым рабочим. Если первый рабочий занят, то деталь поступает на обработку ко второму рабочему при условии, что он не занят. Условие незанятости второго рабочего также определяется с помощью своей пары операторов **SEIZE** и **RELEASE**. Это можно записать так:

```

RABO SEIZE    RAB2
ADVANCE      X5, X6
RELEASE      RAB2
TERMINATE

```

Операторы **SEIZE** и **RELEASE** определяют соответственно занятость канала обслуживания (рабочего) и его освобождение. Эти два оператора взаимно дополняют друг друга. В поле операнда А операторов указывается имя канала обслуживания (символьное или числовое). Если канал обслуживания занят, то требование не может войти в него. Если канал обслуживания освободился, то при прохождении требования через оператор **RELEASE** дается сигнал об освобождении канала, и таким образом, следующее требование может поступить на обслуживание. Оператор **ADVANCE** моделирует время выполнения операции вторым рабочим.

В четвертом секторе моделируется время функционирования системы – 480 мин. Это выполняется с помощью оператора **GENERATE**.

В пятом секторе открывается текстовый файл, в который записываются нужные результаты моделирования. Этот сектор может выглядеть так:

```

Fin      OPEN    ("Transp_konveer_rez.txt") ,,Met1
          WRITE  (Catenate("Facility      =",FC$RAB1)) ,,Met
          WRITE  (Catenate("Util. Facility =",FR$RAB1/1000)) ,,Met
          WRITE  (Catenate("Facility      =",FC$RAB2)) ,,Met
          WRITE  (Catenate("Util. Facility =",FR$RAB2/1000)) ,,Met
CLOSE    Prob, ,,Met2

```

```

TERMINATE 1
Met TERMINATE 1
Met1 TERMINATE 1
Met2 TERMINATE 1
START 1,NP

```

Оператор **OPEN** (Открыть) открывает в нашем примере текстовый файл `Transp_konveer_rez.txt` для вывода конкретных результатов моделирования. Результаты моделирования записываются в открытый текстовый файл с помощью оператора **WRITE** (Записать). В нашем примере записывается число деталей, прошедших через рабочих, и коэффициент использования рабочих. Это выполняется с помощью соответствующих стандартных числовых атрибутов: `FC$RAB1`, `FR$RAB1`, `FC$RAB2`, `FR$RAB2`.

Затем открытый файл закрывается с помощью оператора **CLOSE** (Закреть).

Использование каждого оператора в третьем секторе в случае появления ошибок связано с переходом к оператору завершения – **TERMINATE** (Завершить).

Оператор **TERMINATE** удаляет из системы требования: указанное в поле A число требований покидают систему.

START представляет собой управляющий оператор, обеспечивающий выполнение процесса моделирования.

Окончательно программа моделирования этой системы будет выглядеть так, как показано на рис. 4.2 и 4.3.

Результаты моделирования системы представлены на рис. 4.4.

Первая строка:

```
Facility = 26
```

```

Transp_konv
* GPSSM      Transp_konv.GPS
*****
*   Моделирование работы системы   *
* "Транспортный конвейер – рабочий" *
*****
INITIAL      X1, 9          ; Среднее время поступления деталей на обработку.
INITIAL      X2, 1          ; Отклонение от среднего времени поступления.
INITIAL      X3, 12         ; Среднее время выполнения операции 1-м рабочим.
INITIAL      X4, 1          ; Откл. от среднего времени выполнения операции.
INITIAL      X5, 13         ; Среднее время выполнения операции 2-м рабочим.
INITIAL      X6, 2          ; Откл. от среднего времени выполнения операции.
*****
GENERATE      X1, X2         ; Интервал поступления деталей на обработку.
TRANSFER     BOTH, , RABO   ; Передача деталей 1-му или 2-му рабочему.
SEIZE        RAB1           ; С ожидание освобождения 1-го рабочего.
ADVANCE      X3, X4         ; Время выполнения операции 1-м рабочим.
RELEASE      RAB1           ; Освобождение 1-го рабочего.
TERMINATE    ; Завершение выполнения операции.
*****
RABO SEIZE    RAB2           ; С ожидание освобождения 2-го рабочего.
ADVANCE      X5, X6         ; Время выполнения операции 2-м рабочим.
RELEASE      RAB2           ; Освобождение 2-го рабочего.
TERMINATE    ; Завершение выполнения операции.
*****

```

Рис. 4.2. Начало программы моделирования системы «Транспортный конвейер–рабочий»

```

Transp_konveer
*****
      GENERATE      480      ; Время моделирования системы.
      TEST G       TGI,1,Fin
*****
Fin    OPEN       ("Transp_konveer_rez.txt"),,Met1
      WRITE      (Catenate("Facility      = ",FC$RAB1)),,Met
      WRITE      (Catenate("Util. Facility = ",FR$RAB1/1000)),,Met
      WRITE      (Catenate("Facility      = ",FC$RAB2)),,Met
      WRITE      (Catenate("Util. Facility = ",FR$RAB2/1000)),,Met
      CLOSE      Prob,,Met2
      TERMINATE  1
Met    TERMINATE  1
Met1   TERMINATE  1
Met2   TERMINATE  1
      START     1,NP
*****
    
```

Рис. 4.3. Продолжение программы моделирования системы «Транспортный конвейер-рабочий»

```

Transp_konveer_rez
Facility      = 26
Util. Facility = .6549566083333333
Facility      = 26
Util. Facility = .6749524583333336
    
```

Рис. 4.4. Результаты моделирования системы «Транспортный конвейер-рабочий»

показывает число деталей, поступивших к первому рабочему в течение моделируемого времени, то есть смены.

Вторая строка:

Util. Facility = .6549566083333333

показывает коэффициент использования первого рабочего в течение смены.

Третья строка:

Facility = 26

показывает число деталей, поступивших ко второму рабочему в течение моделируемого времени, то есть смены.

Четвертая строка:

Util. Facility = .6749524583333336

показывает коэффициент использования второго рабочего в течение смены.

Модификация моделируемой системы

Теперь немного модифицируем нашу систему. Введем третьего рабочего. Если первый рабочий занят – деталь подается ко второму; если и он занят – то к третьему рабочему. При этом третий рабочий должен обрабатывать все поступающие

к нему детали. Допустим, что детали поступают на обработку через интервал времени, равный $4 \pm 1,2$ мин. Время изготовления деталей первым, вторым и третьим рабочими соответственно равно 12 ± 1 , 13 ± 2 и 10 ± 2 мин. Требуется смоделировать работу всей линии в течение смены. Надо определить число деталей, обработанных каждым рабочим в течение смены, коэффициенты загрузки (использования) рабочих, максимальную и среднюю длину очередей деталей, ожидающих обработки у третьего рабочего, и среднее время пребывания детали в очереди к третьему рабочему.

Изобразим графически процесс функционирования модифицированной системы «Транспортный конвейер–рабочий» (рис. 4.5).

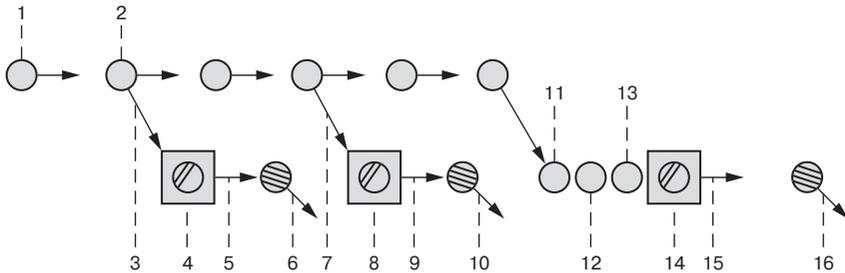


Рис. 4.5. Функционирование модифицированной системы «Транспортный конвейер–рабочий»

Охарактеризуем каждое событие в моделируемой системе:

1. Поступление деталей (требований) на обработку (**GENERATE**).
2. Передача деталей одному из рабочих (**TRANSFER**).
3. Передача детали первому рабочему, если он свободен (**SEIZE**).
4. Время выполнения операции первым рабочим (**ADVANCE**).
5. Освобождение первого рабочего, то есть канала обслуживания (**RELEASE**).
6. Завершение выполнения операции – деталь покидает систему (**TERMINATE**).
7. Передача детали второму рабочему, если он свободен (**SEIZE**).
8. Время выполнения операции вторым рабочим (**ADVANCE**).
9. Освобождение второго рабочего (**RELEASE**).
10. Завершение выполнения операции – деталь покидает систему (**TERMINATE**).
11. Вход в очередь на обслуживание к третьему рабочему (**QUEUE**).
12. Ожидание освобождения третьего рабочего (**SEIZE**).
13. Выход из очереди на обслуживание к третьему рабочему (**DEPART**).
14. Время выполнения операции третьим рабочим (**ADVANCE**).
15. Освобождение третьего рабочего (**RELEASE**).
16. Завершение выполнения операции – деталь покидает систему (**TERMINATE**).

Выявление основных особенностей

Особенности моделирования данной системы заключаются в следующем:

- у первого и второго рабочих очереди поступающих на обслуживание требований (деталей) отсутствуют, то есть рабочие действуют в режиме отказа. Если требование застает канал обслуживания занятым, то оно переходит к следующему каналу;
- третий рабочий обрабатывает все детали, которые не попали на обслуживание к первым двум рабочим. Если третий рабочий занят, то требование (деталь) становится в очередь и ожидает в ней до тех пор, пока не будет обслужено (обработано);
- используется режим передачи деталей (требований) в один из нескольких операторов – режим ALL (Все), а точнее в один из нескольких каналов обслуживания, который в момент передачи детали не занят. Этот режим является обобщением режима передачи требований в один из двух каналов обслуживания – BOTH.

В режиме передачи требований в один из нескольких операторов (ALL) требованию никогда не запрещается входить в оператор. Когда требование входит в оператор **TRANSFER**, оно последовательно проверяет занятость всех каналов обслуживания, и если все они заняты, то встает в очередь в последний канал обслуживания.

Номер оператора в поле операнда C должен быть больше номера оператора в поле операнда B на величину, кратную шагу, указанному в поле операнда D.

Поле операнда D оператора **TRANSFER** в режиме ALL определяет число операторов (шагов), которые требование должно пропустить перед тем, как войти в следующий оператор. Если оператор, в который собиралось войти требование, занят, то требование снова пропускает определенное число операторов и т.д.

Метки промежуточных операторов в программе могут не указываться. В нашей задаче имя RAB2 оператора **SEIZE**, относящегося ко второму каналу обслуживания, опущено.

Модель модифицированной системы

Модель модифицированной системы «Транспортный конвейер–рабочий» представлена на рис. 4.6 и 4.7.

Для просмотра результатов моделирования необходимо открыть текстовый файл `Transp_konv_2_gez` (имя файлу дает пользователь) данной системы. Для этого:

- щелкните по пункту **File** главного меню или нажмите комбинацию клавиш **Alt+F**. Появится выпадающее меню;
- щелкните по пункту **Open** (Открыть) главного меню или нажмите на клавишу **O**. Появится диалоговое окно **Открыть**.

```

Transp_konv_2
* GPSSM Transp_konv_2.GPS
*****
* Моделирование работы системы 2 *
* "Транспортный конвейер - рабочий" *
*****
GENERATE 4,1,2 ; Интервал поступления деталей на обработку.
TRANSFER ALL,RAB_1,RAB_3,4 ; Передача деталей 1-му, 2-му или 3-му рабочему.
RAB_1 SEIZE RAB1 ; Ожидание освобождения 1-го рабочего.
ADVANCE 12,1 ; Время выполнения операции 1-м рабочим.
RELEASE RAB1 ; Освобождение 1-го рабочего.
TERMINATE ; Завершение выполнения операции.
SEIZE RAB2 ; Ожидание освобождения 2-го рабочего.
ADVANCE 13,2 ; Время выполнения операции 2-м рабочим.
RELEASE RAB2 ; Освобождение 2-го рабочего.
TERMINATE ; Завершение выполнения операции.
RAB_3 QUEUE RAB3 ; Вход в очередь к 3-му рабочему.
SEIZE RAB3 ; Ожидание освобождения 3-го рабочего.
DEPART RAB3 ; Выход из очереди к 3-му рабочему.
ADVANCE 10,2 ; Время выполнения операции 3-м рабочим.
RELEASE RAB3 ; Освобождение 3-го рабочего.
TERMINATE ; Завершение выполнения операции.
*****

```

Рис. 4.6. Начало программы моделирования системы «Транспортный конвейер-рабочий»

```

Transp_konv_2
*****
GENERATE 480 ; Время моделирования системы.
TEST G TG1,1,Fin
*****
Fin OPEN ("Transp_konv_2_rez.txt"),Met1
WRITE (Catenate("Facility FC$RAB1 = ",FC$RAB1)),Met
WRITE (Catenate("Util. Facility FR$RAB1/1000 = ",FR$RAB1/1000)),Met
WRITE (Catenate("Facility FC$RAB2 = ",FC$RAB2)),Met
WRITE (Catenate("Util. Facility FR$RAB2/1000 = ",FR$RAB2/1000)),Met
WRITE (Catenate("Max QM$RAB3 = ",QM$RAB3)),Met
WRITE (Catenate("CONT. Q$RAB3 = ",Q$RAB3)),Met
WRITE (Catenate("ENTRY QC$RAB3 = ",QC$RAB3)),Met
WRITE (Catenate("AVE.TIME QT$RAB3 = ",QT$RAB3)),Met
CLOSE Prob,,Met2
TERMINATE 1
Met TERMINATE 1
Met1 TERMINATE 1
Met2 TERMINATE 1
START 1,NP

```

Рис. 4.7. Продолжение программы моделирования системы «Транспортный конвейер-рабочий»

Можно вызвать диалоговое окно **Открыть** нажатием комбинации клавиш **Ctrl+O** или щелчком по кнопке **Open** с изображением открытой папки. Далее в появившемся диалоговом окне:

- раскройте список типов файлов в текстовом поле **Тип файлов:**;
- щелкните по строке **Text File (*.txt)**;
- введите название файла в текстовом поле **Имя файла:**. Это может выглядеть так, как представлено на рис. 4.8;

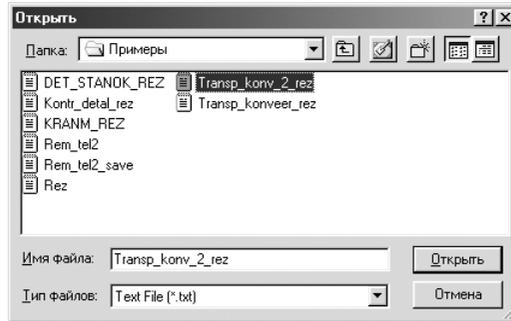


Рис. 4.8. Диалоговое окно **Открыть**

- щелкните по кнопке **Открыть**. Откроется текстовый файл с результатами моделирования, который для нашей задачи будет выглядеть так, как показано на рис. 4.9.

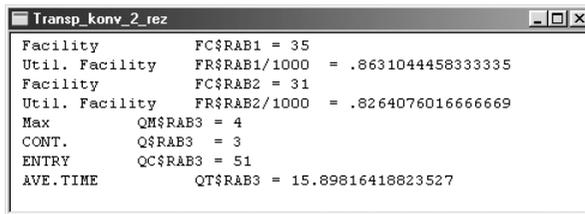


Рис. 4.9. Результаты моделирования системы «Транспортный конвейер–рабочий»

Первая строка:

Facility FC\$RAB1 = 35

показывает число деталей, поступивших к первому рабочему в течение моделируемого времени – смены.

Вторая строка:

Util. Facility FR\$RAB1/1000 = .8631044458333335

показывает коэффициент использования первого рабочего в течение смены.

Третья строка:

Facility FC\$RAB2 = 31

показывает число деталей, поступивших ко второму рабочему в течение моделируемого времени – смены.

Четвертая строка:

Util. Facility FR\$RAB2/1000 = .8264076016666669

показывает коэффициент использования второго рабочего в течение смены.

Пятая строка:

Max QM\$RAB3 = 4

показывает максимальную длину очереди – максимальное число деталей, ожидающих обслуживания, к третьему рабочему в течение смены.

Шестая строка:

CONT. Q\$RAB3 = 3

показывает среднюю длину очереди – среднее число деталей, ожидающих обслуживания, к третьему рабочему в течение смены.

Седьмая строка:

ENTRY QC\$RAB3 = 51

показывает число деталей, поступивших к третьему рабочему в течение смены.

Восьмая строка:

AVE.TIME QT\$RAB3 = 15.89816418823527

показывает среднее время нахождения детали в очереди к третьему рабочему в течение смены.

Оценка надежности работы системы

Постановка задачи

Рассмотрим систему, имеющую один объект (машину, прибор, станок, компьютер и т.д.), находящийся под действием пуассоновского потока отказов с интенсивностью λ . Отказавший объект немедленно начинает ремонтироваться (восстанавливаться). Распределение времени восстановления будем предполагать экспоненциальным с интенсивностью μ . Требуется оценить надежность работы такой системы, то есть вычислить:

- коэффициент использования (готовности) объекта;
- среднее время восстановления объекта.

Выявление основных особенностей

Рассматриваемые в системе потоки отказов наиболее полно отвечают реалиям жизни и характеризуются следующими особенностями:

- отказы объекта происходят по одному, то есть вероятность отказа двух и более элементов объекта одновременно очень мала, и ею можно пренебречь (поток требований ординарный);
- вероятность последующих отказов объекта в любой момент времени не зависит от предыдущих отказов – поток отказов без последствия;
- поток отказов стационарный.

Потоки отказов, в которых имеются все вышеперечисленные особенности, называются простейшими.

Интенсивность отказа объекта определяется как величина, обратная времени безотказной работы объекта, $- t_p$:

$$\lambda = 1/t_p.$$

Интенсивность восстановления объекта определяется как величина, обратная времени восстановления (ремонта) объекта, $- t_o$:

$$\mu = 1/t_o.$$

Рассмотрим сначала решение задачи аналитическим методом. Функционирование такой системы можно представить через все ее возможные состояния и интенсивность перехода системы из одного состояния в другое. Рассматриваемая система может находиться в одном из двух состояний: S_0 – объект работает, S_1 – объект не работает (находится в состоянии отказа и ремонтируется). Обозначим вероятности состояний S_0 и S_1 через $P_0(t)$ и $P_1(t)$.

Очевидно, что для любого момента времени имеет место равенство

$$P_0(t) + P_1(t) = 1.$$

Представим все возможные состояния функционирования системы в виде размеченного графа состояний (рис. 4.10). Каждый прямоугольник графа, количественно оцениваемый вероятностью состояний P_n , определяет одно из всех возможных состояний. Стрелки указывают, в какое состояние объект может перейти и с какой интенсивностью.

Первый прямоугольник с вероятностью $P_0(t)$ определяет состояние системы, при котором объект простаивает из-за отказа. Из этого положения система может перейти только в состояние $P_1(t)$. С интенсивностью μ система может перейти также из состояния $P_1(t)$ в состояние $P_0(t)$ после восстановления объекта.

Ограничимся рассмотрением установившегося режима работы системы, когда основные вероятностные характеристики системы постоянны во времени, например в течение года. Тогда интенсивности входных и выходных потоков для каждого состояния будут сбалансированы. Этот баланс выглядит так:

$$P_0 \times \lambda = P_1 \times \mu.$$

Обозначим величину λ / μ через ψ и назовем ее коэффициентом загрузки.

Таким образом, мы имеем два уравнения, описывающие функционирование нашей системы, которые могут быть представлены так:

$$\begin{aligned} P_0 \times \lambda - P_1 \times \mu &= 0; \\ P_0 + P_1 &= 1. \end{aligned}$$

Из первого уравнения можно найти значение P_1 :

$$P_1 = P_0 \times \lambda / \mu = P_0 \times \psi.$$

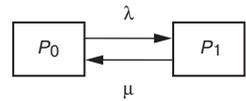


Рис. 4.10
Размеченный граф состояний объекта

Из второго уравнения найдем значение P_0 :

$$P_0 + P_0 \times \lambda / \mu = 1;$$

$$P_0 = 1 / (1 + \lambda / \mu) = \frac{\mu}{\mu + \lambda}.$$

Вероятность простоя объекта определяется так:

$$P_1 = 1 - P_0 = 1 - \frac{\mu}{\mu + \lambda} = \frac{\lambda}{\mu + \lambda}.$$

Величина P_0 есть не что иное, как коэффициент готовности системы, а P_1 – коэффициент простоя. Рассмотрим систему, имеющую следующие характеристики:

- $\lambda = 0,1$ 1/единица времени;
- $\mu = 0,05$ 1/единица времени.

По полученным выше формулам находим коэффициенты готовности системы и простоя:

$$K_r = P_0 = 0,333;$$

$$P_1 = 0,667.$$

Рассмотрим теперь решение этой же задачи имитационным методом.

Построение имитационной модели

Построение имитационной модели начнем с создания заголовка модели, который может быть представлен, например, в таком виде:

```
* GPSSW                Nadejn1_sist.GPS
*****
*      Оценка надежности работы      *
*      одноканальной системы          *
*****
```

Генерирование интервала времени, в течение которого должна быть проведена оценка надежности работы одноканальной системы, например в течение всего срока службы системы, выполняется с помощью оператора **GENERATE**. Он в нашей задаче может быть записан так:

```
GENERATE , , , 1
```

В поле операнда D определяется моделирование одной системы – 1.

Далее используется оператор **ASSIGN**, в котором определяется число отказов, происшедших в системе, например 10000. Это записывается так:

```
ASSIGN 1,10000
```

Далее моделируется интервал времени, через который наступает отказ. Для этого используется оператор **GENERATE** и встроенное экспоненциальное распределение.

Среднее время между наступлениями отказов составляет 10 единиц времени – третий аргумент встроенной функции экспоненциального распределения (Exponential...). Это можно представить так:

```
CiklADVANCE (Exponential(1,0,10))
```

Использование метки **Cikl** предполагает применение этого оператора в цикле. Далее используется оператор **SEIZE** (Занят), который определяет, работоспособен ли объект и выполняется ли сбор соответствующей статистической информации. Это может выглядеть так:

```
SEIZE KANAL_OBSL
```

В поле операнда **A** оператора **SEIZE** дается символьное или числовое имя объекта. В нашей задаче каналу обслуживания дано имя **KANAL_OBSL**. Желательно, чтобы присваиваемое имя отражало суть описываемого элемента системы.

Далее должно быть промоделировано время восстановления объекта. Для моделирования этого процесса используется оператор **ADVANCE** (Задержать), который в нашей задаче будет выглядеть так:

```
ADVANCE (Exponential(1,0,20))
```

В поле операнда **A** выполняется обращение к встроенному в систему экспоненциальному распределению с указанием среднего времени обслуживания машины – 20 единиц времени. После восстановления объекта должно быть послано сообщение об этом. Это делается с помощью оператора **RELEASE** (Освободить), который в нашей задаче записывается так:

```
RELEASE KANAL_OBSL
```

Следует особо подчеркнуть, что парные операторы **SEIZE** и **RELEASE**, обеспечивающие сбор статистической информации по каналу обслуживания, должны иметь одно и то же, но свое уникальное имя.

Для моделирования всех отказов, возникающих за период использования объекта, применим оператор **LOOP** (Цикл). Это действие может быть записано так:

```
LOOP 1,Cikl
```

Моделирование заканчивается операторами **TERMINATE** и **START**, которые записываются в программе в таком виде:

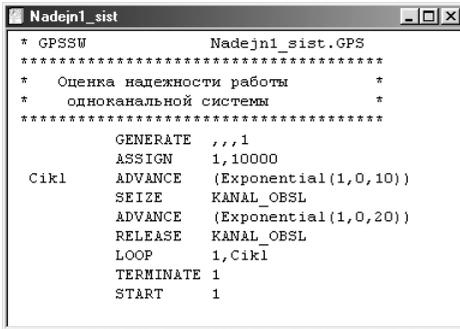
```
TERMINATE 1
START 1
```

Окончательно наша программа будет выглядеть так, как показано на рис. 4.11.

Моделирование системы

Для моделирования системы выполните следующие действия:

- щелкните по пункту **File** главного меню системы. Появится выпадающее меню;



```

* GPSSW      Nadejn1_sist.GPS
*****
* Оценка надежности работы      *
* одноканальной системы         *
*****
      GENERATE      ,, 1
      ASSIGN      1, 10000
Cikl1  ADVANCE      (Exponential(1,0,10))
      SEIZE      KANAL_OBSL
      ADVANCE      (Exponential(1,0,20))
      RELEASE     KANAL_OBSL
      LOOP      1, Cikl1
      TERMINATE 1
      START      1

```

Рис. 4.11. Окно имитационной модели оценки надежности системы

- щелкните по пункту **New** (Создать) выпадающего меню. Появится диалоговое окно **Новый документ**;
- выделите пункт **Model** и щелкните по кнопке **OK**. Появится окно модели, в котором введите данную программу.

Вызвать окно для представления имитационной модели в системе GPSSW можно также, нажав комбинацию клавиш **Ctrl+Alt+S**.

После создания имитационную модель необходимо оттранслировать и запустить на выполнение. Для этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню.

В имитационной модели имеется управляющая команда **START**, следовательно, исходная имитационная модель будет транслироваться, и если в ней нет ошибок, то начнется процесс моделирования системы.

Перед началом моделирования можно установить вывод тех параметров моделирования, которые нужны пользователю. Для этого:

- щелкните по пункту **Edit** (Правка) главного меню системы или нажмите комбинацию клавиш **Alt+E**. Появится выпадающее меню;
- щелкните по пункту **Settings** (Установки) выпадающего меню. Появится диалоговое окно **SETTINGS** для данной модели;
- щелкните по вкладке **Reports** (Отчеты), в которой можно установить нужные выходные данные. Для нашего примера это может выглядеть так, как показано на рис. 4.12.

Наличие галочки в окошках говорит о том, что эта информация будет выведена в окне результатов моделирования. В нашем примере будет выведена информация по каналам обслуживания.

Результаты моделирования представлены в окне **REPORT**, показанном на рис. 4.13.

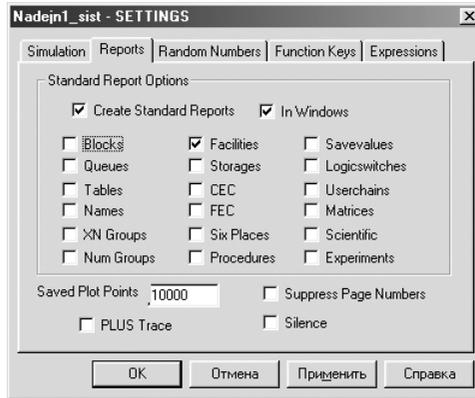


Рис. 4.12. Окно **SETTINGS** с установками для имитационной модели оценки надежности системы

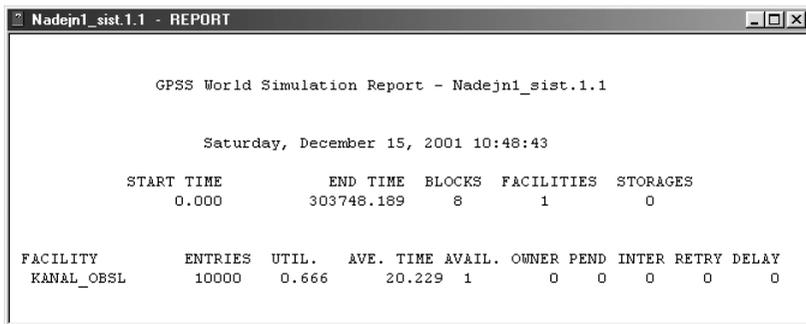


Рис. 4.13. Окно **REPORT** с фрагментом результатов моделирования оценки надежности системы

В верхней строке окна **REPORT** указывается:

- **START TIME** (Начальное время) – 0.000;
- **END TIME** (Время окончания) – 303748.189;
- **BLOCKS** (Число блоков) – 8;
- **FACILITIES** (Число каналов обслуживания) – 1;
- **STORAGES** (Число накопителей) – 0.

Ниже указываются результаты моделирования канала обслуживания под именем Kanal_obs1:

- **ENTRIES** (Число входов) – 10000;
- **UTIL.** (Коэффициент использования) – 0.666;
- **AVE. TIME** (Среднее время обслуживания) – 20.229;

- **AVAIL.** (Доступность) – 1;
- **OWNER** (Возможное число входов) – 0;
- **PEND** – 0;
- **INTER** – 0;
- **RETRY** (Повтор) – 0;
- **DELAY** (Отказ) – 0.

Сравнивая результаты решения задачи аналитическим и имитационным методами, видим, что они практически полностью совпадают. Коэффициент готовности системы при решении аналитическим методом равен 0,333, а имитационным: $1 - 0,666 = 0,334$.

Моделирование работы участка цеха

Постановка задачи

Промоделируем работу участка цеха, состоящего из трех видов оборудования, обслуживающих два потока изделий. Известны интервалы времени между поступлениями изделий каждого типа на обработку. Они соответственно равны 42 ± 5 и 20 ± 5 мин с равномерным законом распределения. Известно время изготовления изделия каждого потока на каждом виде оборудования. Так, время изготовления изделия первого потока на первом виде оборудования составляет 17 ± 2 мин, на втором – 32 ± 4 и на третьем – 22 ± 3 мин. Время изготовления изделия второго потока на первом виде оборудования составляет 19 ± 3 мин, на втором – 27 ± 5 и на третьем – 27 ± 5 мин. Продолжительность изготовления изделий на всех видах оборудования определяется равномерным законом распределения.

Перед нами стоят следующие задачи:

- промоделировать работу участка цеха в течение рабочего дня (8 ч) при двухсменном режиме;
- определить среднюю загрузку каждого вида оборудования, среднее время обработки изделий каждого типа, длину очередей перед каждым видом оборудования, размер склада, необходимый для данного потока изделий;
- предложить способы модификации участка цеха с целью повышения эффективности его работы.

Выявление основных особенностей

Для моделирования заданного производственного процесса необходимо сформировать два входных потока изделий и временной интервал моделирования всего производственного процесса. Но перед этим необходимо выбрать единицу измерения времени. Для данного производственного процесса можно взять в качестве единицы измерения минуту.

Для решения нашей задачи потребуется создать три сегмента модели для моделирования:

- поток изделий первого типа;
- поток изделий второго типа;
- время изготовления изделий в течение рабочего дня (8 ч) при двухсменном режиме.

Поскольку мы выбрали в качестве единицы измерения времени минуту, время моделирования процесса составит $8 \times 60 \times 2 = 960$ мин.

Создание имитационной модели

Построение имитационной модели начнем с создания заголовка модели, который может быть представлен, например, в таком виде:

```
; GPSSW                PR_UCHASTOK.GPS
*****
*   Моделирование производственного   *
*                   участка            *
*****
```

Программу работы одноканальной разомкнутой СМО можно представить в виде трех секторов.

В первом секторе будем моделировать первый поток изделий (требований), поступающих на обработку. Моделирование потока изделий выполним с помощью оператора **GENERATE** (Генерировать). В нашем примере он будет выглядеть так:

```
GENERATE 42,5
```

В поле операнда А указывается средний интервал времени между поступлением на обработку двух идущих одно за другим изделий. В поле операнда В дано отклонение времени поступления изделия на обработку от среднего. Такая запись времени поступления изделия в систему соответствует равномерному распределению поступления изделий на обработку в замкнутом интервале [42–5] мин.

Сбор статистической информации для очереди к первому виду оборудования можно обеспечить с помощью операторов **OUEUE** и **DEPART**. Оператор **OUEUE** может быть записан в таком виде:

```
QUEUE OCHER1
```

В поле операнда А дается символьное или числовое имя очереди. В нашем примере указано символьное имя очереди к первому виду оборудования – **ОCHER1**. Требование будет находиться в очереди до тех пор, пока не поступит сообщение об освобождении объекта. Для этого используется оператор **SEIZE** (Занять), который определяет занятость объекта, и при его освобождении очередное требование выходит из очереди и идет в канал на обслуживание. Это может выглядеть так:

```
SEIZE OBOR1
```

В поле операнда А дается символьное или числовое имя первого вида оборудования – первого объекта. В нашем примере используется символьное имя **OBOR1**.

Выход требования из очереди в объект фиксируется оператором **DEPART** (Выйти) с соответствующим именем очереди. В нашем примере он будет выглядеть так:

```
DEPART OCHER1
```

Далее должно быть промоделировано время обработки изделия первого потока на первом виде оборудования. Это время в нашем примере составляет 17 ± 2 мин. Для моделирования этого процесса используется оператор **ADVANCE** (Задержать), который в нашей задаче будет выглядеть так:

```
ADVANCE 17,2
```

После обработки на первом виде оборудования изделие первого потока переходит на обработку на второй вид оборудования. Но перед этим системе должно быть послано сообщение об освобождении первого вида оборудования – первого объекта. Это делается с помощью оператора **RELEASE** (Освободить), который в нашей задаче записывается так:

```
RELEASE OBOR1
```

Следует особо подчеркнуть, что парные операторы **QUEUE** и **DEPART** для каждой очереди должны иметь одно и то же, но свое уникальное имя или номер. Это же относится и к операторам **SEIZE** и **RELEASE**.

Далее изделие поступает на обработку на второй вид оборудования. Это может быть промоделировано так же, как и для первого вида оборудования, и выглядеть следующим образом:

```
QUEUE OCHER2
SEIZE OBOR2
DEPART OCHER2
ADVANCE 32,4
RELEASE OBOR2
```

После этого изделие поступает на обработку на третий вид оборудования. Это может быть промоделировано аналогично:

```
QUEUE OCHER3
SEIZE OBOR3
DEPART OCHER3
ADVANCE 22,3
RELEASE OBOR3
```

После обработки на трех видах оборудования изделие первого потока с помощью оператора **TERMINATE** покидает систему.

Далее представим второй сектор программы, в котором будем моделировать второй поток изделий (требований), поступающих на обработку. Обработка изделий второго потока производится аналогично и также начинается с оператора **GENERATE**. В нашем примере он будет выглядеть так:

```
GENERATE 20,5
```

В поле операнда А указывается средний интервал времени между поступлением на обработку двух идущих одно за другим изделий. В поле операнда В дано отклонение времени поступления изделия на обработку от среднего. Такая запись времени поступления изделия в систему соответствует равномерному распределению поступления изделий на обработку в замкнутом интервале $[20 \pm 5]$ мин.

Сбор статистической информации для очереди к первому виду оборудования обеспечим с помощью операторов **QUEUE** и **DEPART**. Оператор **QUEUE** записывается в таком виде:

```
QUEUE OCHER1
```

Изделие (требование) будет находиться в очереди до тех пор, пока не поступит сообщение об освобождении объекта – первого вида оборудования. Для этого используется оператор **SEIZE**, который определяет занятость объекта, и при его освобождении очередное изделие выходит из очереди и идет в канал на обслуживание. Это может выглядеть так:

```
SEIZE OBOR1
```

В поле операнда А дается символьное имя первого вида оборудования – первого объекта. В нашем примере используется символьное имя **OBOR1**.

Выход требования из очереди в объект фиксируется оператором **DEPART** с соответствующим именем очереди. В нашем примере это будет выглядеть так:

```
DEPART OCHER1
```

Далее должно быть промоделировано время обработки изделия второго потока на первом виде оборудования. Это время в нашем примере составляет 19 ± 3 мин. Для моделирования этого процесса используется оператор **ADVANCE**, который в нашей задаче будет выглядеть так:

```
ADVANCE 19,3
```

После обработки на первом виде оборудования изделие второго потока переходит на обработку на второй вид оборудования. Но перед этим системе должно быть послано сообщение об освобождении первого вида оборудования – первого объекта. Это делается с помощью оператора **RELEASE**, который в нашей задаче записывается так:

```
RELEASE OBOR1
```

Далее изделие поступает на обработку на второй вид оборудования. Это может быть промоделировано аналогично:

```
QUEUE OCHER2
SEIZE OBOR2
DEPART OCHER2
ADVANCE 27,5
RELEASE OBOR2
```

После этого изделие поступает на обработку на третий вид оборудования, что может быть промоделировано аналогично:

```

QUEUE    OCHER3
SEIZE    OBOR3
DEPART   OCHER3
ADVANCE  27,5
RELEASE  OBOR3

```

После обработки на трех видах оборудования изделие второго потока с помощью оператора **TERMINATE** покидает систему.

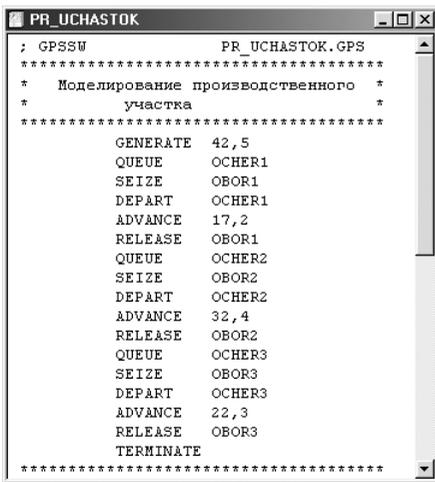
В третьем секторе моделируется время работы системы. Это можно представить с помощью трех операторов:

```

GENERATE 960
TERMINATE 1
START    1

```

Окончательно программа будет выглядеть так, как показано на рис. 4.14 и 4.15.

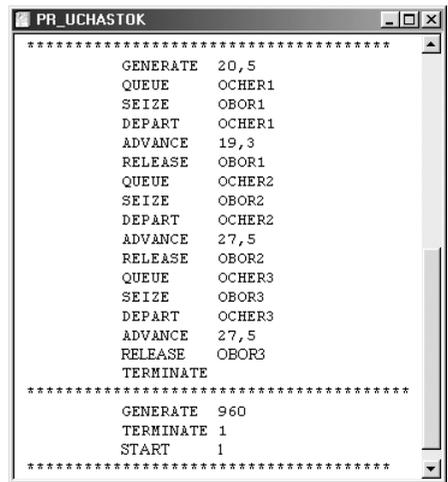


```

PR_UCHASTOK
; GPSSW PR_UCHASTOK.GPS
*****
* Моделирование производственного *
* участка *
*****
GENERATE 42,5
QUEUE OCHER1
SEIZE OBOR1
DEPART OCHER1
ADVANCE 17,2
RELEASE OBOR1
QUEUE OCHER2
SEIZE OBOR2
DEPART OCHER2
ADVANCE 32,4
RELEASE OBOR2
QUEUE OCHER3
SEIZE OBOR3
DEPART OCHER3
ADVANCE 22,3
RELEASE OBOR3
TERMINATE
*****

```

Рис. 4.14. Первая часть программы моделирования производственного участка



```

PR_UCHASTOK
*****
GENERATE 20,5
QUEUE OCHER1
SEIZE OBOR1
DEPART OCHER1
ADVANCE 19,3
RELEASE OBOR1
QUEUE OCHER2
SEIZE OBOR2
DEPART OCHER2
ADVANCE 27,5
RELEASE OBOR2
QUEUE OCHER3
SEIZE OBOR3
DEPART OCHER3
ADVANCE 27,5
RELEASE OBOR3
TERMINATE
*****
GENERATE 960
TERMINATE 1
START 1
*****

```

Рис. 4.15. Вторая часть программы моделирования производственного участка

Проведение имитационного моделирования

Перед началом моделирования откройте вкладку **Reports** (Отчеты), связанную с получением нужных результатов моделирования. Для этого:

- щелкните по пункту **Edit** (Правка) главного меню. Появится выпадающее меню;

- щелкните по пункту **Settings** (Настройки) выпадающего меню. Появится диалоговое окно **SETTINGS**;
- щелкните по вкладке **Reports**, чтобы открыть ее (рис. 4.16).

Установите флажки так, как показано на рис. 4.16, и щелкните по кнопке **OK**. После этого можно переходить к транслированию и выполнению программы.



Рис. 4.16. Диалоговое окно настройки **SETTINGS** с открытой вкладкой **Reports** для имитационной модели производственного участка

Перед началом моделирования, а точнее после трансляции модели, можно графически представить некоторые параметры функционирования системы. Если в модели есть команда управления **START**, она должна быть заблокирована, то есть переведена в комментарии. Для этого в позиции 1 поставьте звездочку. Процесс построения графика включает следующие этапы:

- щелкните по пункту **Window** главного меню системы или нажмите комбинацию клавиш **Alt+W**. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** (Окно моделирования) выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Plot Window** (Окно графика) всплывающего меню. Появится диалоговое окно **Edit Plot Window** (Окно редактирования графика), которое необходимо соответствующим образом заполнить.

Допустим, мы хотим графически представить на всем периоде моделирования, как меняется длина очереди изделий, поступивших на обработку к первому виду оборудования. Для нашей задачи окно **Edit Plot Window** может быть заполнено так, как показано на рис. 4.17.

После заполнения диалогового окна **Edit Plot Window** щелкните по кнопкам **Plot** (График), **Memorize** (Запомнить), а затем – по кнопке **OK**. Появится заготовка графика (рис. 4.18).

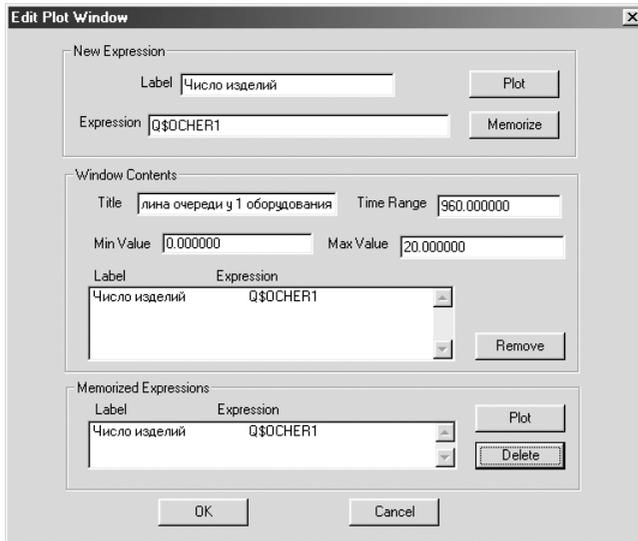


Рис. 4.17. Диалоговое окно **Edit Plot Window** для имитационной модели производственного участка

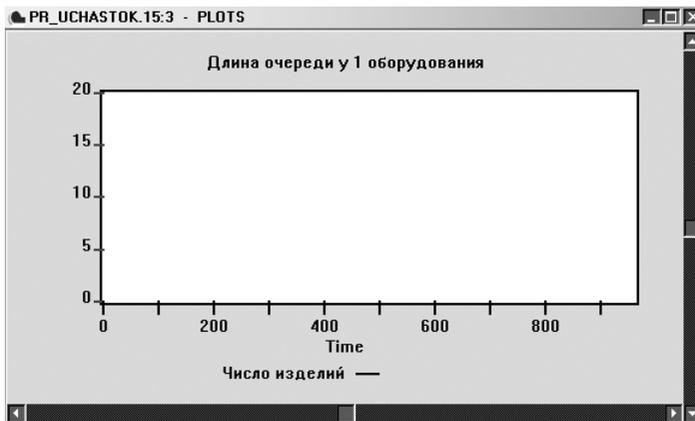


Рис. 4.18. Заготовка графика для представления длины очереди к первому виду оборудования

После этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **START**. Появится диалоговое окно **Start Command**;
- введите в диалоговом окне **Start Command** число 1 для начала моделирования и щелкните по кнопке **OK**. Появится окно **REPORT** с результатами моделирования. На заднем плане будет размещаться график;

- щелкните по графику, расположенному на заднем плане, – он выйдет на первый план;
 - используя горизонтальную и вертикальную полосы прокрутки, вы можете просмотреть построенный график.
- Фрагмент графика для нашего примера представлен на рис. 4.19.

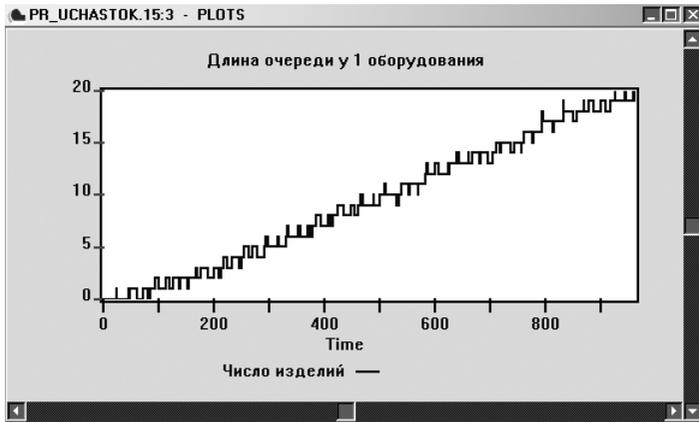


Рис. 4.19. Фрагмент изображения длины очереди к первому виду оборудования

При выводе графика на передний план окно **REPORT** с результатами моделирования переместится на задний план. Для просмотра окна **REPORT** щелкните по нему. Оно вновь перейдет на передний план и будет выглядеть так, как показано на рис. 4.20.

В верхней строке указывается:

- **START TIME** (Начальное время) – 0.000;
- **END TIME** (Время окончания) – 960.000;

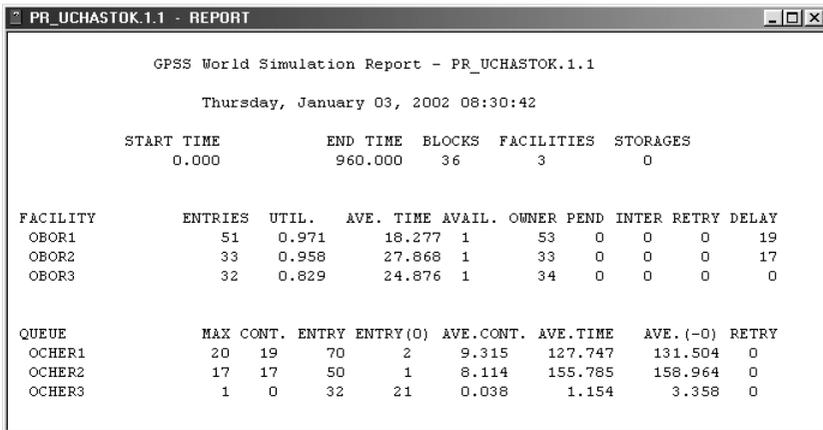


Рис. 4.20. Окно **REPORT** с результатами моделирования системы

- **BLOCKS** (Число блоков) – 36;
- **FACILITIES** (Число каналов обслуживания) – 3;
- **STORAGES** (Число накопителей) – 0.

Ниже указываются результаты моделирования для всех трех каналов обслуживания (FACILITY) под присвоенными нами именами соответственно OBOR1, OBOR2 и OBOR3:

- **ENTRIES** (Число входов) – 51, 33, 32;
- **UTIL.** (Коэффициент использования) – 0.971, 0.958, 0.829;
- **AVE. TIME** (Среднее время обслуживания) – 18.277, 27.868, 24.876;
- **AVAIL.** (Доступность) – 1, 1, 1;
- **OWNER** (Возможное число входов) – 0, 0, 0;
- **PEND** – 0, 0, 0;
- **INTER** – 0, 0, 0;
- **RETRY** (Повтор) – 0, 0, 0;
- **DELAY** (Отказ) – 19, 17, 0.

Еще ниже указываются результаты моделирования для всех трех очередей (QUEUE) под присвоенными нами именами соответственно OCHER1, OCHER2 и OCHER3:

- **MAX** (Максимальное содержание) – 20, 17, 1;
- **CONT.** (Текущее содержание) – 19, 17, 0;
- **ENTRY** (Число входов) – 70, 50, 32;
- **ENTRY(0)** (Число нулевых входов) – 2, 1, 21;
- **AVE.CONT.** (Среднее число входов) – 9.315, 8.114, 0.038;
- **AVE.TIME** – 127.747, 155.785, 1.154;
- **AVE.(–0)** – 131.504, 158.964, 3.358;
- **RETRY** – 0, 0, 0.

Анализ результатов имитационного моделирования

По результатам моделирования данного варианта можно сделать несколько выводов. Во-первых, коэффициенты загрузки всех видов оборудования достаточно высоки и составляют 0,971; 0,958 и 0,829 соответственно для первого, второго и третьего вида оборудования.

Во-вторых, максимальная очередь образуется перед первым и вторым видами оборудования. Ее образуют соответственно 20 и 17 изделий. Следовательно, необходимо перед этими видами оборудования предусмотреть места для размещения достаточного количества изделий.

Моделирование работы автозаправочной станции

Постановка задачи

Допустим, нам надо промоделировать работу автомобильной заправочной станции (АЗС), которая имеет две заправочные колонки. Известны следующие параметры работы АЗС:

- поток автомобилей, поступающих на заправку, подчиняется экспоненциальному распределению вероятностей с параметрами $\lambda = 0$ и $\beta = 6,5$;
- время заправки на первой колонке составляет $10 \pm 2,5$ мин, а на второй – 13 ± 4 мин;
- автомобиль подъезжает к колонке, которая не занята обслуживанием другого автомобиля.

Требуется промоделировать работу АЗС в течение рабочей смены – 8 ч – и определить параметры функционирования АЗС:

- коэффициент загрузки каждой колонки;
- среднее время обслуживания в каждой колонке;
- максимальное, среднее и текущее число автомобилей в очереди к каждой колонке;
- среднее время нахождения автомобиля в каждой очереди и др.

Выявление основных особенностей

Для моделирования работы АЗС необходимо сформировать входной поток автомобилей, подъезжающих к заправке с интервалом времени, соответствующим экспоненциальному закону. Интервал времени рассчитывается по следующей формуле:

$$f(x) = \frac{1}{\beta} e^{-\frac{x-\alpha}{\beta}}, \beta \geq 0.$$

В противном случае его значение равно 0.

В качестве единицы измерения времени примем минуту.

Построение имитационной модели процесса

Построение имитационной модели начнем с создания заголовка модели, который может быть представлен, например, в таком виде:

```
; GPSSW AZS.GPS
*****
*      Моделирование автомобильной      *
*      заправочной станции (АЗС)        *
*****
```

Моделирование потока машин, поступающих на заправку, будем выполнять с помощью оператора **GENERATE** (Генерировать). Нам необходимо сформировать поток машин, поступающих на обслуживание на АЗС, который подчиняется экспоненциальному распределению вероятностей. Это можно сделать двумя способами, используя:

- эмпирические распределения, которые можно создать с помощью команды **FUNCTION** (Функция) языка GPSS, используя дискретные (тип D) или непрерывные (тип C) случайные функции;
- встроенные распределения вероятностей.

В системе GPSSW в библиотеку процедур включено более 20 встроенных распределений вероятности. Эти распределения имеют широкий диапазон использования.

Каждый вызов процедуры распределения вероятности требует, чтобы вы определили параметр потока, номер генератора случайных чисел.

В нашем примере оператор **GENERATE** запишется в таком виде:

```
GENERATE (Exponential(1,0,6.5))
```

В поле операнда A указывается обращение к библиотечной процедуре – экспоненциальному распределению вероятности.

Автомобиль, поступающий на заправку, сначала встает в очередь, если она есть. Это можно промоделировать оператором **QUEUE** (Очередь), который только в совокупности с оператором **DEPART** (Выйти) собирает статистическую информацию о работе моделируемой очереди.

В нашем примере оператор **QUEUE** будет выглядеть так:

```
QUEUE Zapravka
```

В поле операнда A дается символьное или числовое имя очереди. Таких очередей в системе может быть очень много. В нашей задаче дадим очереди имя Zapravka (Заправка). Желательно, чтобы присваиваемое имя отражало суть описываемого элемента системы.

Поскольку на АЗС есть две колонки для заправки, то автомобиль подъезжает к первой колонке, если она свободна, и ко второй, если первая занята. Если обе колонки заняты, то автомобиль ожидает их освобождения в общей очереди. Направление автомобилей на ту или иную колонку может быть обеспечено оператором **TRANSFER** (Передать). В нашем примере он запишется так:

```
TRANSFER Both,Kol_1,Kol_2
```

Слово Both (Оба) означает возможность передачи требования (машины) по двум направлениям: к оператору с меткой Kol_1, если первая колонка (объект) не занята, и к оператору с меткой Kol_2, если занята.

Следуя логике, автомобиль может выйти из очереди только тогда, когда освободится первая или вторая колонка (объект). Для моделирования используются два сегмента. Первый из них связан с работой первой колонки, а второй – с работой второй. Оба сегмента начинаются с оператора **SEIZE** (Занять), и каждый из них имеет свою метку, соответственно Kol_1 и Kol_2, которые определяют занятость первой или второй колонки (канала обслуживания). При освобождении одной из них находящееся впереди требование выходит из очереди и идет в канал на обслуживание.

Начало первого сегмента записывается так:

```
Kol_1 SEIZE Kolonka_1
```

В поле операнда A дается символьное или числовое имя объекта. Таких каналов обслуживания в системе может быть очень много. В нашей задаче каналу дано имя Kolonka_1 (Колонка_1). Желательно, чтобы присваиваемое имя отражало суть описываемого элемента системы.

Выход автомобиля из очереди фиксируется оператором **DEPART** с соответствующим названием очереди. В нашем примере это будет выглядеть так:

```
DEPART Zapravka
```

Далее должно быть промоделировано время заправки автомобиля. Оно в нашем примере составляет $10 \pm 2,5$ мин. Для моделирования этого процесса используется оператор **ADVANCE** (Задержать), который в нашей задаче будет выглядеть так:

```
ADVANCE 10,2.5
```

Затем необходимо сообщить об освобождении колонки под номером 1. Это можно сделать с помощью оператора **RELEASE** (Освободить). Он в нашей задаче записывается так:

```
RELEASE Kolonka_1
```

Далее используется оператор **TRANSFER** (Передать) в режиме безусловной передачи требования к оператору с указанной меткой в поле операнда B. В нашем примере это будет выглядеть так:

```
TRANSFER ,Next
```

Начало второго сегмента запишется так:

```
Kol_2 SEIZE Kolonka_2
```

В поле операнда A дается символическое имя объекта – Kolonka_2.

Выход автомобиля из очереди фиксируется оператором **DEPART** с соответствующим названием очереди. В нашем примере это будет выглядеть так:

```
DEPART Zapravka
```

Далее должно быть промоделировано время заправки автомобиля. Оно в нашем примере составляет 13 ± 4 мин. Для моделирования этого процесса используется оператор **ADVANCE**, который в нашей задаче будет выглядеть так:

```
ADVANCE 13,4
```

После заправки автомобиль покидает колонку. Это можно зафиксировать оператором **DEPART**, который в нашей задаче запишется так:

```
DEPART Total_time
```

Далее необходимо сообщить об освобождении колонки под номером 2. Это можно сделать с помощью оператора **RELEASE**. Он в нашей задаче записывается так:

```
RELEASE Kolonka_2
```

Затем используем оператор **SAVEVALUE** (Сохраняемая величина) с меткой Next (Следующий) для сохранения среднего времени нахождения автомобиля в очереди под именем Zapravka в сохраняемой величине под именем Ave_Queue. Это будет выглядеть так:

```
Next SAVEVALUE Ave_Queue, QT$Zappravka
```

Далее автомобиль покидает АЗС. Для этого используется оператор **TERMINATE** (Завершить):

```
TERMINATE
```

И наконец, мы должны создать сегмент, который будет моделировать работу АЗС в течение рабочей смены, равной 8 ч. Поскольку моделирование работы АЗС проводится в минутах, то время моделирования системы будет равно $8 \times 60 = 480$ мин. Этот сегмент будет выглядеть так:

```
GENERATE 480
TERMINATE 1
START 1
```

Следует особо подчеркнуть, что парные операторы **QUEUE** и **DEPART** для каждой очереди должны иметь одно и то же, но свое уникальное имя. Это же относится и к операторам **SEIZE** и **RELEASE**.

Завершающим оператором в нашей задаче является **START**, дающий команду начать моделирование:

```
START 1
```

Представление имитационной модели

Для представления имитационной модели выполните следующие действия:

- щелкните по пункту **File** главного меню системы. Появится выпадающее меню;

- щелкните по пункту **New** (Создать) выпадающего меню. Появится диалоговое окно **Новый документ**;
- выделите пункт **Model** и щелкните по кнопке **OK**. Появится окно модели, в котором введите данную программу. Оно показано на рис. 4.21.

```

; GPSSW                                AZS.GPSW|
*****
* Моделирование автомобильной          *
* заправочной станции (АЗС)           *
*****
      GENERATE (Exponential(1,0,6.5)) ; Поступление автомобилей на заправку.
      QUEUE   Zapravka                ; Вход автомобиля в очередь на заправку.
      TRANSFER Both,Kol_1,Kol_2      ; Переход автомобиля к свободной колонке.
Kol_1  SEIZE  Kolonka_1                ; Определение занятости 1 колонки.
      DEPART  Zapravka                ; Выход автомобиля из очереди на заправку.
      ADVANCE 10,2.5                  ; Заправка автомобиля на 1 колонке.
      RELEASE Kolonka_1              ; Освобождение 1 колонки.
      TRANSFER ,Next                 ; Переход к оператору с меткой Next.
Kol_2  SEIZE  Kolonka_2                ; Определение занятости 2 колонки.
      DEPART  Zapravka                ; Выход автомобиля из очереди на заправку.
      ADVANCE 13,4                   ; Заправка автомобиля на 2 колонке.
      RELEASE Kolonka_2              ; Освобождение 2 колонки.
Next   SAVEVALUE Ave_Queue,QT$Zapravka ; Сохранение времени пребывания в очереди.
      TERMINATE ; Выход автомобиля из системы.
      GENERATE 480                    ; Моделирование работы АЗС в течение смены.
      TERMINATE 1
      START 1
    
```

Рис. 4.21. Окно имитационной модели АЗС

Вызвать окно для представления имитационной модели в системе GPSSW можно также, нажав комбинацию клавиш **Ctrl+Alt+S**.

Моделирование системы

После создания имитационную модель необходимо оттранслировать и запустить на выполнение. Для этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню.

Так как в имитационной модели имеется управляющая команда **START**, то исходная имитационная модель будет транслироваться, и если в ней нет ошибок, начнется процесс моделирования системы.

Перед началом моделирования можно установить вывод тех параметров моделирования, которые нужны пользователю. Для этого:

- щелкните по пункту **Edit** (Правка) главного меню системы или нажмите комбинацию клавиш **Alt+E**. Появится выпадающее меню;
- щелкните по пункту **Settings** (Установки) выпадающего меню. Появится диалоговое окно **SETTINGS** для данной модели. В нашем примере появится

окно под именем **Azs – SETTINGS**, в котором можно установить нужные выходные данные. Для нашего примера оно может выглядеть так, как показано на рис. 4.22.

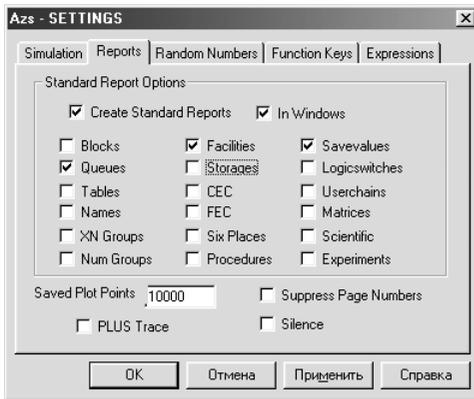


Рис. 4.22. Окно **SETTINGS** с установками для имитационной модели АЗС

Наличие галочек в окошках говорит о том, что эта информация будет выведена в окне результатов моделирования **REPORT** (Отчет). В нашем примере будет выведена информация по следующим объектам:

- **Queues** (Очереди);
- **Savevalues** (Сохраняемые величины);
- **Facilities** (Каналы обслуживания).

Перед началом моделирования, а точнее после трансляции модели, можно графически представить некоторые параметры функционирования системы. Если в модели есть команда управления **START**, она должна быть заблокирована, то есть переведена в комментарии. Для этого в позиции 1 поставьте звездочку. Процесс построения графика включает следующие этапы:

- щелкните по пункту **Window** главного меню системы или нажмите комбинацию клавиш **Alt+W**. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** (Окно моделирования) выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Plot Window** (Окно графика) всплывающего меню. Появится диалоговое окно **Edit Plot Window** (Окно редактирования графика), которое необходимо соответствующим образом заполнить.

Графическое представление результатов моделирования

Допустим, мы хотим графически представить на всем периоде моделирования, как меняется длина очереди машин на заправку. Для нашей задачи окно **Edit Plot Window** может быть заполнено так, как показано на рис. 4.23.

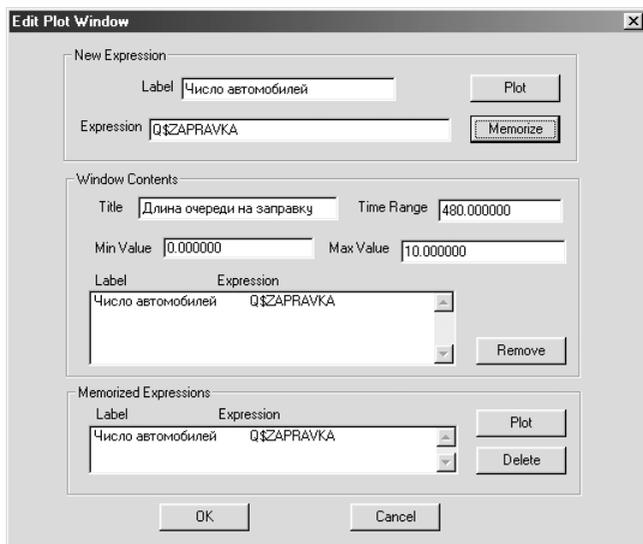


Рис. 4.23. Диалоговое окно **Edit Plot Window** для имитационной модели АЗС

После заполнения диалогового окна **Edit Plot Window** щелкните по кнопкам **Plot** (График), **Memorize** (Запомнить), а затем – по кнопке **OK**. Появится заготовка графика (рис. 4.24).

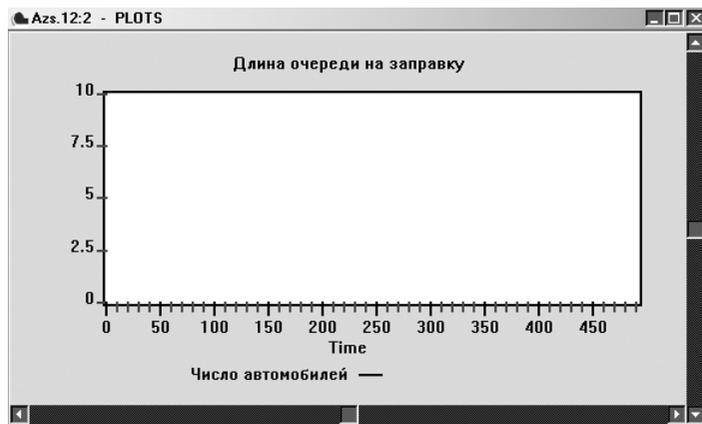


Рис. 4.24. Заготовка графика длины очереди на заправку

После этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **START**. Появится диалоговое окно **Start Command**;

- введите в диалоговом окне **Start Command** число 1 и щелкните по кнопке **ОК**. Появится окно **REPORT** (Отчет) с результатами моделирования. На заднем плане будет размещаться график;
- щелкните по графику, расположенному на заднем плане, – он выйдет на первый план;
- используя горизонтальную и вертикальную полосы прокрутки, вы можете просмотреть построенный график.

Фрагмент графика для нашего примера представлен на рис. 4.25.

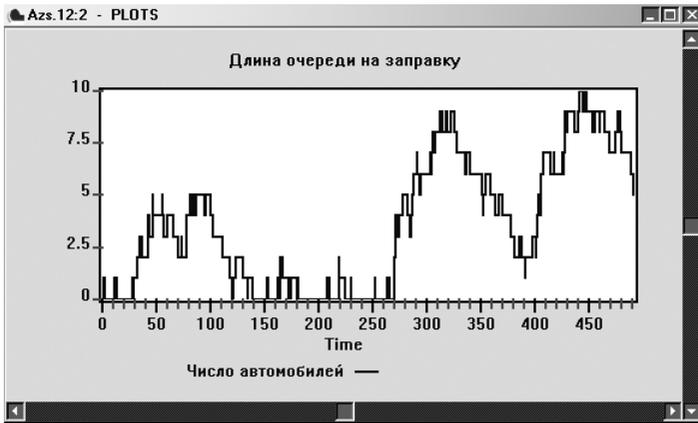


Рис. 4.25. Фрагмент изображения длины очереди на заправку

При выводе графика на передний экран окно **REPORT** с результатами моделирования переместится на задний план. Для просмотра окна **REPORT** щелкните по нему. Оно вновь появится на переднем плане и будет выглядеть так, как показано на рис. 4.26.

В верхней строке указывается:

- **START TIME** (Начальное время) – 0.000;
- **END TIME** (Время окончания) – 480.000;
- **BLOCKS** (Число блоков) – 16;
- **FACILITIES** (Число каналов обслуживания) – 2;
- **STORAGES** (Число накопителей) – 0.

Ниже указываются результаты моделирования каналов обслуживания соответственно под назначенными нами именами **KOLONKA_1** и **KOLONKA_2**:

- **ENTRIES** (Число входов) – 45 и 33;
- **UTIL.** (Коэффициент использования) – 0.901 и 0.875;
- **AVE. TIME** (Среднее время обслуживания) – 9.608 и 12.734;
- **AVAIL.** (Доступность) – 1 и 1;
- **OWNER** (Возможное число входов) – 78 и 79;

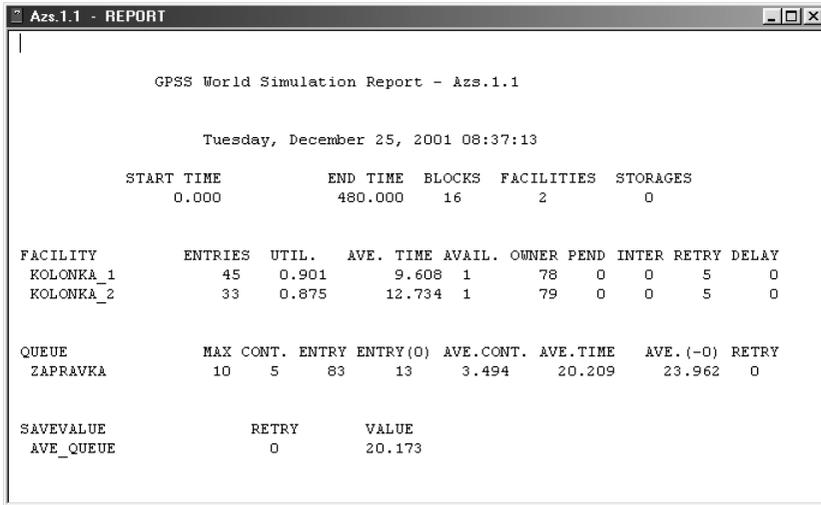


Рис. 4.26. Окно **REPORT** с фрагментом результатов моделирования имитационной модели АЗС

- **PEND** – 0 и 0;
- **INTER** – 0 и 0;
- **RETRY** (Повтор) – 5 и 5;
- **DELAY** (Отказ) – 0 и 0.

Еще ниже указываются результаты моделирования очереди под именем ZAPRAVKA:

- **MAX** (Максимальное содержание) – 10;
- **CONT.** (Текущее содержание) – 5;
- **ENTRY** (Число входов) – 83;
- **ENTRY(0)** (Число нулевых входов) – 13;
- **AVE.CONT.** (Среднее число входов) – 3.494;
- **AVE.TIME** (Среднее время пребывания в очереди) – 20.209;
- **AVE.(-0)** – 23.962;
- **RETRY** – 0.

Еще ниже приведены параметры сохраняемой величины под именем AVE_QUEUE:

- **RETRY** – 0;
- **VALUE** (Значение) – 20.173.

Для получения дополнительной информации можно ввести в начале программы команду **QTABLE** для построения соответствующей гистограммы. Эта команда для нашей задачи может быть записана в таком виде:

```
INFORM QTABLE Zapravka,0,3,35
```

Если в диалоговом окне установок **Azs – SETTINGS** поставить напротив опции **Tables** галочку, то после выполнения программы в окне **Azs – REPORT** появится дополнительная информация (в табличном виде) по функционированию очереди под именем **ЗАПРАВКА**. Для нашего примера она будет выглядеть так, как представлено на рис. 4.27.

TABLE	MEAN	STD. DEV.	RANGE	RETRY	FREQUENCY	CUM. %
INFORM	19.723	15.581		0		
			-	0.000	13	16.67
		0.000	-	3.000	5	23.08
		3.000	-	6.000	3	26.92
		6.000	-	9.000	3	30.77
		9.000	-	12.000	4	35.90
		12.000	-	15.000	5	42.31
		15.000	-	18.000	5	48.72
		18.000	-	21.000	6	56.41
		21.000	-	24.000	4	61.54
		24.000	-	27.000	4	66.67
		27.000	-	30.000	4	71.79
		30.000	-	33.000	3	75.64
		33.000	-	36.000	5	82.05
		36.000	-	39.000	2	84.62
		39.000	-	42.000	3	88.46
		42.000	-	45.000	3	92.31
		45.000	-	48.000	3	96.15
		48.000	-	51.000	3	100.00

Рис. 4.27. Фрагмент содержимого окна **REPORT** – блок **TABLE** по имени **INFORM**

При этом выводятся следующие результаты:

- **MEAN** (Средняя) – 19.723;
- **STD.DEV.** (Среднее квадратическое отклонение) – 15.581;
- **RANGE** (Область);
- **RETRY** – 0;
- **FREQUENCY** (Частота);
- **CUM.%** (Суммарный процент).

Эти результаты могут быть использованы для построения по табличным данным соответствующей гистограммы под именем **INFORM**.

Для этого:

- щелкните по пункту **Window** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** (Окно моделирования) выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Table Window** (Окно гистограммы) во всплывающем меню. Появится диалоговое окно **Open Table Window** (Открыть окно гистограммы). В раскрывающемся списке **Table** щелкните по нужной гистограмме. Для нашей задачи это окно выглядит так, как показано на рис. 4.28;
- щелкните по кнопке **OK**. Появится соответствующая гистограмма. Для нашей задачи она выглядит так, как представлено на рис. 4.29.

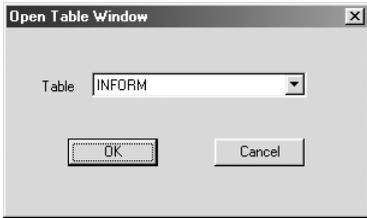


Рис. 4.28. Диалоговое окно выбора гистограммы имитационной модели АЗС

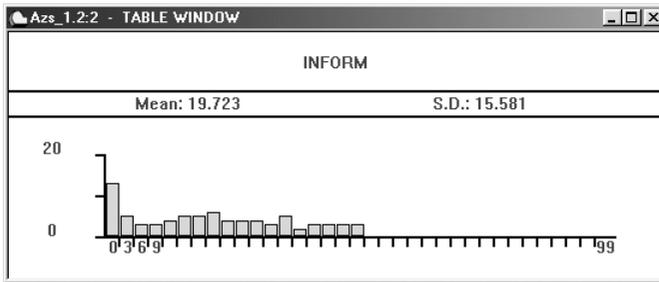


Рис. 4.29. Окно гистограммы имитационной модели АЗС

Визуализация процесса функционирования системы

Перед началом моделирования, а точнее после трансляции модели, система GPSSW обеспечивает возможность визуального наблюдения перемещения активных требований (транзактов) в процессе моделирования. Если в модели есть команда управления **START**, она должна быть заблокирована, то есть переведена в комментарии. Для этого в позиции 1 поставьте звездочку. Процесс визуального наблюдения перемещения активных требований при моделировании включает следующие этапы:

- щелкните по пункту **Window** главного меню системы или нажмите комбинацию клавиш **Alt+W**. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Block Entities** (Блочные элементы) всплывающего меню. Появится диалоговое окно **BLOCK ENTITIES** (рис. 4.30).

Для визуального наблюдения перемещения активных требований в процессе моделирования:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **START**. Появится диалоговое окно **Start Command**;

Loc	Block Type	Current Count	Entry Count	Retry Chain	Line Number	Include-file
1 GEN	GENERATE	0	0	0	6	0
2 QUE	QUEUE	0	0	0	7	0
3 TRA	TRANSFER	0	0	0	8	0
KOL_1	SEIZE	0	0	0	9	0
5 DEP	DEPART	0	0	0	10	0
6 ADV	ADVANCE	0	0	0	11	0
7 REL	RELEASE	0	0	0	12	0
8 TRA	TRANSFER	0	0	0	13	0
KOL_2	SEIZE	0	0	0	14	0
10 DEP	DEPART	0	0	0	15	0
11 ADV	ADVANCE	0	0	0	16	0
12 REL	RELEASE	0	0	0	17	0
NEXT	SAVEVALUE	0	0	0	18	0
14 TER	TERMINATE	0	0	0	19	0
15 GEN	GENERATE	0	0	0	20	0
16 TER	TERMINATE	0	0	0	21	0

Рис. 4.30. Блок-схема моделируемой системы АЗС

- щелкните по кнопке **OK**. Начнется процесс поступления и перемещения активных требований (автомашин) в моделируемой системе (АЗС). Каждое перемещение требования по блокам системы фиксируется в правой части окна;
- щелкните по кнопке **Halt** (Остановить), расположенной на панели кнопок управления в верхней правой части окна **BLOCK ENTITIES**. Одно из состояний моделирования системы, зафиксированное в окне **BLOCK ENTITIES**, представлено на рис. 4.31;

Loc	Block Type	Current Count	Entry Count	Retry Chain	Line Number	Include-file
1 GEN	GENERATE	0	17	0	6	0
2 QUE	QUEUE	0	17	0	7	0
3 TRA	TRANSFER	4	17	0	8	0
KOL_1	SEIZE	0	8	0	9	0
5 DEP	DEPART	0	8	0	10	0
6 ADV	ADVANCE	1	8	0	11	0
7 REL	RELEASE	0	7	0	12	0
8 TRA	TRANSFER	0	7	0	13	0
KOL_2	SEIZE	0	5	0	14	0
10 DEP	DEPART	0	5	0	15	0
11 ADV	ADVANCE	1	5	0	16	0
12 REL	RELEASE	0	4	0	17	0
NEXT	SAVEVALUE	0	11	0	18	0
14 TER	TERMINATE	0	11	0	19	0
15 GEN	GENERATE	0	0	0	20	0
16 TER	TERMINATE	0	0	0	21	0

Рис. 4.31. Одно из состояний моделирования системы АЗС в детальном представлении

- щелкните по кнопке **Continue** (Продолжить) для продолжения моделирования или по кнопке **Step** (Шагнуть), чтобы промоделировать и просмотреть изменения в системе в течение одного шага. Кнопку **Step** можно нажимать многократно для подробного просмотра последовательных шагов моделирования системы.

Можно также наблюдать перемещения активных требований (транзактов) в процессе моделирования в том же окне **BLOCK ENTITIES**, но только без излишних деталей. Для этого:

- щелкните по пункту **View** (Вид) главного меню системы или нажмите комбинацию клавиш **Alt+V**. Появится выпадающее меню;
- щелкните по пункту **Entity Details** (Детальное представление) выпадающего меню. По умолчанию перед пунктом **Entity Details** стоит галочка, что означает активизацию режима детального представления моделируемой системы. После щелчка по этому пункту лишняя информация исчезнет, и окно **BLOCK ENTITIES** примет вид, который показан на рис. 4.32.

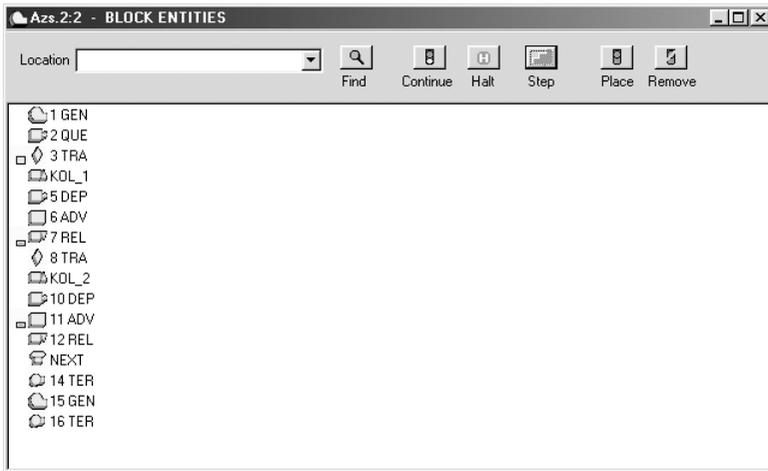


Рис. 4.32. Одно из состояний моделирования системы АЗС в упрощенном представлении

Следует помнить, что включение окна просмотра **BLOCK ENTITIES** значительно замедляет процесс моделирования, так как тратится время на визуализацию процесса моделирования. Для ускорения процесса моделирования и быстрого получения конечного результата целесообразно закрыть окно **BLOCK ENTITIES**. Это можно сделать несколькими способами:

- щелкнуть по кнопке с крестиком, расположенной в правом верхнем углу окна;
- щелкнуть по пиктограмме блока, расположенной в левом верхнем углу окна. Появится всплывающее меню, в котором щелкните по пункту **Close** (Закрыть);
- нажать комбинацию клавиш **Ctrl+F4**;

- дважды щелкнуть по пиктограмме блока, расположенной в левом верхнем углу окна.

Вывод отдельных результатов моделирования

В системе GPSSW возможно вывести значения искомых параметров в окне **JOURNAL** (Журнал), используя команду **SHOW** (Показать) и соответствующие стандартные числовые атрибуты системы. Для этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **SHOW**. Появится диалоговое окно **Show Command** (рис. 4.33);
- введите в поле после команды **SHOW** обращение к искомому параметру

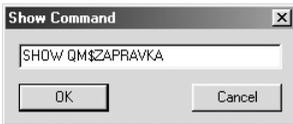


Рис. 4.33. Диалоговое окно **Show Command** с указанием искомого параметра

с помощью стандартного числового атрибута. В нашем примере используется стандартный числовой атрибут **QM<имя>**, обеспечивающий получение максимальной длины очереди под именем **ZAPRAVKA**;

- щелкните по кнопке **OK**. Значение использованного стандартного числового атрибута появится в окне **JOURNAL**.

Многочисленное использование команды **Show** позволяет получить числовые значения искомых параметров. На рис. 4.34 представлено окно **JOURNAL** с несколькими искомыми параметрами для нашей задачи.



Рис. 4.34. Окно **JOURNAL** с фрагментом искомых результатов

Моделирование работы инструментальной кладовой

Постановка задачи

В цехе работает инструментальная кладовая по принципу самообслуживания. Рабочие приходят за инструментом в среднем каждые 8 мин с возможным отклонением от этого интервала ± 2 мин. Поток рабочих (требований) за инструментами равномерный. Каждый рабочий может взять один или несколько инструментов, лежащих на разных стеллажах. Время, необходимое для поиска инструмента на стеллажах, число инструментов, взятых со стеллажа, и вероятности взятия их приведены в табл. 4.1.

Таблица 4.1

Стеллаж	Вероятность взятия инструмента	Время поиска инструмента, мин	Число инструментов, взятых со стеллажа, шт.
1	0,65	10 \pm 4	4 \pm 2
2	0,78	12 \pm 2	2 \pm 1

Взяв инструмент в кладовой, рабочий подходит к учетчику, который делает соответствующую отметку в журнале. Время учета пропорционально числу инструментов, взятых рабочим. На оформление одного наименования инструмента требуется 1 мин. Необходимо определить:

- число посещений рабочими кладовой в течение смены;
- среднее время для взятия инструмента в кладовой;
- коэффициент использования (загрузки) учетчика;
- максимальную длину очереди в кладовой;
- среднюю длину очереди в кладовой;
- число входов в кладовую без ожидания;
- процент входов в кладовую без ожидания;
- среднее время учета взятого инструмента.

Выявление основных особенностей

Для моделирования работы инструментальной кладовой необходимо сформировать входной поток рабочих (требований) и временной интервал моделирования. Но перед этим нужно выбрать единицу измерения времени. Для моделирования работы инструментальной кладовой можно взять в качестве единицы измерения времени минуту.

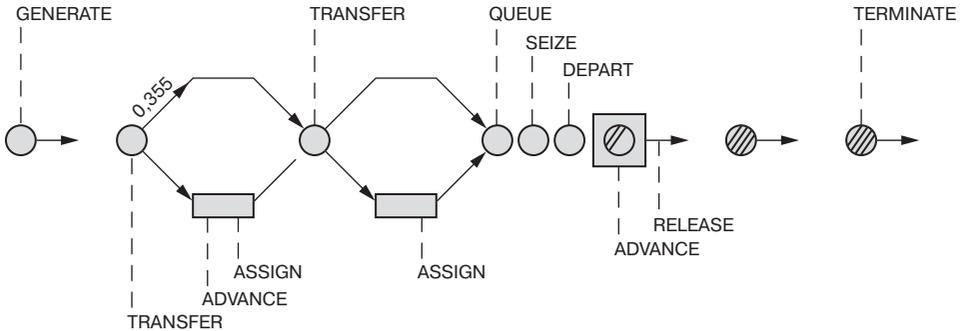


Рис. 4.35. Процесс функционирования системы «Рабочие–кладовая»

Изобразим графически процесс функционирования системы «Рабочие–кладовая» (рис. 4.35).

Создание имитационной модели процесса

Построение имитационной модели начнем с создания заголовка модели, который может быть представлен, например, в таком виде:

```
; GPSSW                               Inst_clad.GPS
*****
*      Моделирование системы «Рабочие–кладовая»      *
*****
```

В программе можно выделить три сектора. В первом секторе формируются необходимые функции. Во втором проводится собственно моделирование работы инструментальной кладовой. В третьем секторе моделируется время работы системы.

В первом секторе создадим две функции и вещественную переменную. Первая функция под именем INST1 будет моделировать взятие рабочим определенного числа инструментов с первого стеллажа. Для определения числа инструментов, взятых рабочим с первого стеллажа, используется дискретная числовая функция, показанная на рис. 4.36.

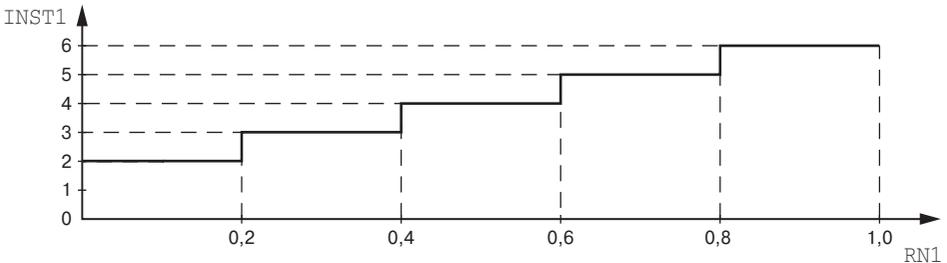


Рис. 4.36. Дискретная функция распределения числа инструментов с первого стеллажа, необходимых рабочему

В модели эту функцию можно записать в таком виде:

```
INST1 FUNCTION RN1, D5
```

```
0.2, 2/0.4, 3/0.6, 4/0.8, 5/1, 6
```

Вторая функция под именем INST2 будет моделировать взятие рабочим определенного числа инструментов со второго стеллажа. Для определения числа инструментов, взятых рабочим со второго стеллажа, используется дискретная числовая функция, показанная на рис. 4.37.

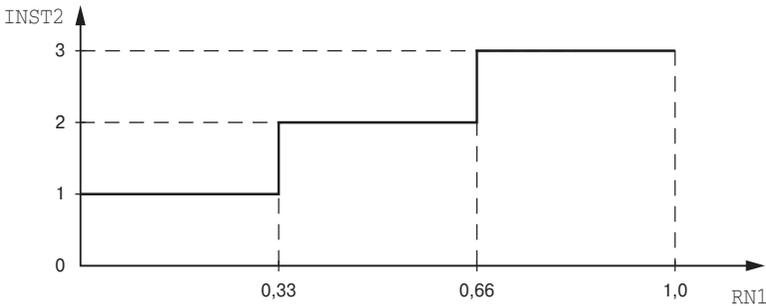


Рис. 4.37. Дискретная функция распределения числа инструментов со второго стеллажа, необходимых рабочему

В модели эту функцию можно записать в таком виде:

```
INST2 FUNCTION RN1, D3
```

```
0.33, 1/0.66, 2/1, 3
```

Переменная по имени T_OBSL будет определять время обслуживания учетчиком рабочего, которое зависит от числа инструментов, взятых рабочим с обоих стеллажей:

```
T_OBSL FVARIABLE (FN$INST1+FN$INST2)#1.2
```

Второй сектор начинается с моделирования потока рабочих в инструментальную кладовую. Это можно выполнить с помощью оператора **GENERATE** (Генерировать). В нашем примере он будет выглядеть так:

```
GENERATE 8, 2
```

В поле операнда A указывается средний интервал времени между прибытием в инструментальную кладовую двух идущих один за другим рабочих (требований, транзактов). В нашем примере среднее время прибытия требований составляет 8 мин. В поле операнда B дано отклонение времени поступления рабочих от среднего. В нашем примере оно составляет 2 мин.

Пришедшие в кладовую рабочие с вероятностью 0,645 будут брать инструменты с первого стеллажа, а остальные – со второго. Это действие можно промоделировать с помощью оператора **TRANSFER** (Передать), используемого в режиме

перехода к заданному оператору с указанной вероятностью. В нашей задаче он будет выглядеть так:

```
TRANSFER .355,,STEL2
```

Такая запись означает, что рабочий с вероятностью 0,355, указанной в поле операнда А, пойдет к оператору, имя метки которого определено в поле операнда С, и с вероятностью $1 - 0,355 = 0,645$ пойдет к оператору, следующему сразу же за оператором **TRANSFER**. Таким образом, данный оператор направляет рабочих к первому или ко второму стеллажу. Сразу же после оператора **TRANSFER** должен быть оператор, моделирующий время поиска нужных инструментов на первом стеллаже. Это время составляет 10 ± 4 мин с равномерным законом распределения. Для этого можно использовать оператор **ADVANCE** (Задержать), который в нашей задаче запишется так:

```
ADVANCE 10,4
```

Далее рабочие, взявшие инструмент с первого стеллажа, с вероятностью 0,779 будут брать инструменты со второго стеллажа. Это действие можно промоделировать с помощью оператора **TRANSFER**, используемого в режиме перехода к заданному оператору с указанной вероятностью. В нашей задаче он будет выглядеть так:

```
STEL2 TRANSFER .221,,KLAD
```

Такая запись означает, что рабочий с вероятностью 0,221, указанной в поле операнда А, пойдет к оператору, имя метки которого определено в поле операнда С, и с вероятностью $1 - 0,221 = 0,779$ пойдет к оператору, следующему сразу же за оператором **TRANSFER**. Таким образом, данный оператор направляет рабочих ко второму стеллажу или к учетчику – кладовщику.

В операторе **TRANSFER** имеется метка **STEL2** – по ней к этому оператору направляются рабочие, которым не нужны инструменты с первого стеллажа.

Сразу же после оператора **TRANSFER** должен быть оператор, моделирующий время поиска нужных инструментов на втором стеллаже. Это время составляет 12 ± 2 мин с равномерным законом распределения. Для этого можно использовать оператор **ADVANCE**, который в нашей задаче запишется так:

```
ADVANCE 12,2
```

Рабочий, взявший все необходимые инструменты, встает в очередь к учетчику, если она есть. Это можно промоделировать оператором **QUEUE** (Очередь), который только в совокупности с оператором **DEPART** (Выйти) собирает статистическую информацию о работе моделируемой очереди. В нашем примере оператор **QUEUE** будет выглядеть так:

```
KLAD QUEUE UCNET
```

В поле операнда А дается символьное или числовое имя очереди. Таких очередей в системе может быть очень много. В нашей задаче дадим очереди имя **UCNET**.

Желательно, чтобы присваиваемое имя отражало суть описываемого элемента системы.

В операторе **QUEUE** имеется метка **CLAD** – по ней к этому оператору направляются рабочие, которым не нужны инструменты со второго стеллажа. Следуя логике, рабочий может выйти из очереди только тогда, когда освободится учетчик (объект). Для этого вводится оператор **SEIZE** (Занять), который определяет занятость объекта, и при его освобождении очередное требование выходит из очереди и идет в канал на обслуживание. Это может выглядеть так:

```
SEIZE CLAD
```

В поле операнда **A** дается символьное или числовое имя объекта. Таких каналов обслуживания в системе может быть очень много. В нашей задаче каналу дано имя **CLAD**. Желательно, чтобы присваиваемое имя отражало суть описываемого элемента системы.

Выход рабочего из очереди к учетчику фиксируется оператором **DEPART** с соответствующим названием очереди. В нашем примере это будет выглядеть так:

```
DEPART UCNET
```

Далее должно быть промоделировано время пребывания в кладовой рабочего, непосредственно обслуживаемого учетчиком. Это время в нашем примере зависит от числа взятых инструментов с двух стеллажей. Время обслуживания рабочего учетчиком определяется с помощью функции по имени **OBSL**. Для моделирования этого процесса используется оператор **ADVANCE**, который в нашей задаче будет выглядеть так:

```
ADVANCE V$T_OBSL
```

Продолжительность обслуживания рабочего учетчиком определяется обращением к переменной по имени **T_OBSL**, которая в зависимости от числа взятых инструментов устанавливает время обслуживания. Эта вещественная переменная вычисляется так:

$$(FN\$INST1+FN\$INST2) \# 1.2$$

Как вы видите, в состав выражения входят значения двух функций **FN\$INST1** и **FN\$INST2**, которые необходимо предварительно определить.

Число инструментов с первого стеллажа, взятых рабочим, определяется следующим образом. С помощью датчика псевдослучайных чисел **RN1** генерируется случайное число. Допустим, оно равно 0,468, тогда, согласно заданным значениям дискретной функции **Inst1** (см. рис. 4.36), число взятых инструментов будет равно 4.

Число инструментов со второго стеллажа, взятых рабочим, определяется аналогично. С помощью датчика псевдослучайных чисел **RN2** генерируется случайное число. Допустим, что оно равно 0,588, тогда, согласно заданным значениям дискретной функции **Inst2** (см. рис. 4.37), число инструментов, взятых со второго стеллажа, будет равно 2.

В нашем примере рабочий взял 6 инструментов. Это число умножается на коэффициент 1,2, чтобы определить время учета взятых инструментов.

После обслуживания рабочего учетчиком должно быть послано сообщение об освобождении объекта. Это моделируется оператором **RELEASE**:

```
RELEASE CLAD
```

Следует особо подчеркнуть, что парные операторы **QUEUE** и **DEPART** для каждой очереди должны иметь одно и то же, но свое уникальное имя. Это же относится и к операторам **SEIZE** и **RELEASE**.

После обслуживания учетчиком рабочий покидает инструментальную кладовую. Это действие может быть представлено оператором **TERMINATE** (Завершить):

```
TERMINATE
```

И наконец, мы должны создать сегмент, который будет моделировать работу инструментальной кладовой в течение рабочей смены, равной 8 ч. Поскольку моделирование работы инструментальной кладовой проводится в минутах, то время моделирования системы будет равно $8 \times 60 = 480$ мин. Этот сегмент будет выглядеть так:

```
GENERATE 480
TERMINATE 1
START 1
```

Представление имитационной модели

Для представления имитационной модели выполните следующие действия:

- щелкните по пункту **File** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **New** (Создать) выпадающего меню. Появится диалоговое окно **Новый документ**;
- выделите пункт **Model** и щелкните по кнопке **ОК**. Появится окно модели, в котором введите данную программу, как показано на рис. 4.38.

Вызвать окно для представления имитационной модели в системе GPSSW можно также, нажав комбинацию клавиш **Ctrl+Alt+S**.

Моделирование системы

После создания имитационную модель необходимо оттранслировать и запустить на выполнение. Для этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню.

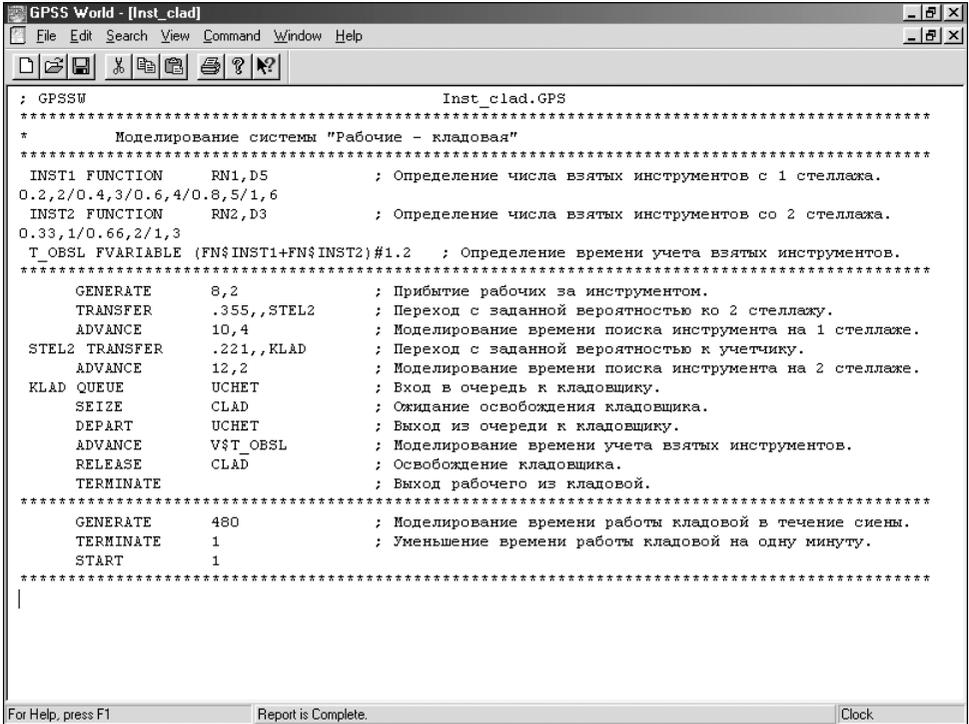


Рис. 4.38. Окно имитационной модели инструментальной кладовой

Поскольку в программе есть управляющая команда **START**, то исходная имитационная модель будет транслироваться, и если в ней нет ошибок, то начнется процесс моделирования системы.

Подготовка к моделированию

Перед началом моделирования можно установить вывод тех параметров моделирования, которые нужны пользователю. Для этого:

- щелкните по пункту **Edit** (Правка) главного меню системы или нажмите комбинацию клавиш **Alt+E**. Появится выпадающее меню;
- щелкните по пункту **Settings** (Установки) выпадающего меню. Появится диалоговое окно **SETTINGS** для данной модели, в котором можно установить нужные выходные данные, например, так, как показано на рис. 4.39.

Наличие галочки в окошках говорит о том, что эта информация будет выведена в окне результатов моделирования. В нашем примере будет выведена информация по следующим объектам:

- **Queues** (Очереди);
- **Facilities** (Каналы обслуживания).

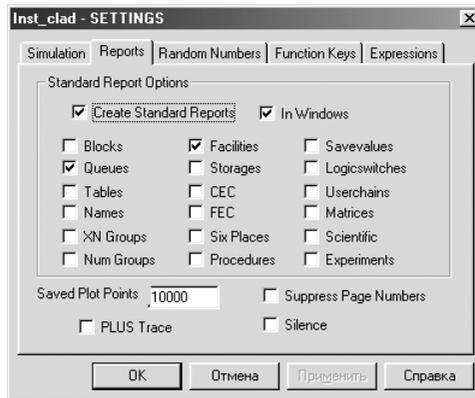


Рис. 4.39. Окно **SETTINGS** с установками для имитационной модели инструментальной кладовой

Для начала моделирования:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** выпадающего меню. Появится окно **JOURNAL** (Журнал) с выводом сообщений о времени выполнения тех или иных действий, а затем – окно **REPORT** (Отчет) с результатами моделирования (рис. 4.40).

GPSS World Simulation Report - Inst_clad.1.1

Sunday, December 23, 2001 22:42:03

	START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
	0.000	480.000	15	1	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
CLAD	57	0.807	6.794	1	57	0	0	0	0

QUEUE	MAX CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
UCHET	3	0	57	17	0.517	4.352	6.201

Рис. 4.40. Окно **REPORT** с фрагментом результатов моделирования работы инструментальной кладовой

В верхней строке указывается:

- **START TIME** (Начальное время) – 0.000;
- **END TIME** (Время окончания) – 480.000;

- **BLOCKS** (Число блоков) – 15;
- **FACILITIES** (Число каналов обслуживания) – 1;
- **STORAGES** (Число накопителей) – 0.

Ниже указываются результаты моделирования канала обслуживания под назначенным именем CLAD:

- **ENTRIES** (Число входов) – 57;
- **UTIL.** (Коэффициент использования) – 0.807;
- **AVE. TIME** (Среднее время обслуживания) – 6.794;
- **AVAIL.** (Доступность) – 1;
- **OWNER** (Возможное число входов) – 57;
- **PEND** – 0;
- **INTER** – 0;
- **RETRY** (Повтор) – 0;
- **DELAY** (Отказ) – 0.

Еще ниже указываются результаты моделирования очереди под присвоенным именем UCNET:

- **MAX** (Максимальное содержание) – 3;
- **CONT.** (Текущее содержание) – 0;
- **ENTRY** (Число входов) – 57;
- **ENTRY(0)** (Число нулевых входов) – 17;
- **AVE.CONT.** (Среднее число входов) – 0.517;
- **AVE.TIME** – 4.352;
- **AVE.(–0)** – 6.201;
- **RETRY** – 0.

Моделирование системы управления качеством

Постановка задачи

Допустим, нам надо промоделировать систему управления качеством производственного процесса, включающего две операции обработки изделия с соответствующим контролем. Известны следующие параметры производственного процесса:

- поток изделий, поступающих на обработку, подчиняется экспоненциальному распределению вероятностей с параметрами $\lambda = 0$ и $\beta = 28$;
- время выполнения первой операции определяется с помощью дискретной числовой функции, а время на контроль этой операции составляет 3 мин;
- время выполнения второй операции определяется с использованием нормального распределения с параметрами $\mu = 22$, $\sigma = 3$, а время на контроль этой операции составляет 2 мин.

Нужно промоделировать работу системы управления качеством в течение рабочей смены – 8 ч. Требуется определить параметры функционирования производственного процесса:

- коэффициент загрузки каждого контролера;
- среднее время обслуживания каждым контролером;
- максимальное, среднее и текущее число изделий у каждого контролера и др.

Выявление основных особенностей

Для моделирования производственного процесса необходимо сформировать входной поток изделий, поступающих на обработку с интервалом времени, соответствующим экспоненциальному закону. Интервал времени вычисляется по следующей формуле:

$$f(x) = \frac{1}{\beta} e^{-\frac{(x-\lambda)}{\beta}}, \beta \geq 0.$$

В противном случае его значение равно 0.

Время выполнения первой операции определяется с помощью дискретной числовой функции, задаваемой семью парами чисел (точек):

$$(0, 0), (.04, 9), (.20, 13), (.35, 17), (.60, 25), (.85, 35), (1.0, 50).$$

Время выполнения второй операции определяется с использованием нормального распределения, который представляется в таком виде:

$$f(x) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}.$$

В качестве единицы измерения времени примем минуту.

Модель будем представлять в виде пяти сегментов. В первом секторе сформируем некоторые исходные и выходные данные.

Во втором и третьем секторах промоделируем соответственно выполнение операций и их контроль. В четвертом секторе будем решать вопрос об устранении брака в проконтролированных изделиях. В пятом секторе промоделируем продолжительность выполнения процесса, например, в течение 8-часовой рабочей смены.

Создание имитационной модели процесса

Построение имитационной модели начнем с создания заголовка модели, который может быть представлен, например, в таком виде:

```
; GPSSW File - CONTROL.GPS
*****
*           Моделирование системы           *
*           управления качеством            *
*****
```

В первом секторе будем формировать некоторые исходные и выходные данные. Используем оператор **RMULT**, чтобы задать начальное значение для работы генератора случайных чисел. В нашей задаче мы будем использовать датчик случайных чисел RN1. По умолчанию, когда не используется оператор **RMULT**, начальное значение для работы генератора случайных чисел равно его номеру – для нашей задачи это 1:

```
RMULT 231
```

Далее используем оператор **TABLE** (Таблица) для сбора информации и вывода ее в виде графика:

```
Time_obrab TABLE M1,100,20,7
```

Эта запись означает, что стандартный числовой атрибут M1 должен быть выведен в табличном виде. Нижний предел переменной (поле B) равен 100, приращение (поле C) равно 20 и число приращений (поле D) равно 7.

Стандартный числовой атрибут M1 – это время всего процесса обработки изделия (требования), включая и операции контроля. Это время определяется как разность абсолютного времени функционирования системы и времени вхождения изделия (требования) в оператор **TABULATE**.

Далее определим дискретную числовую функцию для вычисления времени выполнения первой операции. В системе GPSSW она представляется в таком виде:

```
Oper_1 FUNCTION RN1,D7  
0,0/.04,9/.20,13/.35,17/.60,25/.85,35/1.0,50
```

Создание второго сектора модели начнем с моделирования потока изделий, поступающих на обработку. Это действие будем выполнять с помощью оператора **GENERATE** (Генерировать). Нам необходимо сформировать поток изделий, поступающих на обработку, который подчиняется экспоненциальному распределению вероятностей. Это можно сделать двумя способами, используя:

- эмпирические распределения, которые можно создать с помощью команды **FUNCTION** (Функция) языка GPSS, используя дискретные (тип D) или непрерывные (тип C) случайные функции;
- встроенные процедуры распределений вероятности.

Каждый вызов процедуры распределения вероятности требует, чтобы вы определили параметр потока, номер генератора случайных чисел.

В нашем примере оператор **GENERATE** запишется в таком виде:

```
GENERATE (Exponential(1,0,28))
```

В поле операнда A указывается обращение к библиотечной процедуре – экспоненциальному распределению вероятности.

Далее определим с помощью функции по имени Oper_1 длительность выполнения первой операции и сохраним ее значение в параметре изделия (требования)

под номером 1. Это можно сделать с помощью оператора **ASSIGN** (Присвоить). В нашей задаче этот оператор может быть записан в таком виде:

```
ASSIGN 1, FN$Oper_1
```

В поле операнда А указывается номер требования, в которое записывается значение операнда В. В операнде В выполняется обращение к функции по имени Oper_1 с помощью стандартного числового атрибута FN<имя>. Если используется символическое имя, то между именами стандартного числового атрибута и функции ставится знак \$.

Далее необходимо промоделировать процесс выполнения первой операции на первом оборудовании. Это начинается с оператора **SEIZE** (Занять) с меткой Oborud_1, который определяет занятость первого оборудования, выполняющего первую операцию, но с предшествующим изделием. И при окончании выполнения операции следующее изделие (требование) поступает на обработку на первое оборудование (объект), что моделируется оператором **ADVANCE** (Задержать). Окончание обработки фиксируется оператором **RELEASE** (Освободить) с той же меткой Oborud_1.

В поле операнда А дается символьное или числовое имя оборудования (объекта). Таких каналов обслуживания в системе может быть очень много. Желательно, чтобы присваиваемое имя отражало суть описываемого элемента системы.

Все эти действия могут быть представлены так:

```
Stage1 SEIZE      Oborud_1
        ADVANCE    P1
        RELEASE    Oborud_1
```

Операторы **SEIZE** и **RELEASE** предназначены для сбора статистики по каналу обслуживания, с которым они используются.

Далее моделируется время работы контролера по оценке качества выполненной операции. Это может быть выполнено оператором **ADVANCE**. Время контроля, равное 3 мин, записывается в поле операнда А. Допустим, что 15% всех изделий бракуется. Однако есть возможность повторить операции обработки для некоторых изделий. Эту часть модели можно записать так:

```
ADVANCE 3
TRANSFER .150,,Ustr_brak_1
```

Таким образом, в нашей задаче 15% изделий направляются в четвертый сегмент к оператору с меткой Ustr_brak_1.

Аналогичные действия проводятся с изделием при выполнении второй операции на втором оборудовании. Это можно записать так:

```
Oper2 SEIZE      Oborud_2
      ADVANCE    (Normal(1,22,3))
      RELEASE    Oborud_2
```

```

ADVANCE      2
TRANSFER    .060,,Ustr_brak_2
TABULATE    Time_obrab
TERMINATE    1
    
```

Здесь время выполнения второй операции определяется с использованием нормального закона распределения, встроенного в систему. Оператор **TABULATE** предназначен для контроля времени выполнения всего производственного процесса, а оператор **TERMINATE** выводит обработанное изделие из системы.

Перейдем теперь к четвертому сегменту, в котором определяется возможность устранения брака путем возвращения изделия на повторную обработку. При этом из всех забракованных изделий на первую операцию возвращается 30%, а на вторую – 60%. Это можно записать так:

```

Ustr_brak_1  TRANSFER .300,,Oper1
              TERMINATE
Ustr_brak_2  TRANSFER .600,,Oper2
              TERMINATE
    
```

Далее используется оператор **TRANSFER** (Передать) в вероятностном режиме передачи требования к оператору с указанной меткой в поле операнда С. В этом режиме в поле операнда А указывается вещественное число меньше 1. Активное требование переходит к местоположению, указанному в поле операнда С, с вероятностью, данной операндом А. Если операнд А – неотрицательное целое число, то он интерпретируется как одна тысячная и преобразуется в значение вероятности. Например, строка

```
Ustr_brak_1 TRANSFER .300,,Oper1
```

означает, что когда требование входит в оператор **TRANSFER**, оно переходит к оператору с меткой Oper1 с вероятностью 0,700, а с вероятностью 0,300 – к следующему оператору.

И наконец, мы должны создать пятый сегмент, который будет моделировать работу системы управления качеством в течение рабочей смены, равной 8 ч. Поскольку моделирование производственного процесса проводится в минутах, то время моделирования системы будет равно $8 \times 60 = 480$ мин. Этот сегмент будет выглядеть так:

```

GENERATE     480
TERMINATE    1
    
```

Завершающим оператором в нашей задаче является **START**, дающий команду начать моделирование:

```
START        1
```

Представление имитационной модели

Для представления имитационной модели выполните следующие действия:

- щелкните по пункту **File** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **New** (Создать) выпадающего меню. Появится диалоговое окно **Новый документ**;
- выделите пункт **Model** и щелкните по кнопке **ОК**. Появится окно модели, в котором введите данную программу, как показано на рис. 4.41.

```

*****
RMULT      231
Time_obrab TABLE      H1,100,20,7
Oper_1     FUNCTION     RN1,D7
0,0/.04,9/.20,13/.35,17/.60,25/.85,35/1.0,50
*****
GENERATE   (Exponential(1,0,28))
ASSIGN     1, FN$Oper_1
Oper1     SEIZE      Oborud_1
          ADVANCE    P1
          RELEASE    Oborud_1
          ADVANCE    3
          TRANSFER   .150,,Ustr_brak_1
*****
Oper2     SEIZE      Oborud_2
          ADVANCE    (Normal(1,22,3))
          RELEASE    Oborud_2
          ADVANCE    2
          TRANSFER   .060,,Ustr_brak_2
          TABULATE   Time_obrab
          TERMINATE
*****
Ustr_brak_1 TRANSFER   .300,,Oper1
          TERMINATE
Ustr_brak_2 TRANSFER   .600,,Oper2
          TERMINATE
*****
GENERATE   480
TERMINATE  1
START      1
  
```

Рис. 4.41. Окно имитационной модели системы управления качеством

Моделирование системы

После создания имитационную модель необходимо оттранслировать и запустить на выполнение. Для этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню.

Так как в имитационной модели имеется управляющая команда **START**, то исходная имитационная модель будет транслироваться, и если в ней нет ошибок, начнется процесс моделирования системы.

Перед началом моделирования можно установить вывод тех параметров моделирования, которые нужны пользователю. Для этого:

- щелкните по пункту **Edit** (Правка) главного меню системы или нажмите комбинацию клавиш **Alt+E**. Появится выпадающее меню;
- щелкните по пункту **Settings** (Установки) выпадающего меню. Появится диалоговое окно **SETTINGS** для данной модели. В нашем примере появится окно под именем **Qcontrol – SETTINGS**, в котором можно установить нужные выходные данные, как показано на рис. 4.42.

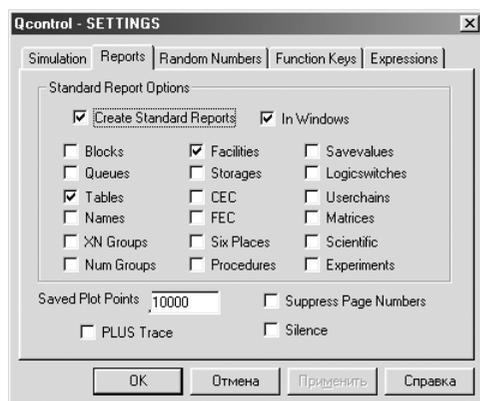


Рис. 4.42. Окно **Qcontrol – SETTINGS** с установками для имитационной модели системы управления качеством

Наличие галочек в окошках говорит о том, что эта информация будет выведена в окне результатов моделирования **REPORT** (Отчет). В нашем примере будет выведена информация по следующим объектам:

- **Facilities** (Каналы обслуживания);
- **Tables** (Таблицы).

После выполнения имитационной модели появится окно **REPORT** с результатами моделирования (рис. 4.43).

В верхней строке указывается:

- **START TIME** (Начальное время) – 0.000;
- **END TIME** (Время окончания) – 480.000;
- **BLOCKS** (Число блоков) – 20;
- **FACILITIES** (Число каналов обслуживания) – 2;
- **STORAGES** (Число накопителей) – 0.

Ниже указываются результаты моделирования каналов обслуживания соответственно под назначенными нами именами **OBORUD_1** и **OBORUD_2**:

- **ENTRIES** (Число входов) – 16 и 13;
- **UTIL.** (Коэффициент использования) – 0.879 и 0.577;

GPSS World Simulation Report - Qcontrol.1.1

Thursday, January 03, 2002 09:08:20

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	480.000	20	2	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
OBORUD_1	16	0.879	26.375	1	0	0	0	0	0
OBORUD_2	13	0.577	21.307	1	0	0	0	0	0

TABLE	MEAN	STD.DEV.	RANGE	RETRY	FREQUENCY	CUM.%
TIME_OBRAB	94.359	23.880		0		
			-	100.000	10	83.33
			100.000 -	120.000	0	83.33
			120.000 -	140.000	1	91.67
			140.000 -	160.000	1	100.00

Рис. 4.43. Окно **REPORT** с фрагментом результатов моделирования системы управления качеством

- **AVE. TIME** (Среднее время обслуживания) – 26.375 и 21.307;
- **AVAIL.** (Доступность) – 1 и 1;
- **OWNER** (Возможное число входов) – 0 и 0;
- **PEND** – 0 и 0;
- **INTER** – 0 и 0;
- **RETRY** (Повтор) – 0 и 0;
- **DELAY** (Отказ) – 0 и 0.

Еще ниже приводится содержимое таблицы под именем `TIME_OBRAB`. При этом выводятся следующие результаты:

- **MEAN** (Средняя) – 94.359;
- **STD.DEV.** (Среднее квадратическое отклонение) – 23.880;
- **RANGE** (Область);
- **RETRY** – 0;
- **FREQUENCY** (Частота);
- **CUM.%** (Суммарный процент).

Эти результаты могут быть использованы для построения по табличным данным соответствующей гистограммы под именем **TIME_OBRAB**.

Построение гистограммы

Для построения гистограммы:

- щелкните по пункту **Window** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** (Окно моделирования) выпадающего меню. Появится всплывающее меню;

- щелкните по пункту **Table Window** (Окно гистограммы) во всплывающем меню. Появится диалоговое окно **Open Table Window** (Открыть окно гистограммы). В раскрывающемся списке **Table** щелкните по нужной гистограмме. Для нашей задачи это окно выглядит так, как показано на рис. 4.44;
- щелкните по кнопке **OK**. Появится соответствующая гистограмма. Для нашей задачи она выглядит так, как представлено на рис. 4.45.

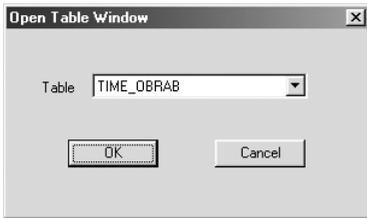


Рис. 4.44. Диалоговое окно выбора нужной гистограммы имитационной модели системы управления качеством

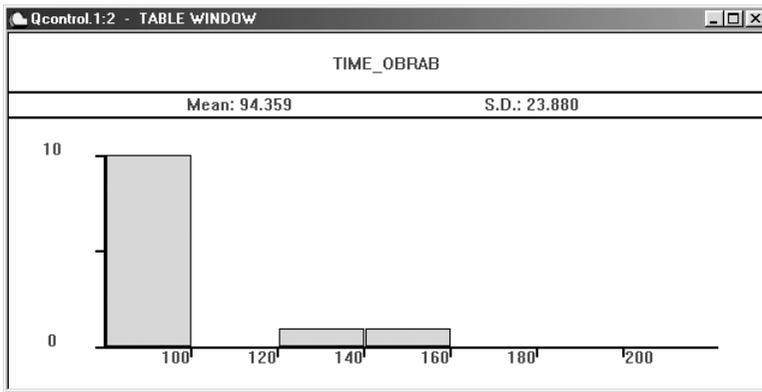


Рис. 4.45. Окно гистограммы имитационной модели системы управления качеством

Моделирование системы управления запасами

Постановка задачи

Допустим, нам надо промоделировать работу склада материалов на предприятии. Вместимость склада составляет 10000 единиц материала. Возможна поставка на склад с периодичностью 5 дней в размере 1000 единиц материала. Начальный запас материалов на складе составляет 1000 единиц. Ежедневный спрос материала изменяется в пределах от 35 до 50 единиц с равной вероятностью. Если текущий

запас равен или больше 800, то никакой поставки материалов на склад в течение недели не производят. Требуется смоделировать работу склада материалов в течение 200 дней.

Выявление основных особенностей

Для моделирования работы склада необходимо сформировать входные потоки заказов материала на склад, определить спрос материалов со склада и временной интервал моделирования работы склада. Но перед этим необходимо выбрать единицу измерения времени. Для моделирования работы склада можно взять в качестве единицы измерения времени день.

Создание имитационной модели процесса

Создание имитационной модели начнем с построения заголовка модели, который может быть представлен, например, в таком виде:

```
; GPSSW Upr_zapasom.GPS
*****
*           Управление запасами           *
*****
```

Далее в имитационной модели создадим четыре сектора:

- 1) сектор определения входных данных;
- 2) моделирование поставки материалов на склад;
- 3) моделирование текущего запаса с учетом спроса;
- 4) установление начального размера запаса.

Рассмотрим последовательно создание каждого из секторов.

Первый сектор может быть представлен в таком виде:

```
* Определение входных данных.
      RMULT      413
Zapas  STORAGE  2000          ; Вместимость склада.
Zapas  TABLE   S$Zapas,100,100,20; Гистограмма запаса.
Material VARIABLE Nach_ur-S$Zapas ; Размер заказа.
Spros_m VARIABLE RN1@15+35      ; Размер ежедневного спроса.
Nach_ur EQU     1500            ; Начальный уровень запаса.
Postavka EQU    1200            ; Размер поставки.
```

Оператор **RMULT** определяет начальное число для генератора случайных чисел.

Оператор **STORAGE** (Накопитель) с меткой *Zapas* определяет вместимость склада – 2000 единиц.

Оператор **TABLE** с меткой *Zapas* предназначен для формирования таблицы текущего (ежедневного) уровня запаса на складе. Она может быть использована

для построения соответствующей гистограммы, если потребуется. Текущий уровень запаса определяется с помощью стандартного числового атрибута `S$Zapas`.

Оператор **VARIABLE** с меткой `Material` определяет количество материала как разность начального и текущего запаса.

Оператор **VARIABLE** с меткой `Spros_m` характеризует ежедневный спрос, который изменяется от 35 до 50 единиц с равной вероятностью. Ежедневный спрос определяется так. Датчик случайных чисел, а точнее датчик псевдослучайных чисел, определяет, используя равномерное распределение вероятностей, значение вероятностей в интервале [0–1]. Допустим, это значение равно 0,150. Тогда величина спроса будет определяться так:

$$35 + (50 - 35) \times 0,15 = 37,25.$$

Оператор **EQU** с меткой `Nach_ur` определяет начальный уровень запасов, который в нашей задаче установлен в размере 1500 единиц. Использование именованной величины облегчает проведение экспериментов с ее различными значениями.

Оператор **EQU** с меткой `Postavka` определяет тот критический уровень материалов на складе, который обуславливает необходимость очередного заказа материалов. Этот размер в нашей задаче составляет 1200 единиц.

Рассмотрим теперь второй сектор модели, который обеспечивает моделирование процесса поставки материала на склад. Он будет выглядеть так:

```

GENERATE      5,,,1           ; Поставка через 5 дней.
TEST L        S$Zapas,Postavka,Out ; Нужна ли поставка.
ASSIGN        2,V$Material     ; Размер заказа в P2.
ADVANCE       5                ; Интервал поставки.
ENTER         Zapas,P2         ; Увеличить запас на P2.
Out TERMINATE ; Завершение заказа.
    
```

Оператор **GENERATE** генерирует очередной момент времени, когда возможна поставка материала на склад. Этот интервал времени составляет, например, 5 дней, обеспечивая еженедельную поставку материала на склад при пятидневной рабочей неделе. Этот интервал времени представляется в операнде `A`. В операнде `E` определен уровень приоритетности поставки материала на склад. Он в нашей задаче равен 1. По умолчанию уровень приоритетности для всех требований (транзактов) равен 0. Чем больше число в поле операнда `E`, тем выше уровень требования.

Оператор **TEST L** определяет необходимость очередной поставки материала на склад. Очередная поставка материалов на склад производится тогда, когда текущий запас, который определяется с помощью стандартного числового атрибута `S$Zapas`, меньше размера поставки, определенного оператором **EQU** с меткой `Postavka` в предыдущем секторе. В противном случае поставка отменяется – требование (транзакт) направляется к оператору **TERMINATE** с меткой `Out`.

Если текущий запас меньше размера поставки, то требование переходит к следующему оператору – **ASSIGN** (Присвоить). В операторе **ASSIGN** в параметре

требования под номером 2 запоминается размер заказа, который определяется переменной под именем `Material`.

Оператор **ADVANCE** моделирует время ожидания требования (поставки) в течение 5 дней.

Далее оператор **ENTER** моделирует поставку после истечения 5-дневного срока и увеличивает текущее содержание склада – `S$Zapas` – на величину, содержащуюся в параметре требования под номером 2.

Оператор **TERMINATE** с меткой `Out` удаляет требование из системы. Этот оператор используется, чтобы определить окончание моделирования состояния.

Рассмотрим теперь третий сектор модели, который обеспечивает моделирование текущего запаса с учетом ежедневного спроса на материал. Он будет выглядеть так:

```

GENERATE      1                ; Генерирование дня.
ASSIGN        1,V$Spros_m      ; Размер спроса в P1.
TABULATE      Zapas            ; Запись тек. запаса.
TEST GE       S$Zapas,P1,Zapasout ; Можно ли заказать.
LEAVE         Zapas,P1         ; Уменьшить запас на P1.
TERMINATE     1                ; Завершения дня.
Zapasout TERMINATE 1          ; Завершения дня.

```

Оператор **GENERATE** генерирует очередной день работы склада.

Оператор **ASSIGN** присваивает параметру требования (очередному дню) под номером 1 размер спроса.

Оператор **TABULATE** с меткой `Zapas` собирает информацию для создания таблицы и гистограммы ежедневных уровней запасов.

Оператор **TEST GE** проверяет возможность удовлетворения спроса. Если текущий запас – `S$Zapas` – больше или равен величине спроса, то процесс моделирования переходит к следующему оператору – **LEAVE** (Оставить):

```

LEAVE         Zapas,P1

```

Если такой возможности нет, то требование направляется к оператору с меткой `Zapasout`.

Оператор **LEAVE** уменьшает текущий запас – `S$Zapas` – на величину спроса в данный день. Это делается так: требование (очередной день), входя в блок **LEAVE**, несет в параметре требования под номером 1 величину спроса (операнд В). Эта величина спроса и вычитается из величины текущего запаса `S$Zapas`.

Рассмотрим теперь четвертый сектор модели, который обеспечивает установление начального размера запаса. Он будет выглядеть так:

```

GENERATE      ,, ,1,10         ; Начальный запас.
ENTER         Zapas,Nach_ur    ; Установка нач. запаса.
TERMINATE     ; Завершение установки.

```

Оператор **GENERATE** генерирует одно требование (транзакт) с приоритетом, равным 10. Оно имеет наибольший приоритет среди всех требований, генерируемых

оператором **GENERATE**, и поэтому становится первым активным требованием в системе.

Оператор **ENTER** определяет величину начального запаса, чтобы установить величину $S\$Zapas$ в начале моделирования.

Оператор **TERMINATE** уничтожает начальное требование без уменьшения индекса завершения.

GPSS позволяет многим требованиям одновременно существовать в различных местах в модели. В этой модели требования создаются в трех секторах.

Представление имитационной модели

Для представления имитационной модели выполните следующие действия:

- щелкните по пункту **File** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **New** (Создать) выпадающего меню. Появится диалоговое окно **Новый документ**;
- выделите пункт **Model** и щелкните по кнопке **ОК**. Появится окно модели, в котором введите данную программу, как показано на рис. 4.46.

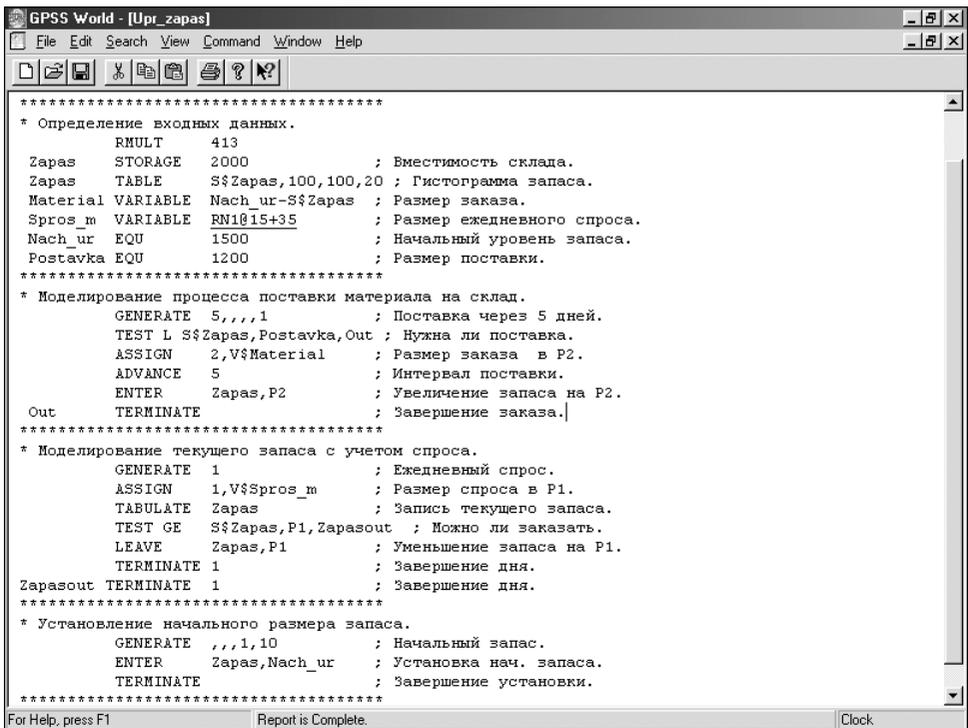


Рис. 4.46. Фрагмент имитационной модели управления запасами

Вызвать окно для представления имитационной модели в системе GPSSW можно также, нажав комбинацию клавиш **Ctrl+Alt+S**.

Моделирование системы

Этот этап включает следующие действия:

- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню. Начнется трансляция модели;
- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **START** выпадающего меню. Появится диалоговое окно, в котором вместо значения 1 введите 200;
- щелкните по кнопке **OK**.

Моделирование закончится, когда 200 ежедневных заказов будут выполнены.

Будет выполнено то количество прогонов процесса моделирования системы, которое указано в поле операнда A команды **START**. Затем появятся окна **JOURNAL** (Журнал) и **REPORT** (Отчет) в стандартном виде, как показано на рис. 4.47.

The screenshot shows the 'REPORT' window in GPSS World. The window title is 'GPSS World - [Upr_zapas.1.1 - REPORT]'. The menu bar includes File, Edit, Search, View, Command, Window, and Help. The toolbar contains icons for file operations, printing, and help. The main area displays two tables of simulation results.

NAME	VALUE
MATERIAL	10007.000
NACH_UR	1500.000
OUT	6.000
POSTAVKA	1200.000
SPROS_M	10008.000
ZAPAS	10006.000
ZAPASOUT	13.000

LABEL	LOC	BLOCK TYPE	ENTRY	COUNT	CURRENT	COUNT	RETRY
	1	GENERATE	0	0	0	0	0
	2	TEST	0	0	0	0	0
	3	ASSIGN	0	0	0	0	0
	4	ADVANCE	0	0	0	0	0
	5	ENTER	0	0	0	0	0
OUT	6	TERMINATE	0	0	0	0	0
	7	GENERATE	1	0	0	0	0
	8	ASSIGN	1	0	0	0	0
	9	TABULATE	1	0	0	0	0
	10	TEST	1	0	0	0	0
	11	LEAVE	1	0	0	0	0
	12	TERMINATE	1	0	0	0	0
ZAPASOUT	13	TERMINATE	0	0	0	0	0
	14	GENERATE	1	0	0	0	0
	15	ENTER	1	0	0	0	0
	16	TERMINATE	1	0	0	0	0

At the bottom of the window, the status bar shows 'For Help, press F1', 'Report is Complete.', and 'Clock'.

Рис. 4.47. Окно **REPORT** для имитационной модели управления запасами

Перед началом моделирования можно установить вывод тех параметров моделирования, которые нужны пользователю. Для этого:

- щелкните по пункту **Edit** (Правка) главного меню системы или нажмите комбинацию клавиш **Alt+E**. Появится выпадающее меню;
- щелкните по пункту **Settings** (Установки) выпадающего меню. Появится диалоговое окно **SETTINGS** для данной модели, в котором можно установить нужные выходные данные. Для нашего примера это окно может выглядеть так, как показано на рис. 4.48.

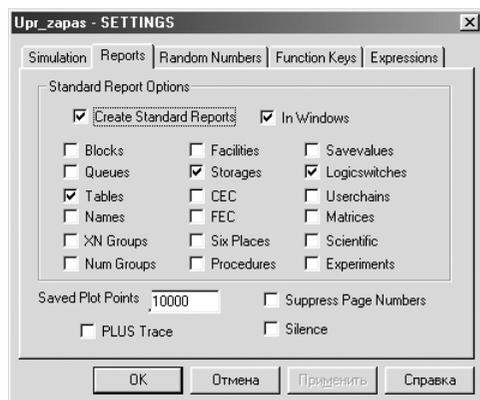


Рис. 4.48. Окно **SETTINGS** с установками для имитационной модели управления запасами

Наличие галочки в окошках говорит о том, что эта информация будет выведена в окне результатов моделирования. В нашем примере будет выведена информация по следующим объектам:

- **Tables** (Таблицы/гистограммы);
- **Storages** (Накопители);
- **Logicswitches** (Логические переключатели).

При таких установках после выполнения моделирования снова появятся окна **JOURNAL** и **REPORT** в урезанном виде, как показано на рис. 4.49.

В верхней строке окна **REPORT** указывается:

- **START TIME** (Начальное время) – 0.000;
- **END TIME** (Время окончания) – 200.000;
- **BLOCKS** (Число блоков) – 16;
- **FACILITIES** (Число каналов обслуживания) – 0;
- **STORAGES** (Число накопителей) – 1.

Ниже указываются результаты моделирования накопителя (**STORAGE**) под назначенным именем **ZAPAS**:

- **CAP.** (Вместимость) – 2000;
- **REM.** – 750;

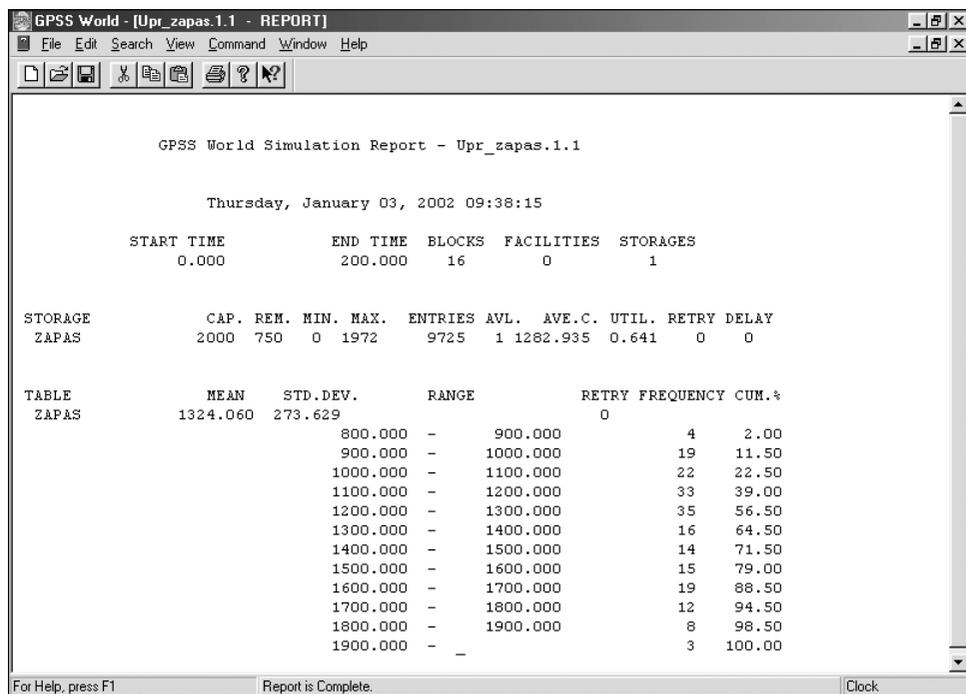


Рис. 4.49. Окно **REPORT** для имитационной модели управления запасами после введения установок

- **MIN.** (Минимальное содержимое) – 0;
- **MAX.** (Максимальное содержимое) – 1972;
- **ENTRIES** (Число входов) – 9725;
- **AVL.** (Доступность) – 1;
- **AVE.C.** (Среднее содержимое) – 1282.935;
- **UTIL.** (Коэффициент использования) – 0.641;
- **RETRY** (Повтор) – 0;
- **DELAY** (Отказ) – 0.

Еще ниже указываются результаты моделирования для построения по табличным данным соответствующей гистограммы функционирования склада по имени ZAPAS:

- **MEAN** (Средняя) – 1324.060;
- **STD.DEV.** (Среднее квадратическое отклонение) – 273.629;
- **RANGE** (Область);
- **RETRY** – 0;
- **FREQUENCY** (Частота);
- **CUM.%** (Суммарный процент).

Построение графиков отдельных параметров функционирования системы

Перед началом моделирования, а точнее после появления окна **JOURNAL**, можно построить графики некоторых параметров функционирования системы. Для этого:

- щелкните по пункту **Window** главного меню системы или нажмите комбинацию клавиш **Alt+W**. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** (Окно моделирования) выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Plot Window** (Окно графика) всплывающего меню. Появится диалоговое окно **Edit Plot Window** (Окно редактирования графика), которое необходимо соответствующим образом заполнить.

Допустим, мы хотим графически представить на всем периоде моделирования, как меняется размер запаса. Для нашей задачи окно **Edit Plot Window** может быть заполнено так, как показано на рис. 4.50.

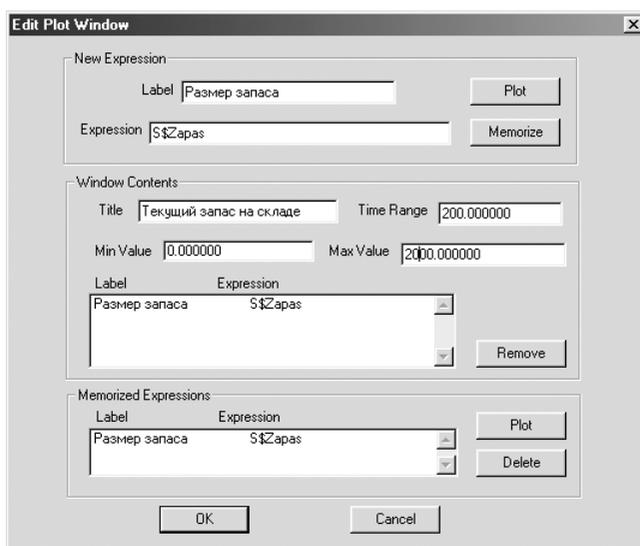


Рис. 4.50. Диалоговое окно **Edit Plot Window** для имитационной модели управления запасами

После заполнения диалогового окна **Edit Plot Window** щелкните по кнопкам **Plot** (График), **Memorize** (Запомнить), а затем – по кнопке **OK**. Появится заготовка графика (рис. 4.51).

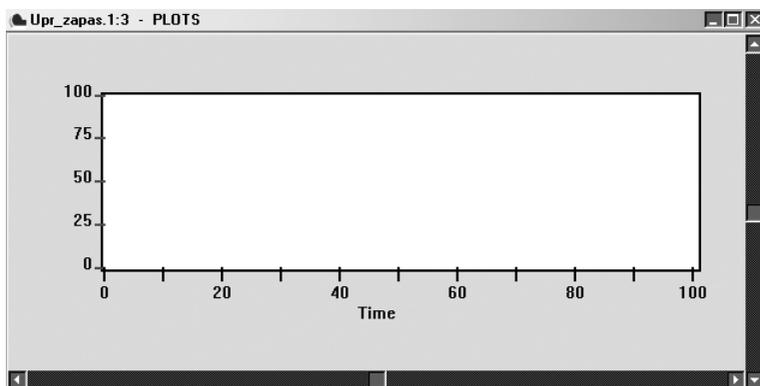


Рис. 4.51. Заготовка графика размера запаса для имитационной модели управления запасами

После этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **START**. Появится диалоговое окно **Start Command**;
- введите в диалоговом окне **Start Command** число 200 – количество дней моделирования работы склада – и щелкните по кнопке **OK**. Появится окно **REPORT** (Отчет) с результатами моделирования. На заднем плане будет размещаться график;
- щелкните по графику, расположенному на заднем плане, – он выйдет на первый план;
- используя горизонтальную и вертикальную полосы прокрутки, вы можете просмотреть построенный график.

Фрагмент графика для нашего примера представлен на рис. 4.52.

При выводе графика на передний план окно **REPORT** с результатами моделирования переместится на задний план. Для просмотра окна **REPORT** щелкните по нему мышью. Оно вновь перейдет на передний план.

Можно также вывести соответствующую гистограмму. Для этого:

- щелкните по пункту **Window** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Table Window** (Окно гистограммы) во всплывающем меню. Появится диалоговое окно **Open Table Window** (Открыть окно гистограммы). В раскрывающемся списке **Table** щелкните по нужной гистограмме. Для нашей задачи это окно выглядит так, как показано на рис. 4.53;

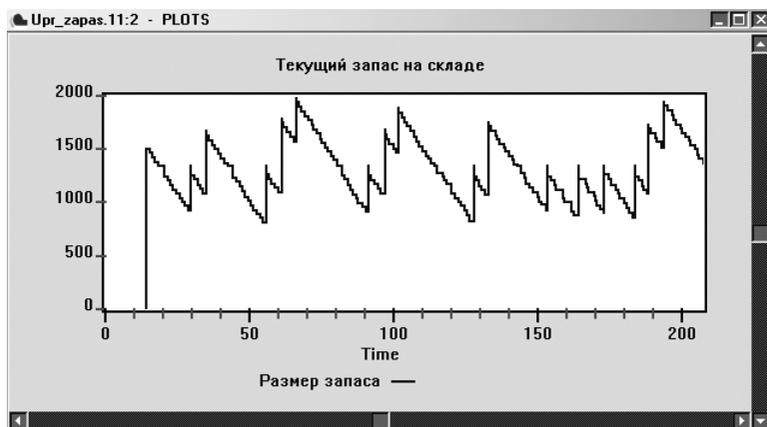


Рис. 4.52. Фрагмент изображения текущего запаса на складе в имитационной модели управления запасами

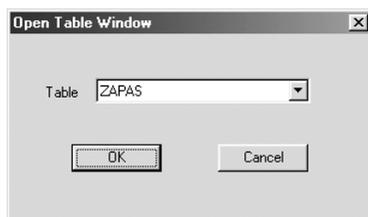


Рис. 4.53. Диалоговое окно выбора нужной гистограммы имитационной модели управления запасами

- щелкните по кнопке **OK**. Появится соответствующая гистограмма. Для нашей задачи она выглядит так, как представлено на рис. 4.54.

Анализ результатов

Сначала давайте воспользуемся окном **Edit Expression Window** (Окно редактирования выражений), чтобы просмотреть две величины сейчас и использовать этот прием в будущем, когда мы пожелаем добавить большее количество переменных для просмотра. Для этого:

- щелкните по пункту **Window** главного меню. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** (Окно моделирования) выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Expressions Window**. Появится диалоговое окно **Edit Expression Window**.

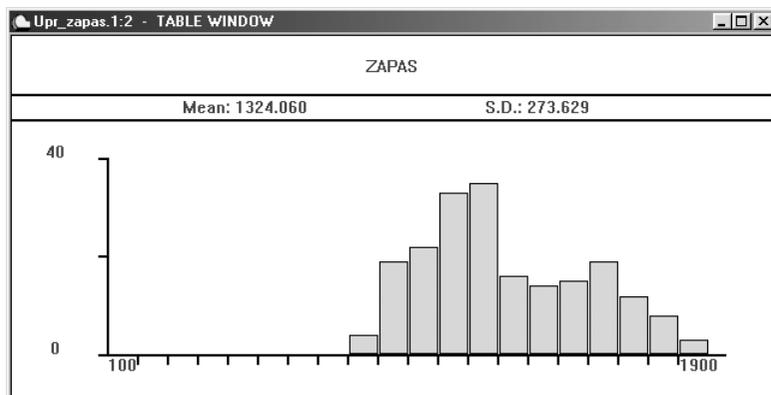


Рис. 4.54. Окно гистограммы имитационной модели управления запасами

В диалоговом окне **Edit Expression Window**:

- щелкните по первому текстовому полю сверху – **Label** (Метка) – и введите следующий текст: «Номер поставки»;
- щелкните по второму текстовому полю сверху – **Expression** – и введите стандартный числовой атрибут $N\$Out$;
- щелкните по кнопкам **View** (Просмотр) и **Memorize** (Запомнить).

Если мы запоминаем введенные выражения и их назначение, то при закрытии окна и повторном его открытии эта информация вновь появляется.

Теперь давайте добавим текущий уровень запасов. В диалоговом окне **Edit Expression Window**:

- в первом текстовом поле введите новую запись – «Текущая вместимость», а во втором – $S\$Zapas$;
- щелкните по кнопкам **View** и **Memorize**.

Затем:

- введите в первом текстовом поле следующий текст: «Максимальное содержание склада», а во втором – стандартный числовой атрибут $SM\$Zapas$;
- щелкните по кнопкам **View** и **Memorize**.

Наконец, давайте введем номер дня:

- в первом текстовом поле введите надпись «День», а во втором – $AC1$;
- щелкните по кнопкам **View** и **Memorize**;
- щелкните по кнопке **OK**. Появится окно **EXPRESSIONS** с искомыми значениями, показанное на рис. 4.55.

После перемещения и изменения размеров окна **TABLE WINDOW** и **EXPRESSIONS** могут выглядеть так, как показано на рис. 4.56.

Теперь начнем процесс моделирования с новым, общим числом дней моделирования. Например, пусть длительность моделирования составит 100 дней. При этом не будем выводить окно **REPORT** с результатами моделирования.

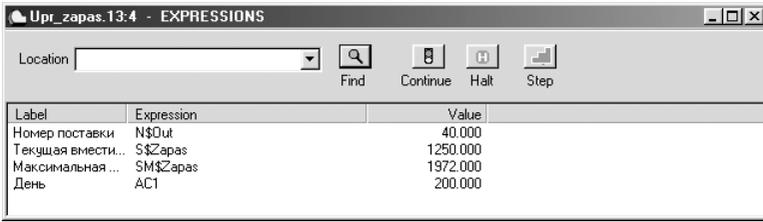


Рис. 4.55. Окно **EXPRESSIONS** со значениями искомых выражений задачи управления запасами

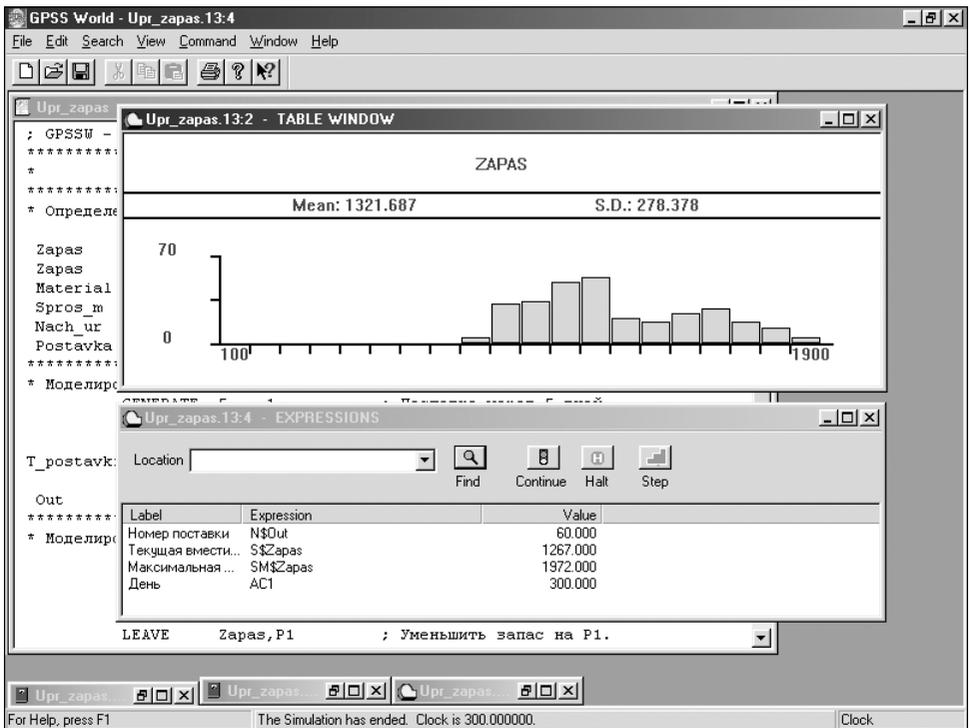


Рис. 4.56. Окна **EXPRESSIONS** и **TABLE WINDOW** для задачи управления запасами

Выполните следующие действия:

- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **START**. Появится диалоговое окно **Start Command**. В этом диалоговом окне вместо 1 введите следующее: 100, NP;
- щелкните по кнопке **OK**.

Обратите внимание, как изменяются гистограмма в окне **TABLE WINDOW** и искомые значения в окне **EXPRESSIONS** в процессе моделирования.

Процесс функционирования данной системы управления запасами весьма прост. Сначала в системе активизируются требования, имеющие наибольший приоритет. В нашей задаче таким является одно требование в четвертом секторе, имеющее приоритет, равный 10. Это наибольший приоритет для всех требований, генерируемых оператором **GENERATE**. Поэтому данное требование становится первым активным требованием в системе. Оно входит в накопитель **STORAGE** под именем *Zapas* с вместимостью 2000 единиц. Во время входа в накопитель в нем устанавливается начальный уровень запаса, величина которого определяется с помощью оператора **EQU**. В нашей задаче он равен 1500 единицам.

Далее активизируются требования (недели), генерируемые оператором **GENERATE** во втором секторе, так как на данный момент они имеют наибольший приоритет, равный 1. В этом секторе определяется время поставки материалов на склад. Если нет поставки, то активное требование второго сектора задерживается в операторе **ADVANCE** на пять дней. В это время активизируются требования, генерируемые оператором **GENERATE** в третьем секторе, которые поступают в систему каждый день. Они имеют нулевой приоритет – самый низкий в системе.

Закройте окна **TABLE WINDOW** и **EXPRESSIONS**, щелкнув по кнопкам с крестиком, расположенным в правом верхнем углу окон.

Теперь откройте окно **BLOCK ENTITIES** в детальном виде. Для этого:

- щелкните по пункту **Window** главного меню. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** (Окно моделирования) выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Blocks Window**. Появится окно **BLOCK ENTITIES** в детальном виде (рис. 4.57).

Теперь можно визуализировать процесс моделирования. Для этого:

- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **Clear** (Очистить). Все значения в окне **BLOCK ENTITIES** обнулятся;
- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **START**. Появится диалоговое окно **Start Command**. Введите в нем такие данные: 100, NP;
- щелкните по кнопке **OK**.

Начнется перемещение требований по блокам системы с соответствующим изменением выводимых в окне величин. Наблюдайте действие в различных частях моделирования. Если модель достаточно большая, то для ее просмотра используйте полосы прокрутки, расположенные в правой части и/или внизу окна. Чтобы просмотреть программу, перемещайте бегунок вдоль полосы прокрутки. Это выполняется точно так же, как в текстовых редакторах.

Loc	Block Type	Current Count	Entry Count	Retry Chain	Line Number	Include-file
1 GEN	GENERATE	0	40	0	15	0
2 TES	TEST	0	40	0	16	0
3 ASN	ASSIGN	0	17	0	17	0
4 ADV	ADVANCE	0	17	0	18	0
5 ENT	ENTER	0	17	0	19	0
6 OUT	TERMINATE	0	40	0	20	0
7 GEN	GENERATE	0	200	0	23	0
8 ASN	ASSIGN	0	200	0	24	0
9 TAB	TABULATE	0	200	0	25	0
10 TES	TEST	0	200	0	26	0
11 LEA	LEAVE	0	200	0	27	0
12 TER	TERMINATE	0	200	0	28	0
ZAPA...	TERMINATE	0	0	0	29	0
14 GEN	GENERATE	0	1	0	32	0
15 ENT	ENTER	0	1	0	33	0
16 TER	TERMINATE	0	1	0	34	0

Рис. 4.57. Окно **BLOCK ENTITIES** в детальном виде для модели управления запасами

Перемещение бегунка можно выполнить тремя способами:

- щелкните по стрелкам прокрутки, расположенным сверху (слева) и внизу (справа) на вертикальной (горизонтальной) полосе прокрутки;
- щелкните по полосе прокрутки под бегунком или над ним (по горизонтальной полосе прокрутки щелкните соответственно слева и справа);
- наведите курсор мыши на бегунок, нажмите левую кнопку мыши и, удерживая ее, перемещайте курсор, а вместе с ним и бегунок.

Теперь попробуем изменить некоторые входные данные в первом секторе программы. Для этого:

- щелкните по пункту **Window** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **Simulation Window** выпадающего меню. Появится всплывающее меню;
- щелкните по пункту **Table Window** (Окно гистограммы) во всплывающем меню. Появится диалоговое окно **Open Table Window** (Открыть окно гистограммы), в текстовом поле которого будет имя Zapas;
- щелкните по кнопке **OK**. Появится соответствующая гистограмма;
- щелкните по пункту **Command** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **Custom** (Заказать) выпадающего меню. Появится диалоговое окно **Simulation Command** (Команда моделирования), которое выглядит так, как показано на рис. 4.58;

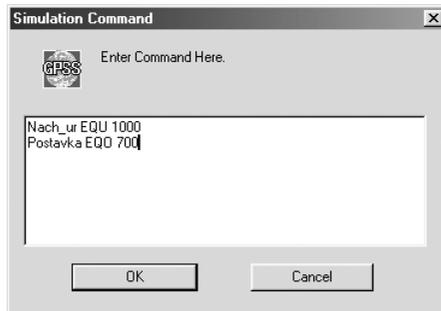


Рис. 4.58. Диалоговое окно **Simulation Command** для ввода измененных данных в модели управления запасами

- введите две строки:

```
Nach_ur EQU 1000
Postavka EQO 700
```

- щелкните по кнопке **OK**;
- щелкните по пункту **Command** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **Clear** выпадающего меню;
- щелкните по пункту **Command** главного меню. Появится выпадающее меню;
- щелкните по пункту **START**. Появится диалоговое окно **Start Command**. Введите в нем такие данные: 100, NP;
- щелкните по кнопке **OK**. Появится новая гистограмма после моделирования. Она выглядит так, как показано на рис. 4.59.

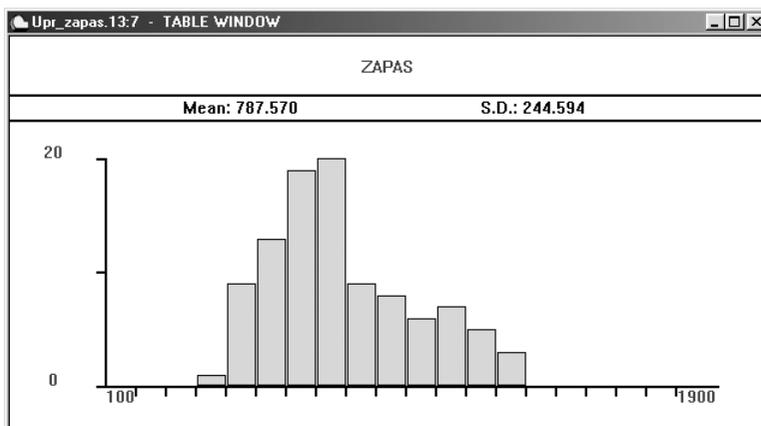


Рис. 4.59. Окно гистограммы имитационной модели управления запасами

Оценка эффективности работы системы с учетом отказов

Постановка задачи

Рассмотрим функционирование системы, имеющей три вида объектов (машины, приборы, станки, компьютеры и т.д.). Объекты поступают в систему на обслуживание с интенсивностью λ . Поток поступления объектов на обслуживание – пуассоновский: $\lambda = 0,1$ 1/единица времени. Объекты обслуживаются в канале обслуживания с интенсивностью μ . Распределение времени обслуживания объектов – экспоненциальное: $\mu = 0,05$ 1/единица времени. Одновременно все объекты находятся под действием пуассоновских потоков отказов с соответствующими интенсивностями: $\lambda_1 = 0,01$ 1/единица времени, $\lambda_2 = 0,008$ 1/единица времени, $\lambda_3 = 0,0125$ 1/единица времени. Отказавшие объекты немедленно начинают ремонтироваться (восстанавливаться). При этом для восстановления отказавших объектов имеется два пункта восстановления (ПВ). Распределения времени восстановления объектов на каждом ПВ будем предполагать экспоненциальным. Для первого ПВ интенсивность восстановления $\mu_1 = 0,033$ 1/единица времени. Для второго ПВ интенсивность восстановления $\mu_2 = 0,025$ 1/единица времени. Требуется оценить эффективность работы такой системы, то есть определить:

- коэффициенты использования (готовности) всех объектов;
- среднее время восстановления объектов;
- коэффициенты использования пунктов восстановления и др.

Выявление основных особенностей

Рассматриваемые в системе потоки отказов наиболее полно отвечают реалиям жизни и характеризуются следующими особенностями:

- отказы объекта происходят по одному, то есть вероятность отказа двух и более элементов объекта в один момент времени очень мала, и ею можно пренебречь (поток требований ординарный);
- вероятность наступления последующих отказов объекта в любой момент времени не зависит от предыдущих отказов – поток отказов без последствия;
- поток отказов стационарный.

Потоки отказов, в которых имеются все вышеперечисленные особенности, называются простейшими.

Интенсивность отказа объекта определяется как величина, обратная времени наступления очередного отказа объекта, – t_p :

$$\lambda = 1/t_p.$$

Интенсивность восстановления объекта определяется как величина, обратная времени восстановления (ремонта) объекта, $-t_o$:

$$\mu = 1/t_o.$$

Создание имитационной модели

Создание имитационной модели начнем с построения заголовка модели, который может быть представлен, например, в таком виде:

```
* GPSSW           Nadejn_mn_sist.GPS
*****
*      Оценка надежности работы      *
*      многоканальной системы        *
*****
```

Модель создадим из трех сегментов. В первом сегменте генерируется поток отказов различных объектов. Во втором сегменте проводится обслуживание не отказавших объектов. В третьем проводится восстановление (ремонт) отказавших объектов.

Первый сегмент состоит из трех наборов операторов – по числу видов объектов. Для первого объекта этот набор операторов выглядит так:

```
GENERATE (Exponential(1,0,100)),,1
PREEMPT Object1,PR,MET4,,RE
ASSIGN 2,Object1
TRANSFER ,COM2
```

Первый оператор набора **GENERATE** (Генерировать) генерирует отказы первого объекта. Поскольку поток отказов всех объектов пуассоновский, то их можно генерировать с помощью встроенной функции экспоненциального распределения отказов. В поле операнда A используется встроенная в систему функция экспоненциального распределения (Exponential...). Третий параметр в экспоненциальной функции – время наступления очередного отказа t_p .

Для первого прибора этот параметр будет равен:

$$t_{p1} = 1/\lambda_1 = 1/0,01 = 100.$$

Для второго прибора этот параметр будет равен:

$$t_{p2} = 1/\lambda_2 = 1/0,008 = 125.$$

Для третьего прибора этот параметр будет равен:

$$t_{p3} = 1/\lambda_3 = 1/0,0125 = 80.$$

В поле операнда D стоит значение 1, которое определяет величину приоритета данного события – отказа объекта. По умолчанию приоритет равен 0.

Второй оператор набора **PREEMPT** (Занять) снимает объект (требование) с обслуживания в связи с наступлением отказа и перемещает его в другое место. В поле операнда А указывается имя канала обслуживания, в котором прерывается обслуживание. В поле операнда С указывается местоположение блока, к которому направляется прерванное в обслуживании требование. В нашем примере это блок с меткой MET4. Объекты, обслуживание которых прерывается, выходят из системы обслуживания. В поле операнда В указываются символы PR – режим учета приоритета.

В поле операнда D определяется параметр, в который заносится оставшееся время обслуживания. В поле операнда E могут быть символы RE. Это означает, что требование направляется в блок, указанный в поле операнда С, с правом на повторное использование канала обслуживания.

Далее используется оператор **ASSIGN** (Присвоить) для присвоения параметру требования под номером 2 имени объекта. Заключительным оператором набора является оператор **TRANSFER** (Передать), используемый в режиме безусловной передачи требования к оператору с меткой COM2 (поле операнда В). Такие наборы операторов создаются для каждого объекта, а в нашей задаче их три.

Второй сегмент нашей модели обеспечивает обслуживание не отказавших объектов. Он включает оператор **GENERATE**, который генерирует поступление объектов на обслуживание:

```
GENERATE (Exponential(1,0,10))
```

Поток поступления объектов на обслуживание – пуассоновский ($\lambda = 0,1$ 1/единица времени), и его можно генерировать с помощью встроенной функции экспоненциального распределения. Третий параметр в экспоненциальной функции – среднее время между входами двух идущих один за другим объектов в систему на обслуживание. Оно будет равно:

$$t_0 = 1/\lambda = 1/0,1 = 10.$$

Далее используются три набора операторов – по числу видов объектов. Для первого объекта этот набор операторов выглядит так:

```
MET1  GATE NU Object1,MET2
      SEIZE  Object1
      ASSIGN 1,Object1
      TRANSFER ,COM1
```

Первый оператор набора **GATE NU** обеспечивает проверку канала обслуживания на занятость с передачей объекта в случае занятости канала на следующий канал либо (при занятости всех каналов обслуживания) на уничтожение. В поле операнда А указывается числовое или символьное имя объекта. В зависимости от состояния объекта выполняется его перемещение к оператору с меткой, указываемой в поле операнда В, в нашем примере это оператор с меткой MET2.

Если канал обслуживания не занят, объект перемещается к следующему оператору – **SEIZE** (Занять), – который определяет занятость канала обслуживания. Если канал обслуживания занят, то объект далее не перемещается. Если канал обслуживания не занят, то объект переходит к следующему оператору – **ASSIGN** – для присвоения параметру требования под номером 1 имени объекта. Заключительным оператором набора является **TRANSFER**, используемый в режиме безусловной передачи требования к оператору с меткой COM1.

Далее используется **ADVANCE** для моделирования времени обслуживания объектов. Среднее время обслуживания объекта определяется, исходя из интенсивности обслуживания объектов – $\mu = 0,05$ 1/единица времени. Распределение времени обслуживания объектов – экспоненциальное. Среднее время обслуживания определяется так:

$$t_{obsl} = 1/\mu = 1/0,05 = 20.$$

Моделирование обслуживания объектов будет выглядеть так:

```
COM1 ADVANCE (Exponential(1,0,20))
```

После обслуживания объекта надо сообщить об освобождении канала обслуживания. Это делается с помощью оператора **RELEASE** (Освободить), который в нашем примере записывается так:

```
RELEASE P1
```

В поле операнда A в параметре требования под номером 1 содержится имя объекта, покинувшего канал обслуживания в данный момент.

Далее используется ряд операторов, обеспечивающих вывод объектов из системы обслуживания и определение числа теряемых объектов. Они в нашем примере будут выглядеть так:

```
TERMINATE 1
MET4 SAVEVALUE 2+,1
TERMINATE 1
```

Число теряемых объектов содержится в сохраняемой величине под номером 2. И наконец, третий сегмент модели будет проводить восстановление (ремонт) отказавших объектов. Это будет выглядеть так:

```
COM2 TRANSFER BOTH,REM1,REM2
REM1 QUEUE Punkt_1
ADVANCE (Exponential(1,0,30))
DEPART Punkt_1
TRANSFER ,COM3
REM2 QUEUE Punkt_2
ADVANCE (Exponential(1,0,40))
DEPART Punkt_2
```

```

COM3  RETURN      P2
      SAVEVALUE   1+,M1
      TERMINATE
      START       5000

```

Первый оператор третьего сегмента модели **TRANSFER** распределяет объекты по двум пунктам восстановления, используя режим BOTH (Оба). Сначала объект направляется к оператору с меткой REM1, а если пункт восстановления занят – то к оператору с меткой REM2.

Далее используются операторы **QUEUE** и **DEPART** для сбора статистической информации по очередям и операторы **SEIZE** и **RELEASE** для сбора статистической информации по каналам восстановления с использованием соответствующих имен в поле операнда A.

Для моделирования времени восстановления объекта используются операторы **ADVANCE** (Задержать), которые в нашей задаче будут выглядеть так:

```

ADVANCE (Exponential(1,0,30))
ADVANCE (Exponential(1,0,40))

```

В поле операнда A выполняется обращение к встроенному в систему экспоненциальному распределению с указанием среднего интервала восстановления объектов – соответственно 30 и 40 единиц времени.

Далее используются операторы **RETURN** и **SAVEVALUE**:

```

COM3  RETURN      P2
      SAVEVALUE   1+,M1

```

Первый из них обеспечивает прерывание обслуживания объекта вследствие отказа и ремонта. Второй обеспечивает сбор суммарного времени прохождения отказавших объектов в сохраняемой величине под номером 1.

Моделирование заканчивается оператором **TERMINATE** (уничтожая отказавшие объекты) и начинается оператором **START**. Окончательно наша программа будет выглядеть так, как представлено на рис. 4.60а,б.

Моделирование системы

Для начала моделирования системы выполните следующие действия:

- щелкните по пункту **File** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **New** (Создать) выпадающего меню. Появится диалоговое окно **Новый документ**;
- выделите пункт **Model** и щелкните по кнопке **ОК**. Появится окно модели, в котором введите данную программу.

а)

```

Nadejn_n_sist
* GPSSW      Nadejn_n_sist.GPS
*****
* Оценка надежности работы *
* многоканальной системы *
*****
GENERATE (Exponential (1,0,100) ) , , , 1
PREEMPT Object1,PR,MET4,,RE
ASSIGN 2, Object1
TRANSFER ,COM2
GENERATE (Exponential (1,0,125) ) , , , 1
PREEMPT Object2,PR,MET4,,RE
ASSIGN 2, Object2
TRANSFER ,COM2
GENERATE (Exponential (1,0,80) ) , , , 1
PREEMPT Object3,PR,MET4,,RE
ASSIGN 2, Object3
TRANSFER ,COM2
*****
GENERATE (Exponential (1,0,10) )
MET1 GATE NU Object1,MET2
SEIZE Object1
ASSIGN 1, Object1
TRANSFER ,COM1
MET2 GATE NU Object2,MET3
SEIZE Object2
ASSIGN 1, Object2
TRANSFER ,COM1

```

б)

```

Nadejn_n_sist
MET3 GATE NU Object3,MET4
SEIZE Object3
ASSIGN 1, Object3
COM1 ADVANCE (Exponential (1,0,20) )
RELEASE P1
TERMINATE 1
MET4 SAVEVALUE 2+, 1
TERMINATE 1
*****
COM2 TRANSFER BOTH,REM1,REM2
REM1 QUEUE Punkt_1
SEIZE Punkt_1
DEPART Punkt_1
ADVANCE (Exponential (1,0,30) )
RELEASE Punkt_1
TRANSFER ,COM3
REM2 QUEUE Punkt_2
SEIZE Punkt_2
DEPART Punkt_2
ADVANCE (Exponential (1,0,40) )
RELEASE Punkt_2
COM3 RETURN P2
SAVEVALUE 1+,M1
TERMINATE 5000
START 5000
*****

```

Рис. 4.60. Фрагменты имитационной модели оценки эффективности работы системы

Чтобы вызвать окно для представления имитационной модели в системе GPSSW, можно также нажать комбинацию клавиш **Ctrl+Alt+S**.

После создания имитационную модель необходимо оттранслировать и запустить на выполнение. Для этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню.

В имитационной модели имеется управляющая команда **START**, следовательно, исходная имитационная модель будет транслироваться, и если в ней нет ошибок, то начнется процесс моделирования системы.

Подготовка к моделированию

Перед началом моделирования можно установить вывод тех параметров моделирования, которые нужны пользователю. Для этого:

- щелкните по пункту **Edit** (Правка) главного меню системы или нажмите комбинацию клавиш **Alt+E**. Появится выпадающее меню;
- щелкните по пункту **Settings** (Установки) выпадающего меню. Появится диалоговое окно **SETTINGS** для данной модели;
- щелкните по вкладке **Reports** (Отчеты), в которой можно установить нужные выходные данные. Для нашего примера это может выглядеть так, как показано на рис. 4.61.

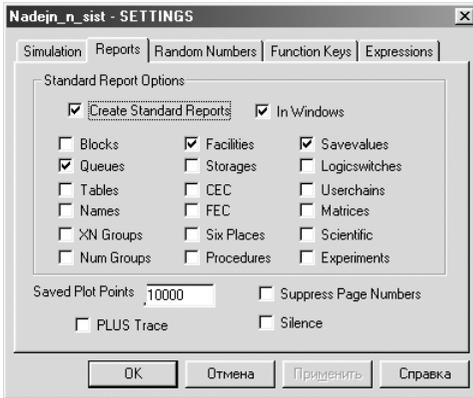


Рис. 4.61. Окно **SETTINGS** с установками для имитационной модели оценки эффективности работы системы

Наличие галочки в окошках говорит о том, что эта информация будет выведена в окне результатов моделирования. В нашем примере будет выведена информация по следующим объектам:

- **Queues** (Очереди);
- **Facilities** (Каналы обслуживания);
- **Savevalues** (Сохраняемые величины).

Результаты моделирования представлены в окне **REPORT**, показанном на рис. 4.62.

GPSS World Simulation Report - Nadejn_n_sist.1.1									
Sunday, December 16, 2001 20:09:18									
START TIME	END TIME	BLOCKS	FACILITIES	STORAGES					
0.000	50482.087	44	4	0					
FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
OBJECT3	959	0.380	19.978	1	5003	0	0	0	0
PUNKT_1	3	0.002	28.380	1	0	0	0	0	0
OBJECT1	1661	0.662	20.135	1	5005	0	0	0	0
OBJECT2	1309	0.534	20.577	1	5001	0	0	0	0
QUEUE	MAX CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY		
PUNKT_1	1	0	3	2	0.000	0.593	1.778	0	
SAVEVALUE	RETRY	VALUE							
1	0	159.205							
2	0	1077.000							

Рис. 4.62. Окно **REPORT** с результатами моделирования имитационной модели оценки эффективности работы системы

Из результатов моделирования видно, что за время моделирования, равное 50482,087 единицам времени, в систему поступило 5005 объектов. Из них потеряно вследствие занятости объекта или отказа 1077 (см. сохраняемую величину под номером 2). Средняя занятость объектов ОБЪЕСТ1, ОБЪЕСТ2 и ОБЪЕСТ3 соответственно равна 0,662, 0,534 и 0,380, а средняя занятость пункта восстановления Punkt_1 составляет 0,002. Максимальная длина очереди на восстановление объектов в пункте 1 составляет один объект. Вся эта информация достаточно полно иллюстрирует эффективность работы системы.

Моделирование систем массового обслуживания

Классификация систем массового обслуживания ...	238
Анализ работы разомкнутых СМО	240
Анализ работы замкнутых СМО	274
Анализ работы многофазных СМО	300

Систему массового обслуживания в общем виде можно представить как совокупность последовательно связанных между собой входящих потоков требований на обслуживание (машин, самолетов, пользователей и т.д.), очередей, каналов обслуживания (станции техобслуживания, аэродрома, ЭВМ и т.д.) и выходящих потоков требований после обслуживания.

Классификация систем массового обслуживания

Системы массового обслуживания по наличию того или иного признака можно классифицировать таким образом:

1. *По характеру поступления требований в систему:* на системы с регулярным и случайным потоками поступления требований в систему.

Если количество требований, поступающих в систему в единицу времени (интенсивность потока), постоянно или является заданной функцией времени, то мы имеем систему с регулярным потоком поступления требований в систему, иначе – со случайным.

Случайный поток требований в систему подразделяется на *стационарный* и *нестационарный*.

Если параметры, потока требований не зависят от расположения рассматриваемого интервала времени на оси времени, то мы имеем стационарный поток требований, в противном случае – нестационарный. Например, если число покупателей, приходящих в магазин, не зависит от времени суток, то поток требований (покупателей) – стационарный.

2. *По количеству требований, поступающих в единицу времени:* на системы с ординарным и неординарным потоками требований. Если вероятность поступления двух или более требований в один момент времени равна нулю или имеет столь малую величину, что ею можно пренебречь, то мы имеем систему с ординарным потоком требований. Например, поток требований – самолетов, поступающих на взлетно-посадочную полосу аэродрома (ВПП), – можно считать ординарным, так как вероятность поступления двух и более самолетов в канал обслуживания (ВПП) в один и тот же момент времени очень мала, и ею можно пренебречь.

3. *По связи между требованиями:* на системы без последствия от поступивших требований и с последствием. Если вероятность поступления требований в систему в некоторый момент времени не зависит от того, сколько требований уже поступило в систему, то есть не зависит от предыстории изучаемого процесса, то мы имеем задачу без последствия, в противном случае – с последствием. Примером задачи с последствием может служить поток студентов на сдачу зачета преподавателю.

4. *По характеру поведения требования в системе:* с отказами, с ограниченным ожиданием и с ожиданием без ограничения:

- если требование, вновь поступившее на обслуживание, застает все каналы обслуживания уже занятыми и покидает систему, то мы имеем систему с отказами. Требование может покинуть систему и в том случае, когда очередь достигла определенных размеров. Если ракета противника появляется во время, когда все противоракетные установки заняты обслуживанием других ракет, то она благополучно покидает область обслуживания;

- если поступившее требование застает все каналы обслуживания занятыми и становится в очередь, но находится в ней ограниченное время, после

чего, не дождавшись обслуживания, покидает систему, то имеем систему с ограниченным ожиданием. Примером такого «нетерпеливого» требования может быть автосамосвал с цементным раствором. Если время ожидания велико, то во избежание затвердения раствора он может быть разгружен в другом месте;

- если поступившее требование, застав все каналы обслуживания занятыми, вынуждено ожидать своей очереди до тех пор, пока оно не будет обслужено, то мы имеем систему с ожиданием без ограничения. Пример: самолет, который находится на аэродроме до тех пор, пока не освободится взлетная полоса.

5. *По способу выбора требований на обслуживание*: с приоритетом, по мере поступления, случайно, последний обслуживается первым. Иногда в таком случае говорят о дисциплине обслуживания:

- если система массового обслуживания охватывает несколько категорий требований и по каким-либо соображениям необходимо соблюдать различный подход к их отбору, то имеем систему с приоритетом. Так, при поступлении изделий на стройплощадку в первую очередь монтируются те, которые необходимы в данный момент;
- если освободившийся канал обслуживает требование, ранее других поступившее в систему, то имеем систему с обслуживанием требований по мере их поступления. Это наиболее распространенный класс систем. Например, покупатель, подошедший первым к продавцу, обслуживается раньше других. Этот способ выбора требований на обслуживание применяется там, где в силу технических, технологических или организационных условий требования не могут опережать друг друга;
- если требования из очереди поступают в канал обслуживания в случайном порядке, то имеем систему со случайным выбором требований на обслуживание. Пример: выбор слесарем-сантехником одной из нескольких заявок, поступивших от жильцов, на устранение неисправностей. Выбор здесь, как правило, определяется местоположением самого слесаря: он предпочтет заявку жильца, расположенного к нему наиболее близко, если никакие другие факторы не определяют иной выбор;
- последний обслуживается первым. Этот способ выбора требований на обслуживание используется в тех случаях, когда удобнее или экономнее брать на обслуживание требование, позже всех поступившее в систему. Так, при складировании строительных изделий в штабель удобнее сначала брать из него изделие, поступившее последним.

6. *По характеру обслуживания требований*: на системы с детерминированным и случайным временем обслуживания. Если интервал времени между моментом поступления требования в канал обслуживания и моментом выхода требования из этого канала постоянен, то мы имеем систему с детерминированным временем обслуживания, иначе – со случайным.

7. *По числу каналов обслуживания*: на одноканальные и многоканальные системы. Так, при монтаже дома может быть использован один подъемный кран (один канал обслуживания) или несколько (много каналов) для обслуживания прибывающих на стройку изделий.

8. *По количеству этапов обслуживания*: на однофазные и многофазные системы. Если каналы обслуживания расположены последовательно и они неоднородны, так как выполняют различные операции обслуживания, то мы имеем многофазную систему массового обслуживания. Примером такой системы может быть, например, обслуживание автомобилей на станции технического обслуживания (мойка, диагностирование и т.д.).
9. *По однородности требований*, поступающих на обслуживание: на системы с однородными и неоднородными потоками требований. Так, если под погрузку прибывают фуры одинаковой грузоподъемности, то такие требования называются однородными, если разной – то неоднородными.
10. *По ограниченности потока требований*: на замкнутые и разомкнутые системы. Если поток требований ограничен и требования, покинувшие систему, через некоторое время в нее возвращаются, то мы имеем замкнутую систему, в противном случае – разомкнутую. Примером замкнутой системы может служить система «Ремонтная бригада» и обслуживаемое ею оборудование.

Если изучены или заданы входящие потоки требований, механизм (число каналов, продолжительность обслуживания и т.д.) и дисциплина обслуживания – это дает основание для построения модели системы.

Анализ работы разомкнутых СМО

Одноканальная разомкнутая СМО с простейшими потоками

Постановка задачи

Рассмотрим сначала одноканальную разомкнутую систему массового обслуживания с неограниченным временем ожидания требований и с простейшими потоками. Простейший поток наиболее полно отвечает реалиям жизни и характеризуется следующими особенностями:

- поступление требований в систему на обслуживание происходит по одному, то есть вероятность прибытия двух и более требований в один момент времени очень мала, и ею можно пренебречь (поток требований ординарный);
- вероятность поступления последующих требований в любой момент времени не зависит от возможности их прибытия в предыдущие моменты – поток требований без последствия;
- поток требований стационарный.

Требуется определить:

- коэффициент использования канала обслуживания;
- среднюю длину очереди, то есть среднее число машин, находящихся в очереди, ожидая освобождения канала обслуживания;
- среднее число требований, находящихся в системе, то есть в очереди и в канале обслуживания.

Выявление основных особенностей

Функционирование любой системы массового обслуживания можно представить через все ее возможные состояния, а также через интенсивность перехода из одного состояния в другое. Основными параметрами функционирования системы массового обслуживания являются вероятности ее состояния, то есть возможности наличия n требований (покупателей, рабочих, заданий, машин, неполадок) в системе – P_n . Так, вероятность P_0 характеризует состояние, когда в системе нет требований и канал обслуживания простаивает.

Важными параметрами функционирования системы массового обслуживания являются также среднее число требований, находящихся в системе, то есть в очереди и на обслуживании, – N_{syst} – и средняя длина очереди – N_{och} . Исходными параметрами, характеризующими систему массового обслуживания, являются:

- число каналов обслуживания (касс, компьютеров, подъемных кранов, ремонтных бригад) – N ;
- число требований (покупателей, заданий, машин, неполадок) – m ;
- интенсивность поступления одного требования на обслуживание, то есть число поступлений требований в единицу времени – λ ;
- интенсивность обслуживания требований – μ .

Интенсивность поступления требования на обслуживание определяется как величина, обратная среднему времени между поступлениями двух смежных требований, – t_p :

$$\lambda = 1/t_p.$$

Интенсивность обслуживания требований определяется как величина, обратная времени обслуживания одного требования, – t_o :

$$\mu = 1/t_o.$$

Рассмотрим сначала решение задачи аналитическим методом.

Решение задачи аналитическим методом

Состояние системы массового обслуживания будем связывать с числом требований, находящихся в системе:

- в системе нет ни одного требования – вероятность состояния P_0 ;
- в системе находится одно требование – вероятность состояния P_1 ;
- в системе находится n требований – вероятность состояния P_n .

Представим все возможные состояния системы массового обслуживания в виде размеченного графа состояний (рис. 5.1). Каждый прямоугольник графа, количественно оцениваемый вероятностью состояний P_n , определяет одно из всех возможных состояний. Стрелки указывают, в какое состояние система может перейти и с какой интенсивностью.

Первый прямоугольник с вероятностью P_0 определяет состояние системы массового обслуживания, при котором канал обслуживания простаивает из-за отсутствия

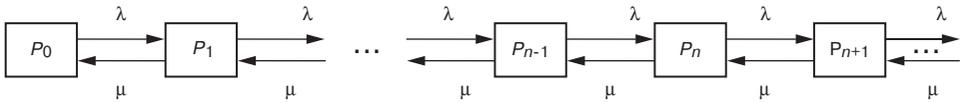


Рис. 5.1. Размеченный граф состояний
одноканальной разомкнутой системы массового обслуживания

требований в системе. Из этого положения система массового обслуживания может перейти только в состояние P_1 . Это означает, что в системе появится одно требование, так как входной поток ординарный. С интенсивностью μ система может перейти также из состояния P_1 в состояние P_0 . Это значит, что единственное находящееся в системе требование было обслужено раньше, чем появилось новое.

Сначала рассмотрим установившийся режим работы системы массового обслуживания, когда основные вероятностные характеристики СМО постоянны во времени, например в течение часа. Тогда интенсивности входных и выходных потоков для каждого состояния будут сбалансированы. Эти сбалансированные потоки могут выглядеть так:

$$\begin{aligned}
 P_0 \times \lambda &= P_1 \times \mu; \\
 P_1 \times (\mu + \lambda) &= P_0 \times \lambda + P_2 \times \mu; \\
 P_2 \times (\mu + \lambda) &= P_1 \times \lambda + P_3 \times \mu; \\
 &\dots \\
 P_n \times (\mu + \lambda) &= P_{n-1} \times \lambda + P_{n+1} \times \mu; \\
 &\dots
 \end{aligned}$$

Обозначим величину λ / μ через ψ и назовем ее коэффициентом загрузки. Из первого уравнения можно найти значение P_1 :

$$P_1 = P_0 \times \lambda / \mu = P_0 \times \psi.$$

Из второго уравнения найдем значение P_2 :

$$P_2 = P_1 + P_1 \times \lambda / \mu - P_0 \times \lambda / \mu.$$

Но первый член

$$P_1 = P_0 \times \lambda / \mu.$$

Следовательно, первый и третий сокращаются:

$$P_2 = P_1 \times \lambda / \mu = P_0 \times \psi^2.$$

Из третьего уравнения найдем значение P_3 :

$$P_3 = P_2 + P_2 \times \lambda / \mu - P_1 \times \lambda / \mu.$$

Но первый член

$$P_2 = P_1 \times \lambda / \mu.$$

Следовательно, первый и третий также сокращаются:

$$P_3 = P_2 \times \lambda / \mu = P \times \psi^3$$

и т.д.:

$$P_n = P_{n-1} \times \lambda / \mu = P_0 \times \psi^n.$$

Используя очевидное равенство

$$\sum_{n=0}^{\infty} P_n = 1,$$

получим:

$$1 = P_0 \times \sum_{n=0}^{\infty} \psi^n.$$

Так как ψ меньше 1, то сумма геометрически убывающей прогрессии равна

$$1 + \psi + \psi^2 + \psi^3 + \dots + \psi^n + \dots = \frac{1 - \psi^{n-1}}{1 - \psi}.$$

При $n \rightarrow \infty$ ($\psi < 1$)

$$\sum_{n=0}^{\infty} P_n = P_0 \frac{1}{1 - \psi} = 1.$$

Отсюда вероятность простоя канала обслуживания определяется так:

$$P_0 = 1 - \psi.$$

Вероятность того, что в системе находится n требований, определяется по формуле

$$P_n = \psi^n \times P_0 = \psi^n \times (1 - \psi).$$

Среднее число обслуживаемых требований N_{syst} , находящихся в системе, может быть определено таким образом:

$$N_{syst} = \sum_{n=0}^{\infty} n \times P_n = \sum_{n=0}^{\infty} n \times \psi^n \times (1 - \psi) = (1 - \psi) \sum_{n=0}^{\infty} n \times \psi^n;$$

$$\begin{aligned} \sum_{n=0}^{\infty} n \times \psi^n &= (1 - \psi) \times (\psi + 2\psi^2 + 3\psi^3 + \dots + n\psi^n + \dots) = \\ &= \psi \times (1 - \psi) \times (1 + 2\psi + 3\psi^2 + \dots + n \times \psi^{n-1} + \dots). \end{aligned}$$

Выражение в последних скобках является производным от следующего выражения:

$$\psi + \psi^2 + \psi^3 + \dots + \psi^n + \dots = \psi \times (1 + \psi + \psi^2 + \dots + \psi^{n-1} + \dots) = \frac{\psi}{1 - \psi},$$

то есть равно $\frac{1}{(1 - \psi)^2}$.

Окончательно среднее число обслуживаемых требований N_{syst} , находящихся в системе, определяется по формуле:

$$N_{syst} = \frac{\Psi}{(1-\Psi)}.$$

Среднее же число требований (машин), находящихся в очереди, будет вычислено так:

$$N_{och} = \frac{\lambda}{\mu} N_{syst} = \frac{\Psi^2}{(1-\Psi)}.$$

Среднее время ожидания требования можно определить, зная среднее число требований, находящихся в системе:

$$T_{syst} = N_{syst} / \lambda = \frac{1}{\mu \times (1-\Psi)}.$$

Используя полученные выражения, определим основные параметры функционирования одноканальной разомкнутой СМО с простейшими потоками на примере функционирования системы «Станок–изделия». Допустим, что входной поток изделий, поступающих на обработку, является простейшим потоком со средним временем ожидания в очереди 10 мин и временем обработки 6 мин. Тогда интенсивность потока изделий, поступающих на обслуживание, составит

$$\lambda = \frac{60}{10} = 6 \text{ изделий в час.}$$

Интенсивность же потока изделий после обслуживания составит

$$\mu = \frac{60}{6} = 10 \text{ изделий в час.}$$

Откуда вероятность простоя канала обслуживания определяется так:

$$P_0 = 1 - \Psi = 1 - \frac{\lambda}{\mu} = 1 - \frac{6}{10} = 0,4.$$

Среднее число обслуживаемых требований N_{syst} , находящихся в системе:

$$N_{syst} = \frac{\Psi}{(1-\Psi)} = \frac{0,6}{1-0,6} = 1,5.$$

Среднее же число требований (машин), находящихся в очереди, будет вычислено так:

$$N_{och} = \frac{\lambda}{\mu} N_{syst} = \frac{\Psi^2}{(1-\Psi)} = \frac{0,6^2}{1-0,6} = \frac{0,36}{0,4} = 0,9.$$

Решение задачи имитационным методом

Рассмотрим теперь решение этой же задачи имитационным методом. Для облегчения построения имитационной модели, изобразим графически процесс функционирования одноканальной разомкнутой системы (рис. 5.2).

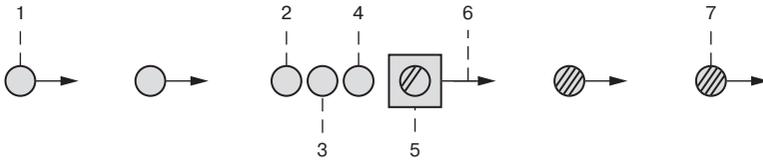


Рис. 5.2. Графическое изображение функционирования одноканальной разомкнутой системы

Рассмотрим все события, происходящие в одноканальной разомкнутой системе:

1. Генерирование требований, входящих в систему (**GENERATE** – Генерировать).
2. Вход требований в очередь (**QUEUE** – Очередь).
3. Проверка занятости канала обслуживания (**SEIZE** – Занять).
4. Выход требования из очереди (**DEPART** – Выйти).
5. Обслуживание требования (**ADVANCE** – Задержать).
6. Освобождение канала обслуживания (**RELEASE** – Освободить).
7. Выход требований из системы (**TERMINATE** – Завершить).

Поскольку требования не возвращаются в систему, то мы имеем одноканальную разомкнутую систему.

Построение имитационной модели процесса

Построение имитационной модели начнем с создания заголовка модели, который может быть представлен, например, в таком виде:

```
* GPSSW                Smo_1_re.GPS
*****
*      Моделирование одноканальной      *
*      разомкнутой системы                *
*      с простейшими потоками            *
*****
```

Поступление изделий к станку моделируется оператором **GENERATE**, который в нашей задаче может быть записан так:

```
GENERATE (Exponential(1,0,10))
```

В поле операнда *A* определяется интервал времени между прибытием двух идущих одно за другим изделий (требований) к станку (каналу обслуживания). В нашем примере используется встроенная функция экспоненциального распределения времени между поступлениями в систему двух идущих одно за другим требований. Среднее время прибытия требований составляет 10 мин – это третий аргумент функции *Exponential*.

Поступившее изделие (требование) встает в очередь для обработки. Это можно промоделировать оператором **QUEUE**, который только в совокупности с соответствующим оператором **DEPART** собирает статистическую информацию о работе моделируемой очереди.

В нашем примере оператор **QUEUE** будет выглядеть так:

```
QUEUE OCHER
```

В поле операнда *A* дается символьное или числовое имя очереди. В нашем примере очереди дано имя *ОЧЕР*. Желательно, чтобы присваиваемое имя отражало суть описываемого элемента системы.

Следуя логике, изделие может выйти из очереди только тогда, когда освободится станок (канал обслуживания). Для этого вводится оператор **SEIZE**, который определяет занятость канала обслуживания, и при освобождении последнего находящееся впереди требование выходит из очереди и идет в канал на обслуживание. Это может выглядеть так:

```
SEIZE STANOK
```

В поле операнда *A* дается символьное или числовое имя канала обслуживания. В нашей задаче каналу дано имя *СТАНОК*. Желательно, чтобы присваиваемое имя отражало суть описываемого элемента системы.

Выход изделия из очереди на обслуживание фиксируется оператором **DEPART** с соответствующим названием очереди. В нашем примере это будет выглядеть так:

```
DEPART OCHER
```

Далее должно быть промоделировано время изготовления изделия на станке. Для моделирования этого процесса используется оператор **ADVANCE**, который в нашей задаче будет выглядеть так:

```
ADVANCE (Exponential(1,0,6))
```

В поле операнда *A* выполняется обращение к встроенному в систему экспоненциальному распределению с указанием среднего времени изготовления изделия – 6 мин. После обработки изделия на станке должно быть послано сообщение об освобождении канала обслуживания. Это делается с помощью оператора **RELEASE**, который в нашей задаче записывается так:

```
RELEASE STANOK
```

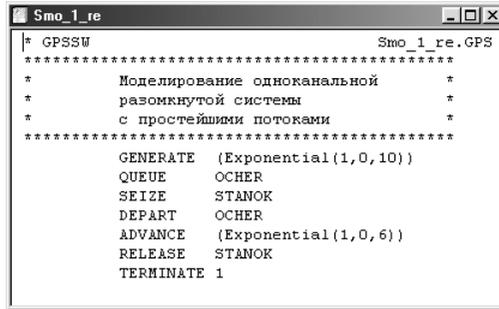
Следует особо подчеркнуть, что парные операторы **QUEUE** и **DEPART** для каждой очереди должны иметь одно и то же, но свое уникальное имя. Это же относится и к операторам **SEIZE** и **RELEASE**.

После изготовления изделие покидает систему. Это действие моделируется оператором **TERMINATE**, который записывается в программе в таком виде:

```
TERMINATE 1
```

Число изделий (требований), прошедших изготовление в нашей системе, прием равным 10000.

Окончательно наша программа будет выглядеть так, как показано на рис. 5.3.



```

* GPSSW Smo_1_re.GPS
*****
* Моделирование одноканальной *
* разомкнутой системы *
* с простейшими потоками *
*****
GENERATE (Exponential(1,0,10))
QUEUE OCHER
SEIZE STANOK
DEPART OCHER
ADVANCE (Exponential(1,0,6))
RELEASE STANOK
TERMINATE 1

```

Рис. 5.3. Окно с моделью одноканальной разомкнутой системы с простейшими потоками

Подготовка к моделированию системы

Для начала моделирования:

- щелкните по пункту **File** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **New** (Создать) выпадающего меню. Появится диалоговое окно **Новый документ**;
- выделите пункт **Model** и щелкните по кнопке **ОК**. Появится окно модели, в котором введите данную программу.

Чтобы вызвать окно для представления имитационной модели в системе GPSSW, можно также нажать комбинацию клавиш **Ctrl+Alt+S**.

Перед началом моделирования можно установить вывод тех параметров моделирования, которые нужны пользователю. Для этого:

- щелкните по пункту **Edit** (Правка) главного меню системы или нажмите комбинацию клавиш **Alt+E**. Появится выпадающее меню;
- щелкните по пункту **Settings** (Установки) выпадающего меню. Появится диалоговое окно **SETTINGS** для данной модели;
- щелкните по вкладке **Reports** (Отчеты), в которой можно установить нужные выходные данные. Для нашего примера это может выглядеть так, как показано на рис. 5.4.

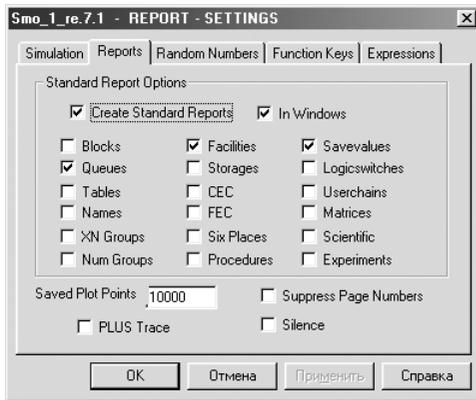


Рис. 5.4. Окно **SETTINGS** с установками для модели одноканальной разомкнутой системы с простейшими потоками

Наличие галочки в окошках говорит о том, что эта информация будет выведена в окне результатов моделирования. В нашем примере будет выведена информация по следующим объектам:

- **Facilities** (Каналы обслуживания);
- **Queues** (Очереди).

Моделирование системы

После создания имитационную модель необходимо оттранслировать. Для этого выполните следующие действия:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню.

После трансляции имитационной модели необходимо запустить ее на выполнение. Для этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **START**. Появится диалоговое окно **Start Command**;
- введите в диалоговом окне **Start Command** нужное число прогонов модели.

Проделаем небольшой эксперимент. Промоделируем нашу систему при различном числе прогонов модели – разном количестве изготавливаемых изделий – и сравним результаты моделирования с полученными при аналитическом расчете.

Если в диалоговом окне **Start Command** ввести число 100, 1000 или 10000 и щелкнуть по кнопке **OK**, то результаты моделирования в окне **REPORT** будут выглядеть так, как показано на рис. 5.5а–в соответственно.

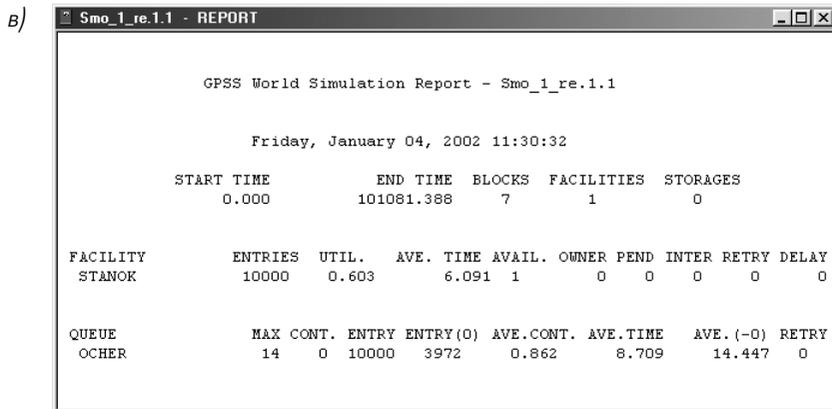
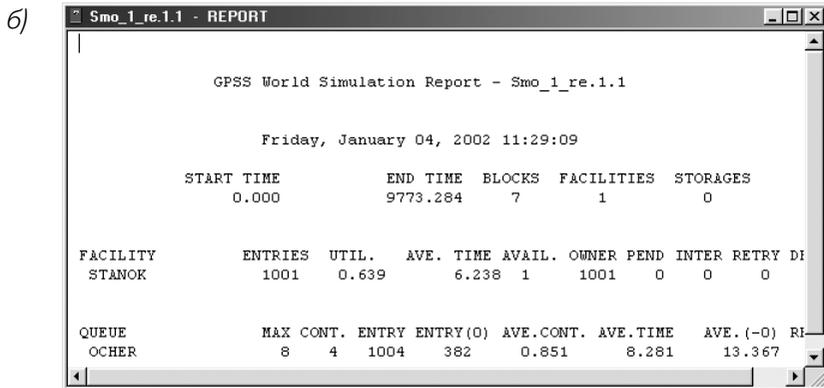
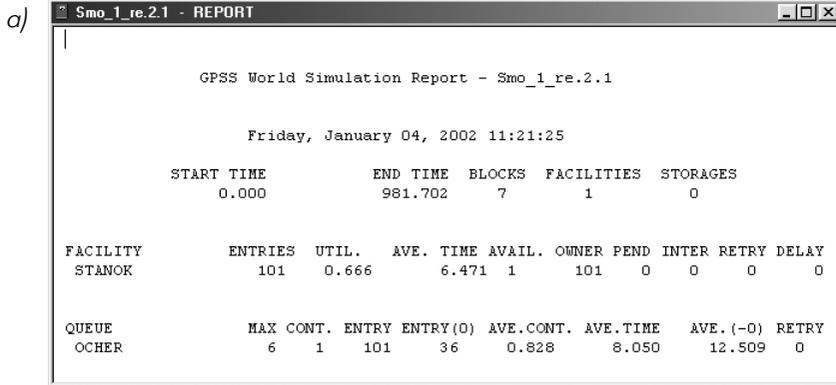


Рис. 5.5. Окно **REPORT** с различными вариантами результатов моделирования для имитационной модели одноканальной разомкнутой системы с простейшими потоками

Ниже указываются некоторые результаты моделирования с изготовлением 100, 1000 и 10000 изделий соответственно:

- **START TIME** (Начальное время) – 0.000, 0.000, 0.000;
- **END TIME** (Время окончания) – 981.702, 9773.284, 101081.388.

Ниже указываются результаты моделирования канала обслуживания (FACILITY) под именем STANOK:

- **ENTRIES** (Число входов) – 101, 1001, 10000;
- **UTIL.** (Коэффициент использования) – 0.666, 0.639, 0.603;
- **AVE. TIME** (Среднее время обслуживания) – 6.471, 6.238, 6.091.

Еще ниже указываются результаты моделирования очереди (QUEUE) под именем ОСНЕР:

- **MAX** (Максимальная длина очереди) – 6, 8, 14;
- **AVE.CONT.** (Среднее содержимое) – 0.828, 0.851, 0.862.

Разместим результаты, полученные путем аналитического расчета и методом имитационного моделирования при различном числе прогонов модели, в табл. 5.1.

Таблица 5.1

Параметры	Имитация изделий, шт.			Аналитический метод
	100	1000	10000	
Время моделирования	981,702	9773,284	101081,388	–
Коэффициент использования станка	0,666	0,639	0,603	0,600
Средняя длина очереди	0,828	0,851	0,862	0,9

Сравнивая результаты решения одной и той же задачи аналитическим и имитационным методами, видим, что при увеличении числа испытаний (имитаций) результаты, полученные имитационным методом, приближаются к результатам аналитического метода.

Одноканальная разомкнутая СМО с равномерными потоками

Решение этой задачи аналитическим методом невозможно. Поэтому ограничимся здесь использованием только имитационного метода.

Постановка задачи

Допустим, нам надо промоделировать работу одноканальной системы массового обслуживания – разомкнутой, с равномерным законом поступления требований на обслуживание (лифт–пассажиры, изделия–контролер, программы–ЭВМ и т.д.). Допустим, что нам известен интервал времени между поступлениями двух смежных требований, равный 8 ± 2 мин, в котором определяется время поступления

требования на обслуживание. Величина интервала времени между поступлениями двух смежных требований в систему подчиняется равномерному распределению. Требуется смоделировать процесс функционирования системы и определить следующие основные ее характеристики:

- коэффициент использования канала обслуживания;
- среднее время использования канала обслуживания;
- число входов в каждый канал обслуживания;
- среднее содержимое очереди;
- среднее время пребывания требования в очереди;
- максимальное содержимое очереди;
- коэффициент использования очереди.

Требования, поступающие в систему на обслуживание, не возвращаются в нее, то есть мы имеем одноканальную разомкнутую систему массового обслуживания.

Выявление основных особенностей

Для облегчения построения имитационной модели, изобразим графически процесс функционирования одноканальной разомкнутой системы с равномерными потоками (рис. 5.6).

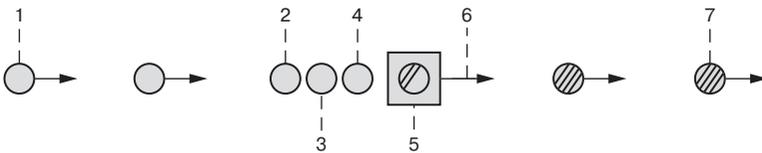


Рис. 5.6. Графическое изображение функционирования одноканальной разомкнутой системы с равномерными потоками

Рассмотрим все события, происходящие в одноканальной разомкнутой системе:

1. Генерирование требований, входящих в систему (**GENERATE** – Генерировать).
2. Вход требования в очередь (**QUEUE** – Очередь).
3. Проверка занятости канала обслуживания (**SEIZE** – Занять).
4. Выход требования из очереди (**DEPART** – Выйти).
5. Обслуживание требования (**ADVANCE** – Задержать).
6. Освобождение канала обслуживания (**RELEASE** – Освободить).
7. Выход требования из системы (**TERMINATE** – Закончить).

Поскольку требования не возвращаются в систему, то мы имеем одноканальную разомкнутую систему.

Создание имитационной модели

Создание имитационной модели начнем с построения заголовка модели, который может быть представлен, например, в таком виде:

```
* GPSSW          Smo_1_re.GPS
*****
*   Моделирование одноканальной   *
*   разомкнутой системы           *
*   с равномерными потоками      *
*****
```

Программу работы одноканальной разомкнутой СМО можно представить в виде последовательности нескольких операторов.

Моделирование потока требований будем выполнять с помощью оператора **GENERATE**. В нашем примере он будет выглядеть так:

```
GENERATE 8,2
```

В поле операнда А указывается средний интервал времени между прибытием двух смежных требований. В поле операнда В дано отклонение времени поступления требований от среднего, которое соответствует равномерному распределению поступления требований в систему в замкнутом интервале [6–10] с.

Сбор статистической информации для очереди в системе можно обеспечить с помощью операторов **OUEUE** и **DEPART**. Оператор **OUEUE** может быть записан в таком виде:

```
OUEUE 1
```

В поле операнда А дается символьное или числовое имя очереди. В нашем примере указан номер очереди – 1. Требование будет находиться в очереди до тех пор, пока не поступит сообщение об освобождении канала обслуживания. Для этого используется оператор **SEIZE**, который определяет занятость канала обслуживания, и при освобождении последнего находящееся впереди требование выходит из очереди и идет в канал на обслуживание. Это может выглядеть так:

```
SEIZE 1
```

В поле операнда А дается символьное или числовое имя канала обслуживания. В нашем примере используется номер 1.

Выход требования из очереди в канал обслуживания фиксируется оператором **DEPART** с соответствующим номером очереди. В нашем примере это будет выглядеть так:

```
DEPART 1
```

Далее должно быть промоделировано время пребывания требования, непосредственно обслуживаемого, в канале обслуживания. Это время в нашем примере составляет 7 ± 1 с. Для моделирования этого процесса используется оператор **ADVANCE**, который в нашей задаче будет выглядеть так:

```
ADVANCE 7,1
```

После обслуживания требование выходит из канала обслуживания. Для этого системе должно быть послано сообщение об освобождении канала обслуживания.

Это делается с помощью оператора **RELEASE**, который в нашей задаче записывается так:

```
RELEASE 1
```

Следует особо подчеркнуть, что парные операторы **QUEUE** и **DEPART** для каждой очереди должны иметь одно и то же, но свое уникальное имя или номер. Это же относится и к операторам **SEIZE** и **RELEASE**.

В нашем примере для идентификации имени очереди и канала использован номер 1.

Далее требование выходит из системы с помощью оператора **TERMINATE**, который записывается в таком виде:

```
TERMINATE 1
```

И наконец, последний оператор – управляющая команда **START** – определяет, сколько требований будет обслуживаться в процессе моделирования системы:

```
START 200
```

Представление имитационной модели

Для представления имитационной модели выполните следующие действия:

- щелкните по пункту **File** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **New** выпадающего меню. Появится диалоговое окно **Новый документ**;
- выделите пункт **Model** и щелкните по кнопке **ОК**. Появится окно модели, в котором введите данную программу. Это будет выглядеть так, как показано на рис. 5.7.

Чтобы вызвать окно для представления имитационной модели в системе GPSSW, можно также нажать комбинацию клавиш **Ctrl+Alt+S**.

```

Smo_1_r
* GPSSW                               Smo_1_re.GPS
*****
*   Моделирование одноканальной      *
*   разомкнутой системы              *
*   с равномерными потоками         *
*****
GENERATE 8,2                          ; Генерирование требования в интервале [6 - 10] ед.времени.
QUEUE 1                                ; Вход требования в очередь под номером 1.
SEIZE 1                                ; Определение занятости канала под номером 1.
DEPART 1                               ; Выход требования из очереди под номером 1.
ADVANCE 7,1                            ; Обслуживание требования в интервале [6 - 8] ед.времени.
RELEASE 1                              ; Освобождение канала обслуживания под номером 1.
TERMINATE 1                            ; Выход требования из системы.
START 200

```

Рис. 5.7. Окно имитационной модели одноканальной разомкнутой СМО

Эта же модель может иметь и другую форму записи. Для этого:

- щелкните по пункту **Edit** главного меню системы или нажмите комбинацию клавиш **Alt+E**. Появится выпадающее меню;
- щелкните по пункту **Settings** выпадающего меню. Появится диалоговое окно **SETTINGS** для данной модели;
- щелкните по вкладке **Function Keys** (Функциональные клавиши). В разделе **Model Tabstops** выберите переключатель **twips**, после чего активизируется кнопка **Применить**. Для нашего примера это может выглядеть так, как показано на рис. 5.8;
- щелкните по кнопке **Применить**, а затем – по кнопке **ОК**.

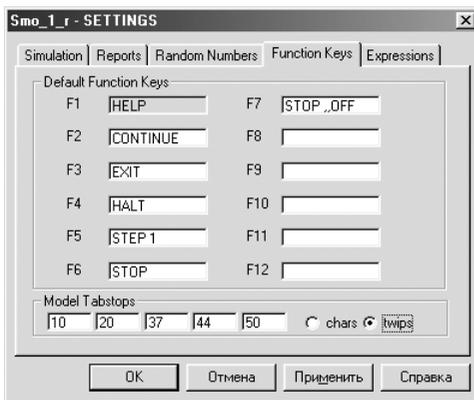


Рис. 5.8. Окно **SETTINGS** с установками для имитационной модели одноканальной разомкнутой СМО

Представленная на рис. 5.7 программа изменит свой вид и может теперь выглядеть так, как показано на рис. 5.9, с небольшими изменениями.

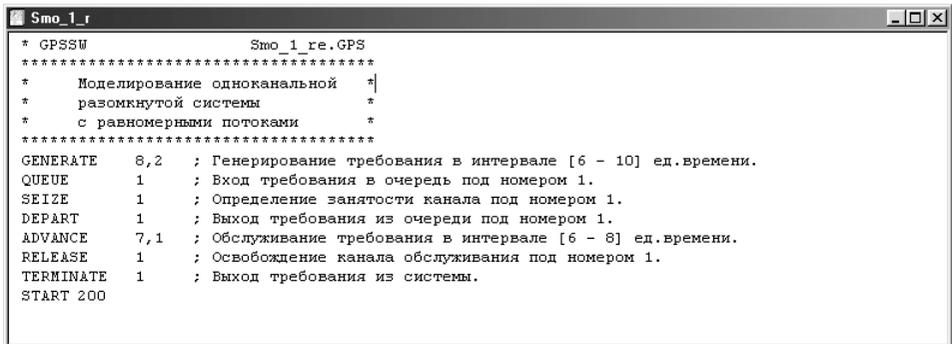


Рис. 5.9. Окно имитационной модели одноканальной разомкнутой СМО с другой формой представления

Подготовка системы к моделированию

Перед началом моделирования можно установить вывод тех параметров, которые необходимо получить в процессе моделирования. Для этого:

- щелкните по пункту **Edit** (Правка) главного меню системы или нажмите комбинацию клавиш **Alt+E**. Появится выпадающее меню;
- щелкните по пункту **Settings** (Установки) выпадающего меню. Появится диалоговое окно **SETTINGS** для данной модели, в котором можно установить нужные выходные данные. Для нашего примера это может выглядеть так, как показано на рис. 5.10.

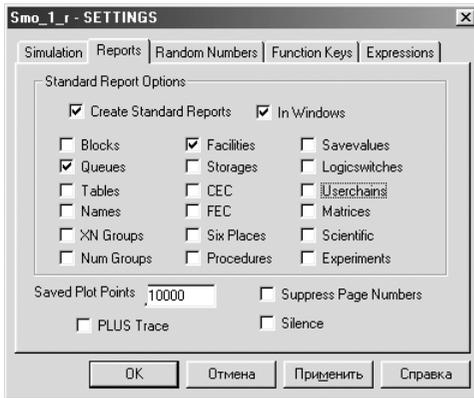


Рис. 5.10. Окно **SETTINGS** с открытой вкладкой **Reports** для имитационной модели одноканальной разомкнутой СМО

Наличие галочки в окошках говорит о том, что эта информация будет выведена в окне результатов моделирования. В нашем примере будет выведена информация для следующих объектов:

- **Facilities** (Каналы обслуживания);
- **Queues** (Очереди).

Моделирование системы

После создания имитационную модель необходимо оттранслировать и запустить на выполнение. Для этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню.

Так как в модели имеется управляющая команда **START**, то исходная имитационная модель будет выполняться после транслирования, если в ней нет ошибок.

Будет выполняться то число прогонов, которое указано в поле операнда **A** команды **START** – 200. Затем появится окно **JOURNAL** (Журнал) и результаты работы программы в окне **REPORT** (Отчет), как показано на рис. 5.11.

В верхней строке указывается:

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	1612.366	7	1	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
1	200	0.868	6.996	1	0	0	0	0	0

QUEUE	MAX CONT.	ENTRY	ENTRY(0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY
1	1	0	200	141	0.034	0.276	0.937

Рис. 5.11. Окно **REPORT** с результатами моделирования одноканальной разомкнутой СМО с равномерными потоками

- **START TIME** (Начальное время) – 0.000;
- **END TIME** (Время окончания) – 1612.366;
- **BLOCKS** (Число блоков) – 7;
- **FACILITIES** (Число каналов обслуживания) – 1;
- **STORAGES** (Число накопителей) – 0.

Ниже указываются результаты моделирования для канала обслуживания (**FACILITY**) под номером 1:

- **ENTRIES** (Число входов) – 200;
- **UTIL.** (Коэффициент использования) – 0.868;
- **AVE. TIME** (Среднее время обслуживания) – 6.996;
- **AVAIL.** (Доступность) – 1;
- **OWNER** – 0;
- **PEND** – 0;
- **INTER** – 0;
- **RETRY** (Повтор) – 0;
- **DELAY** (Отказ) – 0.

Ниже указываются результаты функционирования очереди (**QUEUE**) под номером 1:

- **MAX** (Максимальное содержание) – 1;
- **CONT.** (Текущее содержание) – 0;
- **ENTRY** (Число входов) – 200;

- **ENTRY(0)** (Число нулевых входов) – 141;
- **AVE.CONT.** (Среднее число входов) – 0.034;
- **AVE.TIME** (Среднее время) – 0.276;
- **AVE.(–0)** – 0.937;
- **RETRY** – 0.

Многоканальная разомкнутая СМО с простейшими потоками

Постановка задачи

Допустим, задана многоканальная разомкнутая система массового обслуживания с неограниченным временем ожидания и с простейшими потоками, которая наиболее соответствует действительности. Система характеризуется следующими особенностями:

- поступление требований в систему на обслуживание происходит по одному, то есть вероятность поступления двух и более требований в один момент времени очень мала, и ею можно пренебречь (поток требований ординарный);
- вероятность поступления последующих требований в любой момент времени не зависит от вероятности их поступления в предыдущие моменты – поток требований без последействия;
- поток требований стационарный.

Выявление основных особенностей

Функционирование многоканальной разомкнутой системы массового обслуживания можно описать через все возможные ее состояния и через интенсивность перехода из одного состояния в другое.

Основными параметрами функционирования СМО являются вероятности состояния системы, то есть возможность наличия n требований (покупателей, рабочих, заданий, машин, неполадок) в системе – P_n . Так, вероятность P_0 характеризует состояние, когда в системе нет требований и все каналы обслуживания простаивают, P_1 – когда в системе находится только одно требование, и т.д.

Важным параметром функционирования системы массового обслуживания является также среднее число требований, находящихся в системе (то есть в очереди и на обслуживании), – N_{syst} – и средняя длина очереди – N_{och} . Исходными параметрами, характеризующими систему массового обслуживания, являются:

- число каналов обслуживания (касс, компьютеров, подъемных кранов, ремонтных бригад) – N ;
- интенсивность поступления одного требования на обслуживание – λ ;
- интенсивность обслуживания требования – μ .

Интенсивность поступления на обслуживание одного требования определяется как величина, обратная времени между поступлениями двух смежных требований на обслуживание, $-t_p$:

$$\lambda = 1/t_p$$

Интенсивность обслуживания требований определяется как величина, обратная времени обслуживания одного требования, $-t_o$:

$$\mu = 1/t_o$$

Рассмотрим сначала решение задачи аналитическим методом. Для этого представим все возможные состояния системы массового обслуживания в виде размеченного графа состояний (рис. 5.12). Каждый прямоугольник графа определяет одно из всех возможных состояний, количественно оцениваемое вероятностью состояний P_n . P_n – это возможность наличия в системе n требований. Стрелки на графе указывают, в какое состояние система может перейти и с какой интенсивностью. При этом в многоканальной СМО необходимо различать два случая:

- число требований n , поступивших в систему, меньше количества каналов обслуживания N , то есть все они находятся на обслуживании ($0 \leq n < N$);
- число требований n , поступивших в систему, больше или равно числу каналов обслуживания N ($N \leq n$), то есть N требований обслуживаются, а остальные r ожидают в очереди ($r = 1, 2, \dots, n - N$).

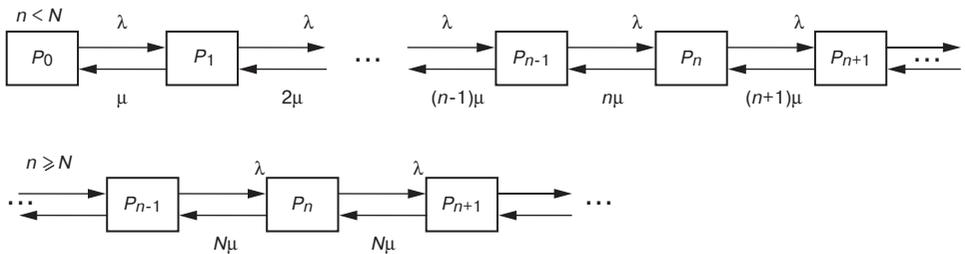


Рис. 5.12. Размеченный граф состояний многоканальной разомкнутой СМО

Первый прямоугольник с вероятностью P_0 определяет состояние системы массового обслуживания, при котором все каналы обслуживания простаивают из-за отсутствия требований в ней. Из этого положения СМО может перейти только в состояние P_1 , и тогда в ней появится одно требование, так как входной поток требований ординарный. С интенсивностью μ система может перейти также из состояния P_1 в состояние P_0 , если единственное требование, находившееся в системе, было обслужено раньше, чем появилось новое, и т.д.

Аналитический метод решения задачи

Ограничимся рассмотрением установившегося режима работы системы массового обслуживания, когда основные вероятностные характеристики СМО постоянны во

времени. Тогда интенсивности входных и выходных потоков для каждого состояния будут сбалансированы. Эти балансы выглядят следующим образом.

Если $0 \leq n < N$, то:

$$P_0 \times \lambda = P_1 \times \mu;$$

$$P_1 \times (\mu + \lambda) = P_0 \times \lambda + P_2 \times 2 \times \mu;$$

$$P_2 \times (2 \times \mu + \lambda) = P_1 \times \lambda + P_3 \times 3 \mu;$$

...

$$P_n \times (n \times \mu + \lambda) = P_{n-1} \times \lambda + P_{n+1} \times (n+1) \times \mu;$$

...

Если $N \leq n \leq \nu$, то:

...

$$P_n \times (N \times \mu + \lambda) = P_{n-1} \times \lambda + P_{n+1} \times N \times \mu$$

...

Обозначим величину λ / μ , как и раньше, через ψ и назовем ее коэффициентом загрузки.

Рассмотрим сначала первый случай, когда $0 \leq n < N$.

Из первого уравнения можно найти значение P_1 :

$$P_1 = P_0 \times \lambda / \mu = P_0 \times \psi.$$

Из второго уравнения найдем значение P_2 :

$$P_2 = P_1 / 2 + P_1 \times \lambda / 2\mu - P_0 \times \lambda / 2\mu.$$

Но первый член

$$P_1 = P_0 \times \lambda / \mu.$$

Следовательно, первый и третий сокращаются:

$$P_2 = P_1 \times \lambda / 2\mu = P_0 \times \psi^2 / 2.$$

Из третьего уравнения найдем значение P_3 :

$$P_3 = P_2 \times 2 / 3 + P_2 \times \lambda / 3\mu - P_1 \times \lambda / 3\mu.$$

Но первый член

$$P_2 = P_1 \times \lambda / 2\mu.$$

Следовательно, первый и третий сокращаются:

$$P_3 = P_2 \times \lambda / 3\mu = P_0 \times \psi^3 / (1 \times 2 \times 3).$$

Аналогичные выражения можно получить и для других вероятностей состояний. Анализируя полученные выражения, вычисляем рекуррентное выражение для определения вероятности состояния системы, когда число требований, находящихся в системе, n , меньше числа каналов обслуживания, N :

$$P'_n = P_{n-1} \times \lambda / (n \times \mu) = P_0 \times \psi^n / (1 \times 2 \times 3 \dots n);$$

$$P'_n = P_0 \times \psi^n / n!.$$

Рассмотрим теперь второй случай, когда $N \leq n \leq \infty$. В этой ситуации рекуррентное выражение для определения вероятности состояния системы будет записано в таком виде:

$$P''_n = P_0 \times \psi^n / N! \times N^{n-N}.$$

Используем очевидное равенство

$$\sum_{n=0}^{\infty} P_n = 1,$$

откуда

$$P_0 = (1 + \sum_{n=1}^{N-1} \psi^n / n! + \sum_{n=N}^{\infty} \psi^n / (N! N^{n-N}))^{-1}.$$

Допустим, что наша система имеет два канала обслуживания: $N = 2$. Интервал времени между поступлениями смежных требований составляет 10 мин. Среднее время обслуживания требования составляет 2 мин. Тогда коэффициент загрузки $\psi = 2/10 = 0,2$.

Требуется определить:

- вероятность отсутствия требований в системе – P_0 :

$$P_0 = (1 + \sum_{n=1}^{2-1} 0,2^n / n! + \sum_{n=2}^{\infty} 0,2^n / (2! \times 2^{n-2}))^{-1} = 0,85;$$

- вероятность наличия одного требования в системе – P_1 :

$$P_1 = P_0 \times \psi = 0,85 \times 0,2 = 0,17;$$

- вероятность наличия двух требований в системе – P_2 :

$$P_2 = P_1 \times \psi / 2 = 0,17 \times 0,2 / 2 = 0,017.$$

Имитационный метод решения задачи

Изобразим графически процесс функционирования двухканальной разомкнутой системы массового обслуживания. На рис. 5.13 представлены основные события, которые возникают в процессе работы многоканальной разомкнутой СМО.

Охарактеризуем каждое событие, возникшее в моделируемой системе:

1. Поступление требований в систему (**GENERATE**).
2. Вход требования в накопитель (**ENTER**).
3. Определение канала обслуживания (**TRANSFER**).
4. Ожидание освобождения одного из каналов обслуживания (**SEIZE**).
5. Выход требования из накопителя (**LEAVE**).
6. Время обслуживания требования в канале обслуживания (**ADVANCE**).

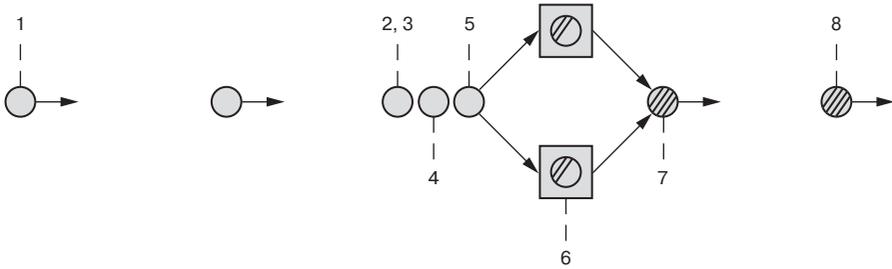


Рис. 5.13. Процесс функционирования многоканальной разомкнутой СМО с простейшими потоками

- 7. Освобождение канала обслуживания (**RELEASE**).
- 8. Выход требования из системы (**TERMINATE**).

Построение имитационной модели процесса

Построение имитационной модели начнем с создания заголовка модели, который может быть представлен, например, в таком виде:

```
; GPSSW File                               Smo_2_re.GPS
*****
*   Моделирование двухканальной разомкнутой   *
*   системы с простейшими потоками           *
*****
```

Программу работы двухканальной разомкнутой СМО можно представить в виде трех секторов.

В первом секторе указывается вместимость СМО. Это можно выполнить с помощью оператора **STORAGE** (Накопитель), который в нашем примере будет выглядеть так:

```
NAK STORAGE 3
```

Далее будем моделировать поток требований в систему и их обслуживание. Моделирование потока требований будем выполнять с помощью оператора **GENERATE** (Генерировать). В нашем примере он будет выглядеть так:

```
GENERATE (Exponential(1,0,10))
```

В поле операнда А осуществляется обращение к встроенному экспоненциальному распределению, с помощью которого можно моделировать простейший поток требований, поступающих в систему. Средний интервал времени между поступлениями смежных требований у нас принят равным 10 единицам времени.

Сбор статистической информации для многоканальной системы можно обеспечить с помощью операторов **ENTER** и **LEAVE**. Оператор **ENTER** может быть записан в таком виде:

```
ENTER NAK
```

В поле операнда А указано имя накопителя, вместимость которого должна быть заранее определена.

Поскольку СМО многоканальная, то необходимо использовать оператор **TRANSFER** для обеспечения возможности направления требований к незанятому каналу:

```
TRANSFER BOTH, KAN1, KAN2
```

Сначала требование направляется к оператору, имеющему символическую метку KAN1. Этим оператором является **SEIZE**, который записывается так:

```
KAN1 SEIZE CAN1
```

Если канал обслуживания с символической меткой KAN1 занят, то требование направляется к оператору, имеющему символическую метку KAN2. Таким образом, следующим оператором будет:

```
KAN2 SEIZE CAN2
```

В свободном канале обслуживания требование будет обслужено. Но предварительно требование должно запомнить канал, в который оно попало на обслуживание. Для этого используется оператор **ASSIGN** (Присвоить) – с его помощью в параметре требования под номером 1 запоминается имя канала, в который требование пошло на обслуживание. Присвоение будет выглядеть так:

```
ASSIGN 1, CAN1
```

Далее, после определения свободного канала и записи его имени с помощью оператора **TRANSFER** требование направляется к оператору с символической меткой COME. Это выглядит так:

```
TRANSFER , COME
```

Требование при направлении в канал сначала попадает в оператор **LEAVE** (Оставить). Это будет выглядеть так:

```
COME LEAVE NAK
```

После выхода из накопителя требование поступает в канал на обслуживание. Это действие выполняется с помощью оператора **ADVANCE** и записывается так:

```
ADVANCE (Exponential(1,0,2))
```

В поле операнда А осуществляется обращение к встроенному экспоненциальному распределению, с помощью которого можно моделировать время обслуживания требования в канале обслуживания. Среднее время обслуживания у нас принято равным 2 единицам времени.

После обслуживания требование выходит из канала обслуживания, и должен появиться сигнал об освобождении последнего. Это делается с помощью оператора **RELEASE** (Освободить):

```
RELEASE P1
```

Оператор требования под номером P1 содержит имя освобождаемого канала обслуживания. Далее требование выходит из системы с помощью оператора **TERMINATE**, который записывается в таком виде:

```
TERMINATE 1
```

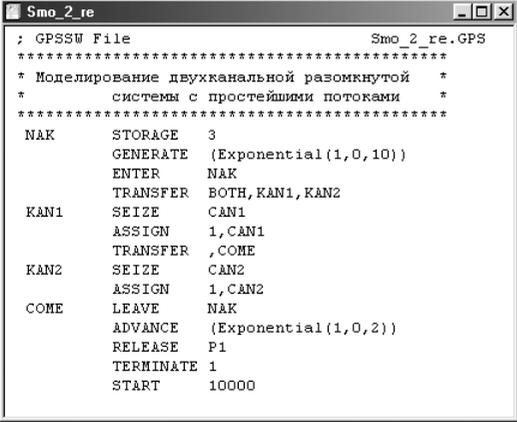
И наконец, последний оператор – управляющая команда **START** – определяет, сколько требований поступит в процессе моделирования системы:

```
START 10000
```

Представление имитационной модели

Для представления имитационной модели выполните следующие действия:

- щелкните по пункту **File** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **New** (Создать) выпадающего меню. Появится диалоговое окно **Новый документ**;
- выделите пункт **Model** и щелкните по кнопке **ОК**. Появится окно модели, в котором введите данную программу. Оно будет выглядеть так, как показано на рис. 5.14.



```

; GPSSW File                               Smo_2_re.GPS
*****
* Моделирование двухканальной разомкнутой *
* системы с простейшими потоками         *
*****
NAK      STORAGE      3
         GENERATE      (Exponential(1,0,10))
         ENTER        NAK
         TRANSFER      BOTH, KAN1, KAN2
KAN1     SEIZE         CAN1
         ASSIGN        1, CAN1
         TRANSFER      ,COME
KAN2     SEIZE         CAN2
         ASSIGN        1, CAN2
COME     LEAVE         NAK
         ADVANCE       (Exponential(1,0,2))
         RELEASE      P1
         TERMINATE    1
         START        10000

```

Рис. 5.14

Окно имитационной модели
двухканальной разомкнутой СМО

Чтобы вызвать окно для представления имитационной модели в системе GPSSW, можно также нажать комбинацию клавиш **Ctrl+Alt+S**.

Подготовка системы к моделированию

Перед началом моделирования можно установить вывод тех параметров, которые необходимо получить в процессе моделирования. Для этого:

- щелкните по пункту **Edit** (Правка) главного меню системы или нажмите комбинацию клавиш **Alt+E**. Появится выпадающее меню;

- щелкните по пункту **Settings** (Установки) выпадающего меню. Появится диалоговое окно **SETTINGS** для данной модели, в котором можно установить нужные выходные данные. Для нашего примера оно может выглядеть так, как показано на рис. 5.15.

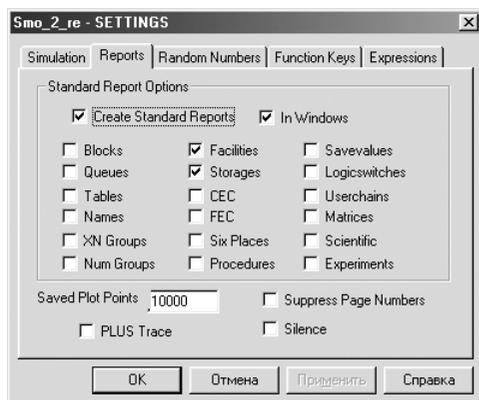


Рис. 5.15. Окно **SETTINGS** с установками для имитационной модели многоканальной разомкнутой СМО

Наличие галочки в окошках говорит о том, что эта информация будет выведена в окне результатов моделирования. В нашем примере будет выведена информация для следующих объектов:

- Facilities** (Каналы обслуживания);
- Storages** (Накопители).

Моделирование системы

После создания имитационную модель необходимо оттранслировать и запустить на выполнение. Для этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню.

Если управляющая команда **START** есть в модели, то исходная имитационная модель будет транслироваться, и если в ней нет ошибок, то начнется процесс моделирования системы. Будет выполняться то число прогонов, которое указано в поле операнда A команды **START**. Затем появится окно **JOURNAL** (Журнал).

Если управляющей команды **START** в модели нет, то исходная имитационная модель будет транслироваться. И если в ней нет ошибок, то будет получена оттранслированная модель – в машинных кодах, готовая к выполнению моделирования, так называемая выполняемая модель.

Для начала моделирования:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;

- щелкните по пункту **Create Simulation** выпадающего меню. Начнется трансляция исходной модели, а затем – и ее выполнение, так как в самой программе имеется управляющая команда **START**, обеспечивающая автоматическое выполнение оттранслированной программы. После выполнения программы появится окно с информацией о трансляции и выполнении – **JOURNAL** – и результаты работы программы в окне **REPORT** (Отчет), как показано на рис. 5.16.

The screenshot shows a window titled "Smo_2_re.1.1 - REPORT" containing the following text:

```

GPSS World Simulation Report - Smo_2_re.1.1

Friday, January 04, 2002 14:14:23

START TIME          END TIME  BLOCKS  FACILITIES  STORAGES
0.000              101188.678    12       2           1

FACILITY           ENTRIES  UTIL.    AVE. TIME AVAIL.  OWNER  PEND  INTER  RETRY  DELAY
CAN1                8382    0.168    2.033  1        0    0    0      0      0
CAN2                1618    0.032    2.001  1        0    0    0      0      0

STORAGE            CAP.  REM.  MIN.  MAX.  ENTRIES  AVL.  AVE.C.  UTIL.  RETRY  DELAY
NAK                 3    3    0    3    10000  1    0.002  0.001  0      0
  
```

Рис. 5.16. Окно **REPORT** с результатами моделирования многоканальной разомкнутой СМО

В верхней строке указывается:

- **START TIME** (Начальное время) – 0,000;
- **END TIME** (Время окончания) – 101188.678;
- **BLOCKS** (Число блоков) – 12;
- **FACILITIES** (Число каналов обслуживания) – 2;
- **STORAGES** (Число накопителей) – 1.

Ниже указываются результаты моделирования двух каналов обслуживания (FACILITY) соответственно под именами CAN1 и CAN2:

- **ENTRIES** (Число входов) – 8382, 1618;
- **UTIL.** (Коэффициент использования) – 0.168, 0.032;
- **AVE. TIME** (Среднее время обслуживания) – 2.033, 2.00.

Ниже указываются результаты функционирования накопителя (STORAGE) под именем NAK:

- **CAP.** (Capacity – Вместимость) – 3;
- **REM.** (Remove – Удален) – 3;
- **MIN.** (Минимальное содержимое) – 0;
- **MAX.** (Максимальное содержимое) – 3;
- **ENTRIES** (Число входов) – 10000;

- **AVL.** (Доступность) – 1;
- **AVE.C.** (Средняя вместимость) – 0.002;
- **UTIL.** (Коэффициент использования) – 0.001;
- **RETRY** (Повтор) – 0;
- **DELAY** (Отказ) – 0.

Сравним результаты, полученные аналитическим и имитационным методами. Вероятность наличия одного требования в системе, определенная аналитическим методом, равна 0,17, а коэффициент использования первого канала обслуживания, определенный имитационным способом, равен 0,168. То есть результаты практически совпадают.

Многоканальная разомкнутая СМО со смешанными потоками

Постановка задачи

Допустим, нам надо промоделировать работу многоканальной системы массового обслуживания – разомкнутой с экспоненциальным законом поступления требований на обслуживание (лифты–пассажиры, изделия–контролеры, программы–ЭВМ и т.д.), – для которой справедливы следующие условия:

- поступление одного требования в систему на обслуживание не зависит от поступления другого (отсутствие последствия);
- в систему одновременно никогда не поступает два или более требований (поток ординарный);
- вероятность поступления требований зависит только от продолжительности периода наблюдения (поток требований стационарный), а не от принятого начала отсчета времени.

Известно среднее время поступления требования на обслуживание, равное 60 с, которое подчиняется экспоненциальному распределению вероятностей. В системе массового обслуживания имеется три канала обслуживания, время обслуживания в которых равномерное. Требуется смоделировать процесс функционирования системы и определить следующие основные ее характеристики:

- коэффициент использования каждого канала обслуживания;
- среднее время использования каждого канала обслуживания;
- число входов в каждый канал обслуживания;
- среднее содержимое накопителя;
- среднее время пребывания требования в накопителе;
- максимальное содержимое накопителя;
- коэффициент использования накопителя.

Требования, поступающие в систему на обслуживание, не возвращаются в нее, то есть мы имеем многоканальную разомкнутую систему массового обслуживания. Для решения этой задачи нет аналитических методов.

Выявление основных особенностей

Изобразим графически процесс функционирования трехканальной разомкнутой системы массового обслуживания. На рис. 5.17 представлены основные события, которые возникают в процессе работы СМО.

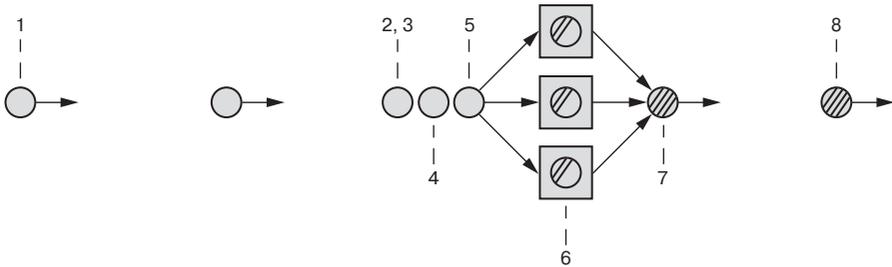


Рис. 5.17. Процесс функционирования многоканальной разомкнутой СМО со смешанными потоками

Охарактеризуем каждое событие, возникшее в моделируемой системе:

1. Поступление требований в систему (**GENERATE**).
2. Вход требования в накопитель (**ENTER**).
3. Передача требования в один из свободных каналов обслуживания (**TRANSFER**).
4. Ожидание освобождения одного из каналов обслуживания (**SEIZE**).
5. Выход требования из накопителя (**LEAVE**).
6. Время обслуживания требования в канале обслуживания (**ADVANCE**).
7. Освобождение канала обслуживания (**RELEASE**).
8. Выход требования из системы (**TERMINATE**).

Создание имитационной модели процесса

Создание имитационной модели начнем с построения заголовка модели, который может быть представлен, например, в таком виде:

```
; GPSSW File                Smo3re.GPS
*****
*   Моделирование трехканальной разомкнутой   *
*           системы со смешанными потоками     *
*****
```

Программу работы СМО можно представить в виде трех секторов.

В первом секторе указывается вместимость СМО. Это можно выполнить с помощью оператора **STORAGE** (Накопитель), который в нашем примере будет выглядеть так:

```
NAK STORAGE 3
```

В этом же секторе используем оператор **FUNCTION** (Функция) для формирования экспоненциального распределения поступления требований в систему. Ввод функции выполняется с использованием непрерывной числовой функции, с помощью значений функции в 24 точках. Это можно представить в таком виде:

```
EXP_G FUNCTION RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915
.7,1.2/.75,1.38/.8,1.6/.84,1.83/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9
.99,4.6/.995,5.3/.998,6.2/.999,7./ .9997,8.
```

Функция – это вычислительный элемент системы моделирования, связывающий зависимую и независимую переменные моделирования. Функция определяется с помощью оператора **FUNCTION**. Перед оператором **FUNCTION** в поле меток ставится символьное или числовое имя (идентификатор) функции. В нашем примере ее имя – **EXP_G**. В поле операнда А задается аргумент (независимая переменная) функции. Аргумент функции может быть любым стандартным числовым атрибутом, за исключением матричной сохраняемой величины. Если в качестве аргумента функции используется случайное число **RN___**, то результатом является дробная величина, равномерно распределенная в интервале $0 \leq \text{RN} \leq 1$. Во всех других случаях $0 \leq \text{аргумент} \leq 999$.

Во втором секторе будем моделировать поток требований в систему и их обслуживание. Моделирование потока требований будем выполнять с помощью оператора **GENERATE** (Генерировать). В нашем примере он будет выглядеть так:

```
GENERATE 12, FN$EXP_G
```

В поле операнда А указывается средний интервал времени между прибытием двух смежных требований. В поле операнда В дано отклонение времени поступления требований от среднего. В нашем примере отклонение от среднего времени прибытия требований подчиняется экспоненциальному распределению.

Сбор статистической информации для многоканальной системы можно обеспечить с помощью операторов **ENTER** и **LEAVE**. Оператор **ENTER** может быть записан в таком виде:

```
ENTER NAK
```

В поле операнда А указано имя накопителя, вместимость которого должна быть заранее определена.

Поскольку СМО многоканальная, то необходимо использовать оператор **TRANSFER** для обеспечения возможности направления требований к неза занятому каналу:

```
TRANSFER ALL, KAN1, KAN3, 3
```

Сначала требование направляется к оператору, имеющему символьную метку **KAN1**. Этим оператором является **SEIZE**, который записывается так:

```
KAN1 SEIZE CAN1
```

Если канал обслуживания с символьной меткой KAN1 занят, то требование направляется к следующему каналу, перешагивая через три оператора. 3 – это число, указанное в поле операнда D в операторе **TRANSFER**. Таким образом, следующим оператором будет:

```
SEIZE CAN2
```

Если и этот канал будет занят, то требование снова перешагнет через три оператора и т.д., пока не найдется незанятый канал обслуживания. В свободном канале обслуживания требование будет обслужено. Но предварительно требование должно запомнить канал, в который оно попало на обслуживание. Для этого используется оператор **ASSIGN** (Присвоить) – с его помощью в параметре требования под номером 1 запоминается имя канала, в который требование пошло на обслуживание. В каждом канале имеется свой оператор **ASSIGN**. Например, для первого канала это присвоение будет выглядеть так:

```
ASSIGN 1,CAN1
```

Далее, после определения свободного канала и записи его имени с помощью оператора **TRANSFER** требование направляется на обслуживание. Это выглядит так:

```
TRANSFER ,COME
```

Однако перед началом обслуживания должно быть подано сообщение о том, что требование оставило накопитель под именем NAK, в котором оно находилось. Это будет выглядеть так:

```
COME LEAVE NAK
```

После выхода из накопителя требование поступает в канал на обслуживание. Это действие выполняется с помощью оператора **ADVANCE**. Время обслуживания определяется с равномерным законом распределения в интервале 8 ± 3 с. Это записывается так:

```
ADVANCE 8,3
```

После обслуживания требование выходит из канала обслуживания, и должен появиться сигнал об освобождении последнего. Это делается с помощью оператора **RELEASE** (Освободить):

```
RELEASE P1
```

Оператор требования под номером P1 содержит имя освобождаемого канала обслуживания. Далее требование выходит из системы с помощью оператора **TERMINATE**, который записывается в таком виде:

```
TERMINATE 1
```

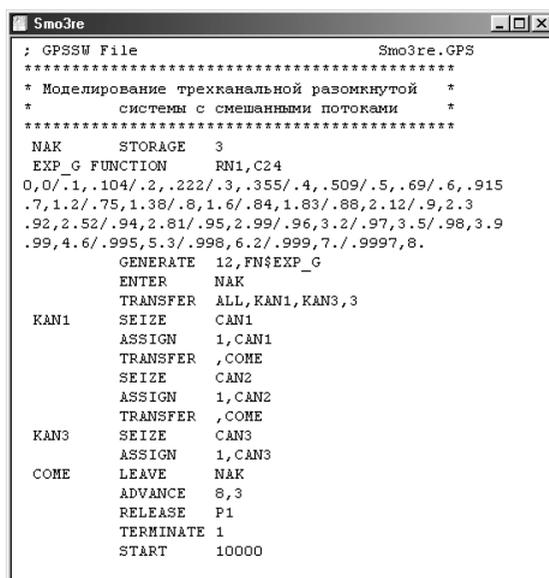
И наконец, последний оператор – управляющая команда **START** – определяет, сколько поступит требований в процессе моделирования системы:

```
START 10000
```

Представление имитационной модели

Для представления имитационной модели выполните следующие действия:

- щелкните по пункту **File** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **New** (Создать) выпадающего меню. Появится диалоговое окно **Новый документ**;
- выделите пункт **Model** и щелкните по кнопке **ОК**. Появится окно модели, в котором введите данную программу. Оно будет выглядеть так, как показано на рис. 5.18.



```

; GPSSW File                               Smo3re.GPS
*****
* Моделирование трехканальной разомкнутой *
* системы с смешанными потоками *
*****
NAK      STORAGE      3
EXP_G FUNCTION  RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915
.7,1.2/.75,1.38/.8,1.6/.84,1.83/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9
.99,4.6/.995,5.3/.998,6.2/.999,7./9997,8.
      GENERATE  12, FN$EXP_G
      ENTER    NAK
      TRANSFER ALL, KAN1, KAN3, 3
KAN1   SEIZE   CAN1
      ASSIGN  1, CAN1
      TRANSFER , COME
      SEIZE   CAN2
      ASSIGN  1, CAN2
      TRANSFER , COME
KAN3   SEIZE   CAN3
      ASSIGN  1, CAN3
COME   LEAVE   NAK
      ADVANCE 8, 3
      RELEASE P1
      TERMINATE 1
      START   10000
  
```

Рис. 5.18. Окно имитационной модели многоканальной разомкнутой СМО

Чтобы вызвать окно для представления имитационной модели в системе GPSSW, можно также нажать комбинацию клавиш **Ctrl+Alt+S**.

Подготовка к моделированию

Перед началом моделирования можно установить вывод тех параметров, которые необходимо получить в процессе моделирования. Для этого:

- щелкните по пункту **Edit** (Правка) главного меню системы или нажмите комбинацию клавиш **Alt+E**. Появится выпадающее меню;
- щелкните по пункту **Settings** (Установки) выпадающего меню. Появится диалоговое окно **SETTINGS** для данной модели, в котором можно установить

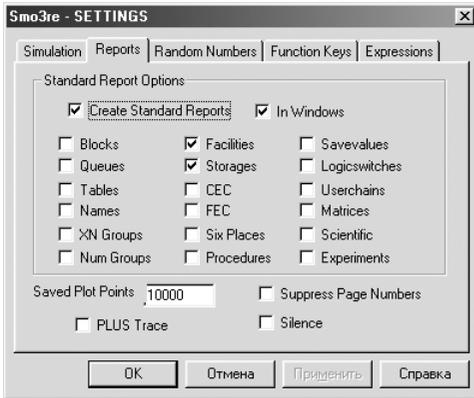


Рис. 5.19. Окно **SETTINGS** с установками для имитационной модели многоканальной разомкнутой СМО

нужные выходные данные. Для нашего примера оно может выглядеть так, как показано на рис. 5.19.

Наличие галочки в окошках говорит о том, что эта информация будет выведена в окне результатов моделирования. В нашем примере будет выведена информация для следующих объектов:

- **Facilities** (Каналы обслуживания);
- **Storages** (Накопители).

Моделирование системы

После создания имитационную модель необходимо оттранслировать и запустить на выполнение.

Для начала моделирования:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню. Начнется трансляция исходной модели, а затем – и ее выполнение, так как в самой программе имеется управляющая команда **START**, обеспечивающая автоматическое выполнение оттранслированной программы. После выполнения программы появится окно с информацией о трансляции и выполнении – **JOURNAL** (Журнал) – и результаты работы программы в окне **REPORT** (Доклад), как показано на рис. 5.20.

В верхней строке указывается:

- **START TIME** (Начальное время) – 0.000;
- **END TIME** (Время окончания) – 12196.149;
- **BLOCKS** (Число блоков) – 15;

GPSS World Simulation Report - Smo3re.1.1

Friday, January 04, 2002 14:58:01

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	121196.149	15	3	1

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
CAN1	6113	0.399	7.909	1	0	0	0	0	0
CAN2	2899	0.192	8.009	1	0	0	0	0	0
CAN3	988	0.066	8.038	1	0	0	0	0	0

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
NAK	3	3	0	3	10000	1	0.006	0.002	0	0

Рис. 5.20. Окно **REPORT** с результатами моделирования многоканальной разомкнутой СМО

- **FACILITIES** (Число каналов обслуживания) – 3;
- **STORAGES** (Число накопителей) – 1.

Ниже указываются результаты моделирования для всех трех каналов обслуживания (FACILITY) соответственно под именами: CAN1, CAN2, CAN3:

- **ENTRIES** (Число входов) – 6113, 2899, 988;
- **UTIL.** (Коэффициент использования) – 0.399, 0.192, 0.066;
- **AVE. TIME** (Среднее время обслуживания) – 7.909, 8.009, 8.038.

Ниже указываются результаты функционирования накопителя (STORAGE) под именем NAK:

- **CAP.** (Capacity – Вместимость) – 3;
- **REM.** (Remove – Удален) – 3;
- **MIN.** (Минимальное содержимое) – 0;
- **MAX.** (Максимальное содержимое) – 3;
- **ENTRIES** (Число входов) – 10000;
- **AVL.** (Доступность) – 1;
- **AVE.C.** (Средняя вместимость) – 0.006;
- **UTIL.** (Коэффициент использования) – 0.002;
- **RETRY** (Повтор) – 0;
- **DELAY** (Отказ) – 0.

Модернизация имитационной модели

Для упрощения создания имитационной модели целесообразно использовать встроенные распределения вероятностей. Тогда наша программа будет выглядеть так, как показано на рис. 5.21.

```

Smo3re_1
; GPSSW File                               Smo3re_1.GPS
*****
* Моделирование трехканальной разомкнутой *
* системы с смешанными потоками          *
*****
NAK      STORAGE      3
          GENERATE     (Exponential(1,0,12))
          ENTER        NAK
          TRANSFER     ALL, KAN1, KAN3, 3
KAN1     SEIZE         CAN1
          ASSIGN       1, CAN1
          TRANSFER     ,COME
          SEIZE        CAN2
          ASSIGN       1, CAN2
          TRANSFER     ,COME
KAN3     SEIZE         CAN3
          ASSIGN       1, CAN3
COME     LEAVE         NAK
          ADVANCE      8, 3
          RELEASE     P1
          TERMINATE   1
          START       1000
    
```

Рис. 5.21. Окно с модернизированной имитационной моделью многоканальной разомкнутой СМО

После выполнения программы появится окно с информацией о трансляции и выполнении – **JOURNAL** – и результаты работы программы в окне **REPORT** (рис. 5.22).

Как вы видите, результаты моделирования в обоих случаях мало различаются. Однако можно заметить и другое: коэффициенты использования всех трех каналов обслуживания достаточно низки, особенно третьего, который составляет менее 7%. Отсюда можно сделать вывод о том, что число каналов обслуживания может быть уменьшено.

GPSS World Simulation Report - Smo3re_1.1.1

Friday, January 04, 2002 15:07:22

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	11601.654	15	3	1

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
CAN1	604	0.414	7.952	1	0	0	0	0	0
CAN2	302	0.206	7.914	1	1001	0	0	0	0
CAN3	95	0.066	8.089	1	0	0	0	0	0

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
NAK	3	3	0	2	1001	1	0.007	0.002	0	0

Рис. 5.22. Окно **REPORT** с фрагментом результатов моделирования имитационной модели многоканальной разомкнутой СМО

Анализ работы замкнутых СМО

Одноканальная замкнутая СМО с простейшими потоками

Постановка задачи

Рассмотрим сначала одноканальную замкнутую систему массового обслуживания с неограниченным временем ожидания требований и с простейшим потоком. Эта система наиболее полно отвечает реалиям жизни и характеризуется следующими особенностями:

- поступление требований в систему на обслуживание происходит по одному, то есть вероятность прибытия двух и более требований в один момент времени очень мала, и ею можно пренебречь (поток требований ординарный);
- вероятность поступления последующих требований в любой момент времени не зависит от возможности их прибытия в предыдущие моменты – поток требований без последействия;
- поток требований стационарный.

Требуется определить:

- коэффициент использования канала обслуживания;
- среднюю длину очереди, то есть среднее число требований, находящихся в очереди, ожидая освобождения канала обслуживания;
- среднее число требований, находящихся в системе, то есть в очереди и в канале обслуживания.

Выявление основных особенностей

Функционирование любой системы массового обслуживания можно представить через все возможные ее состояния и интенсивность перехода из одного состояния в другое. Основными параметрами функционирования системы массового обслуживания являются вероятности ее состояния, то есть возможность наличия n требований (покупателей, рабочих, заданий, машин, неполадок) в системе – P_n . Так, вероятность P_0 характеризует состояние, когда в системе нет требований и канал обслуживания простаивает.

Важным параметром функционирования системы массового обслуживания является также среднее число требований, находящихся в системе, то есть в очереди на обслуживание – N_{syst} , а также средняя длина очереди – $N_{оч}$. Исходными параметрами, характеризующими систему массового обслуживания, являются:

- число требований (покупателей, рабочих, заданий, машин, неполадок) – m ;
- интенсивность поступления одного требования на обслуживание, то есть число поступлений требований в единицу времени – λ ;
- интенсивность обслуживания требований – μ .

Интенсивность поступления требования на обслуживание определяется как величина, обратная времени возвращения требования в систему, $-t_p$:

$$\lambda = 1/t_p$$

Интенсивность обслуживания требований определяется как величина, обратная времени обслуживания одного требования, $-t_o$:

$$\mu = 1/t_o$$

Рассмотрим сначала решение задачи аналитическим методом. Состояние системы массового обслуживания будем связывать с числом требований, находящихся в системе:

- в системе нет ни одного требования – вероятность состояния P_0 ;
- ...
- в системе находится одно требование – вероятность состояния P_1 ;
- в системе находится n требований – вероятность состояния P_n ;
- ...
- в системе находится m требований – вероятность состояния P_m .

Представим все возможные состояния системы массового обслуживания в виде размеченного графа состояний (рис. 5.23). Каждый прямоугольник графа, количественно оцениваемый вероятностью состояний P_n , определяет одно из всех возможных состояний. Стрелки указывают, в какое состояние система может перейти и с какой интенсивностью.

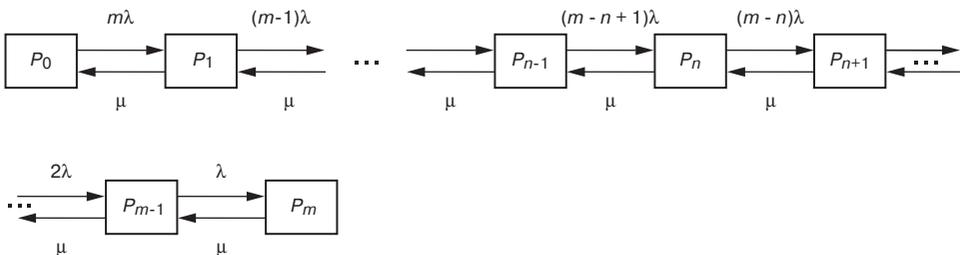


Рис. 5.23. Размеченный граф состояний одноканальной замкнутой СМО

Первый прямоугольник с вероятностью P_0 определяет состояние системы массового обслуживания, при котором канал обслуживания простаивает из-за отсутствия требований в ней. Из этого положения система массового обслуживания может перейти только в состояние P_1 ; тогда в системе появится одно требование, так как входной поток их ординарный. С интенсивностью μ система может перейти также из состояния P_1 в состояние P_0 , если единственное находящееся в системе требование было обслужено раньше, чем появилось новое.

Рассмотрим установившийся режим работы системы массового обслуживания, когда основные вероятностные характеристики СМО постоянны во времени,

например в течение часа. Тогда интенсивности входных и выходных потоков для каждого состояния будут сбалансированы. Эти балансы выглядят так:

$$P_0 \times m \times \lambda = P_1 \times \mu;$$

$$P_1 \times (\mu + (m-1) \times \lambda) = P_0 \times m \times \lambda + P_2 \times \mu;$$

$$P_2 \times (\mu + (m-2) \times \lambda) = P_1 \times (m-1) \times \lambda + P_3 \times \mu;$$

...

$$P_n \times (\mu + (m-n) \times \lambda) = P_{n-1} \times (m-(n-1)) \times \lambda + P_{n+1} \times \mu;$$

...

$$P_m \times \mu = P_{m-1} \times \lambda.$$

Обозначим величину λ / μ через ψ и назовем ее коэффициентом загрузки.

Из первого уравнения можно найти значение P_1 :

$$P_1 = P_0 \times m \times \lambda / \mu = P_0 \times m \times \psi.$$

Из второго уравнения найдем значение P_2 :

$$P_2 = P_1 + P_1 \times (m-1) \times \lambda / \mu - P_0 \times m \times \lambda / \mu.$$

Но первый член

$$P_1 = P_0 \times m \times \lambda / \mu.$$

Следовательно, первый и третий сокращаются:

$$P_2 = P_1 \times (m-1) \times \lambda / \mu = P_0 \times m \times (m-1) \times \psi^2.$$

Из третьего уравнения найдем значение P_3 :

$$P_3 = P_2 + P_2 \times (m-2) \times \lambda / \mu - P_1 \times (m-1) \times \lambda / \mu.$$

Но первый член

$$P_2 = P_1 \times (m-1) \times \lambda / \mu.$$

Следовательно, первый и третий сокращаются:

$$P_3 = P_2 \times (m-2) \times \lambda / \mu = P_0 \times m \times (m-1) \times (m-2) \times \psi^3$$

и т.д.:

$$P_n = P_{n-1} \times (m-(n-1)) \times \lambda / \mu = P_0 \times m \times (m-1) \dots (m-(n-1)) \times \psi^n;$$

$$P_n = P_0 \times \psi^n \times m! / (m-n)!.$$

Используя очевидное равенство

$$\sum_{n=0}^m P_n = 1,$$

получим:

$$1 = P_0 \times \sum_{n=0}^m \psi^n m! / (m-n)!$$

Отсюда:

$$P_0 = \frac{1}{\sum_{n=0}^m \frac{\psi^n \times m!}{(m-n)!}}$$

Для установившегося режима работы системы средняя интенсивность поступления требований во входном потоке равна аналогичной характеристике выхода требований из канала обслуживания:

$$(m - N_{\text{sys}}) \lambda = (1 - P_0) \times \mu,$$

где N_{sys} – среднее число обслуживаемых требований, находящихся в системе. Из данного равенства можно легко найти среднее число требований (покупателей, рабочих, заданий, машин, неполадок), находящихся в системе, N_{sys} :

$$N_{\text{sys}} = m - (1 - P_0) / \psi.$$

Среднее же число требований (машин), находящихся в очереди, будет вычислено так:

$$N_{\text{оч}} = N_{\text{sys}} - (1 - P_0) = m - (1 - P_0) \times (1 / \psi + 1).$$

Пусть задан комплект машин системы «Кран–машины». Кран погружает за один рабочий цикл g_e 1 т груза. Грузоподъемность машины $g_a = 7$ т. Число машин, обслуживаемых краном, $m = 5$. Время рабочего цикла крана $t_c = 18$ с, а время возвращения машины к крану $t_o = 10$ мин. Тогда время погрузки одной машины составит:

$$t_{\text{погр}} = \frac{g_a}{g_e} \times t_c = \frac{7}{1} \times \frac{18}{60} = 2,1 \text{ мин.}$$

Интенсивность погрузки краном машин составит:

$$\mu = \frac{1}{t_{\text{погр}}} = \frac{60}{2,1} = 29 \text{ погрузок в час.}$$

Интенсивность же поступления машин на погрузку составит:

$$\lambda = \frac{1}{t_o} = \frac{60}{10} = 6 \text{ поездов в час.}$$

Коэффициент ψ , определяемый по формуле $\psi = \lambda / \mu$, в нашей задаче будет равен 0,207.

Вероятность простоя крана в этом случае составит:

$$P_0 = \frac{1}{1 + \sum_{n=1}^5 \frac{5!}{(5-n)!} \times \Psi^n} = 0,271.$$

А коэффициент использования крана составит:

$$K = 1 - P_0 = 1 - 0,271 = 0,729.$$

Среднее число машин, находящихся в системе:

$$N_{syst} = m - (1 - P_0) / \Psi = 1,477.$$

Среднее число машин, находящихся в очереди:

$$N_{och} = N_{syst} - (1 - P_0) = m - (1 - P_0) \times (1 / \Psi + 1) = 0,749.$$

Решение задачи имитационным методом

Для облегчения построения имитационной модели, изобразим графически процесс функционирования одноканальной замкнутой системы (рис. 5.24).

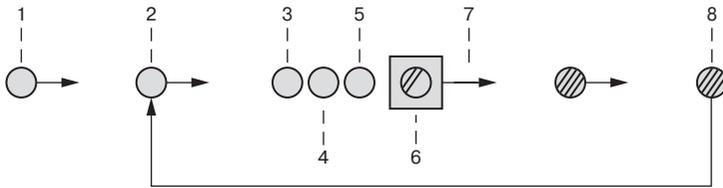


Рис. 5.24. Графическое изображение функционирования одноканальной замкнутой системы

Рассмотрим все события, происходящие в одноканальной замкнутой системе:

1. Генерирование требований, входящих в систему (**GENERATE** – Генерировать).
2. Доставка требования в очередь (**ADVANCE** – Продвинуть).
3. Вход требования в очередь (**QUEUE** – Очередь).
4. Проверка занятости канала обслуживания (**SEIZE** – Захватить).
5. Выход требования из очереди (**DEPART** – Выйти).
6. Обслуживание требования (**ADVANCE** – Продвинуть).
7. Освобождение канала обслуживания (**RELEASE** – Освободить).
8. Возвращение требования в систему (**TRANSFER** – Передать).

Поскольку требования возвращаются в систему, то мы имеем одноканальную замкнутую систему. Для имитационной модели рассмотрим одну из распространенных систем – «Кран–машины».

Создание имитационной модели

Создание имитационной модели начнем с построения заголовка модели, который может быть представлен, например, в таком виде:

```
* GPSSW                KRANM_e.GPS
*****
*   Моделирование одноканальной   *
*           замкнутой системы       *
*           с пуассоновскими потоками *
*****
```

Допустим, что в нашей задаче число машин, обслуживаемых краном, может изменяться. Тогда целесообразно использовать переменную, которая будет представлять число обслуживаемых машин, например сохраняемую величину под именем X\$MASH.

Первоначальное значение X\$MASH определяется оператором **INITIAL**:

```
INITIAL X$MASH,5
```

Особенности моделирования системы

Особенности моделирования данной системы заключаются в следующем:

1. Оператор **GENERATE** используется только для формирования числа машин, которые обслуживает кран. Этот режим использования оператора **GENERATE** предполагает, что поля А, В, С остаются пустыми, то есть ставятся соответственно три запятые, затем в поле Е указывается число машин, которые должен обслуживать кран.
2. Машины после доставки груза в пункт назначения снова возвращаются в систему для погрузки. Возвращение машины в систему происходит при вхождении ее в оператор **TRANSFER**, который используется в режиме безусловной передачи.
3. Время обращения машины моделируется следующим оператором:

```
AVTO ADVANCE (Exponential(1,0,10))
```

К нему направляются машины от оператора **TRANSFER ,AVTO**.

4. Возвращение машин в систему происходит до тех пор, пока время моделирования не превысит время моделирования системы. Определение времени моделирования основано на использовании простой модели измерения времени, состоящей из трех операторов:

```
GENERATE 480
TERMINATE 1
START 1
```

Этот сектор моделирует время работы системы в течение рабочей смены, равной 480 мин.

Моделирование потока машин в количестве $X\$MASH$ начинается с помощью оператора **GENERATE** – в нем определяется число машин, которые будут работать в системе. В нашем примере он будет выглядеть так:

```
GENERATE , , , X$MASH
```

Число работающих машин указывается в поле операнда D с помощью сохраняемой величины $X\$MASH$. Машина, пришедшая в систему, сначала попадает в оператор **ADVANCE**, который моделирует время ее приезда к каналу обслуживания (к крану). Он может быть записан так:

```
AVTO ADVANCE (Exponential(1,0,10))
```

В поле операнда A указывается средний интервал времени между прибытием к крану (каналу обслуживания) двух идущих одна за другой машин (требований, транзактов). В нашем примере среднее время прибытия требований составляет 20 мин. В поле операнда B дано отклонение времени поступления машин от среднего. В нашем примере отклонение от среднего времени прибытия машин на погрузку составляет 4 мин.

Затем машина встает в очередь для погрузки. Это можно промоделировать оператором **QUEUE**, который только в совокупности с соответствующим оператором **DEPART** собирает статистическую информацию о работе моделируемой очереди.

В нашем примере оператор **QUEUE** будет выглядеть так:

```
QUEUE POGR
```

В поле операнда A дается символьное или числовое имя очереди. Дадим нашей очереди имя **POGR** (Погрузка). Желательно, чтобы присваиваемое имя отражало суть описываемого элемента системы.

Следуя логике, машина может выйти из очереди только тогда, когда освободится кран (канал обслуживания). Для этого вводится оператор **SEIZE**, который определяет занятость канала обслуживания, и при освобождении последнего находящееся впереди требование выходит из очереди и идет в канал на обслуживание. Это может выглядеть так:

```
SEIZE KRAN
```

В поле операнда A дается символьное или числовое имя канала обслуживания. В нашей задаче каналу дано имя **KRAN** (Кран). Желательно, чтобы присваиваемое имя отражало суть описываемого элемента системы.

Выход машины из очереди на погрузку к крану фиксируется оператором **DEPART** с соответствующим названием очереди. В нашем примере это будет выглядеть так:

```
DEPART POGR
```

Далее должно быть промоделировано время погрузки машины, непосредственно обслуживаемой краном. Для моделирования этого процесса используется оператор **ADVANCE**, который в нашей задаче будет выглядеть так:

```
ADVANCE (Exponential(1,0,2.1))
```

В поле операнда А выполняется обращение к встроенному в систему экспоненциальному распределению с указанием среднего времени обслуживания машины – 2,1 мин. После обслуживания машины краном системе должно быть послано сообщение об освобождении канала обслуживания. Это делается с помощью оператора **RELEASE**, который в нашей задаче записывается так:

```
RELEASE KРАН
```

Следует особо подчеркнуть, что парные операторы **QUEUE** и **DEPART** для каждой очереди должны иметь одно и то же, но свое уникальное имя. Это же относится и к операторам **SEIZE** и **RELEASE**.

После погрузки краном машина направляется к месту разгрузки. Это может быть промоделировано оператором **TRANSFER**, например, так:

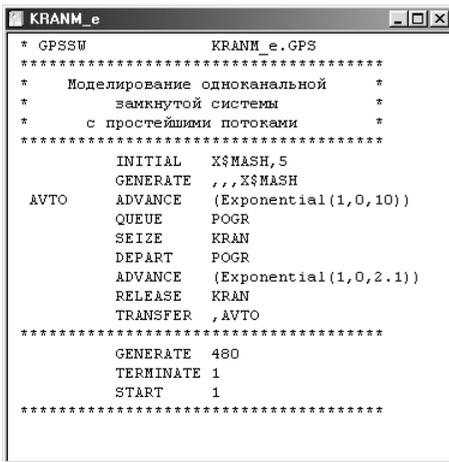
```
TRANSFER ,AVTO
```

Метка AVTO определяет оператор, к которому направляется требование. В нашей задаче это оператор **ADVANCE**, который уже был использован ранее. И в нашем примере он был представлен так:

```
AVTO ADVANCE (Exponential(1,0,10))
```

В данном случае этот оператор моделирует время возвращения машины в систему обслуживания. В поле операнда А выполняется обращение к встроенному в систему экспоненциальному распределению с указанием среднего интервала времени между прибытием к крану (каналу обслуживания) двух идущих одна за другой машин – 10 мин.

Окончательно наша программа будет выглядеть так, как показано на рис. 5.25.



```

KРАНM_e
* GPSSW          KРАНM_e.GPS
*****
*   Моделирование одноканальной   *
*   замкнутой системы             *
*   с простейшими потоками       *
*****
          INITIAL X$MASH,5
          GENERATE , , X$MASH
AVTO     ADVANCE (Exponential(1,0,10))
          QUEUE   POGR
          SEIZE   KРАН
          DEPART  POGR
          ADVANCE (Exponential(1,0,2.1))
          RELEASE KРАН
          TRANSFER ,AVTO
*****
          GENERATE 480
          TERMINATE 1
          START    1
*****

```

Рис. 5.25. Окно имитационной модели одноканальной замкнутой системы с простейшими потоками

Далее выполните следующие действия:

- щелкните по пункту **File** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **New** (Создать) выпадающего меню. Появится диалоговое

окно **Новый документ**;

- выделите пункт **Model** и щелкните по кнопке **ОК**. Появится окно модели, в котором введите данную программу.

Чтобы вызвать окно для представления имитационной модели в системе GPSSW, можно также нажать комбинацию клавиш **Ctrl+Alt+S**.

Подготовка к моделированию системы

Перед началом моделирования можно установить вывод тех параметров моделирования, которые нужны пользователю. Для этого:

- щелкните по пункту **Edit** (Правка) главного меню системы или нажмите комбинацию клавиш **Alt+E**. Появится выпадающее меню;
- щелкните по пункту **Settings** (Установки) выпадающего меню. Появится диалоговое окно **SETTINGS** для данной модели, в котором можно установить нужные выходные данные. Для нашего примера это окно может выглядеть так, как показано на рис. 5.26.

Наличие галочки в окошках говорит о том, что эта информация будет выведена

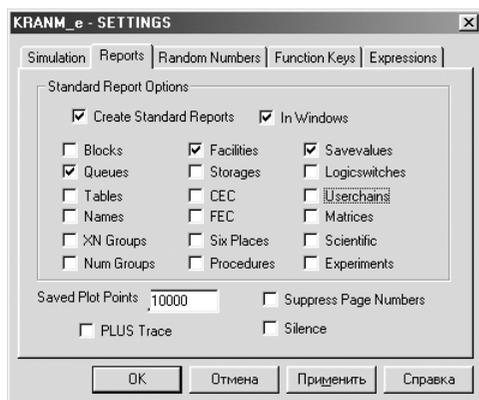


Рис. 5.26. Окно **SETTINGS** с установками для имитационной модели одноканальной замкнутой системы с простейшими потоками

в окне результатов моделирования. В нашем примере будет выведена информация по следующим объектам:

- **Facilities** (Каналы обслуживания);
- **Queues** (Очереди);
- **Savevalues** (Сохраняемые величины).

Моделирование системы

После создания имитационную модель необходимо оттранслировать и запустить на выполнение. Для этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню.

В имитационной модели имеется управляющая команда **START**, следовательно, исходная имитационная модель будет транслироваться, и если в ней нет ошибок, то начнется процесс моделирования системы.

Результаты моделирования представлены в окне **REPORT** (Отчет), показанном на рис. 5.27.

GPSS World Simulation Report - KRAMM_e.1.1									
Friday, January 04, 2002 19:47:58									
START TIME	END TIME	BLOCKS	FACILITIES	STORAGES					
0.000	480.000	10	1	0					
FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
KRAM	160	0.718	2.155	1	1	0	0	0	0
QUEUE	MAX CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE. (-0)	RETRY		
POGR	4	0	160	64	0.782	2.347	3.912	0	
SAVEVALUE	RETRY	VALUE							
MASH	0	5.000							

Рис. 5.27. Окно **REPORT** с результатами моделирования имитационной модели одноканальной замкнутой системы с простейшими потоками

В верхней строке указывается:

- **START TIME** (Начальное время) – 0.000;
- **END TIME** (Время окончания) – 480.000;
- **BLOCKS** (Число блоков) – 10;
- **FACILITIES** (Число каналов обслуживания) – 1;
- **STORAGES** (Число накопителей) – 0.

Ниже указываются результаты моделирования канала обслуживания (**FACILITY**) под именем KRAM:

- **ENTRIES** (Число входов) – 160;
- **UTIL.** (Коэффициент использования) – 0.718;
- **AVE. TIME** (Среднее время обслуживания) – 2.155;
- **AVAIL.** (Доступность) – 1;
- **OWNER** – 1;
- **PEND** – 0;
- **INTER** – 0;
- **RETRY** – 0;
- **DELAY** – 0.

Еще ниже указываются результаты моделирования очереди (QUEUE) по имени PQR:

- **MAX** (Максимальное содержание) – 4;
- **CONT.** (Текущее содержание) – 0;
- **ENTRY** (Число входов) – 160;
- **ENTRY(0)** (Число нулевых входов) – 64;
- **AVE.CONT.** (Среднее число входов) – 0.782;
- **AVE.TIME** (Среднее время) – 2.347;
- **AVE.(–0)** – 3.912;
- **RETRY** – 0.

Далее указывается значение сохраняемой величины (SAVEVALUE) по имени MASH:

- **RETRY** – 0;
- **VALUE** (Значение) – 5.000.

Одноканальная замкнутая СМО с равномерными потоками

Рассмотрим решение этой задачи для системы с равномерными потоками требований.

Постановка задачи

На монтажной площадке функционирует система «Кран–машины». Известно среднее время погрузки машины краном – 2,1 мин. Возможное отклонение от этого времени – 1 мин, среднее время, необходимое для транспортирования груза и возвращения машины, составляет 20 мин и отклонение от него – 3 мин. Число машин, обслуживаемых краном, может варьироваться в пределах 6–9. Требуется определить основные характеристики системы при различном числе машин, обслуживаемых каналом обслуживания – краном:

- коэффициент использования канала обслуживания;
- среднее время пребывания требования (машины) в канале;
- максимальное содержимое (длину) очереди, то есть число машин, ожидающих погрузки (наибольшее из возможных);
- среднее содержимое (длину) очереди, то есть среднее число машин, ожидающих обслуживания (погрузки);
- общее число входов в очередь, то есть общее число поступлений машин на обслуживание в течение смены;
- среднее время пребывания требования в очереди.

Поступление машин на обслуживание – равномерное с интервалом 17–23 мин. Время обслуживания машины также равномерное с интервалом 3–5 мин.

Выявление основных особенностей

Поскольку машины возвращаются в систему, то мы имеем одноканальную замкнутую производственную систему. Число машин в системе может варьироваться.

Создание имитационной модели процесса

Создание имитационной модели начнем с построения заголовка модели, который может быть представлен, например, в таком виде:

```
* GPSSW           KRANM.GPS
*****
*   Моделирование системы   *
*   "Кран-машины"          *
*****
```

Поскольку в нашей задаче число машин, обслуживаемых краном, изменяется, то целесообразно использовать переменную в виде сохраняемой величины X\$MASH.

Первоначальное значение X\$MASH определяется оператором **INITIAL**:

```
INITIAL  X$MASH, 6
```

Особенности моделирования данной системы заключаются в следующем:

1. Оператор **GENERATE** используется только для формирования числа машин, которые обслуживает кран. Этот режим использования оператора **GENERATE** предполагает, что поля A, B, C должны оставаться пустыми, то есть ставятся соответственно три запятые, затем в поле операнда D указывается число машин, которые должен обслуживать кран.
2. Машины после доставки груза в пункт назначения снова возвращаются в систему для погрузки. Возвращение машины в систему происходит при вхождении ее в оператор **TRANSFER**, который используется в режиме безусловной передачи.
3. Время обращения машины моделируется оператором **ADVANCE**. К нему направляются машины от оператора **TRANSFER ,AVTO**.
4. Возвращение машин в систему выполняется до тех пор, пока время моделирования не превысит время моделирования системы. Определение времени моделирования основано на использовании простой модели измерения времени, состоящей из трех операторов:

```
GENERATE  480
TERMINATE 1
START     1
```

Этот сектор моделирует время работы системы в течение рабочей смены, равной 480 мин.

Моделирование потока машин в количестве $X\$MASH$ начинается с помощью оператора **GENERATE** (Генерировать) – в нем определяется число машин, которые будут работать в системе. В нашем примере он будет выглядеть так:

```
GENERATE , , , X$MASH
```

Число работающих машин указывается в поле операнда *D* с помощью сохраняемой величины $X\$MASH$. Машина, пришедшая в систему, сначала попадает в следующий оператор:

```
AVTO ADVANCE 20,4
```

Он моделирует время приезда машины к каналу обслуживания. В поле операнда *A* указывается средний интервал времени между прибытием к крану (каналу обслуживания) двух идущих одна за другой машин (требований, транзактов). В нашем примере среднее время прибытия требований составляет 20 мин. В поле операнда *B* дано отклонение времени поступления машин от среднего. В нашем примере отклонение от среднего времени прибытия машин на погрузку составляет 4 мин.

Затем машина встает в очередь для погрузки. Это можно промоделировать оператором **QUEUE** (Очередь), который только в совокупности с соответствующим оператором **DEPART** (Выйти) собирает статистическую информацию о работе моделируемой очереди.

В нашем примере оператор **QUEUE** будет выглядеть так:

```
QUEUE POGR
```

В поле операнда *A* дается символьное или числовое имя очереди. Дадим нашей очереди имя *POGR* (Погрузка). Желательно, чтобы присваиваемое имя отражало суть описываемого элемента системы. Следуя логике, машина может выйти из очереди только тогда, когда освободится кран. Для этого вводится оператор **SEIZE** (Занять), который определяет занятость канала обслуживания, и при освобождении последнего находящегося впереди требование выходит из очереди и идет в канал на обслуживание. Это может выглядеть так:

```
SEIZE KRAN
```

В поле операнда *A* дается символьное или числовое имя канала обслуживания. В нашей задаче каналу дано имя *KRAN* (Кран). Желательно, чтобы присваиваемое имя отражало суть описываемого элемента системы.

Выход машины из очереди на погрузку к крану фиксируется оператором **DEPART** с соответствующим названием. В нашем примере это будет выглядеть так:

```
DEPART POGR
```

Далее должно быть промоделировано время погрузки машины, непосредственно обслуживаемой краном. Это время в нашем примере составляет $4 \pm 0,7$ мин. Для моделирования этого процесса используется оператор **ADVANCE** (Задержать), который в нашей задаче будет выглядеть так:

```
ADVANCE 4,0.7
```

После обслуживания машины краном системе должно быть послано сообщение об освобождении канала обслуживания. Это делается с помощью оператора **RELEASE** (Освободить). Он в нашей задаче записывается так:

```
RELEASE KRAN
```

Следует особо подчеркнуть, что парные операторы **QUEUE** и **DEPART** для каждой очереди должны иметь одно и то же, но свое уникальное имя. Это же относится и к операторам **SEIZE** и **RELEASE**.

После погрузки краном машина направляется к месту разгрузки. Это может быть промоделировано оператором **TRANSFER**, например, так:

```
TRANSFER ,AVTO
```

Метка **AVTO** определяет оператор, к которому направляется требование. В нашей задаче это оператор **ADVANCE**, который уже был использован ранее. И в нашем примере он был представлен так:

```
AVTO ADVANCE 20,4
```

В данном случае этот оператор моделирует время возвращения машины в канал обслуживания. В поле операнда **A** указывается средний интервал времени между прибытием к крану двух идущих одна за другой машин – 20 мин. В поле операнда **B** дано отклонение времени поступления машин от среднего. В нашем примере отклонение от среднего времени прибытия машин на погрузку составляет 4 мин.

Представление имитационной модели

Для представления имитационной модели выполните следующие действия:

- щелкните по пункту **File** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **New** (Создать) выпадающего меню. Появится диалоговое окно **Новый документ**;
- выделите пункт **Model** и щелкните по кнопке **OK**. Появится окно модели, в котором введите данную программу. Оно будет выглядеть так, как показано на рис. 5.28.

Чтобы вызвать окно для представления имитационной модели в системе GPSSW, можно также нажать комбинацию клавиш **Ctrl+Alt+S**.

Подготовка к моделированию

Перед началом моделирования можно установить вывод тех параметров моделирования, которые нужны пользователю. Для этого:

- щелкните по пункту **Edit** (Правка) главного меню системы или нажмите комбинацию клавиш **Alt+E**. Появится выпадающее меню;
- щелкните по пункту **Settings** (Установки) выпадающего меню. Появится диалоговое окно **SETTINGS** для данной модели, в котором можно установить нужные выходные данные. Для нашего примера это может выглядеть так, как показано на рис. 5.29.

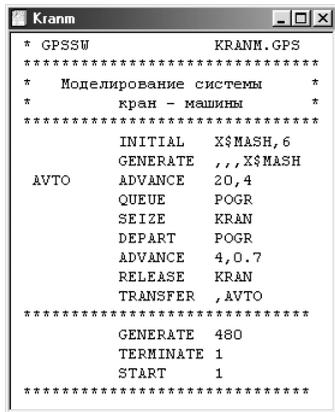


Рис. 5.28. Окно имитационной модели «Кран-машины»

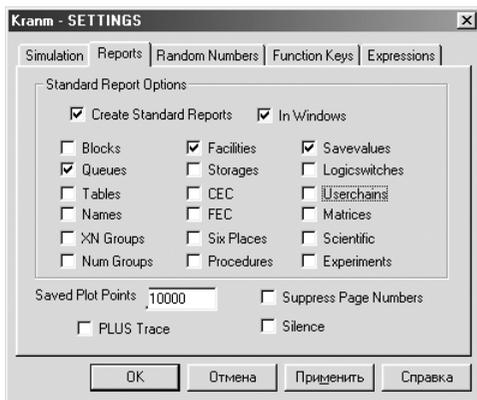


Рис. 5.29. Окно **SETTINGS** с установками для имитационной модели «Кран-машины»

Наличие галочки в окошках говорит о том, что эта информация будет выведена в окне результатов моделирования. В нашем примере будет выведена информация по следующим объектам:

- **Facilities** (Каналы обслуживания);
- **Queues** (Очереди);
- **Savevalues** (Сохраняемые величины).

Моделирование системы

После создания имитационную модель необходимо оттранслировать и запустить на выполнение. Для этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню.

В имитационной модели имеется управляющая команда **START**, следовательно, исходная имитационная модель будет транслироваться, и если в ней нет ошибок, то начнется процесс моделирования системы.

Результаты моделирования представлены в окне **REPORT** (Доклад), показанном на рис. 5.30.

```

Kranm.1.1 - REPORT
GPSS World Simulation Report - Kranm.1.1
Friday, January 04, 2002 19:59:10

START TIME      END TIME  BLOCKS  FACILITIES  STORAGES
0.000           480.000    10         1           0

FACILITY        ENTRIES  UTIL.    AVE. TIME  AVAIL.  OWNER  PEND  INTER  RETRY  DELAY
KRAN            107     0.885    3.970     1        1     0     0     0     0

QUEUE           MAX CONT. ENTRY  ENTRY (0)  AVE. CONT.  AVE. TIME  AVE. (-0)  RETRY
POGR            4       0     107       22        0.490     2.197     2.765     0

SAVEVALUE       RETRY     VALUE
MASH            0         6.000
  
```

Рис. 5.30. Окно **REPORT** с фрагментом результатов моделирования системы «Кран-машины»

В верхней строке указывается:

- **START TIME** (Начальное время) – 0.000;
- **END TIME** (Время окончания) – 480.000;
- **BLOCKS** (Число блоков) – 10;
- **FACILITIES** (Число каналов обслуживания) – 1;
- **STORAGES** (Число накопителей) – 0.

Ниже указываются результаты моделирования канала обслуживания (**FACILITY**) под именем KRAN:

- **ENTRIES** (Число входов) – 107;
- **UTIL.** (Коэффициент использования) – 0.885;
- **AVE. TIME** (Среднее время обслуживания) – 3.970;
- **AVAIL.** (Доступность) – 1;
- **OWNER** – 1;
- **PEND** – 0;
- **INTER** – 0;
- **RETRY** – 0;
- **DELAY** – 0.

Еще ниже указываются результаты моделирования очереди (**QUEUE**) по имени POGR:

- **MAX** (Максимальное содержание) – 4;
- **CONT.** (Текущее содержание) – 0;
- **ENTRY** (Число входов) – 107;
- **ENTRY(0)** (Число нулевых входов) – 22;
- **AVE.CONT.** (Среднее число входов) – 0.490;
- **AVE.TIME** (Среднее время) – 2.197;
- **AVE.(–0)** – 2.765;
- **RETRY** – 0.

Еще ниже указывается значение сохраняемой величины (SAVEVALUE) по имени MASH. При этом выводятся следующие результаты:

- **RETRY** – 0;
- **VALUE** (Значение) – 6.000.

Многоканальная замкнутая СМО с простейшими потоками

Постановка задачи

Допустим, задана многоканальная замкнутая система массового обслуживания с неограниченным временем ожидания и с простейшими потоками. Эта система наиболее полно соответствует действительности. Она характеризуется следующими особенностями:

- поступление требований в систему на обслуживание происходит по одному, то есть вероятность поступления двух и более требований в один момент времени очень мала, и ею можно пренебречь (поток требований ординарный);
- вероятность поступления последующих требований в любой момент времени не зависит от возможности поступления в предыдущие моменты – поток требований без последействия;
- поток требований стационарный.

Выявление основных особенностей

Функционирование многоканальной замкнутой системы массового обслуживания можно описать через все возможные ее состояния и через интенсивность перехода из одного состояния в другое.

Основными параметрами функционирования СМО являются вероятности состояния системы, то есть возможность наличия n требований (покупателей, рабочих, заданий, машин, неполадок) в системе – P_n . Так, вероятность P_0 характеризует состояние, когда в системе нет требований и все каналы обслуживания простаивают, P_1 – когда в системе находится только одно требование, и т.д.

Важным параметром функционирования системы массового обслуживания является также среднее число требований, находящихся в системе (то есть в очереди и на обслуживании), N_{sys} , и средняя длина очереди, N_{och} . Исходными параметрами, характеризующими систему массового обслуживания, являются:

- число каналов обслуживания (касс, компьютеров, подъемных кранов, ремонтных бригад) – N ;
- число требований (покупателей, рабочих, заданий, машин, неполадок) – m ;
- интенсивность поступления одного требования на обслуживание (то есть число возвращений требования в единицу времени) – λ ;
- интенсивность обслуживания требований – μ .

Интенсивность поступления на обслуживание одного требования определяется как величина, обратная времени возвращения требования в систему, – t_p :

$$\lambda = 1/t_p$$

Интенсивность обслуживания требований определяется как величина, обратная времени обслуживания одного требования, – t_o :

$$\mu = 1/t_o$$

Рассмотрим сначала решение задачи аналитическим методом. Для этого представим все возможные состояния системы массового обслуживания в виде размеченного графа состояний (рис. 5.31). Каждый прямоугольник графа определяет одно из всех возможных состояний, количественно оцениваемое вероятностью состояний P_n . Стрелки на графе указывают, в какое состояние система может перейти и с какой интенсивностью. При этом в многоканальной СМО необходимо различать два случая:

- число требований n , поступивших в систему, меньше числа каналов обслуживания N , то есть все требования находятся на обслуживании ($0 \leq n < N$);
- число требований n , поступивших в систему, больше или равно числу каналов обслуживания N ($N \leq n$), то есть N требований обслуживаются, а остальные r ожидают в очереди ($r = 1, 2, \dots, m - N$).

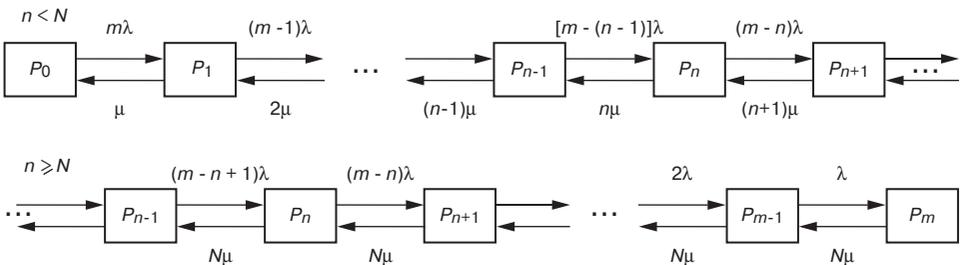


Рис. 5.31. Размеченный граф состояний многоканальной замкнутой СМО

Первый прямоугольник с вероятностью P_0 определяет состояние системы массового обслуживания, при котором все каналы обслуживания простаивают из-за отсутствия требований в ней. Из этого положения СМО может перейти только в состояние P_1 , и тогда в ней появится одно требование, так как входной поток требований ординарный. С интенсивностью μ система может перейти также из состояния P_1 в состояние P_0 , если единственное требование, находившееся в системе, было обслужено раньше, чем появилось новое, и т.д.

Аналитический метод решения

Ограничимся рассмотрением установившегося режима работы системы массового обслуживания, когда основные вероятностные характеристики СМО постоянны во времени. Тогда интенсивности входных и выходных потоков для каждого состояния будут сбалансированы. Эти сбалансированные интенсивности перехода состояния в другие состояния и обратно выглядят так.

Если $0 \leq n < N$, то:

$$P_0 \times m \times \lambda = P_1 \times \mu;$$

$$P_1 \times (\mu + (m-1) \times \lambda) = P_0 \times m \times \lambda + P_2 \times 2 \times \mu;$$

$$P_2 \times (2 \times \mu + (m-2) \times \lambda) = P_1 \times (m-1) \times \lambda + P_3 \times 3 \times \mu;$$

...

$$P_n \times (n \times \mu + (m-n) \times \lambda) = P_{n-1} \times (m-(n-1)) \times \lambda + P_{n+1} \times (n+1) \times \mu;$$

...

Если $N \leq n \leq m$, то:

...

$$P_n \times (N \times \mu + (m-n) \times \lambda) = P_{n-1} \times (m-(n-1)) \times \lambda + P_{n+1} \times N \times \mu;$$

...

$$P_{m-1} \times \lambda = P_m \times N \times \mu.$$

Обозначим величину λ / μ , как и раньше, через ψ и назовем ее коэффициентом загрузки.

Рассмотрим сначала первый случай, когда $0 \leq n < N$.

Из первого уравнения можно найти значение P_1 :

$$P_1 = P_0 \times m \times \lambda / \mu = P_0 \times m \times \psi.$$

Из второго уравнения найдем значение P_2 :

$$P_2 = P_1 / 2 + P_1 \times (m-1) \times \lambda / 2 \times \mu - P_0 \times m \times \lambda / 2 \times \mu.$$

Но первый член

$$P_1 = P_0 \times m \times \lambda / \mu.$$

Следовательно, первый и третий сокращаются:

$$P_2 = P_1 \times (m-1) \times \lambda / 2\mu = P_0 \times m \times (m-1) \times \psi^2 / 2.$$

Из третьего уравнения найдем значение P_3 :

$$P_3 = P_2 \times 2 / 3 + P_2 \times (m-2) \times \lambda / 3\mu - P_1 \times (m-1) \times \lambda / 3\mu.$$

Но первый член

$$P_2 = P_1 \times (m-1) \times \lambda / 2\mu.$$

Следовательно, первый и третий сокращаются:

$$P_3 = P_2 \times (m-2) \times \lambda / 3\mu = P_0 \times m \times (m-1) \times (m-2) \times \psi^3 / (1 \times 2 \times 3).$$

Аналогичные выражения можно получить и для других вероятностей состояний. Анализируя полученные выражения, вычисляем рекуррентное выражение для определения вероятности состояния системы, когда число требований, находящихся в системе, n , меньше числа каналов обслуживания, N :

$$P'_n = P_{n-1} \times (m - (n-1)) \times \lambda / (n \times \mu) = P_0 \times m \times (m-1) \dots (m - (n-1)) \times \psi^n / (1 \times 2 \times 3 \dots n);$$

$$P_n = P_0 \times \psi^n \times m! / ((m-n)! \times n!).$$

Рассмотрим теперь второй случай, когда $N \leq n \leq m$. В этой ситуации рекуррентное выражение для определения вероятности состояния системы будет записано в таком виде:

$$P''_n = P_0 \times \psi^n \times m! / ((m-n)! \times N! \times N^{n-N}).$$

Используем очевидное равенство

$$\sum_{n=0}^m P_n = 1,$$

отсюда

$$P_0 = \left(1 + \sum_{n=1}^{N-1} m! \psi^n / ((m-n)! n!) + \sum_{n=N}^m m! \psi^n / ((m-n)! N! N^{n-N})\right)^{-1}.$$

Допустим, что наша система имеет два канала обслуживания: $N = 2$. Интервал времени между поступлениями двух смежных требований составляет 10 мин. Среднее время обслуживания требования составляет 2 мин. Число обслуживаемых машин m равно 5. Требуется определить вероятность отсутствия требований в системе:

$$P_0 = \left(1 + \sum_{n=1}^{2-1} 5! \times 0,2^n / ((5-n)! n!) + \sum_{n=2}^5 5! \times 0,2^n / ((5-n)! 2! 2^{n-2})\right)^{-1} = 0,4.$$

На рис. 5.32 представлены основные события, которые возникают в процессе работы СМО.

Охарактеризуем каждое событие, возникшее в моделируемой системе:

1. Поступление требований в систему (**GENERATE**).
2. Вход требования в накопитель (**ENTER**).
3. Передача требования в один из свободных каналов обслуживания (**TRANSFER**).
4. Ожидание освобождения одного из каналов обслуживания (**SEIZE**).
5. Выход требования из накопителя (**LEAVE**).
6. Время обслуживания требования в канале обслуживания (**ADVANCE**).
7. Освобождение канала обслуживания (**RELEASE**).
8. Возвращение требования в систему (**TRANSFER**).

Создание имитационной модели процесса

Создание имитационной модели начнем с построения заголовка модели, который может быть представлен, например, в таком виде:

```
; GPSSW File                Smo3ze.GPS
*****
*   Моделирование трехканальной замкнутой   *
*   системы массового обслуживания         *
*****
```

Программу работы СМО можно представить в виде трех секторов.

В первом секторе указывается вместимость СМО. Это можно выполнить с помощью оператора **STORAGE** (Накопитель), который в нашем примере будет выглядеть так:

```
NAK STORAGE 3
```

Во втором секторе будем моделировать поток требований в систему и его обслуживание. Оператор **GENERATE** используем для формирования числа машин, которые обслуживают канал обслуживания. Этот режим использования оператора **GENERATE** предполагает, что поля A, B, C остаются пустыми, то есть ставятся соответственно три запятые, затем в поле операнда D указывается число машин, которые должны обслуживать каналы обслуживания:

```
GENERATE , , , 10
```

Далее машины поступают в канал обслуживания. При этом поток поступления машин на обслуживание простейший со средним интервалом 10 единиц времени. Это можно представить с помощью оператора **ADVANCE**, который в нашей задаче будет записан так:

```
MASH ADVANCE (Exponential(1,0,10))
```

Сбор статистической информации для многоканальной системы можно обеспечить с помощью операторов **ENTER** и **LEAVE**. Оператор **ENTER** может быть записан в таком виде:

```
ENTER NAK
```

В поле операнда А указано имя накопителя, вместимость которого должна быть заранее определена. Поскольку СМО многоканальная, то необходимо использовать оператор **TRANSFER** для обеспечения возможности направления требований к занятому каналу:

```
TRANSFER ALL, KAN1, KAN3, 3
```

Сначала требование направляется к оператору, имеющему символическую метку KAN1. Этим оператором является **SEIZE**, который записывается так:

```
KAN1 SEIZE CAN1
```

Если канал обслуживания с символьной меткой KAN1 занят, то требование направляется к следующему каналу, перешагивая через три оператора. 3 – это число, указанное в поле операнда D в операторе **TRANSFER**. Таким образом, следующим оператором будет:

```
SEIZE CAN2
```

Если и он будет занят, то требование снова перешагнет через три оператора и т.д., пока не найдется незанятый канал обслуживания. В свободном канале обслуживания требование будет обслужено. Но предварительно требование должно запомнить канал, в который оно попало на обслуживание. Для этого используется оператор **ASSIGN** (Присвоить) – с его помощью в параметре требования под номером 1 запоминается имя канала, в который требование пошло на обслуживание. В каждом канале имеется свой оператор **ASSIGN**. Например, для первого канала это присвоение будет выглядеть так:

```
ASSIGN 1, CAN1
```

Далее, после определения свободного канала и записи его имени с помощью оператора **TRANSFER** требование направляется на обслуживание. Это выглядит так:

```
TRANSFER , COME
```

Однако перед началом обслуживания должно быть подано сообщение о том, что требование оставило накопитель под именем NAK, в котором оно находилось. Это будет выглядеть так:

```
COME LEAVE NAK
```

После выхода из накопителя требование поступает в канал на обслуживание. Это действие выполняется с помощью оператора **ADVANCE**. Время обслуживания определяется так:

```
ADVANCE (Exponential(1,0,2))
```

После обслуживания требование выходит из канала обслуживания, и должен появиться сигнал о его освобождении. Это делается с помощью оператора **RELEASE** (Освободить):

```
RELEASE P1
```

Параметр требования под номером P1 содержит имя освобождаемого канала обслуживания. Далее требование входит в оператор **TRANSFER** с безусловным переходом для возвращения в систему. Он записывается так:

```
TRANSFER ,MASH
```

Таким образом, машины после обслуживания снова возвращаются в систему для обслуживания. Возвращение машин в систему выполняется до тех пор, пока время моделирования не превысит время моделирования системы. Определение времени моделирования системы выполняется в третьем секторе модели. Оно определяется с помощью простой модели измерения времени, состоящей из трех операторов:

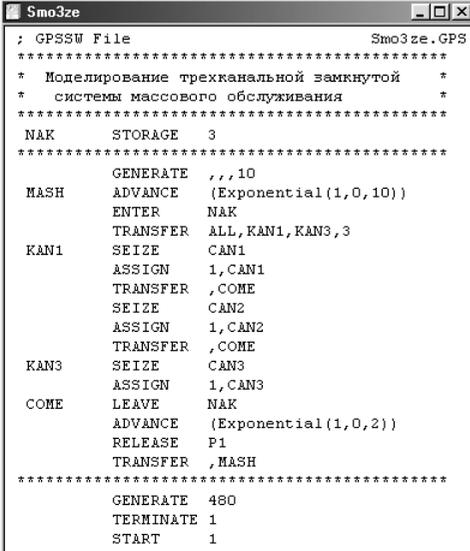
```
GENERATE 480
TERMINATE 1
START 1
```

Этот сектор моделирует время работы системы в течение рабочей смены, равной 480 мин.

Представление имитационной модели

Для представления имитационной модели выполните следующие действия:

- щелкните по пункту **File** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **New** (Создать) выпадающего меню. Появится диалоговое окно **Новый документ**;
- выделите пункт **Model** и щелкните по кнопке **ОК**. Появится окно модели, в котором введите данную программу. Оно будет выглядеть так, как показано на рис. 5.33.



```

Smo3ze
; GPSSW File                               Smo3ze.GPS
*****
* Моделирование трехканальной замкнутой *
* системы массового обслуживания          *
*****
NAK      STORAGE 3
*****
      GENERATE  ,, 10
MASH    ADVANCE  (Exponential(1,0,10))
      ENTER    NAK
      TRANSFER ALL, KAN1, KAN3, 3
KAN1    SEIZE   CAN1
      ASSIGN  1, CAN1
      TRANSFER ,COME
      SEIZE   CAN2
      ASSIGN  1, CAN2
      TRANSFER ,COME
KAN3    SEIZE   CAN3
      ASSIGN  1, CAN3
COME    LEAVE   NAK
      ADVANCE (Exponential(1,0,2))
      RELEASE P1
      TRANSFER ,MASH
*****
      GENERATE 480
      TERMINATE 1
      START    1

```

Рис. 5.33. Окно имитационной модели многоканальной замкнутой СМО с простейшими потоками

Чтобы вызвать окно для представления имитационной модели в системе GPSSW, можно также нажать комбинацию клавиш **Ctrl+Alt+S**.

Подготовка к моделированию

Перед началом моделирования можно установить вывод тех параметров, которые необходимо получить в процессе моделирования. Для этого:

- щелкните по пункту **Edit** (Правка) главного меню системы или нажмите комбинацию клавиш **Alt+E**. Появится выпадающее меню;
- щелкните по пункту **Settings** (Установки) выпадающего меню. Появится диалоговое окно **SETTINGS** для данной модели, в котором можно установить нужные выходные данные. Для нашего примера это окно может выглядеть так, как показано на рис. 5.34.

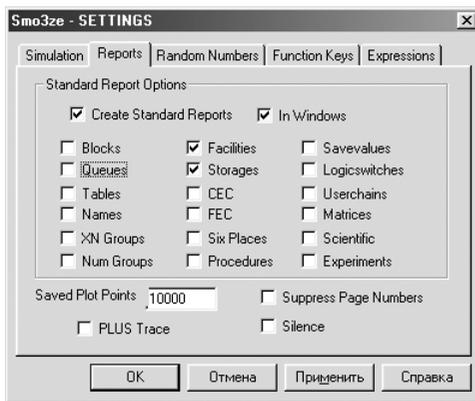


Рис. 5.34. Окно **SETTINGS**

с установками для имитационной модели многоканальной замкнутой СМО

Наличие галочки в окошках говорит о том, что эта информация будет выведена в окне результатов моделирования. В нашем примере будет выведена информация для следующих объектов:

- **Facilities** (Каналы обслуживания);
- **Storages** (Накопители).

Моделирование системы

После создания имитационной модели необходимо ее оттранслировать и запустить на выполнение. Для этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню.

Так как в модели есть управляющая команда **START**, то исходная имитационная модель будет транслироваться, и если в ней нет ошибок, то начнется процесс моделирования системы.

После выполнения программы появится окно с информацией о трансляции и выполнении – **JOURNAL** – и результаты работы программы в окне **REPORT** (рис. 5.35).

The screenshot shows a window titled "Smo3ze.1.1 - REPORT" containing the following text:

```

GPSS World Simulation Report - Smo3ze.1.1
Friday, January 04, 2002 21:29:33

START TIME          END TIME  BLOCKS  FACILITIES  STORAGES
0.000              480.000    18      3           1

FACILITY           ENTRIES  UTIL.   AVE. TIME AVAIL.  OWNER  PEND  INTER  RETRY  DELAY
CAN1               163     0.674   1.984  1        0    0    0      0      0
CAN2               127     0.580   2.192  1        0    0    0      0      0
CAN3                98     0.433   2.119  1        5    0    0      0      0

STORAGE            CAP.  REM.  MIN.  MAX.  ENTRIES  AVL.  AVE.C.  UTIL.  RETRY  DELAY
NAK                 3    3    0    3    388    1    0.169  0.056  0      0

```

Рис. 5.35. Окно **REPORT** с результатами моделирования многоканальной замкнутой СМО

В верхней строке указывается:

- **START TIME** (Начальное время) – 0.000;
- **END TIME** (Время окончания) – 480.000;
- **BLOCKS** (Число блоков) – 18;
- **FACILITIES** (Число каналов обслуживания) – 3;
- **STORAGES** (Число накопителей) – 1.

Ниже указываются результаты моделирования для всех трех каналов обслуживания (**FACILITY**) соответственно под именами CAN1, CAN2, CAN3:

- **ENTRIES** (Число входов) – 163, 127, 98;
- **UTIL.** (Коэффициент использования) – 0.674, 0.580, 0.433;
- **AVE. TIME** (Среднее время обслуживания) – 1.984, 2.192, 2.119;
- **AVAIL.** (Доступность) – 1, 1, 1;
- **OWNER** – 0, 0, 5;
- **PEND** – 0, 0, 0;
- **INTER** – 0, 0, 0;
- **RETRY** (Повтор) – 0, 0, 0;
- **DELAY** (Отказ) – 0, 0, 0.

Ниже указываются результаты функционирования накопителя (STORAGE) под именем НАК:

- **CAP.** (Capacity – Вместимость) – 3;
- **REM.** (Remove – Удален) – 3;
- **MIN.** (Минимальное содержимое) – 0;
- **MAX.** (Максимальное содержимое) – 3;
- **ENTRIES** (Число входов) – 388;
- **AVL.** (Доступность) – 1;
- **AVE.C.** (Средняя вместимость) – 0.169;
- **UTIL.** (Коэффициент использования) – 0.056;
- **RETRY** – 0;
- **DELAY** – 0.

Анализ работы многофазных СМО

Многофазная замкнутая СМО с равномерными потоками

Рассмотрим на конкретном примере моделирование двухфазной замкнутой системы массового обслуживания.

Постановка задачи

Рассмотрим работу системы «Краны–панелевозы». На домостроительном комбинате (ДСК) кран погружает панели в панелевоз, а на строительном объекте другой кран проводит разгрузку доставленных на площадку панелей. Известно среднее время погрузки машины краном на ДСК. Это время составляет 14 мин. Возможные отклонения от среднего времени составляют 1 мин. Среднее время, необходимое для транспортирования груза на строительную площадку, составляет 38 мин. А возможное отклонение от этого времени – 2 мин. Известно среднее время разгрузки панелевоза другим краном на строительной площадке, которое составляет 9 мин. Возможное отклонение от этого времени – 1 мин. Среднее время, необходимое для возвращения машины на ДСК, составляет 28 мин, а возможное отклонение от него – 2 мин. Число машин, занятых доставкой панелей на строительную площадку, равно 6. Требуется определить основные характеристики системы:

- коэффициенты использования первого и второго каналов обслуживания;
- среднее время пребывания требования (машины) в каналах;
- максимальное содержимое (длину) очереди, то есть число машин, ожидающих погрузки и разгрузки (наибольшее из возможных);
- среднее содержимое (длину) очереди, то есть среднее число машин, ожидающих обслуживания (погрузки, разгрузки);

- общее число входов в очередь, то есть поступлений машин на обслуживание в течение смены;
- среднее время пребывания машин в очередях.

Выявление основных особенностей

Поскольку машины возвращаются в систему, то мы имеем замкнутую производственную систему. Каждое требование – машина – проходит две фазы обслуживания: погрузку и разгрузку. Таким образом, мы имеем двухфазную систему обслуживания.

Создание имитационной модели

Создание имитационной модели начнем с построения заголовка модели, который может быть представлен, например, в таком виде:

```
* GPSSW          SMO_2f_z.gps
*****
* Моделирование двухфазной замкнутой *
* системы массового обслуживания   *
*****
```

Особенности моделирования данной системы заключаются в следующем.

1. Оператор **GENERATE** используется только для формирования числа машин, которые обслуживает кран. Этот режим использования оператора **GENERATE** предполагает, что поля А, В, С остаются пустыми, то есть ставятся соответственно три запятые, затем в поле операнда D указывается число машин, которые должен обслуживать кран.
2. Машины доставляют панели на строительную площадку, на которой проводится их разгрузка. Машины после разгрузки панелей снова возвращаются на ДСК на погрузку. Возвращение машины в систему происходит при вхождении ее в оператор **TRANSFER**, который используется в режиме безусловной передачи:

```
TRANSFER ,AVTO
```

3. Время возвращения машины на ДСК моделируется оператором **ADVANCE**:

```
ADVANCE 28,2
```

4. Возвращение машин в систему выполняется до тех пор, пока время моделирования не превысит время моделирования системы. Определение времени моделирования основано на использовании простой модели измерения времени, состоящей из трех операторов:

```
GENERATE 480
TERMINATE 1
START 1
```

Этот сектор моделирует время работы системы в течение рабочей смены, равной 480 мин.

Моделирование потока машин начинается с помощью оператора **GENERATE** (Генерировать) – в нем определяется число машин, которые будут работать в системе. В нашем примере он будет выглядеть так:

```
GENERATE , , , 6
```

Число работающих машин указывается в поле операнда D.

Далее панелевоз (требование) встает в очередь для погрузки. Это можно промоделировать оператором **QUEUE** (Очередь), который только в совокупности с соответствующим оператором **DEPART** (Выйти) собирает статистическую информацию о работе моделируемой очереди.

В нашем примере оператор **QUEUE** будет выглядеть так:

```
QUEUE POGR
```

В поле операнда A дается символьное или числовое имя очереди. Дадим нашей очереди имя POGR (Погрузка). Желательно, чтобы присваиваемое имя отражало суть описываемого элемента системы. Следуя логике, машина может выйти из очереди только тогда, когда освободится кран (канал обслуживания). Для этого вводится оператор **SEIZE** (Занять), который определяет занятость канала обслуживания, и при освобождении последнего находящееся впереди требование выходит из очереди и идет в канал на обслуживание. Это может выглядеть так:

```
SEIZE KRAN1
```

В поле операнда A дается символьное или числовое имя канала обслуживания. В нашей задаче каналу дано имя KRAN1 (Кран). Желательно, чтобы присваиваемое имя отражало суть описываемого элемента системы.

Выход машины из очереди фиксируется оператором **DEPART** с соответствующим названием очереди. В нашем примере это будет выглядеть так:

```
DEPART POGR
```

Далее должно быть промоделировано время погрузки машины, непосредственно обслуживаемой краном. Это время в нашем примере составляет 14 ± 1 мин. Для моделирования этого процесса используется оператор **ADVANCE** (Задержать), который в нашей задаче будет выглядеть так:

```
ADVANCE 14,1
```

После погрузки (обслуживания) машины краном системе должно быть послано сообщение об освобождении канала обслуживания. Это делается с помощью оператора **RELEASE** (Освободить), который в нашей задаче записывается так:

```
RELEASE KRAN1
```

Следует особо подчеркнуть, что парные операторы **QUEUE** и **DEPART** для каждой очереди должны иметь одно и то же, но свое уникальное имя. Это же относится и к операторам **SEIZE** и **RELEASE**.

После погрузки краном панелевоз направляется на строительную площадку к месту разгрузки, где также возможна очередь. Панелевоз (требование) встает в очередь для разгрузки. Это можно промоделировать оператором **QUEUE**, который в нашем примере будет выглядеть так:

```
QUEUE RAZGR
```

В поле операнда А дается символьное или числовое имя очереди. Дадим нашей очереди имя RAZGR (Разгрузка). Машина может выйти из очереди только тогда, когда освободится кран (канал обслуживания). Для этого вводится оператор **SEIZE**, который определяет занятость канала обслуживания, и при освобождении последнего находящееся впереди требование выходит из очереди и идет в канал на обслуживание. Это может выглядеть так:

```
SEIZE KRAN2
```

В поле операнда А дается символьное или числовое имя канала обслуживания. В нашей задаче каналу дано имя KRAN2.

Выход машины из очереди фиксируется оператором **DEPART** с соответствующим названием очереди. В нашем примере это будет выглядеть так:

```
DEPART RAZGR
```

Далее должно быть промоделировано время разгрузки машины краном. Это время в нашем примере составляет 9 ± 1 мин. Для моделирования этого процесса используется оператор **ADVANCE**, который в нашей задаче будет выглядеть так:

```
ADVANCE 9,1
```

После разгрузки (обслуживания) машины краном системе должно быть послано сообщение об освобождении канала обслуживания. Это делается с помощью оператора **RELEASE**, который в нашей задаче записывается так:

```
RELEASE KRAN2
```

Далее используется оператор **TRANSFER** для возвращения панелевоза на ДСК, например, так:

```
TRANSFER ,AVTO
```

Метка AVTO определяет оператор, к которому направляется требование. В нашей задаче это оператор **QUEUE**, который уже был использован ранее. И в нашем примере он был представлен так:

```
AVTO QUEUE POGR
```

Представление имитационной модели

Для представления имитационной модели выполните следующие действия:

- щелкните по пункту **File** главного меню системы. Появится выпадающее меню;

- щелкните по пункту **New** (Создать) выпадающего меню. Появится диалоговое окно **Новый документ**;
- выделите пункт **Model** и щелкните по кнопке **ОК**. Появится окно модели, в котором введите данную программу.

Окончательно наша программа будет выглядеть так, как показано на рис. 5.36.

```

Smo_2f_z
* GPSSM          SMO_2f_z.gps
*****
* Моделирование двухфазной замкнутой *
* системы массового обслуживания   *
*****
      GENERATE  ,, 6
AVTO  QUEUE    POGR
      SEIZE    KRAN1
      DEPART   POGR
      ADVANCE  14, 1
      RELEASE  KRAN1
      ADVANCE  38, 2
*****
      QUEUE    RAZGR
      SEIZE    KRAN2
      DEPART   RAZGR
      ADVANCE  9, 1
      RELEASE  KRAN2
      ADVANCE  28, 2
      TRANSFER , AVTO
*****
      GENERATE 480
      TERMINATE 1
      START    1

```

Рис. 5.36. Окно имитационной модели двухфазной многоканальной замкнутой СМО с равномерными потоками

Подготовка к моделированию системы

Перед началом моделирования можно установить вывод тех параметров моделирования, которые нужны пользователю. Для этого:

- щелкните по пункту **Edit** (Правка) главного меню системы или нажмите комбинацию клавиш **Alt+E**. Появится выпадающее меню;
- щелкните по пункту **Settings** (Установки) выпадающего меню. Появится диалоговое окно **SETTINGS** для данной модели, в котором можно установить нужные выходные данные. Для нашего примера это может выглядеть так, как показано на рис. 5.37.

Наличие галочки в окошках говорит о том, что эта информация будет выведена в окне результатов моделирования. В нашем примере будет выведена информация по следующим объектам:

- **Facilities** (Каналы обслуживания);
- **Queues** (Очереди);
- **Savevalues** (Сохраняемые величины).

Моделирование системы

После создания имитационную модель необходимо оттранслировать и запустить на выполнение. Для этого:

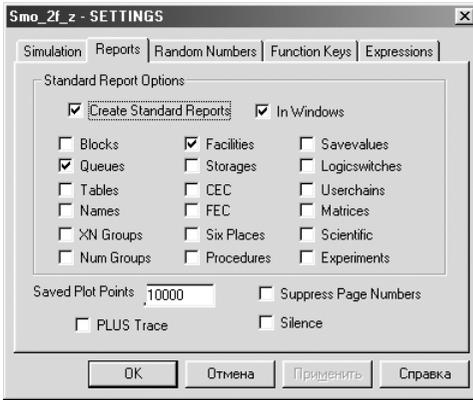


Рис. 5.37. Окно **SETTINGS** с установками для имитационной модели «Краны–панелевозы»

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню.

В имитационной модели имеется управляющая команда **START**, следовательно, исходная имитационная модель будет транслироваться, и если в ней нет ошибок, то начнется процесс моделирования системы.

Результаты моделирования представлены в окне **REPORT** (Отчет), показанном на рис. 5.38.

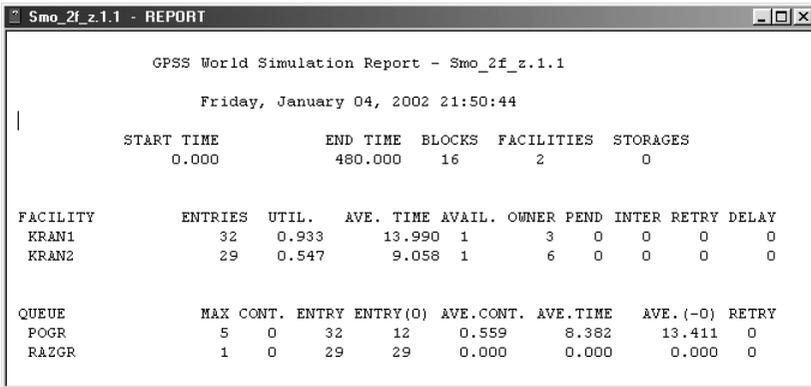


Рис. 5.38. Окно **REPORT** с фрагментом результатов моделирования системы «Краны–панелевозы»

В верхней строке указывается:

- **START TIME** (Начальное время) – 0.000;
- **END TIME** (Время окончания) – 480.000;
- **BLOCKS** (Число блоков) – 16;

- **FACILITIES** (Число каналов обслуживания) – 2;
- **STORAGES** (Число накопителей) – 0.

Ниже указываются результаты моделирования каналов обслуживания (FACILITY) соответственно под именами KРАН1 и KРАН2:

- **ENTRIES** (Число входов) – 32, 29;
- **UTIL.** (Коэф. использования) – 0.933, 0,547;
- **AVE. TIME** (Среднее время обслуживания) – 13.990, 9.058;
- **AVAIL.** (Доступность) – 1, 1;
- **OWNER** – 3, 6;
- **PEND** – 0, 0;
- **INTER** – 0, 0;
- **RETRY** – 0, 0;
- **DELAY** – 0, 0.

Еще ниже указываются результаты моделирования очередей (QUEUE) соответственно под именами ПОГР и РАЗГР:

- **MAX** (Максимальное содержание) – 5, 1;
- **CONT.** (Текущее содержание) – 0, 0;
- **ENTRY** (Число входов) – 32, 29;
- **ENTRY(0)** (Число нулевых входов) – 12, 29;
- **AVE.CONT.** (Среднее число входов) – 0.559, 0.000;
- **AVE.TIME** (Среднее время) – 8.382, 0.000;
- **AVE.(–0)** – 13.411, 0.000;
- **RETRY** – 0, 0.

Многофазная разомкнутая СМО с равномерными потоками

Рассмотрим на конкретном примере моделирование трехфазной разомкнутой системы массового обслуживания с равномерными потоками.

Постановка задачи

Рассмотрим процесс изготовления деталей на предприятии, требующий выполнения трех операций на определенных видах станков. Известно среднее время, через которое детали поступают в систему на изготовление – 23 мин, а также отклонение от среднего времени ± 3 мин. Известно среднее время, затрачиваемое на выполнение каждой операции, и возможные отклонения от среднего времени, которые соответственно равны: 24 ± 1 , 29 ± 3 и 12 ± 3 мин. Известно среднее время для передачи детали от одного станка к другому и возможные отклонения от среднего времени, которые соответственно равны 5 ± 2 и 6 ± 1 мин. Требуется определить основные характеристики системы:

- коэффициенты использования всех станков;
- среднее время пребывания детали у каждого станка (канала);
- максимальное число деталей, ожидающих изготовления у каждого станка;
- средний размер очереди, то есть среднее число деталей, ожидающих изготовления;
- общее число входов в очередь, то есть общее число поступлений деталей на обслуживание в течение смены;
- среднее время пребывания деталей в очередях.

Выявление основных особенностей

Поскольку детали после выполнения последней операции покидают систему, то мы имеем разомкнутую производственную систему. Каждое требование – деталь – проходит обработку на трех станках. Таким образом, мы имеем трехфазную систему обслуживания.

Создание имитационной модели

Создание имитационной модели начнем с построения заголовка модели, который может быть представлен, например, в таком виде:

```
* GPSSW                               SMO_3f_r.gps
*****
* Моделирование трехфазной разомкнутой *
* системы массового обслуживания      *
*****
```

Особенности моделирования данной системы заключаются в следующем:

- после выполнения операции на первом станке деталь поступает на обработку на второй. После выполнения операции на втором станке деталь поступает на обработку на третий. После выполнения операции на третьем станке деталь покидает систему;
- процесс изготовления деталей продолжается до тех пор, пока время моделирования не превысит время моделирования системы.

Этот сектор моделирует время работы системы в течение рабочей смены, равной 480 мин.

Моделирование потока деталей начинается с помощью оператора **GENERATE**. В нашем примере он будет выглядеть так:

```
GENERATE 23,3
```

В поле А указывается среднее время поступления деталей на обработку, равное 23 мин. Отклонение от среднего времени составляет ± 3 мин.

Далее деталь встает в очередь на выполнение первой операции на первом станке. Это можно промоделировать оператором **QUEUE** (Очередь), который только

в совокупности с соответствующим оператором **DEPART** (Выйти) собирает статистическую информацию о работе моделируемой очереди.

В нашем примере оператор **QUEUE** будет выглядеть так:

```
QUEUE OPER1
```

В поле операнда А дается символьное или числовое имя очереди. Дадим нашей очереди имя OPER1. Желательно, чтобы присваиваемое имя отражало суть описываемого элемента системы. Следуя логике, деталь может выйти из очереди только тогда, когда освободится станок (канал обслуживания). Для этого вводится оператор **SEIZE** (Занять), который определяет занятость канала обслуживания, и при освобождении последнего находящаяся впереди деталь (требование) выходит из очереди и идет в канал на обслуживание. Это может выглядеть так:

```
SEIZE STAN1
```

В поле операнда А дается символьное или числовое имя канала обслуживания. В нашей задаче каналу дано имя STAN1. Желательно, чтобы присваиваемое имя отражало суть описываемого элемента системы.

Выход детали из очереди фиксируется оператором **DEPART** с соответствующим названием очереди. В нашем примере это будет выглядеть так:

```
DEPART OPER1
```

Далее должно быть промоделировано время проведения первой операции, выполняемой первым станком. Это время в нашем примере составляет 24 ± 1 мин. Для моделирования этого процесса используется оператор **ADVANCE** (Задержать), который в нашей задаче будет выглядеть так:

```
ADVANCE 24,1
```

После обработки (обслуживания) детали станком системе должно быть послано сообщение об освобождении канала обслуживания. Это делается с помощью оператора **RELEASE** (Освободить), который в нашей задаче записывается так:

```
RELEASE STAN1
```

Следует особо подчеркнуть, что парные операторы **QUEUE** и **DEPART** для каждой очереди должны иметь одно и то же, но свое уникальное имя. Это же относится и к операторам **SEIZE** и **RELEASE**.

После выполнения первой операции на первом станке деталь направляется ко второму станку для второй операции, где также возможна очередь. Время, затрачиваемое на перемещение детали от первого станка ко второму, промоделируем с помощью оператора **ADVANCE**, который в нашей задаче будет выглядеть так:

```
ADVANCE 5,2
```

Далее деталь (требование) встает в очередь для выполнения второй операции на втором станке. Это можно промоделировать оператором **QUEUE**, который в нашем примере будет выглядеть так:

```
QUEUE OPER2
```

В поле операнда А дается символьное или числовое имя очереди. Дадим нашей очереди имя OPER2. Деталь может выйти из очереди только тогда, когда освободится второй станок (канал обслуживания). Для этого вводится оператор **SEIZE**, который определяет занятость канала обслуживания, и при освобождении последнего находящееся впереди требование выходит из очереди и идет в канал на обслуживание. Это может выглядеть так:

```
SEIZE STAN2
```

В поле операнда А дается символьное или числовое имя канала обслуживания. В нашей задаче каналу дано имя STAN2.

Выход детали из очереди фиксируется оператором **DEPART** с соответствующим названием очереди. В нашем примере это будет выглядеть так:

```
DEPART OPER2
```

Далее должно быть промоделировано время выполнения второй операции вторым станком. Это время в нашем примере составляет 29 ± 3 мин. Для моделирования этого процесса используется оператор **ADVANCE**, который в нашей задаче будет выглядеть так:

```
ADVANCE 29,3
```

После выполнения второй операции на втором станке системе должно быть послано сообщение об освобождении канала обслуживания. Это делается с помощью оператора **RELEASE**, который в нашей задаче записывается так:

```
RELEASE STAN2
```

Время, затрачиваемое на перемещение детали от второго станка к третьему, промоделируем с помощью оператора **ADVANCE**, который в нашей задаче будет выглядеть так:

```
ADVANCE 6,1
```

Далее деталь (требование) встает в очередь для выполнения третьей операции на третьем станке. Это можно промоделировать оператором **QUEUE**, который в нашем примере будет выглядеть так:

```
QUEUE OPER3
```

В поле операнда А дается символьное или числовое имя очереди. Дадим нашей очереди имя OPER3. Деталь может выйти из очереди только тогда, когда освободится третий станок (канал обслуживания). Для этого вводится оператор **SEIZE**, который определяет занятость канала обслуживания, и при освобождении последнего находящееся впереди требование выходит из очереди и идет в канал на обслуживание. Это может выглядеть так:

```
SEIZE STAN3
```

В поле операнда А дается символьное или числовое имя канала обслуживания. В нашей задаче каналу дано имя STAN3.

Выход детали из очереди фиксируется оператором **DEPART** с соответствующим названием очереди. В нашем примере это будет выглядеть так:

```
DEPART OPER3
```

Далее должно быть промоделировано время выполнения третьей операции третьим станком. Это время в нашем примере составляет 12 ± 3 мин. Для моделирования этого процесса используется оператор **ADVANCE**, который в нашей задаче будет выглядеть так:

```
ADVANCE 12,3
```

После выполнения третьей операции на третьем станке системе должно быть послано сообщение об освобождении канала обслуживания. Это делается с помощью оператора **RELEASE**, который в нашей задаче записывается так:

```
RELEASE STAN3
```

Определение времени моделирования основано на использовании простой модели измерения времени, состоящей из трех операторов:

```
GENERATE 480  
TERMINATE 1  
START 1
```

Представление имитационной модели

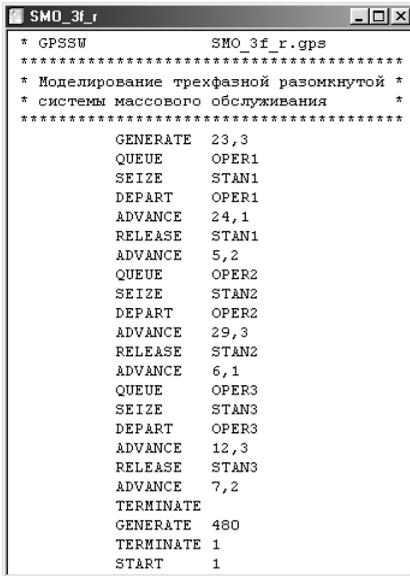
Для представления имитационной модели выполните следующие действия:

- щелкните по пункту **File** главного меню системы. Появится выпадающее меню;
- щелкните по пункту **New** (Создать) выпадающего меню. Появится диалоговое окно **Новый документ**;
- выделите пункт **Model** и щелкните по кнопке **OK**. Появится окно модели, в котором введите данную программу. Это может выглядеть так, как показано на рис. 5.39.

Подготовка к моделированию

Перед началом моделирования можно установить вывод тех параметров моделирования, которые нужны пользователю. Для этого:

- щелкните по пункту **Edit** (Правка) главного меню системы или нажмите комбинацию клавиш **Alt+E**. Появится выпадающее меню;
- щелкните по пункту **Settings** (Установки) выпадающего меню. Появится диалоговое окно **SETTINGS** для данной модели, в котором можно установить нужные выходные данные. Для нашего примера это окно может выглядеть так, как показано на рис. 5.40.



```

SMO_3f_r
* GPSSW          SMO_3f_r.gps
*****
* Моделирование трехфазной разомкнутой *
* системы массового обслуживания      *
*****
      GENERATE 23,3
      QUEUE   OPER1
      SEIZE   STAN1
      DEPART  OPER1
      ADVANCE 24,1
      RELEASE STAN1
      ADVANCE 5,2
      QUEUE   OPER2
      SEIZE   STAN2
      DEPART  OPER2
      ADVANCE 29,3
      RELEASE STAN2
      ADVANCE 6,1
      QUEUE   OPER3
      SEIZE   STAN3
      DEPART  OPER3
      ADVANCE 12,3
      RELEASE STAN3
      ADVANCE 7,2
      TERMINATE
      GENERATE 480
      TERMINATE 1
      START 1

```

Рис. 5.39. Окно имитационной модели трехфазной разомкнутой СМО

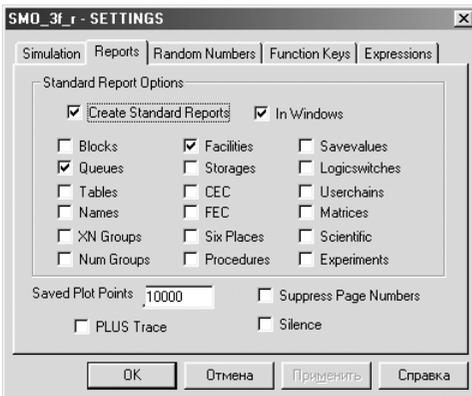


Рис. 5.40. Окно **SETTINGS** с установками для имитационной модели трехфазной разомкнутой СМО

Наличие галочки в окошках говорит о том, что эта информация будет выведена в окне результатов моделирования. В нашем примере будет выведена информация по следующим объектам:

- **Facilities** (Каналы обслуживания);
- **Queues** (Очереди);
- **Savevalues** (Сохраняемые величины).

Моделирование системы

После создания имитационную модель необходимо оттранслировать и запустить на выполнение. Для этого:

- щелкните по пункту **Command** главного меню системы или нажмите комбинацию клавиш **Alt+C**. Появится выпадающее меню;
- щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню.

В имитационной модели имеется управляющая команда **START**, следовательно, исходная имитационная модель будет транслироваться, и если в ней нет ошибок, то начнется процесс моделирования системы.

Результаты моделирования представлены в окне **REPORT** (Отчет), показанном на рис. 5.41.

```

SMO_3f_r.1.1 - REPORT
GPSS World Simulation Report - SMO_3f_r.1.1
Friday, January 04, 2002 22:05:53

START TIME      END TIME  BLOCKS  FACILITIES  STORAGES
0.000           480.000    22      3           0

FACILITY        ENTRIES  UTIL.   AVE. TIME  AVAIL.  OWNER  PEND  INTER  RETRY  DELAY
STAN1           19      0.948   23.945    1       20    0    0     0     1
STAN2           15      0.886   28.355    1       16    0    0     0     3
STAN3           14      0.336   11.517    1       15    0    0     0     0

QUEUE           MAX CONT. ENTRY  ENTRY(0)  AVE. CONT.  AVE. TIME  AVE. (-0)  RETRY
OPER1            1      1     20       1     0.355     8.511     8.959    0
OPER2            4      3     18       1     1.402    37.397    39.597    0
OPER3            1      0     14      14     0.000     0.000     0.000    0

```

Рис. 5.41. Окно **REPORT** с фрагментом результатов моделирования трехфазной разомкнутой СМО

В верхней строке указывается:

- **START TIME** (Начальное время) – 0.000;
- **END TIME** (Время окончания) – 480.000;
- **BLOCKS** (Число блоков) – 22;
- **FACILITIES** (Число каналов обслуживания) – 3;
- **STORAGES** (Число накопителей) – 0.

Ниже указываются результаты моделирования каналов обслуживания (**FACILITY**) соответственно под именами **STAN1**, **STAN2** и **STAN3**:

- **ENTRIES** (Число входов) – 19, 15, 14;
- **UTIL.** (Коэф. использования) – 0.948, 0.886, 0.336;
- **AVE. TIME** (Среднее время обслуживания) – 23.945, 28.355, 11.517;
- **AVAIL.** (Доступность) – 1, 1, 1;
- **OWNER** – 20, 16, 15;
- **PEND** – 0, 0, 0;
- **INTER** – 0, 0, 0;

- **RETRY** – 0, 0, 0;
- **DELAY** – 1, 3, 0.

Еще ниже указываются результаты моделирования очередей (QUEUE) соответственно под именами OPER1, OPER2 и OPER3:

- **MAX** (Максимальное содержание) – 1, 4, 1;
- **CONT.** (Текущее содержание) – 1, 3, 0;
- **ENTRY** (Число входов) – 20, 18, 14;
- **ENTRY(0)** (Число нулевых входов) – 1, 1, 14;
- **AVE.CONT.** (Среднее число входов) – 0.355, 1.402, 0.000;
- **AVE.TIME** (Среднее время) – 8.511, 37.397, 0.000;
- **AVE.(–0)** – 8.959, 39.597, 0.000;
- **RETRY** – 0, 0, 0.

Предметный указатель

В

Время

- машинное 49
- модельное 49
- реальное 49

Д

Диалоговое окно

- BLOCK ENTITIES 78, 86
- Edit Expression Window 33
- Open Table Window 90
- Print 21
- Print Setup 21
- Show Command 81, 89
- Simulation Command 80, 83, 86
- Start Command 78
- Stop Conditions 74, 89

И

Именованные величины 52

Имитационное моделирование 49

Имя 52

Интенсивность

- обслуживания требований 169, 230, 241, 275
- поступления требования на обслуживание 169, 229, 241, 275

К

Канал обслуживания 48

Кнопка

- About 28
- Continue 74
- Copy 28

Cut 28

Halt 74

New file 28

Open file 28

Paste 28

Place 74

Print 28

Remove 74

Save file 28

Step 74

Л

Логические переключатели 53

М

Математическая процедура

ABS 66

ATN 66

COS 66

EXP 66

INT 66

LOG 66

SIN 66

SQR 66

TAN 66

Матрица 53

Матричный элемент 53

Меню

всплывающее 14

выпадающее 14, 16

главное 14, 15

Метка 52

Модель

имитационная 48

исходная 69

Н

Накопитель 48

О

Окно сообщений

EXPRESSIONS 33

Оператор языка PLUS 58

BEGIN 65

CALL 65

DO...WHILE 65

END 65

EXPERIMENT 65

GOTO 65

IF...THEN...ELSE 65

RETURN 65

TEMPORARY 65

П

Параметры требования 50

Переменная

FVARIABLE 55

VARIABLE 54

BVARIABLE 55

пользователя 53

Поле 60

комментария 60

метки 60

операндов 60

оператора 60

Поток требований

входящий 48

выходящий 48

Процедура очереди

QueryXNAssemblySet 66

QueryXNExist 66

QueryXNM1 66

QueryXNParameter 66

QueryXNPriority 66

Пункт

выпадающего меню

About GPSS World 27

Cascade 27

Clear 26

Close 20

Conduct 26

Continue 26

Copy 22

Create Simulation 26

Custom 26

Cut 22

Delete 22

Delete Line 22

Entity Details 25

Exit 22

Expression Window 22

Find/Replace 25

Font 22

Go to Line 25

Halt 26

Help Topics 27

Insert Experiment 23

Insert GPSS Blocks 23

Internet 21

Mark 25

New 17

Next Bookmarks 25

Next Error 25

Notices 25

Open 20

Paste 22

Plot Window 22

Previous Error 25

Print 21

Print Setup 21

Repeat Last Command 26

Reset 26

Retranslate 26

Save 20

Save As 21

Select to Bookmark 25

Settings 23

Show 26

Simulation Clock 25

Simulation Snapshot 27

Simulation Window 27

Start 26

Step1 26

- Tile 27
- ToolBars 25
- Undo 22
- UnMark 25
- UnMark All 25
- главного меню
 - Command 25
 - Edit 22
 - File 16
 - Help 27
 - Search 24
 - View 25
 - Window 26

P

Распределение

- Beta 67
- Binomial 67
- Discrete Uniform 67
- Exponential 67
- Extreme Value A 67
- Extreme Value B 67
- Gamma 67
- Geometric 67
- Inverse Gaussian 67
- Inverse Weibull 67
- Laplace 67
- Logistic 67
- LogLaplace 67
- LogLogistic 67
- LogNormal 67
- Negative Binomial 67
- Normal 67
- Pareto 67
- Pearson Type V 67
- Pearson Type VI 67
- Poisson 67
- Triangular 67
- Uniform 67
- Weibull 67

C

Сервисная процедура

- ANOVA 67
- DoCommand 67

- Символы, специальные 52
- Система
 - массового обслуживания 48
 - меню 15
- Снимок 72
- События 49
- Сохраняемые величины 53
- Строковая
 - константа 53
 - процедура
 - Align 65
 - Catenate 65
 - Copies 65
 - Datatype 65
 - Find 65
 - Left 66
 - Length 66
 - Lowercase 66
 - Place 66
 - Polycatenate 66
 - Right 66
 - String 66
 - StringCompare 66
 - Substring 66
 - Trim 66
 - Uppercase 66
 - Value 66
 - Word 66

T

Требование 48

- активное 51
- выгруженное 51
- пассивное 51
- приостановленное 51
- удаленное 51

Ф

Файл 16

Ц

Цепь

- будущих событий 62
- задержки 63
- повторений 62

пользователя 61
прерываний 61
текущих событий 61

Ш

Шаблон блока GENERATE 24

Э

Элементы выражений 54

Кудрявцев Евгений Михайлович

GPSS World
Основы имитационного моделирования
различных систем

Главный редактор *Захаров И. М.*
editor-in-chief@dmkpress.ru

Выпускающий редактор *Морозова Н. В.*
Верстка *Трубачев М. П.*
Графика *Салимонов Р. В.*
Дизайн обложки *Дудатий А. М.*

ИД № 01903 от 30.05.2000

Подписано в печать 15.08.2003. Формат 70×100¹/₁₆.
Гарнитура «Петербург». Печать офсетная.
Усл. печ. л. 26. Тираж 2000 экз.
Зак. №

Издательство «ДМК Пресс», 105023, Москва, пл. Журавлева, д. 2/8.
Web-сайт издательства: www.dmkpress.ru.
Internet-магазин: www.abook.ru.