Раздел 6. ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ

«Если эксперимент удался, что-то здесь не так…»
(Первый закон Финэйгла)

6.1. Основы имитационного моделирования

6.1.1. Понятие имитационного моделирования

Статистическое моделирование – метод исследования сложных систем, основанный на описании процессов функционирования отдельных элементов в их взаимосвязи с целью получения множества частных результатов, подлежащих обработке методами математической статистики для получения конечных результатов. В основе статистического моделирования лежит метод статистических испытаний – метод Монте-Карло.

Имитационная модель – универсальное средство исследования сложных систем, представляющее собой логико-алгоритмическое описание поведения отдельных элементов системы и правил их взаимодействия, отображающих последовательность событий, возникающих в моделируемой системе.

Если статистическое моделирование выполняется с использованием имитационной модели, то такое моделирование называется *имитационным*.

Понятия «статистическое и имитационное моделирование» часто рассматривают как синонимы. Однако следует иметь в виду, что статистическое моделирование не обязательно является имитационным. Например, вычисление определённого интеграла методом Монте-Карло путем определения подынтегральной площади на основе множества статистических испытаний, относится к статистическому моделированию, но не может называться имитационным.

Наиболее широкое применение имитационное моделирование получило при исследовании сложных систем с дискретным характером функционирования, в том числе моделей массового обслуживания. Для описания процессов функционирования таких систем обычно используются временны*е* диаграммы.

Временная диаграмма – графическое представление последовательности событий, происходящих в системе. Для построения временны*х* диаграмм необходимо достаточно четко представлять взаимосвязь событий внутри системы. Степень детализации при составлении диаграмм зависит от свойств моделируемой системы и от целей моделирования.

Поскольку функционирование любой системы достаточно полно отображается в виде временной диаграммы, *имитационное моделирование можно рассматривать как процесс реализации диаграммы функционирования исследуемой системы на основе сведений о характере функционирования отдельных элементов и их взаимосвязи*.

Имитационное моделирование обычно проводится на ЭВМ в соответствии с программой, реализующей заданное конкретное логико-алгоритмическое описание. При этом несколько часов, недель или лет работы исследуемой системы могут быть промоделированы на ЭВМ за несколько минут. В большинстве случаев модель является не точным аналогом системы, а скорее её символическим отображением. Однако такая модель позволяет производить измерения, которые невозможно произвести каким-либо другим способом.

Имитационное моделирование обеспечивает возможность испытания, оценки и проведения экспериментов с исследуемой системой без каких-либо непосредственных воздействий на нее.

Первым шагом при анализе любой конкретной системы является выделение элементов, и формулирование логических правил, управляющих взаимодействием этих элементов. Полученное в результате этого описание называется **моделью системы**. Модель обычно включает в себя те аспекты системы, которые представляют интерес или нуждаются в исследовании.

Поскольку целью построения любой модели является исследование характеристик моделируемой системы, в имитационную модель должны быть включены средства сбора и обработки статистической информации по всем интересующим характеристикам, основанные на методах математической статистики.

6.1.2. Принципы организации имитационного моделирования

«Даже маленькая практика стоит большой теории» ***(Закон Буккера)***

Рассмотрим принципы имитационного моделирования на примере простейшей базовой модели в виде одноканальной системы массового обслуживания с однородным потоком заявок (рис.6.1), в которую поступает случайный поток заявок с интервалами между соседними заявками, распределёнными по закону $A(\tau)$, а длительность обслуживания заявок в приборе распределена по закону $B(\tau)$.

Процесс функционирования такой системы может быть представлен в виде временных диаграмм, на основе которых могут быть измерены и рассчитаны характеристики обслуживания заявок. Поскольку процессы поступления и обслуживания заявок в системе носят случайный характер, то для построения диаграмм необходимо иметь *генераторы случайных чисел*.

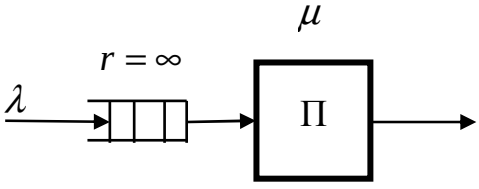


Рис. 6.1. СМО с накопителем неограниченной ёмкости

Положим, что в нашем распоряжении имеются генераторы случай-

ных чисел, формирующие значения соответствующих случайных величин *с заданными законами распределений* $A(\tau)$ и $B(\tau)$. Тогда можно построить временны*е* диаграммы, отображающие процесс функционирования рассматриваемой системы.

На рис.6.2 представлены четыре диаграммы, отображающие:

1) «**процесс поступления заявок**» в виде моментов t_i поступления заявок в систему, формируемых по правилу: $t_i = t_{i-1} + \tau_{a_i}$ $(t_0 = 0)$, где τ_{a_i} $(i = 1, 2, \ldots)$ – интервалы между поступающими в систему заявками, значения которых вырабатываются с помощью генератора случайных величин $A(\tau)$;

2) «**процесс обслуживания в приборе**», представленный в виде длительностей обслуживания τ_{bi}, которых вырабатываются с помощью генератора случайных величин $B(\tau)$, и моментов завершения обслуживания $t_i^{'}$ заявок в приборе, определяемых по следующему правилу:

$t_i^{'} = t_i + \tau_{b_i}$, если на момент поступления *i*-й заявки обслуживающий прибор был свободен;

$t_i^{'} = t_{i-1}^{'} + \tau_{b_i}$, если на момент поступления *i*-й заявки обслуживающий прибор был занят обслуживанием предыдущей заявки ($i = 1, 2, \ldots$; $t_0^{'} = 0$);

3) «**модельное или реальное время**», показывающее дискретное (скачкообразное) изменение времени в реальной системе, каждый момент которого соответствует одному из следующих событий: поступление заявки в систему или завершение обслуживания заявки в приборе; отметим, что в эти моменты времени происходит изменение состояния системы, описываемое числом заявок, находящихся в системе;

4) «**число заявок в системе**», описывающее состояние дискретной системы и изменяющееся по правилу: увеличение на 1 в момент поступления заявки в систему и уменьшение на 1 в момент завершения обслуживания.

При соблюдении выбранного временн*о*го масштаба представленные диаграммы позволяют путем измерения определить значения вероятностно-временны*х* характеристик функционирования моделируемой системы, в частности, как показано на второй диаграмме, время нахождения (пребывания) каждой заявки в системе: τ_{u_i} $(i = 1, 2, \ldots)$.

Очевидно, что время пребывания заявок в системе – величина случайная. В простейшем случае, применяя методы математической статистики, можно рассчитать два первых момента распределения времени пребывания:

- математическое ожидание:

$$u = \frac{1}{N} \sum_{i=1}^{N} \tau_{u_i} ;$$

- второй начальный момент:

$$u^{(2)} = \frac{1}{N-1}\sum_{i=1}^{N}\tau_{u_i}^2 ,$$

где N - количество значений времени пребывания заявок, полученных на диаграмме, то есть количество заявок, отображенных на диаграмме как прошедшие через систему и покинувшие её.

Отсюда легко могут быть получены значения дисперсии, среднеквадратического отклонения и коэффициента вариации времени пребывания заявок в системе.

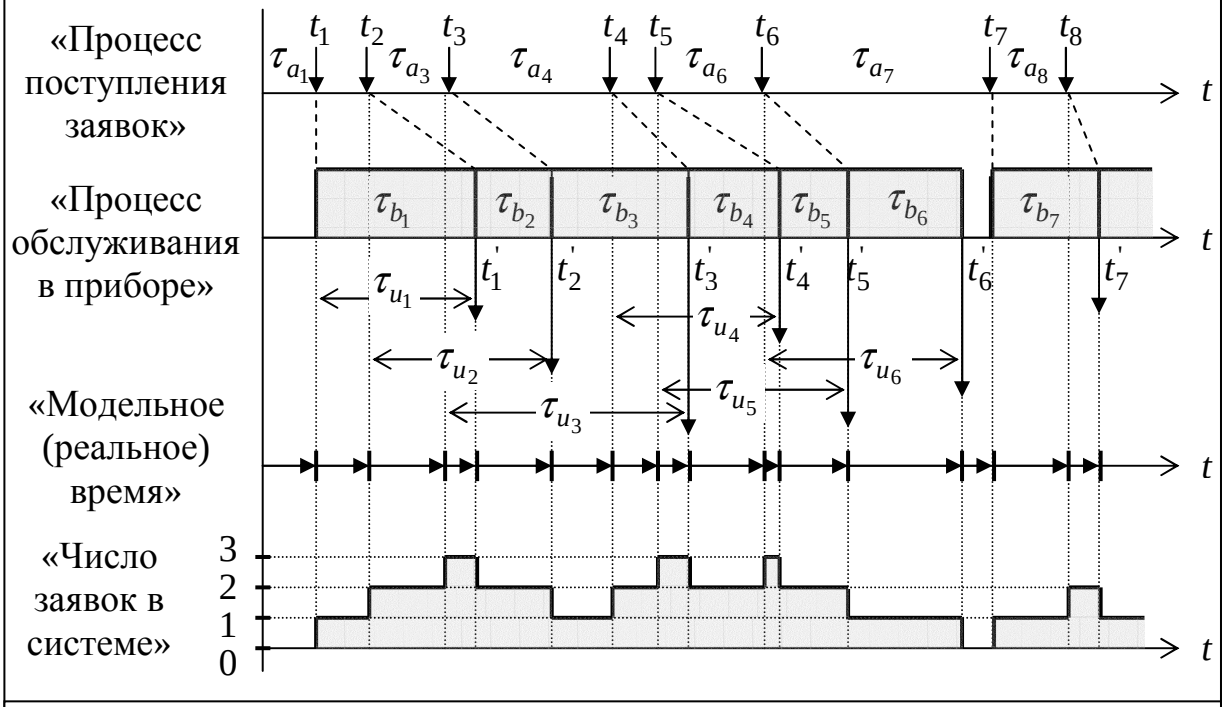


Рис.6.2. Диаграммы функционирования одноканальной СМО

На основе полученных с помощью временны*х* диаграмм значений времени пребывания заявок в системе можно построить гистограмму функции или плотности распределения времени пребывания.

Точность полученных числовых моментов распределения и качество гистограмм существенно зависит от количества значений N времени пребывания заявок, на основе которых они рассчитываются: чем больше N, тем точнее результаты расчета. Значение N может составлять от нескольких тысяч до десятков миллионов. Конкретное значение N зависит от многих факторов, влияющих на скорость сходимости результатов к истинному значению, основными среди которых при моделировании систем и сетей массового обслуживания являются законы распределений интервалов между поступающими заявками и длительностей обслуживания, загрузка системы, сложность модели, количество классов заявок и т.д.

Ясно, что построение вручную таких временных диаграмм с тысячами и более проходящими через систему заявками, нереально. В то же время, использование ЭВМ для реализации временны*х* диаграмм

позволяет существенно ускорить процессы моделирования и получения конечного результата. Поэтому, как сказано выше, имитационное моделирование можно рассматривать как процесс реализации диаграммы функционирования исследуемой системы.

Таким образом, имитационная модель представляет собой алгоритм реализации временной диаграммы функционирования исследуемой системы. Наличие встроенных в большинство алгоритмических языков генераторов случайных чисел значительно упрощает процесс реализации имитационной модели на ЭВМ. Однако при этом остаётся ряд проблем, требующих своего решения. Одна из них заключается в принципе реализации временной диаграммы и, связанной с ней, проблемой организации службы времени в имитационной модели.

В простейшем случае временная диаграмма может быть реализована следующим образом: сначала формируются моменты поступления всех заявок в систему, а затем для каждой заявки определяются длительности обслуживания в приборе и формируются моменты завершения обслуживания (выхода заявок из системы). Очевидно, что такой подход неприемлем, поскольку даже для нашей очень простой системы придётся хранить в памяти ЭВМ одновременно миллионы значений моментов поступления и завершения обслуживания заявок, а также других переменных, причём с увеличением количества классов заявок и количества обслуживающих приборов это число увеличится многократно.

Второй подход, который может быть предложен для реализации временной диаграммы, – пошаговое построение диаграммы. Для этого следует сформировать переменную для модельного времени и выбрать шаг Δt его изменения. В каждый такой момент времени необходимо проверять, какое событие (поступление в систему или завершение обслуживания заявки) произошло в системе за предыдущий интервал Δt.

Этот подход значительно сокращает потребность в памяти, поскольку в этом случае в каждый момент времени необходимо хранить в памяти ЭВМ значения параметров (моментов поступления и завершения обслуживания) только тех заявок, которые находятся в системе на данный момент времени.

Недостатки такого подхода очевидны. Во-первых, проблематичным является выбор длины интервала Δt. С одной стороны, интервал Δt должен быть как можно меньше для уменьшения методической погрешности моделирования, с другой стороны, интервал Δt должен быть как можно больше для уменьшения времени моделирования.

Наиболее эффективным подходом признан подход *с переменным шагом продвижения модельного времени*, который реализуется в соответствии с принципом «до ближайшего события». ***Принцип «продвижения модельного времени до ближайшего события»*** заключается в следующем. По всем процессам, параллельно протекающим в исследуемой системе, в каждый момент времени формируются моменты наступления «бли-

жайшего события в будущем». Затем модельное время продвигается до момента наступления ближайшего из всех возможных событий. В зависимости от того, какое событие оказалось ближайшим, выполняются те или иные действия. Если ближайшим событием является поступление заявки в систему, то выполняются действия, связанные с занятием прибора при условии, что он свободен, и занесение заявки в очередь, если прибор занят. Если же ближайшим событием является завершение обслуживания заявки в приборе, то выполняются действия, связанные с освобождение прибора и выбором на обслуживание новой заявки из очереди, если последняя не пуста. Затем формируется новый момент наступления этого же события. На третьей диаграмме «Модельное (реальное) время» (рис.6.2) продвижение времени в соответствии с этим принципом показано в виде стрелок.

Для того чтобы обеспечить правильную временну*ю* последовательность событий в имитационной модели, используются **системные часы**, хранящие значение текущего *модельного времени*. Изменение значения модельного времени осуществляется в соответствии с принципом «пересчёта времени до ближайшего события». Например, если текущее значение модельного времени равно 25, а очередные события должны наступить в моменты времени 31, 44 и 56, то значение модельного времени увеличивается сразу на 6 единиц и «продвигается» до значения 31. Отметим, что единицы времени в модели не обязательно должны быть конкретными единицами времени, такими как секунда или час. Основной единицей времени в модели можно выбрать любую единицу, которая позволит получить необходимую точность моделирования. Важно помнить, единицы времени выбираются исходя из требований пользователя к точности моделирования. Какая бы единица ни была выбрана, например миллисекунда или одна десятая часа, она должна неизменно использоваться во всей модели.

Кроме рассмотренной службы времени в имитационной модели необходимо реализовать процедуры, связанные с формированием потоков заявок и имитацией обслуживания, с организацией очередей заявок, с организацией сбора и статистической обработки результатов моделирования.

Таким образом, имитационное моделирование дискретных систем со стохастическим характером функционирования, таких как системы и сети массового обслуживания, предполагает использование ряда типовых процедур, обеспечивающих реализацию соответствующих имитационных моделей. К таким процедурам, в первую очередь, относятся следующие процедуры:

1) выработка (генерирование) случайных величин:
   - равномерно распределенных;
   - с заданным законом распределения;
2) формирование потоков заявок и имитация обслуживания;
3) организация очередей заявок;
4) организация службы времени;
5) сбор и статистическая обработка результатов моделирования.

6.2. Методы формирования случайных чисел

«То, что ищешь, найдешь только обыскав все»
(Закон Буба)

Функционирование элементов системы, подверженных случайным воздействиям, задается генераторами (датчиками) случайных чисел: *аппаратными* или *программными*. Генераторы случайных чисел в ЭВМ обычно реализуются программными методами, вырабатывающими псевдослучайные последовательности.

Псевдослучайными последовательностями называются вполне *детерминированные* числа, обладающие:

- *статистическими свойствами случайных чисел*, определяемых путем их проверки специальными тестами,
- *периодичностью*, то есть повторяемостью через определенные промежутки времени.

Количество случайных величин, вырабатываемых между двумя одинаковыми значениями, называется ***длиной периода генератора*** случайных величин.

При моделировании используются интервалы последовательностей псевдослучайных чисел, в которых нет ни одного числа, встречающегося более одного раза.

Для формирования случайных чисел с заданными законами распределений в качестве исходных используют случайные числа, выработанные программными генераторами равномерно распределенных случайных чисел в интервале (0,1), встроенные практически во все языки программирования. Специализированные программные средства, предназначенные для вероятностного моделирования, обычно имеют специальные встроенные процедуры генерирования случайных величин с разными законами распределений.

6.2.1. Формирование равномерно распределённых случайных величин

Для формирования равномерно распределённых случайных чисел в интервале (0; 1) могут использоваться следующие методы:

- метод квадратов;
- метод произведений;
- мультипликативный конгруэнтный метод;
- методы, представляющие модификации перечисленных методов.

Метод квадратов является одним из простейших методов и служит хорошей иллюстрацией принципа алгоритмического формирования равномерно распределённых случайных величин.

Алгоритм формирования равномерно распределённых случайных величин по методу квадратов заключается в выполнении следующих этапов:

1) выбирается некоторое исходное *n*-разрядное целое число, которое должно удовлетворять определённым условиям для получения качественного генератора случайных величин с максимально возможной длиной периода;

2) выбранное *n*-разрядное число возводится в квадрат, в результате чего получается целое число с вдвое большей разрядностью;

3) из полученного 2*n*-разрядного числа выделяются *n* средних разрядов, которые рассматриваются как дробная часть случайного числа, равномерно распределённого в интервале (0; 1);

4) выделенные на предыдущем этапе *n* средних разрядов рассматриваются как новое исходное *n*-разрядное целое число;

5) повторяются этапы 2 – 4.

Проиллюстрируем метод квадратов на следующем примере.

Пример 1. Для простоты будем оперировать десятичными числами, а не двоичными, как это реализуется в программных генераторах.

Пусть выбрано некоторое исходное четырехразрядное целое число, равное 7153. Результаты применения описанного алгоритма представлены в виде следующей таблицы:

Исх.число	Квадрат	Случайное число
7153	51 **1654** 09	0,1654
1654	02 **7357** 16	0,7357
7357	54 **1254** 49	0,1254
1254	01 **5725** 16	0,5725
5725	32 **7756** 25	0,7756
7756	60 **1555** 36	0,1555

Очевидно, что максимальная длина периода генератора, то есть максимальное количество неповторяющихся случайных чисел определяется количеством разрядов в дробной части. В нашем примере максимально возможная длина периода равна 9999 (от 0,0001 до 0,9999). Однако в действительности длина периода меньше максимально возможной и зависит от исходного целого числа. Неудачно выбранное значение исходного числа может привести к двум неприятностям: маленькой длине периода или даже к вырождению генератора, когда значения случайной величины начинают повторяться, как это показано в примере 2.

Пример 2. Исходное четырехразрядное целое число = 1357

Исх.число	Квадрат	Случайное число
1357	01 **8414** 49	0,8414
8414	70 **7953** 96	0,7953
7953	63 **2502** 09	0,2502
2502	06 **2600** 04	0,2600
2600	06 **7600** 00	0,7600
7600	57 **7600** 00	0,7600

Метод произведений аналогичен методу квадратов. Отличие состоит в том, что перемножаются два *n*-разрядных целых числа, одно из которых, называемое *ядром* или *множителем*, не меняется, а второе, называемое *множимым*, формируется из *n* последних (правых) разрядов полученного 2*n*-разрядного числа, представляющего собой произведение ядра и множимого. Естественно, что вначале, как и в методе квадратов, необходимо грамотно выбрать исходные значения ядра и множителя.

Пример 3. Ядро = 5167; множитель = 3729

Множимое	Произведение	Случайное число
3729	19 2677 43	0,2677
7743	40 0080 81	0,0080
8081	41 7545 27	0,7545
4527	23 3910 09	0,3910
1009	05 2135 03	0,2135
3501	….	…

Здесь, в отличие от предыдущего примера, в качестве следующего значения множителя выбираются не средние разряды полученного произведения, а последние *n* разрядов произведения.

Конгруэнтные методы генерирования случайных чисел получили наиболее широкое распространение для формирования на ЭВМ псевдослучайных последовательностей [13].

Два целых числа *a* и *b* называются ***конгруэнтными*** (сравнимыми) по модулю *m*, где *m* – целое число, если разность $(a-b)$ делится на *m* без остатка, а числа *a* и *b* дают одинаковые остатки от деления на *m*. Например, 2568 и 148 (по модулю 10), 1746 и 511 (по модулю 5), 6493 и 2221 (по модулю 2) и т.д.

Конгруэнтные методы описываются в виде рекуррентного соотношения следующего вида:

$$X_{i+1} = \lambda X_i + \mu (\mathrm{mod}\, m) \qquad (i = 0, 1, 2, ...),$$

где X_i, λ, μ, m – неотрицательные целые числа; X_0 – начальное значение псевдослучайной последовательности; λ – множитель; μ – аддитивная константа; m – модуль.

Каждое новое значение X_{i+1} псевдослучайной последовательности представляет собой целочисленный остаток от деления на модуль m суммы произведения предыдущего значения X_i на множитель λ и аддитивной константы μ. Последовательность псевдослучайных чисел в интервале (0; 1) формируется путем деления полученных целочисленных значений X_i на модуль m: $x_i = X_i / m \quad (i = 1, 2, ...)$.

Описанный метод генерирования псевдослучайных чисел получил название **смешанного конгруэнтного метода**.

В некоторых случаях используется более простой метод генерирования псевдослучайных чисел, представляющий собой частный случай смешанного метода, когда $\mu = 0$, и получивший название ***мультипликативного конгруэнтного метода***. В этом случае рекуррентное соотношение имеет вид:

$$X_{i+1} = \lambda X_i (\mathrm{mod}\, m) \qquad (i = 0, 1, 2, ...).$$

На каждом шаге полученное случайное число (множимое) умножается на некоторое постоянное число (множитель) и затем делится на другое постоянное число (делитель). В качестве нового случайного числа принимается остаток от деления, который служит дробной частью случайного числа, равномерно распределённого в интервале (0; 1).

Пример 4. Первое постоянное число (множитель) = 1357; второе постоянное число (делитель) = 5689.

Исходное число	Произведение	Частное, целая часть	Остаток	Случайное число
1357	1 8414 49	323	3902	0,3902
3902	5 2950 14	930	4244	0,4244
4244	5 7591 08	1012	1840	0,1840
1840	…	…	…	…

6.2.2. Проверка генераторов равномерно распределенных псевдослучайных чисел

«Когда не знаешь, что именно ты делаешь, де-лай это тщательно» ***(Правило для лаборантов)***

Достоверность и точность результатов имитационного моделирования в значительной степени определяется качеством используемых в моделях программных генераторов псевдослучайных последовательностей.

Проверка генераторов равномерно распределенных псевдослучайных чисел предполагает формирование большой совокупности или, как говорят, представительной выборки случайных чисел и выполнение множества проверочных тестов, позволяющих оценить качество генераторов.

Различают три вида проверки программных генераторов равномерно распределенных псевдослучайных чисел:

- на периодичность;
- на случайность;
- на равномерность.

Проверка на периодичность требует обязательного определения длины периода, что в значительной степени определяет качество генератора случайных чисел. Чем больше длина периода, тем генератор более качественный.

Проверка на случайность. При проверке на случайность программных генераторов *двоичных* случайных чисел можно использовать

совокупность тестов, а именно тесты проверки:

- частот;
- пар;
- комбинаций;
- серий;
- корреляции.

Тест проверки частот предполагает разбиение диапазона распределения на несколько интервалов и подсчет количества (частот или вероятностей) попаданий случайных чисел в выделенные интервалы.

Тест проверки пар заключается в подсчете количества "1" для каждого *разряда* всей совокупности выработанных генератором двоичных случайных чисел. Очевидно, количество "1" во всех разрядах должно составлять примерно 50% от количества выработанных генератором случайных чисел.

Тест проверки комбинаций сводится к подсчету "1" в случайных числах, количество которых в среднем должно составлять половину от количества разрядов.

Тест проверки серий заключается в подсчете количества различных длин последовательностей одинаковых значений (1 или 0).

Тест проверки корреляции заключается в определении коэффициента корреляции между последовательностями случайных чисел, вырабатываемых двумя разными генераторами.

Проверка на равномерность. При проверке на равномерность можно использовать тест проверки частот, так как гистограмма частот хорошо отражает равномерность распределения случайных чисел по всему диапазону изменения.

6.2.3. Методы формирования псевдослучайных чисел с заданным законом распределения

Методы формирования псевдослучайных чисел с заданным законом распределения основаны на использовании генераторов равномерно распределённых случайных величин. При этом наибольшее распространение получили следующие методы:

- аналитический (метод обратной функции);
- табличный;
- метод композиций, основанный на функциональных особенностях генерируемых распределений.

Аналитический метод заключается в построении математической зависимости, связывающей значения случайной величины с заданным законом распределения со значениями случайной величины, распределённой равномерно в интервале (0; 1).

Суть аналитического метода иллюстрируется на графике (рис.6.3). Пусть задана некоторая функция распределения $F(x)$, значения которой лежат в интервале (0; 1). Положим, что имеется генератор равномерно

распределённых в том же интервале случайных чисел: $S \in (0;1)$. Тогда, генерируя последовательность значений $s_1, s_2, s_3, \ldots$ и откладывая их по оси ординат, можно найти соответствующие значения $x_1, x_2, x_3, \ldots$ случайной величины *X*, распределённой по заданному закону $F(x)$.

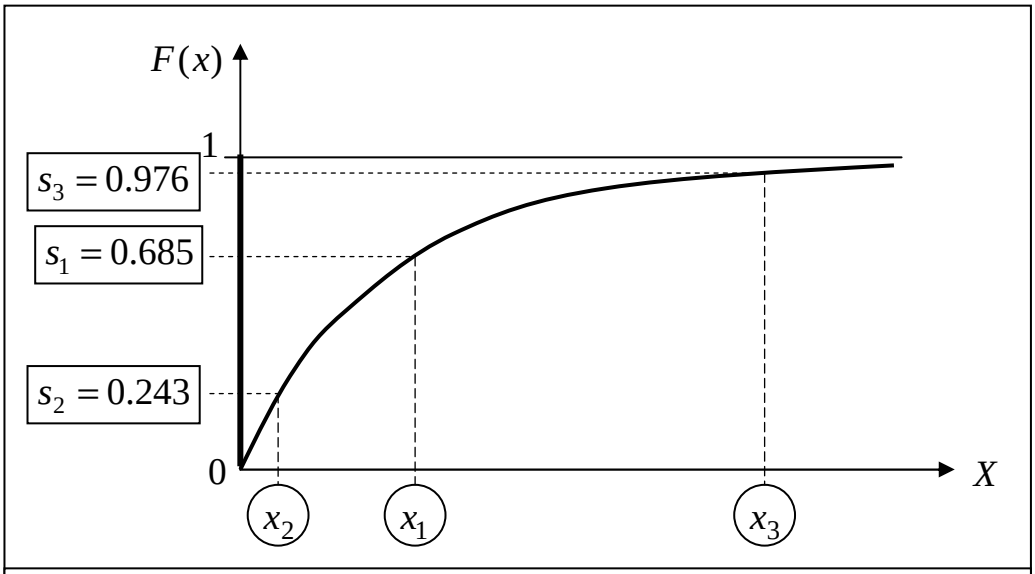


Рис.6.3. Графическая иллюстрация аналитического метода

Выведем аналитическую зависимость для расчета значений случайной величины, распределённой по экспоненциальному закону с функцией: $F(x) = 1 - e^{-\alpha x}$, где $\alpha > 0$ – параметр экспоненциального распределения.

Для этого, в соответствии с выше изложенным, необходимо решить уравнение $F(x) = s$, где s – значение равномерно распределённой в интервале (0; 1) случайной величины. Таким образом, имеем:

$1 - e^{-\alpha x} = s$ или $e^{-\alpha x} = 1 - s$.

Логарифмируя левую и правую части последнего выражения, после некоторых преобразований получим:

$$x = -\frac{1}{\alpha}\ln(1-s) \quad \text{или} \quad \boxed{x = -\frac{1}{\alpha}\ln s}.$$

Отметим, что $\frac{1}{\alpha}$ представляет собой математическое ожидание экспоненциально распределённой случайной величины.

Оба полученных выражения равнозначны, поскольку с позиций теории вероятностей случайные величины s и $(1-s)$ распределены по одному и тому же равномерному закону в интервале (0; 1). В то же время, последнее выражение предпочтительнее, поскольку не требует выполнения «лишней» операции вычитания, что позволяет уменьшить время моделирования с учетом того, что в процессе моделирования генерируются миллионы случайных чисел.

Достоинства аналитического метода:

- высокая точность метода;
- не требуется составления и хранения в памяти таблиц, как в табличном методе.

Недостатки аналитического метода:

- метод распространяется только на те функции, которые позволяют вычислить интеграл от функции плотности аналитически;
- использование численных методов вычисления интегралов приводит к погрешностям и большим затратам машинного времени;
- выражение, используемое для вычислений, содержит в себе функции вычисления логарифмов, возведения в степень, вычисления радикалов, что требует значительных затрат машинного времени.

Табличный метод заключается в формировании таблицы, содержащей пары чисел: значение функции распределения $F(x)$ и соответствующее ему значение x случайной величины. В качестве аргумента при обращении к таблице используется значение $s \in (0;1)$ равномерно распределенной случайной величины S, задающее значение функции распределения $F(x)$, а в качестве функции – значение x случайной величины X с соответствующим законом распределения $F(x)$.

Значение случайного числа, находящегося между узлами табуляции, обычно рассчитывается методом *линейной интерполяции*.

В ранних версиях GPSS для генерирования случайных чисел, распределённых по экспоненциальному закону, использовался табличный генератор со следующими значениями функции распределения $F(x)$ (от 0 до 0.9997) и соответствующими им значениями случайной величины x (от 0 до 8):

$F(x)$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.75	0.8	0.84	0.88
x	0	0.104	0.222	0.335	0.509	0.69	0.915	1.2	1.38	1.6	1.83	2.12

0.9	0.92	0.94	0.95	0.96	0.97	0.98	0.99	0.995	0.998	0.999	0.9997
2.3	2.52	2.81	2.99	3.2	3.5	3.9	4.6	5.3	6.2	7	8

Представленные в таблице значения $F(x)$ и x соответствуют экспоненциальному распределению *с математическим ожиданием, равным единице*. Если математическое ожидание экспоненциально распределённой случайной величины отличается от 1, то полученное с помощью этой таблицы значение случайной величины умножается на значение математического ожидания.

Заметим, что табулирование функции выполнено с переменным шагом: в начале таблицы шаг изменения аргумента (значений экспоненциальной функции распределения $F(x)$) равен 0.5, а в конце – 0.0007. Это обусловлено необходимостью обеспечить приемлемую методическую погреш-

ность, возникающую в результате линейной интерполяции при вычислении значений случайной величины, находящихся между узлами табуляции.

Достоинства табличного метода:

- существует принципиальная возможность построения таблицы для формирования случайных последовательностей с любым законом распределения, в том числе полученного экспериментальным путём;
- можно обеспечить любую заданную точность генерирования случайных чисел за счет увеличения количества интервалов табуляции (уменьшения шага табуляции);
- для генерирования случайных величин с заданным законом распределения вероятностей требуется только генератор равномерно распределенных случайных чисел и выполнение несложных операций, занимающих мало времени.

Недостатки табличного метода:

- значительные затраты памяти для хранения большого числа таблиц с разными законами распределений;
- наличие методической погрешности, обусловленной применением линейной интерполяции для определения значений случайных чисел, находящихся между узлами табуляции;
- для уменьшения методической погрешности формирования случайных последовательностей при использовании линейной интерполяции следует увеличивать количество точек табуляции, что приводит к увеличению размера таблиц и, как следствие, к дополнительным затратам памяти и времени;
- в связи с неодинаковой скоростью изменения функции распределения для обеспечения высокой точности формирования случайных последовательностей табулирование должно выполняться с переменным шагом, выбор которого связан с определёнными проблемами.

Метод композиций основан на функциональных особенностях вероятностных распределений, таких как распределение Эрланга, гипоэкспоненциальное и гиперэкспоненциальное распределения.

Метод используется, как правило, в тех случаях, когда не удаётся получить аналитическим методом решение в явном виде. Например, значения случайных величин, распределённых по закону Эрланга и гипоэкспоненциальному закону могут быть получены путём сложения нескольких экспоненциально распределённых случайных величин, а значения случайных величин, распределённых по гиперэкспоненциальному закону – путём вероятностного формирования смеси из нескольких экспоненциально распределённых случайных величин с разными математическими ожиданиями.

Для оценки качества случайных последовательностей с заданным законом распределения наиболее часто используют тест проверки частот и метод доверительного интервала для математического ожидания.

6.3. Введение в систему имитационного моделирования GPSS World

«Ошибаться человеку свойственно, но окончательно всё запутать может только компьютер»
(Пятый закон ненадежности)

GPSS (General Purpose Simulation System) – общецелевая система имитационного моделирования (СИМ), предназначенная для разработки моделей сложных систем с дискретным и непрерывным характером функционирования и проведения экспериментов с целью изучения свойств и закономерностей процессов, протекающих в них, а также выбора наилучшего проектного решения среди нескольких возможных вариантов.

Среди множества реализаций GPSS одной из наиболее доступных и популярных является GPSS World для работы на персональных компьютерах под управлением ОС Windows. GPSS World обладает удобным многооконным пользовательским интерфейсом, встроенными средствами визуализации и интерактивного управления процессом моделирования, обширной библиотекой встроенных процедур, включающей, в том числе, генераторы случайных величин для более чем двух десятков вероятностных распределений. Все это делает процесс моделирования эффективным и наглядным.

В GPSS World включены специальные средства для моделирования большого класса дискретных систем со стохастическим характером функционирования, в частности, систем и сетей массового обслуживания, что позволяет сделать модели ясными и лаконичными.

6.3.1. Состав системы имитационного моделирования GPSS World

Система имитационного моделирования GPSS World включает:

- язык GPSS – высокоуровневый язык имитационного моделирования;
- язык PLUS (Programming Language Under Simulation) – встроенный в GPSS язык программирования низкого уровня;
- компилятор – программа для трансляции (перевода) с языка высокого уровня на язык компьютера.

Объектами СИМ GPSS World являются:

- **«Модель»** или **«GPSS-модель»** – программа, написанная на *языке GPSS* и представляющая собой последовательность *операторов*, описывающих логику работы моделируемой системы, каждый из которых реализует некоторую конкретную функцию.
- **«Процесс моделирования»** – непосредственно исполняемый объект, создаваемый в результате трансляции объекта «GPSS-модель»; реализация «процесса моделирования» заключается в перемещении в модели некоторых подвижных объектов, называемых *транзактами*.

- **«Отчёт»** – создается автоматически по завершении процесса моделирования и содержит результаты моделирования.
- **«Текстовый объект»** – текстовые файлы, используемые для упрощения разработки больших моделей и формирования библиотеки исходных текстов.

Три первых объекта являются основными и всегда используются при имитационном моделировании.

6.3.2. Элементы языка GPSS World

Элементами языка GPSS World являются:

- ***алфавитно-цифровые символы***: латинские прописные и строчные буквы от «A» до «Z» и цифры от 0 до 9;
- ***имя*** – совокупность алфавитно-цифровых символов (от 1 до 200), *начинающаяся всегда с алфавитного символа*, причем допускается использование букв только *латинского алфавита*; для того чтобы имя не совпало с зарезервированными ключевыми словами (названиями операторов, системными числовыми атрибутами и т.п.), рекомендуется использование символа «_» (подчеркивание); примеры *правильных* имен: AS_27, R25, Pribor, W5Fix, Object_New1;
- ***метка*** – *имя*, расположенное в поле метки оператора для задания имени объекта GPSS-модели (памяти, таблицы, переменной,…) или для обозначения местоположения блока;
- ***переменная*** пользователя – *имя*, используемое в процессе моделирования для хранения числовых и строковых величин;
- ***числа*** – могут быть трёх типов:
  - *целочисленные* 32-разрядные (при переполнении преобразуются в вещественные);
  - *вещественные* 64-разрядные с плавающей точкой двойной точности (порядок может изменяться от –308 до +308, а точность ограничена примерно 15-ю десятичными разрядами),
  - *строковые* – массив символов произвольной длины, определяемой пользователем;
- ***системные числовые атрибуты (СЧА)*** – переменные, описывающие состояния процесса моделирования, автоматически поддерживаемые в GPSS и доступные в течение всего процесса моделирования;
- ***арифметические операторы*** – задают арифметические операции (перечислены в порядке приоритетности выполнения операций):
  - ^ (возведение в степень);
  - # (умножение), может быть изменено пользователем на *,
    / (деление),
    \ (целочисленное деление);
  - @ (остаток от деления);
  - + (сложение),
    - (вычитание);

- ***операторы отношения*** – задают логические условия (перечислены в порядке приоритетности выполнения операций):
  - > или ’G’ (больше),
    >= или ‘GE’ (больше или равно),
    < или ‘L’ (меньше),
    <= или ‘LE’ (меньше или равно);
  - = или ‘E’ (равно),
    != или ‘NE’ (не равно);
- ***логические операторы*** – задают логические операции (перечислены в порядке приоритетности выполнения операций):
  - & или ‘AND’ (логическое «И»);
  - | или ‘OR’ (логическое «ИЛИ»);
- ***выражения*** – часть языка PLUS: представляют собой совокупность *переменных*, *чисел* и *СЧА*, связанных *арифметическими операторами*, *логическими операторами* и *операторами отношения*; могут использоваться в операндах операторов GPSS и в PLUS-процедурах; всегда заключаются в круглые скобки;
- ***процедуры*** – программы на языке PLUS (PLUS-процедуры), встроенные в GPSS World (*стандартная процедура*) или созданные пользователем (*пользовательская процедура*); обращение к процедуре осуществляется путем задания в качестве операнда GPSS-операторов имени процедуры с её параметрами; библиотека стандартных процедур включает:
  - **обслуживающие процедуры** для управления прогонами процессов моделирования и анализа экспериментов;
  - **математические процедуры**: ABS (абсолютное значение), EXP (степень экспоненты), INT (целая часть), LOG (натуральный логарифм), SQR (квадратный корень), SIN (синус), COS (косинус), TAN (тангенс), ATN (арктангенс);
  - **процедуры запроса** для получения информации о состоянии находящегося в модели транзакта;
  - **строковые процедуры** для операций со строками;
  - **процедуры потоков данных** для управления потоками данных внутри PLUS-процедуры;
  - **процедуры динамического вызова** для вызова функций, хранящихся во внешних исполняемых файлах, включая динамически подключаемые библиотеки DLL;
  - **вероятностные распределения**.

6.3.3. Объекты GPSS-модели

> «Машинная программа выполняет то, что вы ей приказали делать, а не то, что бы вы хотели, чтобы она делала» ***(Третий закон Грида)***

GPSS-модель представляет собой написанную на языке GPSS программу и включает в себя множество объектов, которые могут быть

разбиты на 6 групп (рис.6.4):

- основные объекты;
- оборудование;
- числовые объекты;
- генераторы случайных чисел;
- групповые списки;
- потоки данных.

К *основным объектам* GPSS-модели относятся:

- **операторы (блоки и команды)** – основные объекты GPSS-модели, определяющие совокупность действий, которая должна быть выполнена в модели в соответствии с заданными в операторе параметрами, называемыми ***операндами***;
- **транзакты** – динамические объекты, движущиеся в GPSS-модели от одного оператора (блока) к другому в заданной последовательности.

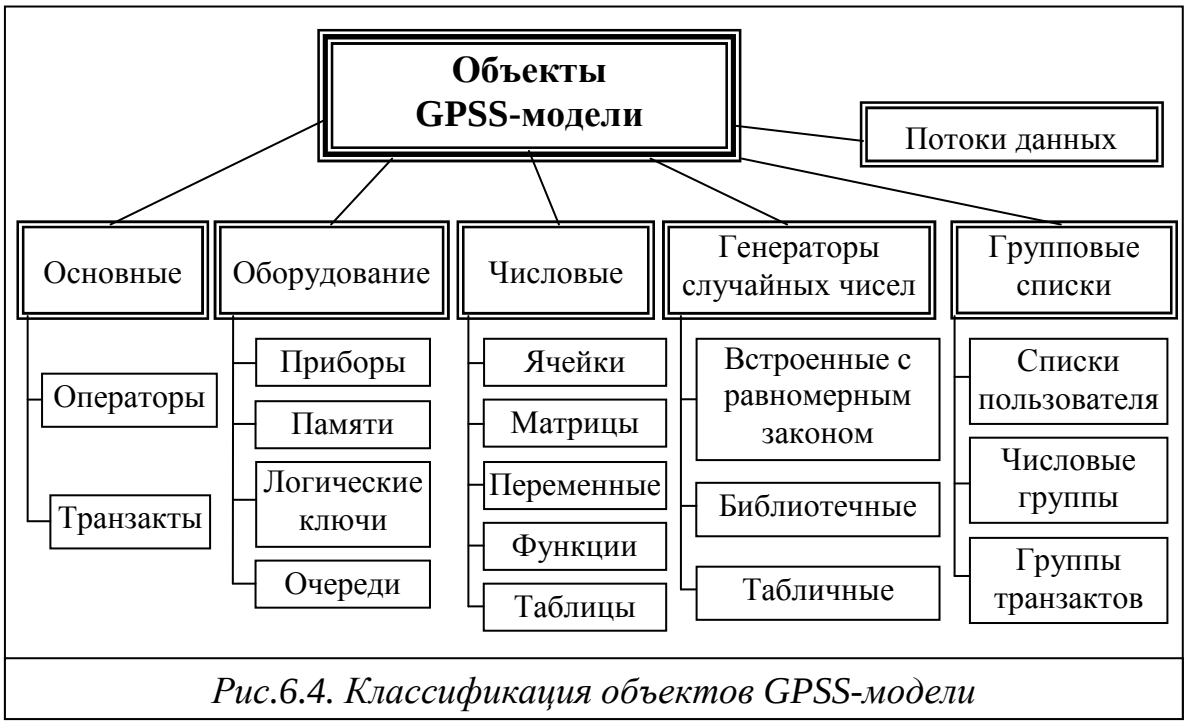


Рис.6.4. Классификация объектов GPSS-модели

Объектами *оборудования* являются:

- **приборы (одноканальные устройства)** – объекты, которые могут находиться в одном из двух состояний: свободном или занятом; при моделировании систем массового обслуживания используются для имитации процессов занятия и освобождения прибора, то есть для моделирования одноканальных СМО; занятие и освобождение прибора транзактом выполняется в GPSS-модели с помощью операторов SEIZE и RELEASE;
- **памяти (многоканальные устройства)** – объекты, состоящие из множества элементов, которые занимаются и освобождаются транзактами, при этом один транзакт может занять один или несколько элементов памяти, но не более чем её *ёмкость*; при моделировании систем массового обслуживания «память» используется для имитации процессов занятия и

освобождения приборов многоканальных СМО; ёмкость памяти задается в области описания GPSS-модели с помощью оператора STORAGE, а занятие и освобождение элементов «памяти» транзактом – с помощью операторов ENTER и LEAVE;

- **очереди** – объекты, используемые для накапливания транзактов, находящихся в состоянии ожидания какого-то события, например освобождения прибора или памяти; при моделировании систем массового обслуживания «очередь» используется для имитации процессов ожидания перед обслуживающими приборами; следует иметь в виду, что понятие «очередь» весьма относительное, поскольку в действительности транзакты, ожидающие освобождения прибора или памяти, заносятся в *«список задержки»* соответствующего прибора или памяти, при этом формирование списков задержки, то есть занесение в очередь и удаление из очереди, происходит автоматически, независимо от наличия операторов QUEUE и DEPART; последние используются только с целью сбора статистики по очередям путем фиксирования моментов поступления транзакта в очередь и удаления его из очереди;
- **логические ключи** – объекты, которые могут находиться только в двух состояниях: «установлен» или «сброшен»; установка, сброс или инвертирование ключа осуществляется с помощью оператора LOGIC.

К *числовым объектам* GPSS-модели относятся:

- **ячейки** – объекты для хранения величин, которым могут быть присвоены некоторые значения;
- **матрицы** – объекты для хранения массивов элементов размерности от 2 до 6;
- **переменные** – объекты для хранения величин, значения которых вычисляются на основе некоторого заданного выражения; переменные описываются с помощью операторов VARIABLE (арифметическая переменная), FVARIABLE (арифметическая переменная с плавающей точкой), BVARIABLE (булева переменная);
- **функции** – объекты, позволяющие вычислять значения в зависимости от некоторого аргумента; функции описываются с помощью оператора FUNCTION;
- **таблицы** – объекты, используемые для построения гистограммы плотности распределения случайной величины и представляющие собой набор чисел, отображающих частоту попадания значений случайной величины в тот или иной частотный диапазон (интервал); таблицы описываются с помощью оператора TABLE.

Генераторы случайных (точнее, псевдослучайных) чисел представляют собой объекты GPSS-модели, которые можно разделить на *три* группы:

- **встроенные** генераторы равномерно распределённых в интервале (0; 1) случайных чисел, основанные на мультипликативном конгруэнтном методе, с длиной периода 2 147 483 646; количество таких генераторов равно 999, причём номер генератора (от 1 до 999) определяет начальное

число для запуска генератора; при обращении к генератору с помощью системного числового атрибута (СЧА) RNj, где j – номер генератора, вырабатываются целочисленные случайные величины в интервале (0; 999);

- **библиотечные** генераторы случайных чисел с конкретными законами распределений, реализованные в виде встроенных библиотечных процедур количеством более 20;
- **табличные** генераторы случайных чисел с произвольными законами распределений, реализуемые пользователем в виде таблиц с помощью оператора описания FUNCTION.

Кроме перечисленных объектов при разработке больших сложных GPSS-моделей дополнительно могут использоваться:

- *групповые списки*, включающие в себя:
  - списки пользователя;
  - числовые группы;
  - группы транзактов,
- *потоки данных*.

Объекты в GPSS-модели могут формироваться автоматически, либо должны объявляться с использованием специальных команд – операторов описания. К объявляемым объектам относятся: памяти, переменные, матрицы, таблицы, функции, а также параметры транзактов.

6.3.4. Состав и структура GPSS-модели

GPSS-модель представляет собой программу, написанную на *языке GPSS* в виде последовательности *операторов*, описывающих логику работы моделируемой системы.

Операторы GPSS-модели делятся на две группы:

- GPSS-операторы;
- PLUS-операторы.

В свою очередь, GPSS-операторы делятся на *команды* и *блоки*.

Команды предназначены:

- для описания (определения) некоторых объектов, таких как памяти, переменные, функции, матрицы, таблицы; эти команды называются также *операторами описания*;
- для управления процессом моделирования (запуск, остановка и продолжение процесса моделирования, сброс статистики, завершение моделирования и т.п.); некоторые из этих команд могут находиться как в GPSS-модели, так и задаваться пользователем в процессе моделирования извне в качестве интерактивных операторов с использованием соответствующих пунктов меню GPSS World; эти команды называются также *операторами управления.*

Все команды делятся на:

- *срочные*, выполнение которых начинается в момент их появления;
- *несрочные*, которые заносятся в специальную очередь команд и последовательно выбираются для выполнения в процессе моделирования.

Операторы блоков или просто ***блоки*** представляют собой *исполняемые операторы* и реализуют в процессе моделирования некоторые действия, предписанные этими операторами. Название «блок» происходит от так называемых *блок-диаграмм*, которые используются для графического представления GPSS-моделей. **Блок-диаграмма** представляет собой набор стандартных блоков, связанных между собой в последовательности, определяемой логикой работы моделируемой системы. Каждому такому блоку в языке GPSS соответствует определенный оператор, реализующий некоторую конкретную функцию. При этом полагается, что реализация оператора связана с выполнением достаточно большой совокупности действий (блока). В GPSS World изображение этих блоков и перемещение транзактов в процессе моделирования можно увидеть в окне «BLOCK ENTITIES», доступ к которому выполняется из главного меню: **Window/Simulation Window/Blocks Window**.

Система имитационного моделирования GPSS, предназначенная для моделирования сложных систем со стохастическим характером функционирования, обладает эффективными встроенными средствами для описания параллельных процессов, протекающих в исследуемой системе. Эти средства позволяют выполнять описание каждого из параллельных процессов независимо друг от друга в виде отдельных программных модулей, что существенно упрощает процесс программирования и создания модели. В то же время, качество модели в значительной степени зависит от того, насколько достоверно и корректно отражены в модели связи между модулями, отображающие логику функционирования моделируемой системы.

Таким образом, укрупнено структуру GPSS-модели можно представить в виде множества модулей (рис.6.5), каждый из которых описывает один из протекающих в исследуемой системе параллельных процессов. Здесь же приведены некоторые операторы описания (команды) и исполняемые операторы (блоки), используемые в соответствующих модулях.

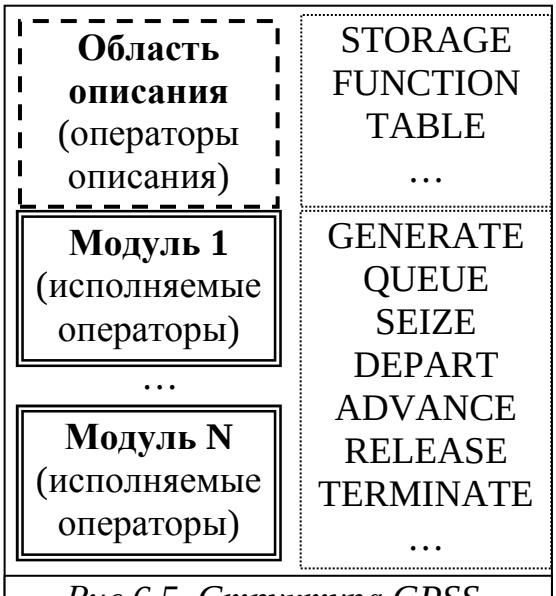


Рис.6.5. Структура GPSS-

Следует отметить, что выделение специальной области описания является желательным, но не обязательным. Такое выделение позволяет создать хорошо структурированную GPSS-модель, наглядную и легко понимаемую. В принципе, операторы описания могут быть в любом месте исполняемой области. При этом в GPSS World даже необязательно, чтобы оператор описания объекта находился до того, как соответствующий объект будет использован.

Оператор GPSS World, в общем случае, содержит 4 поля:

<Метка>	**<Операция>**	**<Операнды>**	**; <Комментарий>**

Поле **<Метка>** содержит *имя*, которое может быть присвоено оператору блока и оператору описания.

Каждый оператор занимает вполне определенное место в GPSS-модели. Для изменения естественного порядка выполнения процесса моделирования *любому исполняемому оператору* может быть присвоена *метка* в виде *имени*. В поле метки *оператора описания* указывается *имя* описываемого объекта (памяти, таблицы, функции и т.д.).

Поле **<Операция>** содержит зарезервированное слово GPSS World, определяющее функциональное назначение блока и задающее совокупность действий, которые должны быть выполнены.

Названия операторов (операции) в GPSS World обычно записываются прописными (но могут и строчными) буквами, не допускают сокращений и не могут использоваться в качестве переменных или имён объектов. Примеры операций: GENERATE (ГЕНЕРИРОВАТЬ), TERMINATE (ЗАВЕРШИТЬ), QUEUE (СТАТЬ В ОЧЕРЕДЬ), DEPART (ПОКИНУТЬ ОЧЕРЕДЬ), SEIZE (ЗАНЯТЬ), RELEASE (ОСВОБОДИТЬ), ENTER (ВОЙТИ), LEAVE (ВЫЙТИ), ADVANCE (ЗАДЕРЖАТЬ), PRIORITY (НАЗНАЧИТЬ ПРИОРИТЕТ), GATE (ВПУСТИТЬ), ASSIGN (НАЗНАЧИТЬ), TEST (ПРОВЕРИТЬ).

В поле **<Операнды>** задаются данные, необходимые для выполнения операции и представляющие собой параметры (операнды) оператора, разделяемые запятыми или пробелами, например: <**A,B,C,D**> или <**A B C D**>. При этом некоторые операнды являются *обязательными*, то есть должны быть всегда заданы, а другие – *необязательными*, то есть могут быть опущены при записи оператора. В последнем случае значения этих операндов определяются транслятором по умолчанию.

Между двумя соседними операндами может находиться только запятая или пробел: <**A,B**> или <**A B**>. Если между операндами **A** и **B** после запятой появится пробел: <**A, B**> или между ними будет находиться два пробела, то это равносильно двум запятым, и операнд **B** будет восприниматься транслятором как третий операнд, а значение второго операнда будет определяться по умолчанию.

Большинство операторов содержат один или два операнда.

В некоторых операторах в качестве операндов могут использоваться операторы отношения, задающие условия, выполнение которых проверяется в процессе выполнения операции.

Поле **«Комментарий»** располагается после операндов, от которых отделяется символом «точка с запятой».

GPSS-модель может содержать *комментарий*, который занимает всю строку. В этом случае признаком комментария служит символ «звездочка»

<*> или «точка с запятой» <;>, располагающийся в первой позиции строки, что говорит о наличии в этой строке только комментария. Если комментарий располагается после оператора в той же самой строке, то в качестве разделителя (признака комментария) используется только символ «точка с запятой». В поле **«Комментарий»** могут использоваться как латинские, так и русские буквы, а также любые другие символы.

Все операторы, кроме оператора описания FUNCTION, записываются в одну строку и могут содержать до 250 символов, включая комментарий.

6.4. Процесс моделирования в среде GPSS World

> «Если отладка - процесс удаления ошибок, то программирование должно быть процессом их внесения» ***(Э.Дейкстра)***

6.4.1. Запуск процесса моделирования

В результате трансляции (компиляции) GPSS-модели с использованием пунктов меню **Command / Create Simulation** создаётся исполняемый объект, реализующий **процесс моделирования**.

Для запуска процесса моделирования используется команда START, которая может находиться в GPSS-модели в качестве последнего оператора или может быть задана интерактивно после трансляции.

Если команда START находится в GPSS-модели, процесс моделирования запускается сразу же после трансляции автоматически. В противном случае, запуск процесса моделирования осуществляется путем задания команды START с использованием пунктов меню **Command / Start** системы имитационного моделирования GPSS World.

6.4.2. Транзакты

Реализация процесса моделирования заключается в перемещении в модели некоторых подвижных объектов, называемых *транзактами*. Транзакты последовательно перемещаются от блока к блоку в заданной алгоритмом моделирования последовательности.

Транзакты создаются и уничтожаются в модели с помощью операторов (блоков): GENERATE и TERMINATE.

В начале моделирования в GPSS-модели нет ни одного транзакта. В процессе моделирования транзакты формируются в модели в определенные моменты времени в соответствии с условиями, заданными с помощью блока GENERATE. Транзакты покидают модель (уничтожаются), попадая в блок TERMINATE. В общем случае, *в модели может находиться множество транзактов*, однако *в один и тот же момент времени продвигается только один транзакт*. Транзакт, попадая в определенный блок, вызывает к исполнению совокупность действий, предписанных соответствующим оператором, и затем пытается войти в следующий по

порядку блок. Такое продвижение транзакта продолжается до тех пор, пока не произойдет одно из следующих событий:

- транзакт входит в блок, функцией которого является задержка транзакта на некоторое заданное время (блок ADVANCE);
- транзакт пытается войти в блок, который "отказывается" принять его до тех пор, пока в модели не изменятся некоторые условия (например, блоки SEIZE, ENTER);
- транзакт входит в блок, функцией которого является удаление транзакта из модели (блок TERMINATE).

При возникновении одного их перечисленных событий транзакт прекращает движение и начинается перемещение в модели *другого* транзакта, то есть моделирование продолжается. Таким образом, моделирование заключается в перемещении транзактов между блоками GPSS-модели и выполнении соответствующих действий.

Для изменения последовательности движения транзактов используются условные и безусловные операторы, такие как TRANSFER, TEST, SELECT.

Транзакт, продвигаемый в модели в данный момент времени, называется ***активным***.

Интервал времени, в течение которого транзакт находится в модели, называется ***резидентным временем транзакта***.

Интервал времени, в течение которого транзакт проходит от одной произвольно выбранной точки модели до другой точки, называется ***транзитным временем*** перехода между двумя этими точками.

Каждому транзакту в модели присваивается порядковый номер, начиная с *единицы*.

6.4.3. Модельное время

Работа реальных систем протекает во времени, для отображения которого в GPSS-модели используется ***таймер модельного времени***. Изменение модельного времени происходит путем его продвижения *до ближайшего события*, связанного с изменением состояния моделируемой системы. При моделировании СМО и СеМО такими событиями являются:

- поступление заявок в систему;
- завершение обслуживания заявок в узле СеМО (обслуживающем приборе).

Таким образом, моделирование заключается в определении ближайшего момента наступления каждого события.

Таймер модельного времени корректируется *автоматически* в соответствии с логикой, предписанной моделью.

Таймер GPSS World может принимать любые значения. Единица времени (секунды, минуты, часы или их доли) для таймера задается разработчиком модели. Так как единица времени не сообщается транслятору, то все данные, связанные со временем, должны быть

выражены разработчиком через эту выбранную единицу.

Рассмотрим более подробно механизм изменения таймера модельного времени и логику процесса моделирования на примере моделирования системы массового обслуживания с неоднородным потоком заявок. Для формирования неоднородного потока заявок GPSS-модель будет содержать несколько операторов GENERATE по числу классов заявок.

В начале моделирования значение таймера модельного времени устанавливается в 0. Для всех классов заявок, поступающих в моделируемую систему, в каждом из блоков GENERATE определяется по одному ближайшему моменту появления транзакта, что соответствует моменту поступления очередной заявки данного класса. Очевидно, что число таких моментов будет равно количеству классов заявок. Среди всех этих моментов определяется момент с наименьшим значением, то есть момент, соответствующий ближайшему событию, и значение таймера модельного времени устанавливается равным значению (продвигается до) этого момента. Такое изменение значения таймера модельного времени приводит к тому, что соответствующий транзакт с моментом поступления, равным значению таймера, начинает движение в модели от блока GENERATE к следующему по порядку блоку. Движение транзакта в модели продолжается до тех пор, пока он не попадет в блок, функцией которого является задержка на некоторое заданное время, или в блок, который "отказывается" принять его до тех пор, пока в модели не изменятся некоторые условия. В последнем случае транзакт остаётся в предыдущем блоке. Если транзакт входит в блок, функцией которого является удаление транзакта из модели, то этот транзакт уничтожается.

Если в модели имеется ещё транзакт с таким же моментом формирования, то начинается его продвижение, причем значение таймера модельного времени не изменяется. Изменение таймера модельного времени происходит только в том случае, если в модели больше нет ни одного транзакта с таким же моментом формирования.

6.4.4. Списки

Описанный принцип продвижения транзактов в GPSS-модели реализуется с помощью так называемых *списков* или *цепей* (Chain).

В каждый момент модельного времени все транзакты, находящиеся в модели, соотносятся с одним из списков. В зависимости от принадлежности транзакта тому или иному списку, он может продвигаться в модели, быть готовым к дальнейшему продвижению, либо ожидать наступления заданного момента модельного времени или выполнения некоторого условия.

Такими списками в GPSS-модели являются:

- **список текущих событий (СТС)** – содержит транзакты, которые могут продвигаться в модели в текущий момент модельного времени;
- **список будущих событий (СБС)** – содержит транзакты, ожидаю-

щие наступления более позднего момента модельного времени;

- **списки повторных попыток (СПП)** – содержат транзакты, не удовлетворяющие условиям входа в блок, причем каждый объект GPSS-модели имеет свой СПП;
- **списки прибора** (одноканального устройства), включающие:
  - *список отложенных прерываний*, в котором находятся транзакты, ожидающие занятия устройства по приоритету с возможностью вытеснения транзакта, ранее занимавшего устройство;
  - *список прерываний*, в котором находятся транзакты, вытесненные из данного устройства, то есть обслуживание которых было прервано более высокоприоритетным транзактом;
  - *список задержки*, в котором находятся транзакты, ожидающие занятия устройства;
  - *список повторных попыток*;
- **списки памяти** (многоканального устройства), включающие:
  - *список задержки*,
  - *список повторных попыток*;
- **списки пользователя**, используемые для построения моделей с разнообразными функциональными возможностями, в частности, для организации моделирования различных алгоритмов формирования очередей (дисциплин буферизации и дисциплин обслуживания заявок в моделях массового обслуживания).

В любой модели всегда формируется *один список текущих* и *один список будущих событий*. Остальные списки формируются по мере необходимости.

В каждый момент модельного времени последовательно один за другим продвигаются только те транзакты, которые находятся в СТС, причем только один транзакт, который продвигается в данный момент реального времени, является активным. Продвижение каждого транзакта осуществляется до тех пор, пока это возможно. Например, если транзакт попадает в блок ADVANCE, функцией которого является задержка на некоторое время, то он переводится в СБС. Транзакт будет находиться в СБС до тех пор, пока модельное время не станет равным моменту, когда он может покинуть блок ADVANCE. В этом случае транзакт будет переведён в СТС.

Транзакты, расположенные в СТС с учётом их приоритетов, выбираются последовательно один за другим. Когда в СТС не остаётся транзактов, которые могут быть продвинуты в текущий момент модельного времени, происходит изменение модельного времени, которое продвигается до ближайшего запланированного момента времени для транзакта, находящегося первым в СБС. Этот, а также все другие транзакты, движение которых может быть возобновлено в тот же момент модельного времени, переносятся из СБС в СТС, где размещаются в порядке убывания приоритетов.

Каждый транзакт может иметь множество параметров, называемых

атрибутами транзакта, которые сопровождают его в течение «жизни» в модели. К ним, в частности, относятся:

- *параметры*, закрепляемые пользователем за каждым транзактом, число которых не ограничено; идентификатором параметра может служить его *номер* (целое положительное число) или *имя*; параметры транзакта должны быть определены, до того как они будут востребованы;
- *приоритет* – преимущественное право на использование общего ресурса, причём более высокому приоритету соответствует большее значение; транзакты с одинаковым приоритетом обычно выбираются в порядке поступления;
- *время входа транзакта в систему* – значение абсолютного времени в момент первого входа транзакта в модель или в блок MARK без операнда A;
- *текущий блок* – номер блока, в котором находится транзакт;
- *следующий блок* – номер следующего блока, в который должен перейти данный транзакт;
- *список*, в котором находится транзакт в некоторый момент времени:
  - ACTIVE – транзакт находится в СТС и является активным;
  - SUSPENDED – транзакт находится в СБС или в СТС и ожидает возможности стать активным;
  - PASSIVE – транзакт находится в состоянии ожидания: в списке пользователя, списке задержки или списке отложенных прерываний;
  - PREEMPTED – обслуживание транзакта в устройстве прервано, и он находится в списке прерываний;
  - TERMINATED – транзакт удаляется из модели и больше не участвует в процессе моделирования.

6.4.5. Завершение моделирования

Достоверность результатов моделирования в значительной степени определяется продолжительностью процесса моделирования, которое устанавливается разработчиком модели или пользователем.

В GPSS World завершение процесса моделирования может быть реализовано:

- принудительно с помощью срочной команды HALT, задаваемой из подменю COMMAND; использование этой команды позволяет принудительно остановить процесс моделирования в любой момент времени;
- по некоторому условию, задаваемому командой STOP, которая может находиться в GPSS-модели;
- по достижению содержимого «счётчика завершений» значения меньше или равного нулю.

Последний способ, используемый наиболее часто при моделировании систем и сетей массового обслуживания, рассмотрим более подробно.

Начальное значение «счётчика завершений» устанавливается с помощью команды START, которая запускает процесс моделирования. В процессе моделирования всякий раз при попадании транзакта в какой-либо блок TERMINATE (таких блоков в модели может быть несколько) из содержимого «счётчика завершений» вычитается значение, указанное в качестве параметра в соответствующем блоке TERMINATE. При достижении нулевого или отрицательного значения «счётчика завершений» процесс моделирования останавливается. Отметим, что при отсутствии параметра в блоке TERMINATE содержимое «счётчика завершений» не изменяется. Если во всех блоках TERMINATE отсутствуют параметры или их значения равны нулю, содержимое «счётчика завершений» не будет изменяться, и процесс моделирования (при отсутствии в модели команды STOP) будет длиться до тех пор, пока не будет введена команда HALT.

В одной и той же модели может быть предусмотрено несколько способов завершения моделирования. Например, в модели может находиться несколько команд STOP, задающих разные условия, и предусмотрено завершение моделирования по достижению содержимого «счётчика завершений» значения равного нулю. В этом случае завершение процесса моделирования происходит при достижении ближайшего по времени наступления условия.

По завершению процесса моделирования формируется и выводится на экран стандартный отчет, содержащий основные результаты моделирования, в том числе характеристики основных объектов – очередей, приборов, многоканальных устройств и т.д. Состав включаемых в отчёт результатов моделирования может быть изменён на вкладке **Reports** меню **EDIT/SETTINGS**.

Кроме отчета, содержащего числовые значения характеристик моделируемых систем, GPSS World предоставляет возможность получения результатов в графическом виде, в частности, путём формирования гистограмм плотностей распределений вероятностных характеристик. Для этого в GPSS-модели используются команды TABLE и QTABLE.

Более подробно результаты моделирования, представленные в отчете и в виде гистограмм, рассматриваются ниже при описании GPSS-моделей массового обслуживания.

6.4.6. Системные числовые атрибуты

Числовые и строковые переменные, используемые в процессе моделирования, называются ***атрибутами***. Атрибуты могут использоваться в операндах операторов GPSS и в выражениях.

Числовые атрибуты, *автоматически поддерживаемые в GPSS* и доступные в течение процесса моделирования, называются ***системными числовыми атрибутами (СЧА)*** (System Numerical Attributes – SNA). Их значения могут изменяться в процессе моделирования и доступны пользователю за счет использования специальных наименований этих атрибутов.

В GPSS используются СЧА трёх типов:

- СЧА *объектов*, описывающие состояние таких объектов GPSS-модели как приборы (одноканальные устройства), памяти (многоканальные устройства), очереди, таблицы и др.;
- СЧА *системы*, описывающие состояние модели в целом;
- СЧА *транзактов*, описывающие их свойства и параметры.

Имя СЧА объектов состоит из двух частей:

- первая часть указывает *групповое имя*, идентифицирующее *тип объекта* (прибор, многоканальное устройство, очередь, таблица) и *тип информации* (количество входов в объект, загрузка объекта, среднее время занятия объекта и т.д.);
- вторая часть (число или имя) идентифицирует *конкретного члена группы*.

Если конкретный член группы задан в виде имени, то вторая часть имени СЧА отделяется от первой (группового имени) символом $. Таким образом, имя СЧА может иметь вид:

<Групповое имя><*Число*> или **<Групповое имя>$**<*Имя*>.

Например: **F***5*, **QT***23*, **FR$***Pribor*, **SR$***New_System*.

Групповые имена (наименования) и значения основных ***СЧА объектов***, таких как приборы, многоканальные устройства, очереди и таблицы приведены ниже в табл.1-3.

Таблица 1

СЧА приборов

Групповое имя	Значение
F	1, если прибор занят; 0, если свободен
FC	Число занятий прибора транзактами
FR	Загрузка прибора, выраженная в долях тысячи
FT	Среднее время занятия прибора транзактом

Таблица 2

СЧА многоканальных устройств

Групповое имя	Значение
R	Количество незанятых приборов (каналов)
S	Количество занятых приборов (каналов)
SA	Среднее количество занятых приборов
SC	Счетчик числа входов в многоканальное устройство (при каждом выполнении блока ENTER значение счетчика увеличивается на величину операнда B блока)
SM	Максимальное количество занятых приборов (максимальное значение Sj или S$ имя)
SR	Загрузка многоканального устройства, выраженная в долях тысячи
ST	Среднее время нахождения транзакта в устройстве.

Таблица 3

СЧА очередей

Групповое имя	Значение
Q	Текущее значение длины очереди
QA	Среднее значение длины очереди
QC	Количество входов в очередь (увеличивается на величину операнда B блока QUEUE)
QM	Максимальное значение длины очереди
QT	Среднее время пребывания в очереди с учетом нулевых входов
QX	Среднее время пребывания в очереди для входов без учета нулевых входов
QZ	Количество нулевых входов в очередь, при которых время ожидания было равно нулю

Кроме того, могут использоваться следующие ***СЧА объектов***:

- СЧА таблиц:
  - **TB$**<*Имя*> – Среднее значение элементов таблицы
  - **TC $**<*Имя*> – Количество учтенных в таблице элементов
  - **TD $**<*Имя*> – Стандартное отклонение элементов таблицы
- СЧА функции:
  - **FN$**<*Имя*> – результат вычисления функции;
- СЧА переменной:
  - **V$**<*Имя*> – результат вычисления переменной.

Примеры СЧА объектов:

FR3 – возвращает значение загрузки прибора с номером **3**;

FT$Auto_Master – возвращает среднее время занятия транзактом прибора с именем **Auto_Master**.

S22 – возвращает количество занятых приборов в многоканальном устройстве с номером **22**;

SM$Kassa_2m – возвращает максимальное количество занятых приборов в многоканальном устройстве с именем **Kassa_2m**.

V$F_5 – возвращает значение переменной **F_5**.

К ***СЧА системы*** относятся такие глобальные переменные как:

- **AC1** – значение абсолютного модельного времени (с момента начала моделирования или последней команды CLEAR);
- **C1** – значение относительного модельного времени (с момента последней команды RESET);
- **TG1** – текущее значение счетчика завершения;
- **Z1** – свободная оперативная память ЭВМ в байтах.

К ***СЧА транзактов*** относятся:

- **MP**<*Число*> или **MP$**<*Имя*> – транзитное время транзакта

(абсолютное модельное время минус значение, содержащееся в параметре <*Число*> или <*Имя*>);

- **P**<*Число*> или **P$**<*Имя*> – значение параметра <*Число*> или <*Имя*>;
- **PR** – приоритет транзакта;
- **M1** – резидентное время транзакта (абсолютное модельное время минус время появления транзакта в модели);
- **XN1**– номер активного транзакта.

6.4.7. Встроенные вероятностные распределения

Встроенная библиотека процедур GPSS World содержит более 20 вероятностных распределений, в том числе:

- равномерное (Uniform);
- экспоненциальное (Exponential);
- геометрическое (Geometric);
- Пуассона (Poisson);
- Бета (Beta);
- Гамма (Gamma);
- биномиальное (Binomial);
- дискретно-равномерное (Discrete Uniform);
- треугольное (Triangular);
- нормальное (Normal);
- Парето (Pareto); …

Для обращения к вероятностному распределению необходимо указать имя библиотечной процедуры и её параметры, заключённые в круглые скобки и отделённые друг от друга запятой:

<Имя процедуры>(G,A,B, …)

Здесь G – номер генератора равномерно распределённых случайных чисел (от 1 до 999) – используется в качестве аргумента для формирования случайных величин с заданным законом распределения. Остальные параметры **A, B, …,** количество которых для разных распределений составляет от 1 до 4, задают непосредственно параметры вероятностного распределения.

Ниже рассматриваются только некоторые из перечисленных распределений, наиболее часто используемые в моделях массового обслуживания.

1. *Равномерное распределение*:

UNIFORM (G,Min,Max),

где Min и Max – соответственно минимальное и максимальное значение равномерно распределённой случайной величины.

2. *Экспоненциальное распределение*:

EXPONENTIAL (G,Min,Mean),

где Mean – математическое ожидание (среднее значение) случайной

величины, распределённой по экспоненциальному закону;
Min – смещение распределения относительно нуля (минимальное значение случайной величины).

3. *Распределение Пуассона*:

POISSON (G,Mean),

где Mean – математическое ожидание (среднее значение) случайной величины.

4. *Геометрическое распределение*:

GEOMETRIC (G,P),

где **P** – параметр распределения, принимающий значения в интервале (0;1).

Библиотечные процедуры вероятностных распределений могут использоваться в выражениях, в том числе арифметических, а также в качестве операнда A в операторах GENERATE и ADVANCE. В последнем случае они рассматриваются как выражения языка PLUS и должны быть заключены в круглые скобки.

6.5. Операторы блоков GPSS World

6.5.1. Общие сведения

В GPSS World используются 53 оператора блоков. Среди операторов блоков имеются так называемые взаимодополняющие операторы, представляющие собой пару операторов, каждый из которых является зеркальным отображением другого оператора, означающим, что совокупность действий, реализуемых одним оператором, является противоположной по отношению к совокупности действий, реализуемых другим оператором. Примерами взаимодополняющих операторов могут служить операторы GENERATE и TERMINATE, SEIZE и RELEASE, QUEUE и DEPART, ENTER и LEAVE, PREEMPT и RETURN.

Для построения имитационных моделей *простейших* систем и сетей массового обслуживания в среде GPSS World оказывается достаточным использование примерно половины из всех операторов блоков, которые по функциональному назначению могут быть разбиты на следующие группы:

1. Операторы генерирования, задержки и удаления транзактов: GENERATE, ADVANCE, TERMINATE.
2. Операторы одноканальных устройств (приборов): SEIZE, RELEASE.
3. Операторы многоканальных устройств (памятей): ENTER, LEAVE.
5. Операторы очередей: QUEUE, DEPART.
4. Условные операторы: TEST, TRANSFER, GATE.
5. Операторы приоритетного обслуживания: PRIORITY, PREEMPT, RETURN.
6. Оператор логических ключей: LOGIC.
7. Прочие операторы: ASSIGN, MARK, TABULATE.

Операторы могут быть без операндов или содержать от 1 до 7 операндов, некоторые из которых могут быть необязательными, то есть могут отсутствовать. В последнем случае значения необязательных операндов принимаются по умолчанию. Если после отсутствующего операнда в операторе имеются другие операнды, то признаком отсутствия необязательного операнда служит лишняя запятая. Например, следующая запись в поле операций: **<„C>** означает, что операнды **A** и **B** не используются, и их значения принимаются по умолчанию.

Ниже представлены краткие описания операторов, наиболее часто используемых при построении имитационных моделей массового обслуживания. При изображении структуры оператора *необязательные операнды заключены в квадратные скобки.*

Отсутствие обязательных операндов приводит к ошибке.

6.5.2. GENERATE (ГЕНЕРИРОВАТЬ)

Назначение оператора: генерирование транзактов в соответствии с заданным правилом формирования интервалов между транзактами.

Формат оператора:

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

Значения операндов:

A – средний интервал времени между генерируемыми транзактами или вероятностное распределение интервала из встроенной библиотеки процедур, заключённое в круглые скобки;; [по умолчанию – ноль];

B – величина полуинтервала равномерно распределенного интервала или модификатор таблично заданной функции; [по умолчанию – ноль];

C – смещение – момент формирования первого транзакта; [по умолчанию – ноль];

D – ограничитель – число генерируемых данным оператором транзактов; [по умолчанию – не ограничено];

E – уровень приоритета от 0 до 127 (чем больше номер, тем выше приоритет); [по умолчанию – ноль].

Примечание. Несмотря на то, что операнды A и D – необязательные операнды, в операторе **GENERATE** обязательно должен использоваться один из них: либо операнд **A** , либо операнд **D**.

Примеры:

GENERATE 25; интервал времени между генерируемыми транзактами – величина детерминированная равная 25, количество генериру-емых транзактов не ограничено.

GENERATE „„25;операнды **A**, **B** и **C** не используются, и их значения равны нулю по умолчанию; это означает, что в нулевой момент модельного времени будут сгенерированы ровно 25 транзактов.

GENERATE 25,10; интервал времени между транзактами – величина случайная, равномерно распределенная в интервале (25±10), т.е. от 15 до 35.

GENERATE 25,FN$Erlang; интервал времени между транзактами – величина случайная, распределенная по закону, заданному в виде табличной функции Erlang.

GENERATE 25,10,100,250,5; интервал времени между транзактами – равномерно распределенная величина в интервале от 15 до 35; момент формирования первого транзакта равен 100 единицам модельного времени; всего за время моделирования этим оператором будет сгенерировано 250 транзактов, после чего формирование транзактов прекратится; всем сгенерированным транзактам будет присвоен приоритет, равный 5.

GENERATE (Exponential(1,0,50)); интервал времени между транзактами – величина случайная, распределенная по экспоненциальному закону со средним значением 50.

Следует обратить внимание, что в последнем примере имя библиотечной процедуры с параметрами **Exponential(1,0,50)** заключается в круглые скобки. Параметры процедуры **Exponential** имеют следующий смысл: первый параметр – номер встроенного генератора равномерно распределённых в интервале (0; 1) случайных чисел (может иметь значения от 1 до 999); второй и третий параметры – соответственно смещение (минимальное значение) и среднее значение (математическое ожидание) случайной величины, распределённой по экспоненциальному закону.

6.5.3. TERMINATE (ЗАВЕРШИТЬ)

Назначение оператора: удаление транзактов из модели.

Формат оператора:

TERMINATE [A]

Значения операндов:

A – указатель уменьшения счетчика завершений (целое положительное число); [по умолчанию – ноль].

Примеры:

TERMINATE 1; транзакт, поступивший в данный блок, удаляется из модели, и счетчик завершения процесса моделирования, начальное значение которого устанавливается командой START, уменьшается на 1.

TERMINATE; транзакт удаляется из модели, при этом значение счетчика завершения процесса моделирования не изменяется.

6.5.4. ADVANCE (ЗАДЕРЖАТЬ)

Назначение оператора: задержка транзакта на заданное время.

Формат оператора:

ADVANCE [A],[B]

Значения операндов:

A – среднее время задержки или вероятностное распределение из встроенной библиотеки процедур, заключённое в круглые скобки; [по умолчанию – ноль];

B – величина полуинтервала равномерно распределенного интервала задержки или модификатор таблично заданной функции; [по умолчанию – ноль].

Примеры:

ADVANCE 50; поступивший транзакт задерживается в данном блоке на 50 единиц времени.

ADVANCE 50,10; время задержки транзакта – величина случайная, равномерно распределенная в интервале от 40 до 60 (50 ± 10).

ADVANCE 50,FN$Erl_1; время задержки транзакта – величина случайная, распределенная по закону, заданному в виде табличной функции **Erl_1,** со средним значением 50**.**

ADVANCE (Exponential(33,10,50)); время задержки – величина случайная, распределенная по экспоненциальному закону (из встроенной библиотеки процедур) со средним значением 50; номер встроенного генератора равномерно распределённых случайных чисел равен 33; смещение равно 10, то есть случайная величина, распределённая по экспоненциальному закону, принимает значения, начиная от 10.

6.5.5. SEIZE (ЗАНЯТЬ)

Назначение оператора: занятие транзактом прибора.

Формат оператора:

SEIZE A

Значения операндов:

A – идентификатор (число или имя) занимаемого прибора.

Примеры:

SEIZE 4; транзакт пытается занять прибор с номером 4; если прибор занят другим транзактом, то поступивший транзакт помещается в список задержки этого прибора, где находится до момента освобождения прибора, после чего этот транзакт занимает освободившийся прибор и продолжает свое движение к следующему блоку.

SEIZE Pribor_Disk; транзакт пытается занять прибор с именем Pribor_Disk; далее по аналогии с предыдущим примером.

6.5.6. RELEASE (ОСВОБОДИТЬ)

Назначение оператора: удаление транзакта из прибора (освобождение прибора).

Формат оператора:

RELEASE A

Значения операндов:

A – идентификатор (число или имя) освобождаемого прибора.

Примеры:

RELEASE 4; транзакт освобождает прибор с номером 4.

RELEASE Pribor_Disk; транзакт освобождает прибор с именем Pribor_Disk.

6.5.7. QUEUE (СТАТЬ В ОЧЕРЕДЬ)

Назначение оператора: занесение транзакта в очередь (точнее – регистрация статистики очереди, связанная с фиксацией момента поступления транзакта в очередь и увеличением ее длины).

Формат оператора:

QUEUE A,[B]

Значения операндов:

A – идентификатор (число или имя) очереди;

B – количество элементов, на которое должна увеличиться длина очереди; [по умолчанию – один].

Примеры:

QUEUE 3; присоединение транзакта к очереди с номером 3 и увеличение ее длины на 1 (по умолчанию).

QUEUE Jeck,5;присоединение транзакта к очереди с именем Jeck и увеличение ее длины на 5.

6.5.8. DEPART (ПОКИНУТЬ ОЧЕРЕДЬ)

Назначение оператора: удаление транзакта из очереди (точнее – регистрация статистики очереди, связанная с уменьшением ее длины и фиксацией момента удаления транзакта из очереди с целью определения времени ожидания).

Формат оператора:

DEPART A,[B]

Значения операндов:

A – идентификатор (число или имя) очереди;

B – количество элементов, на которое должна уменьшиться длина очереди; [по умолчанию – один].

Примеры:

DEPART 3; удаление транзакта из очереди с номером 3 и уменьшение ее длины на 1 (по умолчанию).

DEPART Jeck,5;удаление транзакта из очереди с именем Jeck и уменьшение ее длины на 5.

6.5.9. ENTER (ВОЙТИ)

Назначение оператора: вход транзакта в многоканальное устройство.

Формат оператора:

ENTER A,[B]

Значения операндов:

A – идентификатор (число или имя) многоканального устройства;

B – количество занимаемых приборов многоканального устройства; [по умолчанию – один].

Примеры:

ENTER 5; транзакт поступает в многоканальное устройство с

номером 3 и занимает один прибор (по умолчанию).

ENTER MANY,4; транзакт, поступая в многоканальное устройство с именем MANY, занимает 4 прибора.

6.5.10. LEAVE (ВЫЙТИ)

Назначение оператора: удаление транзакта из многоканального устройства.

Формат оператора:

LEAVE A,[B]

Значения операндов:

A – идентификатор (число или имя) многоканального устройства;

B – количество освобождаемых приборов многоканального устройства; [по умолчанию – один].

Примеры:

LEAVE 5; транзакт покидает многоканальное устройство с номером 3 и освобождает 1 прибор (по умолчанию).

LEAVE MANY,4; транзакт, покидая многоканальное устройство с именем MANY, освобождает 4 прибора.

6.5.11. TEST (ПРОВЕРИТЬ)

Назначение оператора: поверка значения (обычно СЧА) и передача активного транзакта в блок, отличный от последующего, если указанное условие не выполняется.

Формат оператора:

TEST X A,B,[C],

Значения операндов:

A – проверяемое значение;

B – контрольное значение;

C – имя (метка) блока назначения **C**; [по умолчанию – Режим отказа];

X – оператор отношения, определяющий условие проверки операнда **A** с операндом **B**:

Значения X	Интерпретация в смысле блока TEST
G	**A** больше **B**?
GE	**A** больше или равно **B**?
E	**A** равно **B**?
NE	**A** не равно **B**?
LE	**A** меньше или равно **B**?
L	**A** меньше **B**?

Блок **TEST** может функционировать в двух режимах:

- в режиме альтернативного выхода (если задан операнд **C**);
- в режиме отказа (если операнд **C** не задан).

Когда транзакт пытается войти в блок **TEST** в режиме альтернативного выхода и проверяемое условие не выполняется, транзакт

передается блоку, указанному в операнде **C**.

Когда транзакт пытается войти в блок **TEST** в режиме отказа (при отсутствии операнда **C**) и заданное условие не выполняется, транзакт блокируется до тех пор, пока условие не будет выполнено.

Примеры:

TEST LE Q1,5,Otk_1; если проверяемое условие «длина очереди 1 меньше или равна 5?» выполняется, то активный транзакт передается следующему оператору, в противном случае он направляется к оператору с меткой Otk_1.

TEST G Q1,5; если проверяемое условие «длина очереди 1 больше 5?» выполняется, то активный транзакт передается следующему оператору, в противном случае он блокируется до тех пор, пока условие не будет выполнено.

6.5.12. TRANSFER (ПЕРЕДАТЬ)

Назначение оператора: передача транзакта в блок, отличный от последующего.

Режимы использования оператора TRANSFER:

1) режим безусловной передачи;
2) режим статистической передачи;
3) режим BOTH (ОБА);
4) режим ALL (ВСЕ);
5) режим PICK (выборочный);
6) режим FN (функциональный);
7) режим P (параметрический);
8) режим SBR (подпрограммный);
9) режим SIM (одновременный).

Далее рассматриваются только два первых режима, используемые ниже при построении GPSS-моделей.

1. Режим безусловной передачи

Назначение оператора: безусловная передача транзакта в блок, отличный от последующего.

Формат оператора:

TRANSFER ,B

Значения операндов:

A – НЕ используется, что является признаком режима безусловной передачи;

B – имя блока, к которому направляется активный транзакт.

Пример:

TRANSFER ,UZEL_3; всякий раз активный транзакт будет направляться к блоку с меткой **UZEL_3.**

2. Режим статистической передачи

Назначение оператора: передача транзакта в один из блоков случайным образом.

Формат оператора:

TRANSFER A,[B],C

Значения операндов:

A – частота (вероятность) передачи транзакта в блок **C**;

B – имя блока B; [по умолчанию – Следующий по порядку блок];

C – имя блока C.

Примечание. Частота (вероятность) передачи транзакта в операнде **A** может быть указана двумя способами:

- в виде вероятности – дробного числа с десятичной точкой, принимающего значения строго меньше 1;
- в виде целого положительного числа, принимающего значения от 0 до 1000 и интерпретируемого как доля от тысячи.

Заметим, что значение операнда **A**, равное 1, будет соответствовать вероятности 0,001, а не 1, поскольку транслятор воспринимает любое целое число как долю от 1000.

Следует также отметить, что транслятор не выдаст ошибку, если операнд A будет задан в виде числа с десятичной точкой, имеющего значение больше 1. В этом случае транслятор выделяет целую часть числа и интерпретирует её как долю от тысячи.

Примеры:

TRANSFER 250,UZEL_2,UZEL_3; активный транзакт с вероятностью 250/1000 = 0,25 будет направляться к блоку с меткой **UZEL_3** и с вероятностью 0,75 – к блоку с меткой **UZEL_2** .

TRANSFER 25,UZEL_2,UZEL_3; активный транзакт с вероятностью 25/1000 = 0,025 будет направляться к блоку с меткой **UZEL_3** и с вероятностью 0,975 – к блоку с меткой **UZEL_2** .

TRANSFER .95,,BOX7; активный транзакт с вероятностью 0,95 будет направляться к блоку с меткой **BOX7** и с вероятностью 0,05 – к следующему по порядку блоку (по умолчанию).

6.5.13. PRIORITY (НАЗНАЧИТЬ ПРИОРИТЕТ)

Назначение оператора: изменение уровня приоритета активного транзакта в процессе моделирования.

Формат оператора:

PRIORITY A,[B]

Значения операндов:

A – уровень приоритета, присваиваемый активному транзакту;

B – может принимать только одно значение: **BU** (задает специальный режим, при котором активный транзакт помещается в список текущих событий позади транзактов с таким же приоритетом); [по умолчанию – Транзакт помещается перед транзактами с таким же приоритетом].

6.5.14. PREEMPT (ЗАХВАТИТЬ)

Назначение оператора: захват прибора вновь прибывшим

транзактом.

Формат оператора:

PREEMPT A,[B],[C],[D],[E]

Значения операндов:

A – идентификатор (число или имя) прибора, подлежащего захвату;

B – определяет условие, при котором разрешён захват прибора: **PR** – *приоритетный режим*: захват разрешён, если активный транзакт имеет более высокий приоритет, чем обслуживаемый транзакт; [по умолчанию – *Режим прерывания*: захват разрешён, если обслуживаемый транзакт не является захватчиком];

C – метка блока, в который направляется транзакт, вытесненный из прибора более высокоприоритетным транзактом; [по умолчанию – Транзакт помещается в СБС];

D – номер параметра вытесненного транзакта, в который записывается оставшееся время обслуживания в приборе, если транзакт направляется к блоку **C**; используется совместно с операндом **C**;

E – может принимать только одно значение: **RE**, означающий *режим удаления*: вытесненный транзакт удаляется из состязания за прибор; [по умолчанию – вытесненный транзакт не удаляется из состязания за прибор].

Примечание. Следует обратить внимание, что приоритетный захват возможен только для *прибора*, но невозможен для многоканального устройства.

6.5.15. RETURN (ВЕРНУТЬ)

Назначение оператора: освобождение прибора активным транзактом и выбор нового транзакта.

Формат оператора:

RETURN A

Значения операнда:

A – идентификатор (число или имя) прибора, подлежащего освобождению.

Примечание. В прибор новый транзакт выбирается из списков прибора в строго определённой последовательности: сначала выбирается транзакт из списка отложенных прерываний; если он пуст, то транзакт выбирается из списка прерываний и, наконец, если и список прерываний пуст, то транзакт выбирается из списка задержки.

6.5.16. LOGIC (ИЗМЕНИТЬ)

Назначение оператора: изменение состояния логического ключа.

Формат оператора:

LOGIC X A

Значения операндов:

A – идентификатор (число или имя) логического ключа;

X – логический оператор, указывающий тип операции изменения

состояния: **R** – сбросить (выключить), **S** – установить (включить), **I** – инвертировать.

6.5.17. GATE (ВПУСТИТЬ)

Назначение: изменение маршрута движения транзактов в зависимости от состояния некоторого объекта.

Формат оператора:

GATE X A,[B]

Значения операндов:

A – идентификатор (число или имя) проверяемого объекта;

B – номер блока, к которому переходит транзакт, если объект находится в положении, не отвечающем условию проверки; [по умолчанию – Проверка происходит в режиме отказа];

X – условный оператор, содержащий условие, которому должен удовлетворять объект для успешного завершения теста; может принимать множество значений, в соответствии с которыми проводится проверка состояния некоторого объекта (прибора, многоканального устройства, логического ключа), в том числе:

- **FV** – прибор доступен;
- **FNV** – прибор недоступен;
- **I** – прибор в прерванном состоянии;
- **NI** – прибор в непрерванном состоянии;
- **U** – прибор используется;
- **NU** – прибор не используется;
- **SE** – многоканальное устройство пусто;
- **SNE** – многоканальное устройство не пусто;
- **SF** – многоканальное устройство заполнено;
- **SNF** – многоканальное устройство не заполнено;
- **SV** – многоканальное устройство доступно;
- **SNV** – многоканальное устройство не доступно;
- **LS** – логический ключ установлен (включен);
- **LR** – логический ключ сброшен (выключен).

6.5.18. MARK (ОТМЕТИТЬ)

Назначение оператора: запись значения абсолютного времени в качестве одного из параметров активного транзакта (отметка транзакта).

Формат оператора:

MARK [A]

Значения операндов:

A – номер параметра, в который записывается значение таймера абсолютного времени; [по умолчанию – Значение абсолютного времени помещается на место ранее записанного времени входа транзакта в модель].

6.5.19. ASSIGN (НАЗНАЧИТЬ)

Назначение оператора: назначение и изменение параметра транзакта.

Формат оператора:

ASSIGN A,B,[C]

Значения операндов:

A – номер модифицируемого параметра и вид модификации: присвоение, увеличение (+), уменьшение (-);

B – величина, используемая для модификации;

C – номер функции для модификации.

Примеры:

ASSIGN 4,10.5; параметру с номером 4 присваивается значение 10.5.

ASSIGN 4+,10.5; значение параметра с номером 4 увеличивается на величину 10.5.

ASSIGN 4-,10.5; значение параметра с номером 4 уменьшается на величину 10.5.

ASSIGN 3+,5,7; 1) рассчитывается значение функции 7; 2) это значение умножается на 5; 3) целая часть этого произведения прибавляется к значению параметра 3 вошедшего (активного) транзакта.

ASSIGN 3+,5,FN7; в отличие от предыдущего примера сначала рассчитывается значение функции 7, целая часть которого будет использоваться как **C**, то есть номер функции.

ASSIGN 3+,5,Erl; 1) рассчитывается значение функции с именем Erl; 2) это значение умножается на 5; 3) целая часть этого произведения прибавляется к значению параметра 3 вошедшего (активного) транзакта.

Три последних примера показывают, что в качестве операнда C может использоваться номер функции или её имя (без FN или FN$). Если же операнд C содержит FN или FN$, то это означает косвенное определение номера функции для модификации.

6.5.20. TABULATE (ТАБУЛИРОВАТЬ)

Назначение оператора: занесение значений в таблицу.

Формат оператора:

TABULATE A,[B]

Значения операндов:

A – имя таблицы, в которую заносится соответствующее значение и которая должна быть описана с помощью оператора описания (команды) **TABLE**;

B – весовой коэффициент; [по умолчанию – Коэффициент равен 1].

При попадании активного транзакта в оператор **TABULATE** обновляется статистика, связанная с таблицей, указанной в операнде A (см. пример в п.6.6.4).

6.6. Команды GPSS World

6.6.1. Общие сведения

В GPSS World используются 24 команды (операторов описания и операторов управления).

Для построения и реализации имитационных моделей *простейших* систем и сетей массового обслуживания в среде GPSS World оказывается достаточным использование немногим более половины из всех команд, которые по функциональному назначению могут быть разбиты на две группы:

1. Операторы (команды) описания: FUNCTION, TABLE, QTABLE, STORAGE, VARIABLE.

2. Операторы (команды) управления: CLEAR, CONTINUE, HALT, INCLUDE, REPORT, RESET, SHOW, START, STEP, STOP.

Команды управления используются в процессе моделирования для интерактивного взаимодействия пользователя с GPSS-моделью и управления процессом моделирования.

Команды, как и операторы блоков, могут быть без операндов или содержать от 1 до 5-и операндов, некоторые из которых могут быть необязательными. В последнем случае значения необязательных операндов принимаются по умолчанию. При изображении структуры оператора *необязательные операнды заключены в квадратные скобки.*

Отсутствие обязательных операндов приводит к ошибке.

6.6.2. FUNCTION (ФУНКЦИЯ)

Назначение: описание функции.

Формат:

<Имя> FUNCTION A,B

Здесь: **<Имя>** – имя функции.

Значения операндов:

A – аргумент функции;

B – задаёт тип функции и количество пар данных в виде:

<тип функции><количество пар данных>,

где **<тип функции>** может принимать следующие значения: **C** – непрерывная функция, **D** – дискретная функция, **E** – дискретная атрибутивно-значимая функция, **L** – списковая числовая функция, **M** – списковая атрибутивно-значимая функция;

<количество пар данных> определяет количество пар данных (аргумента и соответствующего ему значения функции) в списке данных функции, который располагается после оператора с первой позиции новой строки и может занимать несколько строк; каждая пара данных определяет значение аргумента X и значение функции Y (или СЧА), разделённые запятой.

Пример. При моделировании систем массового обслуживания функция типа **C** может использоваться для табличного представления

вероятностных законов распределения случайных величин. В частности, в предыдущих версиях GPSS для генерирования случайных чисел, распределённых по экспоненциальному закону, использовался табличный генератор, заданный в виде следующей функции:

EXP1FUNCTION RN100, C24
0,0/.1, .104/.2, .222/.3, .335/.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

Здесь:

EXP1 – имя табличной функции, которое используется в СЧА класса FN при обращении к функции: FN$EXP1;

RN100 – генератор равномерно распределённых случайных чисел с номером 100, используемый в качестве аргумента функции для вычисления значений экспоненциально распределённых случайных величин; путём изменения номера генератора равномерно распределённых случайных чисел (от 1 до 999) можно создавать множество генераторов экспоненциально распределённых случайных величин;

C24 – тип функции – **C**, означающий, что значения функции для любого значения аргумента определяются с использованием линейной интерполяции; таблица содержит 24 пары значений аргумента и функции, причём каждая пара отделена от другой наклонной чертой.

6.6.3. STORAGE (МНОГОКАНАЛЬНОЕ УСТРОЙСТВО)

Назначение: описание ёмкости многоканального устройства (памяти).

Формат:

<Имя> STORAGE A

Здесь: **<Имя>** – имя многоканального устройства.

Значения операнда:

A – количество приборов (каналов) в многоканальном устройстве.

6.6.4. TABLE (ТАБЛИЦА)

Назначение: описание таблицы, используемой в модели для накопления частоты попадания некоторой случайной величины в заданные частотные интервалы и построения гистограммы плотности распределения.

Формат:

<Имя> TABLE A,B,C,D

Здесь: **<Имя>** – имя таблицы (не более 32-х алфавитно-цифровых символов).

Значения операндов:

A – имя случайной величины (СЧА), значения которой должны учитываться в таблице; операнд **A** игнорируется дисперсионным анализом, но должен быть определен, когда используется блоками **TABULATE**;

B – ширина первого частотного интервала;

C – ширина всех промежуточных частотных интервалов;

D – количество частотных интервалов таблицы, включая левый и правый (целое положительное число).

Пример:

TU_5 TABLE M1,5,10,4

в таблице с именем **TU_5** будет накапливаться частота попаданий значений резидентного времени транзактов в четыре (D=4) частотных интервала шириной 5 единиц времени для первого интервала и 10 – для остальных трёх интервалов: 0-5; 5-15; 15-25; 25-35; когда активный транзакт попадает в блок **TABULATE TU_5**, в соответствии с операндом **A** в команде **TABLE**, заданным в виде СЧА **M1**, вычисляется время нахождения этого транзакта в модели как разница между текущим моментом модельного времени и моментом поступления транзакта в модель; в зависимости от полученного значения резидентного времени прибавляется единица к накапливаемому значению соответствующего частотного интервала; *для всех значений, превышающих правую границу последнего частотного интервала, единица добавляется в последний интервал*.

6.6.5. QTABLE (ТАБЛИЦА ОЧЕРЕДИ)

Назначение: описание таблицы очереди, используемой в модели для накопления частоты попадания времени нахождения транзакта в очереди (времени ожидания) в заданные частотные интервалы и построения гистограммы плотности распределения.

Формат:

<Имя> TABLE A,B,C,D

Здесь: **<Имя>** – имя таблицы (не более 32-х алфавитно-цифровых символов), в которой будут накапливаться значения частот.

Значения операндов:

A – имя очереди, для которой формируется таблица;

B – ширина первого частотного интервала;

C – ширина всех промежуточных частотных интервалов;

D – количество частотных интервалов таблицы, включая левый и правый (целое положительное число).

Пример:

Gis2u TABLE Stell,10,10,40

в таблице с именем **Gis2u** будет накапливаться частота попаданий значений времени нахождения транзактов в очереди с именем **Stell** в сорока (D=40) частотных интервалах шириной по 10 единиц времени, то есть охватывается временной интервал от 0 до 400 единиц времени; значения, превышающие 400 единиц времени, попадут в последний интервал.

6.6.6. VARIABLE (АРИФМЕТИЧЕСКАЯ ПЕРЕМЕННАЯ)

Назначение: описание арифметической переменной.

Формат:

<Имя> VARIABLE X

Здесь: **<Имя>** – имя арифметической переменной.

Значения операнда:

X – арифметическое выражение для вычисления значения переменной **<Имя>**.

Пример:

Vara1 VARIABLE 5#EXP(V$Grad+2)

когда активный транзакт попадает в блок, в котором используется переменная Vara1, (точнее, ссылка на эту переменную в виде СЧА: V$Vara1), например:

ADVANCE V$Vara1,

вычисляется значение переменной **Vara1** в соответствии с заданным арифметическим выражением как $5e^{Grad+2}$, где **V$Grad** – ссылка на другую арифметическую переменную **Grad**, которая тоже должна быть определена с помощью другого оператора **VARIABLE.**

6.6.7. CLEAR (ОЧИСТИТЬ)

Назначение: возврат процесса моделирования в исходное состояние с возможностью сохранения значений некоторых объектов GPSS-модели.

Формат команды:

CLEAR [A]

Операнд **A** может принимать только два значения: **ON** или **OFF**; необязательный операнд [по умолчанию – **ON**].

Когда операнд **A** равен OFF, ячейки, логические ключи и элементы матриц остаются без изменений.

6.6.8. CONTINUE (ПРОДОЛЖИТЬ)

Назначение: возобновление прерванного процесса моделирования.

Формат команды:

CONTINUE

6.6.9. HALT (ОСТАНОВИТЬ)

Назначение: прерывает процесс моделирования и очищает очередь команд. Является *срочной* командой.

Формат команды:

HALT

6.6.10. INCLUDE (ВКЛЮЧИТЬ)

Назначение: вставка в исходную модель и трансляция файла с операторами.

Формат команды:

INCLUDE A

A – полный путь доступа к указанному файлу.

Если **A** – имя файла (без указания пути доступа), то предполагается, что вставляемый файл находится в той же папке, что и исходная модель.

6.6.11. REPORT (СОЗДАТЬ ОТЧЁТ)

Назначение: немедленное создание отчета.

Формат команды:

REPORT

6.6.12. RESET (СБРОСИТЬ)

Назначение: сброс в ноль статистики и атрибутов системы.

Формат команды:

RESET

6.6.13. SHOW (ПОКАЗАТЬ)

Назначение: отображает значение выражения в строке состояния окна «Model».

Формат команды:

SHOW X

Операнд **X** представляет собой выражение (арифметическое или логическое), значение которого необходимо отобразить в строке состояния окна «Model».

6.6.14. START (НАЧАТЬ)

Назначение: запуск процесса моделирования.

Формат команды:

START A,[B],,[D]

A – начальное значение «счетчика завершений»;

B – признак вывода статистики: значение NP (no printout) блокирует вывод стандартной статистики; необязательный операнд;

D – признак вывода списков: значение **1** включает вывод списков будущих и текущих событий в стандартный отчет; необязательный операнд.

Операнд **C** остался от предыдущих версий GPSS и *не используется* в GPSS World.

6.6.15. STEP (ШАГАТЬ)

Назначение: остановка процесса моделирования по определенному количеству входов транзактов в блоки.

Формат команды:

STEP A

A – количество входов в блок (положительное целое число).

Пример: команда

STEP 1

приводит к приостановке процесса моделирования всякий раз, когда транзакт входит в очередной блок.

6.6.16. STOP (ОСТАНОВИТЬ)

Назначение: устанавливает или снимает условие прерывания моделирования.

Формат команды:

STOP [A],[B],[C]

A – номер транзакта (положительное целое число);

B – номер блока (положительное целое число) или метка блока (имя);

C – флаг состояния команды:

- ON – устанавливает условие прерывания;
- OFF – снимает условие прерывания;
- по умолчанию ON.

При отсутствии:

- операнда **A** – любой транзакт, входящий в блок с номером **B**, вызывает условие прерывания;
- операнда **B** – транзакт с номером **A**, при входе в любой блок вызывает условие прерывания;
- операндов **A** и **B** – процесс моделирования немедленно прерывается.

Пример: команда

STOP 100,21

определяет условие прерывания процесса моделирования: при входе транзакта с номером 100 в блок с номером 21. Продолжение моделирования – команда CONTINUE.

6.7. GPSS-модели массового обслуживания

«Если программа полностью отлажена, ее нужно скорректировать» ***(Законы программирования)***

Рассмотрим принципы построения GPSS-моделей на примерах моделей систем (СМО) и сетей (СеМО) массового обслуживания с однородным и неоднородным потоком заявок. GPSS-модели представлены в порядке возрастания сложности. Вначале рассматриваются и подробно комментируются простейшие GPSS-модели, имитирующие работу СМО с однородным потоком заявок и позволяющие получить представление об основных операторах GPSS World. По мере усложнения моделей вводятся новые операторы, необходимые для построения более сложных GPSS-моделей.

Для каждой модели представлено подробное описание моделируемой системы с указанием конкретных значений параметров. Далее приводится текст GPSS-модели и детально рассматривается каждый оператор. Все операторы GPSS-моделей сопровождаются комментариями. Для некоторых моделей приводятся и подробно описываются стандартные отчеты, формируемые автоматически по завершению моделирования и содержащие результаты моделирования.

6.7.1. Модель 1: одноканальная СМО с детерминированным потоком заявок и равномерно распределенной длительностью обслуживания (D/U/1)

Положим, что система содержит один обслуживающий прибор (рис.6.6). В СМО поступает детерминированный поток заявок с интервалом 10 секунд. Заявки выбираются на обслуживание из накопителя неограниченной ёмкости в порядке поступления, то есть по правилу «первым пришел – первым обслужен» (дисциплина обслуживания FIFO – First In First Out).

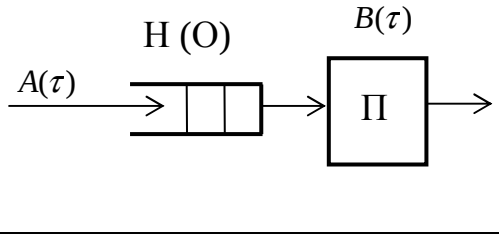


Рис.6.6. Одноканальная СМО

Длительность обслуживания заявок в приборе – величина случайная, распределенная по равномерному закону в интервале от 4 до 12 секунд (8±4 секунды) со средним значением 8 секунд.

Краткое описание рассматриваемой СМО имеет следующий вид:

- количество обслуживающих приборов: 1;
- количество потоков (классов) заявок: 1;
- ёмкость накопителя: *не ограничена* (равна бесконечности);
- интервалы между заявками в потоке: 10 секунд;
- поток заявок: *детерминированный*;
- значение длительности обслуживания заявок в приборе: 8±4 секунд;
- закон распределения длительности обслуживания заявок в приборе: *равномерный*.

Текст GPSS-модели:

```
**********************************************************************
      GENERATE    10; формирование детерминированного потока заявок
      QUEUE       1; отметка момента поступления заявки в очередь 1
      SEIZE       uzel; занятия прибора с именем uzel
      DEPART      1; отметка момента покидания заявкой очереди 1
      ADVANCE     8,4; задержка на время 8±4 единицы времени
      RELEASE     uzel; освобождение прибора с именем uzel
      TERMINATE   1; удаление заявки из модели
**********************************************************************
      START       100000
**********************************************************************
```

Рассмотрим подробно представленную модель и прокомментируем каждый оператор GPSS-модели, сопоставив их с реально протекающими в системе процессами.

Первый оператор **GENERATE** формирует в модели через каждые 10 единиц модельного времени транзакты. Множество формируемых таким образом транзактов моделируют процесс поступления заявок в

систему, образующих детерминированный поток с интервалом 10 секунд.

Когда модельное время становится равным моменту формирования очередного транзакта, последний начинает движение в модели к следующему по порядку оператору **QUEUE**, который заносит транзакт (заявку) в очередь с именем «1». (В действительности же, все транзакты сохраняются в очереди даже при отсутствии оператора QUEUE. Оператор QUEUE отмечает момент поступления транзакта в очередь с целью сбора статистики по очередям).

Далее транзакт продолжает движение к следующему оператору **SEIZE**, в соответствии с которым выполняет попытку занять одноканальное устройство (прибор) с именем «uzel». При этом проверяется занятость устройства. Если прибор занят обслуживанием ранее поступившего транзакта, то рассматриваемый транзакт приостанавливает свое движение и остается в очереди до тех пор, пока не освободится прибор. Если прибор свободен, то рассматриваемый транзакт продвигается к следующему оператору DEPART.

Оператор **DEPART** отмечает момент покидания транзактом очереди с именем (номером) «1» с целью сбора статистики по очередям (определяется время нахождения транзакта в очереди, то есть время ожидания заявки). Двигаясь дальше, транзакт попадает в оператор ADVANCE. Оператор **ADVANCE** задерживает транзакт на случайную величину, формируемую по равномерному закону распределения из интервала 8±4, моделируя, таким образом, процесс обслуживания заявок в приборе. Дальнейшее движение транзакта в модели возможно только тогда, когда значение модельного времени достигнет момента завершения обслуживания заявки в приборе.

При попадании транзакта в операторе **RELEASE** выполняется совокупность действий по освобождению прибора с именем «uzel».

Затем транзакт попадает в последний оператор **TERMINATE**, который выводит транзакт из модели (уничтожает транзакт), при этом из «**Счетчика завершений»** вычитается значение, указанное в качестве операнда A оператора TERMINATE и равное 1 в нашем примере.

Процесс моделирования продолжается до тех пор, пока значение «**Счетчика завершений»** не станет равным нулю.

Начальное значение «**Счетчика завершений»**, указываемое в качестве операнда A, устанавливается с помощью команды **START**, которая одновременно запускает процесс моделирования. Таким образом, моделирование в данном примере завершится после прохождения через модель 100 тысяч транзактов (после обслуживания в моделируемой системе 100 тысяч заявок).

Команда START может находиться *непосредственно в модели* или же может быть задана отдельно, *после трансляции* модели. В первом случае, после трансляции модели сразу же начинается ее выполнение. Во втором случае, выполнение модели начинается только после запуска

команды START.

По завершению моделирования результаты формируются автоматически в виде стандартного отчета, представленного на рис.6.7.

```
                GPSS World Simulation Report - CMO_DU1.2.1

                    Wednesday, January 25, 2006 11:58:53

          START TIME      END TIME  BLOCKS  FACILITIES  STORAGES
              0.000   1000005.010     7        1          0

                        NAME                VALUE
                       UZEL               10000.000

     LABEL          LOC  BLOCK TYPE     ENTRY COUNT CURRENT COUNT RETRY
                1    GENERATE         100000             0       0
                2    QUEUE            100000             0       0
                3    SEIZE            100000             0       0
                4    DEPART           100000             0       0
                5    ADVANCE          100000             0       0
                6    RELEASE          100000             0       0
                7    TERMINATE        100000             0       0

FACILITY    ENTRIES  UTIL.   AVE. TIME AVAIL. OWNER PEND INTER RETRY DELAY
UZEL        100000    0.801       8.008  1        0    0     0     0      0

QUEUE         MAX CONT. ENTRY ENTRY(0) AVE.CONT. AVE.TIME   AVE.(-0) RETRY
 1               1    0 100000  69780     0.040      0.405      1.339   0

   FEC XN    PRI      BDT      ASSEM  CURRENT  NEXT  PARAMETER    VALUE
             100001    0     1000010.000   100001      0      1
```

Рис.6.7. Стандартный отчет Модели 1

Стандартный отчет рассматриваемой модели содержит следующую информацию.

1. Заголовок с именем GPSS-модели:

```
GPSS World Simulation Report - CMO_DU1.2.1
```

2. Дату и время проведения имитационного моделирования (эксперимента):

```
Wednesday, January 25, 2006 11:58:53
```

3. Время старта и завершения моделирования, количество блоков (операторов), одноканальных устройств (приборов) и многоканальных устройств (памятей) в GPSS-модели:

```
START TIME          END TIME  BLOCKS  FACILITIES  STORAGES
    0.000        1000005.010     7        1          0
```

4. Перечень заданных в модели символических имен (блоков, устройств, памятей) и присвоенные им числовые значения (начиная с 10000):

```
NAME                          VALUE
UZEL                          10000.000
```

5. Перечень (BLOCK TYPE) пронумерованных (LOC) блоков с присвоенными им в модели метками (LABEL):

```
LABEL              LOC  BLOCK TYPE     ENTRY COUNT CURRENT COUNT RETRY
                    1    GENERATE         100000             0       0
                    2    QUEUE            100000             0       0
```

```
                    3     SEIZE          100000          0       0
                    4     DEPART         100000          0       0
                    5     ADVANCE        100000          0       0
                    6     RELEASE        100000          0       0
                    7     TERMINATE      100000          0       0
```

Кроме того, для каждого блока указывается:

ENTRY COUNT – количество транзактов, вошедших в данный блок за время моделирования;

CURRENT COUNT – количество транзактов, в данном блоке на момент завершения моделирования;

RETRY – количество транзактов, ожидающих выполнения некоторого специфического условия.

6. Результаты моделирования и дополнительная информация по устройствам:

```
FACILITY   ENTRIES   UTIL.    AVE.TIME  AVAIL. OWNER PEND INTER RETRY DELAY
 UZEL       100000   0.801       8.008  1        0    0    0     0      0
```

Здесь:

FACILITY – символическое имя или номер устройства;

ENTRIES – количество транзактов, вошедших в данное устройство за время моделирования;

UTIL. – коэффициент использования (загрузка) устройства;

AVE.TIME – среднее время занятия устройства одним транзактом (средняя длительность обслуживания заявок);

AVAIL. – состояние устройства на момент завершения моделирования: 1 – устройство доступно (не занято), 0 - устройство недоступно (занято);

OWNER – номер транзакта, находящегося в устройстве на момент завершения моделирования;

PEND – количество транзактов, ожидающих выполнения с прерыванием других транзактов;

INTER – количество прерванных транзактов на момент завершения моделирования (в списке прерываний);

RETRY – количество транзактов, ожидающих выполнения некоторого специфического условия;

DELAY – количество транзактов, ожидающих занятия устройства.

7. Результаты моделирования и дополнительная информация по очередям:

```
QUEUE          MAX CONT. ENTRY ENTRY(0) AVE.CONT. AVE.TIME   AVE.(-0) RETRY
 1               1    0 100000  69780     0.040      0.405      1.339   0
```

Здесь:

QUEUE – имя или номер очереди;

MAX – максимальное количество транзактов в очереди за время моделирования;

CONT. – текущее количество транзактов в очереди на момент завершения моделирования;

ENTRY – количество транзактов, прошедших через очередь за время моделирования;

ENTRY(0) – количество транзактов, прошедших через очередь за время моделирования с нулевым временем ожидания;
AVE.CONT. – средняя длина очереди за время моделирования;
AVE.TIME – среднее время нахождения транзакта в очереди (среднее время ожидания заявок);
AVE.(-0) – среднее время нахождения транзакта в очереди без учета транзактов с нулевым временем ожидания;
RETRY – количество транзактов, ожидающих выполнения некоторого специфического условия;

8. Список будущих событий (FEC):

```
FEC XN   PRI          BDT      ASSEM  CURRENT  NEXT  PARAMETER    VALUE
100001    0     1000010.000   100001       0      1
```

Здесь:
FEC – Future Events Chain;
XN – номера всех транзактов, находящихся в списке будущих событий (в данном примере это единственный транзакт с номером 100001);
PRI – приоритет транзакта;
BDT – момент времени, когда транзакт должен покинуть блок, а, следовательно, и список будущих событий);
ASSEM – номер семейства данного транзакта;
CURRENT – номер блока, в котором находился транзакт на момент завершения моделирования;
NEXT – номер следующего блока, в который будет передан транзакт;
PARAMETER – имя или номер параметра транзакта;
VALUE – значение параметра.

6.7.2. Модель 1.A: одноканальная СМО с простейшим потоком заявок (M/U/1)

Положим теперь, что в рассмотренную выше одноканальную СМО поступает простейший поток заявок, интервалы между которыми распределены по экспоненциальному закону со средним значением 10 секунд.

Текст GPSS-модели:

```
**************************************************************
      GENERATE     (Exponential(5,0,10))
      QUEUE        1
      SEIZE        uzel
      DEPART       1
      ADVANCE      (Uniform(25,4,12))
      RELEASE      uzel
      TERMINATE    2
**************************************************************
      START        100000
**************************************************************
```

Рассмотрим изменения, внесенные в предыдущую модель и выделенные жирным шрифтом.

Первое изменение – в операторе GENERATE, в первом операнде которого указан закон распределения интервалов между генерируемыми транзактами в виде библиотечной процедуры **Exponential(5,0,10),** обеспечивающей формирование случайных величин с экспоненциальным законом распределения. Три параметра процедуры **Exponential(5,0,10)** задают соответственно: 5 – номер исходного стандартного (встроенного) генератора равномерно распределенных случайных величин; 0 – смещение вырабатываемой случайной величины; 10 – среднее значение интервала.

Второе изменение в операторе ADVANCE, в котором используется другая форма задания равномерно распределенной случайной величины – в виде библиотечной процедуры **(Uniform(25,4,12))** из встроенной в GPSS библиотеки вероятностных распределений. Первый параметр процедуры **Uniform** определяет номер встроенного генератора равномерно распределённых в интервале (0; 1) случайных величин, а второй и третий параметры – границы интервала (соответственно нижняя и верхняя) формируемой равномерно распределенной случайной величины.

Третье изменение в операторе TERMINATE, в котором параметр A задан равным 2. Это значит, что при каждом попадании транзакта в этот оператор из счетчика завершений будет вычитаться не 1, как в предыдущем примере, а 2. Следовательно, моделирование завершится после прохождения через модель не 100 000 транзактов, а только 50 000.

6.7.3. Модель 2: многоканальная СМО с накопителем ограниченной ёмкости и обслуживанием заявок по закону Эрланга ($M/E_2/1/r$)

Положим теперь, что в предыдущую модель внесены следующие изменения (рис.6.8):

1) система содержит K=4 идентичных обслуживающих приборов, причём заявка может занять любой свободный прибор;
2) накопитель имеет ограниченную ёмкость $r = 10$, при этом заявка, заставшая накопитель заполненным, получает отказ в обслуживании и теряется;
3) длительность обслуживания заявок в одном приборе распределена по закону Эрланга 2-го порядка со средним значением 40 секунд.

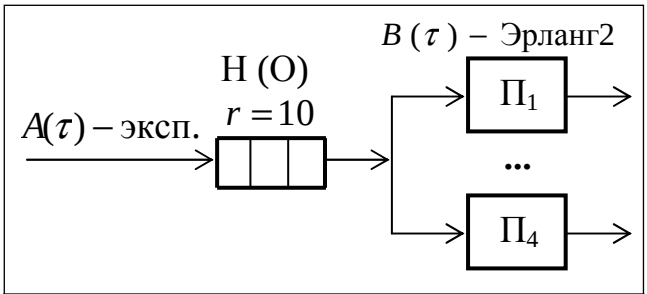


Рис.6.8. Многоканальная СМО с накопителем ограниченной ёмкости

Текст GPSS-модели с *комментариями* (*выделены курсивом*):

```
Uzel    STORAGE     4; задание числа приборов в устройстве с именем Uzel
**************************************
*Область исполняемых блоков (Основной модуль)
        GENERATE    (Exponential(11,0,10)); формирование простейшего потока
        TEST  L     Q$ch_1,10,Otkaz; проверка длины очереди
        QUEUE       ch_1; регистрация момента поступления заявки в очередь ch_1
        ENTER       Uzel; попытка занять один из приборов устройства Uzel
        DEPART      ch_1; регистрация момента покидания заявки очереди ch_1
        ADVANCE     (Exponential(21,0,20)+Exponential(31,0,20)); задержка заявки
*                                  в среднем на 40 единиц модельного времени
        LEAVE       Uzel; освобождение одного прибора многоканального
*                   устройства Uzel
        TERMINATE   1; удаление обслуженной заявки из модели и уменьшение
*                  счетчика завершений
Otkaz   TERMINATE   1; удаление заявки, получившей отказ
```

Рассмотрим изменения, внесенные в предыдущую GPSS-модель и выделенные жирным шрифтом.

Первое изменение заключается в появлении в GPSS-модели «Области описания», которая содержит оператор **STORAGE**, задающий имя (**Uzel**) многоканального устройства (памяти) и количество обслуживающих приборов (ёмкость памяти), равное 4.

Второе изменение заключается в появлении в GPSS-модели нового оператора (блока) **TEST**, позволяющего смоделировать накопитель с ограниченной ёмкостью перед многоканальным устройством.

Рассмотрим оператор **TEST** более подробно в контексте данного примера. Для этого сопоставим оператор **TEST**, записанный в общем виде, с оператором **TEST** в нашей модели:

TEST X A, B, C
TEST L Q$ch_1, 10, Otkaz

Здесь:

X – условный оператор (в нашем примере **L** означает «меньше»);

A – СЧА, значение которого проверяется в соответствии с заданным условным оператором (в нашем примере **Q$ch_1** означает проверку длины очереди с именем **ch_1**);

B – контрольное значение, с которым сравнивается значение числового атрибута, указанного в параметре A (в нашем примере длина очереди **ch_1** сравнивается во значением 10);

C – имя альтернативного оператора, которому передается транзакт, если указанное условие не выполняется (в нашем примере транзакт будет передан оператору **TERMINATE** с именем **Otkaz**).

Таким образом, транзакт, попав в указанный опреатор **TEST**, перейдет к следующему по порядку оператору при условии, что длина очереди **ch_1** меньше 10, и к оператору **TERMINATE** с меткой **Otkaz**, если в очереди **ch_1** уже находятся 10 заявок.

Третье изменение состоит в использовании операторов **ENTER** и **LEAVE,** моделирующих занятие и освобождение многоканального устройства, вместо операторов SEIZE и RELEASE, использующихся для одноканального устройства. Заметим, что в операторах **ENTER** и **LEAVE,** в отличие от SEIZE и RELEASE, могут использоваться два операнда A и B, где второй операнд B определяет количество занимаемых или освобождаемых приборов (каналов), причем при отсутствии операнда B его значение по умолчанию принимается равным 1.

В операторе ADVANCE реализуется случайная задержка заявки в соответствии с законом распределения Эрланга 2-го порядка в виде суммы двух экспоненциально распределенных случайных величин со средними значениями в 20 секунд (одна единица модельного времени равна одной секунде) так, что средняя задержка заявки в приборе составляет 40 секунд.

Еще одной особенностью данной модели является наличие двух операторов **TERMINATE**. Первый оператор удаляет из модели *обслуженные* заявки (транзакты), при этом из «**Счетчика завершений»** вычитается единица. Второй оператор удаляет из модели *необслуженны*е заявки, то есть заявки, заставшие при поступлении в систему накопитель заполненным и получившие отказ в обслуживании, при этом из «**Счетчика завершений»** также вычитается единица. Возникает вопрос: «Можно ли для вывода из модели обслуженных и необслуженных заявок использовать только один оператор **TERMINATE**?». Ответ: «Да, можно!». Зачем же тогда надо было использовать 2 оператора **TERMINATE**? Ответ достаточно простой. Второй оператор нужен только для того, чтобы получить информацию о доле обслуженных и доле потерянных (не обслуженных) заявок. Из стандартного отчета (рис.6.9) видно, что число обслуженных транзактов, прошедших через первый оператор **TERMINATE,** равно 938291, а число необслуженных (потерянных) транзактов, прошедших через второй оператор **TERMINATE,** равно 61709. Таким образом, вероятность потери заявки в моделируемой системе составляет 61709/(938291+61709)=0,061709, то есть 6,2% от общего числа поступивших в систему заявок. Отметим, что наличие в обоих операторах **TERMINATE** операнда, равного 1, означает, что моделирование завершится при достижении суммарного числа обслуженных и необслуженных заявок, покинувших систему, значения, указанного в операнде A команды **START** (в данной модели это значение равно 500000). Если в *первом* операторе **TERMINATE** операнд будет отсутствовать, что по умолчанию соответствует значению 0, то моделирование завершится, когда число *необслуженных (потерянных*) заявок достигнет указанного в команде **START** значения. И наоборот, если операнд будет отсутствовать во *втором* операторе **TERMINATE**, то моделирование завершится, когда число *обслуженных* заявок достигнет указанного в команде **START** значения.

На рис.6.9 представлен стандартный отчет, полученный для рассмотренной модели при задании команды

START 1000000,

означающем, что моделирование завершается после прохождения через систему *миллиона* заявок (транзактов).

```
               GPSS World Simulation Report - Untitled Model 2.1

                    Thursday, September 21, 2006 20:48:27

           START TIME          END TIME  BLOCKS  FACILITIES  STORAGES
                0.000      10007445.339     9         0          1

              NAME                       VALUE
          CH_1                         10001.000
          OTKAZ                            9.000
          UZEL                         10000.000

 LABEL             LOC  BLOCK TYPE     ENTRY COUNT CURRENT COUNT RETRY
                   1    GENERATE       1000006             0       0
                   2    TEST           1000006             0       0
                   3    QUEUE           938297             2       0
                   4    ENTER           938295             1       0
                   5    DEPART          938294             0       0
                   6    ADVANCE         938294             3       0
                   7    LEAVE           938291             0       0
                   8    TERMINATE       938291             0       0
OTKAZ              9    TERMINATE        61709             0       0

QUEUE          MAX CONT. ENTRY ENTRY(0) AVE.CONT. AVE.TIME   AVE.(-0) RETRY
 CH_1           10    3 938297 138746     4.231     45.125     52.955   0

STORAGE        CAP. REM. MIN. MAX.  ENTRIES AVL.  AVE.C. UTIL. RETRY DELAY
 UZEL            4    0   0     4   938295   1    3.753  0.938    0    2

CEC XN   PRI        M1        ASSEM  CURRENT  NEXT  PARAMETER    VALUE
1000004    0  10007414.480   1000004      4      5

FEC XN   PRI         BDT      ASSEM  CURRENT  NEXT  PARAMETER    VALUE
1000007   0    10007447.972   1000007     0      1
999995    0    10007482.391   999995      6      7
1000003   0    10007495.279   1000003     6      7
1000001   0    10007496.268   1000001     6      7
```

Рис.6.9. Стандартный отчет к Модели 2

Следует обратить внимание на то, что завершение процесса моделирования происходит по числу транзактов, прошедших через операторы TERMINATE, а не по числу транзактов, сформированных оператором GENERATE. В нашей модели через операторы TERMINATE с номерами (LOC) 8 и 9 прошли соответственно 938291 и 61709 транзактов (см. раздел LABEL отчета), что в сумме составляет ровно 1000000 транзактов, как указано в команде START. В то же время, количество транзактов, сформированных в операторе GENERATE равно 1000006, то есть на 6 транзактов больше, чем покинуло модель. Эти шесть транзактов на момент завершения моделирования остались в модели и, как видно в том же разделе отчета, они находятся в блоках QUEUE (2 транзакта), ENTER (1 транзакт), ADVANCE (3 транзакта).

6.7.4. Модель 2.A: дополнительная статистика в виде гистограмм

Стандартный отчёт, формируемый автоматически после завершения моделирования, если в команде START не был задан параметр NP, содержит основные результаты моделирования, состав которых может быть задан перед моделированием. Для этого в главном меню нужно выбрать пункт «Edit/Settings …» и на странице «Reports» («Отчёты») журнала настроек модели с помощью набора флажков задать состав результатов, включаемых в отчёт.

В некоторых случаях требуется получить результаты моделирования не только в виде средних значений вероятностных характеристик, но и в виде гистограмм, отображающих законы (плотности) распределений случайных величин.

Положим, что в рассмотренной модели 2 результаты моделирования должны быть представлены в виде гистограмм плотностей распределений времени ожидания и времени пребывания заявок.

Ниже представлен **текст GPSS-модели** с соответствующими добавлениями:

```
**************************************************************************
* Область описания
Uzel     STORAGE   4; задание числа приборов в устройстве с именем Uzel
T_w      QTABLE    ch_1,15,15,10
T_u      TABLE     M1,30,30,10
  ****************************************
*  Область исполняемых блоков (Основной модуль)
         GENERATE  (Exponential(11,0,10)); формирование простейшего потока
         TEST L    Q$ch_1,10,Otkaz; проверка длины очереди  ch_1
         QUEUE     ch_1; регистрация момента поступления заявки в очередь
         ENTER     Uzel; попытка занять один из приборов устройства Uzel
         DEPART    ch_1;  регистрация момента покидания очереди ch_1
         ADVANCE   (Exponential(21,0,20)+Exponential(31,0,20)); задержка
*                  заявки в среднем на 40 единиц модельного времени
         LEAVE     Uzel; освобождение одного прибора многоканального
*                  устройства Uzel
         TABULATE  T_u
         TERMINATE 1; удаление обслуженной заявки из модели и уменьшение
*                  счетчика завершений
Otkaz    TERMINATE 1; удаление заявки, получившей отказ
**************************************************************************
```

Рассмотрим изменения, внесенные в предыдущую модель и выделенные жирным шрифтом.

Во-первых, в области описания появились два новых оператора – команды: **QTABLE** и **TABLE**.

Сопоставим эти операторы, записанные в общем виде, с операторами в нашей модели:

<Имя> QTABLE A, B, C, D
T_w QTABLE ch_1, 15, 15, 10

<Имя> TABLE A, B, C, D
T_u TABLE M1, 30, 30, 10

Первый оператор (команда) **QTABLE** формирует таблицу для гистограммы плотности распределения времени ожидания заявок в очереди, имя которой указано в операнде A.

Имя **T_w** задаёт имя таблицы (гистограммы), а операнды A, B, C и D задают соответственно:

A=**ch_1** – имя очереди, для которой формируется гистограмма;

B=**15** – верхнюю (правую) границу первого частотного интервала гистограммы;

C=**15** – величину всех остальных частотных интервалов;

D=**10** – количество частотных интервалов.

Второй оператор **TABLE** формирует таблицу для гистограммы плотности распределения времени пребывания заявок в системе.

Имя **T_u**, как и в предыдущем случае, задает имя таблицы (гистограммы), а операнды A, B, C и D задают соответственно:

A=**M1** – величину, для которой формируется гистограмма; в нашем примере **M1** представляет собой СЧА, определяющее резидентное время, вычисляемое как разность между текущим значением модельного времени, определяемым в момент вхождения транзакта в блок **TABULATE**, и временем появления транзакта в модели, то есть временем поступления заявки в систему, являющимся одним из параметров транзакта;

B=**30** – верхнюю границу первого частотного интервала;

C=**30 –** величину всех остальных частотных интервалов;

D=**10** – количество частотных интервалов.

Таким образом, команда **TABLE** используется совместно с блоком **TABULATE**, который регистрирует момент прохождения транзактом (заявкой) определенного места в модели. Соответственно блок **TABULATE** должен находиться в модели в том месте, относительно которого измеряется искомое время. Таким местом при измерении времени пребывания заявки в моделируемой системе является точка выхода заявки из системы, когда транзакт покидает прибор многоканальной системы. В качестве параметра A оператора **TABULATE** выступает имя соответствующей таблицы (гистограммы). В нашем случае эта таблица и соответствующая ей гистограмма имеет имя **T_u.**

Оператор **TABLE** так же, как и **QTABLE**, позволяет сформировать гистограмму плотности распределения случайной величины и имеет аналогичную структуру. Основное отличие **TABLE** от **QTABLE** состоит в том, что оператор **TABLE** позволяет формировать гистограмму плотности распределения случайной величины между двумя, в общем случае, *произвольными моментами времени*, в то время как **QTABLE** всегда формирует гистограмму плотности распределения *времени ожидания в очереди*.

На рис.6.10 представлен фрагмент стандартного отчета, полученного для рассмотренной модели при задании команды

START 100000,

означающем, что моделирование завершено после прохождения через систему *ста тысяч* заявок (транзактов).

```
                                  ...
 LABEL            LOC   BLOCK TYPE     ENTRY COUNT CURRENT COUNT RETRY
                  1     GENERATE         100006             0       0
                  2     TEST             100006             0       0
                  3     QUEUE             93548             2       0
                  4     ENTER             93546             1       0
                  5     DEPART            93545             0       0
                  6     ADVANCE           93545             3       0
                  7     LEAVE             93542             0       0
                  8     TABULATE          93542             0       0
                  9     TERMINATE         93542             0       0
OTKAZ            10    TERMINATE          6458              0       0

QUEUE         MAX CONT. ENTRY ENTRY(0) AVE.CONT. AVE.TIME   AVE.(-0) RETRY
 CH_1          10    3  93548  13027     4.365     46.404     53.912   0

STORAGE       CAP. REM. MIN. MAX.  ENTRIES AVL.  AVE.C. UTIL. RETRY DELAY
 UZEL           4    0   0     4    93546   1    3.766  0.941    0    2

TABLE          MEAN    STD.DEV.       RANGE            RETRY FREQUENCY CUM.%
 T_W          46.405   35.680                            0
                                    _ -       15.000          23766    25.41
                              15.000 -       30.000          11890    38.12
                              30.000 -       45.000          11926    50.87
                              45.000 -       60.000          12229    63.94
                              60.000 -       75.000          11917    76.68
                              75.000 -       90.000           9946    87.31
                              90.000 -      105.000           6324    94.07
                             105.000 -      120.000           3365    97.67
                             120.000 -      135.000           1428    99.19
                             135.000 -  _                      754   100.00
 T_U          86.443   45.393                              0
                                    _ -       30.000           9684    10.35
                              30.000 -       60.000          19501    31.20
                              60.000 -       90.000          22921    55.70
                              90.000 -      120.000          20461    77.58
                             120.000 -      150.000          12584    91.03
                             150.000 -      180.000           5605    97.02
                             180.000 -      210.000           1954    99.11
                             210.000 -      240.000            594    99.75
                             240.000 -      270.000            168    99.93
                             270.000 -  _                       70   100.00
                                  ...
```

Рис.6.10. Фрагмент отчета к модели 2.А

Жирным шрифтом в отчёте выделены результаты формирования двух таблиц для построения гистограмм плотности распределения:

T_W – *времени ожидания* заявок в очереди;

T_U – *времени пребыв*ания заявок в системе.

На рис. 6.11 показаны гистограммы плотностей распределений времени пребывания **T_U** и времени ожидания **T_W** заявок, полученные для рассматриваемой модели.

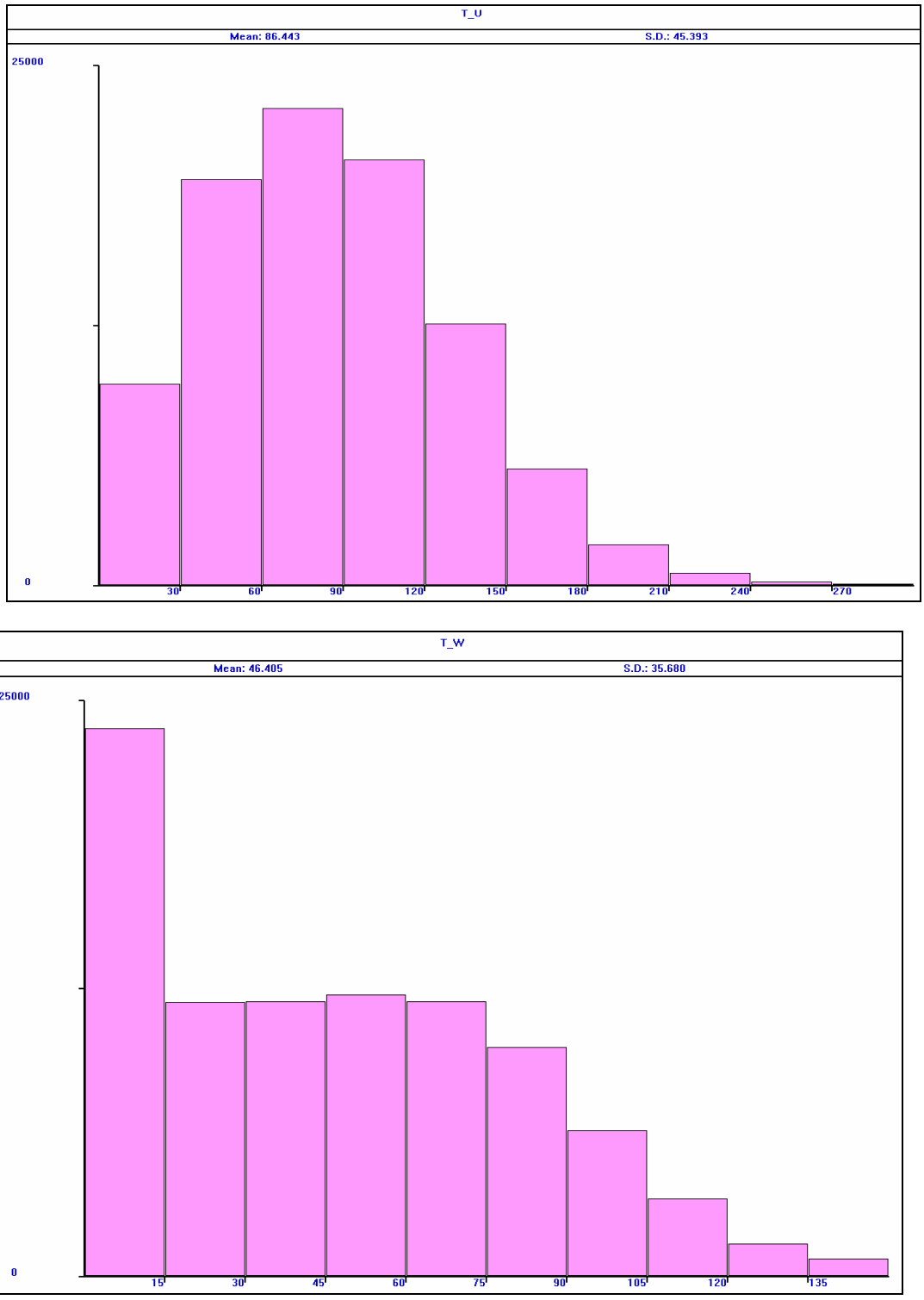


Рис.6.11. Гистограммы плотностей распределений времени пребывания T_U и времени ожидания T_W заявок

Для каждой из таблиц в отчёте приведены следующие данные:

MEAN – среднее значение соответствующей случайной величины;

STD.DEV. – стандартное отклонение случайной величины;

RANGE – нижние и верхние границы частотного класса (интервала);

RETRY – количество транзактов, ожидающих выполнения специфического условия, зависящего от состояния данной таблицы;

FREQUENCY – количество случайных значений, попавших в данный интервал; всякий раз увеличивается на единицу, если значение случайной величины больше нижней границы и меньше или равно верхней границе данного интервала; нижняя граница последнего интервала принимается равной бесконечности, то есть все случайные величины, значения которых больше нижней границы последнего частотного интервала, "попадают" в последний интервал;

CUM.% - накопленная частота, выраженная в процентах от общего количества случайных значений.

Следует отметить наличие определенных проблем, возникающих при задании длин и количества частотных интервалов, задаваемых в качестве операндов команд QTABLE и TABLE. Очевидно, что наглядность и, вытекающая отсюда, информативность гистограмм распределения случайных величин существенно зависит от количества частотных интервалов. Естественно, что для наглядности желательно иметь большое количества частотных интервалов. Однако, чем больше частотных интервалов, тем большую выборку случайных величин необходимо иметь, для того чтобы получить объективную картину, что не всегда возможно и целесообразно. В то же время небольшое количество частотных интервалов (в пределе только 1 интервал) не даёт объективной картины, позволяющей судить о законе распределения анализируемой случайной величины. Таким образом, задание длин и количества частотных интервалов является непростой задачей. Обычно их значения подбираются экспериментальным путем в процессе нескольких реализаций имитационной модели или же на основе предполагаемых значений математического ожидания и среднеквадратического отклонения соответствующей случайной величины.

По значениям среднеквадратического отклонения (S.D.) и математического ожидания (Mean) можно рассчитать коэффициенты вариации времени пребывания ν_u и времени ожидания ν_w заявок: $\nu_u = \frac{45{,}393}{86{,}443} \cong 0{,}525$ и $\nu_w = \frac{35{,}680}{46{,}405} \cong 0{,}769$, значения которых свидетельствуют о близости соответствующих законов распределений к распределению Эрланга 4-го и 2-го порядка соответственно ($k = 1/\nu^2$).

6.7.5. Модель 3: многоканальная СМО с неоднородным потоком заявок и накопителем ограниченной емкости

Внесем теперь в предыдущую модель четырехканальной СМО следующие изменения (рис.6.12):

1) ёмкость накопителя ограничена и равна 4;

2) в систему поступают два класса заявок:

- заявки 1-го класса образуют простейший поток со средним значением интервалов между заявками 20 секунд, и

длительность их обслуживания в приборе постоянна и равна 50 секундам;

- заявки 2-го класса образуют случайный равномерный поток с интервалами между заявками 18±10 секунд и длительностью их обслуживания в приборе, распределённой по экспоненциальному закону со средним значением 40 секунд.

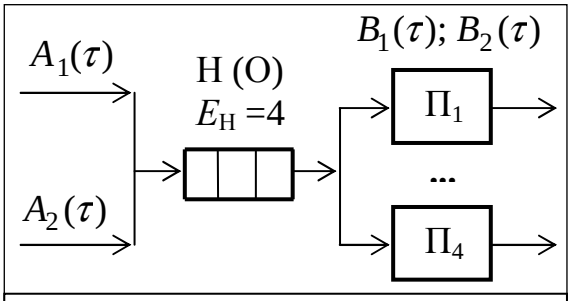


Рис.6.12. Многоканальная СМО с неоднородным потоком заявок

Заявки обоих классов поступают в один и тот же накопитель и выбираются на обслуживание в порядке поступления, то есть в соответствии с дисциплиной обслуживания FIFO.

Текст GPSS-модели с *комментариями* (*выделены курсивом*):

```
**********Модель 3**********
* Область описания
Uzel    STORAGE   4; задание числа приборов в устройстве с именем Uzel
Tw      QTABLE    1,2,2,40
Tu_1    TABLE     M1,50,4,40
Tu_2    TABLE     M1,7,7,40
**************************************
*Модуль 1: моделирование процессов поступления и обслуживания заявок 1-го класса
        GENERATE  (Exponential(1,0,20)); формирование простейшего потока
        TEST  L   Q1,4,Otk_1; проверка длины очереди
        QUEUE     1; регистрация момента поступления заявки в очередь 1
        ENTER     Uzel; попытка занять один из приборов устройства Uzel
        DEPART    1; регистрация момента покидания заявки очереди 1
        ADVANCE   50; задержка заявки на 50 единиц модельного времени
        LEAVE     Uzel; освобождение прибора Uzel
        TABULATE  Tu_1
        TERMINATE 1; удаление из модели обслуженной заявки 1-го класса
Otk_1   ERMINATE  1; удаление не обслуженной заявки 1-го класса
*Модуль 2: моделирование процессов поступления и обслуживания заявок 2-го класса
        GENERATE  18,10; формирование равномерно распределенного потока
        TEST  L   Q1,4,Otk_2; проверка длины очереди
        QUEUE     1; регистрация момента поступления заявки в очередь 1
        ENTER     Uzel; попытка занять один из приборов устройства Uzel
        DEPART    1; регистрация момента покидания заявки очереди 1
        ADVANCE   (Exponential(25,0,40)); задержка заявки 2-го класса
        LEAVE     Uzel; освобождение прибора Uzel
        TABULATE   Tu_2
        TERMINATE 1; удаление из модели обслуженной заявки 2-го класса
Otk_2   TERMINATE 1; удаление не обслуженной заявки 2-го класса
**************************************
        START     500000; запуск модели
```

Краткое описание рассматриваемой СМО:

- количество обслуживающих приборов – 4;
- емкость накопителя – 4;
- количество потоков (классов) заявок – 2;
- закон распределения интервалов между заявками 1-го класса – простейший со средним значением 10 секунд;
- длительность обслуживания заявок 1-го класса – детерминированная и равна 50 секундам;
- закон распределения интервалов между заявками 2-го класса – равномерный с интервалами между заявками 18±10 секунд;
- закон распределения длительности обслуживания заявок 2-го класса – экспоненциальный со средним значением 40 секунд;
- дисциплина буферизации – бесприоритетная с потерей заявки, заставшей в момент поступления накопитель заполненным;
- дисциплина обслуживания – бесприоритетная в порядке поступления (FIFO).

Рассмотрим подробнее представленную GPSS-модель.

В области описания заданы 3 таблицы для построения гистограмм плотностей распределений:

Tw – времени ожидания заявок обоих классов в общей очереди;

Tu_1 – времени пребывания в системе заявок 1-го класса;

Tu_2 – времени пребывания в системе заявок 2-го класса.

Отметим, что таблица **Tw** содержит информацию об усредненном времени ожидания заявок обоих классов.

Исполняемая область модели состоит из двух модулей, каждый из которых моделирует процессы поступления и обслуживания заявок 1-го и 2-го классов.

Последним оператором модели является команда START, задающая значение счетчика завершений равным 500000. Поскольку во всех четырех операторах TERMINATE задано значение операнда A равным 1, то моделирование завершится после прохождения через систему 500 тысяч заявок обоих классов, включая как обслуженные в системе заявки, так и потерянные (не обслуженные) заявки, которые в момент поступления в систему застали накопитель заполненным до конца.

Если в операторах TERMINATE с метками Otk_1 и Otk_2 операнд A не будет указан, то моделирование завершится после прохождения через систему 500 тысяч обслуженных заявок обоих классов, то есть без учета потерянных заявок.

Поскольку команда START включена в состав модели, то после создания модели (трансляции с помощью команды меню «Command/Create Simulation») процесс моделирования начнется автоматически сразу же после завершения трансляции.

6.7.6. Модель 3.A: многоканальная СМО с раздельными накопителями для заявок разных классов

В предыдущей модели время ожидания заявок определяется безотносительно к какому-либо классу, то есть полученное значение является усредненным временем ожидания заявок 1-го и 2-го класса. При этом отсутствует возможность оценки времени ожидания заявок каждого класса в отдельности.

Для определения времени ожидания заявок каждого класса в отдельности можно воспользоваться двумя способами:

1) собирать информацию о времени ожидания заявок 1-го и 2-го классов с помощью двух разных таблиц с использованием операторов TABLE и TABULATE, причем последний должен располагаться перед оператором ADVANCE в обоих исполняемых модулях, отображающих процесс прохождения заявок каждого класса; если при этом сохраняется оператор QTABLE, то в соответствующей таблице будет накапливаться информация об усредненном значении времени ожидания заявок обоих классов;
2) использовать для ожидания заявок 1-го и 2-го классов разные накопители.

Рассмотрим, какие изменения необходимо внести в предыдущую GPSS-модель 3 для реализации второго способа, когда заявки разных классов ожидают в разных накопителях. При этом будем полагать, что ёмкости обоих накопителей одинаковы: $E_{H1}=E_{H2}=2$, а их суммарная ёмкость осталась прежней, равной 4 (рис.6.13).

Текст GPSS-модели с изменениями, выделенными жирным шрифтом, приведён на следующей странице. По сравнению с предыдущей моделью в эту GPSS-модель внесены такие изменения.

В области описания появился второй оператор **QTABLE** с именем таблицы **Tw_2**, в котором в качестве операнда **A** указано имя (номер) накопителя **2**.

В операторах **TEST** модулей 1 и 2 в качестве операндов **A** используются СЧА **Q1** и **Q2**, означающие проверку длин очередей 1 и 2, а в качестве операндов **B** заданы ёмкости соответствующих накопителей, равные в обоих случаях 2. Таким образом, в момент поступления в систему заявки первого или второго класса текущие длины очередей сравниваются с заданными ёмкостями соответствующих накопителей.

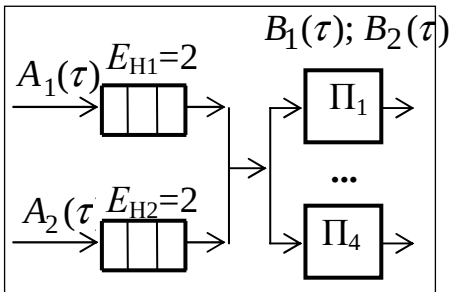


Рис.6.13. СМО с раздельными накопителями

Кроме того, в модуле 2 операнды **A** в операторах **QUEUE** и **DEPART** задают теперь номер очереди 2 для хранения заявок 2-го класса.

```
**********************************************************************
* Область описания
Uzel     STORAGE    4; задание числа приборов в устройстве с именем Uzel
Tw_1     QTABLE     1,2,2,40
Tw_2     QTABLE     2,2,2,40
Tu_1     TABLE      M1,50,4,40
Tu_2     TABLE      M1,7,7,40
**********************************************************************
*Модуль 1: моделирование процессов поступления и обслуживания заявок 1-го класса
         GENERATE   (Exponential(1,0,20)); формирование простейшего потока
         TEST  L    Q1,2,Otk_1; проверка длины очереди
         QUEUE      1; регистрация момента поступления заявки в очередь 1
         ENTER      Uzel; попытка занять один из приборов устройства Uzel
         DEPART     1; регистрация момента покидания заявки очереди 1
         ADVANCE    50; задержка заявки на 50 единиц модельного времени
         LEAVE      Uzel; освобождение прибора Uzel
         TABULATE   Tu_1
         TERMINATE  1; удаление из модели обслуженной заявки 1-го класса
Otk_1    TERMINATE  1; удаление не обслуженной заявки 1-го класса
**********************************************************************
*Модуль 2: моделирование процессов поступления и обслуживания заявок 2-го класса
         GENERATE   18,10; формирование равномерно распределенного потока
         TEST  L    Q2,2,Otk_2; проверка длины очереди 2
         QUEUE      2; регистрация момента поступления заявки в очередь 2
         ENTER      Uzel; попытка занять один из приборов устройства Uzel
         DEPART     2; регистрация момента покидания заявки очереди 2
         ADVANCE    (Exponential(25,0,40)); задержка заявки 2-го класса
         LEAVE      Uzel; освобождение прибора Uzel
         TABULATE   Tu_2
         TERMINATE  1; удаление из модели обслуженной заявки 2-го класса
Otk_2    TERMINATE  1; удаление не обслуженной заявки 2-го класса
**********************************************************************
         START      500000; запуск модели
```

Следует отметить, что результаты моделирования, полученные для СМО с общим накопителем ограниченной ёмкости (модель 3) и с раздельными накопителями ограниченной ёмкости (модель 3.А) для заявок разных классов, будут различны. В то же время, в случае накопителей *с неограниченной ёмкостью* модели 3 и 3.А дадут одинаковые результаты. Для того чтобы убедиться в этом, достаточно в представленных GPSS-моделях закомментировать операторы TEST (поставив в первой позиции символ *), используемые для проверки длины очереди и изменения направления движения транзактов при заполненном накопителе.

6.7.7. Модель 4: одноканальная СМО с относительными приоритетами

Рассмотрим одноканальную СМО с накопителем неограниченной ёмкости с неоднородным потоком заявок и приоритетным обслуживанием заявок разных классов (рис.6.14).

Положим, что в систему поступают 2 класса заявок. Заявки 1-го класса образуют детерминированный поток с интервалом между заявками 30 минут, заявки 2-го класса образуют равномерный поток с интервалами между заявками 15±5,5 минут.

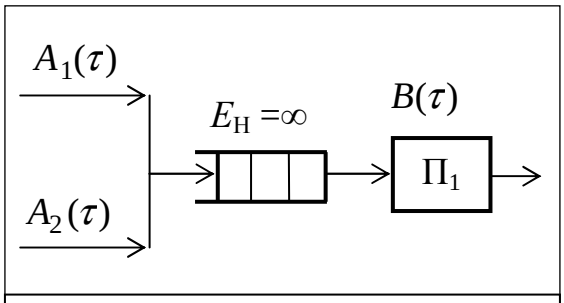


Рис.6.14. Одноканальная СМО с приоритетным обслуживанием

Длительность обслуживания в приборе заявок 1-го и 2-го классов является случайной величиной со средним значением 7 минут 30 секунд и среднеквадратическим отклонением 4 минуты 20 секунд.

Заявки обоих классов поступают в один и тот же накопитель, ёмкость которого не ограничена, и выбираются на обслуживание из накопителя в соответствии с дисциплиной обслуживания с относительными приоритетами, причём заявки 1-го класса имеют более высокий приоритет.

Для формирования в GPSS-модели закона распределения длительности обслуживания заявок воспользуемся аппроксимацией по двум моментам распределения: среднему значению $b = 7{,}5$ и среднеквадратическому отклонению $\sigma_b = 4{,}33$. Для выбора аппроксимирующего распределения рассчитаем коэффициент вариации длительности обслуживания:

$$\nu_b = \frac{\sigma_b}{b} = \frac{4{,}33}{7{,}5} \approx 0{,}577 .$$

В качестве аппроксимирующего распределения случайной величины с коэффициентом вариации, принимающим значения в интервале от 0 до 1, можно воспользоваться распределением Эрланга, коэффициент вариации которого определяется как $\nu = 1/\sqrt{k}$, где k – порядок распределения Эрланга. Тогда:

$$k = \frac{1}{\nu^2} = \frac{1}{0{,}577^2} \approx 3 .$$

Таким образом, в качестве закона распределения длительности обслуживания заявок будем использовать распределение Эрланга 3-го порядка, в соответствии с которым случайная величина формируется как сумма 3-х экспоненциально распределённых случайных величин с математическим ожиданием равным 2,5.

Краткое описание моделируемой СМО:

- количество обслуживающих приборов – 1;
- ёмкость накопителя – не ограничена;
- количество потоков (классов) заявок – 2;
- поток заявок 1-го класса – детерминированный с интервалами между заявками 30 минут;
- поток заявок 2-го класса – случайный с равномерно распределёнными интервалами между поступающими заявками в пределах от 9,5 до 20,5 минут;

- длительность обслуживания заявок 1-го и 2-го класса – случайная, распределённая по закону Эрланга 3-го порядка со средним значением 7,5 минут;
- дисциплина обслуживания заявок – с **относительными** приоритетами.

Текст GPSS-модели с *комментариями* (*выделены курсивом*):

```
* Модуль 1: моделирование процессов поступления и обслуживания заявок 1-го класса
      GENERATE  30,,,,2; детерминированный поток и приоритет, равный 2
      QUEUE     QUzel_1;  поступление заявки в очередь QUzel_1
      SEIZE     Uzel; занятие прибора Uzel
      DEPART    QUzel_1; покидание очереди QUzel_1
      ADVANCE   (Exponential(1,0,2.5)+Exponential(2,0,2.5)+Exponential(3,0,2.5));
      RELEASE   Uzel; освобождение прибора Uzel
      TERMINATE 1; удаление из модели обслуженной заявки 1-го класса
**************************************
* Модуль 2: моделирование процессов поступления и обслуживания заявок 2-го класса
      GENERATE  15,5.5; равномерный поток и приоритет, равный 0
      QUEUE     QUzel_2;  поступление заявки в очередь QUzel_2
      SEIZE     Uzel;  попытка занять один из приборов устройства Uzel
      ADVANCE   (Exponential(4,0,2.5)+Exponential(5,0,2.5)+Exponential(6,0,2.5));
      DEPART    QUzel_2;  покидание очереди QUzel_2
      RELEASE   Uzel; освобождение прибора Uzel
      TERMINATE 1; удаление из модели обслуженной заявки 2-го класса
```

Рассмотрим подробнее представленную GPSS-модель.

Модель включает в себя два модуля, каждый из которых моделирует процессы поступления и обслуживания заявок 1-го и 2-го классов и содержит стандартный набор операторов, используемых для моделирования простейшей одноканальной СМО с однородным потоком заявок.

Особенности данной модели, на которые следует обратить внимание, заключаются в следующем.

В первом операторе **GENERATE** в качестве пятого операнда, используемого для задания уровня приоритета формируемых транзактов (заявок первого класса), задано значение 2. Во втором операторе **GENERATE** значение пятого операнда не задано, что по умолчанию означает нулевой уровень приоритета формируемых транзактов (заявок второго класса). Таким образом, заявки 1-го класса имеют более высокий уровень приоритета по сравнению с заявками 2-го класса, что предоставляет им преимущественное право на занятие того или иного объекта модели: прибора (в нашей модели) или многоканального устройства.

При необходимости предоставить более высокий приоритет заявкам 2-го класса достаточно во втором операторе **GENERATE** задать значение пятого операнда большее, чем 2.

Следует также обратить внимание на второй оператор **GENERATE**, где в поле операндов присутствует запятая, разделяющая операнды A и B, и точка, отделяющая дробную часть десятичного числа от целой части.

Необходимо помнить, что запятая в поле операндов используется только в качестве разделителя операндов, а точка – для отделения дробной части десятичного числа от целой части.

Оператор **ADVANCE** в обоих модулях GPSS-модели в качестве операнда A содержит выражение, реализующее формирование случайной задержки транзактов (заявок) в соответствии с распределением Эрланга 3-го порядка путём сложения трёх экспоненциально распределённых случайных величин. Для уменьшения корреляции между случайными значениями, вырабатываемыми при обращении к библиотечной процедуре **Exponential**, используются разные генераторы равномерно распределённых величин, номера которых указаны в качестве первого параметра библиотечной процедуры **Exponential**.

Следует обратить внимание на местоположение оператора **DEPART** во втором модуле GPSS-модели, который, в отличие от первого модуля, располагается после оператора **ADVANCE**. Таким способом можно в стандартном отчёте получить информацию не о времени ожидания заявок в очереди, как это сделано для заявок 1-го класса, а о времени пребывания заявок 2-го класса в системе (в очереди и на обслуживании). Это легко понять, если вспомнить, что функцией оператора **DEPART** является отметка времени и сбор статистики по времени между моментами прохождения транзакта через операторы QUEUE и DEPART. Описанная ситуация соответствует случаю, когда заявка, находясь на обслуживании в приборе, остаётся в накопителе, который она покидает только после завершения обслуживания. Примером такой ситуации может служить модель передачи пакетов по каналу связи в сети передачи данных. Пакет в процессе передачи по каналу связи остаётся в выходной буферной памяти передающего узла до тех пор, пока не будет получено подтверждение от принимающего узла о безошибочном приёме пакета.

Для одновременного сбора статистики по времени пребывания заявок в системе и по времени ожидания в очереди можно использовать две пары операторов QUEUE и DEPART, как это показано ниже для второго модуля рассматриваемой GPSS-модели.

```
                                        . . .
Gis_U   QTABLE      Q_U,10,10,20
Gis_W   QTABLE      Q_W,10,10,20
                                        . . .
* Модуль 2: моделирование процессов поступления и обслуживания заявок 2-го класса
        GENERATE    15,5.5; равномерный поток и приоритет, равный 0
        QUEUE       Q_U;  регистрация момента занесения в очередь Q_U
        QUEUE       Q_W;  регистрация момента занесения в очередь Q_W
        SEIZE       Uzel;  попытка занять один из приборов устройства Uzel
        DEPART      Q_W;  регистрация момента покидания очереди Q_W
        ADVANCE     (Exponential(4,0,2.5)+Exponential(5,0,2.5)+Exponential(6,0,2.5));
        DEPART      Q_U;  регистрация момента покидания очереди Q_U
        RELEASE     Uzel; освобождение прибора Uzel
        TERMINATE   1; удаление из модели обслуженной заявки 2-го класса
```

При этом для получения соответствующих гистограмм плотностей распределений достаточно в модели описать две таблицы с помощью команды QTABLE.

Здесь пара операторов **QUEUE Q_U** и **DEPART Q_U** используется для сбора статистики по времени пребывания транзактов (заявок), а пара операторов **QUEUE Q_W** и **DEPART Q_W** – для сбора статистики по времени ожидания заявок.

Команда START не включена в состав модели и для запуска процесса моделирования должна быть задана с использованием меню GPSS World.

Подводя итог, следует отметить, что для реализации относительных приоритетов в GPSS не предусмотрены специальные операторы. Единственным средством для отображения относительных приоритетов служит присвоение определённого уровня приоритета разным транзактам. Однако для реализации абсолютных приоритетов в GPSS используются специальные операторы, которые рассматриваются в следующей модели.

6.7.8. Модель 4.А: одноканальная СМО с абсолютными приоритетами

Положим, что в той же одноканальной СМО (рис.6.14), обслуживание неоднородного потока заявок осуществляется на основе *абсолютных* приоритетов, допускающих прерывание обслуживания низкоприоритетной заявки при поступлении в систему высокоприоритетной заявки.

Текст GPSS-модели с *комментариями* (*выделены курсивом*):

```
* Модуль 1: моделирование процессов поступления и обслуживания заявок 1-го класса
      GENERATE  30,,,,2; детерминированный поток и приоритет, равный 2
      QUEUE     QUzel_1;  поступление заявки в очередь QUzel_1
      PREEMPT   Uzel;  занятие прибора с вытеснением низкоприоритетной заявки
      DEPART    QUzel_1; покидание очереди QUzel_1
      ADVANCE   (Exponential(1,0,2.5)+Exponential(2,0,2.5)+Exponential(3,0,2.5));
      RETURN    Uzel;  освобождение прибора и возврат прерванной заявки
      TERMINATE 1; удаление из модели обслуженной заявки 1-го класса
 ****************************************
* Модуль 2: моделирование процессов поступления и обслуживания заявок 2-го класса
      GENERATE  15,5.5; равномерный поток и приоритет, равный 0
      QUEUE     QUzel_2;  поступление заявки в очередь QUzel_2
      SEIZE     Uzel;  попытка занять один из приборов устройства Uzel
      ADVANCE   (Exponential(4,0,2.5)+Exponential(5,0,2.5)+Exponential(6,0,2.5));
      DEPART    QUzel_2;  покидание очереди QUzel_2
      RELEASE   Uzel; освобождение прибора Uzel
      TERMINATE 1; удаление из модели обслуженной заявки 2-го класса
```

Краткое описание моделируемой СМО:

- количество обслуживающих приборов – 1;
- емкость накопителя – не ограничена;
- количество потоков (классов) заявок – 2;

- поток заявок 1-го класса – детерминированный с интервалами между заявками 30 минут;
- поток заявок 2-го класса – случайный с равномерно распределёнными интервалами между заявками от 9,5 до 20,5 минут;
- длительность обслуживания заявок 1-го и 2-го класса – случайная, распределённая по закону Эрланга 3-го порядка со средним значением 7,5 минут;
- дисциплина обслуживания – с **абсолютными** приоритетами.

Для моделирования абсолютных приоритетов в GPSS World используются специальные операторы: PREEMPT (ЗАХВАТИТЬ) и RETURN (ВЕРНУТЬ), позволяющие реализовать прерывание обслуживания в приборе низкоприоритетного транзакта с последующим его восстановлением, а также оператор PRIORITY, позволяющий изменять приоритет транзакта в модели.

Оператор PREEMPT реализует захват прибора с вытеснением находящегося в приборе транзакта с более низким приоритетом (прерывание обслуживания низкоприоритетной заявки).

Если при поступлении в этот оператор транзакта прибор с именем, указанным в операнде A, свободен, то оператор PREEMPT инициирует такие же действия, что и оператор SEIZE, то есть транзакт занимает прибор независимо от уровня его приоритета. Если же прибор занят, то сравниваются уровни приоритета поступившего транзакта и транзакта, находящегося в приборе. Если уровень приоритета поступившего транзакта выше, чем находящегося в приборе, то в *приоритетном режиме*, признаком которого служит операнд B, имеющий значение PR, выполняется вытеснение из прибора (прерывание обслуживания) низкоприоритетного транзакта, и поступивший высокоприоритетный транзакт захватывает прибор. Отсутствие операнда по умолчанию означает *режим прерывания*, при котором захват прибора высокоприоритетным транзактом возможен только в том случае, если обслуживаемый транзакт не является захватчиком. В этом случае поступивший транзакт помещается в список отложенных прерываний. Транзактам из списка отложенных прерываний предоставляется возможность занять устройство раньше, чем вытесненным транзактам или транзактам из списка задержки. Если уровень приоритета поступившего транзакта ниже, чем находящегося в приборе, то он помещается в список задержки прибора с учётом приоритета.

Таким образом, наличие двух режимов оператора PREEMPT предоставляет возможность смоделировать разные стратегии реализации абсолютных приоритетов.

Оператор RETURN реализует удаление из прибора обслуженного транзакта (освобождение прибора) и, при необходимости, возврат низкоприоритетного транзакта в прибор для продолжения обслуживания. При этом очередной транзакт на обслуживание выбирается из списков в следующем порядке: сначала из списка отложенных прерываний, затем из

списка прерываний и, наконец, из списка задержки.

Заметим, что во втором модуле занятие прибора и его освобождение реализуется с помощью операторов SEIZE и RELEASE, поскольку формируемые там транзакты имеют самый низкий приоритет и не могут вытеснять никакие другие транзакты.

Использование вместо операторов SEIZE и RELEASE операторов PREEMPT и RETURN, а также операндов C, D и E в операторе PREEMPT позволяет смоделировать более сложные стратегии реализации абсолютных приоритетов, при которых, например, вытесненные транзакты могут быть направлены в другие блоки модели или исключены из состязания за прибор.

Следует помнить, что приоритетный захват возможен только для прибора, но невозможен для многоканального устройства!

Как и в предыдущей модели, команда START должна быть задана с использованием меню GPSS World.

6.7.9. Модель 5: двухузловая разомкнутая СеМО с однородным потоком заявок

Положим, что линейная разомкнутая СеМО с однородным потоком заявок содержит два узла (рис.6.15).

В РСеМО из внешней среды, обозначенной на рисунке как «0», в узел 1 поступает простейший поток заявок со средним интервалом 100 секунд. После обслуживания в узле 1 заявки с вероятностью $p_{12} = 0{,}8$ переходят на обслуживание в узел 2 и с вероятностью $p_{10} = 0{,}2$ покидают СеМО, возвращаясь во внешнюю среду. Из условия линейности СеМО следует, что $p_{10} + p_{12} = 1$, поскольку заявки в сети не теряются и не размножаются.

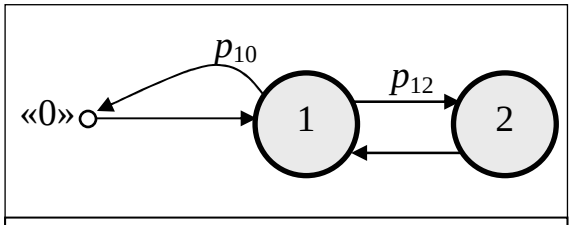


Рис.6.15. Разомкнутая СеМО

Длительность обслуживания заявок в узле 1, представляющем собой двухканальную СМО, – величина случайная, распределенная по равномерному закону в интервале от 10 до 20 секунд, то есть в интервале 15±5 с (со средним значением 15 с).

Длительность обслуживания заявок в узле 2, представляющем собой одноканальную СМО, – величина случайная, распределенная по экспоненциальному закону со средним значением 20 с.

Краткое описание рассматриваемой РСеМО:

- количество потоков (классов) заявок: $H = 1$;
- количество узлов в сети: $n = 2$;
- количество обслуживающих приборов в узле 1: $K_1 = 2$;
- количество обслуживающих приборов в узле 2: $K_2 = 1$;

- ёмкость накопителей в узлах сети – не ограничена, то есть в сети не может быть потерь заявок, что обусловливает линейность сети;
- поток заявок – простейший;
- средний интервал между поступающими в сеть заявками: $a = 100\,\text{с}$;
- длительность обслуживания заявок в узле 1 распределена равномерно в интервале от 10 до 20 с (15±5 с);
- длительность обслуживания заявок в узле 2 распределена по экспоненциальному закону со средним значением 20 с.

Рассмотрим **GPSS-модель** разомкнутой СеМО и прокомментируем некоторые операторы модели.

```
**************************************************************************
* Модель 5: разомкнутая однородная СеМО с двумя узлами
**************************************************************************
* Модуль 0:   область описания
Uz_1    STORAGE    2; задание числа приборов в узле 1
Tw_1    QTABLE     1,0,1,20; время ожидания в узле 1
Tw_2    QTABLE     2,50,50,20; время ожидания в узле 2
T_U     TABLE      M1,150,150,20; время пребывания в сети
**************************************************************************
* Модуль 1: моделирование процессов поступления и обслуживания заявок в узле 1
        GENERATE   (Exponential(10,0,100)); формирование простейшего потока
Met_1   QUEUE      1;  регистрация момента поступления заявки в очередь узла 1
        ENTER      Uz_1; попытка занять один из приборов узла 1
        DEPART     1;  регистрация момента покидания заявки очереди узла 1
        ADVANCE    15,5; задержка (обслуживание) заявки в узле 1
        LEAVE      Uz_1; выход обслуженной заявки из узла 1
        TRANSFER   .8,,Met_2; передача транзакта с вероятностью 0,8 в узел 2
        TABULATE   T_U
        TERMINATE  1; удаление из модели (СеМО) обслуженной заявки
**************************************************************************
* Модуль 2:  моделирование процесса обслуживания заявок в узле 2
Met_2   QUEUE      2;  регистрация момента поступления заявки в очередь узла 2
        SEIZE      2;  попытка занять прибор узла 2
        DEPART     2;  регистрация момента покидания заявки очереди узла 2
        ADVANCE    (Exponential(50,0,20));  обслуживание заявки в узле 2
        RELEASE    2;  освобождение прибора и выход заявки из узла 2
        TRANSFER   ,Met_1; безусловная передача транзакта в узел 1
**************************************************************************
        START      100000; запуск модели
```

Модуль 0 содержит описание двухканального устройства первого узла СеМО с именем Uz_1 и трех таблиц для формирования гистограмм плотностей распределений следующих случайных величин:

Tw_1 – времени ожидания заявок в узле 1 СеМО;

Tw_2 – времени ожидания заявок в узле 2 СеМО;

T_U – полного времени пребывания заявок в СеМО.

Отметим, что для каждого узла в модели определяется так называе-

мое ***единичное время ожидания*** заявок в узле СеМО, то есть время, соответствующее одному попаданию заявки в узел. В отличие от единичного, ***полное время ожидания*** заявки в узле СеМО учитывает, сколько раз заявка попала в данный узел за время ее нахождения в СеМО.

Как уже отмечалось, выбор длин и числа частотных интервалов, задаваемых в качестве операндов операторов QTABLE и TABLE, является непростой задачей, если мы хотим получить гистограмму приемлемого вида, дающего достаточно хорошее представление о законе (плотности) распределения. Обычно их значения подбираются экспериментальным путем в процессе нескольких реализаций имитационной модели. В частности, для рассматриваемой модели таким способом были подобраны операнды B=0, C=1 и D=20 в операторе:

Tw_1 QTABLE 1,0,1,20 .

Значение операнда B, определяющего длину первого частотного класса, было выбрано равным 0, поскольку в узле 1 почти 80% заявок имели нулевое время ожидания.

Модули 1 и 2 моделируют функционирование соответственно узлов 1 и 2 СеМО.

В модулях 1 и 2 используется новый оператор – оператор передачи **TRANSFER,** который реализует передачу активного транзакта к другому блоку (оператору) и может работать в 9 режимах. Активным является транзакт, поступивший в рассматриваемый момент времени в оператор **TRANSFER**.

В нашей модели оператор **TRANSFER** используется в двух режимах: *статистическом* и *безусловном*.

В общем случае оператор **TRANSFER** в *статистическом режиме* имеет вид:

TRANSFER A,[B],C ,

где **A** – частота (которая может трактоваться как вероятность) передачи активного транзакта к оператору с именем (меткой), указанной в качестве операнда **C**; может задаваться двумя способами: в виде *дробной величины* в интервале (0; 1) либо в виде неотрицательного *целого числа*, которое интерпретируется как доля от тысячи;

B – метка альтернативного оператора, которому будет передан активный транзакт (с вероятностью p=1-**A)**; если операнд не указан, то транзакт передается следующему по порядку оператору;

C – метка оператора, к которому передается активный транзакт с вероятностью, заданной в операнде **A**.

В нашем случае оператор **TRANSFER** в статистическом режиме имеет вид:

TRANSFER .8,,Met_2

Активный транзакт с частотой (вероятностью) 0,8 направляется к оператору с меткой **Met_2** и с вероятностью 0,2 – к следующему по порядку оператору.

Оператор **TRANSFER 800,,Met_2** эквивалентен предыдущему оператору и отличается только тем, что операнд **A** задан в виде целого числа, которое интерпретируется как доля от тысячи. Заметим, что оператор **TRANSFER 8,,Met_2** не эквивалентен двум предыдущим, поскольку вероятность перехода в данном случае будет равна 0,008.

Если в операторе **TRANSFER** операнд **A** отсутствует, то это означает, что оператор применяется в *безусловном режиме*: активный транзакт всякий раз будет направляться к оператору с меткой, указанной в качестве операнда **B.**

В нашем случае оператор **TRANSFER** в безусловном режиме имеет вид:

TRANSFER ,Met_1 .

Активный транзакт всякий раз направляется к оператору с меткой **Met_1**.

Команда **START**, включённая в состав модели и содержащая в качестве операнда **A** значение 100000, обеспечивает автоматический запуск процесса моделирования сразу же после завершения трансляции GPSS-модели. Процесс моделирования завершится после прохождения через систему 100 тысяч заявок (транзактов).

На рис. 6.16 представлен фрагмент отчёта, из которого могут быть получены основные характеристики функционирования РСеМО (наиболее интересные и важные результаты выделены жирным шрифтом).

Время моделирования (**END TIME**), в течение которого через РСеМО прошло $N_0 = 100000$ заявок, равно 9 994 872.283 секунд, что составляет без малого почти 2 777 часов или более 115 суток работы моделируемой системы. Все 100 тысяч обслуженных заявок, в конечном счете, попали в блок TERMINATE и были удалены из модели. В то же время количество сгенерированных транзактов в блоке GENERATE составляет 100 016. Возникает вопрос: почему было сгенерировано транзактов больше 100 000 и где «лишние» 16 транзактов? Ответ достаточно прост: поскольку процесс моделирования был остановлен по числу покинувших сеть ста тысяч транзактов, то на момент завершения моделирования в РСеМО остались транзакты, которые поступили в сеть, но не успели полностью пройти все этапы обслуживания. Где же эти транзакты? Из представленного отчёта видно, что на момент завершения моделирования в 10-м блоке QUEUE находится 15 транзактов и в 13-м блоке ADVANCE – один транзакт, всего 16 «лишних» транзактов на момент завершения процесса моделирования оказались в узле 2 СеМО: один в приборе и 15 – в очереди.

Через первый узел (блок **ENTER**) транзакты прошли $N_1 = 501310$ раз, через второй (блок **SEIZE**) – $N_2 = 401295$ раз. Эти значения позволяют рассчитать коэффициенты передач для обоих узлов РСеМО:

$$\alpha_1 = \frac{N_1}{N_0} = \frac{501310}{100000} \cong 5 \qquad \text{и} \quad \alpha_2 = \frac{N_2}{N_0} = \frac{401295}{100000} \cong 4,$$

что соответствует теоретическим значениям, которые могут быть найдены путём решения системы уравнений (4.16), как это описано в п.4.4.2.

Загрузки узлов (**UTIL.**) соответственно равны: $\rho_1 = 0{,}376$ и $\rho_2 = 0{,}800$.

```
START TIME              END TIME  BLOCKS  FACILITIES  STORAGES
             0.000       9994872.283    15        1          1

LABEL            LOC  BLOCK TYPE     ENTRY COUNT CURRENT COUNT RETRY
                   1    GENERATE         100016             0       0
MET_1              2    QUEUE            501310             0       0
                   3    ENTER            501310             0       0
                   4    DEPART           501310             0       0
                   5    ADVANCE          501310             0       0
                   6    LEAVE            501310             0       0
                   7    TRANSFER         501310             0       0
                   8    TABULATE         100000             0       0
                   9    TERMINATE        100000             0       0
MET_2             10    QUEUE            401310            15       0
                  11    SEIZE            401295             0       0
                  12    DEPART           401295             0       0
                  13    ADVANCE          401295             1       0
                  14    RELEASE          401294             0       0
                  15    TRANSFER         401294             0       0

FACILITY    ENTRIES    UTIL.   AVE. TIME AVAIL. OWNER PEND INTER RETRY
DELAY
 2           401295    0.800      19.936   1    99992    0    0     0
15

QUEUE          MAX CONT. ENTRY ENTRY(0) AVE.CONT. AVE.TIME   AVE.(-0) RETRY
 1               8    0 501310 398681     0.073      1.457      7.115   0
 2              35   15 401310  81329     3.243     80.776    101.307   0

STORAGE       CAP. REM. MIN. MAX.  ENTRIES AVL.  AVE.C. UTIL. RETRY DELAY
 UZ_1           2    2   0     2   501310   1    0.752  0.376    0    0

TABLE         MEAN     STD.DEV.        RANGE            RETRY FREQUENCY CUM.%
 TW_1         1.457    3.795                              0
                                  _  -        0.000          398681    79.53
                              0.000  -        1.000            8939    81.31

                                     ...
                             18.000  -  _                      4203   100.00
 TW_2        80.772   98.691                                  0
                                  _  -       50.000          207020    51.59
                             50.000  -      100.000           75108    70.30

                                     ...
                            900.000  -      950.000               5   100.00
 T_U        486.446  736.978                                  0
                                  _  -      150.000           38624    38.62
                            150.000  -      300.000           17615    56.24

                                        ...
                           2850.000  -  _                      1819   100.00


  FEC XN   PRI         BDT      ASSEM  CURRENT  NEXT  PARAMETER    VALUE
  99992    0     9994893.688    99992     13     14
 100017    0     9994946.528   100017      0      1
```

Рис.6.16. Фрагмент отчета модели разомкнутой СеМО

Средние длины очередей (**AVE.CONT.**) в узлах СеМО составляют: $l_1 = 0{,}073$ и $l_2 = 3{,}243$.

Использование в модели таблиц для построения гистограмм плотностей распределений времён ожидания заявок в узлах СеМО и времени пребывания заявок в сети, кроме средних значений временных характеристик, позволяет получить их среднеквадратические отклонения:

$w_1 = 1{,}457$ с; $\sigma_{w_1} = 3{,}795$ с;

$w_2 = 80{,}772$ с; $\sigma_{w_2} = 98{,}691$ с;

$U = 486{,}446$ с; $\sigma_U = 736{,}978$ с.

6.7.10. Модель 6: многоузловая разомкнутая СеМО с однородным потоком заявок

В данном пункте рассматривается модель, в которой заявки, покидающие некоторый узел, должны перемещаться с заданными вероятностями по 3-м и более направлениям. Реализация соответствующей имитациионной модели связана с определёнными проблемами, обусловленными особенностями имитационного моделирования.

Положим, что линейная разомкнутая СеМО с однородным потоком заявок содержит три узла (рис.6.17).

Пусть, как и в предыдущей модели, в РСеМО из внешней среды «0» в узел 1 поступает простейший поток заявок со средним интервалом 100 секунд. После обслуживания в узле 1 заявки с вероятностью $p_{12} = 0{,}1$ переходят на обслуживание в узел 2, с вероятностью $p_{13} = 0{,}3$ – на обслуживание в узел 3 и с вероятностью $p_{10} = 0{,}6$ покидают сеть, причём $p_{10} + p_{12} + p_{13} = 1$.

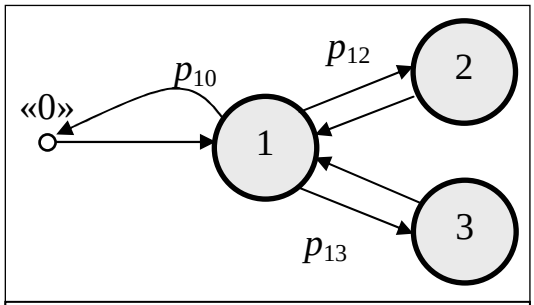


Рис.6.17. Трёхузловая РСеМО

Положим, что все узлы СеМО – одноканальные, а длительности обслуживания заявок в узлах 1, 2 и 3 представляют собой детерминированные величины, равные 10, 20 и 30 секундам соответственно.

Краткое описание рассматриваемой СеМО:

- количество потоков (классов) заявок: $H = 1$;
- количество узлов в сети: $n = 3$;
- все узлы одноканальные: $K_1 = K_2 = K_3 = 1$;
- емкость накопителей в узлах сети – не ограничена;
- поток заявок – простейший;
- средний интервал между поступающими в сеть заявками: $a = 100$ с;
- длительности обслуживания заявок в узлах 1, 2 и 3 детерминированные величины, равные соответственно 10 с, 20 с и 30 с.

Текст GPSS-модели с *комментариями* (*выделены курсивом*):

```
**************************************************************************
* Модель 6 линейной разомкнутой однородной СеМО с тремя узлами
**************************************************************************
*Модуль 1: моделирование процессов поступления и обслуживания заявок в узле 1
        GENERATE    (Exponential(10,0,100)); формирование простейшего потока
Met_1   SEIZE       1;  попытка занять прибор узла 1
        ADVANCE     10;  обслуживание заявки в узле 1
        RELEASE     1;  выход обслуженной заявки из узла 1
        TRANSFER    .6,,Met_0;  передача заявки с вероятностью 0,6 в узел «0»
        TRANSFER    .75,,Met_3;  передача транзакта с вероятностью 0,75 в узел 3
**************************************************************************
* Модуль 2: моделирование процесса обслуживания заявок в узле 2
Met_2   SEIZE       2;  попытка занять прибор узла 2
        ADVANCE     20;  обслуживание заявки в узле 2
        RELEASE     2;  освобождение прибора и выход заявки из узла 2
        TRANSFER    ,Met_1;  безусловная передача транзакта в узел 1
**************************************************************************
* Модуль 3:  моделирование процесса обслуживания заявок в узле 3
Met_3   SEIZE       3;  попытка занять прибор узла 3
        ADVANCE     30;  обслуживание заявки в узле 3
        RELEASE     3;  освобождение прибора и выход заявки из узла 3
        TRANSFER    ,Met_1; безусловная передача транзакта в узел 1
Met_0   TERMINATE   1; удаление из модели (СеМО) обслуженной заявки
```

Рассмотрим представленную GPSS-модель, уделив основное внимание операторам TRANSFER.

В модуле 1, реализующем процесс обслуживания заявки в узле 1 СеМО, используются два оператора TRANSFER. Первый оператор:

TRANSFER .6,,Met_0

направляет активный транзакт с заданной вероятностью 0,6 к оператору TERMINATE, что соответствует удалению обслуженной заявки из СеМО, и с вероятностью 0,4 – к следующему по порядку оператору:

TRANSFER .75,,Met_3 .

Назначение этого оператора – реализация процесса выбора узла, в который будет направлена заявка после завершения обслуживания в узле 1. Активный транзакт с вероятностью 0,75 будет направлен в модуль 3, реализующий процесс обслуживания заявок в узле 3, и соответственно с вероятностью 0,25 – в модуль 2, реализующий процесс обслуживания заявок в узле 2. Естественно, возникает вопрос: почему вероятности равны 0,25 и 0,75, если вероятности передачи в узлы 2 и 3 СеМО заданы равными 0,1 и 0,3? Ответ становится очевидным, если иметь в виду следующее: транзакт, попадающий в статистический TRANSFER, с заданной в операнде A вероятностью p направляется к оператору с меткой, указанной в операнде C, и с вероятностью $(1-p)$, являющейся дополнением до 1, к оператору с меткой, указанной в операнде B, или, по умолчанию, к следующему по порядку оператору, если операнд B опущен. Таким

образом, если в рассматриваемом операторе TRANSFER в качестве операнда A указать вероятность 0,3, то активный транзакт с вероятностью 0,3 будет направлен в модуль 3, и с вероятностью $(1-0{,}3)=0{,}7$ – в модуль 2. Это приведёт к тому, что большая часть заявок после обслуживания в узле 1 будет направляться в узел 2, в то время как в соответствии с исходными данными должно быть наоборот – в узел 3 должно направляться в 3 раза больше заявок, чем в узел 2.

Итак, каким же образом должны рассчитываться новые значения вероятностей? Необходимо, чтобы соотношение между вероятностями сохранялось. Этого можно достичь путём *нормирования вероятностей* так, чтобы сумма вероятностей была равна 1.

Например, пусть имеется N направлений передачи транзактов с вероятностями $p_1, p_2, \ldots, p_N$, причём $p_1 + p_2 + \ldots + p_N = 1$. Указанное вероятностное разветвление потока заявок может быть реализовано в GPSS-модели с помощью последовательности из $(N-1)$ операторов TRANSFER:

TRANSFER p_1,,Napravlenie_1

TRANSFER p_2^*,,Napravlenie_2

TRANSFER p_3^*,,Napravlenie_3

. . .

TRANSFER p_{N-1}^*,,Napravlenie_N-1

В первом операторе указывается заданное значение вероятности p_1 для передачи транзакта к оператору с меткой Napravlenie_1.

Во втором операторе для передачи транзакта к оператору с меткой Napravlenie_2 указывается нормированное значение вероятности, рассчитываемое как $p_2^* = \dfrac{p_2}{p_2 + p_3 + \ldots p_N}$ или, что то же самое, как $p_2^* = \dfrac{p_2}{1-p_1}$.

Аналогично, значение вероятности для третьего оператора:

$$p_3^* = \frac{p_3}{p_3 + p_4 + \ldots p_N} = \frac{p_3}{1-p_1-p_2}$$ и т.д.

И, наконец, для последнего оператора:

$$p_{N-1}^* = \frac{p_{N-1}}{p_{N-1} + p_N} = \frac{p_{N-1}}{1-p_1-\ldots-p_{N-2}}.$$

6.7.11. Модель 7: замкнутая СеМО с однородным потоком заявок

Положим, что рассмотренная выше линейная разомкнутая СеМО с однородным потоком заявок и двумя узлами преобразована в замкнутую

СеМО (рис.6.18), в которой циркулирует постоянное число заявок: $M = 5$.

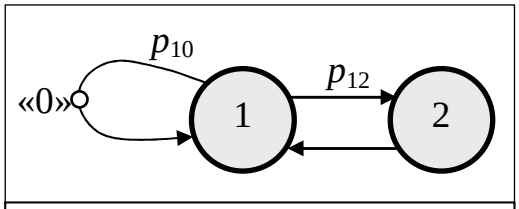


Рис.6.18. Замкнутая СеМО

Как и в предыдущей модели, после обслуживания в узле 1 заявки с вероятностью $p_{12} = 0{,}8$ переходят на обслуживание в узел 2 и с вероятностью $p_{10} = 0{,}2$ возвращаются в узел 1, причем $p_{10} + p_{12} = 1$. Пусть нулевая точка выбрана на дуге, выходящей из узла 1 и входящей снова в узел 1 так, как это показано на рис.6.18. Относительно этой точки будут измеряться такие характеристики сети, как производительность ЗСеМО и время пребывания заявок в сети. Длительность обслуживания заявок в двухканальном узле 1 распределена по равномерному закону в интервале от 10 до 20 секунд, а длительность обслуживания заявок в одноканальном узле 2 распределена по экспоненциальному закону со средним значением 20 с.

Таким образом, краткое описание рассматриваемой замкнутой СеМО имеет следующий вид:

- количество потоков (классов) заявок: $H = 1$;
- количество узлов в сети: $n = 2$;
- количество заявок, циркулирующих в замкнутой сети: $M = 5$;
- количество обслуживающих приборов в узле 1: $K_1 = 2$;
- длительность обслуживания заявок в узле 1 распределена равномерно в интервале от 10 до 20 с (15±5 с);
- количество обслуживающих приборов в узле 2: $K_2 = 1$;
- длительность обслуживания заявок в узле 2 распределена по экспоненциальному закону со средним значением 20 с.
- ёмкость накопителей в узлах сети достаточна для того, чтобы в сети не было потерь заявок, что обусловливает линейность сети; в нашем случае можно считать, что ёмкость каждого накопителя совпадает с числом циркулирующих в сети заявок.

Для того чтобы упростить процесс разработки GPSS-модели замкнутой СеМО, воспользуемся представленной выше GPSS-моделью разомкнутой СеМО. Основное отличие замкнутой СеМО от разомкнутой состоит в отсутствии внешнего источника заявок, при этом в GPSS-модели замкнутой СеМО необходимо реализовать циркуляцию в сети постоянного числа заявок (в нашей модели – пяти заявок).

Рассмотрим представленную ниже **GPSS-модель** и прокомментируем только выделенные жирным шрифтом изменения (операторы), которые были внесены в модель разомкнутой СеМО для ее преобразования в модель замкнутой СеМО.

1) Модель содержит на один модуль больше, чем модель разомкнутой СеМО. Дополнительный третий модуль реализует завершение процесса моделирования путем задания временного интервала функционирования реальной (исследуемой) системы.

```
*************************************************************************
* Модель 7 линейной замкнутой однородной СеМО с двумя узлами и М=5
*************************************************************************
* Модуль 0:   область описания
Uz_1    STORAGE     2; задание числа приборов в узле 1
Tw_1    QTABLE      1,0,0.5,30; время ожидания в узле 1
Tw_2    QTABLE      2,10,10,30; время ожидания в узле 2
T_U     TABLE       M1,40,40,30; время пребывания в сети
****************************************
* Модуль 1:  моделирование процесса обслуживания заявок в узле 1
        GENERATE    ,,,5;  формирование в нулевой момент времени пяти заявок
Met_1   MARK            ; отметка момента времени поступления заявки в сеть
Met_3   QUEUE       1; регистрация момента поступления заявки в очередь узла 1
        ENTER       Uz_1; попытка занять один из приборов узла 1
        DEPART      1;  регистрация момента покидания заявки очереди узла 1
        ADVANCE     15,5; задержка (обслуживание) заявки в узле 1
        LEAVE       Uz_1; выход обслуженной заявки из узла 1
        TRANSFER    .8,,Met_2; передача транзакта с вероятностью 0,8 в узел 2
        TABULATE    T_U
        TRANSFER    ,Met_1; безусловная передача транзакта в узел 1
****************************************
* Модуль 2:  моделирование процесса обслуживания заявок в узле 2
Met_2   QUEUE       2; регистрация момента поступления заявки в очередь узла 2
        SEIZE       2;  попытка занять прибор узла 2
        DEPART      2;  регистрация момента покидания заявки очереди узла 2
        ADVANCE     (Exponential(50,0,20));  обслуживание заявки в узле 2
        RELEASE     2;  освобождение прибора и выход заявки из узла 2
        TRANSFER    ,Met_3; безусловная передача транзакта в узел 1
*************************************************************************
* Модуль 3: завершение процесса моделирования по длительности моделирования
        GENERATE    10000000; задание единичной длительности моделирования
        TERMINATE   1; уменьшение счетчика завершения на 1
*************************************************************************
```

2) Для получения более наглядных временных гистограмм в операторах QTABLE и TABLE модуля 0 изменены параметры, задающие длину и число частотных интервалов, значения которых были подобраны экспериментальным путем.
3) В модуль 1 внесены 3 изменения.
   - Во-первых, изменился оператор **GENERATE**, который принял вид:

     GENERATE ,,,5

     В этом операторе указан только четвертый операнд, определяющий число генерируемых им транзактов за все время моделирования. Поскольку первый параметр отсутствует, то в нулевой момент модельного времени будут сформированы 5 транзактов, которые поступят в очередь первого узла. Таким образом, в моделируемой СеМО появятся 5 заявок.
   - Во-вторых, появился новый оператор с меткой:

Met_1 MARK .

Оператор **MARK (ОТМЕТИТЬ)** предназначен для записи значения абсолютного времени в качестве одного из параметров транзакта (отметка транзакта) и, в общем случае, имеет вид:

MARK A .

Единственный операнд **A** задает *имя* или *номер параметра* активного транзакта, в который записывается значение таймера абсолютного времени, причём при его отсутствии значение абсолютного времени по умолчанию помещается на место ранее записанного времени входа транзакта в модель.

Этот оператор в рассматриваемой GPSS-модели используется для отметки момента прохождения заявкой «нулевой точки», относительно которой измеряется время пребывания заявок в замкнутой СеМО.

- В-третьих, в конце модуля 1 вместо оператора TERMINATE вставлен оператор

TRANSFER ,Met_1 ,

реализующий безусловную передачу транзакта к блоку с меткой **Met_1**, что соответствует в модели возврату заявки в узел 1.

4) Дополнительный модуль 3 состоит только из двух операторов:

```
GENERATE    10000000
TERMINATE   1
```

Такой модуль применяется в GPSS-моделях для реализации завершения процесса моделирования *по времени*, прошедшему в моделируемой системе, а не по числу обслуженных в системе заявок (прошедших через модель транзактов).

В соответствии с этими операторами в момент модельного времени (времени, который наступит в реальной исследуемой системе), равный 10000000, в блоке **GENERATE** модели появится транзакт, который сразу же попадет в блок **TERMINATE** и будет уничтожен. При этом значение счётчика завершений будет уменьшено на единицу. Если модель была запущена командой **START 1**, установившей начальное значение счётчика завершений в 1, то после вычитания 1 значение счётчика завершений станет равным 0 и процесс моделирования завершится. Таким образом, если в предыдущей модели завершение моделирования осуществлялось по числу заявок, покинувших СеМО, то в данной модели использовалось другое условие завершения моделирования – по времени, прошедшему в моделируемой системе.

На рис.6.19 представлен фрагмент отчёта, из которого могут быть получены все основные характеристики функционирования замкнутой СеМО (наиболее интересные и важные результаты моделирования выделены жирным шрифтом).

```
            GPSS World Simulation Report - Untitled Model 1.1.1

         START TIME       END TIME    BLOCKS  FACILITIES  STORAGES
            0.000      10000000.000     18        1          1

                                  ...
LABEL          LOC   BLOCK TYPE     ENTRY COUNT   CURRENT COUNT  RETRY
                 1    GENERATE            5             0        0
MET_1            2    MARK           124309             0        0
MET_3            3    QUEUE          622086             0        0
                 4    ENTER          622086             0        0
                 5    DEPART         622086             0        0
                 6    ADVANCE        622086             1        0
                 7    LEAVE          622085             0        0
                 8    TRANSFER       622085             0        0
                 9    TABULATE       124304             0        0
                10    TRANSFER       124304             0        0
MET_2           11    QUEUE          497781             3        0
                12    SEIZE          497778             0        0
                13    DEPART         497778             0        0
                14    ADVANCE        497778             1        0
                15    RELEASE        497777             0        0
                16    TRANSFER       497777             0        0
                17    GENERATE            1             0        0
                18    TERMINATE           1             0        0

FACILITY  ENTRIES   UTIL.   AVE.TIME  AVAIL. OWNER PEND INTER RETRY DELAY
 2         497778  0.993     19.949     1      4    0     0     0    3

QUEUE     MAX CONT. ENTRY  ENTRY(0) AVE.CONT. AVE.TIME   AVE.(-0) RETRY
 1         4    0  622086  444845     0.145      2.330      8.178   0
 2         4    3  497781   12528     2.929     58.836     60.355   0

STORAGE   CAP. REM. MIN. MAX.  ENTRIES AVL.  AVE.C. UTIL. RETRY DELAY
 UZ_1       2    1   0     2   622086   1    0.933  0.467    0    0

TABLE     MEAN    STD.DEV.       RANGE        RETRY FREQUENCY CUM.%
TW_1      2.330    4.796                        0
                              _  -     0.000        444845    71.51
                          0.000  -     0.500          6254    72.51
                          0.500  -     1.000          6350    73.53
                                       . . .
                         14.000  -  _                28460   100.00
TW_2     58.836   39.993                            0
                              _  -    10.000         38373     7.71
                         10.000  -    20.000         39655    15.68
                         20.000  -    30.000         49849    25.69
                                       . . .
                        290.000  -  _                   71   100.00
T_U     402.214  438.431                            0
                              _  -    40.000         24771    19.93
                         40.000  -    80.000          4846    23.83
                         80.000  -   120.000          8414    30.60
                                       . . .
                       1160.000  -  _                8158   100.00
```

Рис.6.19. Фрагмент отчета к модели замкнутой СеМО

Видно, что время завершения моделирования (**END TIME**) в точности совпадает с временем, заданным в модуле 3 GPSS-модели.

Оператор **GENERATE** сгенерировал за время моделирования только

5 транзактов, которые постоянно циркулировали в модели. При этом через первый узел (блок **ENTER**) транзакты прошли $N_1 = 622086$ раз, через второй (блок **SEIZE**) – $N_2 = 497778$ раз, а через нулевую точку (блок **TABULATE**) – $N_0 = 124304$ раз. Последнее значение позволяет рассчитать одну из основных сетевых характеристик замкнутой СеМО – производительность сети, как отношение числа заявок (транзактов), прошедших через нулевую точку СеМО за время моделирования $T = 10000000$, к этому времени:

$$\lambda_0 = \frac{N_0}{T} = \frac{124304}{10000000} = 0{,}0124304\ \text{с}^{-1} \approx 44{,}75\ \text{ч}^{-1},$$

то есть примерно 45 заявок в час.

Коэффициенты передач для каждого из узлов могут быть рассчитаны следующим образом:

$$\alpha_1 = \frac{N_1}{N_0} = \frac{622086}{124304} \cong 5 \quad \text{и} \quad \alpha_2 = \frac{N_2}{N_0} = \frac{497778}{124304} \cong 4.$$

Загрузки узлов (**UTIL.**) соответственно равны: $\rho_1 = 0{,}467$ и $\rho_2 = 0{,}993$.

Средние длины очередей (**AVE.CONT.**) в узлах СеМО составляют: $l_1 = 0{,}144$ и $l_2 = 2{,}929$.

Использование в модели таблиц для построения гистограмм плотностей распределений времён ожидания заявок в узлах СеМО и времени пребывания заявок в сети, кроме средних значений временных характеристик, позволяет получить их среднеквадратические отклонения:

$w_1 = 2{,}33$ с; $\sigma_{w_1} = 4{,}796$ с;

$w_2 = 58{,}836$ с; $\sigma_{w_2} = 39{,}993$ с;

$U = 402{,}214$ с; $\sigma_U = 438{,}431$ с.

6.7.12. Модель 8: разомкнутая СеМО с неоднородным потоком заявок

Положим, что в линейную разомкнутую СеМО с двумя узлами поступает неоднородный поток заявок двух классов (рис.6.20). Заявки класса 1 (сплошная линия) и класса 2 (пунктирная линия) поступают в узел 1 и образуют простейшие потоки со средними интервалами 100 и 50 секунд соответственно. После обслуживания в узле 1 заявки класса 1 с вероятностью $p_{12} = 0{,}8$ переходят на обслуживание в узел 2 и с вероятностью $p_{10} = 0{,}2$ покидают СеМО. Заявки класса 2 обслуживаются только в узле 1, после чего покидают СеМО.

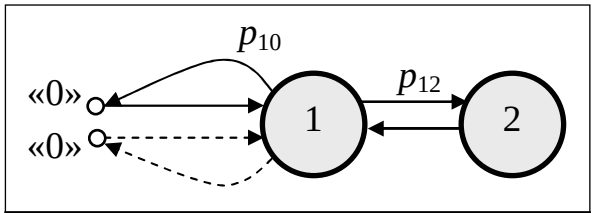


Рис.6.20. Разомкнутая СеМО с неоднородным потоком заявок

Длительности обслуживания заявок класса 1 и 2 в двухканальном узле 1 представляют собой равномерно распределённые случайные величины в интервалах (15±5) и (10 ±5) секунд соответственно.

Длительность обслуживания заявок класса 2 в одноканальном узле 2 – величина случайная, распределенная по экспоненциальному закону со средним значением 20 секунд.

Краткое описание рассматриваемой СеМО:

- количество потоков (классов) заявок: $H = 2$;
- количество узлов в сети: $n = 2$;
- количество обслуживающих приборов в узле 1: $K_1 = 2$;
- количество обслуживающих приборов в узле 2: $K_2 = 1$;
- емкость накопителей в узлах сети – не ограничена, то есть в сети не может быть потерь заявок, что обусловливает линейность сети;
- потоки заявок класса 1 и класса 2 – простейшие;
- средний интервал между поступающими заявками класса 1: $a_0(1) = 100\,\text{с}$;
- средний интервал между поступающими заявками класса 2: $a_0(2) = 50\,\text{с}$;
- длительность обслуживания заявок класса 1 в узле 1 распределена равномерно в интервале от 10 до 20 с: $b_1(1) = 15 \pm 5\,\text{с}$;
- длительность обслуживания заявок класса 2 в узле 1 распределена равномерно в интервале от 5 до 15 с: $b_1(2) = 10 \pm 5\,\text{с}$;
- длительность обслуживания заявок класса 1 в узле 2 распределена по экспоненциальному закону со средним значением 20 с: $b_2(1) = 20\,\text{с}$.

Текст GPSS-модели разомкнутой СеМО с неоднородным потоком заявок представлен на следующей странице.

В рассматриваемой GPSS-модели, в отличие от модели 5 двухузловой РСеМО с однородным потоком заявок, появился третий модуль, моделирующий процессы поступления и обслуживания заявок класса 2 в узле 1.

Таким образом, при моделировании СеМО с неоднородным потоком заявок количество исполняемых модулей GPSS-модели определяется как произведение количества классов заявок на количество узлов моделируемой СеМО.

На рис 6.21 представлен отчет с результатами имитационного моделирования разомкнутой СеМО с двумя классами заявок для значения 1000000 операнда А в команде START, заданного при запуске процесса моделирования.

Анализ представленного отчета позволяет получить основные характеристики функционирования разомкнутой СеМО с неоднородным потоком заявок (наиболее интересные и важные результаты моделирования выделены жирным шрифтом).

```
*************************************************************************
* Модель 8 разомкнутой СеМО с неоднородным потоком заявок
*************************************************************************
* Модуль 0:          область описания
Uzel_1      STORAGE    2; задание числа приборов в узле 1
****************************************
* Модуль 1:  поступление и обслуживание заявок класса 1 в узле 1
            GENERATE     (Exponential(10,0,100)); формирование потока заявок класса 1
Met_1       QUEUE        Quz1_k1;  момент поступления в очередь узла 1
            ENTER        Uzel_1; попытка занять один из приборов узла 1
            DEPART       QUz1_k1;  момент покидания очереди узла 1
            ADVANCE      15,5; задержка (обслуживание) в узле 1
            LEAVE        Uzel_1; выход обслуженной заявки из узла 1
            TRANSFER     .8,,Met_2; передача заявки с вероятностью 0,8 в узел 2
            TERMINATE    1; удаление из модели (СеМО) обслуженной заявки класса 1
****************************************
* Модуль 2:  моделирование процесса обслуживания заявок класса 1 в узле 2
Met_2       QUEUE        QUz2_k1;  момент поступления в очередь узла 2
            SEIZE        Uzel_2;  попытка занять прибор узла 2
            DEPART       QUz2_k1;  момент покидания очереди узла 2
            ADVANCE      (Exponential(50,0,20));  обслуживание в узле 2
            RELEASE      Uzel_2;  выход обслуженной заявки из узла 2
            TRANSFER     ,Met_1; безусловная передача транзакта в узел 1
****************************************
* Модуль 3:  поступление и обслуживание заявок класса 2 в узле 1
            GENERATE     (Exponential(10,0,50)); формирование потока заявок класса 2
            QUEUE        QUz1_k2;  момент поступления в очередь узла 1
            ENTER        Uzel_1; попытка занять один из приборов узла 1
            DEPART       QUz1_k2;  момент покидания очереди узла 1
            ADVANCE      10,5; задержка (обслуживание) в узле 1
            LEAVE        Uzel_1; выход обслуженной заявки из узла 1
            TERMINATE    1; удаление из модели (СеМО) обслуженной заявки класса 2
```

В процессе моделирования через разомкнутую СеМО прошло $N_0 = 1000000$ заявок обоих классов. Все обслуженные заявки попали в два блока TERMINATE и были удалены из модели.

По количеству транзактов каждого класса, прошедших через соответствующие блоки ENTER, SEIZE и TERMINATE можно рассчитать коэффициенты передач для заявок класса 1 ($\alpha_1(1), \alpha_2(1)$) и 2 ($\alpha_1(2), \alpha_2(2)$) в узлах 1 и 2 разомкнутой СеМО соответственно:

$$\alpha_1(1) = \frac{1671938}{333403} \cong 5 \quad \text{и} \quad \alpha_2(1) = \frac{1338533}{333403} \cong 4,$$

$$\alpha_1(2) = \frac{666597}{666597} = 1 \quad \text{и} \quad \alpha_2(2) = 0,$$

что соответствует теоретическим значениям, которые могут быть рассчитаны путём решения системы линейных алгебраических уравнений (4.16), как это описано в п.4.4.2.

Загрузки узлов СеМО (**UTIL.**) равны: $\rho_1 = 0{,}476$ и $\rho_2 = 0{,}802$.

```
           START TIME          END TIME  BLOCKS  FACILITIES  STORAGES
                0.000      33345310.868    21        1          1

            NAME                          VALUE
         MET_1                             2.000
         MET_2                             9.000
         QUZ1_1                        10002.000
         QUZ1_2                        10001.000
         QUZ2                          10003.000
         UZEL_1                        10000.000
         UZEL_2                        10004.000

LABEL              LOC  BLOCK TYPE     ENTRY COUNT CURRENT COUNT RETRY
                    1    GENERATE        333406             0       0
MET_1               2    QUEUE          1671938             0       0
                    3    ENTER          1671938             1       0
                    4    DEPART         1671937             0       0
                    5    ADVANCE        1671937             1       0
                    6    LEAVE          1671936             0       0
                    7    TRANSFER       1671936             0       0
                    8    TERMINATE       333403             0       0
MET_2               9    QUEUE          1338533             0       0
                   10    SEIZE          1338533             0       0
                   11    DEPART         1338533             0       0
                   12    ADVANCE        1338533             1       0
                   13    RELEASE        1338532             0       0
                   14    TRANSFER       1338532             0       0
                   15    GENERATE        666597             0       0
                   16    QUEUE           666597             0       0
                   17    ENTER           666597             0       0
                   18    DEPART          666597             0       0
                   19    ADVANCE         666597             0       0
                   20    LEAVE           666597             0       0
                   21    TERMINATE       666597             0       0

FACILITY  ENTRIES  UTIL.   AVE. TIME AVAIL. OWNER PEND INTER RETRY DELAY
UZEL_2    1338533  0.802      19.984  1    999991   0    0     0     0

QUEUE     MAX CONT.  ENTRY ENTRY(0) AVE.CONT. AVE.TIME   AVE.(-0) RETRY
 QUZ1_2     6   0   666597  465509     0.047     2.332      7.732   0
 QUZ1_1    10   1  1671938 1158921     0.120     2.396      7.809   0
 QUZ2      40   0  1338533  272711     3.232    80.521    101.124   0

STORAGE   CAP. REM. MIN. MAX.  ENTRIES  AVL.  AVE.C. UTIL. RETRY DELAY
 UZEL_1     2    0   0    2    2338535   1    0.952  0.476    0    0

CEC XN  PRI         M1      ASSEM  CURRENT  NEXT  PARAMETER    VALUE
999980    0    33344706.455   999980      3      4

FEC XN   PRI         BDT      ASSEM  CURRENT  NEXT  PARAMETER    VALUE
999993    0    33345317.746   999993      5      6
999991    0    33345318.791   999991     12     13
1000005   0    33345330.450   1000005     0     15
1000003   0    33345459.221   1000003     0      1
```

Рис.6.21. Отчет к модели разомкнутой СеМО с неоднородным

Среднее число заявок класса 1 в очереди (**AVE.CONT.**) в узлах 1 и 2 СеМО: $l_1(1) = 0{,}120$ и $l_2(1) = l_2 = 3{,}232$. Среднее число заявок класса 2 в

очереди узла 1 СеМО: $l_1(2) = 0{,}047$. Заметим, что в узле 2 очередь образуют только заявки первого класса. Суммарная длина очереди заявок в узле 1 равна $l_1 = l_1(1) + l_1(2) = 0{,}167$. Суммарное число заявок, находящихся в состоянии ожидания в СеМО: $L = l_1 + l_2 \cong 3{,}4$.

Средние времена ожидания (**AVE.TIME**) заявок класса 1 в узлах 1 и 2 СеМО соответственно равны: $w_1(1) = 2{,}4\,\text{с}$ и $w_2(1) = 80{,}5\,\text{с}$. Среднее время ожидания заявок класса 2 в узле 1 СеМО: $w_1(2) = 2{,}33\,\text{с}$.

Следует заметить, что стандартный GPSS-отчёт по результатам моделирования содержит информацию не по всем характеристикам, которые могут представлять интерес для исследователя. В частности, представленный выше отчёт не содержит информацию о временах пребывания заявок в узлах и в СеМО в целом.

Эти характеристики могут быть рассчитаны на основе имеющихся в отчёте данных с использованием фундаментальных соотношений, представленных в п.3.4.3. Так, например, легко могут быть рассчитаны средние времена пребывания заявок каждого класса в узлах СеМО:

$u_1(1) = w_1(1) + b_1(1) = 17{,}4\,\text{с}$; $\quad u_2(1) = w_2(1) + b_2(1) = 100{,}5\,\text{с}$;

$u_1(2) = w_1(2) + b_1(2) = 12{,}3\,\text{с}$.

С учётом того, что за время нахождения в сети заявки класса 1 в среднем пройдут через узел 1 $\alpha_1(1) = 5$ раз, а через узел 2 – $\alpha_2(1) = 4$ раза, можно найти среднее время пребывания заявок класса 1 в сети:

$U(1) = \alpha_1(1)u_1(1) + \alpha_2(1)u_2(1) = 489\,\text{с}$.

Аналогично, среднее время пребывания в сети заявок класса 2:

$U(2) = \alpha_1(2)u_1(2) + \alpha_2(2)u_2(2) = 12{,}3\,\text{с}$.

Среднее число заявок каждого класса в СеМО может быть найдено по формулам Литтла, связывающим безразмерные и временны*ы*е сетевые характеристики:

$$M(1) = \lambda_0(1)U(1) = \frac{U(1)}{a_0(1)} = 4{,}89 \quad \text{и} \quad M(2) = \lambda_0(2)U(2) = \frac{U(2)}{a_0(2)} \cong 0{,}25.$$

Для получения дополнительных результатов, например в виде гистограмм плотностей распределений времён ожидания и пребывания заявок в СеМО с целью детального анализа свойств исследуемой системы, в область описания GPSS-модели следует включить команды TABLE и QTABLE:

Tw1_k1	QTABLE	QUz1_k1,0,.25,40; *время ожидания в узле 1 заявок класса 1*
Tw1_k2	QTABLE	QUz1_k2,0,.25,40; *время ожидания в узле 1 заявок класса 2*
Tw2_k1	QTABLE	QUz2_k1,0,10,40; *время ожидания в узле 2 заявок класса 1*
TU_k1	TABLE	M1,50,50,40; *время пребывания в сети заявок класса 1*
TU_k2	TABLE	M1,1,1,40; *время пребывания в сети заявок класса 2*

Для двух последних таблиц TU_k1 и TU_k2, в которых накапливается статистика по временам пребывания заявок обоих классов в СеМО, дополнительно в GPSS-модель необходимо вставить два оператора:

```
TABULATE    TU_k1; удаление из модели (СеМО) обслуженной заявки класса 1
TABULATE    TU_k2; удаление из модели (СеМО) обслуженной заявки класса 2
```

Первый оператор должен быть вставлен в модуль 1 перед оператором TERMINATE для отметки времени выхода из СеМО заявки класса 1, а второй – в модуль 3 перед оператором TERMINATE для отметки времени выхода из СеМО заявки класса 2.

В этом случае кроме средних значений временных характеристик могут быть получены значения среднеквадратических отклонений соответствующих характеристик:

$\sigma_{w_1(1)} = 5{,}02$ с; $\sigma_{w_2(1)} = 98{,}90$ с; $\sigma_{w_2(1)} = 4{,}94$ с;

$\sigma_{U(1)} = 737{,}3$ с; $\sigma_{U(2)} = 5{,}72$ с .

6.8. Резюме

1. Универсальным и наиболее эффективным методом исследования сложных систем со стохастическим характером функционирования является имитационное моделирование, предоставляющее возможность исследования систем *любой сложности с любой степенью детализации и получения наиболее полных результатов*.

Имитационная модель представляет собой программу для ЭВМ, реализующую заданное логико-алгоритмическое описание исследуемой системы.

Имитационное моделирование часто называют статистическим, поскольку сбор и обработка результатов имитационного моделирования реализуется методами математической статистики, позволяющими получить результаты в любом объёме – от средних значений и нескольких первых начальных или центральных моментов *до законов распределений*.

К основным процедурам имитационного моделирования относятся:

- организация службы времени;
- сбор и статистическая обработка результатов моделирования;
- выработка (генерирование) случайных величин с заданными законами распределений.

Одна из основных проблем имитационного моделирования – организация службы времени, определяющей способ изменения (продвижения) модельного времени, протекающего в моделируемой системе. В настоящее время наиболее эффективным признан способ «*продвижения модельного времени с переменным шагом до ближайшего события*», реализуемый в большинстве систем имитационного моделирования.

2. Случайные величины в имитационных моделях формируются *программными генераторами* (датчиками), вырабатывающими *псевдослучайные последовательности*, представляющие собой детерминированные числа и обладающие *статистическими свойствами случайных чисел*.

Основными методами формирования равномерно распределённых в интервале (0; 1) псевдослучайных последовательностей являются:

- метод квадратов;
- метод произведений;
- мультипликативный конгруэнтный метод.

Оценка качества генераторов равномерно распределенных псевдослучайных последовательностей проводится с использованием трёх видов проверки:

- на периодичность;
- на случайность;
- на равномерность.

При проверке на случайность программных генераторов *двоичных* псевдослучайных последовательностей используются тесты проверки частот, пар, комбинаций, серий, корреляций.

Псевдослучайные последовательности с заданным законом распределения формируются в ЭВМ на основе программных генераторов равномерно распределённых случайных величин одним из следующих методов:

- аналитическим (обратной функции);
- табличным;
- методом композиций.

3. Общецелевая система имитационного моделирования GPSS World является одной из наиболее доступных и популярных для работы на персональных компьютерах под управлением ОС Windows. GPSS World обладает специальными средствами, которые делают процесс моделирования эффективным и наглядным. GPSS World включает в себя языки программирования GPSS и PLUS и компилятор.

Основными объектами системы имитационного моделирования GPSS World, которые всегда используются при моделировании, являются:

- программа, написанная на языке *GPSS* (*GPSS-модель*),
- исполняемый объект, создаваемый в результате трансляции GPSS-модели и реализующий *процесс моделирования* на ЭВМ,
- *отчёт* с результатами моделирования.

Элементами языка GPSS World являются алфавитно-цифровые символы, имена, метки, переменные, числа, системные числовые атрибуты (СЧА), арифметические операторы, операторы отношения, логические операторы, выражения, процедуры.

Объекты GPSS-модели могут быть разбиты на следующие группы:

- основные объекты (операторы и транзакты);
- оборудование (приборы или одноканальные устройства, памяти или многоканальные устройства, очереди, логические ключи);
- числовые объекты (ячейки, матрицы, переменные, функции, таблицы);
- генераторы случайных чисел (встроенные, библиотечные, табличные);
- групповые списки (списки пользователя, числовые группы,

группы транзактов);

- потоки данных.

Объекты в GPSS-модели могут формироваться автоматически, либо должны объявляться с использованием специальных команд – операторов описания. К объявляемым объектам относятся: памяти, переменные, матрицы, таблицы, функции, а также параметры транзактов.

4. GPSS-модель представляет собой последовательность операторов, описывающих логику работы моделируемой системы, которые могут быть разбиты на две группы: GPSS-операторы и PLUS-операторы.

GPSS-операторы делятся на исполняемые операторы, операторы описания и операторы управления.

Исполняемые операторы, называемые также операторами блоков или просто *блоками*, непосредственно реализуют процесс моделирования.

Операторы описания и *операторы управления*, называемые в GPSS World *командами*, используются соответственно для *описания* многоканальных устройств, переменных, функций, матриц, таблиц и для *управления* процессом моделирования (запуск, остановка и продолжение процесса моделирования, сброс статистики, завершение моделирования и т.п.).

Команды могут быть *срочными и несрочными*.

Оператор GPSS World, в общем случае, содержит 4 поля: поле *метки*, поле *операции*, поле *операндов* и поле *комментариев*.

Все операторы, кроме оператора описания FUNCTION, записываются в одну строку и могут содержать до 250 символов, включая комментарий.

5. Для запуска процесса моделирования используется команда START, которая может находиться в GPSS-модели в качестве последнего оператора или может быть задана после трансляции из меню GPSS World.

Реализация процесса моделирования заключается в перемещении в модели *транзактов, которые* последовательно переходят от блока к блоку в заданной алгоритмом моделирования последовательности.

Транзакты создаются и уничтожаются в модели с помощью специальных операторов: GENERATE и TERMINATE.

В общем случае, *в модели может находиться множество транзактов*, однако *в один и тот же момент времени двигается только один транзакт*.

Транзакт, продвигаемый в модели в данный момент времени, называется *активным*.

Интервал времени, в течение которого транзакт находится в модели, называется *резидентным временем* транзакта.

Интервал времени, в течение которого транзакт проходит между двумя произвольно выбранными точками модели, называется *транзитным временем*.

Каждому транзакту в модели присваивается порядковый номер по мере появления их в модели, начиная с *единицы*.

Работа реальных систем протекает во времени, для отображения которого в GPSS-модели используется *таймер модельного времени*, содержание которого корректируется *автоматически* в соответствии с логикой, предписанной моделью. Единица времени (секунды, минуты, часы или их доли) задается разработчиком модели.

6. Для реализации перемещения транзактов в GPSS-модели используются следующие *списки* (*цепи*):

- *список текущих событий (СТС)*;
- *список будущих событий (СБС)*;
- *списки повторных попыток (СПП)*;
- списки одноканального устройства, включающие *список отложенных прерываний*, *список прерываний*, *список задержки*, *список повторных попыток*;
- списки многоканального устройства, включающие *список задержки*, *список повторных попыток*;
- *списки пользователя*.

В любой модели всегда формируется *один список текущих* и *один список будущих событий*. Остальные списки формируются по мере необходимости.

7. Каждый транзакт имеет множество параметров, называемых *атрибутами транзакта*, к которым относятся:

- *параметры*, число которых не ограничено;
- *приоритет* транзакта;
- *время входа транзакта в систему*;
- *текущий блок*, в котором находится транзакт;
- *следующий блок*, в который должен перейти данный транзакт;
- *список*, в котором находится транзакт в некоторый момент времени.

8. В GPSS World завершение процесса моделирования может быть реализовано:

- принудительно с помощью срочной команды HALT;
- по некоторому условию, задаваемому командой STOP;
- по достижению содержимого «счётчика завершений» значения меньше или равного нулю.

Последний способ используется наиболее часто при моделировании систем и сетей массового обслуживания.

По завершению моделирования формируется и выводится на экран стандартный отчет, содержащий основные результаты моделирования.

9. Переменные, используемые в операндах операторов GPSS и в выражениях, называются *атрибутами*.

Числовые атрибуты, *автоматически поддерживаемые в GPSS* и доступные в течение всего процесса моделирования, называются *системными числовыми атрибутами (СЧА)*. Их значения в любой момент

в процессе моделирования доступны пользователю за счет использования специальных наименований этих атрибутов.

В GPSS используются СЧА трёх типов: СЧА *объектов*, СЧА *системы*, СЧА *транзактов*.

Имя СЧА объектов состоит из двух частей: *группового имени* (идентифицирующего тип объекта и тип информации) и имени или номера *конкретного члена группы*.

10. Встроенная библиотека процедур GPSS World содержит более 20 вероятностных распределений, в том числе равномерное (Uniform), экспоненциальное (Exponential) и др.

Для обращения к вероятностному распределению необходимо указать имя библиотечной процедуры и её параметры, заключённые в круглые скобки и отделённые друг от друга запятой.

Библиотечные процедуры вероятностных распределений могут использоваться в выражениях, а также в качестве операнда A в операторах GENERATE и ADVANCE.

11. В GPSS World имеется 53 операторов блоков, из которых примерно половина используется для построения имитационных моделей простейших систем и сетей массового обслуживания.

Операторы могут быть без операндов или содержать от 1 до 7 операндов, некоторые из которых могут быть необязательными. При отсутствии операндов их значения принимаются по умолчанию. *Отсутствие обязательных операндов приводит к ошибке*.

К операторам *генерирования*, *задержки* и *удаления* транзактов относятся:

- GENERATE (генерирование транзактов);
- ADVANCE (задержка транзакта на заданное время);
- TERMINATE (удаление транзактов из модели).

Операторы *одноканальных устройств* (приборов):

- SEIZE (занятие транзактом прибора);
- RELEASE (удаление транзакта из прибора).

Операторы *многоканальных устройств* (памятей):

- ENTER (вход транзакта в многоканальное устройство);
- LEAVE (удаление транзакта из многоканального устройства).

Операторы *очередей*:

- QUEUE (фиксация момента поступления транзакта в очередь);
- DEPART (фиксация момента удаления транзакта из очереди).

Условные операторы:

- TEST (поверка значения СЧА и передача активного транзакта в блок, отличный от последующего);
- TRANSFER (передача транзакта в блок, отличный от последующего);
- GATE (изменение маршрута движения транзактов в зависимости от состояния некоторого объекта).

Операторы *приоритетного обслуживания*:

- PRIORITY (изменение уровня приоритета активного транзакта);
- PREEMPT (захват прибора поступившим транзактом);
- RETURN (освобождение прибора активным транзактом).

Оператор *логических ключей*:

- LOGIC (изменение состояния логического ключа).

К *прочим* операторам относятся:

- ASSIGN (назначение и изменение параметра транзакта),
- MARK (запись значения абсолютного времени в качестве одного из параметров активного транзакта),
- TABULATE (занесение значений в таблицу – обновление статистики).

12. В GPSS World используются 24 команды (описания и управления), из которых для построения и реализации имитационных моделей *простейших* систем и сетей массового обслуживания оказывается достаточным использование немногим более половины. Команды, как и операторы блоков, могут быть без операндов или содержать от 1 до 5-и операндов, некоторые из которых могут быть необязательными. Значения необязательных операндов при их отсутствии принимаются по умолчанию. *Отсутствие обязательных операндов приводит к ошибке.*

К командам описания относятся:

- FUNCTION (описание функции);
- TABLE (описание таблицы);
- QTABLE (описание таблицы очереди);
- STORAGE (описание ёмкости многоканального устройства);
- VARIABLE (описание арифметической переменной).

К командам управления относятся:

- CLEAR (сброс процесса моделирования в исходное состояние);
- CONTINUE (возобновление прерванного процесса моделирования);
- HALT (прерывает процесс моделирования и очищает очередь команд);
- INCLUDE (вставка в исходную модель и трансляция файла с операторами);
- REPORT (немедленное создание отчета);
- RESET (сброс в ноль статистики и атрибутов системы);
- SHOW (отображает значение выражения в строке состояния окна «Model»);
- START (запуск процесса моделирования);
- STEP (остановка процесса моделирования по определенному количеству входов транзактов в блоки);
- STOP (устанавливает или снимает условие прерывания моделирования).

6.9. Практикум: обсуждение и решение задач

Вопрос 1. Каково соотношение между терминами «имитационное» и «статистическое» моделирование? Эквивалентны ли эти термины?

Обсуждение. Чаще всего подразумевается, что термины «имитационное» и «статистическое» моделирование эквивалентны, и используются как равнозначные термины. Однако, исходя из смыслового содержания этих терминов, всё-таки следует различать, что «имитационное» моделирование означает, что моделирование основано на подражании исследуемому объекту, а «статистическое» моделирование указывает на то, что результаты моделирования накапливаются и обрабатываются методами математической статистики. Эти же результаты могли бы быть получены точно так же на реальной системе путём многократных измерений и их статистической обработки. Таким образом, можно считать, что имитационное моделирование всегда является статистическим. Однако статистическое моделирование не всегда является имитационным. Как показано выше, вычисление определённого интеграла методом Монте-Карло относится к статистическому моделированию, но не является имитационным.

Вопрос 2. Какими достоинствами обладает имитационное моделирование по сравнению с другими методами моделирования?

Обсуждение. Основным достоинством имитационного моделирования является возможность всестороннего детального исследования системы любой сложности и с любой степенью детализации, что невозможно при аналитическом и численном моделировании. Имитационное моделирование позволяет построить математическую модель, максимально приближенную к оригиналу (реальной системе), и, фактически, заменить измерения на реальной системе измерениями на модели. Степень соответствия имитационной модели оригиналу ограничивается только возможностями ЭВМ (производительностью, ёмкостью памяти), на которой проводится моделирование. Естественно, чем сложнее модель, тем более мощной должна быть ЭВМ. Одна из причин, по которой разрабатываются всё более мощные суперЭВМ, – повышенные требования к производительности при решении задач моделирования реальных систем и процессов. Таким образом, можно считать, что имитационное моделирование является универсальным инструментом исследования реальных систем и процессов.

Вопрос 3. Имеют ли результаты имитационного моделирования методическую погрешность и, если да, то чему она равна и как её оценить?

Обсуждение. Имитационному моделированию присущ статистический разброс результатов, который означает, что получаемые значения характеристик имеют методическую погрешность, обусловленную, прежде всего, такими факторами, как длительность моделирования и качество генераторов случайных величин. Наличие методической погрешности проявляется в том, что значения одних и тех же характеристик могут разли-

чаться при использовании в одной и той же модели разных генераторов случайных чисел, а также при различной длительности имитационного эксперимента, причём с увеличением длительности моделирования методическая погрешность уменьшается и лежит обычно в пределах 1-3%.

Следует отметить, что *методическая погрешность различна для разных характеристик*, в чём легко убедиться на следующем гипотетическом примере.

Положим, что в одноканальной СМО с однородным потоком заявок измеряются две характеристики: время ожидания w и время пребывания $u = w + b$ заявок в системе, где b – детерминированная длительность обслуживания заявок в приборе.

Пусть в результате одного эксперимента было получено следующее значение времени ожидания: $w_1 = 10$. Если длительность обслуживания равна $b = 10$, то время пребывания окажется равным $u_1 = 10 + 10 = 20$.

Пусть в результате другого эксперимента было получено значение времени ожидания: $w_2 = 20$. Тогда время пребывания окажется равным $u_2 = 20 + 10 = 30$.

Разница между полученными значениями, представляющая собой погрешность имитационного моделирования, будет составлять:

$$\delta_w = \frac{|w_2 - w_1|}{w_1} 100\% = 100\%, \qquad \delta_u = \frac{|u_2 - u_1|}{u_1} 100\% = 50\%.$$

Итак, погрешность времени ожидания оказалась существенно больше погрешности времени пребывания заявок в системе, что, если подумать, выглядит вполне логично.

Уменьшение методической погрешности имитационного моделирования при использовании качественных генераторов случайных величин достигается за счёт увеличения длительности имитационного эксперимента. При этом некоторые характеристики имеют минимальную погрешность даже при небольшой длительности моделирования (быстрая сходимость результатов к своему истинному значению), в то время как другие характеристики требуют гораздо большей длительности моделирования (медленная сходимость). При моделировании систем и сетей массового обслуживания быстрой сходимостью обычно обладает загрузка, а для времени ожидания характерна медленная сходимость.

Оценить методическую погрешность характеристик моделируемой системы можно «методом срезов», который заключается в следующем. Проводится моделирование длительностью T, и фиксируются полученные на первом срезе значения $\{h_1^{(1)},...,h_N^{(1)}\}$ характеристик (обычно обладающих медленной сходимостью). Затем моделирование продолжается в течение того же времени T, и фиксируются новые полученные на втором срезе значения $\{h_1^{(2)},...,h_N^{(2)}\}$ тех же характеристик. Рассчитывается относительная разность между значениями одноимённых характеристик:

$\delta_i = \frac{\left|h_i^{(2)} - h_i^{(1)}\right|}{h_i} 100\% \quad (i = 1,..., N)$, где в качестве h_i принимается минимальное из двух значений: $h_i = \min(h_i^{(1)}, h_i^{(2)})$ или их среднее значение: $h_i = (h_i^{(1)} + h_i^{(2)})/2$. Значение $\delta = \max(\delta_1,..., \delta_N)$ может рассматриваться как максимальная погрешность имитационного моделирования. Для получения достоверной оценки погрешности рекомендуется выполнить не менее трёх срезов и, если максимальные относительные разности между первым и вторым срезами и между вторым и третьим срезами значительно отличаются, следует продолжить моделирование до тех пор, пока, как минимум, два (а ещё лучше три) соседних среза не дадут приемлемую и примерно одинаковую погрешность.

Вопрос 4. Для чего и каким образом формируются предположения и допущения при разработке модели?

Обсуждение. Предположения и допущения, формируемые в процессе разработки модели, преследуют две цели. Во-первых, это позволяет, во многих случаях, упростить модель и уменьшить её размерность за счёт отбрасывания несущественных факторов и параметров, оказывающих незначительное влияние на процесс функционирования исследуемой системы и, соответственно, на конечные результаты. Во-вторых, при отсутствии каких-либо исходных данных или недостаточно полных сведений о некоторых из них могут и должны вводиться предположения и допущения, позволяющие решить (пусть и упрощённо) поставленную задачу. Действительно, на практике при разработке моделей и исследовании реальных систем зачастую известны только средние значения нагрузочных параметров, представляющих собой случайные величины, и, возможно, их дисперсии. Закон распределения этих параметров обычно не известен. В этом случае для первоначальных оценочных расчётов можно ввести некоторые предположения о законах распределений, позволяющие получить конечные результаты аналитическими методами. При необходимости, дополнительные исследования влияния закона распределения на характеристики функционирования системы могут быть выполнены с использованием имитационного моделирования. Например, если в задаче не оговаривается характер потока заявок и длительности обслуживания заявок в системе массового обслуживания, то может быть введено предположение о том, что поток заявок – простейший, а длительность их обслуживания распределена по экспоненциальному закону. Если не указано количество приборов в узлах сетевой модели, то может быть введено предположение о том, что оно равно 1. В то же время, при решении задач, как это часто бывает на практике, могут иметься «избыточные» исходные данные, которые, вполне возможно, и не нужны для получения результата, поскольку не влияют на характеристики функционирования системы.

Вопрос 5. Если имитационное моделирование является универсальным инструментом исследования, то не значит ли это, что другие методы моделирования не нужны? Или же имитационное моделирование имеет какие-то недостатки?

Обсуждение. Несмотря на универсальность, имитационное моделирование не может полностью заменить другие методы моделирования – аналитические или численные, что обусловлено присущими ему недостатками.

Первый очевидный недостаток связан с высокими требованиями к производительности ЭВМ, на которой проводится моделирование реальных систем, обладающих сложностью и большой размерностью. Естественно, что имитационные модели таких систем представляют собой большие программные комплексы, разработка которых под силу только высококвалифицированным специалистам, владеющим не только приёмами программирования, но и имеющим опыт разработки имитационных моделей, позволяющий разрабатывать модели, учитывающие все существенные особенности структурно-функциональной организации, не перегружая её незначительными подробностями, не влияющими на конечный результат. Всё это делает имитационное моделирование *дорогостоящим* и требующим *значительных временных затрат* как на разработку модели, так и на её реализацию на ЭВМ с учётом того, что для детального исследования свойств системы и получения достоверных результатов необходимо провести большое множество экспериментов, число которых может достигать десятков тысяч.

Последнее связано со вторым недостатком, который заключается в том, что всякий раз при каждом эксперименте результат моделирования получается в точке, то есть справедлив только для заданных в данном эксперименте структурно-функциональных и нагрузочных параметров системы. Поэтому для получения зависимости характеристик системы только от одного параметра требуется провести несколько экспериментов, а для получения доверительного интервала этой зависимости количество экспериментов возрастает многократно. К тому же количество таких параметров может быть значительным.

Третий существенный недостаток имитационного моделирования состоит в невозможности получить приемлемые результаты для систем и сетей массового обслуживания, работающих в области малых (близких к нулю) и больших (близких к единице) загрузок. Действительно, при загрузке системы менее 0,01 вероятность появления очереди очень мала, и в процессе имитационного моделирования даже после прогона достаточно большого числа заявок через систему может оказаться, что ни одна заявка не ждала в очереди, то есть время ожидания будет строго равно нулю. В то же время, точный аналитический расчёт показывает, что время ожидания, хотя и очень маленькое, но не равно нулю. Для того чтобы имитационная модель выдала результат отличный от нуля, возможно потребуется

провести достаточно длительное моделирование, при котором хотя бы одна заявка окажется в состоянии ожидания.

Покажем это на примере простейшей СМО типа M/M/1. Положим, что интенсивность поступления заявок в СМО $\lambda = 0{,}0001\,\text{с}^{-1}$, а длительность обслуживания $b = 100\,\text{с}$. Тогда загрузка системы $\rho = 0{,}001$. В п.5.4.5 показано, что вероятность нахождения в системе *k* заявок определяется по формуле: $p_k = \rho^k(1-\rho)$ $(k = 0, 1, 2, \ldots)$. Тогда вероятность образования очереди (того, что в системе будет две и более заявок) $p_{k>2} = 1 - p_0 - p_1 = 1 - 0{,}99 - 0{,}0099 = 0{,}0001$. Таким образом, для того чтобы в очереди оказалась хотя бы одна заявка, необходимо при имитационном моделировании пропустить через систему более 10 тысяч заявок.

Ещё больше проблем возникает при имитационном моделировании систем, загрузка которых близка к единице. В этом случае практически невозможно получить результат (например, время ожидания заявок в системе), близкий к реальному. Это связано с характером зависимости характеристик СМО от загрузки системы, которая при загрузке, близкой к 1, резко возрастает и стремится к бесконечности (см. рис.4.2).

Действительно, если загрузка той же СМО M/M/1 будет равна $\rho = 0{,}99$ (например за счёт интенсивности $\lambda = 0{,}0099\,\text{с}^{-1}$), среднее время ожидания в соответствии с (4.1) будет равно $w = \dfrac{\rho b}{1-\rho} = \dfrac{0{,}99 * 100}{0{,}01} = 9900\,\text{с}$.

Однако, как сказано выше, имитационному моделированию присущ статистический разброс результатов, в результате которого может оказаться, что в момент завершения процесса моделирования значение загрузки системы будет равно $\rho = 0{,}98$. Тогда среднее время ожидания $w' = \dfrac{\rho b}{1-\rho} = \dfrac{0{,}98 * 100}{0{,}02} = 4900\,\text{с}$, то есть значение времени ожидания будет отличаться более чем в два раза от действительного значения.

Ещё один существенный недостаток, присущий имитационному моделированию, состоит в том, что для сложных систем с большим количеством структурно-функциональных параметров практически невозможно решать задачи оптимального синтеза. Имитационное моделирование позволяет выбрать наилучший вариант структурно-функциональной организации проектируемой системы из нескольких вариантов, но не предоставляет возможностей для решения оптимизационных задач. Для решения этих задач обычно используется аналитическое моделирование.

Задача 1. Для заданной *GPSS-модели*:

а) нарисовать и подробно описать модель исследуемой системы с указанием всех параметров и законов распределений;

б) пояснить, когда (по какому условию) завершится моделирование;

в) определить, существует ли стационарный режим в системе (с

необходимыми обоснованиями, расчетами и пояснениями).

GPSS-модель:

(начало модели)

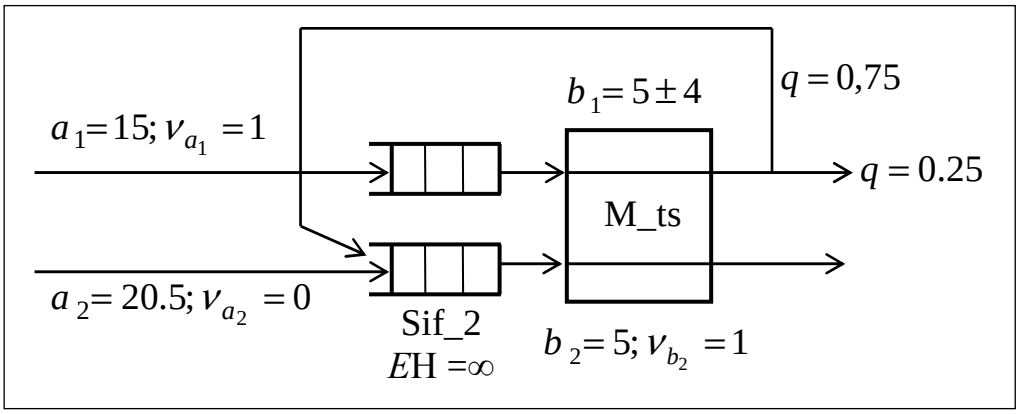
```
GENERATE    (Exponential(2,0,15))
QUEUE       Sif_1
SEIZE       M_ts
DEPART      Sif_1
ADVANCE     5,4
RELEASE     M_ts
TRANSFER    0.25,Noh_1,Noh_2
```

(продолжение модели)

```
            GENERATE    20.5
Noh_1       QUEUE       Sif_2
            SEIZE       M_ts
            DEPART      Sif_2
            ADVANCE     (Exponential(1,0,5))
            RELEASE     M_ts
Noh_2       TERMINATE   2
            START       100000
```

Решение.

а) Наличие двух операторов GENERATE свидетельствует о том, что в моделируемой системе формируется два потока (класса) заявок. Заявки первого класса образуют простейший поток со средним интервалом между заявками 15 единиц времени, а заявки второго класса – детерминированный поток с интервалом 20,5 единиц времени. Формируемые заявки поступают в разные накопители неограниченной ёмкости с именами Sif_1 и Sif_2 соответственно и далее в один и тот же прибор с именем M_ts, где задерживаются на случайное время: заявки класса 1 – на время, равномерно распределённое в интервале (5±4), а заявки класса 2 – на время, распределённое по экспоненциальному закону со средним значением 5 единиц времени. После обслуживания в приборе заявки класса 1 с вероятностью 0,25 направляются к блоку TERMINATE с меткой Noh_2 (удаляются из модели) и с вероятностью 0,75 – к блоку QUEUE с меткой Noh_1 (в накопитель с именем Sif_2) и далее снова попадают в прибор M_ts, где задерживаются на экспоненциально распределённое время со средним значением 5 единиц, то есть обслуживаются уже как заявки класса 2. Таким образом, моделируемая система, показанная на рисунке, представляет собой одноканальную СМО с двумя классами заявок, причём после обслуживания в приборе 75% заявок первого класса переходит во второй класс.



б) Завершение моделирования реализуется оператором TERMINATE и командой START. В момент запуска процесса моделирования в счётчик

завершений заносится значение 100000, указанное в команде START. Всякий раз, когда транзакт (заявка) покидает модель, из счётчика завершений вычитается значение 2, указанное в качестве параметра A оператора TERMINATE. Таким образом, моделирование завершится после обслуживания 50 тысяч заявок.

в) Для того чтобы определить, существует ли стационарный режим в системе, рассчитаем загрузку системы как сумму загрузок, создаваемых заявками классов 1 и 2: $R = \rho_1 + \rho_1$.

Загрузка системы заявками класса 2 рассчитывается как $\rho_2 = \lambda_2 b_2 \approx 0{,}25$, где $\lambda_2 = 1/a_2 = 1/20{,}5 \approx 0{,}05$. Для заявок класса 1 при расчёте загрузки следует учесть, что после первого обслуживания 75% заявок класса 1 остаётся в системе, которые обслуживаются как заявки класса 2 со средним значением $b_2 = 5$ единиц времени. Тогда загрузка системы заявками класса 1 может быть рассчитана как сумма загрузок, создаваемых при первом и втором обслуживании в приборе:

$$\rho_1 = \rho_1^{(1)} + \rho_1^{(2)} = \lambda_1 b_1 + 0{,}75\lambda_1 b_2 = 5/15 + 0{,}75 * 5/15 \approx 0{,}58.$$

Таким образом, загрузка системы $R \approx 0{,}58 + 0{,}05 < 1$, следовательно, система работает без перегрузок, то есть стационарный режим существует.

6.10. Самоконтроль: перечень вопросов и задач

1. Понятия статистического и имитационного моделирования.
2. Основное достоинство имитационного моделирования.
3. Недостатки имитационного моделирования.
4. Основные процедуры имитационного моделирования.
5. По какому принципу осуществляется продвижение модельного времени в имитационной модели.
6. Классификация генераторов случайных величин в зависимости от способа их реализации.
7. Почему величины, вырабатываемые программными генераторами случайных величин, являются псевдослучайными?
8. Перечислить методы генерирования равномерно распределенных случайных величин.
9. Пояснить суть метода квадратов (произведений) для генерирования равномерно распределенных случайных величин.
10. Пояснить суть мультипликативного конгруэнтного метода генерирования равномерно распределенных случайных величин.
11. Понятие длины периода генератора случайных величин.
12. Типы проверок генераторов случайных величин.
13. В чем заключается проверка на периодичность (случайность, равномерность) генераторов случайных величин.
14. Перечислить тесты проверки на случайность генераторов случайных величин.
15. В чем заключается тест проверки частот, пар, комбинаций, серий,

корреляций генераторов равномерно распределённых случайных величин.

16. Перечислить методы генерирования случайных величин с заданным законом распределения.
17. В чем суть аналитического метода (метода обратных функций) генерирования случайных величин с заданным законом распределения?
18. Проиллюстрировать на графике идею аналитического метода генерирования случайных величин с заданным законом распределения?
19. Проиллюстрировать идею аналитического метода (метода обратных функций) генерирования случайных величин на примере экспоненциального закона распределения?
20. Достоинства и недостатки аналитического метода (метода обратных функций) генерирования случайных величин с заданным законом распределения?
21. В чем суть табличного метода генерирования случайных величин с заданным законом распределения?
22. Достоинства и недостатки табличного метода генерирования случайных величин с заданным законом распределения?
23. В чем суть метода генерирования случайных величин с заданным законом распределения, основанного на функциональных особенностях распределений (метод композиций)?
24. Привести примеры генерирования случайных величин с заданным законом распределения, основанного на функциональных особенностях распределений (метод композиций)?
25. Достоинства и недостатки метода генерирования случайных величин с заданным законом распределения, основанного на функциональных особенностях распределений (метод композиций)?
26. Состав системы имитационного моделирования GPSS World.
27. Перечислить элементы языка GPSS World.
28. Классификация объектов системы имитационного моделирования GPSS World.
29. Понятие транзакта.
30. Сколько транзактов может находиться в GPSS-модели одновременно?
31. Сколько транзактов может двигаться в GPSS-модели в один и тот же момент времени?
32. В каких случаях прекращается движение транзакта в GPSS-модели?
33. Какие события в GPSS-моделях массового обслуживания приводят к изменению модельного времени?
34. Какая статистика отражается в стандартном отчёте GPSS-модели?
35. Структура оператора GPSS.
36. Типы GPSS-операторов.
37. В чём отличие операторов блоков от команд?

38. Назначение операторов GENERATE, TERMINATE, ADVANCE, SEIZE, RELEASE, QUEUE, DEPART, ENTER, LEAVE, TEST, TRANSFER, PRIORITY, PREEMPT, RETURN, LOGIC, GATE, MARK, ASSIGN, TABULATE.

39. С помощью какого оператора создаются (уничтожаются) транзакты в GPSS-модели?

40. С помощью каких операторов осуществляется задержка транзакта на определенный период времени (сбор статистики по очередям, изменение маршрута движения транзакта, …) в GPSS-модели?

41. Назначение команд FUNCTION, STORAGE, TABLE, QTABLE, VARIABLE, CLEAR, CONTINUE, HALT, INCLUDE, REPORT, RESET, SHOW, START, STEP, STOP.

42. Задан фрагмент GPSS-модели:

```
GENERATE    20, 10
SEIZE       DIC
ADVANCE     10.5
RELEASE     DIC
TERMINATE
GENERATE    100000
TERMINATE   1
START       10
```

А) Нарисовать и подробно описать модель исследуемой системы (с указанием всех параметров). Б) Пояснить, когда (по какому условию) завершится моделирование. В) Определить, существует ли стационарный режим в системе (с необходимыми обоснованиями, расчетами и пояснениями). Г) Рассчитать среднее число заявок, которые пройдут через систему за время моделирования.

43. Задан фрагмент GPSS-модели:

(начало модели)

```
Met_kom  STORAGE   5
         GENERATE  4.3,1.3
Div_1    ENTER     Met_kom
         ADVANCE   0.5
         LEAVE     Met_kom
         TRANSFER  750, ,Div _2
         TERMINATE
```

(продолжение модели)

```
Div_2    SEIZE     1
         ADVANCE   (Exponential(12,0,4))
         RELEASE   1
         TRANSFER  ,Div _1
         GENERATE  100000
         TERMINATE 2
         START     10
```

А) Нарисовать и подробно описать модель исследуемой системы с указанием всех параметров. Б) Пояснить, когда (по какому условию) завершится моделирование. В) Определить, существует ли стационарный режим в системе (с необходимыми обоснованиями, расчетами и пояснениями). Г) Рассчитать среднее число заявок, которые пройдут через систему за время моделирования.