

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**В.С.ЩЕКЛЕИН**

**МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ**  
Конспект лекций  
для студентов направления 652100 «Авиастроение»

**Ульяновск 2002**

**УДК 621.9.06-229(035)**

**ББК**

**Щ**

Рецензент :

Одобрены секцией методических пособий научно-методического совета университета

**Щеклеин В.С.**

**Щ** Моделирование информационных систем: конспект лекций/  
В.С.ЩЕКЛЕИН. - Ульяновск: УлГТУ, 2002. - с.

Конспект лекций представляет собой подборку материала, использованного в 1999/2000 учебном году при проведении занятий по дисциплине "Моделирование информационных систем". Предназначен для студентов специализаций: 130107 «Программная обработка конструкционных материалов» и 130111 «Проектный менеджмент авиационного производства». Это пособие не является завершенным, в него планируется включить новый разработанный материал, подборка и оформление которого осуществляется в соответствии с утвержденной программой дисциплины.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ МОДЕЛИРОВАНИЯ .....	4
2. СУЩНОСТЬ МЕТОДА СТАТИСТИЧЕСКИХ ИСПЫТАНИЙ И ЕГО РЕАЛИЗАЦИИ С ПОМОЩЬЮ КОМПЬЮТЕРА .....	7
3. ОБОБЩЕННЫЕ АЛГОРИТМЫ СТАТИСТИЧЕСКОГО МОДЕЛИРОВАНИЯ .....	9
4. МОДЕЛИРОВАНИЕ СЛУЧАЙНЫХ ВЕЛИЧИН С ЗАДАННЫМ ЗАКОНОМ РАСПРЕДЕЛЕНИЯ. МОДЕЛИРОВАНИЕ СЛУЧАЙНЫХ СОБЫТИЙ .....	
5. ПОДХОД К МОДЕЛИРОВАНИЮ СИСТЕМ .....	15
6. ЗАДАНИЕ СЛУЧАЙНЫХ ВЕЛИЧИН И СЛУЧАЙНЫХ СОБЫТИЙ В EXCEL .....	21
7. МОДЕЛИРОВАНИЕ МАРКОВСКИХ ЦЕПЕЙ .....	23
8. МОДЕЛИРОВАНИЕ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ .	25
9. СТРУКТУРА ИНФОРМАЦИОННО–ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ .....	26
9.1. Понятие процесса .....	28
9.2. Рабочая нагрузка .....	29
10. ПОКАЗАТЕЛИ ЭФФЕКТИВНОСТИ ИНФОРМАЦИОННЫХ СИСТЕМ .....	30
11. ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ КОМПОНЕНТОВ СИСТЕМЫ .....	31
12. ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМЫ В ЦЕЛОМ .....	32
13. ВЛИЯНИЕ РЕЖИМА ОБРАБОТКИ ДАННЫХ .....	35
14. ХАРАКТЕРИСТИКИ НАДЕЖНОСТИ .....	36
15. ПОСТРОЕНИЕ МАТЕМАТИЧЕСКОЙ МОДЕЛИ ИНФОРМАЦИОННОЙ СИСТЕМЫ .....	40
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	46

## ВВЕДЕНИЕ

Полезность математического моделирования для решения практических задач вообще не вызывает сомнений. Может возникнуть вопрос, а для чего необходимо осваивать моделирование информационных систем (а сейчас эти системы невозможно представить без вычислительной техники) авиастроителям, ориентированным на технологию производства самолетов? Современная технология становится все более и более автоматизированной. Современный авиастроитель, будь он конструктор или технолог должен использовать компьютеры в своей работе. Существует опасность неадекватной оценки возможностей компьютера при решении инженерных задач. Это может привести или к отказу от автоматизации того или иного фрагмента технологического процесса, или к неоправданным расходам на средства вычислительной техники, возможности которых сильно завышены по сравнению с необходимыми. При этом так называемый здравый смысл может приводить к серьезным ошибкам в оценке. Целью дисциплины является вооружение молодого специалиста аппаратом оценки информационно - вычислительных систем для того, чтобы он мог грамотно вписывать средства автоматизации в контуры производства или управления. Кроме того, моделируя те или иные системы, студенты получают опосредованный опыт оптимизации систем и закрепляют навыки использования компьютера при решении профессиональных задач.

## 1. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ МОДЕЛИРОВАНИЯ

Моделированием называется замещение одного объекта другим с целью получения информации о важнейших свойствах объекта – оригинала с помощью объекта – модели.

Модель (франц. *modele* от лат. *modulus* – мера, образец) :

- 1) образец для массового изготовления изделия; марка изделия;
- 2) изделие, с которого снимается форма (шаблоны, лекала, плазы);
- 3) изображаемый художником человек или предмет;
- 4) устройство, воспроизводящее строение или действие какого-либо другого устройства;
- 5) любой образ объекта, процесса или явления, используемый в качестве представителя оригинала (изображение, схема, чертеж, карта);
- 6) математический аппарат, описывающий объект, процесс или явление;
- 7) приспособление для получения отпечатка в литейной форме.

В дальнейшем, если это не будет оговорено особо, под моделью будем понимать математический аппарат.

Всем моделям присуще наличие некоторой структуры (статической или динамической, материальной или идеальной), которая подобна структуре объекта – оригинала. В процессе работы модель выступает в роли относительно самостоятельного квазиобъекта, позволяющего получить при исследовании некоторые знания о самом объекте. Если результаты такого исследова-

дования (моделирования) подтверждаются и могут служить основой для прогнозирования в исследуемых объектах, то говорят, что модель адекватна объекту. При этом адекватность модели зависит от цели моделирования и принятых критериев.

Процесс моделирования предполагает наличие:

- объекта исследования;
- исследователя, имеющего конкретную задачу;
- модели, создаваемой для получения информации об объекте, необходимой для решения задачи.

По отношению к модели исследователь является экспериментатором. Надо иметь в виду, что любой эксперимент может иметь существенное значение в конкретной области науки и техники только при специальной обработке его результатов. Одним из наиболее важных аспектов моделирования систем является проблема цели. Любую модель строят в зависимости от цели, которую ставит перед ней исследователь, поэтому одна из основных проблем при моделировании – это проблема целевого назначения. Подобие процесса, протекающего в модели, реальному процессу, является не самоцелью, а условием правильного функционирования модели. В качестве цели должна быть поставлена задача изучения какой-либо стороны функционирования объекта.

Если цели моделирования ясны, то возникает следующая проблема, проблема построения модели. Это построение оказывается возможным, если имеется информация или выдвинуты гипотезы относительно структуры, алгоритмов и параметров исследуемого объекта. Следует подчеркнуть роль исследователя в процессе построения модели, этот процесс является творческим, базирующимся на знаниях, опыте, эвристике. Формальные методы, позволяющие достаточно точно описать систему или процесс являются неполными или просто отсутствуют. Поэтому выбор той или иной аналогии полностью основывается на имеющемся опыте исследователя, и ошибки исследователя могут привести к ошибочным результатам моделирования.

Когда модель построена, то следующей проблемой можно считать проблему работы с ней, реализацию модели. Здесь основные задачи – минимизация времени получения конечных результатов и обеспечение их достоверности. Для правильно построенной модели характерным является то, что она выявляет лишь те закономерности, которые нужны исследователю, и не рассматривает свойства системы – оригинала, несущественные в данный момент.

Классификация видов моделирования систем приведена на рис. 1.1.

Математическое моделирование – это построение и использование математических моделей для исследования поведения систем (объектов) в различных условиях, для получения (расчета) тех или иных характеристик оригинала без проведения измерений или с небольшим их количеством. В рамках математического моделирования сложились два подхода:

- аналитический;
- имитационный.

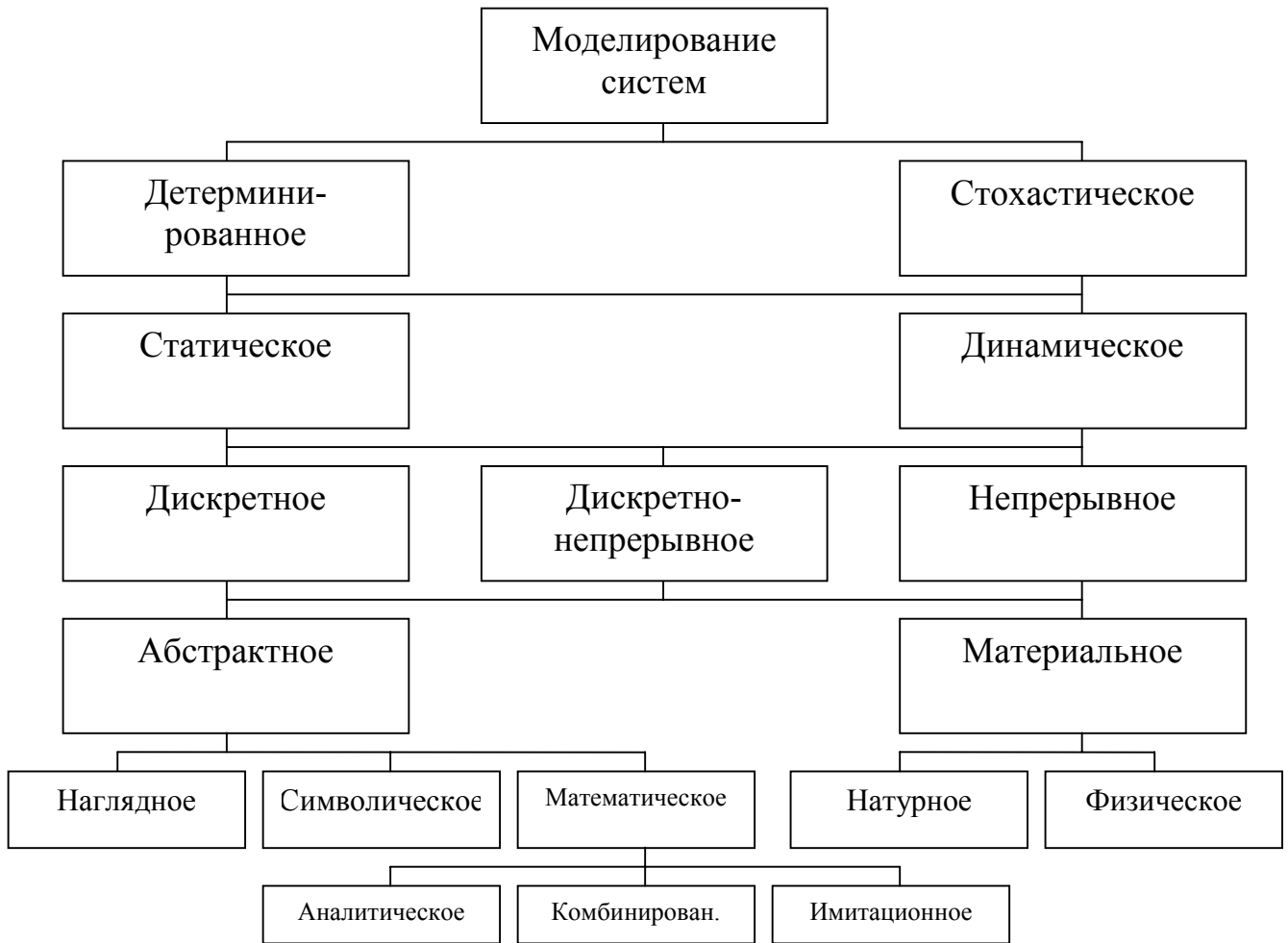


Рис. 1.1.

Аналитический подход основывается на построении формульных зависимостей, связывающих параметры и элементы системы. Такой подход долгое время и был собственно математическим подходом. Однако при рассмотрении сложных систем строгие математические зависимости весьма сложны, требуется большое количество измерений для получения требуемых значений параметров.

Анализ характеристик процессов функционирования сложных систем с помощью только аналитических методов исследования наталкивается на значительные трудности, приводящие к необходимости существенного упрощения моделей либо на этапе их построения, либо в процессе работы с моделью, что снижает достоверность результатов.

Имитационный (статистический) подход в моделировании базируется на использовании предельной теоремы Чебышева при вероятностном представлении параметров системы. На основе предварительного изучения моделируемой системы достаточно просто определяются виды и значения законов распределения случайных величин параметров. В рамках имитационного подхода используются аналитические зависимости между параметрами элементов системы, однако эти зависимости имеют более обобщенный, упрощенный характер. Они значительно проще, нежели зависимости в рамках аналитического подхода.

Математическое моделирование систем, в том числе и информационных, имеет целью оптимизацию структуры систем, выбор наиболее оптимальных режимов функционирования систем, определение требуемых характеристик аппаратного оборудования и программного обеспечения.

Математическое моделирование технологических процессов, в том числе и информационных, имеет основными целями нахождение оптимальных или приемлемых характеристик самого объекта, нахождение оптимальных режимов обработки, обучение персонала, обеспечение определенных функций управления.

В любом случае моделирование должно отвечать следующим требованиям:

- модели должны быть адекватны соответствующим системам или технологическим задачам;
- должна обеспечиваться необходимая точность;
- должно обеспечиваться удобство работы пользователя – специалиста по технологии или по обработке информации (управлению);
- понятный интерфейс управления моделированием;
- достаточная скорость работы;
- наглядность результатов;
- приемлемая стоимость разработки и использования средств моделирования.

## **2. СУЩНОСТЬ МЕТОДА СТАТИСТИЧЕСКИХ ИСПЫТАНИЙ И ЕГО РЕАЛИЗАЦИИ С ПОМОЩЬЮ КОМПЬЮТЕРА**

Метод статистического моделирования заключается в воспроизведении исследуемого процесса при помощи вероятностной математической модели и вычислении характеристик этого процесса. Основан метод на многократном проведении испытаний построенной модели с последующей статистической обработкой полученных данных с целью определения характеристик рассматриваемого процесса в виде статистических оценок его параметров.

Рассмотрим уравнение :

$$y = f(x, t, \xi) , \quad (2.1)$$

где  $y$  - параметр системы, требующий определения,

$x$  - фазовая переменная,

$t$  - время,

$\xi$  - случайный параметр, закон распределения которого нам известен.

Если функция  $f$  существенно нелинейна, то для решения данной задачи нет универсальных методов решения, и достаточно полно отработанные регулярные методы поиска оптимальных решений можно применить только поставив во главу угла видимость использования математики, упрощения приведут к серьезной потере точности. Математическая модель станет не-

адекватной исследуемой системе, и моделирование будет только формой заблуждения.

Однако, если удастся построить функцию  $y = \varphi(\xi)$  и датчик случайных чисел  $\xi_1, \xi_2, \dots, \xi_N$  с заданным законом распределения, то значение  $y$  может быть вычислено как

$$y = \sum \varphi(\xi_i) / N, \quad (2.2)$$

где  $\varphi(\xi_i)$  - значение  $i$ -ой реализации.

Если  $f(x, t, \xi)$  является аналитической моделью процесса преобразования информации или технологического процесса обработки детали, то  $\varphi(\xi)$  будет статистической моделью. Некоторые принципы и приемы построения статистических моделей будут рассмотрены позднее. Важно то, что при построении функции  $y = \varphi(\xi)$  и датчика случайных чисел  $\xi_1, \xi_2, \dots, \xi_N$  на бумаге в подавляющем большинстве случаев достаточно легко реализовать их на ЭВМ в рамках соответствующего программного обеспечения. При этом результаты будут содержать ошибку, но эта ошибка меньше, нежели ошибки из-за допущений в аналитической модели. Кроме того, ошибка из-за применения статистической модели может быть количественно оценена.

Этот прием распространяется и на более сложные случаи, когда уравнение (2.1) содержит не только случайные параметры, но и случайные функции.

После получения на ЭВМ  $N$  реализаций следует этап обработки статистики, позволяющий рассчитать, наряду с математическим ожиданием (2.2) и другие параметры  $\varphi(\xi)$ , например дисперсию

$$D = 1/N^* \sum x.i - 1/N^{2*} (\sum x.i).$$

В методе статистических испытаний для получения достаточно надежных результатов необходимо обеспечивать большое число реализаций  $N$ , кроме того, с изменением хотя бы одного исходного параметра задачи необходимо производить серию из  $N$  испытаний заново. При сложных моделях неоправданно большая величина  $N$  может стать фактором, задерживающим получение результата. Поэтому важно правильно оценить необходимое число результатов. Доверительный интервал  $\varepsilon$ , доверительная вероятность  $\alpha$ , дисперсия  $D$  и число реализаций  $N$  связаны соотношением

$$\varepsilon = D / N \Phi^{-1}(\alpha),$$

где  $\Phi^{-1}(\alpha)$  - функция, обратная функции Лапласа.

На практике можно воспользоваться соотношением

$$N \leq D / \varepsilon^2 * 6,76$$

для  $\alpha \geq 0,99$  принимая, с целью надежности, наибольшее значение  $N$  из соотношения (). Оценка дисперсии  $D$  может быть получена предварительно с помощью той же статистической модели при числе реализаций  $n$ ,  $n \ll N$ .



При построении статистических моделей информационных систем используется общий и прикладной математический аппарат. В качестве примера можно привести аппарат систем массового обслуживания. Система массового обслуживания (СМО) - система, предназначенная для выполнения потока однотипных требований случайного характера. Статистическое моделирование СМО заключается в многократном воспроизведении исследуемого процесса (технического, социального и т.д.) при помощи вероятностной математической модели и соответствующей обработке получаемой при этом статистики. Существуют пакеты программ статистического моделирования СМО, однако они требуют определенных усилий для их освоения и не всегда доступны. Поэтому в рамках дисциплины предлагается достаточно простой подход, позволяющий с наименьшими затратами моделировать простые СМО. При этом предполагается, что пользователь ознакомлен с теорией массового обслуживания и имеет навыки работы на компьютере. Следует помнить, что массовое обслуживание - важный, но далеко не единственный предмет статистического моделирования. На основе этого метода решаются, например, задачи физики (ядерной, твердого тела, термодинамики), задачи оптимизации маршрутов, моделирования игр и т.п.

### **3. ОБОБЩЕННЫЕ АЛГОРИТМЫ СТАТИСТИЧЕСКОГО МОДЕЛИРОВАНИЯ**

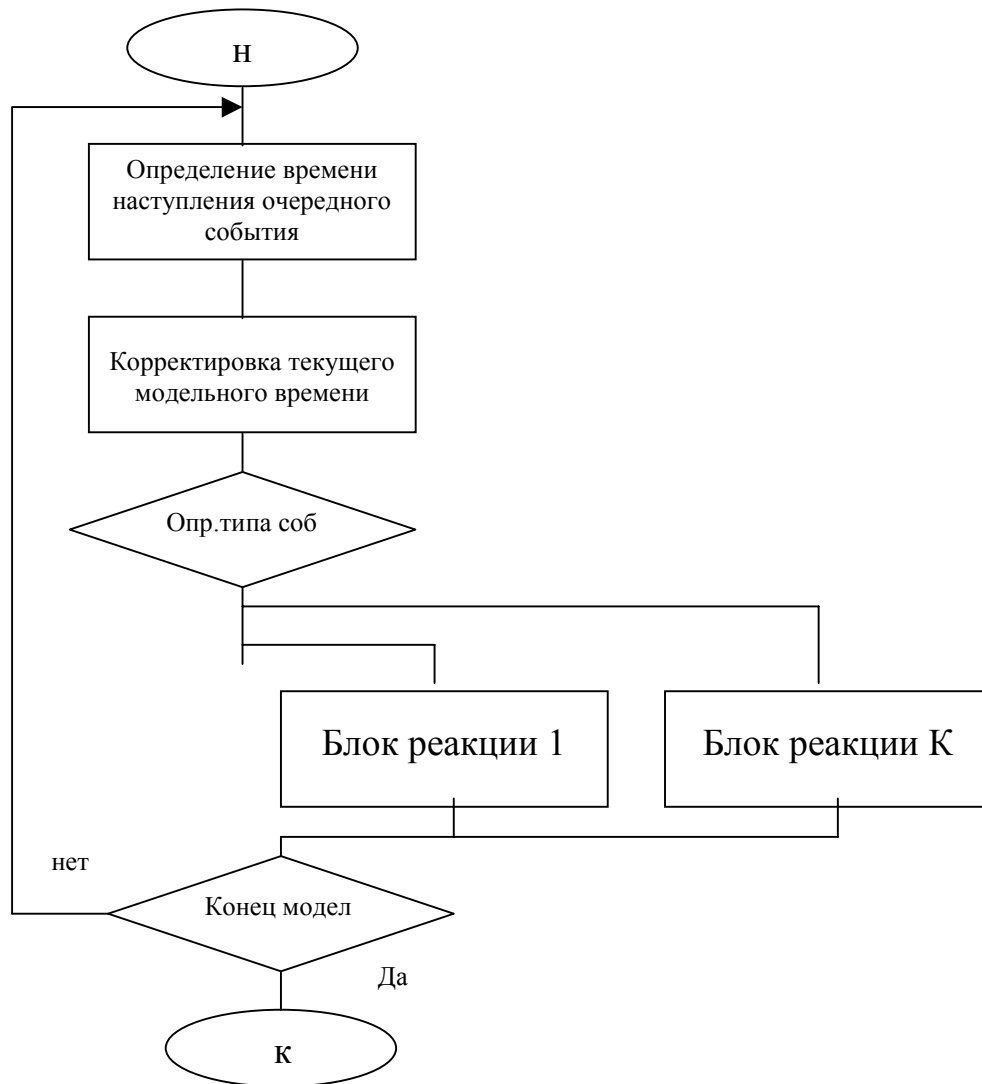
Существуют две схемы статистического моделирования:

- моделирование по принципу особых состояний;
- моделирование по принципу  $\wedge t$ .

Порядок моделирования по принципу особых состояний заключается в выполнении следующих действий:

- 1) случайным образом определяется событие с минимальным временем - более раннее событие;
- 2) модельному времени присваивается значение времени наступления наиболее раннего события;
- 3) определяется тип наступившего события;
- 4) в зависимости от типа наступившего события осуществляется выполнение тех или иных блоков математической модели;
- 5) перечисленные действия повторяются до истечения времени моделирования.

В процессе моделирования производится измерение и статистическая обработка значений выходных характеристик. Эта схема моделирования хорошо подходит для систем массового обслуживания в традиционном их описании. Обобщенный алгоритм моделирования по принципу особых состояний представлен схемой на рис. 3.1.



Моделирование по принципу  $\Delta t$  осуществляется следующим образом :

- 1) устанавливаются начальные состояния, в т. ч.  $t = 0$ ;
- 2) модельному времени дается приращение  $t = t + \Delta t$ ;
- 3) на основе вектора текущих состояний элементов модели и нового значения времени рассчитываются новые значения этих состояний; за  $\Delta t$  может наступить одно событие, несколько событий или же может вообще не происходить событий; пересчет состояния всех элементов системы – более трудоемкая процедура, нежели любой из блоков реакции модели, построенной по принципу особых состояний;
- 4) если не превышено граничное время моделирования, предыдущие пункты повторяются.

В процессе моделирования производится измерение и статистическая обработка значений выходных характеристик. Эта схема моделирования применима для более широкого круга систем, нежели моделирование по принципу особых событий, однако есть проблемы с определением  $\Delta t$ . Если задать его слишком большим - теряется точность, слишком малым - возрастает время моделирования.

На основе базовых схем моделирования можно строить комбинированные и диалоговые схемы, в которых моделирование идет под контролем опе-

ратора с возможностью внесения изменений как в исходные данные, так и в математическую модель процесса.

Структура алгоритма модели, построенной по принципу  $\Delta t$ , проще структуры, построенной по принципу особых состояний. Программная реализация моделей, построенных по одной и той же реальной системе в одной и той же системе программирования, но по разным принципам статистического моделирования, имеет практически одинаковую трудоемкость.

### **Технология моделирования**

В процессе разработки модели можно условно выделить такие этапы описания, как концептуальный, математический и программный. Кроме того, кроме собственно разработки модели необходимо выполнить еще ряд действий, без которых моделирование не может привести к требуемому результату. Итак, рекомендуемый порядок работ в рамках статистического моделирования. Этот порядок не является жестким, но все рекомендованные действия придется выполнять (пусть не всегда в указанной последовательности), к некоторым в процессе работы придется возвращаться.

- 1) создание концептуальной модели. Концептуальная модель - это абстрактная модель, определяющая состав и структуру исследуемой системы, свойства элементов и связей. Строится обычно в словесно-графической форме;
- 2) подготовка исходных данных. Этот этап включает в себя :
  - сбор фактических данных (измерения, анализ документов, метод экспертных оценок);
  - подбор законов распределения случайных величин (по численным значениям параметра строится гистограмма распределения, затем она аппроксимируется кривой, потом эта кривая сравнивается с кривыми плотности распределения различных теоретических законов, выбирается наиболее подходящий из них и проводится оценка степени совпадения эмпирического и теоретического распределения).

В случае необходимости на этом этапе возможны аппроксимация функций, описывающих связи между элементами и выдвижение гипотез по значению новых элементов или параметров;

- 3) разработка математической модели. Создание математической модели преследует две основные цели: - дать формализованное описание структуры и процесса функционирования системы для однозначности их понимания и - попытаться представить процесс функционирования в виде, допускающем аналитическое исследование системы. Разработка единой методики создания математических моделей не представляется возможной. Для начинающих исследователей эффективен путь адаптации одной из уже известных математических моделей к условиям стоящей перед ними задачи;
- 4) выбор метода моделирования. Разработанная математическая модель может быть исследована аналитически и статистически, во втором случае - по принципу особых состояний или по принципу  $\Delta t$ . Если модель позволяет,

лучше провести аналитическое исследование, но это возможно далеко не всегда. Как правило, не имеет значения, моделировать по принципу особых состояний или по принципу  $\wedge t$ , это часто определяется выбором средства моделирования.

5) выбор средств моделирования. Рекомендуется следующая последовательность выбора:

- применить готовые специальные программы, здесь - их освоение, подготовка данных, анализ результатов. Эти средства должны быть корректными. Является грубейшей ошибкой жертвовать адекватностью модели с целью применения того или иного средства моделирования. Если готовых средств нет, то..
- применить средства хорошо зарекомендовавших себя пакетов программ, лучше - ориентированных на моделирование. В данном случае придется изучать возможности и порядок работы, готовить данные в специфическом виде, но это не самый тяжелый путь;
- разрабатывать свое программное средство, используя те или иные среды программирования.

6) проверка адекватности и корректировка модели. Проверка адекватности модели необходима, так как по неверным результатам моделирования могут быть приняты неверные решения. Проверка может производиться путем сравнения показателей, полученных на модели, с реальными, а также путем экспертного анализа. Желательно проведение такого анализа независимым экспертом. Если по результатам проверки адекватности выявляются недопустимые расхождения между системой и ее моделью, в модель вносят необходимые изменения;

7) проведение экспериментов с моделью. Этот этап связан с выполнением предыдущего этапа, а также с определением необходимой точности и, как следствие, числа прогонов программы модели (см п. 1.1.). Кроме того, рекомендуется не включать в статистику результаты начала моделирования, пока модель не войдет в стационарный режим (ориентировочно 5-7 событий,  $100 - 200 \wedge t$ ).

И главное - получение статистических оценок функционирования системы в результате выполнения программы модели на ЭВМ; анализ и использование результатов моделирования. Это целиком зависит от того, с какой целью проводилось моделирование.

#### **4. МОДЕЛИРОВАНИЕ СЛУЧАЙНЫХ ВЕЛИЧИН С ЗАДАНЫМ ЗАКОНОМ РАСПРЕДЕЛЕНИЯ. МОДЕЛИРОВАНИЕ СЛУЧАЙНЫХ СОБЫТИЙ**

Метод статистических испытаний базируется на использовании случайных чисел - возможных значений некоторой случайной величины с заданным распределением вероятностей.

При использовании для моделирования ЭВМ в подавляющем большинстве применений генерация любых случайностей начинается с генерации случайных чисел с равномерным распределением в диапазоне (0, 1) с помощью программы - датчика случайных чисел. Наиболее просто случайные числа в диапазоне (0, 1) получаются из рекуррентного соотношения

$$\gamma_i = (A\gamma_{i-1} + B) \bmod M ,$$

где  $A$  и  $B$  - константы;

$M$  - достаточно большое целое положительное число.

При соответствующем выборе констант и задания некоторого исходного значения  $\gamma_0$  эта формула позволяет получить последовательность целых чисел, равномерно распределенных в интервале (0,  $M-1$ ). Последовательность имеет период повторения, равный  $M$ , поэтому точнее называть эти числа псевдослучайными. Наличие периода повторений может отрицательно сказываться при моделировании с числом реализаций, превышающим значение периода повторений. Для увеличения периода повторения без изменения  $M$  часто применяют специальные приемы "взбадривания" датчика, обычно это заключается в пересмотре  $\gamma_{i-1}$ .

Случайные числа  $\beta_i$ , равномерно распределенные в диапазоне (0, 1) получают из чисел диапазона (0,  $M-1$ ) с помощью масштабного преобразования.

Все доступные системы программирования имеют встроенные подпрограммы - датчики случайных чисел диапазона (0, 1), поэтому при статистическом моделировании получение *равномерно распределенных в диапазоне (a, b) случайных величин* сводится к пересчету

$$\eta = (b - a) * \beta + a ,$$

здесь  $\beta$  - обращение к машинному датчику случайных чисел.

Для экспоненциально распределенных величин

$$\eta = -1/\lambda * \ln \beta .$$

Для распределения Релея

$$\eta = -2\sigma \ln \beta .$$

Для нормального закона распределения можно воспользоваться положением центральной предельной теоремы. Если набрать сумму из  $N$  значений  $\beta$ , то эта сумма будет случайной величиной со средним  $N/2$  и дисперсией  $N/12$ . Вычитание из суммы значения  $N/2$  и деление полученной разности на  $\sqrt{N/12}$  дают случайную величину  $\eta_0$ , распределенную (при достаточно больших  $N$ ) по нормальному закону с нулевым матожиданием и единичной дисперсией. Для ускорения вычислений рекомендуется выбирать  $N = 12, 48, 108$ .

Тогда

$$\eta_0 = (\sum \beta_i - N/2), \text{ для } N = 12;$$

$$\eta_0 = (\sum \beta_i - N/2)/2, \text{ для } N = 48;$$

$$\eta_0 = (\sum \beta_i - N/2)/3, \text{ для } N = 108.$$

Увеличение  $N$  ведет к возрастанию точности, но замедляет процесс моделирования. Для простых задач достаточно  $N = 12$ . Переход к нормальным распределениям с ненулевым ожиданием и дисперсией, отличной от 1, осуществляется по следующей формуле:

$$\eta = \sigma\eta_0 + \mu,$$

где  $\sigma$  - среднеквадратическое отклонение требуемого нормального распределения;

$\mu$  - его математическое ожидание.

До сих пор рассматривалось получение непрерывных случайных величин. Для получения дискретных случайных величин в ряде редких случаев применяют известные соотношения, как например для биномиального распределения

$$b(x, n, \theta) = C_n \theta^x (1 - \theta)^{n-x},$$

$x = 0, 1, \dots, n$ , случайные величины определяются

$$x' = \sum k_i, \text{ где } k_i = 0, \text{ если } \beta_i < \theta; k_i = 1, \text{ если } \beta_i \geq \theta.$$

Когда расчетные соотношения неизвестны и нет возможности их вывести, можно прибегнуть к табличному заданию случайной величины, используя  $\beta$  для определения индекса - указателя элемента таблицы (это годится как для непрерывных, так и для дискретных величин).

### Моделирование случайных событий

В реальных системах объекты могут находиться в том или ином дискретном состоянии, что влияет на работу системы в целом. Например, дискковод компьютера занят или свободен в данный момент, программа загружена или нет, идет этап технологического процесса, или он уже завершен и т.д. При моделировании факт смены (или подтверждения) объектом своего состояния называют событием. Моделирование случайных событий обычно заключается в формировании определенного дискретного значения на основе одной или нескольких случайных величин с соответствующими законами распределения. Случайные события могут быть одиночными или образовывать группы, они могут быть зависимыми или независимыми.

Моделирование одиночного события  $A$ , наступающего с вероятностью  $p$ , определяется как событие, состоящее в том, что выбранное значение случайной величины  $\beta$ ,  $\beta \in [0, 1]$  удовлетворяет неравенству  $\beta < p$ . В дальнейшем принимают  $A = 1$ , если  $\beta < p$  и  $A = 0$  в противном случае.

Моделирование полной группы несовместных событий  $A_m$ ,  $m = 1, 2, \dots, s$ , наступающих с вероятностями  $P(A_m) = p_m$ ,  $\sum p_m = 1$ , определяется как событие, состоящее в том, что выбранное значение  $\beta_i$  случайной величины  $\beta \in [0, 1]$  удовлетворяет неравенству  $l_{m-1} < \beta_j \leq l_m$ , где величины  $l_m = \sum p_i$ ,  $m = 1, 2, \dots, s$ , представляют собой границы интервалов, определяемых как

$$p_m = \sum p_i - \sum p_{i-1}.$$

Процедура моделирования испытаний рассматриваемого вида состоит в выборе значений  $\beta_i$  и сравнении их с величинами  $l_m$ . Исходом испытания является событие  $A_m$ , соответствующее номеру  $m$ -го интервала, в которое попало число  $\beta$ .

Моделирование сложных независимых событий возможно в двух вариантах. В варианте 1 последовательно проверяются условия  $\beta \leq P_A$  и  $\beta \leq P_B$  (см. моделирование одиночного события). Если оба условия выполняются- фиксируется наступление сложного события. Вариант 2 основывается на методике моделирования полной группы событий (нужно строить интервалы). По скорости предпочтительнее вариант 1.

Моделирование сложных зависимых событий рекомендуется производить в следующей последовательности (для двух событий):

- если  $\beta_j \leq P_A$  и  $\beta_{j+1} \leq P_{B/A}$ , то фиксируется событие  $AB$ ;
- если  $\beta_j \leq P_A$  и  $\beta_{j+1} > P_{B/A}$ , то фиксируется событие  $A\bar{B}$ ;
- если  $\beta_j > P_A$  и  $\beta_{j+1} \leq P_{B/A}$ , то фиксируется событие  $\bar{A}B$ ;
- если  $\beta_j > P_A$  и  $\beta_{j+1} > P_{B/A}$ , то фиксируется событие  $\bar{A}\bar{B}$ .

Этот прием легко распространяется и на большее число событий.

## 5. ПОДХОД К МОДЕЛИРОВАНИЮ СИСТЕМ

Целями моделирования информационных систем чаще всего являются оценка их производительности и надежности. В ответственных случаях оцениваются и другие показатели. Оценка обычно производится в интересах решения задач оптимизации.

Производительность и надежность информационных систем связаны с временными аспектами функционирования и зависят в первую очередь от параметров аппаратно-программных средств, которые обобщающе называются вычислительной системой. При оценке производительности перво-степенное значение имеет продолжительность вычислительных процессов. При оценке надежности исследуется продолжительность пребывания системы в различных состояниях, которые меняются из-за отказов в устройствах и программах и последующего восстановления работоспособности. Для вычислительных систем типично наличие случайных факторов, влияющих на характер протекания процессов. Продолжительность процессорной обработки, число, порядок и параметры обращений к периферийным устройствам зависят от исходных данных, которые порождаются вне системы и носят для

нее случайный характер. Случайными являются потоки отказов и времена восстановления отказавших элементов. В связи с этим при оценке функционирования вычислительных систем используется вероятностный подход. Этот подход предполагает, что на процессы воздействуют случайные факторы и свойства процессов и системы в целом проявляется статистически на множестве их реализаций.

Процессы, происходящие в вычислительных системах, представляются в моделях как непрерывные или дискретные случайные процессы. При исследовании вычислительных систем чаще всего приходится иметь дело с дискретными случайными процессами, определенными на конечном множестве состояний, причем процессы рассматриваются или в непрерывном, или в дискретном времени.

Вероятностный подход к описанию функционирования вычислительных систем приводит к использованию аппарата теории вероятностей и математической статистики в качестве математической базы методов исследования. Случайные величины, соответствующие параметрам элементов моделей, могут представляться на разных уровнях, среди которых наиболее широко используются четыре:

- 1) статистическая выборка  $a_1 \dots a_n$ , определяющая случайную величину набором значений;
- 2) закон распределения случайной величины с его параметрами;
- 3) математическое ожидание и дисперсия;
- 4) математическое ожидание.

На первом уровне случайная величина определяется наиболее полно (при достаточности статистической выборки), на последнем – наименее детально.

При моделировании вычислительных систем применяют различный математический аппарат, наиболее часто - теорию марковских процессов и аппарат теории массового обслуживания.

### **Краткое описание марковских процессов**

Марковским называется случайный процесс, состояние которого в очередной момент времени зависит только от текущего состояния и не зависит от предыстории процесса. В классе марковских процессов выделяют процессы с дискретными состояниями, называемые марковскими цепями. Когда множество состояний процесса  $S = \{s_1 \dots s_k\}$  конечно, марковская цепь называется конечной. Конечная марковская цепь может быть определена в непрерывном или дискретном времени. В первом случае переходы процесса из одного состояния в другое связываются с произвольными моментами времени  $t_0, t_1, t_2 \dots$  и цепь называют непрерывной; во втором переходы идут только в фиксированные моменты времени, обозначаемые порядковыми номерами  $t = 0, 1, 2, \dots$ , и цепь называется дискретной.

Дискретная марковская цепь определяется :

- 1) множеством состояний  $S = \{s_1, \dots, s_k\}$ ;



- 2) матрицей вероятностей переходов (или переходных вероятностей), характеризующей вероятности перехода процесса с текущим состоянием  $s_i$  в следующее состояние  $s_j$ ;
- 3) вектором начальных вероятностей  $P = \{p_1, \dots, p_k\}$ , определяющим вероятность  $p_i$  того, что в начальный момент времени  $t = 0$  процесс находится в состоянии  $s_i$ .

Марковские цепи классифицируются в зависимости от возможности перехода из одних состояний в другие. Основными являются два класса: поглощающие и эргодические цепи.

Поглощающая марковская цепь содержит поглощающее состояние, достигнув которого процесс уже никогда его не покидает, по сути дела это моделирует прекращение процесса. Из какого бы состояния ни начался процесс, при  $n \rightarrow \infty$ : с вероятностью 1 он окажется в поглощающем состоянии  $s_0$ . Основная характеристика процесса, порождаемого поглощающей марковской цепью, - число пребывания процесса в состояниях  $s_1, \dots, s_k$  до момента поглощения. Поглощающие марковские цепи используются в качестве моделей программ. При моделировании программы состояния цепи отождествляются с блоками программы, а матрица переходных вероятностей определяет порядок переходов между блоками, зависящий от структуры программы и распределения исходных данных, значения которых влияют на развитие вычислительного процесса. В результате удается вычислить число обращений к блокам программы и время выполнения программы. Аналогично можно представить и работу с аппаратурной частью, когда состояния отображают использование отдельных устройств компьютера.

Эргодическая (возвратная) марковская цепь представляет собой множество состояний, связанных матрицей переходных вероятностей таким образом, что из какого бы состояния процесс ни исходил, после некоторого числа шагов он может оказаться в любом состоянии, в том числе и исходном. Процесс, порожденный эргодической цепью, начавшись в некотором состоянии, никогда не завершается, а последовательно переходит из одного состояния в другое, попадая в различные состояния с разной частотой, зависящей от переходных вероятностей. Поэтому основная характеристика эргодической цепи – вероятности пребывания процесса в состояниях  $s_1, \dots, s_k$ . Эргодические цепи используются в качестве моделей надежности систем. При этом состояния системы, различающиеся составом исправного и отказавшего оборудования, трактуются как состояния эргодической цепи, переходы между которыми связаны с отказами и восстановлениями устройств и реконфигурацией связей между ними, проводимой для сохранения работоспособности системы в целом. Оценки характеристик эргодической цепи дают представление о надежности поведения системы. Кроме того, эргодические цепи используются в качестве базовых моделей взаимодействия устройств с задачами, поступающими на обработку.

Марковский процесс, в котором переходы между состояниями разрешаются в любой момент времени, называется непрерывной марковской цепью. Однородная непрерывная марковская цепь, поведение которой в любой

момент времени подчиняется одному и тому же закону, задается матрицей интенсивности переходов  $Q = [q_{ij}]$ ;  $i, j = 1 \dots K$ .

Интенсивность переходов определяется следующим образом:

$$q_{i,i} = \lim_{\Delta t \rightarrow 0} \frac{p_{i,i}(\Delta t) - 1}{\Delta t}$$

где  $p_{i,j}(\Delta t)$  – вероятность перехода процесса из состояния  $s_i$  в состояние  $s_j$  за время  $\Delta t$ .

Вероятность перехода процесса в любое новое состояние равна  $-q_{i,i}\Delta t$ .

Основная характеристика непрерывной марковской цепи – стационарное (финальное) распределение вероятностей состояний  $A = \{a_1, \dots, a_k\}$ , где  $a_i$  – вероятность пребывания процесса в состоянии  $s_i$ .

### Определение систем массового обслуживания

В теории массового обслуживания изучаются системы, на вход которых поступает случайный поток заявок (требований), приходящихся в общем случае на случайные моменты времени. Поступившая заявка обслуживается в системе путем предоставления ей некоторых ресурсов на какое-то время и, будучи в той или иной мере обслуженной, покидает систему. Наиболее характерный момент функционирования систем массового обслуживания – это наличие очередей, в которых заявки ждут момента освобождения ресурсов, занятых обслуживанием других заявок.

В простейшем случае система массового обслуживания (СМО) определяется потоком заявок, длиной очереди и дисциплиной обслуживания (порядком выбора заявок из очереди), числом каналов (приборов) обслуживания, распределением длительности обслуживания. В более сложных случаях рассматривается надежность приборов обслуживания. Исследование СМО заключается в определении ее пропускной способности, времени ожидания заявки в очереди, загрузки каналов обслуживания. Структура многоканальной однофазной СМО приведена на рис. 5.1. Многоканальной называется такая СМО, где заявка может получить определенный тип обслуживания в одном из нескольких каналов. Многофазной называется СМО, характер обслуживания в которых является многоэтапным, например переход заготовки от станка к станку, на совокупности которых реализуется технологический процесс изготовления детали.

Поток заявок физически представляет собой явления одной природы, например покупатели в магазине, посетители в парикмахерской, попытки позвонить по телефону, желания решить задачу с использованием компьютера и т.д. С математической точки зрения поток заявок на обслуживание характеризуется законом распределения случайной величины – времени между появлением соседних заявок.

Число мест в очереди – это число заявок, которые могут ожидать обслуживания, находясь в СМО. По ограничению очереди СМО разделяются на системы с нулевой длиной очереди, на системы с конечной (определенной) длиной очереди и системы с неограниченной длиной очереди. В системах с

нулевой и конечной длиной очереди имеют место отказы – ситуации, когда в системе нет свободных каналов и все места в очереди уже заняты. Заявка, попавшая на отказ, считается потерянной.

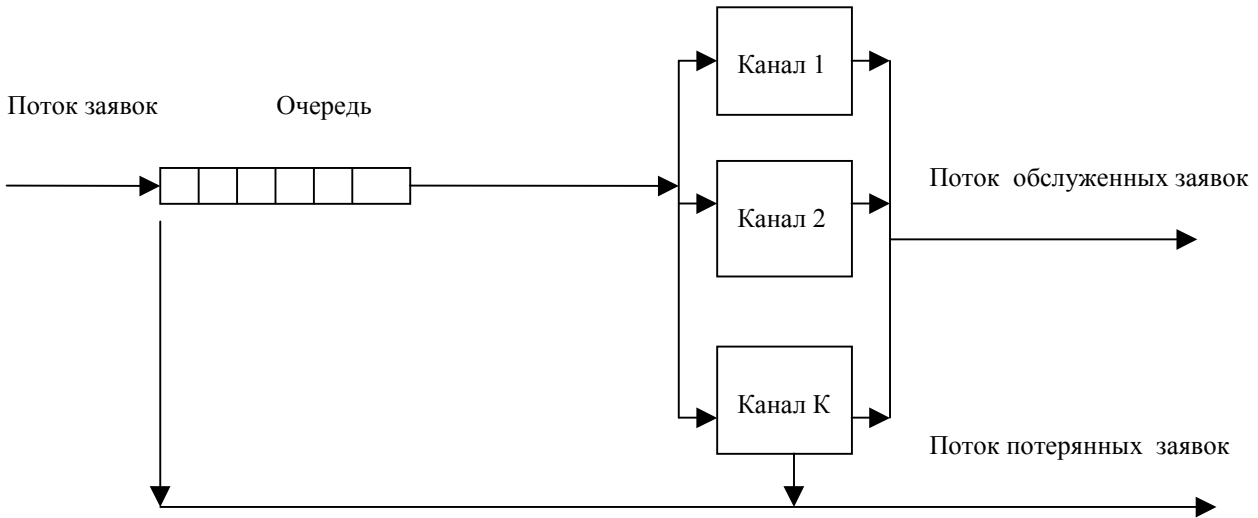


Рис. 5.1.

Дисциплина очереди может быть без приоритетов или с приоритетами. И в том, и в другом случае выборка заявок на обслуживание может осуществляться по правилам: «первый пришел - первый вышел», «последний пришел - первый вышел», выбор заявки из очереди может быть организован случайным образом.

Время обслуживания заявки является случайной величиной. В общем случае производительность различных каналов СМО является различной, и время обслуживания конкретной заявки зависит от того, на какой канал она попала. В простейшем случае все каналы одинаковы. При анализе СМО определяется коэффициент загрузки канала – отношение времени, когда канал обслуживал заявки, к общему времени функционирования. В процессе работы каналы могут выходить из строя и восстанавливаться. В этом случае вводится понятие потока отказов и потока восстановлений, а также коэффициента готовности и коэффициента исправного действия. В общем случае заявка, во время обслуживания которой произошел отказ, считается потерянной. Можно смоделировать повторное выполнение этой заявки или продолжение обслуживания на другом канале.

Успешно обслуженные заявки образуют поток обслуженных заявок. Заявки, не принятые на обслуживание по занятости всех каналов и мест в очереди, а также необслуженные из-за выхода каналов из строя, образуют поток потерянных заявок. Сумма обслуженных и потерянных заявок равна числу заявок, поступивших на вход системы.

Таким образом, СМО характеризуется следующим набором параметров:

- 1) распределением длительности интервалов между заявками входящего потока  $p(a)$ ;
- 2) числом мест в очереди;

- 3) дисциплиной обслуживания заявок  $D$ ;
- 4) числом обслуживающих приборов (каналов)  $K$ ;
- 5) распределение длительности обслуживания заявок приборами  $p(b)$ ;
- 6) надежностью обслуживающих приборов.

Указанный набор параметров полностью определяет порядок функционирования системы. Процесс функционирования количественно оценивается следующим набором основных характеристик:

- 1) загрузкой – средним по времени числом приборов, занятых обслуживанием;
- 2) длиной очереди – средним числом заявок, ожидающих обслуживания;
- 3) числом заявок, находящихся в системе (в очереди и на приборах);
- 4) временем ожидания – средним временем пребывания заявок в очереди;
- 5) временем пребывания заявки в системе;
- 6) временем исправной работы приборов;
- 7) количеством обслуженных и потерянных заявок.

В многофазных СМО очереди могут быть на каждой фазе обслуживания. В тех случаях, когда возможно возвращение заявки после обслуживания на последующей фазе к обслуживанию на одной из предшествующих, говорят о более сложных образованиях - сетях массового обслуживания. Сеть массового обслуживания можно представить как совокупность узлов – отдельных СМО. Сеть массового обслуживания задается следующим набором параметров:

- 1) параметрами источника заявок;
- 2) структурой, определяющей конфигурацию связей и вероятности передачи заявок между узлами сети;
- 3) параметрами СМО, образующих сеть.

Функционирование сети массового обслуживания определяется совокупностью узловых и сетевых характеристик. Узловые характеристики оценивают работу каждой СМО и включают в себя весь набор их характеристик. Сетевые характеристики оценивают функционирование сети в целом и включают в себя:

- 1) загрузку – среднее по времени число заявок, обслуживаемых сетью и среднее число каналов, занятых обслуживанием;
- 2) число заявок, ожидающих обслуживания в сети;
- 3) число заявок, находящихся в сети в состоянии обслуживания или ожидания;
- 4) суммарное время ожидания заявки в сети;
- 5) суммарное время пребывания заявки в сети.

В рамках дисциплины пока не запланировано рассмотрение сетей массового обслуживания, но надобность в таком рассмотрении отвергать нельзя. Могут возникнуть научные или производственные задачи, требующие такого аппарата.

## 6. ЗАДАНИЕ СЛУЧАЙНЫХ ВЕЛИЧИН И СЛУЧАЙНЫХ СОБЫТИЙ В EXCEL

Задание случайной величины, равномерно распределенной в диапазоне (0,1), осуществляется с помощью функции СЛЧИС(). При этом значение ячейки изменяется при внесении в таблицу (в другие ячейки) любого изменения. Изменения происходят и в том случае, если в другую ячейку вновь записывается то же самое значение, если происходит очистка содержимого, удаление, т.е. любой пересчет таблицы.

Проверка датчика может быть осуществлена в рамках процедуры, подобной приведенной ниже.

В ячейки таблицы записываются числа и формулы (см. табл. 6.1).

Таблица 6.1

Адрес	Содержимое ячейки	Комментарий
A1	0	Для установки в исх. положение
B1	=СЛЧИС()	Функция датчика случ. чисел
A2	1	Начальное и
B2	10000	Граничное число обращений
C2	=ЕСЛИ(A1=0;A2; ЕСЛИ(C2<B2;C2+A1;B2))	Текущее число обращений к ДСЧ
D2	=ЕСЛИ(A1=0;0; ЕСЛИ(C2<B2;D2+B1;D2))	Сумма значений ДСЧ
D3	=D2/C2	Оценка математич. ожидания

После этого необходимо перевести Excel в режим циклических расчетов с количеством итераций, равным 1 (Меню “Сервис – Параметры - Вычисления”). При задании предельного числа итераций больше 1, значение ДСЧ будет пересмотрено лишь в первой итерации, а в дальнейшем в рамках цикла изменяться не будет.

Для запуска процедуры необходимо записать в ячейку A1 значение шага 1 и после этой записи нажать и удерживать клавишу F9. Пока нажата эта клавиша, происходит пересмотр значения датчика, накопление суммы различных значений датчика, вычисление оценки математического ожидания. За несколько десятков итераций оценка математического ожидания выходит на уровень 0.5 и в дальнейшем, колеблясь, приближается к теоретическому 0.5.

Получение равномерно распределенных в диапазоне (a, b) случайных величин сводится к пересчету

$$\eta = (b - a) * \beta + a$$

здесь  $\beta$  - обращение к машинному датчику случайных чисел.

В Excel для задания в ячейке такого рода случайной величины необходимо записать в ячейку формулу:  $= (адр2-адр1)*СЛЧИС() + адр1$ , где  $адр1$  – адрес ячейки, содержащей число  $a$ ,  $адр2$  – адрес ячейки, содержащей число  $b$  ( $a$  и  $b$  – границы интервала).

Экспоненциально распределенные случайные числа получаются на основе датчика СЛЧИС() в соответствии с соотношением

$$\eta = -1/\lambda * \ln \beta .$$

В Excel для задания в ячейке такого рода случайной величины необходимо записать в соответствующую ячейку формулу: = -1/адр1\*LN(СЛЧИС()), где адр1 – адрес ячейки, содержащей число  $\lambda$ .

Аналогично в электронной таблице Excel можно задать любую случайную величину с известным законом распределения.

### Моделирование случайных событий в Excel

Моделирование одиночного события А, наступающего с вероятностью  $p$ , определяется как событие, состоящее в том, что выбранное значение случайной величины  $\beta$ ,  $\beta \in [0, 1]$  удовлетворяет неравенству  $\beta < p$ .

В Excel для моделирования одиночного события достаточно задать формулу типа

$$= \text{ЕСЛИ} (\text{ДСЧ} < P; 1; 0) ,$$

где ДСЧ – адрес ячейки, содержащей случайное число или функция датчика;

P – заданная вероятность события;

1 и 0 – константы, определяющие, наступило событие или нет (могут быть любые константы).

Моделирование полной группы несовместных событий  $A_m$ ,  $m = 1, 2, \dots, s$ , наступающих с вероятностями  $P(A_m) = p_m$ ,  $\sum p_m = 1$ , определяется как событие, состоящее в том, что выбранное значение  $\beta_j$  случайной величины  $\beta \in [0, 1]$  удовлетворяет неравенству  $l_{m-1} < \beta_j \leq l_m$ , где величины  $l_m = \sum p_i$ ,  $m = 1, 2, \dots, s$ , представляют собой границы интервалов, определяемых как

$$P_m = \sum p_i - \sum p_{i-1} .$$

Процедура моделирования испытаний рассматриваемого вида состоит в выборе значений  $\beta_j$  и сравнении их с величинами  $l_m$ . Исходом испытания является событие  $A_m$ , соответствующее номеру  $m$ -го интервала, в которое попало число  $\beta$ . Данную процедуру можно оптимизировать, проводя проверку не двух границ интервала  $l_{m-1} < \beta_j \leq l_m$ , а только одной. При этом проверку следует начинать с наименьшего по значению интервала, с его большей границы.

Рассмотрим пример. Пусть полную группу несовместных событий образуют события с вероятностями 0,3 0,5 0,2. Тогда для определения того, какое событие наступило, достаточно записать формулу

$$= \text{ЕСЛИ}(\text{ДСЧ} < 0,3; 1; \text{ЕСЛИ}(\text{ДСЧ} < 0,3 + 0,5; 2; 3))$$

где 1, 2, 3 – номера событий.

При использовании данного приема требуемое число проверок ЕСЛИ на 1 меньше, чем число событий в группе. Значение датчика определяется только адресом ячейки. Если в этой формуле использовать непосредственные

вызовы функции датчика, то в разных проверках могут быть использованы разные значения случайной величины, что недопустимо.

Моделирование сложных зависимых событий рекомендуется производить в следующей последовательности (для двух событий):

- если  $\beta_j \leq P_A$  и  $\beta_{j+1} \leq P_{B/A}$ , то фиксируется событие  $AB$ ;
- если  $\beta_j \leq P_A$  и  $\beta_{j+1} > P_{B/A}$ , то фиксируется событие  $\bar{A}\bar{B}$ ;
- если  $\beta_j > P_A$  и  $\beta_{j+1} \leq P_{B/A}$ , то фиксируется событие  $AB$ ;
- если  $\beta_j > P_A$  и  $\beta_{j+1} > P_{B/A}$ , то фиксируется событие  $\bar{A}\bar{B}$ .

В Excel программирование зависимых событий можно произвести по следующей схеме

= ЕСЛИ (И(ДСЧ <=  $P_A$ ; ДСЧ <=  $P_{B/A}$ ); АВ;  
 ЕСЛИ (И(ДСЧ <=  $P_A$ ; ДСЧ >  $P_{B/A}$ ); АнеВ;  
 ЕСЛИ (И(ДСЧ >  $P_A$ ; ДСЧ <=  $P_{B/A}$ ); неАВ; неАнеВ)))

ДСЧ задается адресом ячейки, содержащей случайное число, вероятности можно задавать как числовыми константами, так и адресами ячеек (последнее лучше, так как позволяет менять исходные данные без изменения формул). События можно задавать или числовыми, или текстовыми константами.

## 7. МОДЕЛИРОВАНИЕ МАРКОВСКИХ ЦЕПЕЙ

При моделировании марковских с помощью табличного процессора Excel необходимо разрешить ряд проблем, связанных с фиксацией состояния цепи и обеспечением цикличности при переходе от одного состояния цепи к другому. Задание переходных вероятностей и накопление статистики – операции достаточно простые.

Возможную схему моделирования рассмотрим на примере эргодической цепи с пятью состояниями. Переходные вероятности заданы табл. 6.1.

Таблица 7.1

Состояния	1	2	3	4	5
1	0,1	0,2	0	0,7	0
2	0	0,8	0,2	0	0
3	0,3	0,2	0,1	0,2	0,2
4	0,2	0,3	0,3	0	0,2
5	0,4	0,01	0	0	0,599

Вектор начальных вероятностей – (0,2; 0,1; 0,3; 0,3; 0,1). Для упрощения работы вначале можно детерминировано выделить любое состояние с ненулевой начальной вероятностью, например, первое.

Необходимо выделить ячейку для отображения номера текущего состояния цепи, пусть это будет  $A1$ . Для задания случайных чисел нужно выделить отдельную ячейку,  $B1$ . В  $B1$  записывается выражение датчика

случайных чисел, для первого примера используем простейший датчик =СЛЧИС().

В ячейках *A2..E6* поместим значения переходных вероятностей из табл. 2.

В процессе расчета цепи для текущего состояния необходимо на основе значения датчика случайных чисел и переходных вероятностей выбрать очередное состояние.

Возможный вариант для первого состояния :

=ЕСЛИ(D\$1=1;F2;ЕСЛИ(A\$1<>1;0;ЕСЛИ(B\$1<=A2;1;ЕСЛИ(B\$1<=A2+B2;2;ЕСЛИ(B\$1<=A2+B2+C2;3;ЕСЛИ(B\$1<=A2+B2+C2+D2;4;5))))))

для второго состояния :

=ЕСЛИ(D\$1=1;F3;ЕСЛИ(A\$1<>2;0;ЕСЛИ(B\$1<=A3;1;ЕСЛИ(B\$1<=A3+B3;2;ЕСЛИ(B\$1<=A3+B3+C3;3;ЕСЛИ(B\$1<=A3+B3+C3+D3;4;5))))))

и так далее.

В этих формулах проверяется, является ли текущим состояние, переходные вероятности для которого приведены в данной строке. Если нет, то в соответствующей ячейке столбца *F* записывается 0. Если же строка соответствует текущему состоянию, в ячейке формируется номер очередного состояния. После просмотра всех строк в ячейках *F2..F6* будут нули и число – номер очередного события. Теперь этот номер необходимо передать в ячейку *A1*. Однако ячейка не может одновременно задавать номер текущего события и принимать номер очередного. Эти действия необходимо разнести. Для этого введены две дополнительные ячейки:

*C1* содержащая 0 при формировании таблицы и 1 в процессе моделирования; *D1* с содержимым = ЕСЛИ(*C1*= 0;1; *D1*\*(-1)).

Тогда запись в *A1* формулы = ЕСЛИ(*D1*= 1;*A1*; МАКС(*F2:F6*)) обеспечивает поочередно фиксацию текущего состояния при просмотре строк переходных вероятностей (*D1* = 1) и пересылку номера очередного состояния (*D1* = -1).

Накопление статистики по количеству пребывания цепи в том или ином состоянии можно обеспечить через запись в ячейки *G2..G6* формул типа (для *G2*)

=ЕСЛИ(*C*\$1=0;0;ЕСЛИ(*F2*<>0;*G2*+1;*G2*)).

На основе интервала *G2..G6* можно построить диаграмму.

При подготовке таблицы необходимо записать в *C1* 0, затем внести в остальные ячейки формулы и константы, в меню “Сервис – Параметры - Вычисления” установить флаг “Итерации” и задать предельное число итераций равным 1. После этого записать в *C1* значение 1. Процесс моделирования запускается нажатием и удержанием клавиши *F9* (файл Случай\_вел.xls).

Общий вид модельной таблицы и результаты моделирования в виде гистограммы приведены в табл. 7.2 и на рис. 7.1. Файл модели МАРКОВ находится на компьютерах кл. 308.



Таблица 7.2

**МОДЕЛЬ**  
дискретной марковской цепи

	5	0,302	1	-1	0	0	Статистика
	0,1	0,2	0	0,7	0	0	378
	0,1	0,7	0,2	0	0	0	780
	0,3	0,2	0,1	0,2	0,2	0	302
	0,2	0,3	0,3	0	0,2	0	308
	0,4	0,01	0	0	0,599	1	301

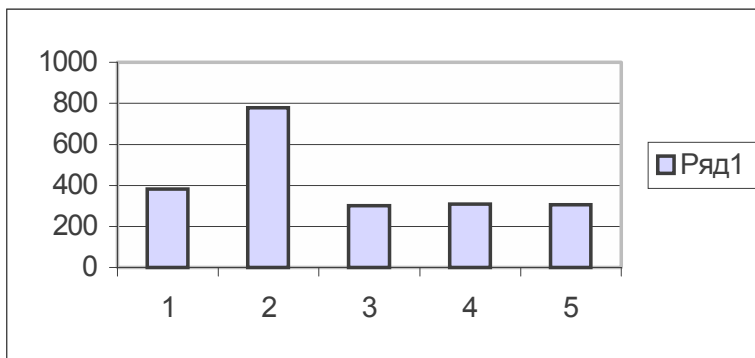


Рис. 7.1.

## 8. МОДЕЛИРОВАНИЕ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ

Практическое моделирование систем массового обслуживания предлагается освоить на примере трехканальной однофазной системы массового обслуживания без очередей. Такая модель соответствует, например АТС небольшого предприятия, имеющей до 100 абонентов и всего три выхода на городскую телефонную сеть. Структура модельной таблицы приведена в табл. 8.1.

Таблица 8.1

	A	B	C	D	E	F	G	H
1				<b>МОДЕЛЬ</b>				
2			трехканальной СМО с нулевой длиной очереди					
3								
4	дельта Т	Т модел	Т гран	№ шага	Лямбда	Мю	Сигма	Ключ
5	1	490	10000	490	0,05	10	5	-1
6								
7	Поток заявок			Каналы			Результаты	
8	флаг заявки	0		флаг освоб	1		обслужен.	33
9	кол-во	35		всего	3		потеряно	2
10	t заявки	490,8897		занято	1			
11				сумма зан	805			
12				загрузка	1,646217			
13								
14	Длит обслуживания			Вспомогательные				Захват
15	сумма	5,298713		магазин	507,1898	10000		0
16	Ню нулев	1,260031		времен	10000	507,1898		
17	Ню	16,30016		освобожд.	10000	10000		

Исходными данными для моделирования являются интенсивность потока заявок "лямбда" (количество заявок, поступающих в единицу времени), математическое ожидание длительности обслуживания "мю" и среднеквадратическое отклонение длительности обслуживания "сигма". Входной поток заявок описывается пуассоновским распределением, длительность обслуживания - нормальным. В табл 8.2 приведены расчетные соотношения для основных ячеек. Полная информация о расчетных соотношениях может быть получена при анализе файла СМО на компьютерах кл. 308. Группа ячеек B15..B17 обеспечивает получение случайных чисел, распределенных по нормальному закону с заданными параметрами. Ячейки E15..H17 обеспечивают сортировку каналов по времени их освобождения, что необходимо для моделирования ситуаций завершения обслуживания заявок.

Таблица 8.2

B10 ЕСЛИ(A5=0;0; ЕСЛИ(\$H\$5=1; ЕСЛИ(B5>B10;B10-1/E5\*LN(СЛЧИС()));B10);B10))

E8 ЕСЛИ(\$A\$5=0;0; ЕСЛИ(\$H\$5=1; ЕСЛИ(\$B\$5>E15;1;0);E8))

E10 ЕСЛИ(\$A\$5=0;0; ЕСЛИ(\$H\$5=1; ЕСЛИ(B8=1; ЕСЛИ(E9>E10;E10+1;E10); ЕСЛИ(E8=1;E10-1;E10));E10))

E11 ЕСЛИ(\$A\$5=0;0;ЕСЛИ(\$H\$5=1;(E11+E10);E11))

E12 ЕСЛИ(\$H\$5=1;E11/D5;E12)

H8 ЕСЛИ(\$A\$5=0;0;ЕСЛИ(\$H\$5=1;ЕСЛИ(E8=1;H8+1;H8);H8))

H9 ЕСЛИ(\$A\$5=0;0;ЕСЛИ(\$H\$5=1;ЕСЛИ(И(B8=1;E8=0;E10=E9);H9+1;H9);H9))

## 9. СТРУКТУРА ИНФОРМАЦИОННО – ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Информационно – вычислительная система (далее - информационная система) – это совокупность аппаратных средств и программного обеспечения, предназначенная для информационного обслуживания пользователей – операторов и технических объектов. Существует и другое название – система обработки данных.

Функции ИС состоят в выполнении требуемых действий обработки данных: ввода, хранения, преобразования, передачи и вывода. ИС может быть представлять собой одиночный компьютер с необходимым набором устройств или соответствующую вычислительную сеть. Пример фрагмента ИС (компьютер, подключенный к локальной и глобальной сетям) приведен на рис. 1.

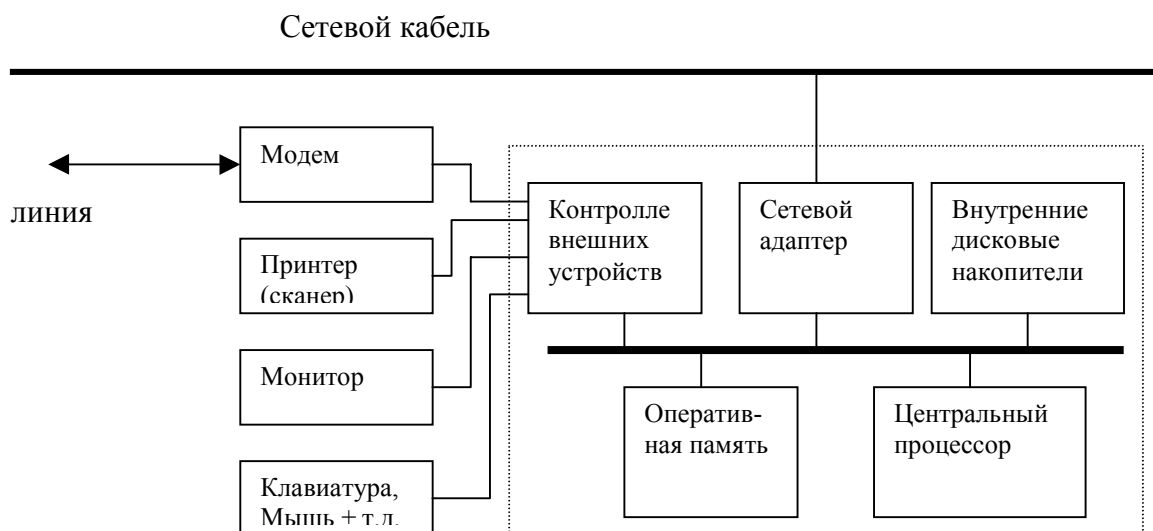


Рис. 1.

Различные структуры компьютеров и сетей (дать обзор: ОКОД, МКОД, ОКМД, МКМД, иерархическая, линейная, кольцевая, звездообразная).

Математической формой представления структуры ИС являются графы, вершины которых соответствуют элементам системы, а ребра (дуги) – связям между элементами. Инженерной формой представления систем являются схемы. На рис.1 показана именно схема. Структура ИС дает общее представление о составе технических средств и связей между ними. Дополнительные сведения даются в форме спецификаций, где для каждого элемента и каждого типа связей указываются наименование, тип устройства, технические характеристики.

Структура сложных систем при представлении ее на уровне устройств может оказаться настолько сложной, что теряет обзримость и выходит за рамки возможностей методов исследования, используемых при анализе и синтезе систем. В таких случаях структура описывается на более высоком уровне, когда в качестве элементов выступают отдельные компьютеры, комплексы и подсистемы.

В связи с обработкой данных технические средства рассматриваются как ресурсы двух типов: устройства и память.

Устройство – ресурс, используемый для преобразования и ввода – вывода данных, разделяемый между процессами во времени. В каждый момент времени устройство может использоваться одним процессом, реализуя соответствующие операции преобразования или ввода – вывода данных. Основная характеристика устройства – производительность.

Память – ресурс, используемый для хранения данных и разделяемый между процессами по объему и по времени. Основные характеристики памяти – емкость и быстродействие.

## Понятие процесса

Использование ресурсов в ИС осуществляется в соответствии с принципом программного управления. Общая схема самоподдерживаемой совокупности процессов в системе приведена на рис. 2.

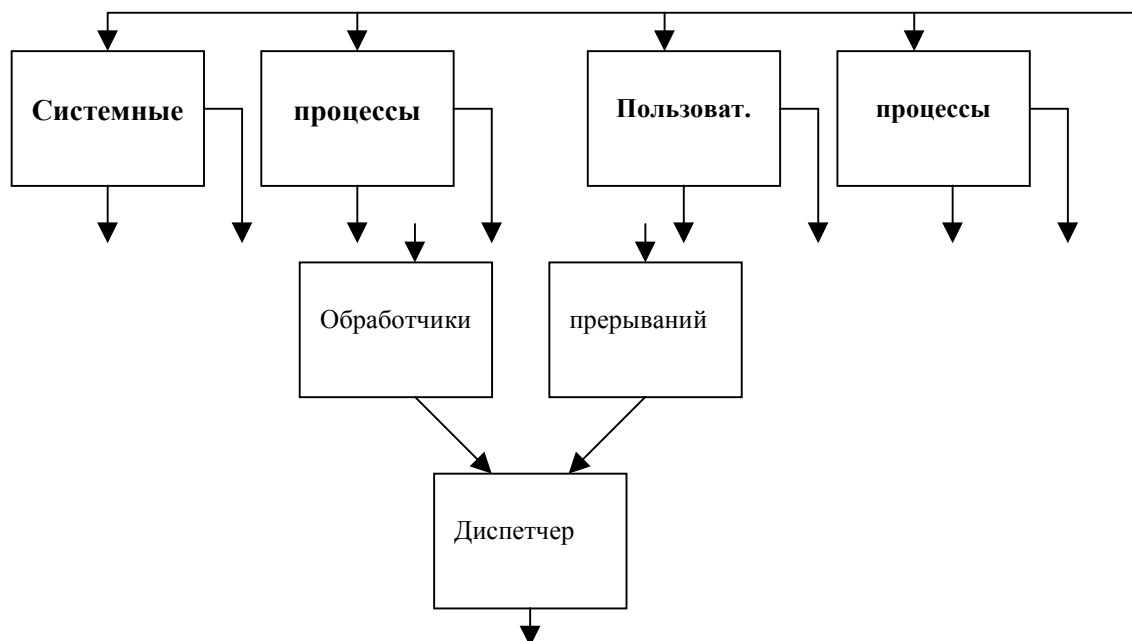


Рис. 2.

Процессом называется идентифицируемое выполнение программного кода, реализующее целенаправленный акт обработки данных. Таким образом, задача может рассматриваться как некоторая совокупность пользовательских процессов. Процессы создаются, управляются и удаляются специальными системными процессами по запросам других процессов. Такой запрос может исходить от программы или пользователя, и связан с соответствующими данными. Если запрос исходит от пользователя, то в качестве данных может быть, например, текст команды на запуск определенной программы.

Процесс может находиться в различных состояниях: работы, ожидания процессора в состоянии готовности, ожидания другого ресурса, останова, запуска, удаления. Процесс может завершиться естественным путем (выход по стрелкам вниз), а может быть остановленным по прерыванию, инициированному в рамках этого или другого процесса.

Прерывание – чисто аппаратный переход к выполнению специальных программ – обработчиков прерываний, которые анализируют складывающуюся в системе ситуацию и вносят изменения во внутренние управляющие данные (таблицы) системы. После этого управление процессором получает программа – диспетчер, которая определяет, какому из процессов передать управление в данный момент времени. И обработчики прерываний, и диспетчер являются системными процессами, их выделение обусловлено их функциями в управлении вычислительным процессом.

Обычно процесс описывается в специальной таблице, используемой системными процессами, следующим набором данных: имя источника, инициировавшего процесс, имя пользователя, время инициирования, режим обработки, приоритет. Приоритет процесса может динамически изменяться вследствие изменения общей ситуации в системе. Сравнивая приоритеты различных процессов, диспетчер выбирает процесс для передачи ему управления. В ряде случаев может создаваться трасса процесса – запись последовательности событий, связанных с изменением состояния процесса. При этом каждое событие связывается с моментом возникновения, программой, реализующей процесс, и ресурсами, обслуживающими процесс.

В информационных системах очень часто запрос оператора порождает цепочку (название упрощенное, на самом деле структура связей напоминает “вытянутую сеть”) взаимосвязанных по данным процессов в нескольких компьютерах. При анализе сложных информационных систем запросы операторов часто называют транзакциями, подчеркивая факт создания взаимосвязанных по данным, времени и/или памяти процессов в различных узлах сети.

Порождение и функционирование цепочек процессов связано с взаимодействием процессов. Взаимодействие процессов осуществляется посредством передачи и приема сообщений – последовательностей данных, имеющих законченное смысловое значение. Ввод сообщений в процесс и вывод сообщений из процесса производится через логические (программно-организованные) точки, называемые портами. Порты подразделяются на входные и выходные. Таким образом, процесс как объект представляется совокупностью портов, через которые он взаимодействует с другими процессами сети. Промежуток времени, в течение которого взаимодействуют процессы, называется сеансом или сессией. На одном и том же компьютере время сеанса на несколько порядков меньше, чем время выполнения транзакции, и им пренебрегают. При работе в сети время сеанса взаимодействия процессов на разных компьютерах становится существенным, часто оно становится определяющим при оценке времени выполнения транзакций.

### **Рабочая нагрузка**

Рабочая нагрузка. Процесс функционирования ИС существенно зависит от состава задач и исходных данных, поступающих на ее вход. Весь объем поступающей информации принято называть рабочей нагрузкой. При проектировании и эксплуатации системы наибольший интерес представляет потребность заданий в ресурсах: оперативной и дисковой памяти, процессорном времени, устройствах ввода – вывода. Поэтому рабочую нагрузку, относящуюся к промежутку времени  $T$ , можно определить в виде множества характеристик заданий

$$L = \{l_1 \dots l_K\}; \quad l_i = \{A_i, \theta_{i1} \dots \theta_{iN}\},$$

где  $l_i$  - описание  $i$ -го задания, устанавливающее его атрибуты  $A_i$  и потребности задания  $\theta_{i1} \dots \theta_{iN}$  в ресурсах  $1 \dots N$ .

Такой прием реален для компьютера оператора, но уже для сервера локальной сети такое описание может оказаться громоздким. Для представления рабочей нагрузки в компактной форме потребности заданий в ресурсах характеризуются в среднестатистическими значениями ресурсов, приходящимися на одно задание. Рабочая нагрузка зависит от сферы применения ИС и оказывается различной для систем, оперирующих с разными классами задач: инженерно-техническими, экономическими, учетно-статистическими и т.п.

## **10. ПОКАЗАТЕЛИ ЭФФЕКТИВНОСТИ ИНФОРМАЦИОННЫХ СИСТЕМ**

Степень соответствия системы своему назначению называется эффективностью (качеством) системы. Для сложных систем, какими являются информационные системы, эффективность не удастся определить одной величиной, и поэтому ее представляют набором характеристик. Основными характеристиками ИС являются производительность, время ответа, надежность и стоимость. Кроме этих общепринятых характеристик, значения которых можно выразить количественно, могут использоваться другие показатели, отображающие специфические требования пользователей.

Характеристики зависят от организации системы – структуры, параметров оборудования, состава программного обеспечения, режима функционирования.

Производительность – характеристика вычислительной мощности системы, определяющая количество вычислительной работы, выполняемой системой за единицу времени.

Время ответа – длительность промежутка времени от момента поступления задания в систему до момента окончания его выполнения (время выполнения транзакции). Кроме этой характеристики используют время отклика – длительность временного промежутка между вводом оператора и выдачей ему системой некоторых промежуточных данных. По времени отклика можно судить о реактивности системы, о ее способности быстро реагировать на действия оператора.

Надежность – свойство системы выполнять возложенные на нее функции в заданных условиях функционирования с заданными показателями качества: достоверностью результатов, пропускной способностью, временем ответа и т.п..

Стоимость информационной системы – это суммарная стоимость аппаратных и программных средств (включая расходы на обслуживание и ремонт), а также накопленных в системе данных.

## 11. ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ КОМПОНЕНТОВ СИСТЕМЫ

Производительность – характеристика вычислительной мощности системы, определяющая количество вычислительной работы, выполняемой системой за единицу времени. Производительность вычислительных систем общего назначения оценивается в зависимости от области применения номинальной, комплексной, системной производительностью и производительностью на рабочей нагрузке.

Она может определяться интегрально для всей системы и дифференцировано по отдельным компонентам.

Аппаратурные средства ИС обладают производительностью вне связи с операционной системой, прикладным программным обеспечением и режимом эксплуатации системы. Эта производительность оценивается быстродействием – числом операций, выполняемых процессором, памятью или иным устройством в секунду. Совокупность значений быстродействия устройств ( $V_1 \dots V_N$ ), образующих аппаратный костяк системы, характеризует номинальную производительность системы. Чтобы оценка номинальной производительности была по возможности простой, стремятся уменьшить число устройств, участвующих в оценке. Для этого, во-первых, суммируют быстродействие устройств, выполняющих одинаковые операции, и, во-вторых, отбрасывают оценки быстродействия устройств, менее влияющих на интегральную производительность системы. Номинальная производительность характеризует только потенциальные возможности устройств, которые не могут быть использованы полностью. Этому препятствует влияние взаимосвязей между устройствами, что проявляется в ожиданиях завершения предшествующих операций, выполняемых другими устройствами.

Показателем использования устройства в процессе работы системы является загрузка. Загрузка  $i$ -го устройства определяется отношением  $p_i = T_i / T$ , где  $T_i$  – время, в течение которого устройство работало, и  $T$  – продолжительность работы системы. В течение времени  $T - T_i$  устройство простаивает. Совокупность значений  $p_1 V_1 \dots p_N V_N$  характеризует производительность технических средств с учетом простоев, возникающих в процессе функционирования системы. На загрузку устройств существенно влияет режим обработки задач, реализуемый управляющими программами операционной системы. Влияние ОС проявляется, например, в следующем. Организация ввода – вывода связана с использованием процессора и запоминающих устройств для промежуточного хранения вводимых или выводимых наборов данных. В результате этого часть времени процессора и емкости запоминающих устройств тратится на обслуживание ввода – вывода. Такая же ситуация возникает при организации в системе виртуальной памяти, режима разделения времени и обеспечения других вспомогательных функций.

Общепринятым способом оценки реальных характеристик отдельных устройств (процессоров, накопителей на дисках и т.п.) является использование смесей команд. Например, производительность устройства по Гиббсону оценивается как

$$p = \sum_{i=1}^K \sum_{j=1}^{L_i} t_i$$

где  $t_i$  - время выполнения устройством  $i$ -ой операции;

$K$  - число типов операций в смеси;

$L_i$  - количество повторений операции данного типа в смеси.

В рамках этого способа оценка производительности зависит от набора операций и их удельного веса в смеси. В мировой практике для сравнительной оценки процессоров используются различные смеси, в ряде случаев применяют различные смеси, взятые с различными весами.

## 12. ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМЫ В ЦЕЛОМ

Интегрально ИС может оцениваться системной производительностью или использованием смеси операций.

Системная производительность наиболее просто оценивается числом задач, решаемых системой за единицу времени. Эта оценка информативна только для конкретной области применения ИС или в случае специализированного, целевого использования системы. Например, сервер, обеспечивающий работу операторов – билетных кассиров, может оцениваться числом запросов, которые могут быть обработаны за час работы системы при максимальной нагрузке со стороны кассиров. Обычно задачи поступают на обработку в случайные моменты времени, и время пребывания задач в системе зависит от набора задач, одновременно обрабатываемых системой. В результате этого число задач  $n$ , обработанных системой за время  $T$ , - случайная величина, и производительность  $\lambda$  в интервале  $T$  оценивается с погрешностью, имеющей статистическую природу и зависящей от случайной величины  $n$  и ее дисперсии. С увеличением длительности интервала  $T$  значение  $n$  возрастает, и погрешность оценки  $\lambda$  стремится к нулю при  $T \rightarrow \infty$ .

$$\tau = \frac{1}{N} \sum_{i=1}^N \tau_i$$

Другой способ определения производительности  $\lambda$  - через среднее значение интервала между моментами окончания обработки задач. В этом случае в течение времени  $T$  регистрируются интервалы между моментами завершения обработки задач  $\tau_1 \dots \tau_N$ . Среднее значение этого интервала определяется интенсивностью выходного потока задач, и производительность системы  $\lambda = 1/\tau$ .

Рассмотрим зависимость между средним числом задач, поступающих на вход системы в единицу времени (интенсивность входного потока задач  $\Lambda$ ) и средним числом задач, покидающих систему за единицу времени (интенсивность выходного потока задач  $\lambda$ ) (рис. 1).



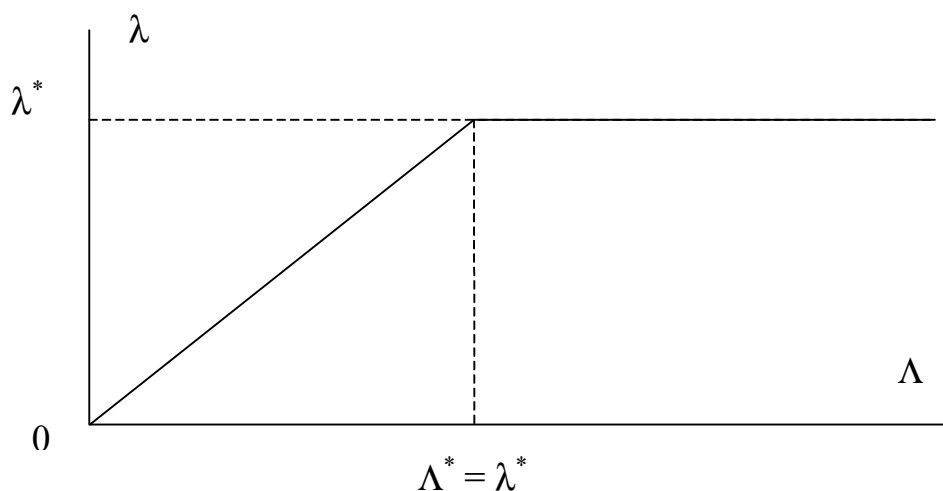


Рис. 1

В области  $0 \leq \Lambda \leq \Lambda^*$  интенсивность выходного потока полностью определяется интенсивностью входного потока:  $\lambda = \Lambda$ . При  $\Lambda > \Lambda^*$  система из-за ограниченности ресурсов – числа и быстродействия устройств, а также емкости памяти – не может в течение единицы времени обслужить все поступившие задания. Интенсивность выходного потока  $\lambda$ , достигнув предельного значения  $\lambda^*$ , при дальнейшем увеличении интенсивности входного потока не увеличивается. Значение  $\lambda^*$  определяет максимальную производительность системы для заданного класса задач и является характеристикой самой системы, не зависящей от интенсивности входного потока задач.

### Определение времени ответа

Время ответа – длительность промежутка времени от момента поступления задания в систему до момента окончания его выполнения (время выполнения транзакции). Кроме этой характеристики используют время отклика – длительность временного промежутка между вводом оператора и выдачей ему системой некоторых промежуточных данных. По времени отклика можно судить о реактивности системы, о ее способности быстро реагировать на действия оператора.

В общем случае время ответа – случайная величина, что обусловлено следующими факторами:

- 1) влиянием исходных данных на число операций ввода, обработки и вывода данных и непредсказуемостью значений исходных данных;
- 2) влиянием состава смеси задач, одновременно находящихся в системе, и непредсказуемостью состава смеси из-за случайности момента поступления задач на обработку.

Время ответа как случайная величина наиболее полно характеризуется функцией распределения  $P(u < x)$  или функцией плотности вероятностей  $p(u)$ .

Чаще всего время ответа оценивается средним значением, которое определяется как статистическое среднее случайной величины  $u_i, i = 1, \dots, n$ , наблюдаемой для достаточного количества задач :

$$U = \frac{1}{n} \sum_{i=1}^n u_i$$

Время ответа складывается из двух составляющих: времени выполнения задачи и времени ожидания. Время выполнения задачи при отсутствии параллельных процессов равно суммарной длительности всех этапов процесса – ввода, обращений к дисковой памяти, процессорной обработки и вывода.

Время выполнения в данном случае зависит от требований к ресурсам,  $\theta_{i1} \dots \theta_{iN}$  и быстродействия  $V_1 \dots V_N$  устройств:

$$\vartheta = \sum_{i=1}^N \frac{\theta_i}{V_i}$$

Время ожидания – сумма промежутков времени, в течение которых задача находилась в состоянии ожидания требуемых ресурсов. Ожидание возникает при мультипрограммной обработке, когда ресурс, необходимый задаче, занят другой задачей. Время ожидания зависит в первую очередь от режима обработки и интенсивности входного потока задач.

Таким образом, время ответа зависит от тех же параметров, что и производительность: структуры и характеристик технических средств, режима обработки и характеристик задач. Зависимость среднего времени ответа  $U$  от интенсивности входного потока задач  $\Lambda$  приведена на рис.4.

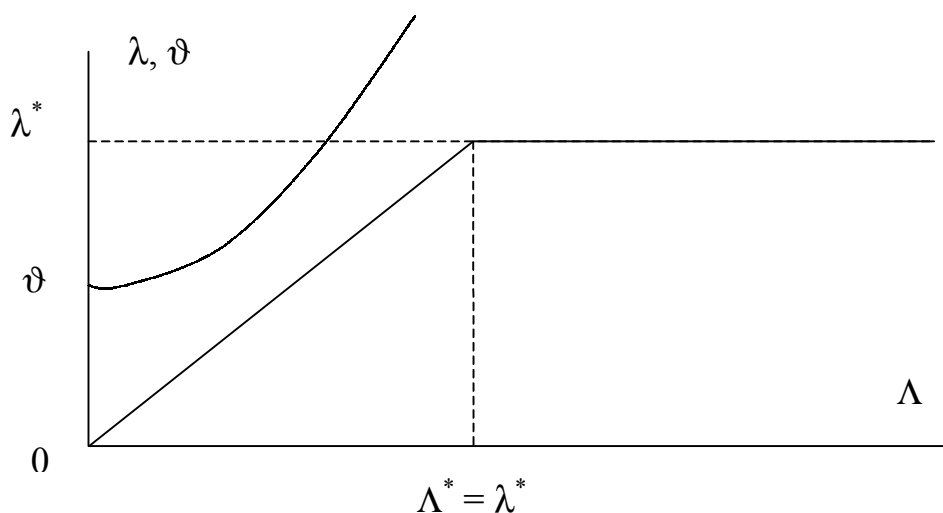


Рис. 4

При  $\Lambda \rightarrow 0$  время ответа  $U \rightarrow \vartheta$ . С увеличением интенсивности входного потока среднее время ответа монотонно возрастает и может принимать сколь угодно большие значения, если интенсивность входного потока  $\Lambda$  пре-

вышает производительность системы  $\lambda^*$  в течение сколь угодно большого периода времени.

Среднее время ответа характеризует быстроту реакции системы на входные воздействия: задания, запросы корреспондентов и т.п. Качество системы тем выше, чем меньше среднее время ответа.

### 13. ВЛИЯНИЕ РЕЖИМА ОБРАБОТКИ ДАННЫХ

Режим обработки данных – это способ выполнения задач, характеризующийся порядком распределения ресурсов между задачами. Требуемый режим обработки данных обеспечивается управляющей программой операционной системы. В первом приближении следует рассмотреть однопрограммный и многопрограммный режимы (существуют различные разновидности многопрограммного режима). Однопрограммный режим в настоящее время применяется в контроллерах (специализированных процессорах, обеспечивающих работу исполнительных механизмов внешних устройств) и, ограниченно – на персональных компьютерах под управлением операционных систем типа MS-DOS вне сети. Основным режимом работы компьютеров, в том числе и персональных – мультипрограммная обработка.

Мультипрограммная обработка – режим, при котором на компьютере выполняется одновременно несколько задач. При этом процессы обработки, относящиеся к разным задачам, одновременно выполняются различными устройствами системы, способными функционировать параллельно. Цель мультипрограммирования – увеличение производительности системы.

Число задач, находящихся в системе, называется уровнем мультипрограммирования. Уровень мультипрограммирования влияет на производительность и время ответы системы следующим образом (Рис.5).

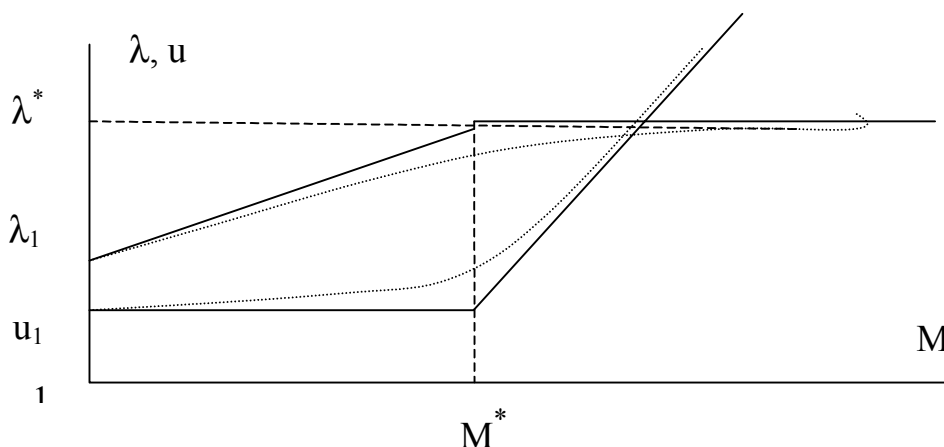


Рис. 5

С увеличением уровня мультипрограммирования  $M$  от 1 увеличивается вероятность того, что большее число устройств занято выполнением задач. Вместе с тем вероятность того, что несколько задач одновременно обращаются к одному устройству, достаточно мала, и поэтому время

ожидания оказывается незначительным. Однако при уровне мультипрограммирования  $M = M^*$  возникает ситуация, когда по крайней мере одно устройство оказывается полностью загруженным. Дальнейшее увеличение числа задач не приводит к росту производительности  $\lambda$ , которая определяется производительностью  $\lambda^*$  этого устройства, но при  $M > M^*$  начинает резко возрастать время ответа  $U$ , поскольку все большее число задач ожидает момента освобождения устройств.

Значение  $M^*$  называется точкой насыщения мультипрограммной смеси (точкой насыщения системы) и зависит от числа устройств, которые в составе системы могут функционировать параллельно. Чем больше таких устройств, тем больше может быть  $M^*$ . Кроме того, на значение  $M^*$  существенно влияют свойства задач. Если задачи преимущественно используют одно устройство, то значение  $M^*$  невелико, близко к 1. Работа системы при уровне мультипрограммирования  $M > M^*$  неэффективна, поскольку нет выигрыша в производительности и увеличивается время ответа.

Производительность  $\lambda$  и среднее время ответа  $U$  связаны между собой зависимостью  $\lambda = M/U$ , которая называется формулой Литтла и является фундаментальным законом теории массового обслуживания.

$$m = \sum_{i=1}^N \rho_i$$

В системе, состоящей из  $N$  устройств, загрузка которых равна  $\rho_1, \dots, \rho_N$ , среднее число одновременно выполняемых задач равно суммарной загрузке устройств. Остальные  $M - m$  задач находятся в состоянии ожидания.

Число одновременно выполняемых задач  $m$  называется коэффициентом мультипрограммирования и равно отношению производительности системы в мультипрограммном режиме  $\lambda$  к производительности в однопрограммном режиме  $\lambda_i = 1/\theta$ , если затраты ресурсов на организацию мультипрограммирования возрастают пропорционально числу одновременно выполняемых задач, то есть  $M = \lambda / \lambda_i$ .

Таким образом, коэффициент мультипрограммирования  $m$  является показателем увеличения производительности системы за счет мультипрограммирования, при этом  $1 \leq m \leq N$ , где  $N$  – число устройств системы, способных функционировать параллельно с каждым из  $N - 1$  устройств.

Следует отметить, что при разработке компьютеров принимаются меры к повышению  $N$ , для этого устройства дисковой памяти и устройства ввода – вывода оснащают «интеллектуальными» контроллерами с достаточным объемом внутренней памяти.

## 14. ХАРАКТЕРИСТИКИ НАДЕЖНОСТИ

Надежность – свойство системы выполнять возложенные на нее функции в заданных условиях функционирования с заданными показателями качества: достоверностью результатов, пропускной способностью, временем

ответа и т.п. Работоспособность системы или ее отдельных частей нарушается из-за отказов аппаратуры – выхода из строя элементов или соединений, а также из-за ошибок в программном обеспечении. Так как природа ошибок в аппаратуре и программах различна, рассмотрим отказы из-за аппаратуры и из-за программ отдельно. При этом вначале будем полагать, что все отказы – из-за аппаратуры.

Важнейшей характеристикой надежности является интенсивность отказов, обычно измеряемая в числе отказов за час или за сутки. Интенсивность отказов зависит от числа элементов и соединений. Если любой отказ носит катастрофический характер, то есть приводит к полной потере работоспособности системы, то интенсивность отказов в системе :

$$\lambda_0 = \sum_{i=1}^n \lambda_i$$

где  $\lambda_i$  - интенсивность отказов  $i$  – го элемента или соединения;  
 $n$  - число элементов и соединений в системе.

Средний промежуток времени между двумя смежными отказами называется средней наработкой на отказ и равен  $T_0 = 1/\lambda_0$ .

Промежутки времени между отказами – случайные величины со средним значением  $T_0$ , которые, как правило, распределены по экспоненциальному закону. При этом вероятность того, что за время  $t$  произойдет отказ,  $P(t < x) = 1 - e^{-t/T}$ .

Например, если наработка компьютера на отказ составляет 1000 ч, то вероятность того, что компьютер выйдет из строя за 100 ч работы составляет 0,095; а после 500 ч работы эта вероятность будет оцениваться уже в 0,4.

Работоспособность системы, нарушенная в результате отказа, восстанавливается путем ремонта системы. Ремонт состоит в выявлении причины нарушения работоспособности – диагностике системы и в восстановлении работоспособности путем замены неисправного элемента, восстановления потерянного контакта или устранения ненужного. Промежуток времени, затрачиваемый на восстановление работоспособности системы, называется временем восстановления. Его длительность зависит от сложности системы, степени совершенства средств диагностики и уровня ремонтпригодности системы. Время восстановления – случайная величина, характеризующаяся средним значением  $T_B$  – средним временем восстановления.

С учетом средней наработки на отказ  $T_0$  и среднего времени восстановления  $T_B$  надежность системы характеризуется коэффициентом готовности  $K_T = T_0 / (T_0 + T_B)$ , определяющим долю времени, в течение которого система работоспособна.

Естественно, что приведенный аппарат оценки справедлив лишь статистически, на большом количестве реализаций. Однако он вполне приемлем и для предварительных оценок по одиночным компьютерам или компьютерным классам.

Надежность системы может повышаться различными путями, наиболее простой – резервирование элементов. Однако этот путь приводит к существенному увеличению стоимости и применяется главным образом в ответственных случаях, в качестве примера можно привести так называемые “зеркальные” накопители серверов в серьезных организациях.

### **Надежность функционирования программ**

Надежность систем определяется в основном двумя факторами: надежностью компонент и ошибками в конструкции. Относительная невысокая надежность аппаратных элементов, их глубокая взаимозависимость и способность к разрушению, старению привели к тому, что этот фактор стал определяющим при оценке надежности аппаратуры.

Надежность программных комплексов определяется теми же факторами, однако их соотношение иное. Хранение программ на магнитных носителях при выполнении элементарных правил характеризуется очень высокой надежностью. Поэтому при оценке надежности программ доминирующим становится второй фактор.

Программа любой сложности при строго фиксированных исходных данных и надежной аппаратуре исполняется по однозначно определенному маршруту и дает на выходе строго определенный результат. Однако большое множество вариантов исходных данных и большое множество условных переходов в программе создают огромное количество различных маршрутов исполнения программного комплекса. Такое количество вариантов исполнения программы нельзя проверить полностью из-за ограничений по длительности отладки и приемочных испытаний. Источниками ненадежности программы являются непроверенные сочетания данных, при которых программа дает неверные результаты или некорректно прекращает свою работу.

Отказ при использовании программ может проявиться как следствие:

- нарушения кода программы в памяти; искажения данных ;
- нарушение нормального хода вычислительного процесса.

Сбой при использовании программ трактуется как самоустраняющийся отказ, не требующий внешнего вмешательства. В процессе обработки данных при сбоях и отказах чаще всего отсутствует физическое разрушение программ и не требуется замена или ремонт каких-либо материальных компонент. Применительно к аппаратуре понятия отказ и сбой отличаются степенью физического разрушения компонент и необходимостью их замены. Применительно к программам сбои и отказы разделяются по временному показателю – длительности восстановления после любого искажения программы, данных или вычислительного процесса. При длительности восстановления, меньшей заданного порога, аномалии при функционировании программ относят к сбоям. При восстановлении, превышающем по длительности пороговое значение, искажения соответствуют отказу.

### **Правильный и надежный комплекс программ**

Понятие корректной программы может рассматриваться статистически вне временного функционирования. Правильность программы не определена

вне данных, заданных спецификацией и не зависит от динамики функционирования. Неправильность программы определяется вероятностью совмещения событий:

- 1) попадания данных в область, заданную требованиями спецификации, но непроверенную при отладке и испытаниях;
- 2) появление ошибки в программе при обработке таких данных.

### **Надежная программа**

Быстрое реагирование на искажения программ, данных или вычислительного процесса и восстановление работоспособности за время меньше, чем порог между сбоем и отказом, позволяют обеспечить высокую надежность программы. Может случиться и так, что абсолютно надежно будет функционировать неправильная программа. В реальных условиях исходные данные могут попадать в область, не соответствующую требованиям спецификаций и не проверенную при отладке. Если восстановление происходит за время меньше порогового, то такие события не влияют на основной показатель надежности – наработку на отказ. Следовательно, отказ программы является понятием динамическим. Он происходит с вероятностью, определяющейся совмещением следующих событий:

- появление данных, попадающих в непроверенные при тестировании и испытаниях области;
- проявление ошибки при обработке этих данных, достаточной для вызова отказовой ситуации;
- превышение длительности восстановления после возникновения ситуации порогового значения.

В области данных, определенной спецификациями, правильность и надежность программ оказываются взаимосвязанными.

### **Восстановление**

Отсутствие физического разрушения позволяет добиться высокой автоматизации программного восстановления. Главной задачей становится восстановление за время, меньше порогового. Это позволяет преобразовывать отказы в сбои, что улучшает общие показатели функционирования вычислительной системы. Для решения этой задачи программный комплекс должен иметь средства, позволяющие:

- проводить систематический контроль и обнаруживать аномалии процесса;
- диагностировать обнаруженные искажения;
- выбирать методы и средства оперативного восстановления (рестарта);
- регистрировать сбои и отказы для выявления систематических случаев, требующих доработки программ или аппаратуры.

Реализация средств с такими функциями осуществляется за счет введения избыточности в программы, данные и процесс функционирования. Избыточность может быть:

- программной, включающей программные компоненты, предназначенные для контроля, обнаружения, диагностики и восстановления работоспособности вычислительного комплекса;

- информационной, заключающейся в дублированном хранении данных и средств кодовой защиты информации;
- временной, состоящей в выделении необходимых резервов процессорного времени на выполнение программ, обеспечивающих оперативный контроль и восстановление правильного функционирования комплекса.

Перечисленные виды избыточности используются совместно и требуют некоторых ресурсов по объему памяти и процессорного времени. Доля этих ресурсов обычно находится в пределах 5..10% от максимальных значений. Однако даже при таких относительно небольших затратах надежность функционирования программ (наработка на отказ) возрастает на один – два порядка.

### Критерии надежности программ

Используются: наработка на отказ и коэффициент готовности. Эти критерии применительно к программному обеспечению имеют тот же смысл, что и по отношению к аппаратуре. Однако со временем значение данных показателей программного обеспечения меняется по другому закону, нежели у аппаратуры (Рис.1). Это определяется тем, что со временем аппаратура стареет, а программное обеспечение освобождается от части ошибок.

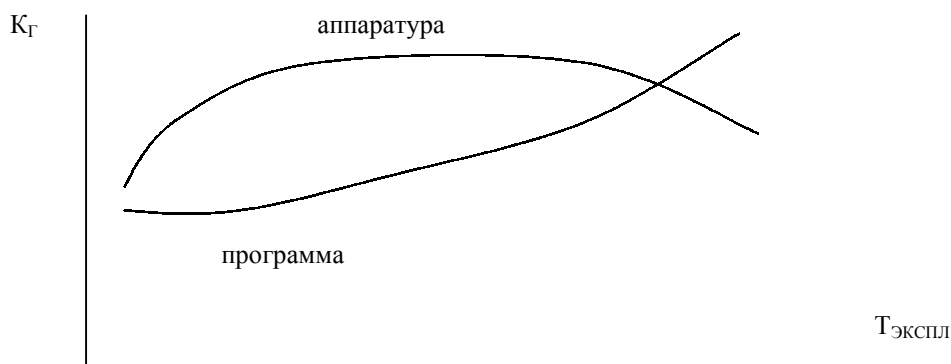


Рис.1

## 15. ПОСТРОЕНИЕ МАТЕМАТИЧЕСКОЙ МОДЕЛИ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Построим аналитическую модель для информационной системы на базе локальной сети типа "файл-сервер". В основу положим отождествление функционирования системы с простым непрерывным марковским процессом.

Процесс взаимодействия пользователей и системы складывается из отдельных повторяющихся этапов, каждый из которых делится на системную и пультовую фазы.

Запрос пользователя находится в системной фазе, если в системе имеется вся необходимая информация для обработки запрошенной программы. В этой фазе производится обработка запрошенной программы, при этом время сервера делится квантами между всеми пользовательскими программами, на-



ходящимися в системной фазе. Если для дальнейшего выполнения программы необходим ввод с пульта новой входной информации, взаимодействие переходит в пультовую фазу.

В пультовой фазе пользователь наблюдает за выводом информации на его терминал, обдумывает свою реакцию на выведенную информацию и вводит с пульта новую входную информацию, что следует рассматривать как ввод в систему нового запроса на обслуживание.

Таким образом, процесс взаимодействия пользователя и системы может находиться в одном из двух состояний: либо система имеет программу, которую она должна выполнять для пользователя, а пользователь ждет ответа системы на свой запрос (системная фаза), либо система фактически не имеет программы, которую могла бы дальше выполнять для данного пользователя, и ждет сообщения от пользователя (пультовая фаза).

Примем, что случайные величины - время работы пользователя за пультом (т.е. продолжительность пультовой фазы)  $\tau_n$  и продолжительность чистого времени обработки запроса (без учета времени пребывания в очереди к процессору)  $t_{об}$  распределены по экспоненциальному закону со средними значениями соответственно  $\tau_{n, cp}$  и  $t_{об, cp}$ .

Функции плотностей для случайных величин  $\tau_n$  и  $t_{об}$  имеют вид :

$$f_n(t) = \frac{1}{\tau_{n, cp}} e^{-t/\tau_{n, cp}} \quad (t \geq 0)$$

$$f_n(t) = \frac{1}{\tau_{об, cp}} e^{-t/\tau_{об, cp}} \quad (t \geq 0)$$

В системах без совмещения вычислений с обменом информацией с внешним ЗУ время обмена должно включаться в  $t_{об}$ .

Предполагаем, что переключение программ происходит мгновенно и издержки, т.е. потери времени отсутствуют.

Модель строится с целью определения основных характеристик системы : среднего числа пользователей, ожидающих ответа, и средней продолжительности ожидания ответа на запрос пользователя.

Примем, что состояние системы определяется числом запросов  $j$ , находящихся в данный момент в системной фазе. Тогда система с  $n$  пользователями имеет  $(n + 1)$  состояний ( $0 \leq j \leq n$ ). Считаем, что вероятность перехода из состояния  $j$  в  $i$  зависит только от самих состояний  $j$  и  $i$  и не зависит от предыстории, приведшей систему в состояние  $j$ . Сами переходы совершаются только через случайные промежутки времени.

При принятых допущениях, в том числе относительно законов распределения для  $\tau_n$  и  $t_{об}$ , функционирование системы может быть представлено простым непрерывным (по времени) марковским процессом.

Пусть в данный момент времени  $j$  запросов находятся в системной фазе, т.е. ожидают обслуживания, а соответственно  $(n - j)$  запросов находятся в пультовой фазе. Тогда число запросов, переходящих в системную фазу в единицу времени, или, иначе говоря, интенсивность (скорость) перехода из состояния с  $j$  запросами в состояние с  $j + 1$  запросами в системной фазе (ожидающими обслуживания) будет :

$$a_{j(j+1)} = \frac{n - j}{\tau_{n.cp}} \quad (0 \leq j \leq n - 1)$$

Вместе с тем происходят переходы запросов из системной в пультовую фазу. Если бы каждому из  $j$  запросов, находящихся в системной фазе, было выделено все время работы сервера, то интенсивность перехода запросов из системной в пультовую фазу составила бы  $j / t_{об, cp}$ . Однако на самом деле время сервера делится поровну (квантами) между всеми  $j$  пользователями и поэтому интенсивность перехода системы из состояния с  $j$  пользователями в состояние с  $j-1$  пользователями :

$$a_{j(j-1)} = \frac{1}{t_{об.cп}} \quad (1 \leq j \leq n)$$

Функционирование рассматриваемой системы разделения времени может быть описано цепью Маркова, соответствующей графу состояний системы, представленному на рис. 15.1. На нем указаны интенсивности переходов системы из одних состояний в другие.

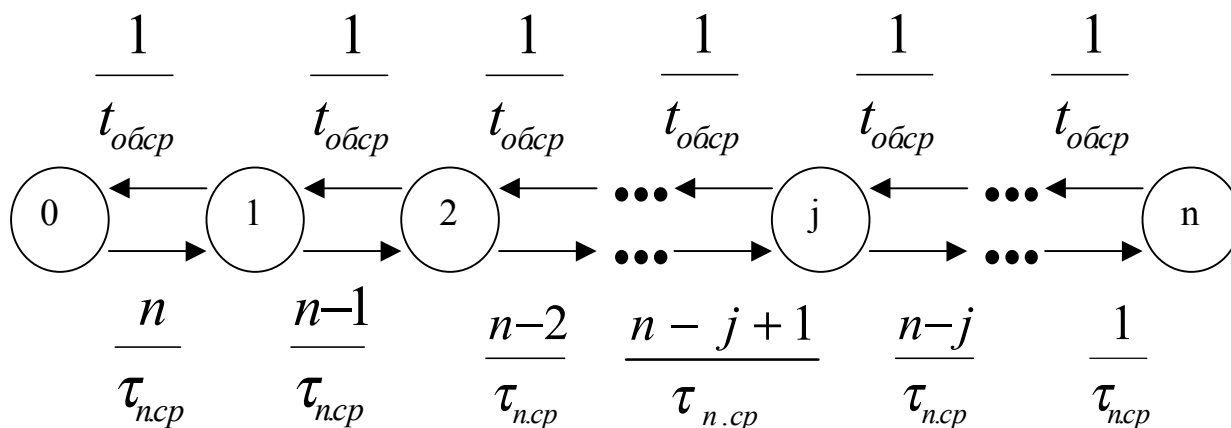


Рис.15.1

Обозначим через  $\Pi = \{\pi_0, \pi_1, \dots, \pi_n\}$  вектор предельных вероятностей состояний системы. Составляющие этого вектора  $\pi_j$  ( $0 \leq j \leq n$ ) есть вероятности пребывания системы в состоянии  $j$  в стационарном режиме, причем

$$\sum_{j=0}^n \pi_j = 1$$



$$\pi_0 = \frac{1}{\sum_{j=0}^n \frac{n}{(n-j)!} \chi^j}$$

Соответственно

$$\pi_i = \frac{\frac{n!}{(n-i)!} \chi^i}{\sum_{j=0}^n \frac{n}{(n-j)!} \chi^j}$$

Среднее количество запросов, ожидающих ответа :

$$L_{cp} = \sum_{i=0}^n i \pi_i = \frac{\sum_{i=0}^n \frac{in!}{(n-i)!} \chi^i}{\sum_{i=0}^n \frac{n!}{(n-i)!} \chi^i}$$

Обозначим через  $t_{омв.ср}$  среднее время ожидания ответа (среднее время пребывания запроса в системной фазе). Среднее число запросов в системной фазе пропорционально общему числу пользователей и отношению  $t_{омв.ср}$  к средней продолжительности этапа взаимодействия  $t_{омв.ср} + \tau_{н.ср}$  :

$$t_{омв.ср} = \frac{L_{cp}}{n - L_{cp}} \tau_{н.ср} ,$$

откуда

$$t_{омв.ср} = \frac{\sum_{i=0}^n \frac{in^i}{(n-i)!}}{\sum_{i=0}^n \frac{\chi^i}{(n-i-1)!}} \tau_{н.ср} .$$

Подставляя в эту формулу выражения для  $L_{cp}$  с учетом выражения  $\pi_i$  после преобразований получаем :

$$L_{cp} = n \frac{t_{омв.ср}}{t_{омв.ср} + \tau_{н.ср}} .$$

Используя выражение для  $\pi_0$  , полученное выражение можно преобразовать к виду, обеспечивающему получение значения времени ответа

$$t_{омв.ср}/t_{обр.ср} = n/(1-\pi_0) - 1/\chi .$$

На основе вышеприведенных выкладок была построена машинная модель, пользовательский интерфейс которой приведен ниже.

**МОДЕЛЬ**  
информационно-справочной системы  
на базе вычислительной сети типа "файл-сервер"

**ИСХОДНЫЕ ДАННЫЕ**

Средняя продолжительность пультовой фазы запроса, с	5
Средняя продолжительность обработки запроса сервером, с	2
Количество пользовательских терминалов	15

**РЕЗУЛЬТАТЫ**

Коэффициент загрузки сервера	0,999999946
Среднее время получения ответа (справки), с	35,00000175
Среднее время пребывания в системной фазе, с	30,00000162
Среднее число запросов в системной фазе	12,85714291

**ПРОМЕЖУТОЧНЫЕ ДАННЫЕ**

Отношение времени обработки к пультовому времени	X	0,4
Вероятность незанятости сервера (предельная)	P0	5,02167E-08
X в степени j, j = 0 ... n	X**j	1,07374E-06
Факториал	(n - j)!	1
Сумма		1,63021E-05
Текущее значение индекса	j	15
Ячейка старта G22		1

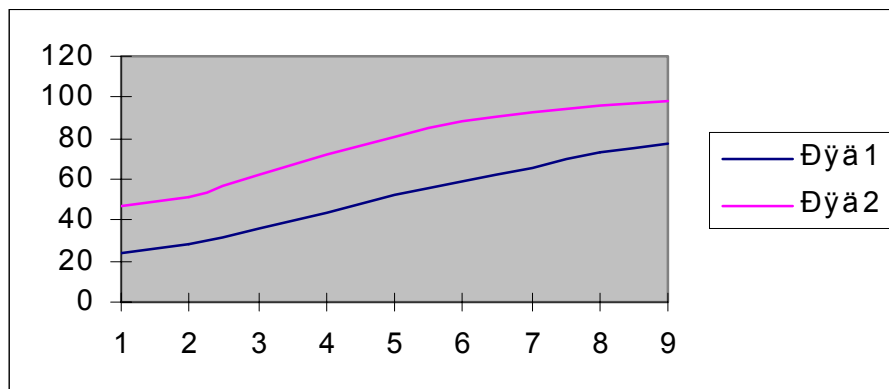


Рис.15.2

С помощью этой модели могут быть получены зависимости оценок времени ответа от продолжительности обработки запроса при различных количествах пользователей, коэффициент загрузки сервера и другие сведения. Например, на рис. 15.2 показаны зависимости времени получения ответа от длительности пультовой фазы для 10 и 20 терминалов при времени обработки сервером в 1 с.

**БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

1. Вентцель Е.С. Исследование операций: задачи, принципы, методология. – М.: Наука, 1980. –208с., ил.
2. Каган Б.М. Электронные вычислительные машины и системы: Учеб. пособие для вузов. - М.: Энергия, 1979. - 528с., ил.
3. Кулаков Ю.А., Луцкмй Г.М. Компьютерные сети. – Киев: Юниор, 1998. - 384с., ил.
4. Куликовский Л.Ф., Мотов В.В. Теоретические основы информационных процессов. - М.: Высшая школа, 1987. -248с., ил.
5. Ларионов А.М., Майоров С.А., Новиков Г.И. Вычислительные комплексы, системы и сети - Л.: Энергоатомиздат, 1987. -288с., ил.
6. Липаев В.В. Проектирование программных средств: Учебное пособие для вузов. – М.: Высшая школа, 1990. – 303 с., ил.
7. Советов Б.Я., Яковлев С.А. Моделирование систем: Учебник для вузов – 3-е изд., перераб. и доп. –М.: Высшая школа, 2001. –343с.: ил.