**Основные понятия теории планирования эксперимента.**

Следующим этапом после создания математической модели и её программной реализации является постановка вычислительного эксперимента. В теории планирования эксперимента исследуемый объект рассматривается как черный ящик, имеющий входы X и выходы Y. Переменные Х принято называть факторами. Факторы в эксперименте могут быть – качественными и количественными.

Качественные факторы можно квантифицировать или прописать им числовые обозначения, тем самым перейти к количественным значениям. В дальнейшем будем полагать, что все факторы являются количественными, представленными непрерывными величинами.

Переменным Х можно сопоставить геометрическое понятие факторного пространства, т.е. пространства, координатные оси которого соответствуют значениям факторов. Совокупность конкретных значений всех факторов образует точку в многомерном факторном пространстве.

Примеры факторов: интенсивность потока запросов в базе данных, скорость передачи данных по каналу, объем запоминающего устройства и т.д.

Но, к сожалению не все так хорошо. На объект воздействуют возмущающие факторы. Они являются случайными и не поддаются управлению.

Область планирования задается интервалами возможного изменения фактора:, , k - количество факторов.

***Нормализация факторов*** – преобразование натуральных значений факторов в безразмерные кодированные величины. Переход i-ого значения задается следующей формулой: - переход безразличного фактора Xi

Xi - натуральное значение фактора

Xi0 - натуральное значение основного уровня фактора, соответствующее нулю в безразмерной шкале

 - интервал варьирования

Совокупность основных уровней всех факторов представляет собой точку в пространстве параметров, называемой центральной точкой плана или центром эксперимента.

С геометрической точки зрения нормализация факторов равноценна линейному преобразованию пространства факторов, при котором проводятся две операции.

1. Перенос начала координат в точку соответствующую значениям основных уровней факторов
2. Сжатие/растяжение пространства в направлении координатных осей

*Активный эксперимент* включает: систему воздействий, при которых воспроизводится функционирование объекта и регистрация отклика объекта.

План эксперимента задаёт совокупность данных, определяющих количество, условия и порядок реализации опытов.

Опыт составляет элементарную часть эксперимента и предусматривает воспроизведение исследуемого явления в конкретных условиях с последующей регистрацией результатов.

В условиях случайности при одних и тех же условиях проводятся параллельные или повторные опыты, для получения статистически устойчивых результатов.

Опыт U предполагает задание конкретных значений фактора Х: . Совокупность значений факторов во всех N точках плана эксперимента образует матрицу плана:



Строки матрицы соответствуют опытам, столбцы - фактором.

Элемент матрицы  задает значение i-ого фактора в j-ом опыте.

Реализовав испытание N факторного пространства, определенным фактором эксперимента, получим вектор наблюдений, имеющий следующий вид:

, где текущий  соответствует i-ой точке плана

Зависимость отклика от факторов носит название ***функции отклика***, а геометрическое представление функции отклика - ***поверхностью отклика***. Функция отклика рассматривается как показатель качества или эффективности объекта. Этот показатель является функцией от параметров, в качестве которых выступают факторы.

;

 – вектор неизвестных параметров модели:

, h+1

 - вектор заданных базисных функций.

 - математическое ожидание функции отклика

Такое представление функции отклика соответствует линейной по параметрам модели регрессионного анализа, т.е. функция отклика есть линейная комбинация базисных функций от фактора.

Из-за влияния на результаты эксперимента случайных воздействий, истинное значение коэффициентов можно определить только приближенно. Оценку вектора  находят по результатам экспериментов. В ходе, которых мы получаем значение  при заданных значениях факторов . Эти оценки обычно оцениваются с помощью метода наименьших квадратов. Если не принимать специальных мер, то оценки коэффициентов  станут взаимозависимыми и полученное выражение для функции отклика можно рассматривать как интерполяционную формулу, что затрудняет её физическую интерпретацию и последующие расчеты.

*Получение независимых результатов.*

Для этого нужно формировать специальным образом матрицу плана. И эти величины будут характеризовать вклад каждого фактора в значение функции отклика.

*Основная задача* - определение основных формул функций отклика .

В большинстве случаев вид этой функции, получаемой из теоретических соображений, является сложным для практического применения.

Функции принято обозначать в некотором универсальном виде - в виде полинома.

Тогда системой базисной функции является совокупность степенных функций с целыми неотрицательными значениями показателя степени:



 - случайная величина, характеризующая ошибку опыта.

Такая функция отклика линейна относительно неизвестных коэффициентов и будет полностью определена, если задана степень полинома и коэффициенты. Степень полинома обычно задается исследователем априорно. На практике широкое распространение имеют полиномы первого и второго порядка. Коэффициенты полинома принято называть *эффектами факторов*. К большинству сложных систем применим принцип Павето, согласно которому 20% факторов определяют свойства системы на 80%. Поэтому первоначальной задачей при исследовании имитационной модели является отсеивание несущественных факторов, позволяющие упростить вычисления функции отклика. Одним из методов решения этой задачи является ***метод дисперсионного анализа***.

**Виды планирования эксперимента.**

Можно выделить ***стратегическое*** и ***тактическое*** планирование эксперимента

(..!..) Тактическое планирование: дать понятие и пример.

**Вычислительная система, как объект моделирования.**

В теории проектирования ВТ принято выделять уровни проектирования. Если рассматривать процесс проектирования электронной техники, как иерархический процесс, то самый верхний уровень будет: *системное проектирование*.

*>> Информационная система - ..на самостоятельную разработку..*

В качестве объектов на системном уровне проектирования нужно рассматривать процессор, память, каналы и т.д., так же нужно рассматривать ОС.

Далее идет функционально-логический уровень проектирование (ФЛУП). Входными характеристиками являются выходные параметры из системного проектирования.

ФЛУП делится на два уровня:

1. подуровень регистровых передач (нужно так же рассматривать физические характеристики устройств)
2. логический уровень

Следующий уровень: Схемотехнический уровень проектирования (УП).

Здесь и возникает проектирование интегральных схем

Далее: Конструкторский УП. Здесь рассматриваются вопросы о теплообмене, охлаждении и т.д.

*Вопрос: можно ли начать проектирование этой схемы снизу вверх?*

*Ответ:*

**Моделирование на системном уровне.**

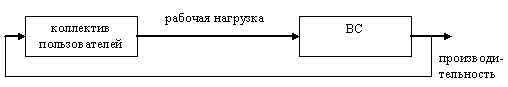
При моделировании новых и модернизации существующих вычислительных систем и сетей необходимо предварительно оценивать эффективность их функционирования с учетом различных вариантов структурной организации. Эти варианты могут отличаться составом и характеристиками устройств. Структурой межмодульных связей, режимами работы и алгоритмами управления. Для оценок таких структур используют модели вычислительных систем. Под *вычислительной системой* будем понимать комплект аппаратных (процессор, память, ву) и программных средств ( ОС ), которые в совокупности выполняют определенные рабочие функции.

>> ОС – множество согласованных управляющих программ [опр: на примитивном уровне]

***Коллектив пользователей*** – это сообщество таких людей, которые используют вычислительную систему для удовлетворения своих нужд по обработке информации.

Входные сигналы (программы, команды, данные), которые создаются коллективом пользователей, называются ***рабочей нагрузкой***.

*Схема вычислительной установки:*



***Индекс производительности*** (ИП) – описатель, который используется для представления производительности системы. Различают:

* Качественные ИП. Тип процессора – RISC/CISC, мощность системы команд.
* Количественные ИП. ***Пропускная способность*** – объем информации обрабатываемый в единицу времени. ***Время ответа (реакции)*** – время между предъявлением системе входных данных и появлением соответствующей выходной информации. ***Коэффициент использования оборудования*** – отношение времени использования указанной части системы в течение заданного интервала времени к длительности этого интервала.

**Концептуальная модель** включает в себя сведения о выходных и конструктивных параметрах системы, ёё структуре, особенности работы каждого ресурса (элемента системы), характере взаимодействия между ресурсами. Как правило, включается постановка прикладной задачи, определяющей цели моделирования исходной системы, а так же исходные данные для исследования системы.

**Формализованная схема** представляет собой, как правило, некоторую сложную систему массового обслуживания.

Основные задачи, которые необходимо решить:

1. Определение принципов организации вычислительной системы;
2. Выбор архитектуры, уточнение функции и их разделение на подфункции, реализуемое аппаратным или программным способом;
3. Разработка структурной схемы, т.е. определение состава устройств и способов их взаимодействия;
4. Определение требований к выходным параметров устройств и формирование технического задания для разработки отдельных устройств.

**Непрерывно стохастические модели (Q-схемы)**

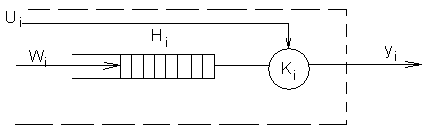
Особенность непрерывно стохастической модели будем рассматривать на примере систем массового обслуживания (СМО) в качестве типовых математических моделей. При этом используемая система формализуется как некая система обслуживания. Характерным для таких объектов является случайное появление требований (заявок) на обслуживание и завершение обслуживания в случайные моменты времени. Т.е. характер функционирования устройств носит стохастический порядок.

Основные понятия теории массового обслуживания**.**

В любом элементарном акте обслуживания можно выделить две основные составляющие:

1. Ожидание обслуживания
2. Собственно, обслуживание

Некоторые виды обслуживания некоторого оборудования:



ОА – обслуживающий аппарат

К – канал

Прибор обслуживания (i-ый) состроит из:

* накопителя заявок, в котором может одновременно находится Li = {0, }, где  – емкость i-ого накопителя
* канала обслуживания заявок.

***Потоком событий*** называется последовательность событий происходящих одно за другим в какие-то случайные моменты времени.

Поток событий называется ***однородным***, если он характеризуется только моментами поступления этих событий (вызывающие моменты) и задается временной последовательностью: , 

Поток называется ***неоднородным***, если он задается следующей совокупностью , где tn – вызывающий моменты, fn – набор признаков события( наличие приоритета, принадлежность к тому или иному типу заявки).

Если интервал времени между сообщениями независимыми между собой являются случайными величинами, то такой поток называется потоком с ***ограниченным*** последействием.

Поток событий называется ***ординарным***, если вероятность того, что на малый интервал времени  примыкающий к моменту времени t попадает более одного события, пренебрежительно мала по сравнению с вероятностью того что на этот же интервал  попадает ровно одно событие.

Поток называется ***стационарным***, если вероятность появления того или иного числа событий на некотором интервале времени зависит лишь от длины интервала и не зависит от того, где на оси времени взят этот участок.

Для ординарного потока среднее число сообщений наступивших на участке  примыкающих к некоторому моменту времени t будет равно .

Тогда среднее число сообщений наступивших на участке времени  составит:  - *интенсивность ординарного потока*.

Для стационарного потока – его интенсивность не зависит от времени и представляет собой постоянное значение равное среднему числу событий наступающих в единицу времени.

***Поток заявок*** (), т.е. интервалы времени между моментами появления заявок на входе канала (это подмножество неуправляемых переменных)

***Поток обслуживания*** () - т.е. интервалы времени между началом и окончанием обслуживанием заявок, принадлежат подмножеству управляемых заявок.

Заявки обслуженные каналом или заявки покинувшие прибор необслуженными, образуют выходной поток. Процесс функционирования i-ого прибора можно представить как процесс изменения состояний его элементов во времени.

Переход в новое состояние для i-ого прибора означает изменение количества заявок, которые находятся в накопителе или канале: 

Где  – *состояние накопителя*, если он = 0, то накопитель пуст (нет заявок), если количество заявок совпадает с емкостью накопителя, то накопитель полон;  - *состояние канала* (0 – свободен или 1 - занят).

В практике моделирования элементарные Q-схемы обычно объединяют, при этом, если каналы различных приборов обслуживания соединены параллельно, то имеет место ***многоканальное обслуживание***. А если последовательно – ***многофазное обслуживание***. Таким образом для задания Q-схемы необходимо использовать оператор сопряжения R, отражающий взаимосвязь элементов структуры. Различаются *разомкнутые* и *замкнутые* Q-схемы.

*Разомкнутые* – выходной поток заявок не может поступить к какому либо элементу, т.е. отсутствует обратная связь

*Замкнутые* – есть обратная связь.

Собственными внутренними параметрами Q-схемы будут являться:

* количество фаз
* количество каналов в каждой фазе
* количество накопителей каждой фазы
* ёмкость накопителя.

В зависимости от ёмкости накопителя в теории массового обслуживания применяют следующую терминологию: если емкость равна нулю (т.е. накопитель отсутствует, а есть только канал), то *система с потерями*. Если ёмкость стремится к бесконечности, то *система с ожиданием*, т.е. очередь заявок неограниченна.

**Система смешанного типа.**

Для задания Q-схемы так же необходимо описать алгоритм её функционирования, который определяет набор правил поведения заявок в системе в различных ситуациях. Неоднородность заявок, отражающая процессы в той или иной реальной системе, учитывается с помощью введения классов приоритетов.

Весь набор возможных алгоритмов поведения заявок в Q-схеме можно представить в виде оператора:

Q = (W, U, R, H, Z, A)

Где W - подмножество входных потоков;

U - подмножество потока обслуживания;

R - оператор сопряжения элементов структуры;

H - подмножество собственных параметров;

Z - множество состояний системы;

A - оператор алгоритмов поведения и обслуживания заявок;

Для получения соотношений связывающих характеристики, которые определяют функционирование Q-схемы, вводят некоторые допущения относительно входных потоков, функций распределения, длительности обслуживания запросов, дисциплин обслуживания.

Для математического описания функционирования устройств, процесс функционирования которого развивается в случайном порядке, могут быть применены математические модели для описания так называемых ***Марковских случайных процессов***.

**Ма́рковский проце́сс** — [случайный процесс](http://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D0%B9_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81), эволюция которого после любого заданного значения временно́го параметра tне зависит от эволюции, предшествовавшей t, при условии, что значение процесса в этот момент фиксировано («будущее» процесса не зависит от «прошлого» при известном «настоящем»; другая трактовка ([Вентцель](http://ru.wikipedia.org/w/index.php?title=%D0%92%D0%B5%D0%BD%D1%82%D1%86%D0%B5%D0%BB%D1%8C,_%D0%90%D0%BB%D0%B5%D0%BA%D1%81%D0%B0%D0%BD%D0%B4%D1%80_%D0%94%D0%BC%D0%B8%D1%82%D1%80%D0%B8%D0%B5%D0%B2%D0%B8%D1%87&action=edit&redlink=1" \o "Вентцель, Александр Дмитриевич (страница отсутствует))): «будущее» процесса зависит от «прошлого» лишь через «настоящее»).

Процесс Маркова — модель авторегрессии AR(1): xt=ψ1\*xt-1+εt

Случайный процесс называется Марковским, если он обладает следующим свойством – для каждого момента времени  вероятность любого состояния системы в будущем (т.е. в какой-то момент времени ) зависит только от состояния системы в настоящем и не зависит от того, когда и каким образом система пришла в это состояние. Иначе, в Марковском случайном процессе будущее его развитие зависит только от его настоящего состояния и не зависит от исторического процесса.

/\* реально таких систем, конечно, не существует. Но существуют механизмы, которые позволяют свести к этим процессам.\*/

Для Марковских процессов обычно составляют уравнения Колмогорова.

В общем виде уравнения Колмогорова выглядят следующим образом:

;

где  - вектор, определяющий некоторый набор коэффициентов присущих системе

Для стационарного соотношения:

,

что дает возможность для стационарной зависимости получить

.

А затем связать выходные характеристики через набор коэффициентов соответствующих системе:



Последнее соотношение представляет собой зависимость выходных параметров от некоторых внутренних параметров модели, и имеют название ***базисной модели***.

В результате всего нам нужно найти:

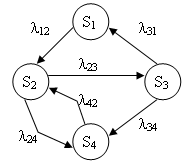


- которая будет называться ***интерфейсной моделью***.

Следовательно, математическая модель системы строится как совокупность базисной и интерфейсной модели, что позволяет использовать одни и те же базисные модели, для различных задач проектирования осуществляя настройку на соответствующую задачу посредством изменения только интерфейсной модели. Для Q-схем математическая модель должна обеспечивать вычисление времени реакции и определения производительности системы.

Пример: пусть есть некоторая система S, имеющая конечный набор состояний (будем рассматривать для 4 состояний).

Получаем ориентированный граф:

 - плотности вероятностей для множества состояний.

Найдем вероятность, т.е. вероятность того что в момент t система будет находиться в состоянии .

Придадим t малое приращение  и найдем, что в момент времени  система будет находится в состоянии .

Это может быть реализовано двумя способами:

1. В момент t система S уже была в состоянии  и за время  не вышла из него.
2. В момент t система была в состоянии  и за время  перешла из него в состояние .

Вероятность первого способа найдем как произведение вероятности  на условную вероятность того, что будучи в состоянии  система за время  не перейдет из него в состояние. Это условная вероятность с точностью до бесконечно малых величин высших порядков будет равна:



Аналогично вероятность второго способа равна вероятности того что в следующий момент t была в состоянии умноженную на условную вероятность перехода в состояния, т.е.:







=> 

Мы вывели уравнение колмагорова для первого состояния.

Выведем далее для 2, 3 и 4 состояний.











Интегрирование данной системы дает искомые вероятности системы как ф-ции времени. Начальные условия берутся в зависимости от того какого было начальное состояние системы. Например, если в момент времени t = 0, система находилась в состоянии, то начальное условие будет .

Кроме того, необходимо добавлять *условие нормировки* (сумма вероятностей = 1).

Уравнение Колмогорова строится по следующему правилу: в левой части каждого уравнения стоит производная вероятности состояния, а правая часть содержит столько членов сколько стрелок связано с данным состоянием. Если стрелка направлена из состояния, то соответствующий член имеет знак "-", в состояние – "+". Каждый член равен произведению плотности вероятности перехода (интенсивности) соответствующий данной срелке, умноженной на вероятность того состояния, из которого исходит стрелка.

**Лабораторная работа №1.**

Определить среднее относительное время прибывания системы в предельном стационарном состоянии. Интенсивности переходов из состояния в состояние задаются в виде матрицы размером ≤ 10.

Отчет: название, цель, теоретическая часть и расчеты.

Рассмотрим многоканальную систему массового обслуживания с отказами.

Будем нумеровать состояние системы по числу занятых каналов. Т.е. по числу заявок в системе.

Обзовем состояния:

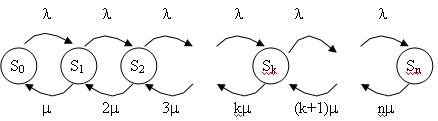
- все каналы свободны

 - занят один канал, остальные свободны

 - занято k каналов, остальные свободны

- заняты все n каналов

Граф состояний:



Разметим граф, т.е. расставим интенсивности соответствующих событий.

По стрелкам слева направо система переводит один и тот же поток с интенсивностью .

Определим интенсивность потоков событий, переводящих систему справа на лево.

Пусть система находится в . Тогда, когда закончится обслуживание заявки занимающей этот канал, система перейдет в  => поток, переводящий систему в другое состояние, будет иметь интенсивность перехода *μ*. Если занято 2 канала, а не один, то интенсивность перехода составит 2*μ*.

Уравнения Колмогорова:



Предельные вероятности состояний *p0*и *pn* характеризуют установившийся режим работы системы массового обслуживания при *t*→ ∞.





 - среднее число заявок, приходящих в систему за среднее время обслуживания одной заявки.





Зная все вероятности состояний *p0*, … , *pn* , можно найти характеристики СМО:

* *вероятность отказа –* вероятность того, что все n каналов заняты



* *относительная пропускная способность –* вероятность того, что заявка будет принята к обслуживанию



* среднее число заявок, обслуженных в единицу времени



Полученные соотношения могут рассматриваться как базисная модель оценки характеристик производительности системы. Входящий в эту модель параметр , является усредненной характеристикой пользователя. Параметр *μ* является функцией технических характеристик компьютера и решаемых задач.

Эта связь может быть установлена с помощью соотношений, называемых интерфейсной моделью. Если время ввода/вывода информации по каждой задачи мало по сравнению со временем решения задачи, то логично принять, что время решения равно *1 / μ* и равно отношению среднего числа операций, выполненных процессором при решении одной задачи к среднему быстродействию процессора.



Самостоятельно: Метод вложенных цепей Маркова

Требования к отчету: название, цель, краткие теоретические сведения (писать то что не знаешь), пример, текст программы.

***Не Марковские случайные процессы, сводящиеся к Марковским.***

Реальные процессы весьма часто обладают последействием и поэтому не являются Марковским. Иногда при исследовании таких процессов удается воспользоваться методами, разработанными для Марковских цепей. Наиболее распространенными являются:

1. Метод разложения случайного процесса на фазы (метод псевдо состояний)
2. Метод вложенных цепей

Метод псевдо состояний**.**

Сущность метода заключается в том, что состояние системы, потоки переходов из которых являются немарковскими, заменяются эквивалентной группой фиктивных состояний, потом переходы, из которых уже являются Марковскими.

Условие статистической эквивалентности реального и фиктивного состояния могут в каждом конкретном случае выбираться по-разному. Очень часто может использоваться следующее: , где  - эквивалентная интенсивность перехода в i-ой группе переходов, заменяющей реальный переход, обладающий интенсивностью.

За счет расширения числа состояний системы некоторые процессы удается точно свести к Марковским. Созданная таким образом система статистически эквивалентна или близка к реальной системе, и она подвергается обычному исследованию с помощью аппарата Марковских цепей.

К числу процессов, которые введением фиктивных состояний можно точно свести к Марковских относятся процессы под воздействием потоков Эрланга. В случае потока Эрланга k-ого порядка интервал времени между соседними событиями представляет собой сумму k независимых случайных интервалов, распределенных по показательному закону. Поэтому с введением потока Эрланга k-го порядка к Пуассоновскому осуществляется введением k псевдо состояний. Интенсивности переходов между псевдо состояниями равны соответствующему параметру потока Эрланга. Полученный таким образом эквивалентный случайный процесс является Марковским, т.к. интервалы времени нахождения его в различных состояниях подчиняются показательному закону.

*Пример*. Устройство S выходит из строя с интенсивностью , причем поток отказов Пуассоновский. После отказа устройство восстанавливается. Время восстановления распределено по закону Эрланга 3 порядка с функцией плотности .

Найти предельные вероятности возможных состояний системы.

*Решение*.

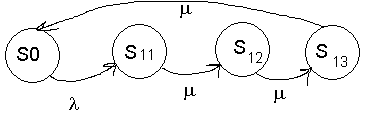
Пусть система может принимать 2 возможных состояния:

 - устройство исправно;

 - устройство отказало и восстанавливается

Переход из  в  осуществляется под воздействием пуассоновского потока, а из  в - потока Эрланга.

Представим случайное время восстановления в виде суммы 3х случайных временных интервалов, распределенных по показательному закону с интенсивностью .







Ответ: , 

**Метод вложенных цепей Маркова.**

Вложенные цепи Маркова образуются следующим образом. В исходном случайном процессе выбираются такие случайные процессы, в которых характеристики образуют Марковскую цепь. Моменты времени обычно являются случайными и зависят от свойств исходного процесса. Затем обычными методами теории Марковских цепей исследуются процессы только в эти характерные моменты. Случайный процесс называется ***полумарковским*** (с конечным или счетным множеством состояний) если заданы переходы состояний из одного состояния в другое и распределение времени пребывания процессов в каждом состоянии. Например, в виде функции распределения или функции плотности распределения.

<остальное самостоятельно>

**Метод статистических испытаний. Метод Монте-Карло.**

В СМО поток заявок редко бывает Пуассоновским и еще реже наблюдается распределенный закон.

Для произвольных потоков событий переводящих систему из состояния в состояние. Аналитические решения получены только для отдельных частных случаев. Когда построение аналитической модели является по той или иной причине трудно осуществимым ставится метод статистических испытаний.

\**Когда этот метод нашел реальное применение: с развитием компьютеров.*

**Идея метода:** вместо того чтобы описывать случайные явления с помощью аналитической зависимости производится т.н. «розыгрыш», т.е. моделирование «случайного» явления с помощью некоторой процедуры дающей «случайный» результат. Проведя такой розыгрыш достаточно большое количество раз, получаем ***статистический материал***, т.е. множество реализаций случайного явления. Дальше эти результаты могут быть обработаны статическими методами математической статистики.

Метод Монте-Карло был предложен в 1948 году Фон-Нейманом как метод численного решения некоторых математических задач.



Введем в некотором единичном квадрате случайную величину. Задача стоит в определении её площади.

**Суть метода**:

1. Вводим в некотором единичном квадрате любую поверхность S.
2. Любым способом получаем 2 числа xi, yi, подчиняющиеся равномерному закону распределения случайной величины на интервале [0, 1].
3. Полагаем, что одно число определяет координату x, второе – координату y
4. Анализируем принадлежность точки (x, y) фигуре. Если принадлежит, то увеличиваем значение счетчика на 1.
5. Повторяем n раз процедуру генерации 2х случайных чисел с заданным законом распределения и проверку принадлежности точки поверхности S.
6. Определяем площадь фигуры как количество попавших точек, к количеству сгенерированных.

Фон-Нейман доказал, что погрешность .

*Преимущество* метода статистических испытаний: в его универсальности, которая обуславливает его возможность статистического исследования объекта, причем всестороннего. Но для реализации этого исследования необходимы довольно полные статистические сведения о параметрах элемента.

*Недостаток*:

Большой объем требующихся вычислений, равный количеству обращений к модели. Поэтому вопрос выбора величины n имеет важнейшее значение. Уменьшая n – повышаем экономичность расчетов, но одновременно ухудшаем их точность.

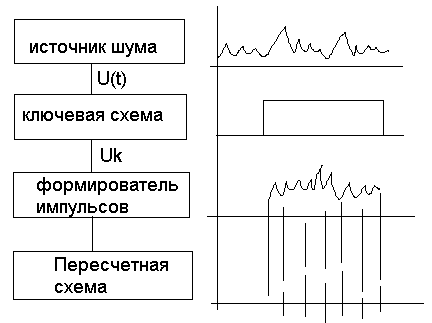
***Способы получения псевдослучайных чисел.***

При имитационном моделировании системы одним из основных вопросов является стохастических воздействий. Для этого метода характерно большое число операций со случайными числами или величинами и зависимость результатов от качества последовательностей случайных чисел. На практике используется 3 основных способа:

1. **Аппаратный** (физический)
2. **Табличный** (файловый)
3. **Алгоритмический** (программный)

**Аппаратный способ.**

Аппаратный представляет из себя шум.



Случайные числа вырабатываются случайной электронной приставкой (генератор случайных чисел), служащей, как правило, в качестве одного из внешних устройств. Реализация этого способа обычно не требует дополнительных вычислений, а необходима только операция обращения к ВУ. В качестве физического эффекта, лежащего в основе таких генераторов чаще всего используют шумы в электронных приборах.

Т.е. необходимо: источник шума, ключевая схема, формирователь импульсов, пересчетная (см. рис.)

**Табличная схема.**

Случайные числа оформляются в виде таблицы и помещаются во внешнюю или оперативную память.

**Алгоритмический способ.**

Способ основан на формировании случайных чисел с помощью специальных алгоритмов.

**Преимущества и недостатки типов генерации случайных чис**

|  |  |  |
| --- | --- | --- |
| Способ | Достоинства | Недостатки |
| Аппаратный | 1. Запас чисел неограничен 2. Расходуется мало операций 3. Не занимается место в оперативной памяти. | 1. Требуется периодическая проверка на случайность 2. Нельзя воспроизводить последовательности 3. Используются специальные устройства. Надо стабилизировать |
| Табличный | 1. Требуется однократная проверка 2. Можно воспроизводить последовательности | 1. Запас чисел ограничен 2. Занимает место в оперативной памяти и требуется время на обращение к памяти |
| Алгоритмический | 1. Однократная проверка 2. Можно многократно воспроизводить последовательности чисел 3. Относительно малое место в оперативной памяти 4. Не используются внешние устройства | 1. Запас чисел последовательности ограничен её периодом 2. Требуются затраты машинного времени |

**Лабораторная работа №2.**

Сравнить эти два способа. Причем сравниваем их по критерию случайности. Т.е. придумать свой критерий случайности (можно конечно и в книжке посмотреть, но лучше самому)..

количественная оценка..

К пятнице:

1. Алгоритм Марселя Зейнмана (?). (целочисленная арифметика) (3-я группа)

2. Функция, увеличивающая период последовательности стандартного генератора rand.

3. Способ Ленмара (?)

Лекция 10.

В настоящем времени с помощью рекуррентных математических уравнений реализовано несколько алгоритмов генерирования псевдослучайных чисел. ***Псевдослучайными*** эти числа называются потому, что фактически они, даже пройдя все статистические испытания на случайность и равномерность распределения, остаются полностью ***детерминированными***. Т.е. если каждый цикл работы генератора начинается с одними и теми же с исходными данными (константами и начальными условиями и значениями), то на выходе мы получаем одни и те же последовательности.

*Сферы применения*.

Генератор случайных чисел используется в программных приложениях связанных с конструированием ядерных реакторов, радиолокационного оповещения и обнаружения, поисков полезных ископаемых, многоканальные связи и т.д.

***Простейшие алгоритмы генерации последовательности псевдослучайных чисел***

Одним из первых способов получения последовательности псевдослучайных чисел было выделение дробной части многочлена первой степени: 

Если n пробегает значения натурального ряда чисел, то поведение yn выглядит весьма хаотично. Физик Якобит доказал, что при рациональном коэффициенте *a* множество *y* конечно, а при иррациональном – бесконечно и всюду плотно в интервале [0, 1]. Для многочленов больших степеней такую задачу решил Герман Вей, т.е. он предложил критерий равномерности распределения любой функции от натурального ряда чисел. Называется это ***эргодичностью*** и заключается в том, что среднее по реализациям псевдослучайных чисел равно среднему по всему их множеству с вероятностью 1. Эти результаты далеки от практики, поэтому она используется только для действительных чисел, что затрудняет практическую её реализацию. Попытки замены настоящего иррационального числа его приближением на компьютере привели к тому, что полученные последовательности оканчиваются циклом с коротким периодом.

1. **1946 год, Фон Нейман**.   
   Каждое последующее число образуется возведением предыдущего в квадрат и отбрасыванием цифр. Способ с точки зрения случайности оказался нестабильным.
2. **Лимер**  
     
   Для подбора коэффициентов k, c, m потрачены десятки лет. Подбор почти иррациональных k ничего не дает. Установили, что при c = 0 и  наибольший период достигается при нечетном начальном числе и при k = 3 + 8i, k = 5 + 8i.
3. **Форсайд**

В 1977 году показал, что тройки последовательности чисел лежат на 15 параллельных плоскостях.

От отчаяния используют 2 и даже 3 разных генератора, смешивая их значения. Если бы разные генераторы не зависели, то сумма их последовательностей обладала дисперсией равной сумме дисперсией. Иначе случайность рядов возрастет при суммировании. Сейчас в системах программирования обычно используют конгруэнтные генераторы по алгоритму, предложенному национальному бюро стандартов США, который имеет длину .

Генерация случайных чисел по алгоритму Зеймана.

{1, 1, 2, 3, 5, 8, 13, 21, …}

mod 10

{1, 1, 2, 3, 5, 8, 3, 1, …}

Переименовываем с помощью какого-либо ГСЧ; пусть всё так и осталось:

{1, 1, 2, 3, 5, 8, 3, 1, …}

С начала CF = 1.



*Пример*:

randomize(231)

x = rnd();

randomize(231)

y = rnd();

// x != y

x = rnd(-231)

y = rnd(-231)

// x = y

(Серега рассказывает про то как можно пытаться смешивать генерацию случайных чисел)

**Лабораторная работа №3.**

Суть: изучить 2 стандартных распределения по всем свойствам распределения

Ф-ция распределения, плотность распределения, мат. ожидание, дисперсия, …

Равномерное распределение изучают все.

По списку с периодом 4 изучают

1.

2. экспоненциальное

3. нормальное распределение (Гауссовское)

4. k – распределение Эрланга

Построить графики:

1. Теоретического распределения (функция и плотность распределения)
2. Экспериментального по «своему» закону распределения (ф-ция и плотность)

Лекция 11

Программа генерации случайных чисел на Фортране для машин ES (~IBM 360)

SUBROUTINE RANDUM( IX, IY, RN) // была придумана для 32 разрядной машины

IY = IX \* 1220703125

IF (IY) 3,4,4 // if ( IY < 0) then

3 IY = IY + 2147483647 + 1

4 RN = IY

RN = RN \* 0.4656613E-9

IX = IY

RETURN

END

// обращение к данной процедуре

CALL RANDUM(IX, IY, YFL)

IX – число, которое при первом обращении должно содержать нечетное целое число, состоящее менее чем из 9 цифр

IY - полученное случайное число, используемое при последующих обращениях к программе

YFL - полученное равномерно распределенное в интервале [0, 1] случайное число

Для имитации **равномерного распределения в интервале от [a, b]** используется обратное преобразование функции плотности вероятности:



где R – равномерно распределенное псевдослучайное число на [0, 1].

В основе построения программы, генерирующей случайные числа с законом распределения отличным от равномерного лежит **метод преобразования последовательности случайных чисел с равномерным законом распределения в последовательность случайных чисел с заданным законом**.

 (1)

Метод основан на утверждении, что случайная величина *x,* принимающая значения, равные корню уравнения (1) имеет плотность распределения *f(x)*. *R* - равномерная случайная величина от 0 до 1.

Значение случайной величины распределенной по показательному закону исходя из (1) может быть вычислено следующим образом:



***Распределение Пуассона.***

*Распределение Пуассона* относится к числу дискретных, т.е. таких при которых переменная может принимать лишь целочисленные значения, включая норму с мат. ожиданием и дисперсией равной λ > 0.

Для генерации Пуассоновских переменных можно использовать метод точек, в основе которого лежит генерируемое случайное значение *Ri* , равномерно распределенное на [0, 1], до тех пор, пока не станет справедливым



При получении случайной величины, функция распределения которой не позволяет найти решение уравнения (1) в явной форме можно произвести кусочно-линейную аппроксимацию, а затем вычислять приближенное значение корня. Кроме того, при получении случайных величин часто используют те или иные свойства распределения.

***Распределение Эрланга.***

*Распределение Эрланга* характеризуется двумя параметрами: λ и k. Поэтому при вычислении случайной величины в соответствии с данным законом воспользуемся тем, что поток Эрланга может быть получен прореживанием потока Пуассона k раз. Поэтому достаточно получить k значений случайной величины распределенной по показательному закону и усреднить их.



***Нормальное (Гауссово) распределение.***

Нормально распределенная случайная величина может быть получена как сумма большого числа случайных величин распределенных по одному и тому же закону и с одними и теми же параметрами.

Случайная величина X имеющая нормальное распределение с математическим ожиданием MX и среднеквадратичным отклонением σX может быть получена по следующей формуле:



Для сокращения вычислений по нормальному закону распределения на практике часто принимают N = 12. Что в дает довольно точные результаты.

**Процедура генерирования псевдослучайных чисел (равномерный и нормальный законы распределения):**

var n, i:integer;

x,R:double;

Const m34: double = 28395423107.0;

m35: double = 34359738368.0;

m36: double = 68719476736.0;

m37: double = 137438953472.0;

function Rand(n:integer):double;

var S, W: double;

i: integer;

begin

if n = 0 then

begin

x := m34; Rand := 0; exit;

end;

S := -2.5;

for i := 1 to 5 do

begin

x := 5.0 \* x;

if x > m37 then x := x - m37;

if x > m36 then x := x - m36;

if x > m35 then x := x - m35;

w := x / m35;

if n = 1 then

begin

Rand := W; exit

end;

S := S + W;

end;

S := S \* 1.54919;

Rand := ( sqr(S) - 3.0 ) \* S \* 0.01 + S;

end;

begin

R := Rand(0);

for i := 1 to 200 do

writeln( Rand(2):18:10)

end.

При n = 0 происходит параметрическая настройка или т.н. «установка».

При n = 1 будем получать равномерно распределенную случайную величину.

При n = 2 будет гауссово (нормальное) распределение.

["всем, пожалуйста, поиграться с этим алгоритмом и построить график" (с) by Рудаков]

***Методика построения программной модели ВС.***

Для разработки программной модели исходная система должна быть представлена как *стохастическая система массового обслуживания*. Это можно объяснить следующим: информация от внешней среды поступает в случайные моменты времени, длительность обработки различных типов информации может быть в общем случае различна. Т.о. внешняя среда является *генератором сообщений*. А комплекс вычислительных устройств (ВС) – *обслуживающими устройствами*.

**Обобщенная структурная схема ВС.**



**ИИ** – источники информации – выдают на вход буферной памяти (БП) независимые друг от друга сообщения. Закон появления сообщений – произвольный, но задан на перед.

В **БП** (буферной памяти) сообщения записываются «в навал» и выбираются по одному в обслуживающий аппарат (ОА) по принципу FIFO/LIFO. Длительность обработки одного сообщения в **ОА** в общем случае так же может быть случайной, но закон обработки сообщений должен быть задан. Т.к. быстродействие ОА ограничено то на входе системы в БП возможно сложение данных ожидающих обработки.

**А** – абоненты.

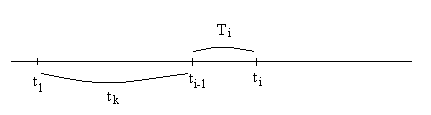
Программная модель из этой системы создается следующим образом:



Должна быть обязательно программа сбора статистики (**ПБССт –** программный блок сбора статистики). Причем статистику программа должна собирать по каждому из объектов модели. Так же должна быть программа, которая позволит "оживить" систему – это **программа синхронизации** (**блок синхронизации**), которая покажет когда и в какое время будут активизированы те или иные фрагменты модели.

**Моделирование работы источника информации (ИИ).**

Поток сообщений обычно имитируется моментами времени, отображающими появление очередного сообщения в потоке.





где Ti – интервал времени между появлением i-го и (i-1)-го сообщения.

Программа – имитатор выработки таких интервалов:

* + 1. Обратиться к генератору равномерно распределенных случайных величин на [a,b]
    2. Ti – по заданному закону
    3. К текущему времени + Ti

// процедурка равномерного распределения псевдослучайных чисел на итервале [a,b]

// U - равном. распр. на [0, 1]

// x = a + (b - a)U

double get\_time (int i)

{

double S = 0;

srand(seek);

if ( i > 1 ) S += get\_time(i - 1);

S += a + (b - a)get\_u();

return S;

}

или

double get\_time (int i)

{

double S = 0;

srand(seek);

for (int i = 0; i < .. ; i++)

S += a + (b - a)rand();

return S;

}

Выражения для вычисления времени с разлчным распределением:

|  |  |
| --- | --- |
| **Вид распределения** | **Выражение** |
| равномерное на [a,b] |  |
| Нормальное | , n ~= 12 |
| Экспоненциальное |  |
| Эрланга |  |

**Моделирование работы Обслуживающего Аппарата.**

Программа-имитатор работы ОА представляет собой комплекс, вырабатывающий случайные отрезки времени, соответствующие длительностям обслуживания требований. Например, если требования от источника обрабатываются в ОА по нормальному закону с параметрами *Mx* и *σx*, то длительность обработки *i*-ого требования:



**Схема алгоритма имитатора.**



Ri – случайное число с равномерным законом распределения

ТОБР – время обработки очередного сообщения

T – время освобождения ОА

XM – Мат ожидание для заданного закона обратки

DX – СКО (средне квадратичное отклонение) для заданного закона обратоки



**Моделирование работы абонентов.**

Абонент может рассматриваться как Обслуживающий Аппарат, поток информации, который поступает от процессора.

Для имитации работы абонентов необходимо составить программу выработки длительности обслуживания требования. Кроме того, абонент сам может быть источником заявок на те или иные ресурсы вычислительной системы. Эти заявки могут моделироваться с помощью генератора сообщений, распределенными по заданному закону. Таким образом, абонент либо имитируется как ОА, либо как генератор.

**Моделирование работы буферной памяти.**

**Память** - относится к электромеханическому устройству, включающее в себя: среду для запоминания, устройство управления, (информация находится по адресу) база + смещение + [индекс].

**Свойства памяти**: предназначена для хранения, чтения и записи информации.

В блок статистики: ошибки записи, ошибки чтения.

Блок буферной памяти должен производить запись и считывание чисел, выдавать сигналы переполнения и отсутствия данных в любой момент времени располагать сведениями о количестве требований (заявок) в блоке. Сама запоминающая среда в простейшем случае имитируется одномерным массивом, размер которого определяет ёмкость памяти. Каждый элемент этого массива может быть либо свободен и в этом случае мы считаем, что он равен 0, либо «занят», в этом случае в качестве эквивалента требования ему присваивается значение времени появления требования.

**Структурная схема модели программной памяти**:



Алгоритм реализации работы буферной памяти:



|  |  |  |  |
| --- | --- | --- | --- |
| *P* | массив сообщений | *LM* | объем буферной памяти |
| *WYB* | признак обращения к буф. памяти  = 1 – режим выборки сообщений  = 0 – режим записи | *NPOS* | номер последнего сообщения, поступившего в память |
| *NP* | число сообщений в памяти | *NPER* | номер первого сообщения в памяти |
| *POLN* | признак переполнения памяти  = 1 – нет свободных ячеек | *PUST* | признак отсутствия сообщений  = 1 – в памяти нет сообщений |
| *NPOS* | = NPOS + 1, если NPOS < LM  = NPOS – LM + 1, иначе | *NPER* | = NPER – 1, если NPER < 1  = NPER – LM + 1, иначе |
| *X* | ячейка для сообщения |  |  |

**Разработка программы для сбора статистики.**

Задача блока статистики заключается в накоплении численных значений необходимых для вычисления статистических оценок, заданных параметров работы моделируемой системы. При моделировании простейшей модели СМО, как правило, оценивают ***среднее время ожидания в очереди***. Для каждого сообщения *время ожидания в очереди* равно разности между моментами времени когда оно было выбрано на обработку обслуживающим аппаратом и моментом времени когда оно пришло в систему от источника информации.

Суммируя количество сообщений в блоке памяти через небольшие промежутки времени и разделив полученную сумму на число суммирований, получим *среднее значение длины очереди*.

*Коэффициент загрузки* обслуживающего аппарата (ОА) определяется как отношение времени работы ОА, к общему времени моделирования.

Чтобы определить *вероятность потери сообщений* в системе, нужно разделить кол-во потерянных сообщений на сумму потерянных и обработанных сообщений в системе.

**Управляющая программа имитационной модели.**

Если программа-имитатор работы источника или буферной памяти обслуживающего аппарата имитируют работу отдельных устройств, то ***управляющая программа*** имитирует алгоритм взаимодействия отдельных устройств системы.

Управляющая программа реализуется в основном по двум принципам:

1. **Принцип Δt**
2. **Событийный принцип**

***Принцип Δt.***

Принцип Δt заключается в последовательном анализе состояний всех блоков в момент t + Δt по заданному состоянию блоков в момент t. При этом новое состояние блоков определяется в соответствии с их алгоритмическим описанием с учетом действующих случайных факторов, задаваемых распределениями вероятности. В результате такого анализа принимается решение о том, какие общесистемные события должны имитироваться программной моделью на данный момент времени.

Основной **недостаток** этого принципа: значительные затраты машинного времени на реализацию моделирования системы. А при недостаточно малом Δt появляется опасность пропуска отдельных событий в системе, что исключает возможность получения адекватных результатов при моделировании.

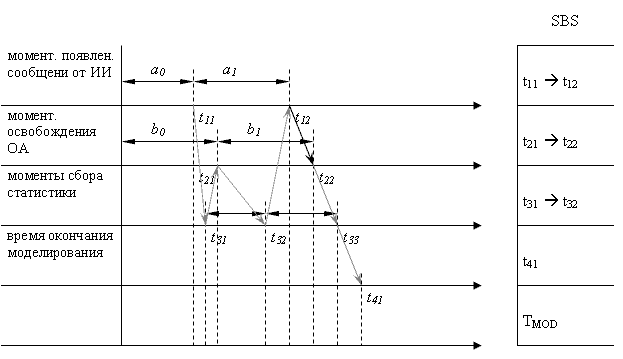
**Достоинство**: равномерная протяжка времени.

***Событийный принцип.***

Характерное свойство систем обработки информации то, что состояние отдельных устройств изменяются в дискретные моменты времени, совпадающие с моментами времени поступления сообщений в систему, временем поступления окончания задачи, времени поступления аварийных сигналов и т.д. Поэтому моделирование и продвижение времени в системе удобно проводить, используя ***событийный принцип***, при котором состояние всех блоков имитационной модели анализируется лишь в момент появления какого-либо события. Момент поступления следующего события определяется минимальным значением из списка будущих событий, представляющего собой совокупность моментов ближайшего изменения состояния каждого из блоков системы.

Недостаток событийного принципа: (самостоятельная обработка)

**Схема событийного принципа**:



Первая ось: момент появления сообщений

Вторая ось: момент освобождения обслуживающего аппарата

Третья ось: момент сбора статистики (здесь абсолютно равные интервалы, мы сами определяем, когда собирать статистику)

Четвертая ось: время окончания моделирования

Пятая ось: текущее время

***t11*, *t12*** – моменты появления сообщений на выходе генератора (источника информации)

***b1*** – интервал времени обслуживания первого сообщения

***t3 n*** – момент сбора статистики

***t41*** – момент окончания моделирования

SBS – список будущих событий.

Списки второго уровня и выше являются масшабирующими списками, кол-во эл-тов в кот.=некоторому константному значению N2кот хар-ет коэф масштабирования временных интералов.

**Методика реализации событийной модели.**

1. Для всех активных блоков (блоки, порождающие события) заводят свой элемент в одномерном массиве – в списке будущих событий (СБС).
2. В качестве подготовительной операции в СБС заносят время ближайшего события от любого активного блока. Активизируя программный имитатор источника событий вырабатывают псевдослучайную величину *a0*, определяющую момент появления первого сообщения *t11* от источника информации и эту величину заносят в СБС. Активизируя программу-имитатор, ОА вырабатывает псевдослучайную величину *b0*, определяющую момент времени *t21*, которую также заносят в SBS. В момент времени *t31* (момент первого сбора статистики) определяется равным стандартному шагу сбору статистики *tСТАТ* и заносится так же в СБС. В этот же список заносим время окончания моделирования *t41*. На этом подготовительный этап заканчивается и далее протяжка времени осуществляется по следующему алгоритму:

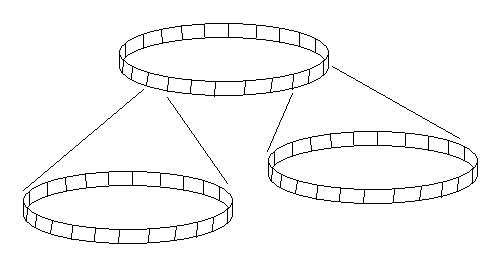


Алгоритм:

1. В SBS определяется минимальное числовое значение и его номер.
2. Реализуется событие, порождаемое блоком с соответствующим номером, т.е. модельное время = *t11*. Далее реализуется событие с номером 1, связанное с появлением нового сообщения в ИИ. Реализация этого события заключается в том, что само сообщение записывается в память, а с помощью имитатора ИИ, вырабатывается момент появления следующего события *t12*. Это время помещается в соответствующую ячейку SBS вместо *t11*.
3. Затем вновь организуется поиск минимального элемента в SBS. Для данного примера реализуется событие 3, после чего выражение момента времени t32 – новое время сбора статистики. Так до тех пор, пока минимальное время не станет равным *t41*.

***Комбинированный метод.***

Два приведенных метода являются универсальными алгоритмами протяжки модального времени. Причем для некоторых предметных областей один принцип может работать быстро и без потерь, а другой будет работать неэффективно. Выбор метода необходимо производить исходя из распределения событий по времени. В реальных системах распределение событий, как правило, *неоднородно*. События, как бы группируются по времени. Образование таких групп связано с наступлением какого-то «значимого» события, которое начинает определенную последовательность действий с соответствующими событиями, имеющими высокую плотность на следующем временном интервале. Такой интервал называется ***пиковым***. А распределение событий ***квази-синхронным***. Примером может являться – *цифровая сеть*, в которой синхронизирующие сигналы переключают большое количество триггеров. Для сложных дискретных систем, в которых присутствуют *квазисинхронное* распределение событий, был разработан алгоритм с название **Delft**. Особенностью данного метода является автоматическая адаптация к распределению событий. Метод реализуется таким образом, что на пиковых интервалах он приближается к *методу Δt*, а вне пиковых к *событийному*. В основе лежит использование иерархической структуры циркулярных списков.



Список уровня 1 содержит n1 элементов и описывает планируемое событие в пиковых интервалах. Число n1 представляет собой разбиение пикового интервала на более мелкие участки, с каждым из которых связан список событий происшедших за этот интервал. Списки второго уровня и выше являются *масштабирующими списками*, количество элементов которого равно константному значению n2, которое характеризует коэффициент масштабирования временных интервалов.

Собственно алгоритм протяжки времени заключается в последовательном поиске непустых элементов в самом верхнем циркулярном списке с большим шагом и дальнейшим спуском на нижние уровни (иерархические), вследствие чего уменьшается шаг протяжки модельного времени.