
Федеральное агентство по образованию
САНКТ–ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

**Институт компьютерных наук и технологий
Высшая школа программной инженерии**

О.В. Александрова, В.П. Котляров

Архитектура вычислительных систем

**Методические указания для курсового
проектирования**

Санкт-Петербург
2018

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	6
ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ (IEEE Std 610.12-1990).....	9
1. ОСНОВЫ ФОРМАЛЬНОЙ МОДЕЛИ ФУНКЦИОНИРОВАНИЯ ВС.....	11
2. ПРИНЦИПЫ ПОСТРОЕНИЯ МОДЕЛИРУЮЩЕГО АЛГОРИТМА.....	14
3. ОБЩАЯ ПОСТАНОВКА ЗАДАЧИ НА ПРАКТИЧЕСКУЮ РАБОТУ.....	16
3.1. ФОРМУЛИРОВКА ОБЩЕЙ ПОСТАНОВКИ ЗАДАЧИ.....	16
3.2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ РАБОТЫ И ИССЛЕДУЕМЫМ ХАРАКТЕРИСТИКАМ МОДЕЛИРУЕМОЙ ВС.....	17
4. ЭТАПЫ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ, ОТЧЁТНОСТЬ ПО ЭТАПАМИ КОНТРОЛЬНЫЕ ВОПРОСЫ	19
4.1 ПЕРВЫЙ ЭТАП ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ	19
4.2 ВТОРОЙ ЭТАП ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ	22
4.3. ТРЕТИЙ ЭТАП ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ	26
4.4 ЧЕТВЕРТЫЙ ЭТАП ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ	28
Приложение 1	34
Формулы заданий на курсовую работу «Создание имитационной модели вычислительной системы».....	34
Приложение 2	36
Варианты отображения результатов работы имитационной модели (в пошаговом и автоматическом режимах)	36
Приложение 3	39
Дополнительная информация к формуле задания	39
Приложение 4.....	46
Приложение 5.....	47
Приложение 6.....	61
Метод особых событий.....	61

ВВЕДЕНИЕ

Имитационное моделирование [1] — метод, позволяющий строить модели, описывающие процессы так, как они проходили бы в действительности. Такую модель можно «проиграть» во времени как для одного испытания, так и заданного их множества. При этом результаты будут определяться случайным характером процессов. По этим данным можно получить достаточно устойчивую статистику.

Имитационное моделирование — это метод исследования, при котором изучаемая система заменяется моделью, с достаточной точностью описывающей реальную систему и с ней проводятся эксперименты с целью получения информации об этой системе. Экспериментирование с моделью называют имитацией (имитация — это постижение сути явления, не прибегая к экспериментам на реальном объекте).

Имитационная модель — логико-математическое описание объекта, которое может быть использовано для экспериментирования на компьютере в целях проектирования, анализа и оценки функционирования объекта.

К имитационному моделированию прибегают, когда:

- дорого или невозможно экспериментировать на реальном объекте;
- невозможно построить аналитическую модель: в системе есть время, причинные связи, последствия, нелинейности, стохастические (случайные) переменные;
- необходимо симитировать поведение системы во времени.

Цель имитационного моделирования состоит в воспроизведении поведения исследуемой системы на основе результатов анализа наиболее существенных взаимосвязей между ее элементами или другими словами — симулятора (**simulation modeling**) исследуемой предметной области для проведения различных экспериментов.

Имитационное моделирование позволяет имитировать поведение системы во времени. Причём плюсом является то, что временем в

модели можно управлять: замедлять в случае с быстропротекающими процессами и ускорять для моделирования систем с медленной изменчивостью.

Применение имитационных моделей дает множество преимуществ по сравнению с выполнением экспериментов над реальной системой [www.xjtek.ru/consulting/what_is_simulation/].

Стоимость. Принятие обоснованного решения с помощью имитационной модели снижает затраты, которые состоят лишь из цены программного обеспечения и стоимости консалтинговых услуг.

Время. В реальности оценить эффективность, например, телекоммуникационной сети можно лишь через месяцы эксплуатации. Имитационная модель позволяет определить оптимальность таких изменений за считанные минуты, необходимые для проведения эксперимента.

Повторяемость. С помощью имитационной модели можно провести неограниченное количество экспериментов с разными параметрами, чтобы определить наилучший вариант.

Точность. Традиционные расчетные математические методы требуют применения высокой степени абстракции и не учитывают важные детали. Имитационное моделирование позволяет описать структуру системы и её процессы в естественном виде, не прибегая к использованию формул и строгих математических зависимостей.

Наглядность. Имитационная модель обладает возможностями визуализации процесса работы системы во времени, схематичного задания её структуры и выдачи результатов в графическом виде.

Универсальность. Имитационное моделирование позволяет решать задачи из любых областей: науки и техники. В каждом случае модель имитирует, воспроизводит, реальную жизнь и позволяет проводить широкий набор экспериментов без влияния на реальные объекты.

Применение метода имитационного моделирования в области изучения свойств вычислительных систем (ВС) и многомашинных

ассоциаций является обычной практикой. Поэтому учебно-методические материалы, ориентированные на выработку навыков имитационного моделирования ВС, актуальны для подготовки бакалавров по направлению «Информатика и вычислительная техника» и специалистов по специальностям «Автоматизированные системы обработки информации и управления» и «Программное обеспечение вычислительной техники и автоматизированных систем».

ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ (IEEE Std 610.12-1990)

Simulation modeling — программная модель системы, имитирующая ее поведение во времени, когда наблюдение за поведением обеспечивается наблюдением за изменением состояний системы или ее компонент, инициируемом соответствующими множествами входных и выходных сигналов или сообщений, которыми обмениваются компоненты, система и окружение

Emulation modeling — модель системы на основе аппаратного аналога, имитирующая поведение во времени при поступлении сигналов окружения, обеспечивающая наблюдение за изменением состояний

Диаграммы События-время (Waveform) — отображают последовательности наблюдаемых сигналов или сообщений во времени.

Особые события (ОС) --- события, изменяющие состояние системы.

Методы фиксации особых событий:

- **Метод дельта-Т** — изменение и фиксация состояний объектов моделирования осуществляется с постоянным шагом (тактом) модельного времени.
- **Метод особых состояний** — изменение и фиксация состояний объектов моделирования осуществляется с переменным шагом, определяемым ближайшим от текущего момента (модельного) времени, характеризующимся изменением состояния объектов моделирования (рис.1)

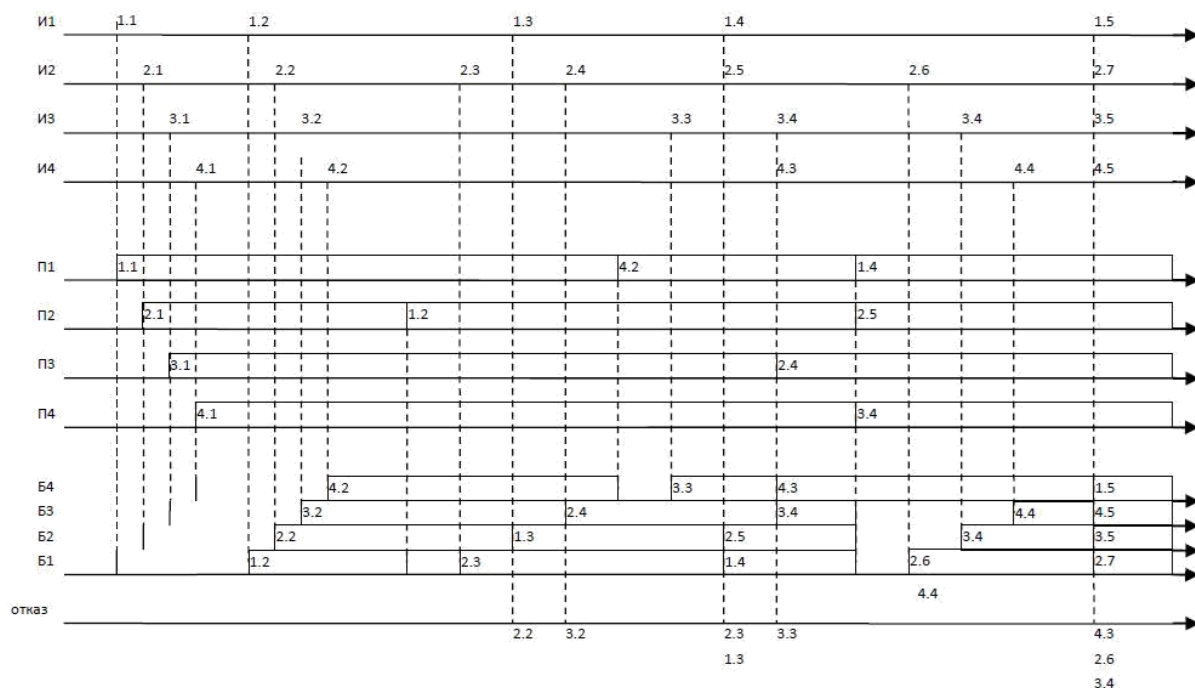


Рис.1 Пример временной диаграммы для метода особых состояний

На рис.1 отображены последовательности событий, поступающих от источников $И_n$ на приборы $П_m$, последовательность накопления событий в накопителе $Б_u$, последовательности фиксируемых событий в методе особых состояний, последовательность событий отказа и такты модельного времени. В методе дельта-Т, где события фиксируются каждый такт, учитывается много не изменяющихся состояний системы.

1. ОСНОВЫ ФОРМАЛЬНОЙ МОДЕЛИ ФУНКЦИОНИРОВАНИЯ ВС

Целью практической работы является создание модели ВС или ее компоненты на некотором уровне детализации, описывающей и имитирующей ее структуру и функциональность.

Каждый реальный объект ВС обладает огромной сложностью, определяемой множеством состояний, множеством внутренних и внешних связей, множеством анализируемых характеристик. Модель дает приближенное описание объекта с целью получения требуемых результатов с определенной точностью и достоверностью. Степень приближения модели к описываемому объекту может быть различной и зависит от требований задачи.

Существуют различные типы моделей ВС: аналитические, аналоговые, физические и имитационные. Последний тип моделей является предметом нашего изучения.

Одним из подходов к построению имитационной модели является построение ее в виде системы массового обслуживания (СМО) (рис.2), с характерной для СМО терминологией:

- источник (И);
- буфер (БП);
- прибор (П);
- диспетчер (Д);
- заявка (требование).

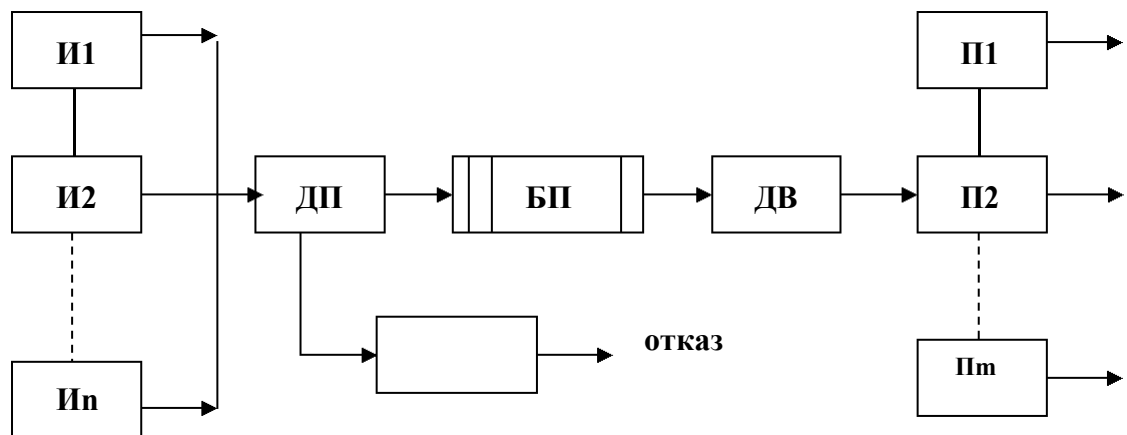


Рис. 2 Пример структуры системы массового обслуживания

На рис. 2 $И_i$ ($i=1...n$) — источник заявок, который генерирует заявки, а все вместе n источников создают входной поток заявок в систему. Источники могут быть двух типов — конечные и бесконечные. Различие лишь в способе генерации заявок.

Первый тип источников — «бесконечный». Принцип его работы состоит в следующем: источник генерирует заявку, а затем определяет интервал (по детерминированному или случайному закону) для генерации следующей заявки. Таким образом, заявка попадает в систему в момент генерации и проходит по ней свой индивидуальный путь.

Второй тип источников называется «конечным». Принцип его работы следующий: источник в определенный момент генерирует и отправляет в систему пакет заявок, состоящий из конечного числа заявок, каждая из которых тоже проходит свой индивидуальный путь по ВС. Момент генерации следующего пакета определяется событием, когда последняя заявка этого пакета удаляется из системы после обслуживания или в результате отказа. Затем пакет отправляет

источнику, который его сгенерировал, сообщение об окончании обслуживания. Это событие инициирует генерацию следующего пакета. Момент генерации определяется случайным интервалом между заявками. Новый сгенерированный пакет состоит из того же конечного числа заявок.

- П — приборы, которые обслуживают заявки и создают выходной поток заявок после обслуживания.
- БП — буферная память (место для хранения очереди заявок).

Буферная память может быть двух типов: общая и зонная.

В общей памяти хранятся заявки от различных источников. Порядок их записи в БП определяется только дисциплиной буферизации.

Зонная память представляет собой буфер, разделенный на зоны, в каждую из которых записываются заявки только от конкретного источника. Таким образом, количество зон соответствует числу источников.

- ДП — диспетчер постановки заявок в очередь.
- ДВ — диспетчер выбора заявок из очереди.

Функционирование ВС моделируется через прохождение заявки, сгенерированной источником, по системе. Проследим путь заявки, вошедшей в модель системы:

1. Постановка заявки в буфер;
2. Отказ или выбивание (удаление) заявки из переполненного буфера;
3. Выбор заявки из БП на обслуживание;
4. Поиск свободного прибора;
5. Обслуживание заявки прибором;
6. Выход заявки из СМО.

При этом источники только генерируют заявки, приборы их только обслуживают, а логика прохождения заявок по системе определяется диспетчерами ДП и ДВ.

- ДП — диспетчер постановки, выполняет две задачи:

- отправляет заявку на обслуживание или в буферную память в случае отсутствия свободных приборов;
- организует отказ или выбивание заявки из БП, если в буфере не осталось свободных мест.
- ДВ — диспетчер выбора, выполняет две задачи:
 - выбирает прибор, на котором обслуживает заявку;
 - выбирает заявку из БП, если она там есть.

Более подробно функции диспетчеров ДП И ДВ приведены в Приложении 3.

Рассмотренный принцип функционирования ВС основан на имитации единого многомерного процесса обработки заявок. Процессы моделируются через последовательность состояний системы во времени.

Моделирующий алгоритм позволяет имитировать поведение, т. е. изменение во времени состояний объектов и событий, составляющих исследуемый процесс.

2. ПРИНЦИПЫ ПОСТРОЕНИЯ МОДЕЛИРУЮЩЕГО АЛГОРИТМА.

Существуют два подхода к построения моделирующего алгоритма ВС для описанных двух методов моделирования:

1. Подход Дельта-Т.

Универсальный метод построения моделирующего алгоритма, когда состояние объекта проверяется через фиксированный интервал модельного времени. Суть его заключается в следующем: в каждый момент времени $t_i = t_{i-1} + \Delta t_{i-1}$ получают приближенные значения характеристик исследуемого объекта. Δt можно получить детерминированным способом.

Основной критерий выбора Δt — он должен быть настолько мал, чтобы не пропустить событие в моделируемой системе, которое

должно быть учтено при выбранной детальности моделирования. Метод неэффективен, т. к. при его использовании постоянно проверяется состояние объектов моделирования, не изменяющихся при этом, особенно при малых Δt .

2. Подход особых событий.

При исследовании реальной системы интервалы, в которых состояние не меняется, не представляют интереса. Имеют значение только значимые (изменяющие состояние) переходы системы в некоторые моменты времени. Эти переходы определяются особыми состояниями или событиями.

Рассмотрим некоторые типы особых событий, которые изменяют состояние системы:

1. Поступление заявки в СМО (момент генерации заявки источником);
2. Освобождение прибора (готовность прибора взять заявку на обслуживание);
3. Окончание процесса моделирования, т. е. момент прекращения генерации заявок источниками.

Использование принципа особых состояний для построения имитационной модели ВС наиболее эффективно. В настоящей работе предлагается использовать именно этот принцип (Приложение).

Моделирование работы ВС (рис.2) существенным образом зависит от характеристик элементов системы и их свойств, т. е. от параметров системы.

Параметрами моделируемой ВС являются:

1. Источники (количество и тип);
2. Закон генерации заявок источниками (детерминированный или случайный);
3. Приборы (количество);
4. Закон обслуживания заявок приборами (детерминированный или случайный);
5. Емкость и тип буферной памяти (БП);

Каждая заявка, сгенерированная источником, имеет свои характеристики:

$T_{ВХ}$ — время генерации заявок и N — номер заявки, составленный из номера источника, сгенерировавшего заявку, и порядкового номера заявки, сгенерированной этим источником. Например, заявка с номером (2,3) — третья заявка от второго источника.

3. ОБЩАЯ ПОСТАНОВКА ЗАДАЧИ НА ПРАКТИЧЕСКУЮ РАБОТУ.

Индивидуальное задание на работу представляется формализованным описанием вычислительной системы в терминах СМО в виде некоторой закодированной строчки — формулы задания (Приложение 1). Информация, необходимая для расшифровки формулы, приведена в Приложениях 2 и 3.

3.1. ФОРМУЛИРОВКА ОБЩЕЙ ПОСТАНОВКИ ЗАДАЧИ.

Исходные данные к работе:

1. Характеристика входного потока:

- 1.1. Тип источников
- 1.2. Количество источников
- 1.3. Закон генерации заявок
- 1.4. Среднее время между заявками
- 1.5. Приоритеты входных потоков

2. Характеристики ВС:

2.1. Буферная память:

- 2.1.1. Тип буфера
- 2.1.2. Объем буфера (количество мест хранения заявок)

2.2. Обслуживающие приборы:

- 2.2.1. Количество приборов
- 2.2.2. Закон обслуживания

- 2.2.3. Среднее время обслуживания
- 2.3. Дисциплины диспетчеризации:
 - 2.3.1. Дисциплины буферизации
 - 2.3.2. Дисциплина отказа (выбивания)
 - 2.3.3. Дисциплина выбора заявок на обслуживание
 - 2.3.4. Дисциплина занятия устройств
- 3. Характер текущей отображаемой информации (пошаговый режим):
 - 3.1. календарь событий, буфер, текущее состояние,
 - 3.2. формализованная схема модели, текущее состояние,
 - 3.3. временные диаграммы, текущее состояние.
- 4. Представление окончательных результатов моделирования (автоматический режим):
 - 4.1. сводная таблица результатов,
 - 4.2. основные графики (графики зависимости основных входных характеристик моделируемой системы от изменения перечисленных выше входных параметров).

3.2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ РАБОТЫ И ИССЛЕДУЕМЫМ ХАРАКТЕРИСТИКАМ МОДЕЛИРУЕМОЙ ВС.

В работе требуется построить моделирующий алгоритм ВС, реализовать его на языке программирования, отладить модель и представить в отчете результаты, полученные с относительной точностью 10 % и доверительной вероятностью 0,9, сопровождаемые анализом, графиками и выводами.

Таким образом, необходимо получить и проанализировать следующие характеристики ВС в зависимости от общей загрузки системы (ρ):

1. количество сгенерированных каждым источником требований;
2. вероятность отказа в обслуживании заявок (требований) каждого источника ($P_{отк}$);

3. среднее время пребывания заявок каждого источника в системе ($T_{вс}$);
4. среднее время ожидания заявок каждого источника в системе ($T_{ож}$);
5. среднее время обслуживания заявок каждого источника ($T_{обсл}$);
6. дисперсии двух последних характеристик;
7. коэффициент использования устройств ($K_{исп}$) (время работы каждого прибора / время реализации).

Эта часть задания относится к исследованию формально заданной ВС, у которой значения входным характеристикам задаются произвольно и определяются требованиями к выходным характеристикам ВС.

Следующим этапом работы является процесс исследования конкретной вычислительной системы (или ее компоненты) на полученной ранее модели с заданным уровнем доверительной вероятности и точности результатов.

На основе этого исследования, ориентируясь на предъявляемые требования к результатам работы системы, т. е. к ее выходным характеристикам, необходимо найти нужную архитектуру системы, состоящую из ее элементов и работающую в соответствии с заданными требованиями. Элементами ВС в нашем случае являются ее входные параметры:

- количество источников;
- интенсивность работы источников (либо среднее время между заявками);
- количество приборов;
- интенсивность обслуживания (либо среднее время обслуживания заявки);
- количество мест в буферной памяти.

4. ЭТАПЫ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ, ОТЧЁТНОСТЬ ПО ЭТАПАМ И КОНТРОЛЬНЫЕ ВОПРОСЫ.

Выполнение работы условно поделено на четыре этапа. Завершение каждого из них фиксируется представлением результатов, которые используются для ежемесячной аттестации.

4.1 ПЕРВЫЙ ЭТАП ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ.

Для получения зачёта по первому этапу необходимо:

1. расшифровать формулу своего варианта задания (см. Приложения 1,2,3),
2. освоить функционирование СМО на уровне формализованной схемы (рис.2),
3. иметь чёткое представление о типах источников и принципах построения моделирующего алгоритма,
4. понять функционирование дисциплин постановки, выбора и обслуживания заявок для заданного варианта задания (Приложение 3),
5. Построить временную диаграмму для заданных дисциплин (Приложение 2).

Форма отчётности по первому этапу.

1. Распечатанный бланк задания с заполненными исходными данными к работе.
2. Формализованная схема ВС.
3. Временная диаграмма своего варианта.
4. Краткие ответы на контрольные вопросы задания.

На временной диаграмме необходимо показать:

- моменты постановки заявок на приборы,
- заполнение буферной памяти по заданной дисциплине постановки заявки в буфер,

- отказ заявке или выбивание её при отсутствии свободных мест в БП,
- функционирование дисциплины выбора заявок из буфера и дисциплины выбора приборов.

Пример временной диаграммы приведён в Приложении 2.

Для определения вероятной возможности переполнения БП, что приводит к отказам или выбиванию заявок, используется оценка интенсивности генерации и обслуживания заявок.

Интенсивность генерации заявок измеряется количеством заявок, сгенерированных источниками за единицу времени, а интенсивность обслуживания заявок — количеством заявок, обслуженных прибором за единицу времени.

Очевидно, что для появления отказов или переполнения буферной памяти, необходимо, чтобы суммарный входной поток заявок (количество заявок, вошедших в систему от всех источников в единицу времени) превышал суммарный выходной поток заявок (количество заявок, покинувших систему после обслуживания на приборах плюс заявки, ушедшие в отказ) за ту же единицу времени.

Таким образом, загрузку системы (ρ) можно определить как отношение суммарного входного потока к суммарному выходному за единицу времени.

$$\rho = \frac{\sum_{i=1}^n \lambda_i}{\sum_{i=1}^m \mu_i}, \quad (1)$$

где λ_i — интенсивность источников; μ_i — интенсивность приборов; n — количество источников; m — количество приборов.

Нужно учитывать, что формула (1) справедлива только для случая одной и той же единицы времени для входного и выходного потока, причем во входном потоке необходимо учитывать заявки, сгенерированные всеми источниками, а в выходном потоке заявки,

обслуженные всеми приборами плюс заявки, ушедшие из системы в отказ.

К сожалению, последнее утверждение по поводу выходного потока не учитывается в формуле (1), которая справедлива только для систем без потери заявок, т. е. таких, где все сгенерированные источниками заявки будут обслужены, ни одной из них не будет отказано в обслуживании.

Иначе говоря, формула (1) справедлива для систем с бесконечным буфером. Для реальных ВС модель бесконечного буфера возможна далеко не всегда, поэтому приходится использовать модели с отказами.

Поскольку в реальных ВС всегда возникает потребность в минимальном количестве отказов в обслуживании, то необходимо стремиться к созданию моделей таких систем, у которых входной и выходной потоки будут почти одинаковыми (сбалансированными) и их отношение станет близким к единице. В таких системах наряду с хорошей загрузкой отказов будет немного и буферная память потребуется небольшая.

Менять загрузку системы для того, чтобы моделировать отказы, можно различными способами: изменяя количество источников, их интенсивности, количество приборов и интенсивности обслуживания. В результате путем увеличения или уменьшения входного и выходного потоков можно изменять и анализировать загрузку системы, приближая ее значения к единице.

Контрольные вопросы к первому этапу курсовой работы.

1. Назовите типы источников, опишите принципы их работы, различия между ними.
2. Можно ли сказать, что бесконечный источник есть частный случай конечного?
3. Опишите два принципа построения моделирующего алгоритма, их преимущества и недостатки.

4. Опишите дисциплины буферизации и постановки заявки на обслуживание, заданные в вашем варианте.

5. Назовите некоторые варианты (комбинации) значений входных параметров, при которых на представленной временной диаграмме могут появиться отказы из БП и будут хорошо проиллюстрированы дисциплины выбора приборов и выбора заявок. Рекомендуемый вид временной диаграммы представлен в Приложении 2.

4.2 ВТОРОЙ ЭТАП ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ.

Этап состоит в разработке и отладке программы, моделирующей поведение заданной конкретной ВС с определенной точностью. Пример программной реализации одной из возможных моделей ВС представлен в Приложении 4.

Все варианты задания предусматривают относительную точность результатов моделирования, равную 10 %, и доверительную вероятность, равную 0,9. Для того, чтобы выполнить эти условия, потребуется определить необходимое и достаточное количество заявок, которое должно быть сгенерировано источниками. Очевидно, что чем больше заявок пройдет по системе, тем с большей точностью будут получены результаты — выходные характеристики.

Существует количественная связь между количеством реализаций (количество заявок, проходящих через ВС), относительной точностью, достоверной вероятностью и случайной величиной $p(A)$ — вероятностью некоторого события A . Эта связь выражается формулой

$$N = \frac{t^2(1-p)}{p\delta^2}, \quad (2)$$

Где p — вероятность отказа заявкам в обслуживании; $t_\alpha = 1,643$ для $\alpha = 0,9$; $\delta = 0,1$ — относительная точность.

Формула представляет собой уравнение с двумя неизвестными (N и p). Поскольку искомой величиной является N , то необходимо иметь

представление о значении p . Для этого обычно производят приблизительную оценку («пристрелку»), назначая какое-либо значение N (например, $N_0 = 100$), с которым проводят процесс моделирования, т.е. через систему пропускают 100 заявок, получают на выходе программной модели рассчитанные выходные характеристики, в этом числе и p_0 , которое подставляют в формулу (2) и получают N_1 , с которым снова проводят процесс моделирования.

Полученное на этом этапе значение p_1 сравнивают со значением p_0 . Если разница $|p_0 - p_1|$ меньше 10 % от значения p_0 , то $N=100$ удовлетворяет заданной точности результатов. Если же $|p_0 - p_1| \geq 10\%$ от p_0 , то процесс моделирования продолжается с новым N_2 и т. д. до достижения необходимой точности.

Весь этот итерационный процесс нахождения необходимого количества заявок нужно реализовать в автоматическом режиме. После окончания моделирования на экран выдается таблица результатов, полученных с заданной точностью.

Форма отчётности по второму этапу.

В каждом индивидуальном задании предусмотрены два вида отражения работы программной модели:

1. Отображение динамики функционирования модели в пошаговом режиме.
2. Отображение результатов работы программной модели в автоматическом режиме.

В п. 1 необходимо отразить поведение ВС, т.е. изменение состояния системы при каждом наступлении особого события. Фиксация состояний осуществляется в пошаговом режиме, где шагом является расстояние по времени от одного особого события до ближайшего другого. В задании указан один из трёх видов отражения работы программной модели в пошаговом режиме:

- календарь событий,
- формализованная схема,
- временная диаграмма.

В п. 2 необходимо продемонстрировать результаты работы программной модели ВС в автоматическом режиме.

В процессе работы программной модели при прохождении по ВС каждой заявки происходит сбор статистической информации для расчёта следующих характеристик системы:

- Количество заявок, сгенерированных каждым источником
- Вероятность отказа в обслуживании заявок каждого источника

$$p = \frac{m}{n},$$

где n — общее количество заявок, сгенерированных источником;
 m — количество заявок этого источника, получивших отказ.

- Среднее время пребывания заявки каждого источника в системе.

$$T_{\text{преб}} = T_{\text{обсл}} + T_{\text{БП}},$$

где $T_{\text{преб}}$ — среднее время пребывания заявки в системе (время ответа на запрос); $T_{\text{обсл}}$ — среднее время обслуживания заявки данного источника; $T_{\text{БП}}$ — среднее время пребывания заявки в БП или. среднее время ожидания заявки каждого источника.

- Дисперсии $T_{\text{обсл}}$ и $T_{\text{БП}}$;
- Коэффициенты использования приборов (суммарное время занятости каждого прибора/общее время реализации).

После завершения процесса моделирования должны быть получены две таблицы результатов:

Таблица 1

Характеристики источников ВС.							
№ источника	КОЛИЧЕСТВО заявок	p отк	$T_{\text{преб}}$	$T_{\text{БП}}$	$T_{\text{обсл}}$	$D_{\text{БП}}$	$D_{\text{обсл}}$
И1							
И2							
...							
Иm							

Таблица 2

Характеристики приборов ВС.

№ прибора	Коэффициент использования
-----------	------------------------------

П1

П1

...

Пк

Окончание процесса генерации заявок (конец моделирования) происходит в момент генерации последней заявки. Но в СМО могут остаться заявки как на приборах, так и в буферной памяти, поэтому процесс обслуживания заявок продолжается до момента выхода из системы последней заявки.

Время, которым заканчивается обслуживание последней заявки, называется общим временем реализации.

Его значение используется при расчете коэффициентов использования приборов.

Контрольные вопросы по второму этапу.

1. Какие изменения состояния ВС должна фиксировать программа для вашего задания в пошаговом режиме?
2. Что должно являться шагом при работе программной модели в режиме динамического отражения результатов?
3. Из чего складывается время пребывания заявки в системе и как рассчитывать среднее время пребывания заявки в системе?
4. Как рассчитать коэффициент использования приборов $K_{\text{исп}}$)?
5. Объясните разницу между окончанием моделирования и общим временем реализации.
6. Какие сведения должны быть выведены на экран по истечении времени реализации?

Критерием успешного выполнения этапа является демонстрация работы моделирующей программы в двух режимах (пошаговом и

автоматическом) и соответствие значений характеристик, полученных при моделировании, ожидаемым значениям.

4.3. ТРЕТИЙ ЭТАП ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ.

На этом этапе необходимо привести пример реальной ВС, которая полностью соответствовала бы варианту задания. В концептуальном описании примера ВС необходимо показать, что ни одна из дисциплин задания, ни случайные законы генерации и обслуживания заявок не противоречат дисциплинам и законам, используемым в примере реальной ВС.

Примером системы может быть вполне конкретная, существующая ВС или гипотетическая система, но имеющая право на существование. Непременным условием при выборе такой системы является то, что приборами, обслуживающими заявки, должны быть исключительно средства вычислительной техники, т. е. предлагаемая для исследования система должна быть не просто системой массового обслуживания, а обязательно вычислительной системой.

На этом этапе необходимо иметь представление о количественных значениях входных параметров, т. е. представление о моделируемой системе должно быть достаточно четким и конкретным.

Задача третьего этапа — синтез конкретной вычислительной системы с заданными выходными характеристиками. Все элементы синтезируемой системы должны быть подробно описаны, и здесь важно сформулировать требования к результатам работы.

Эффективность работы конкретной ВС будем определять по трем ее основным выходным характеристикам:

- вероятность отказа в обслуживании заявок;
- $P_{\text{отк}}$ — среднее время пребывания заявок в СМО;
- $\frac{I}{K_{\text{исп}}}$ — коэффициент использования приборов.

Задавая желаемые количественные значения этим трем выходным характеристикам, мы формируем требования к работе создаваемой ВС.

Имея представление о том, как должна работать эта система, можно определить и конфигурацию такой системы через выбор конкретных значений ее входных параметров. Иначе говоря, одной из задач этого этапа является определение того, что нужно подать на вход программной модели, чтобы на выходе получить затребованные выходные характеристики.

Этого можно добиться, задавая различные комбинации входных параметров. Вариантов таких комбинаций может оказаться бесконечно много, поэтому на множество значений входных параметров придется наложить некоторые ограничения, исходя из особенностей конкретных задач, для которых предназначена ВС. Таким образом, необходимо спроектировать модель архитектуры системы, которая будет работать по заданным требованиям.

Форма отчетности по третьему этапу.

1. Демонстрация действующей программной модели (в пошаговом и автоматическом режимах). Возможность задания различных комбинаций входных параметров.

2. Интерпретация поведения выходных характеристик системы при различных значениях загрузки системы и отбор вариантов систем, работающих по заданным требованиям.

Контрольные вопросы по третьему этапу.

1. Опишите элементы СМО для примера своего варианта ВС по следующей схеме:

Источник — тип и характеристики.

Прибор — количество и характеристики.

Буфер — тип и характеристики.

Заявка — транзакция конкретного размера.

2. Соответствуют ли случайные законы и дисциплины постановки и выбора вашей ВС законам и дисциплинам, указанным в строчке задания вашего варианта?

3. Значениями каких выходных характеристик определяется качество работы системы?

4. Составьте требования к работе ВС (значения, которые вы хотите получить для выходных характеристик).

5. Напишите ограничения (область изменения) для всех пяти входных параметров исследуемой ВС.

Критерием окончания этапа является представление результатов, демонстрирующих полное соответствие моделируемой ВС варианту задания.

4.4 ЧЕТВЕРТЫЙ ЭТАП ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ.

Этот этап посвящен исследованию работы модели ВС и выбору ее оптимальной конфигурации.

Для разных комбинаций входных параметров необходимо продемонстрировать на модели соответствующее поведение системы, т.е. отследить изменения её выходных характеристик. Если полученные в результате моделирования значения этих характеристик будут удовлетворять составленным ранее требованиям к работе системы (всем трем одновременно), следовательно, такая комбинация входных параметров (конфигурация системы) является одним из решений поставленной задачи. Таких решений на выбранных областях изменения входных параметров может быть очень много. Задача заключается в том, чтобы найти все возможные сочетания этих параметров, т. е. все конфигурации системы, удовлетворяющие сформулированным требованиям. Этого можно добиться, если правильно выбрать шаг изменения параметра внутри заданной области существования этого параметра.

В исследуемой ВС параметры могут меняться как дискретно, так и непрерывно. К дискретно изменяющимся параметрам могут относиться такие входные параметры как количество источников,

количество приборов, количество мест в буферной памяти, а параметры случайных законов генерации и обслуживания заявок могут быть как первого, так и второго типов, т. е. могут меняться как дискретно, так и непрерывно.

В первом случае изменять значения входных параметров можно просто перебирая все их значения на заданном отрезке.

В случае же непрерывно изменяющихся параметров оптимальный шаг приходится подбирать, отслеживая характер изменения выходных характеристик. Если при изменении входного параметра на некоторую величину выходные характеристики изменяются меньше, чем на 10 % от предыдущего значения (заданная относительная точность результатов), то шаг можно увеличить. В противном случае, если выходные характеристики изменяются резко и по ним сложно построить плавную кривую, то шаг постепенно уменьшают. Исходя из этого, будем стремиться к такому шагу изменения, который даст приблизительно 10 %-е изменение выходных характеристик. Если же различные выходные характеристики изменяются по-разному, то в таком случае есть смысл выбрать минимальный шаг. В результате можно получить некоторое множество решений исследуемой системы, т.е. некоторое количество конфигураций входных параметров, которые будут соответствовать сформулированным требованиям.

На этом же этапе требуется определять количество заявок, которое должно пройти по системе, чтобы результаты моделирования, т. е. выходные характеристики, были получены с заданной точностью и достоверной вероятностью. Кроме того, из некоторого полученного множества решений (вариантов архитектуры ВС) необходимо на основе анализа результатов отобрать оптимальный вариант системы.

Форма отчетности по четвертому этапу.

1. Отчет по работе, таблицы результатов, графики изменения выходных характеристик в зависимости от изменяющейся конфигурации системы.
2. Анализ результатов, выводы и рекомендации по выбору оптимальной конфигурации системы.

Требования к отчету и его содержанию.

1. Постановка задачи.
2. Формализованная схема и описание СМО.
3. Пример временной диаграммы функционирования системы для своего варианта задания.
4. Пример технической системы (ВС или части ВС), удовлетворяющей формализованному описанию и описанию ее элементов.
5. Документация на ПО:
 - 5.1. Обобщенная блок-схема.
 - 5.2. Модульная структура.
 - 5.3. Описание модулей.
6. Результаты работы имитационной модели в виде таблиц и графиков.
 - 6.1. Определение количества реализаций.
 - 6.2. Анализ результатов, выводы и рекомендации по выбору конфигурации системы.

При анализе результатов необходимо пояснить причины выбора полученного решения.

Чем руководствоваться при этом?

Рассмотрим множество возможных решений, которые дают требуемые результаты. Оценим качество работы полученных систем. Они будут работать наиболее эффективно в случае, если максимальных результатов удастся добиться минимально возможными средствами. В нашем случае этими средствами (ресурсами) являются количество мест в буфере, количество приборов и их производительность, т. к. именно эти параметры формируют выходной поток обработанных заявок.

Такой результат достигается при общей загрузке системы около 1, т. е. при относительном равновесии входного и выходного потоков. Допустимы отклонения, свойственные реальным системам, от оптимальной загрузки системы на 20-25 % .

Поскольку формула расчета общей загрузки справедлива для систем с бесконечным буфером, иначе говоря, для систем без отказа заявок в обслуживании, то ее следует применять для систем, где эффективность работы определяется стремлением к нулю вероятности отказов.

Для каждого из полученных решений системы необходимо рассчитать загрузку. Среди этих систем выберем ту, загрузка которой ближе других к единице. Таким образом, мы выбираем систему, ориентируясь на более высокую эффективность ее работы. Если же существуют несколько вариантов таких систем, то следующим критерием отбора станет отбор по минимуму использования ресурсов (количество приборов, интенсивность их работы и количество мест в буфере), а это, как правило, предполагает минимальную цену на эти ресурсы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Гмурман В.Е. Теория вероятностей и математическая статистика : учеб. пособие для вузов / В. Е. Гмурман .— 12-е изд., перераб. — М. : Юрайт, 2010 .— 478 с.
2. Бусленко Н.П. Моделирование сложных систем. М.: Наука, 1978.
3. Бутомо И.Д., Дробинцев Д.Ф., Котляров В.П. Методы имитационного моделирования вычислительных систем: Учебное пособие, -Л.: ЛПИ, 1979 – 72с.
4. ГОСТ 24.702-85. Эффективность АСУ. Основные положения. М. : Издательство стандартов, 1985.

ПРИЛОЖЕНИЯ

Формулы заданий на курсовую работу «Создание имитационной модели вычислительной системы»

1.	ИБ	ИЗ1	ПЗ2	Д1031	Д10О1	Д2П1	Д2Б1	ОР2	ОД1	
2.	ИБ	ИЗ2	ПЗ1	Д1031	Д10О3	Д2П2	Д2Б2	ОР2	ОД2	
3.	ИБ	ИЗ1	ПЗ2	Д1031	Д10О3	Д2П1	Д2Б3	ОР1	ОД3	
4.	ИБ	ИЗ2	ПЗ1	Д1031	Д10О4	Д2П2	Д2Б4	ОР2	ОД1	
5.	ИБ	ИЗ1	ПЗ2	Д1031	Д10О1	Д2П1	Д2Б5	ОР1	ОД2	
6.	ИБ	ИЗ2	ПЗ1	Д1032	Д10О2	Д2П2	Д2Б5	ОР1	ОД3	
7.	ИБ	ИЗ1	ПЗ2	Д1032	Д10О3	Д2П1	Д2Б2	ОР1	ОД1	
8.	ИБ	ИЗ2	ПЗ1	Д1032	Д10О4	Д2П2	Д2Б4	ОР1	ОД2	
9.	ИБ	ИЗ1	ПЗ2	Д1032	Д10О5	Д2П1	Д2Б5	ОР1	ОД3	
10.	ИБ	ИЗ2	ПЗ1	Д1033	Д10О2	Д2П2	Д2Б5	ОР1	ОД1	
11.	ИБ	ИЗ1	ПЗ2	Д1033	Д10О3	Д2П1	Д2Б2	ОР1	ОД2	
12.	ИБ	ИЗ2	ПЗ1	Д1033	Д10О4	Д2П2	Д2Б3	ОР1	ОД3	
13.	ИБ	ИЗ2	ПЗ1	Д1033	Д10О5	Д2П1	Д2Б4	ОР1	ОД1	
14.	ИБ	ИЗ1	ПЗ2	Д1033	Д10О3	Д2П2	Д2Б5	ОР1	ОД2	
15.	ИБ	ИЗ1	ПЗ2	Д1031	Д10О2	Д2П1	Д2Б5	ОР1	ОД3	
16.	ИБ	ИЗ2	ПЗ1	Д1031	Д10О3	Д2П2	Д2Б2	ОР1	ОД1	
17.	ИБ	ИЗ1	ПЗ2	Д1031	Д10О4	Д2П1	Д2Б3	ОР1	ОД2	
18.	ИБ	ИЗ2	ПЗ1	Д1031	Д10О1	Д2П2	Д2Б4	ОР1	ОД3	
19.	ИБ	ИЗ1	ПЗ2	Д1031	Д10О1	Д2П1	Д2Б5	ОР1	ОД1	
20.	ИБ	ИЗ2	ПЗ1	Д1032	Д10О4	Д2П2	Д2Б1	ОР1	ОД2	
21.	ИБ	ИЗ1	ПЗ1	Д1031	Д10О3	Д2П1	Д2Б3	ОР2	ОД3	
22.	ИБ	ИЗ2	ПЗ1	Д1032	Д10О4	Д2П2	Д2Б4	ОР1	ОД1	
23.	ИБ	ИЗ1	ПЗ2	Д1032	Д10О5	Д2П1	Д2Б5	ОР1	ОД2	

24.	ИБ	ИЗ2	ПЗ1	Д1033	Д10О2	Д2П2	Д2Б5	ОР1	ОД3
25.	ИБ	ИЗ1	ПЗ2	Д1033	Д10О3	Д2П1	Д2Б2	ОР1	ОД1
26.	ИБ	ИЗ2	ПЗ1	Д1033	Д10О4	Д2П2	Д2Б3	ОР1	ОД2
27.	ИБ	ИЗ2	ПЗ1	Д1033	Д10О5	Д2П1	Д2Б4	ОР1	ОД3
28.	ИБ	ИЗ1	ПЗ2	Д1033	Д10О3	Д2П2	Д2Б5	ОР1	ОД1
29.	ИБ	ИЗ1	ПЗ1	Д1031	Д10О3	Д2П2	Д2Б3	ОР2	ОД2

Варианты отображения результатов работы имитационной модели (в пошаговом и автоматическом режимах).

Варианты отображения результатов работы модели в пошаговом режиме.

- 1. Календарь событий, буфер, текущее состояние.
- 2. Формализованная схема.
- 3. Временные диаграммы.

Варианты отображения результатов работы модели в автоматическом режиме.

- 4. Сводная таблица результатов.
- 5. Графики по значению параметров.

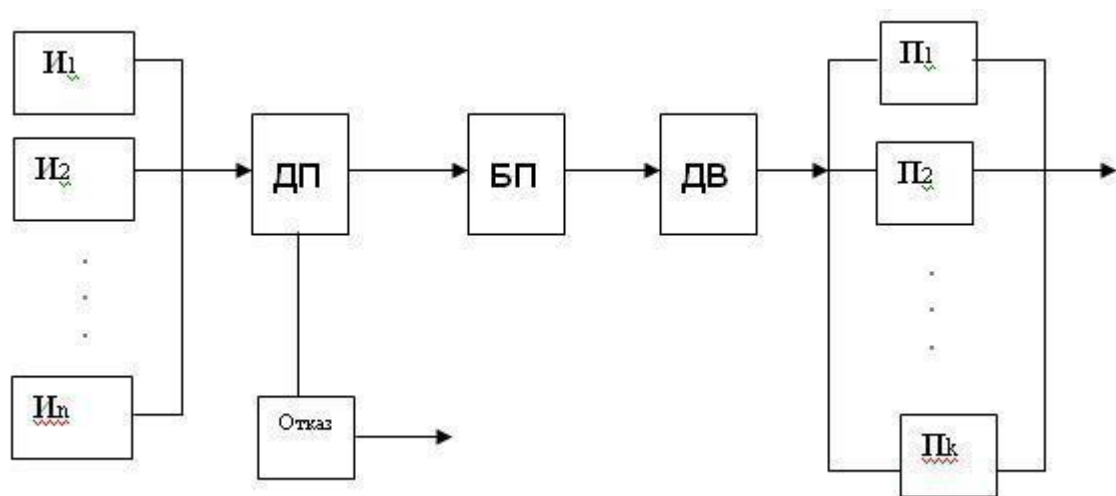
1. Календарь событий, буфер, текущее состояние

Календарь событий			Число заявок	Число отказов
Событие	Время	Признак		
И1	6,1	0	7	0
И2	6,3	0	3	0
И3	7,0	0	6	0
И4	6,4	0	7	0
И5	9,6	0	9	0
И6	8,0	0	5	0
И7	6,0	0	8	1
П1	6,3	0		
П2	8,8	0		
П3	6,5	0		
П4	6,3	0		
П5	7,1	0		
П6	6,3	0		
Конец мод.	999,0	0		

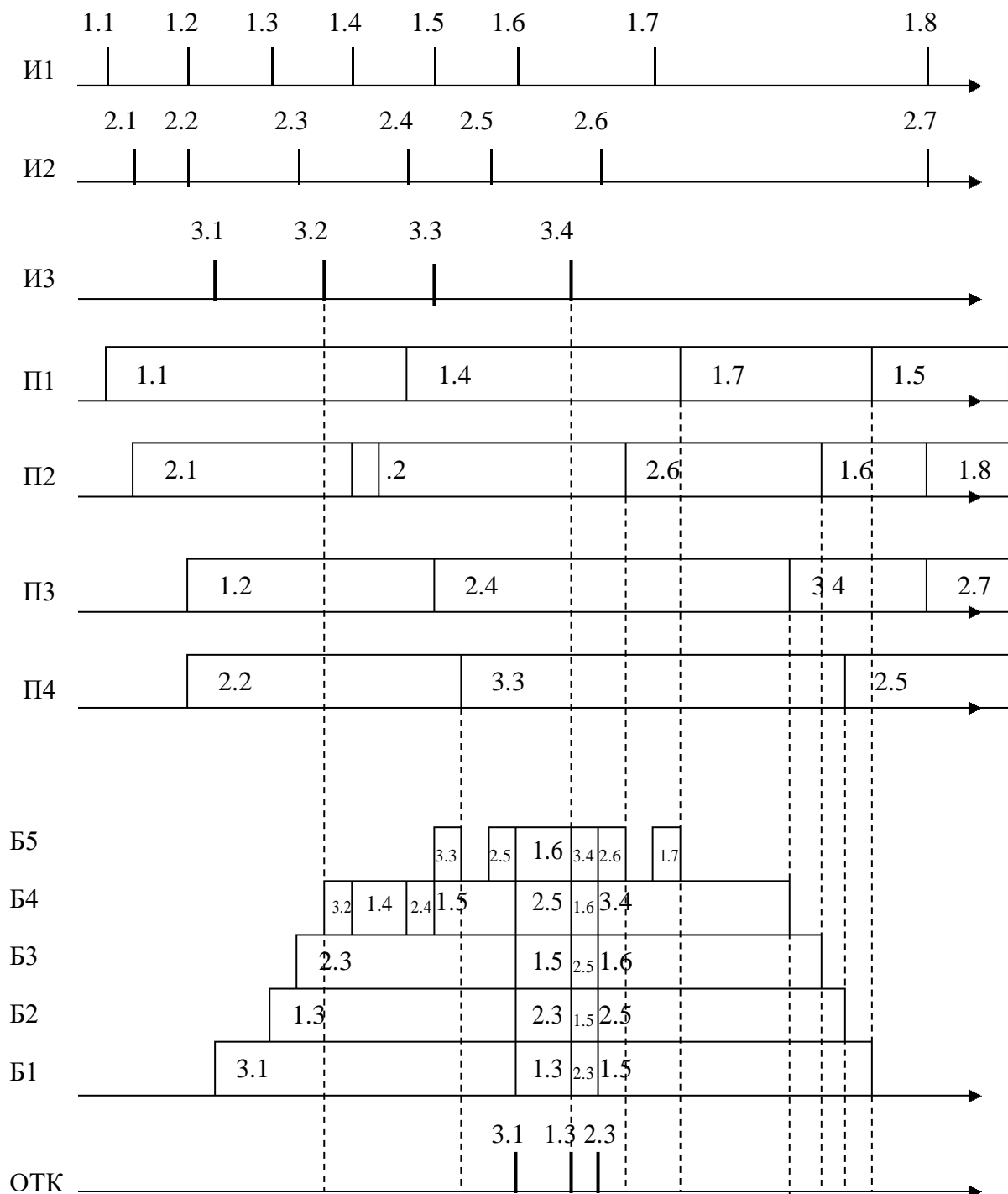
Буфер

Позиция	Время	Источник	Заявка
1	5,6	6	5
2	5,7	1	6
3	5,8	1	7
4	5,8	5	9
5	0,0	0	0
6	5,8	5	9

2. Формализованная схема



3. Временные диаграммы



Дополнительная информация к формуле задания.

- 1. Параметры элементов модели.
- 2. Описание дисциплин постановки и выбора.
- 3. Виды отображения результатов работы программной модели.

1. Параметры элементов модели.

1.1. Источники.

ИБ_n — бесконечный;

ИК_n — конечный;

ИЗ — законы распределения;

ИЗ1 — пуассоновский для бесконечных,
экспоненциальная задержка для конечных;

ИЗ2 — равномерный;

1.2. Приборы.

П_n — приборы;

ПЗ — законы распределения времени
обслуживания; ПЗ1 — экспоненциальный; ПЗ2 —
равномерный;

2. Описание дисциплин постановки и выбора.

Описание дисциплин (функции диспетчеров).

Буферизации Д1О31-Д1О33

Отказа Д1О01-Д1О05

Выбора заявки на обслуживание Д2Б1-Д2Б5

Выбора обслуживающего прибора Д2П1-Д2П2

2.1. Дисциплины буферизации.

Д1 — постановки в буфер;

Д1О — относительные приоритеты на обслуживание;

Д1ОЗ — записи в буфер, если есть
место; Д1ОЗ1 — по кольцу;

Д1О32 — в порядке поступления;

Д1О33 — на свободное место;

Д1О31 – заполнение буферной памяти «по кольцу».

При необходимости поставить заявку на очередь в БП поиск свободного места в буфере осуществляется, начиная с номера места, следующего за последним занятым.

Введем понятие «указателя», который будет принимать значение, равное номеру места, следующего за последним занятым в БП. Покажем это на примере таблицы:

Х	Х		Х	Х	Х		Х	Х	Х
1	2	3	4	5	6	7	8	9	10
УБ									

Буферная память имеет 10 мест. Свободны места 3 и 7. Указатель фиксирован на месте с номером 5, т. е. заявка, поставленная последней в буфер, занимает место с номером 4. В случае необходимости вновь поставить какую-либо из пришедших заявок в очередь, поиск свободного места в БП начинается с указателя, т. е. с 5-ого места. Указатель станет последовательно передвигаться по буферу и искать первое свободное место, на которое будет помещена заявка. Если такое место найдется (в нашем случае это 7), заявка займет его, а указатель станет равным номеру места БП, следующего за занятым (т. е. УБ=8).

Если же, передвигаясь по буферу до конца, свободного места не найдется, указатель вернется на начало БП и продолжит поиск (добежит до места с номером 3 и поставит заявку туда). При этом он примет значение 4.

Возможна ситуация, когда указатель, передвигаясь «по кольцу», не найдет свободного места в БП и вновь примет то же самое значение. В этом случае начинают действовать дисциплины, организующие отказ или выбивание заявки из БП.

Д1032 — постановка заявки в буфер в порядке поступления.

Если в момент поступления заявок в систему все приборы оказываются занятыми, заявка последовательно занимает места в буфере памяти, начиная с первого. В случае освобождения какого-либо места в БП с номером N (заявка уходит на обслуживание или получает отказ), все заявки, стоящие на местах, начиная с (N+1), сдвигаются на одно место. Следующая заявка, вынужденная встать в очередь, всегда будет ставиться в ее конец, пока есть свободные места.

Д1033 — постановка заявки на свободное место.

Заявка встанет в очередь на первое от начала свободное место, если такое найдется. Сдвига очереди в этом случае не происходит.

2.2. Дисциплины отказа.

Д100 — дисциплины отказа;

Д1001 — под указателем;

Д1002 — приоритет по номеру источника;

Д1003 — самая старая в буфере;

Д1004 — последняя поступившая в буфер;

Д1005 — вновь пришедшая;

Д1А — абсолютные приоритеты на обслуживание;

Д1А3х=Д1О3х, Д1АОх=Д1ООх.

Заявки могут получить отказ в обслуживании только в том случае, если к моменту прихода в систему очередной заявки все приборы и все места в буферной памяти окажутся занятыми. Тогда пришедшая заявка может либо сама уйти из системы (получить отказ), либо она имеет право занять место одной из заявок, стоящих в буферной памяти (выбить заявку из БП). <Права> и <возможности> этих заявок определяют дисциплины отказа.

Д1001 — отказ под указателем.

Такая дисциплина возможна только при записи в буфер «по кольцу». В этом случае указатель, пробежав всю буферную память и не найдя свободного места, останется на прежней отметке. Заявка,

стоящая на этом месте, получит отказ в обслуживании и уйдет из ВС, а на ее место встанет пришедшая заявка, после чего указатель передвинется на следующее место.

Д1002 — приоритет по номеру источника.

При этой дисциплине отказ получает заявка с наименьшим приоритетом среди тех, что на данный момент находятся в БП (приоритет заявки определяется номером источника, который её сгенерировал). Если к этому времени в буфере имеется несколько заявок от источника с минимальным приоритетом, то встаёт вопрос, какую из этих заявок отправить в отказ.

При этом необходимо вспомнить, что заявка имеет две характеристики: номер источника и время генерации. В нашем случае первая характеристика у этих заявок одинаковая и различаются они только временем генерации. Если в рассматриваемой ВС выбор заявки на обслуживание происходит по времени (FIFO, LIFO), то это может оказаться подсказкой для определения очередности при отказе.

Д1003 — самая старая в буфере.

Эта дисциплина рассматривает только время прихода заявок в систему (момент генерации заявок источником). Заявка, раньше других вставшая в буфер получает отказ, уходит из системы и на её место встает пришедшая заявка.

Д1004 — последняя поступившая в буфер.

Самая последняя заявка из поступивших в буфер, т. е. заявка, меньше других простоявшая в очереди, выбивается из БП, и на её место встаёт пришедшая заявка.

Д1005 — вновь пришедшая заявка.

Заявка, сгенерированная источником и не нашедшая свободного места в буфере, уходит из системы, не изменяя состояния буфера. При этом она обязательно учитывается при подсчете общего количества сгенерированных источником заявок.

2.3. Дисциплины постановки на обслуживание.

Д2 — постановки на обслуживание;
Д2Б — выбор заявки из буфера;
 Д2Б1 — FIFO;
 Д2Б2 — LIFO;
 Д2Б3 — по кольцу;
 Д2Б4 — приоритет по номеру источника, по одной
 заявке;
 Д2Б5 — приоритет по номеру источника, заявки в
 пакете;
Д2П — выбор прибора;
 Д2П1 — приоритет по номеру прибора;
 Д2П2 — по кольцу;

Освобождение прибора или его простой означает, что прибор готов взять заявку на обслуживание. Если в буфере есть очередь, то заявка поступает на прибор в момент его освобождения. Какую заявку поставить на обслуживание на освободившийся прибор определяют дисциплины выбора заявок.

2.3.1. Дисциплины выбора заявок на обслуживание.

Д2Б1 — FIFO (первым пришел — первым обслужен).

Заявка, дольше всех простоявшая в буфере, будет выбрана на обслуживание раньше других.

Д2Б1 — LIFO (последним пришел — первым обслужен).

В этом случае раньше других будет выбрана из буфера на обслуживание та заявка, которая пришла последней.

Выбор заявки из буфера «по кольцу».

Эта дисциплина выбора работает аналогично дисциплине постановки в БП «по кольцу». Здесь также вводится понятие «Указателя», который принимает значение номера места, с которого начинается поиск заявок в БП. Как только будет найдено место, занятое заявкой, эта заявка ставится на освободившийся прибор, а указатель передвигается на место, следующее за тем, откуда была

выбрана на обслуживание заявка. Поиск следующего занятого места буфера будет производиться также, начиная с указателя.

Д2Б4 — приоритет по номеру источника, по одной заявке.

Приоритет заявки определяется номером источника, её сгенерировавшего. Из буфера выбирается заявка, приоритет которой выше остальных. Если в буфере имеется несколько заявок самого высшего на данный момент приоритета, то обращают внимание на вторую характеристику заявок, т. е. на время их прихода в систему, а также на другие дисциплины, которые организуют, например, отказ заявок по времени поступления в буфер. Выбор заявки должен определяться очевидной логикой. Если в системе из БП уходят в отказ более старые заявки, то на обслуживание логично выбирать самые новые, т. е. среди заявок одного приоритета есть смысл выбрать последнюю поступившую.

Д2Б5 — приоритет по номеру источника, заявки в пакете.

Назовем «пакетом» совокупность заявок одного источника, находящихся в буфере на момент освобождения одного из приборов.

Количество пакетов в БП может меняться от 0 до n , где n — количество источников.

Когда при освобождении прибора происходит выбор первой заявки из буфера, вначале определяется самый приоритетный на данный момент пакет и происходит обслуживание заявок только этого пакета до тех пор, пока к моменту очередного освобождения прибора в БП не останется ни одной заявки этого пакета. Затем снова определяется самый приоритетный на данный момент пакет и далее повторяется весь процесс обслуживания этого пакета. Таким образом, происходит динамическая смена приоритетов обслуживания заявок, причем приоритетность пакетов можно регулировать, изменяя интенсивность генерации заявок источниками.

2.3.2. Дисциплины выбора прибора.

Д2П1 — приоритет по номеру прибора.

Приоритеты приборов, также как и приоритеты источников определяются номерами приборов. Поэтому поиск свободного прибора ведется последовательным перебором, каждый раз начиная с самого приоритетного.

Д2П2 — выбор прибора по кольцу.

Эта дисциплина производит выбор свободного прибора таким же способом, как и аналогичная дисциплины выбора заявок из буфера по кольцу, т. е. поиск свободных приборов каждый раз начинается с указателя и заявка встает на обслуживание на первый из найденных приборов.

3. Виды отображения результатов работы программной модели.

3.1. Динамическое отражение результатов (пошаговый режим).

ОД — отображение динамики функционирования модели;
ОД1 — календарь событий, буфер и текущее состояние;
ОД2 — формализованная схема модели, текущее состояние; ОД3 — временные диаграммы, текущее состояние.

Шаг в этом случае — интервал модельного времени от одного особого события до другого ближайшего по времени особого события.

3.2. Отражение результатов после сбора статистики ОР1-ОР2 (автоматический режим).

ОР — отображение результатов;
ОР1 — сводная таблица результатов;
ОР2 — графики по значениям параметров;

Поток случайных событий

Когда событий много и они следуют друг за другом, то они образуют *поток*.

Интенсивность потока λ — это среднее число событий в единицу времени. Интенсивность потока можно рассчитать экспериментально по формуле: $\lambda = N/T_n$, где N — число событий, произошедших за время наблюдения T_n .

Если интервал между событиями t_j равен константе или определен какой-либо формулой в виде: $t_j = f(t_j - 1)$, то поток называется **детерминированным**. Иначе поток называется **случайным**.

Случайные потоки бывают:

- **ординарные**: вероятность одновременного появления двух и более событий равна нулю;
- **стационарные**: частота появления событий $\lambda(t) = \text{const}(t)$;
- **без последствия**: вероятность появления случайного события не зависит от момента совершения предыдущих событий.

Пуассоновский поток

Случайный ординарный, без последствия поток, интервалы между двумя случайными событиями которого имеют вид:

$$\tau = -1/\lambda \cdot \text{Ln}(r),$$

где r — равномерно распределенное от 0 до 1 случайное число, которое берут из ГСЧ,

τ — интервал между случайными событиями (случайная величина τ_j).

Санкт-Петербургский политехнический университет Петра Великого
ИКНТ
Высшая школа ПИ

Отчет по работе
“Методы имитационного моделирования”

Выполнил

студент гр.-----

ФИО

Руководитель

Дробинцев Д.Ф.

2022 г.

Постановка задачи

Целью курсовой работы является создание модели вычислительной системы (ВС) или ее части на некотором уровне детализации, описывающей и имитирующей ее структуру и функциональность.

Каждый реальный объект (реальная ВС) обладает бесконечной сложностью, множеством характеристик, внутренних и внешних связей. Модель есть приближенное описание объекта с целью получения требуемых результатов с определенной точностью и достоверностью.

При необходимости исследования поведенческих характеристик ВС в процессе исследования выгодно использовать не сам объект, а его модель. Степень приближения модели к описываемому объекту может быть различной и зависит от требований задачи.

Существуют различные типы моделей:

- Аналитические (математические) модели
- Аналоговые модели
- Физические модели
- Имитационные модели

Последний тип моделей является предметом нашего изучения.

Одним из подходов к построению имитационной модели является построение ее в виде системы массового обслуживания (СМО), с характерной для СМО терминологией: источник, буфер, прибор, диспетчер, заявка (требование).

Существуют два подхода к построению моделирующего алгоритма:

Принцип Δt

Универсальный метод построения моделирующего алгоритма, когда состояние объекта проверяется через фиксированный интервал модельного времени. Суть его заключается в следующем: в каждый момент времени $t_i = t_{i-1} + \Delta t_{i-1}$ получают приближенные значения

характеристик исследуемого объекта. Δt можно получить детерминированным способом.

Основной критерий выбора Δt — он должен быть настолько мал, чтобы не пропустить событие в моделируемой системе, которое должно быть учтено при выбранной детальности моделирования. Метод неэффективен, т.к. постоянно проверяет состояние объектов моделирования, не изменяющихся при этом, особенно при малых Δt .

Принцип особых состояний.

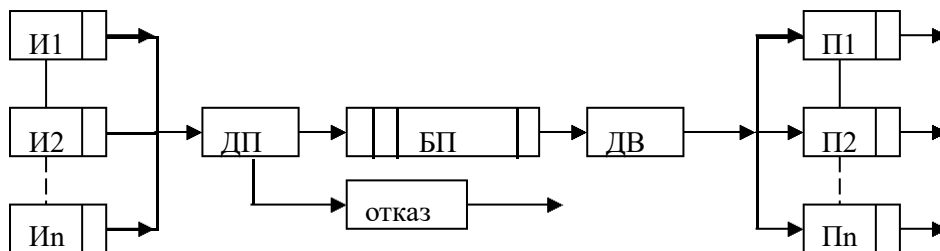
При исследовании реальной системы интервалы, в которых состояние ее не меняется, не представляют интереса. Имеют значение только переходы системы из одного состояния в другое в некоторые моменты времени. Эти переходы определяются особыми состояниями или событиями.

Рассмотрим некоторые типы особых событий, которые изменяют состояние системы:

- Поступление заявки в СМО (момент генерации заявки источником).
- Освобождение прибора (готовность прибора взять заявку на обслуживание).
- Окончание процесса моделирования.

Использование принципа особых событий для построения имитационной модели наиболее эффективно. В настоящей курсовой работе предлагается использовать именно этот принцип.

Формализованная схема и описание СМО.



Здесь I_i ($i=1..n$) – источник заявок, который генерирует заявки, а все вместе n источников создают входной поток заявок в систему.

Каждая заявка приходит в СМО со своими характеристиками. Это $T_{вх}$ — время генерации заявки (время поступления её в СМО) и номер заявки составленный из номера источника, сгенерировавшего заявку, и порядкового номера заявки от этого источника. Например, (2.3) – третья заявка от второго источника.

Π — приборы, которые обслуживают заявки и создают выходной поток заявок после обслуживания.

$БП$ — буферная память (место для хранения очереди заявок).

В общей памяти хранятся заявки от различных источников. Порядок их записи в $БП$ определяется только дисциплиной буферизации.

$ДП$ — диспетчер постановки заявок.

$ДВ$ — диспетчер выбора заявок.

4.	ИБ	ИЗ2	ПЗ1	Д1031	Д1004	Д2П2	Д2Б4	ОР2	ОД1
----	----	-----	-----	-------	-------	------	------	-----	-----

Источники:

ИБ – бесконечный источник;

ИЗ2 – равномерный закон распределения заявок;

Приборы:

ПЗ1 – экспоненциальный закон распределения времени обслуживания;

Описание дисциплин постановки и выбора:

Дисциплина буферизации:

Д1031 – заполнение буферной памяти по кольцу;+ Дисциплина отказа:

Д1004 – последняя поступившая в буфер; +

Дисциплина постановки на обслуживание:

Д2П2 – выбор прибора по кольцу;

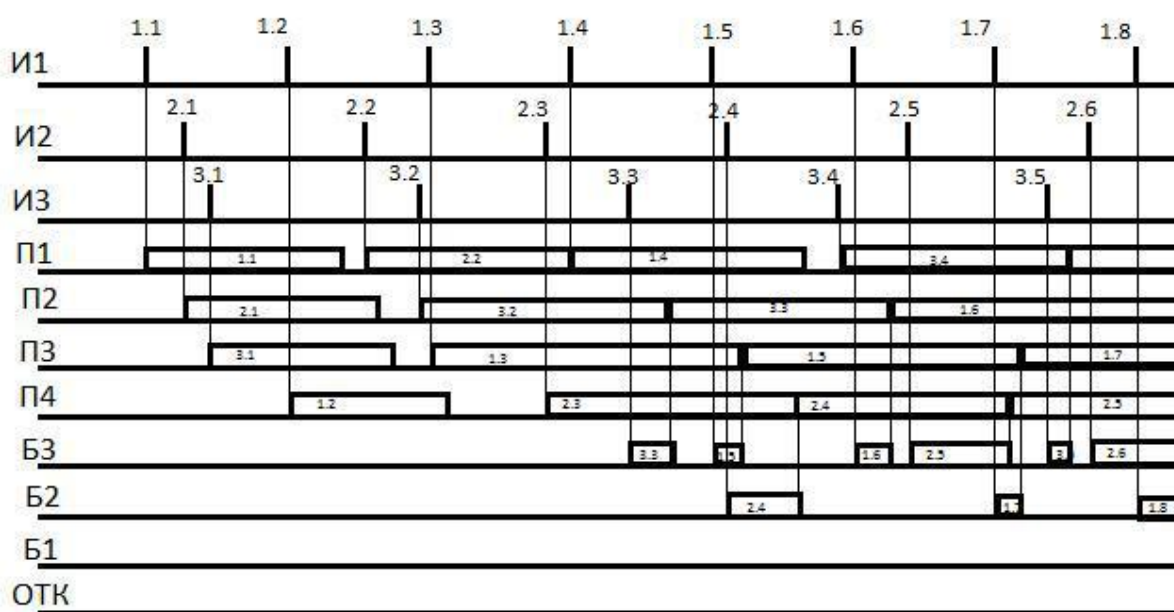
Д2Б4 – приоритет по номеру источника, по одной заявке(среди заявок одного приоритета имеет смысл выбирать последнюю поступившую); +

Виды отображения результатов работы программной модели:

Динамическое отражение результатов:

ОД1 – календарь событий, буфер и текущее состояние; Отражение результатов после сбора статистики: ОР2 – графики по значениям параметров.

Пример временной диаграммы функционирования системы



Пример технической системы (ВС или части ВС), удовлетворяющей формализованному описанию

Техническая система	Система сбора данных об окружающей среде.
---------------------	---

Источники	<p>Источниками являются датчики, каждый из которых собирает информацию о температуре, атмосферном давлении и силе ветра.</p> <p>Датчики посылают информацию на обработку в виде сетевого пакета размером 64 Кбайт. В целях снижения нагрузки на устройства обработки данных информация посылается лишь при отклонении одной из измеряемых величин более чем на 1 %.</p> <p>Количество датчиков может быть от 4 до 10. Желательно получать и обрабатывать информацию с максимального числа датчиков.</p>
Приборы	<p>Приборами являются ЭВМ, которые получают и обрабатывают информацию с датчиков. Полученные данные передаются на устройство отображения, а также записываются в базу данных для статистического анализа.</p>
Буфер	<p>Буфером является буфер сетевого коммутатора, который в базовой конфигурации составляет 640 Кбайт (10 заявок), но может быть наращен до 2 Мбайт (32 заявки) с шагом 64 Кб.</p>
Дисциплина постановки в буфер	<p>В порядке поступления, ни один из датчиков не имеет приоритета.</p>
Дисциплина выбора из буфера	<p>LIFO, т.к. самая последняя информация является наиболее важной, т.к. показывает состояние параметров в последний момент времени.</p>

Дисциплина отказа	Самая старая заявка в буфере идёт в отказ.
Дисциплина постановки на обслуживание	Приоритет по номеру прибора.

Ограничения и требуемые характеристики:

Вероятность отказа должна составлять не более 10 %.

Загрузка приборов более 90 %.

Время пребывания заявки в системе ограничено 60 мс.

Данное значение не может быть ниже времени обработки заявки в ЭВМ, которое зависит от типа установленного в ЭВМ процессора. Доступные типы процессоров и характеристики программного-аппаратного комплекса, построенного на данном типе процессора приведены ниже в таблице компонентов системы.

Количество датчиков	От 4 до 10
Размер заявки	64 Кб
Размер буфера	От 640 Кб до 2 Мбайт
Количество приборов	От 1 до 4.
Скорость работы датчиков	Простейший поток с $\lambda=0,1$ мс
Скорость работы приборов	Интервал: 7 мс (2-12 мс), 9 мс (3-15 мс), 9 мс (7-11 мс), 10 мс (5-15 мс).

Примечание:

1. Чем выше скорость работы прибора, тем выше его стоимость (задана таблицей).
2. Стоимость наращивания буфера также задана в таблице.
3. Стоимость датчиков не учитывается, так как уже имеется фиксированное число (10) датчиков одного типа. Желательно задействовать максимальное число датчиков.

Стоимость компонентов системы

Наименование	Характеристики	Цена
ЭВМ с процессором Intel Celeron 1300 MHz	Время обслуживания заявки: 10 мс (5-15 мс)	7000 р.
ЭВМ с процессором Intel Pentium 1600 MHz	9 мс (7-11 мс),	11000 р.
ЭВМ с процессором Intel Atom 1600 MHz	9 мс (3-15 мс)	11000 р.
ЭВМ с процессором Intel Xeon 2000 MHz	7 мс (2-12 мс)	16000 р.
Блок памяти для сетевого коммутатора	Объем 640 Кб	200 р.

Документация на ПО

Обобщенная блок-схема

При начале моделирования задаем состояние системы на данный момент. В нашем случае выбираем состояние «Начало функционирования системы» :

1. Моменты наступления ОС в будущем (Известны моменты поступления Первых Заявок от Источников).
2. Не известны моменты освобождения Приборов в будущем (все приборы в данный момент свободны).
3. Состояние БУФЕРА (буфер пуст).

Эта функция реализуется в блоке «Задание начальных данных определяющих одну реализацию».

На каждом шаге моделирования рассматриваются моменты наступления всех Особых Событий в БУДУЩЕМ (момент наступления больше текущего момента модельного времени (ТМОД)) и выбираем минимальное из них (ТМИН). ОС - моменты наступления которых в будущем не известны при этом не рассматриваются.

Эта функция реализуется в блоке - БООС.

Все моменты наступления ОС хранятся в информационной структуре КАЛЕНДАРЬ (массив, хранящий два параметра - ТНАСТ - момент наступления ОС, ПР - признак (0 - известен момент наступления ОС в БУДУЩЕМ, 1 – не известен))
В блоке БООС определяется ТМИН и соответствующий ему номер строки НМИН (тип ОС).

В блоке БАС на основании НМИН определяется блок, который обрабатывает соответствующее ОС и вызывает требуемый БМС (блок модификации состояния системы).

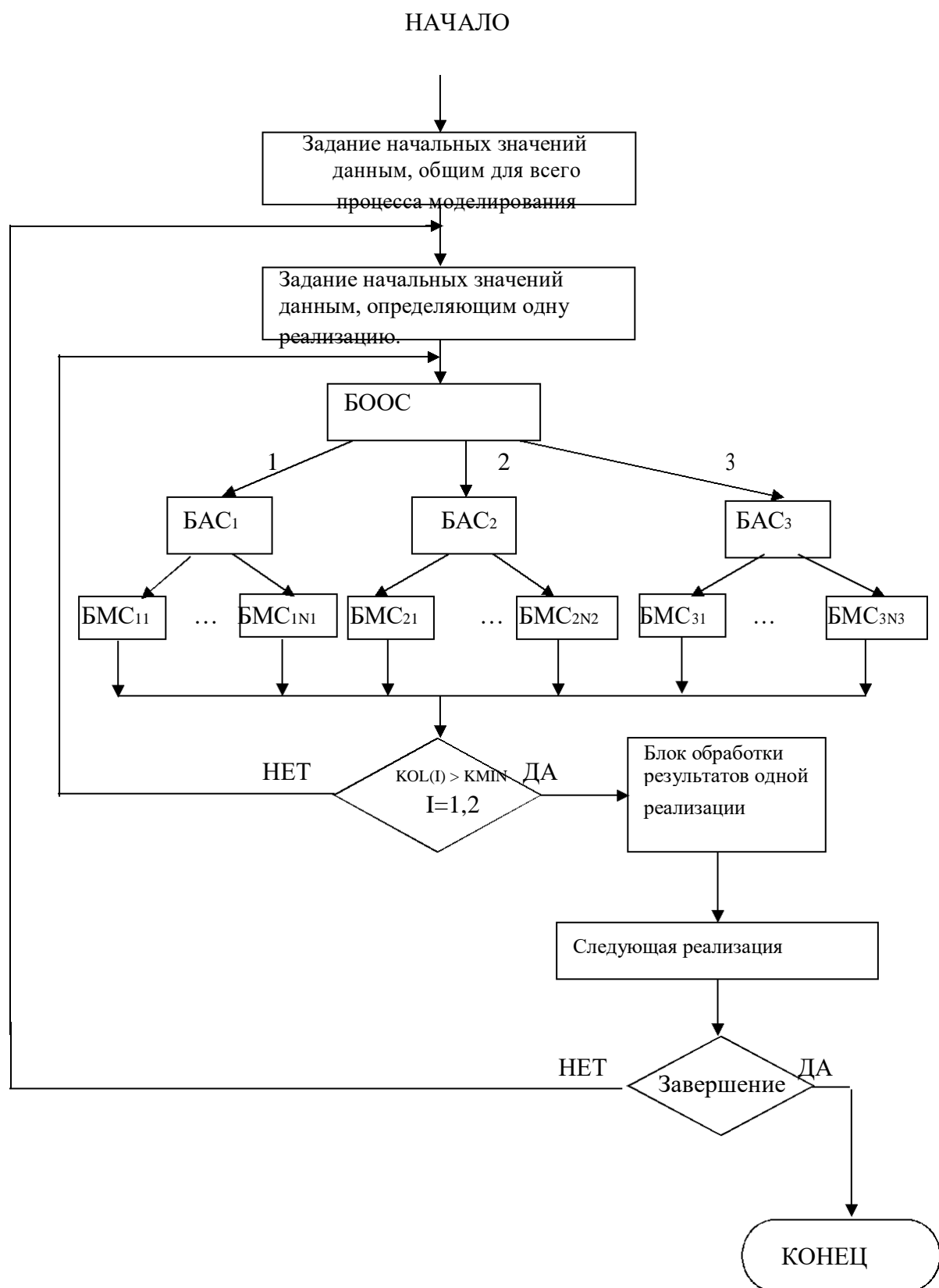
БМС1 - поступила ЗАЯВКА от источника «К» (номер И по строке в структуре КАЛЕНДАРЬ):

1. Формирование следующей заявки от данного источника (И - бесконечный)
2. ЕСЛИ есть свободный П, выбор П и постановка на обслуживание
ИНАЧЕ - постановка ЗАЯВКИ в БУФЕР (если в буфере есть место – запись в буфер, иначе – отказ заявке)

БМС2 - освободился П

1. ЕСЛИ в Б есть ЗАЯВКИ - выбор из Б и постановка на обслуживание
ИНАЧЕ - П в простой

БМС3 - обработка результатов моделирования



Модульная структура

Разработка производилась в среде Visual Studio 2005 на языке C++ с использованием графической библиотеки Qt.

Приложение использует объектно-ориентированную парадигму программирования и содержит набор классов:

- class Bid – класс заявки
- class Producer – класс источника
- class Device – класс прибора
- class Buffer – класс буфера

Программа содержит точку входа в файле main.cpp. Основное действие процедуры main – создание объекта окна программы и его отображение. После появления окна программы, запускается цикл обработки событий (действий пользователя). Отображение результатов в автоматическом режиме:

Отображение результатов в пошаговом режиме:

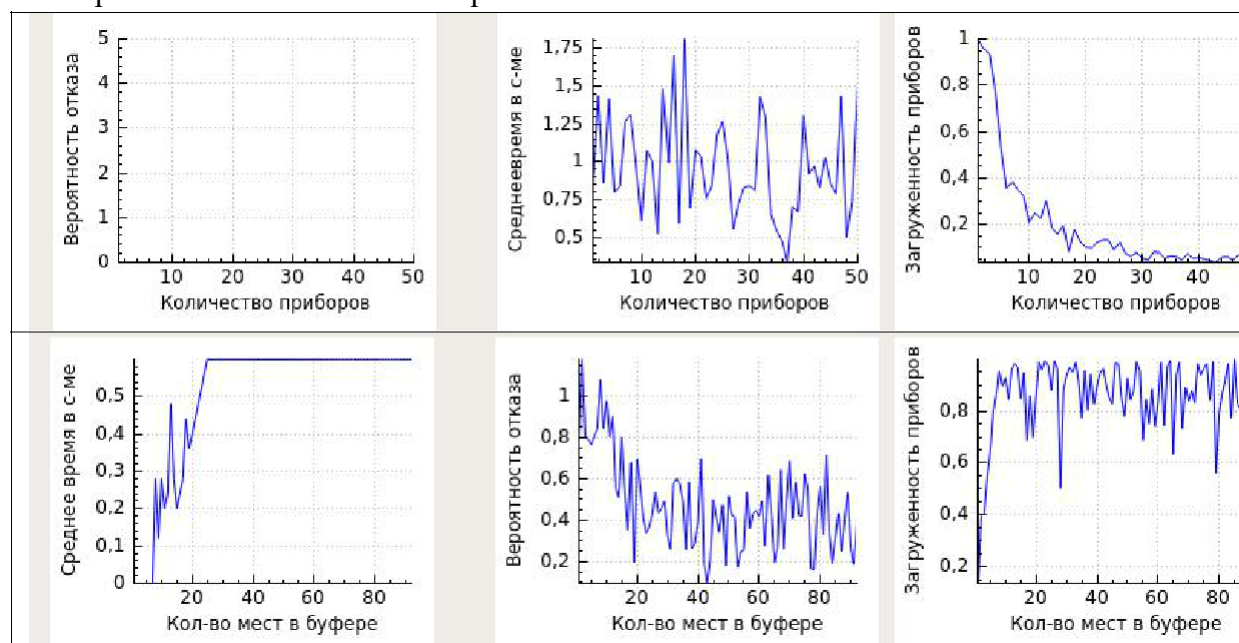
	Время	Признак	Число заявок	Число отказов
И0	2,62092	Ожидает	1	0
И1	2,62092	Ожидает	1	0
И2	2,62092	Ожидает	1	0
И3	2,62092	Ожидает	1	0
И4	2,62092	Ожидает	1	0
И5	2,62092	Ожидает	1	0
И6	2,62092	Ожидает	1	0
И7	2,62092	Ожидает	1	0
И8	2,62092	Ожидает	1	0
И9	2,62092	Ожидает	1	0
П0	0,534848	Занят	1	-
П1	0,904837	Занят	2	-

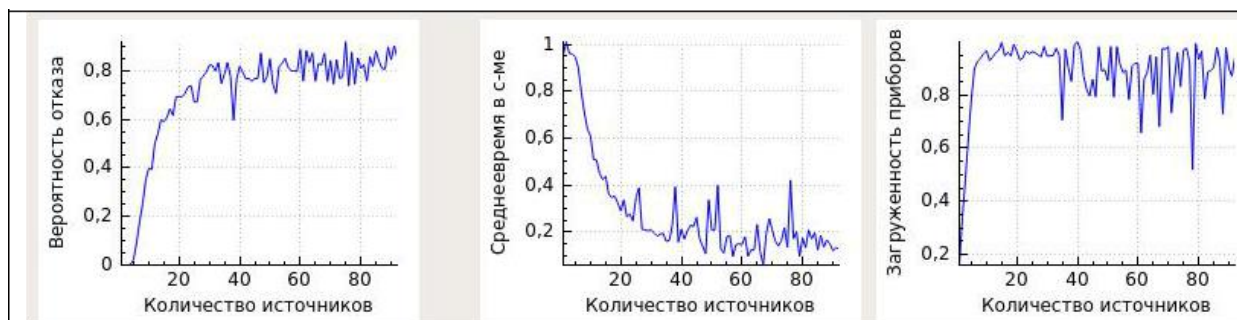
	Время	Источник
0	-	-
1	0,758166	3
2	0,758166	4
3	1,663	5
4	1,663	6
5	0,758166	7
6	0,758166	8
7	0,758166	9

Step

Finish

Отображение в автоматическом режиме





Результаты работы имитационной модели

Определение количества реализаций.

Количество реализаций, необходимое для получения нужной точности при заданной доверительной вероятности, можно оценивать по формуле:

$$N = \frac{t_{\alpha}^2 (1 - p)}{p \delta^2},$$

где p — вероятность отказа заявкам в обслуживании,

$t_{\alpha} = 1.643$ для $\alpha=0.9$,

$\delta=0.1$ — относительная точность.

По результатам работы программы получено, что в большинстве случаев для достижения заданной точности необходимо от 2000 до 6000 заявок. Однако, в случаях, когда p мало (<0.05) для достижения точности в 10% может потребоваться существенно больше заявок (20000-30000).

Анализ результатов, выводы и рекомендации по выбору конфигурации системы.

Т. к. целью моделирования является выбор конфигурации системы, требующей наименьшее количество ресурсов и обрабатывающей максимальный поток информации, то начнем с

проверки конфигурации с макс. числом источников и минимальным числом приборов и мин. размером буфера.

Число источников	Число приборов	Duration	Lambda	Размер буфера	P отк	T треб	Загруженность
10	1	5	5	10	0.517	0.4973	0.99788
10	2	5	5	10	0.118	0.8824	0.88065
10	3	5	5	10	0.003433	0.96487	0.643255

Из таблицы видно, что в последнем случае мы получили необходимую вероятность отказов, но загруженность приборов, в свою очередь, упала.

В таком случае можно попробовать изменить три вещи:

- увеличить количество источников;
- уменьшить интервал генерирования заявки источниками;
- заменить эти приборы на меньшее количество приборов с большей производительностью.

Число источников	Число приборов	Duration	Lambda	Размер буфера	P отк	T треб	Загруженность
15	3	5	5	10	0.18	0.72528	0.716123
15	3	5	5	20	0.3191	0.92301	0.921754
10	3	5	5	10	0.091205	0.88171	0.971557

Как видно из таблицы, мы удовлетворили всем условиям, кроме времени пребывания заявки в системе. Попробуем поставить приборы с большей производительностью.

Число источников	Число приборов	Duration	Lambda	Размер буфера	P отк	T треб	Загруженность
10	3	5	3	10	0.099193	0.85881	0.954282
10	2	5	1	8	0.099754	0.57035	0.95058

Последний случай удовлетворяет всем условиям. Стоимость такой системы порядка 32000р.

ПРИЛОЖЕНИЕ 6

Метод особых событий

В основе реализации моделирования методом особых событий лежит реализация перемещения по оси времени от особого события к особому событию на основе информационной структуры - календарь особых событий (КАОС).

КАОС – список ОС, где по каждому ОС приведены параметры:

- тип ОС (поступление заявки на вход системы, освобождение прибора, конец моделирования)
- Тос момент времени наступления очередного ОС данного типа
- ПР признак, который принимает значение :
 - 0 – известен момент наступления очередного ОС
 - 1 – не известен момент наступления очередного ОС

В нашем случае 1 у ОС «Поступление заявки» указывает на отсутствие очередной заявки от данного источника, а 1 у ОС «Освобождение прибора» означает, что не известен момент освобождения в будущем, т.е. прибор уже свободен и находится в состоянии простоя.

Для определения следующего относительно текущего положения ОС необходимо в КАОС выбрать ОС с минимальным Тос среди всех ОС, у которых ПР=0. Именно эту функцию реализует блок БООС.

На основании результата БООС производится анализ типа ОС в блоке БАС, который подключает соответствующий БМС для модификации состояния системы.

