МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра информационной безопасности

КУРСОВАЯ РАБОТА

по дисциплине «Моделирование систем»

Тема: Анализ подходов к моделированию систем

Студентка гр. 1361	_________________	Токарева У.В.
Преподаватель	_________________	Шульженко А.Д.

Санкт-Петербург
2024

ЗАДАНИЕ
НА КУРСОВУЮ РАБОТУ

Студентка Токарева У.В.

Группа 1361

Тема работы: анализ подходов к моделированию систем

Исходные данные:

- Количество процессов в системе – 1;
- Количество приоритетов процессов – 8;
- Квант времени, отводимый на выполнение процесса – 136 мс;
- Количество процессоров, выполняющих процессы – 1;
- Стохастическая аппроксимация правила обращения процессов к процессору имеет распределение Эрланга.

Содержание пояснительной записки:
«Содержание», «Введение», «Непрерывно-детерминированный подход», «Дискретно-детерминированный подход», «Дискретно-стохастический подход», «Непрерывно-стохастический подход», «Сетевой подход», «Обобщенный подход», «Заключение», «Список использованных источников».

Предполагаемый объем пояснительной записки:
Не менее 00 страниц.

Дата выдачи задания: 16.10.2024

Дата сдачи реферата: 00.00.2000

Дата защиты реферата: 00.00.2000

Студентка	________________	Токарева У.В.
Преподаватель	________________	Шульженко А.Д.

АННОТАЦИЯ

Кратко (в 8-10 строк) указать основное содержание курсового проекта (курсовой работы), методы исследования (разработки), полученные результаты.

SUMMARY

Briefly (8-10 lines) to describe the main content of the course project, research methods, and the results.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Кратко описать цель работы, основные задачи им методы их решения.

1. НЕПРЕРЫВНО-ДЕТЕРМИНИРОВАННЫЙ ПОДХОД

1.1. Краткое теоретическое описание

В данном подходе в качестве математической модели используются дифференциальные уравнения. Обычно в таких математических моделях в качестве независимой переменной, от которой зависят неизвестные искомые функции, служит время t.

$$\vec{y}'(t) = \vec{f}(\vec{y}(t), t),$$
$$\vec{y}(t_0) = \overrightarrow{y_0},$$

где

$$\vec{y}(t) = (y_1, y_2, \dots, y_n), \vec{y}' = \frac{d\vec{y}}{dt},$$

$\vec{f} = (f_1, f_2, \dots, f_n)$ – вектор-функция, непрерывная на некотором (n+1)-мерном множестве.

Так как математические схемы такого вида отражают динамику изучаемой системы, то есть ее поведение во времени, то они называются D-схемами (англ. dynamic). В простейшем случае обыкновенное дифференциальное уравнение имеет вид:

$$y' = f(y, t)$$

1.2. Определение ограничений, накладываемых областью применимости этого подхода

В непрерывно-детерминированный подход переменные системы представлены в виде непрерывных функций времени. Поведение системы предсказуемо, так как все изменения определяются заранее известными закономерностями. Более того она обладает линейными или относительно простыми нелинейными взаимосвязями.

1.3. Доказательство неприменимости подхода к моделированию

Для абстрактной системы с определенными параметрами не применим непрерывно-детерминированный подход, так как нет определенной цели применения системы и физических параметров. Использование распределения Эрланга для модуляции обработки указывает на случайность в процессе обращения. Непрерывно-детерминированная модель не способна точно отразить стохастические явления. Квант времени для каждого режима фиксирован (136 мс). В непрерывно-детерминированной модели обычно используются непрерывные временные интервалы. Фиксированные интервалы не могут точно соответствовать таким моделям. Система имеет ограниченное количество процессоров (1). Это означает, что одновременно может выполняться только один процесс. Непрерывно-детерминированная модель не соблюдает ограничений по ресурсам. Присутствие 8 уровней приоритетов требует соблюдения порядка выполнения задач. Непрерывно-детерминированная модель обычно не включает в себя механизмы приоритета.

Учитывая все эти факторы, можно отметить, что непрерывно-детерминированный подход не подходит для исследования данной абстрактной вычислительной системы.

2. ДИСКРЕТНО-ДЕТЕРМИНИРОВАННЫЙ ПОДХОД

2.1. Краткое теоретическое описание

Дискретно-детерминированный подход основан на теории конечных автоматов. Конечный автомат – это автомат с конечным множеством внутренних состояний и входных (а следовательно, и выходных) сигналов.

Дискретно-детерминированный подход используется в тех случаях, когда необходимо смоделировать систему с дискретными состояниями, переходящими из одного состояния в другое в конкретные фиксированные моменты времени, то есть конечный автомат (система) переходит в другое состояние под воздействием внешних сигналов. Каждый переход происходит мгновенно и может рассматриваться как отдельное событие.

Таким образом, дискретно-детерминированный подход позволяет моделировать обработку заявок по приоритетам, а также следить за состоянием процессора и очереди в каждый момент времени.

2.2. Определение ограничений, накладываемых областью применимости этого подхода

Дискретно-детерминированный подход накладывает ряд ограничений на вычислительную систему, а именно:

- Отсутствие случайностей: все переходы между состояниями предсказуемы.
- Мгновенные переходы: переход из одного состояния в другое считается мгновенным.
- Дискретные события: все события происходят в дискретные моменты времени, то есть квант времени фиксирован.

Таким образом, мы может сделать вывод, что дискретно-детерминированный подход применим для нашей абстрактной вычислительной системы, так как явления могут быть рассмотрены как поочередные процессы с определенным шагом.

2.3. Определение цели моделирования

Цель моделирования нашей абстрактной вычислительной системы с использованием дискретно-детерминированного подхода – создание точной и устойчивой математической модели, которая позволяет анализировать и прогнозировать работу системы или процесса с определенными входными данными и правилами поведения.

2.4. Доопределение исходных данных, недостающих для применения этого подхода

Исходные данные с доопределенными данными:

- Количество процессов в системе – 1;
- Количество приоритетов процессов – 8;
- Квант времени, отводимый на выполнение процесса – 136 мс;
- Количество процессоров, выполняющих процессы – 1;
- Стохастическая аппроксимация правила обращения процессов к процессору имеет распределение Эрланга.
- Количество очередей – 1;
- Количество мест в очереди – 5.

2.5. Проведение моделирования

Определим характеристики для начального состояния S_0:

- Время: t = 0;
- Процесс N = free;
- Приоритет P_n= free;
- Очередь: BUF[0 … 4] = free.
- Запросы Z_n.

Начальное состояние – это момент, когда система не обрабатывает никаких запросов, то есть очередь и процесс пустые. Начальное состояние системы представлено на рисунке 1.

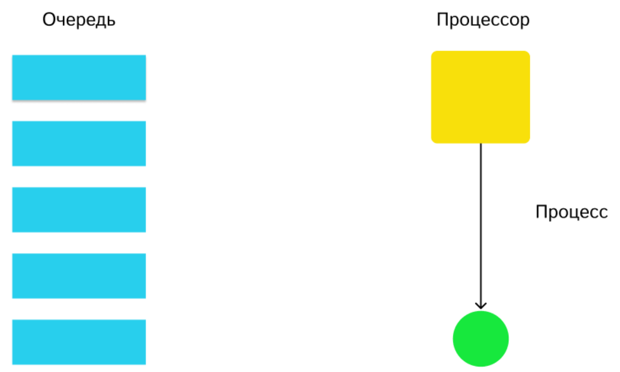

Рисунок 1 – Начальное состояние системы.

Система начинает свою работу, когда происходит первое отправление запроса. Так как в системе один процесс, в момент времени может прийти только один запрос. Таким образом система примет этот запрос и перейдет в состояние частичной занятости S_1. Состояние S_1 – тот период, когда запрос находится в обработке, но очередь пустая.

Рассмотрим поступление первого запроса:

t = 0;

P_1=3;

N = Z_1;

BUF[0 … 4] = free;

$S_0(Z_1) \rightarrow S_1$.

В результате данных преобразований система находится в состоянии частичной занятости, то есть в состоянии S_1. Состояние S_1 представлено на рисунке 2.

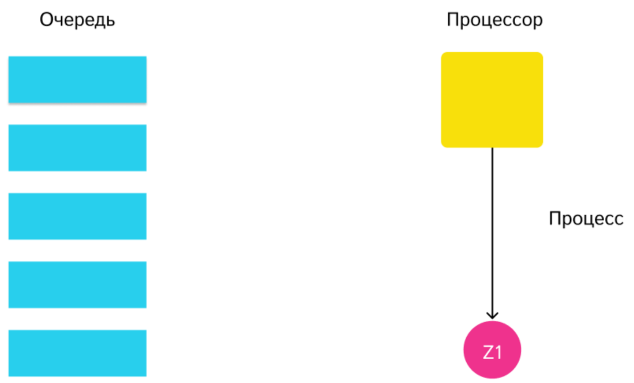

Рисунок 2 – Состояние S_1.

Далее в систему может быть направлен следующий запрос. Разность между временем поступления запросов определяется распределением Эрланга, следовательно время увеличивается на Δt_1. Новый запрос не прерывает обработку запроса, даже если его приоритет выше. Новый запрос помещается в очередь. Таким образом система переходит в новое состояние, когда процесс занят, а очередь частично занята.

Рассмотрим поступление нового запроса:

t = Δt_1;

P_1=3;

N = Z_1;

P_2=4;

BUF[0] = Z_2;

BUF[1 … 4] = free;

$S_1(Z_2) \rightarrow S_2$.

Состояние S_2 представлено на рисунке 3.

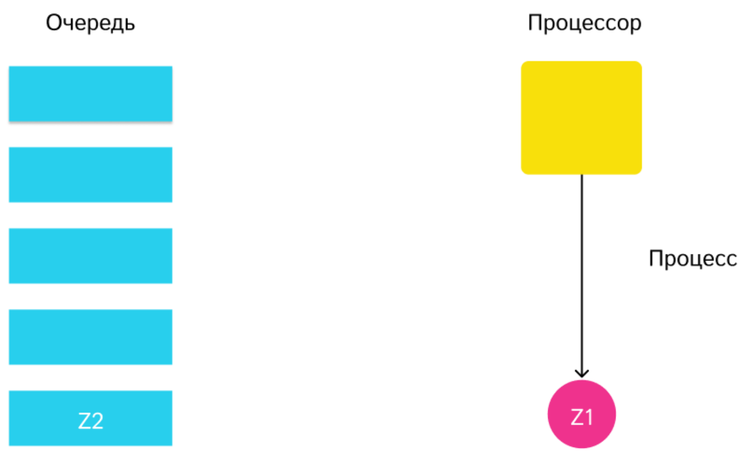

Рисунок 3 – Состояние S_2.

Далее рассмотрим поступление следующих запросов. Система работает следующим образом: если процесс занята, а в очереди есть пустые слоты, то новый запросы помещаются в очередь в зависимости от своего приоритета. Чем выше приоритет запроса, тем ближе к началу очереди он располагается.

Также каждый новый запрос увеличивает время. При этом состояние системы от этого не меняется

Рассмотрим поступление нового запроса:

t = Δt_1 + Δt_2;

P_1=3;

N = Z_1;

P_2=4;

BUF[0] = Z_3;

P_3=8;

BUF[1] = Z_2;

BUF[2 … 4] = free;

$S_2(Z_3) \rightarrow S_2$.

Таким образом, новый запрос имеет больший приоритет, чем запрос, который ранее находился в очереди, поэтому первое место в очереди достается новому запросу. Текущее состояние системы представлено на рисунке 4.

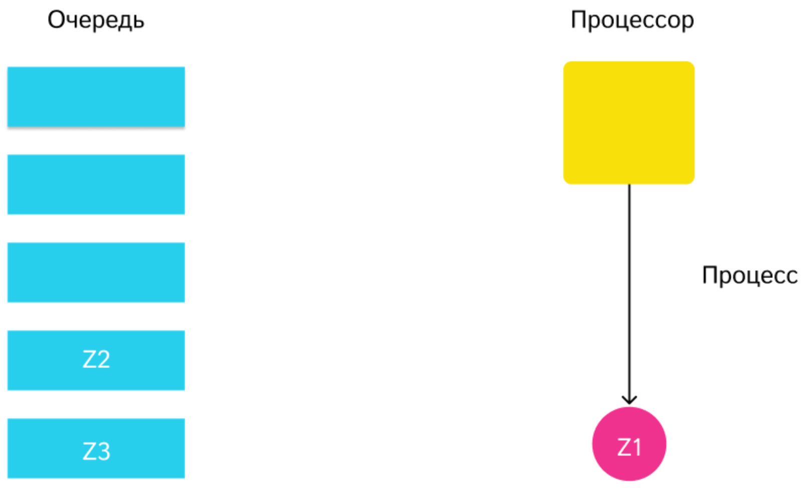

Рисунок 4 – Текущее состояние системы.

Пока процесс занят, все новый запросы отправляются в очередь, если там есть свободные места. Если все места в очереди заняты, а новый запрос имеет приоритет выше, чем приоритет последнего запроса в очереди, то он помещается в очередь согласно своему приоритету, а последний запрос из очереди отклоняется. Если же новый запрос имеет приоритет меньше, чем последний запрос в очереди, то новый запрос отклоняется.

Рассмотри поступление четвертого запроса:

t = Δt_1 + Δt_2 + Δt_3;

P_1=3;

P_2=4;

P_3=8;

P_4=6;

N = Z_1;

BUF[0] = Z_3;

BUF[1] = Z_4;

BUF[2] = Z_2;

BUF[3 … 4] = free;

$S_2(Z_4) \rightarrow S_2$.

Рассмотри поступление пятого запроса:

t = Δt_1 + Δt_2 + Δt_3 + Δt_4;

P_1=3;

P_2=4;

P_3=8;

P_4=6;

P_5=4;

N = Z_1;

BUF[0] = Z_3;

BUF[1] = Z_4;

BUF[2] = Z_2;

BUF[3] = Z_5;

BUF[4] = free;

$S_2(Z_5) \rightarrow S_2$.

Рассмотри поступление шестого запроса:

$t = \Delta t_1 + \Delta t_2 + \Delta t_3 + \Delta t_4 + \Delta t_5$;

P_1=3;

P_2=4;

P_3=8;

P_4=6;

P_5=4;

P_6=2;

N = Z_1;

BUF[0] = Z_3;

BUF[1] = Z_4;

BUF[2] = Z_2;

BUF[3] = Z_5;

BUF[4] = Z_6;

$S_2(Z_6) \rightarrow S_3$.

Мы видим, что система перешла в новое состояние, в котором система полностью загружена. Состояние системы S_3 представлено на рисунке 5.

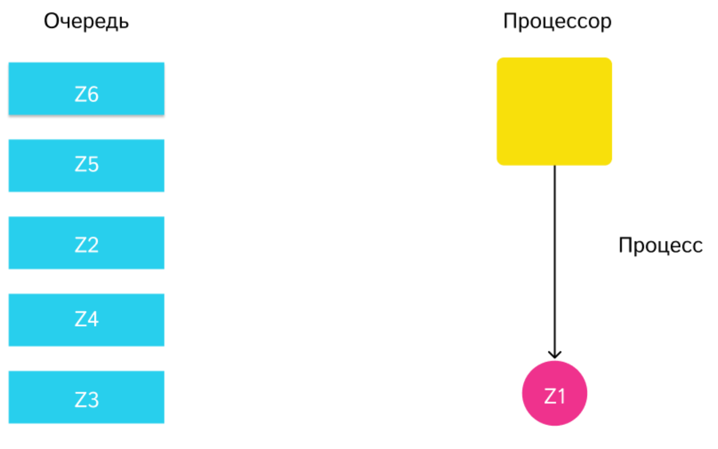

Рисунок 5 – Состояние системы S_3.

Как только процесс освободится, первый запрос из очереди поступит в обработку, а все остальные запросы сдвинутся в очереди за одну позицию вперед. Рассмотри ситуацию, в которой запрос обработан:

$t = \Delta t_1 + \Delta t_2 + \Delta t_3 + \Delta t_4 + \Delta t_5$ + 136 мс;

P_2=4;

P_3=8;

P_4=6;

P_5=4;

P_6=2;

N = Z_2;

BUF[0] = Z_4;

BUF[1] = Z_2;

BUF[2] = Z_5;

BUF[3] = Z_6;

BUF[4] = free;

$S_3 \rightarrow S_2$.

После выхода первого запроса из обработки система вернулась в состояние 2, а время увеличилось на 136 мс.

2.6. Анализ результатов

В процессе моделирования были рассмотрены все возможные состояния системы, а именно:

- S_0 – Состояние, в котором система полностью свободна;
- S_1 – Состояние, в котором процесс занят, а очередь полностью свободна;
- S_2 – Состояние, в котором процесс занят, а очередь частично занята;
- S_3 – Состояние, в котором система полностью занята.

Также были рассмотрены все возможные переходы между состояниями, а именно:

- Поступление первого запроса в систему;
- Поступление последующих запросов в систему, когда процесс еще занят;

- Поступление в систему запроса, который занимает последнее место в очереди;
- Выход из обработки первого запроса;
- Выход из обработки последнего система.

Принцип работы системы таков: когда система пустая и в нее поступает первый запрос, он моментально поступает в обработку. Временная разница между поступлениями запросов описывается распределением Эрланга и отличается от запроса к запросу. Время, которое каждый запрос находится в обработке равно 136 мс. Когда процесс занят, запросы помещаются в очередь согласно своему приоритету. Если очередь занята, то новый запрос либо помещается в очередь согласно своему приоритету и последний запрос из очереди отклоняется, либо же новый запрос отклоняется. Когда запрос выходит из обработки, первый запрос из очереди занимает процесс. Система заканчивает свою работу, когда последний процесс выходит из обработки.

2.7. Выводы

Дискретно-детерминированный подход оказался достаточно эффективным для моделирования моей системы. С помощью этого метода возможно оценить влияние приоритетов и проанализировать поведение системы.

3. ДИСКРЕТНО-СТОХАСТИЧЕСКИЙ ПОДХОД

3.1. Краткое теоретическое описание

Дискретно-стохастический подход – это подход к моделированию систем, в котором переменные принимают случайные значения, то есть данный подход используется в ситуациях, когда изменения происходят в определенные моменты времени, а результаты зависят от вероятностного распределения. При применении данного подхода переходы между состояниями происходят не непрерывно, а с некоторой вероятностью.

3.2. Определение ограничений, накладываемых областью применимости этого подхода

Данный подход применим к системам, в которых переходы подвержены случайным влияниям, а также дискретно-стохастический подход накладывает ряд ограничений:

- Для систем с большим количеством состояний данный подход плохо применим, так как моделирование систем становится сложно-управляемым;
- Дискретно-стохастический подход не применим для систем непрерывного характера, так как в данном подходе время и состояния представляют собой отдельные значения, определенные конкретным образом.
- Данных подход плохо применим для систем, обладающих сложной динамикой.

3.3. Определение цели моделирования

Цель моделирования абстрактной вычислительной системы при помощи дискретно-стохастического подхода – создание точной математической модели, которая необходима для анализа и прогнозирования работы системы в условиях распределения частоты запросов по заданным правилась аппроксимации.

3.4. Доопределение исходных данных, недостающих для применения этого подхода

Все необходимые данные, необходимые для моделирования остаются прежними. Для данного подхода важно более подробно рассмотреть распределение Эрланга.

Рассмотрим основные формулы, описывающие распределение Эрланга:
Плотность:

$$f(t) = \frac{\mu(\mu t)^{r-1}}{(r-1)!} e^{-\mu t}$$

Дополнительная функция распределения:

$$\bar{F}(t) = \sum_{i=0}^{r-1} \frac{(\mu t)^i}{i!} e^{-\mu t}$$

Начальные моменты:

$$f_k = \frac{r(r+1)\dots(r+k-1)}{\mu^k}$$

При проведении моделирования системы с использованием дискретно-стохастического подхода необходимо будет доработать модель, полученную в результате моделирования с использованием дискретно-детерминированного подхода, однако теперь необходимо более детально рассмотреть при помощи распределения Эрланга время поступления запроса.

3.5. Проведение моделирования

При проведении моделирования, как уже было сказано ранее, нужно более подробно рассмотреть распределение Эрланга, которое будет определять, с какой периодичностью поступают запросы.

Распределение Эрланга контролирует моменты поступления запросов. Соответственно в момент поступления нового запроса время будет увеличиваться на $\Delta t_n = \frac{\lambda}{k} * \ln U_i$, где λ – количество поступающих запросов за единицу времени, примем его равным трем, k – количество фаз, а U_i –

случайная величина. Таким образом мы видим, что время поступления запроса определяется случайным образом.

После описания времени поступления запросов, необходимо дополнить систему. Дополненная система представлена на рисунке 6.

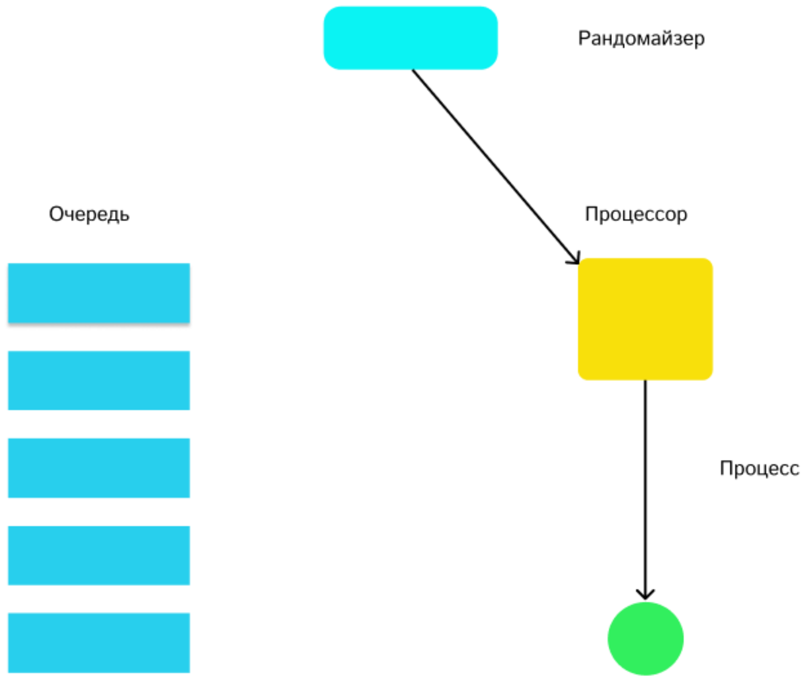

Рисунок 6 – Дополненная система.

Важно отметить, что состояния системы и переходы между состояниями остались прежними, однако теперь мы можем учитывать время поступления запросов, которое определено распределением.

3.6. Анализ результатов

В результате моделирования пропала неопределенность времени поступления запросов. Учет неопределенности позволяет моделировать случайные события и неопределенности. Еще одним из плюсов дискретно-стохастического подхода является простота формализации, то есть модели могут быть построены с учетом четких математических формул.

3.7. Выводы

Дискретно-стохастический подход оказался достаточно эффективным для моделирования моей системы. С помощью этого метода можно не только оценить влияние приоритетов и проанализировать поведение системы, но и формализовать время поступления запросов в систему.

4. НЕПРЕРЫВНО-СТОХАСТИЧЕСКИЙ ПОДХОД

4.1. Краткое теоретическое описание

Непрерывно-стохастический подход – еще один подход к моделированию абстрактных вычислительных систем. При данном подходе в качестве типовых математических схем рассматривается система массового обслуживания, называемая Q-схемами. Системы массового обслуживания представляют собой класс математических схем, разработанных в теории массового обслуживания и различных приложениях для формализации процессов.

В качестве процесса обслуживания могут быть представлена различные по своей физической природе процессы функционирования, для которых характерны случайное появление заявок на обслуживание и завершение обслуживания в случайные моменты времени, то есть стохастический характер процесса.

Можно выделить две основные составляющие модели: ожидание обслуживания и заявки и само обслуживание. Потоком событий называется последовательность событий, происходящий одно за другим в какие-то случайные моменты времени.

4.2. Определение ограничений, накладываемых областью применимости этого подхода

Для непрерывно-стохастических моделей характерно: непрерывное изменение времени и присутствие случайностей. При применении данного подхода время рассматривается как непрерывный процесс.

При имитировании непрерывно-стохастической системы, непрерывно-изменяющееся время заменяют на дискретное время и малым изменением шага Δt.

Непрерывно-стохастический подход имеет ряд ограничений в области применимости:

- Непрерывно-стохастические модели часто предполагают линейные зависимости между переменными, что ограничивает применение данного подхода в ситуациях, когда имеются значительные нелинейные отношения;

- Часто предполагается, что случайные процессы являются стационарными или имеют определенные свойства. Если эти условия не выполняются, подход становится сложно-применимым;
- Моделирование и оценка параметров непрерывно-стохастической системы может требовать значительных вычислительных ресурсов.
- Для данного подхода характерен марковский характер процессов, так как часто предполагается, что будущее состояние процесса зависит только от его текущего состояния.

4.3. Определение цели моделирования

Цель моделирования абстрактной вычислительной системы с использованием непрерывно-стохастического подхода заключается в создании гибкой и адаптивной математической модели, которая позволяет не только анализировать и прогнозировать поведение системы в условиях неопределенности, но и обеспечивать оценку влияния различных факторов на динамику системы. Модель будет использовать непрерывные вероятностные процессы для оценки вероятностных характеристик системы, что позволяет глубже понять взаимодействия между компонентами системы и делать более обоснованные прогнозы. Далее необходимо провести моделирование системы путем программирования.

4.4. Доопределение исходных данных, недостающих для применения этого подхода

Все необходимые данные были определены ранее, однако наиболее важным из них является то, что стохастическая аппроксимация правила обращения процессов к процессору имеет распределение Эрланга.

При проведении моделирования с помощью непрерывно-стохастического подхода будет использоваться та же модель, что и для дискретно-детерминированного подхода, но несколько доработанная. Отличие состоит в том, что теперь время отправления следующего запроса от каждого процесса к процессору представляет собой случайную величину, подчиняющуюся распределению Эрланга.

4.5. Проведение моделирования

Для моделирования системы при помощи непрерывно-стохастического подхода необходимо реализовать программу. Перед написанием кода необходимо рассмотреть математическую модель. Принципы работы системы были описаны ранее, но некоторые нюансы необходимо рассмотреть более подробно.

Во-первых, время между поступлениями запросов моделируется распределением эрланга с параметрами k и λ, от которых много зависит в работе системы. Более подробно влияние этих значений на работу системы будет рассмотрено в следующем разделе. Формула плотности распределения Эрланга:

$$f(t,k,\lambda) = \frac{\lambda^k t^{k-1} e^{-\lambda t}}{(k-1)!}$$

Во-вторых, необходимо рассмотреть основные формулы, необходимые для формирования статистики для анализа результатов.

Среднее время ожидания заявок:

$$W = \frac{\sum W_i}{N_{\text{обработанные}}}, \text{где}$$

W_i – время ожидания каждого запроса;

$N_{\text{обработанные}}$ – количество заявок, которые были обработаны системой за время работы;

$N_{\text{отклоненные}}$ – количество заявок, которые были отклонены системой за время работы.

Эффективность системы:

$$\eta = \frac{T_{\text{занятости}}}{T_{\text{общее}}}$$

Для реализации кода так же будем указывать время, которое система находится в рабочем состоянии.

Для моделирования абстрактной вычислительной системы был реализован алгоритм на языке python. Исходный код программы представлен в приложении к отчету.

4.6. Анализ результатов

Рассмотрим несколько результатов моделирования с разными исходными данные. Первый вариант входных данных:

- $\lambda = 10$
- k = 2
- Максимальное время моделирования = 10
- Максимальное количество запросов = 100

Результат моделирования № 1 представлен на рисунке 7.

```
Результаты моделирования:
Обработано заявок: 33
Отброшено заявок: 7
Среднее время ожидания: 0.7024 сек
Суммарное время занятости процессора: 4.6240 сек
Эффективность системы: 45.96%
```

Рисунок 7 – Результат моделирования № 1.

Занятость очереди для моделирования № 1 представлена на рисунке 8.

Рисунок 8 – Занятость очереди для моделирования № 1.

Второй вариант входных данных:

- λ = 10
- k = 5
- Максимальное время моделирования = 30
- Максимальное количество запросов = 200

Результат моделирования № 2 представлен на рисунке 9.

```
Результаты моделирования:
Обработано заявок: 57
Отброшено заявок: 0
Среднее время ожидания: 0.5588 сек
Суммарное время занятости процессора: 7.8880 сек
Эффективность системы: 26.20%
```

Рисунок 9 – Результат моделирования № 2.

Занятость очереди для моделирования № 2 представлена на рисунке 10.

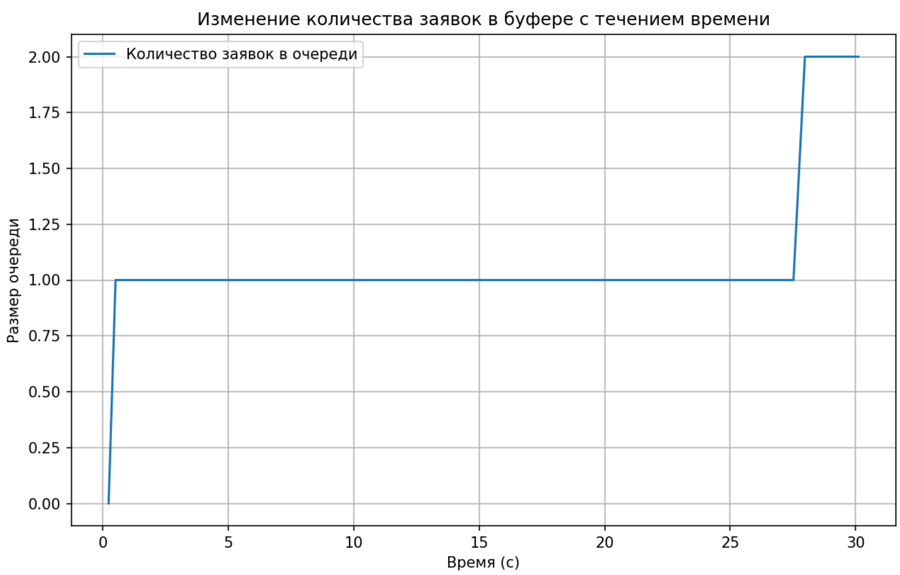

Рисунок 10 – Занятость очереди для моделирования № 2.

Третий вариант входных данных:

- λ = 25
- k = 2
- Максимальное время моделирования = 3000
- Максимальное количество запросов = 1000

Результат моделирования № 3 представлен на рисунке 11.

```
Результаты моделирования:
Обработано заявок: 516
Отброшено заявок: 484
Среднее время ожидания: 0.4775 сек
Суммарное время занятости процессора: 70.3120 сек
Эффективность системы: 68.56%
```

Рисунок 11 – Результат моделирования № 3.

Занятость очереди для моделирования № 3 представлена на рисунке 12.

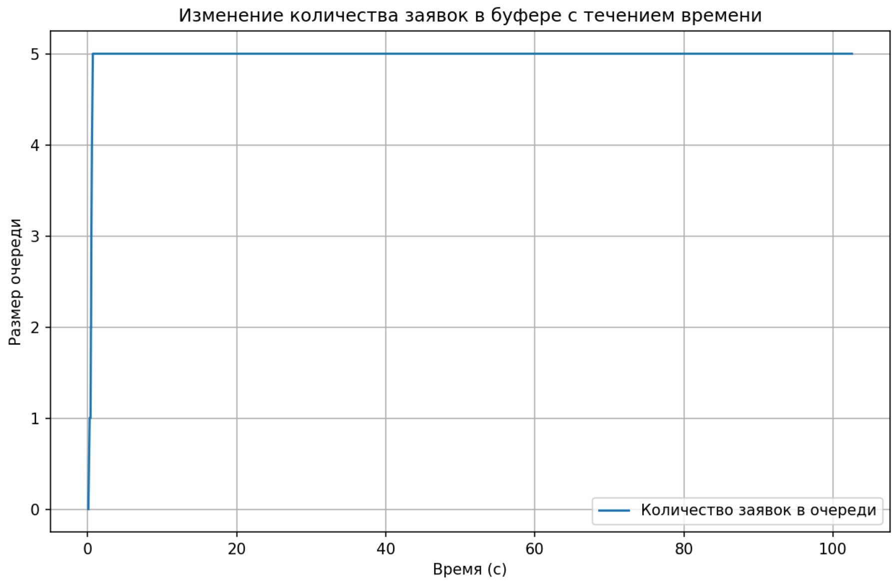

Рисунок 12 – Занятость очереди для моделирования № 3.

Четвертый вариант входных данных:

- $\lambda = 50$
- k = 10
- Максимальное время моделирования = 300
- Максимальное количество запросов = 3000

Результат моделирования № 4 представлен на рисунке 13.

```
Результаты моделирования:
Обработано заявок: 1289
Отброшено заявок: 117
Среднее время ожидания: 1.0336 сек
Суммарное время занятости процессора: 175.4400 сек
Эффективность системы: 58.45%
```

Рисунок 13 – Результат моделирования № 4.

Занятость очереди для моделирования № 4 представлена на рисунке 14.

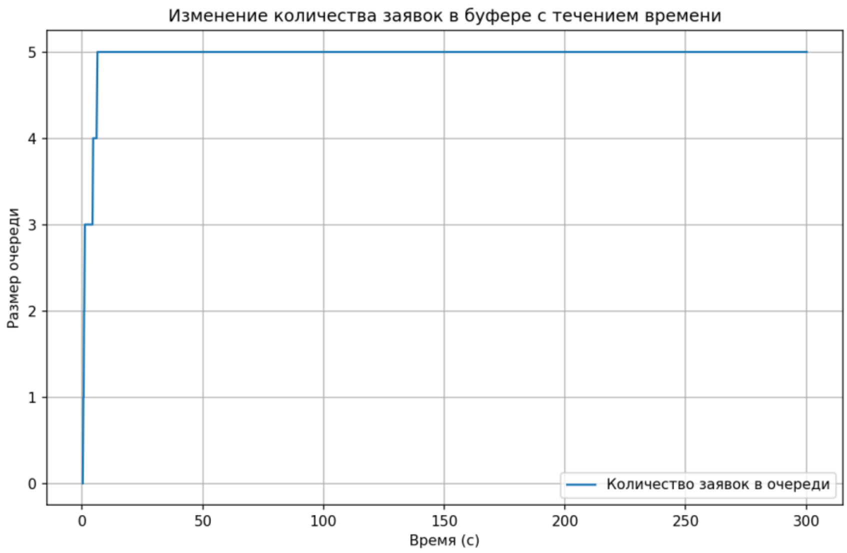

Рисунок 14 – Занятость для моделирования № 4.

По приведенным выше рисункам можно увидеть, что в зависимости от исходных данных эффективность системы меняется. В результате изменения интенсивность прибытия заявок (λ), количества фаз экспоненциального распределения (k), времени работы системы и количества максимального количества заявок, можно сделать вывод, что наиболее эффективно система работает при низких значениях k и относительно высоких значениях λ. Так же при рассмотрении работы системы на большем промежутке времени мы видим, что очередь полностью занята, запросы не успевают обрабатывать и из-за этого идут отказы в обслуживании.

4.7. Выводы

В результате моделирования можно сделать вывод, что непрерывно-стохастический подход отлично подходит для моделирования абстрактной вычислительной системы. Была разработана математическая модель, а также был разработан код на языке python. В ходе моделирования было проанализировано влияние параметров распределения Эрланга на эффективность системы.

5. СЕТЕВОЙ ПОДХОД

5.1. Краткое теоретическое описание

Сетевой подход к моделированию – это метод системного анализа, который используется для изучения и анализа сложных систем, процессов и проектов. Он основывается на построении графических моделей, которые отображают логические связи между элементами системы.

Сетевой подход используется для анализа систем путем построения графа, в котором вершины описывают состояния системы, которые были определены ранее, а ребра – переходы между состояниями.

Данный метод позволяет формализовать работу системы, оценить вероятности состояний и ключевые метрики, провести моделирование стохастических событий.

5.2. Определение ограничений, накладываемых областью применимости этого подхода

Сетевой подход применим для систем, где:

- Состояния четко определены, то есть известны состояния системы, возможные переходы между состояниями, приоритеты запросов.
- Также должны быть возможность описать события правилами переходов.
- Известно распределение событий, в моем случае появление запросов в системе описывается распределением Эрланга.
- Также должен быть ограниченный размер очередь.

Все эти ограничения делают подход плохо-применимым для сложных систем с большим количеством состояний.

5.3. Определение цели моделирования

Основной целью моделирования абстрактной вычислительной системы при помощи сетевого подхода является построение формальной модели работы системы, которая отражает обработку запросов, управление очередью на основании приоритетов и принцип отклонения или вытеснения запросов. Также к целям моделирования системы при помощи сетевого подхода можно отнести оценку вероятности и проведение оптимизации управления системой, то есть провести проверку того, насколько текущие правила работы эффективны.

5.4. Доопределение исходных данных, недостающих для применения этого подхода

Все необходимые исходные данные, необходимые для проведения моделирования, были определены ранее и остаются неизменными.

Для расчета вероятности событий будут использованы следующие формулы:

- λ – средняя интенсивность поступления запросов;
- $\mu = \frac{1}{136 \text{ мс}} \approx 7.35$ обрабатываемых запросов в секунду – интенсивность обработки;
- $p = \frac{\lambda}{\mu}$ – коэффициент загрузки системы;
- $P(S_0) = \frac{1}{\sum_{n=0}^{5} \frac{(p)^n}{n!} + \frac{(p)^5}{5!} * \frac{1}{1-p}}$ – вероятность состояния системы.

5.5. Проведение моделирования

Места:

- Поступление запроса;
- Свободный поток;
- Выход запроса;
- Отказ.

Переходы:

- Буфер;
- Обработка;
- Буфер переполнен.

Создадим граф, который будет визуализировать места и переходы. Места и переходы представлены на рисунке 15.

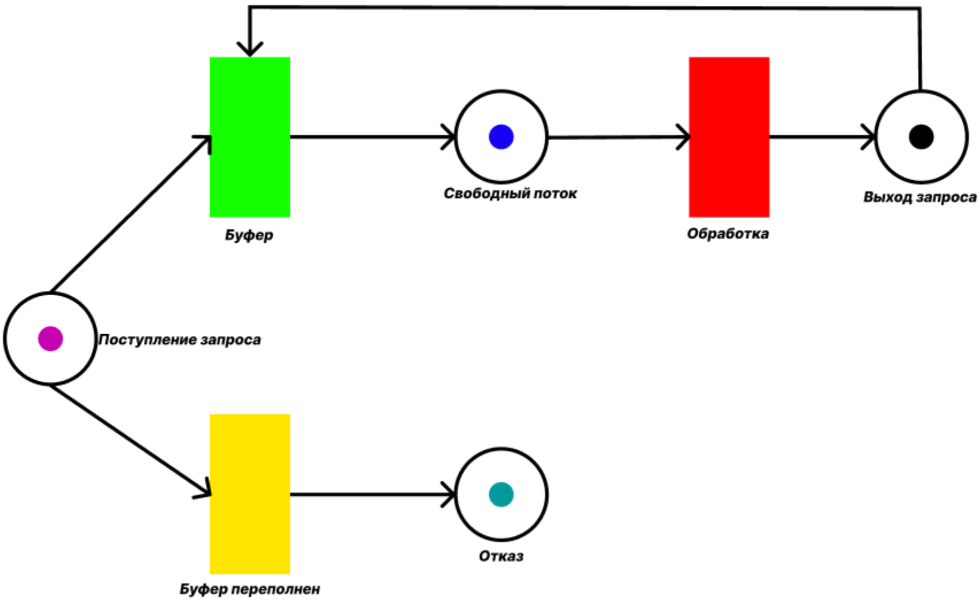

Рисунок 15 – Места и переходы.

5.6. Анализ результатов

В результате моделирования была получена визуализация работы системы при помощи графа. Можно сделать вывод, что сетевой подход отлично подходит для систем, которые имеют небольшое количество состояний. Также сетевой подход необходим для визуализации работы системы.

5.7. Выводы

Сетевой подход отлично подходит для моделирования абстрактных вычислительных систем, которые имеют небольшое количество состояний, а также помогает в визуализации системы для анализа принципа работы системы и внесения изменений в структуру работы, если это необходимо.

6. ОБОБЩЕННЫЙ ПОДХОД

6.1. Краткое теоретическое описание

Обобщенный подход к моделированию абстрактной вычислительной системы необходим для описания системы используя математические модели. Подход используется для исследования основных свойства и поведения системы. К основным этапам данного подхода можно отнести:

- Формализацию системы, то есть определить ее параметры, состояния и события;
- Математическое описание поведения системы;
- Анализ системы и проверка гипотез.

6.2. Определение ограничений, накладываемых областью применимости этого подхода

Обобщенный подход имеет ряд ограничений относительно систем, к которым он может быть применим, а именно:

- Подход плохо применим для систем с высокой сложностью событий, так как могут потребоваться ресурсоемкие вычисления;
- Системы с нелинейным поведением плохо описываются обобщенным подходом;
- При данном подходе описание ограничивается общими свойствами, то есть, когда в системе есть какие-то универсальные события, это сложно учесть.

6.3. Определение цели моделирования

Основные цели моделирования абстрактной вычислительной системы с использованием обобщенного подхода – это анализ производительности и оптимизация параметров. То есть в результате моделирования можно увидеть несовершенства системы, а также лучше изучить поведение системы и, соответственно, оценить ее производительность.

6.4. Доопределение исходных данных, недостающих для применения этого подхода

Все необходимые данные были определены ранее, однако добавим, что время, в течение которого мы будем рассматривать работу системы равняется 10 секундам. Остальные параметры остаются прежними.

6.5. Проведение моделирования

Определим обозначения для схемы:

- Z – текущий запрос;
- p – приоритет текущего запроса;
- p0, p1, p2, p3, p4 – приоритеты запросов на позициях буфера;
- t – текущее время;
- delta t – время, определенное распределением Эрланга.

На рисунке 16 представлена схема работы системы.

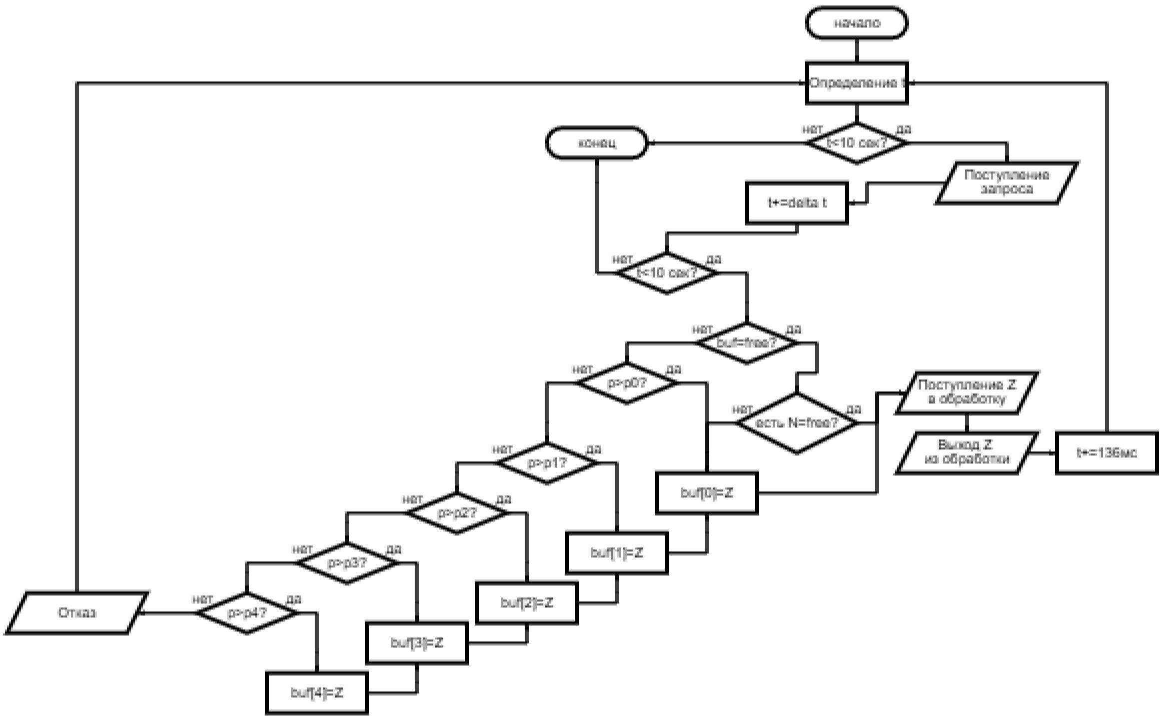

Рисунок 16 – Схема работы системы.

6.6. Анализ результатов

В результате была получена схема, которая отлично визуализирует весь цикл работы системы. В схеме представлено, как именно пришедший запрос обрабатывается системой.

6.7. Выводы

Обобщенный подход хорошо применим для моделирования систем, однако важно, чтобы исходные данные были достаточно полноценные, а также система не была слишком сложной, потому что, в первом варианте, моделирование невозможно, а во втором случае, схема получится слишком сложной, запутанной и анализ результатов будет невозможен.

ЗАКЛЮЧЕНИЕ

Кратко подвести итоги, проанализировать соответствие поставленной цели и полученного результата.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Ниже представлены примеры библиографического описания, В КАЧЕСТВЕ НАЗВАНИЯ ИСТОЧНИКА в примерах приводится вариант, в котором применяется то или иное библиографическое описание.

1. Иванов И. И. Книга одного-трех авторов. М.: Издательство, 2010. 000 с.

2. Книга четырех авторов / И. И. Иванов, П. П. Петров, С. С. Сидоров, В. В. Васильев. СПб.: Издательство, 2010. 000 с.

3. Книга пяти и более авторов / И. И. Иванов, П. П. Петров, С. С. Сидоров и др.. СПб.: Издательство, 2010. 000 с.

4. Описание книги под редакцией / под ред. И.И. Иванова СПб., Издательство, 2010. 000 с.

5. Иванов И.И. Описание учебного пособия и текста лекций: учеб. пособие. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2010. 000 с.

6. Описание методических указаний / сост.: И.И. Иванов, П.П. Петров. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2010. 000 с.

7. Иванов И.И. Описание статьи с одним-тремя авторами из журнала // Название журнала. 2010, вып. (№) 00. С. 000–000.

8. Описание статьи с четырьмя и более авторами из журнала / И. И. Иванов, П. П. Петров, С. С. Сидоров и др. // Название журнала. 2010, вып. (№) 00. С. 000–000.

9. Иванов И.И. Описание тезисов доклада с одним-тремя авторами / Название конференции: тез. докл. III международной науч.-техн. конф., СПб, 00–00 янв. 2000 г. / СПбГЭТУ «ЛЭТИ», СПБ, 2010, С. 000–000.

10. Описание тезисов доклада с четырьмя и более авторами / И. И. Иванов, П. П. Петров, С. С. Сидоров и др. // Название конференции: тез. докл. III международной науч.-техн. конф., СПб, 00–00 янв. 2000 г. / СПбГЭТУ «ЛЭТИ», СПБ, 2010, С. 000–000.

11. Описание электронного ресурса // Наименование сайта. URL: http://east-front.narod.ru/memo/latchford.htm (дата обращения: 00.00.2010).

12. ГОСТ 0.0–00. Описание стандартов. М.: Изд-во стандартов, 2010.

13. Пат. RU 00000000. Описание патентных документов / И. И. Иванов, П. П. Петров, С. С. Сидоров. Опубл. 00.00.2010. Бюл. № 00.

14. Иванов И.И. Описание авторефератов диссертаций: автореф. дисс. канд. техн. наук / СПбГЭТУ «ЛЭТИ», СПБ, 2010.

15. Описание федерального закона: Федер. закон [принят Гос. Думой 00.00.2010] // Собрание законодательств РФ. 2010. № 00. Ст. 00. С. 000–000.

16. Описание федерального постановления: постановление Правительства Рос. Федерации от 00.00.2010 № 00000 // Опубликовавшее издание. 2010. № 0. С. 000–000.

17. Описание указа: указ Президента РФ от 00.00.2010 № 00 // Опубликовавшее издание. 2010. № 0. С. 000–000.

ПРИЛОЖЕНИЕ 1
ИСХОДНЫЙ КОД ПРОГРАММЫ

Непрерывно-стохастический подход:

```
import numpy as np
import heapq
import time
import itertools
import matplotlib.pyplot as plt

# Параметры модели
LAMBDA = 80
K = 2
PROCESSING_TIME = 0.136
QUEUE_SIZE = 5
MAX_SIM_TIME = 50
MAX_REQUESTS = 500

# Статистика
processed_requests = 0
rejected_requests = 0
waiting_times = []
total_busy_time = 0.0


queue = []
current_request = None


current_time = 0.0
```

```
counter = itertools.count()

# Для построения графика
time_points = []
buffer_sizes = []

# Генерация времени между заявками
def erlang_time(lambda_, k):
    return np.sum(np.random.exponential(1 / lambda_, k))

# Генерация новой заявки
def generate_request():
    priority = np.random.randint(1, 9)  # Приоритет от 1
до 8
    return {
        "priority": priority,
        "arrival_time": current_time,
    }

# Обработка очереди
def enqueue_request(request):
    global queue, rejected_requests
    count = next(counter)
    if len(queue) < QUEUE_SIZE:
        heapq.heappush(queue, (-request["priority"],
count, request))
    else:
        lowest_priority, _, _ = queue[0]
        if -lowest_priority < request["priority"]:
```

```
            heapq.heappop(queue)
            heapq.heappush(queue,                         (-
request["priority"],           count,            request))
        else:
            rejected_requests            +=            1

#                  Моделирование                   системы
start_time                   =                 time.time()
while current_time < MAX_SIM_TIME and processed_requests
+        rejected_requests         <        MAX_REQUESTS:
    interarrival_time     =    erlang_time(LAMBDA,     K)
    current_time          +=          interarrival_time

    request              =              generate_request()
    if          current_request          is          None:
        current_request              =              request
        end_processing_time     =     current_time      +
PROCESSING_TIME
        total_busy_time        +=        PROCESSING_TIME
    else:
        enqueue_request(request)
    time_points.append(current_time)
    buffer_sizes.append(len(queue))
    while    current_request    and    current_time    >=
end_processing_time:
        processed_requests             +=             1
        waiting_times.append(current_time                -
current_request["arrival_time"])
        if                                         queue:
            _, _, current_request = heapq.heappop(queue)
```

```
            end_processing_time = current_time + PROCESSING_TIME
            total_busy_time += PROCESSING_TIME
        else:
            current_request = None


average_waiting_time = np.mean(waiting_times) if waiting_times else 0.0
efficiency = total_busy_time / current_time if current_time > 0 else 0


print("Результаты моделирования:")
print(f"Обработано заявок: {processed_requests}")
print(f"Отброшено заявок: {rejected_requests}")
print(f"Среднее время ожидания: {average_waiting_time:.4f} сек")
print(f"Суммарное время занятости процессора: {total_busy_time:.4f} сек")
print(f"Эффективность системы: {efficiency:.2%}")

# Построение графика
plt.figure(figsize=(10, 6))
plt.plot(time_points, buffer_sizes, label="Количество заявок в очереди")
plt.xlabel("Время (с)")
plt.ylabel("Размер очереди")
plt.title("Изменение количества заявок в буфере с течением времени")
```

```
plt.legend()
plt.grid()
plt.show()
```