|  |  |  |
| --- | --- | --- |
| **КУРСОВОЙ ПРОЕКТ** | | |
| МОДЕЛИРОВАНИЕ СИСТЕМЫ, ФОРМАЛИЗОВАННОЙ КАК СИСТЕМА МАССОВОГО ОБСЛУЖИВАНИЯ | | |
| по дисциплине «Архитектура программных систем» | | |
| Выполнил | | |
| студент гр. 3530904/70104 |  | Споденейко Т.А. |
| Руководитель | | |
| старший преподаватель |  | Александрова О.В. |
| «2» декабря 2019г. | | |

Оглавление

[Введение 3](#_Toc26145098)

[Исходные данные 4](#_Toc26145099)

[Временная диаграмма 5](#_Toc26145100)

[Вывод законов распределения 6](#_Toc26145101)

[Ограничения и требуемые характеристики 6](#_Toc26145102)

[Обобщенная блок-схема 7](#_Toc26145103)

[Модульная структура 8](#_Toc26145104)

[Описание работы программы 9](#_Toc26145105)

[Результаты работы 11](#_Toc26145106)

[Анализ результатов: 11](#_Toc26145107)

[Вывод 11](#_Toc26145108)

# Введение

Целью практической курсовой является создание модели ВС или ее компонентов 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. В данной работе будет использоваться имитационная модель ВС. Одним из подходов к построению имитационной модели является построение ее в виде системы массового обслуживания (СМО).

# Исходные данные

Изображение выглядит как снимок экрана

Автоматически созданное описание

**Источники:**

ИБ — бесконечный источник;

И31 — пуассоновский закон распределения заявок;

**Приборы:**

П32 — равномерный закон распределения времени обслуживания;

**Описание дисциплин постановки и выбора:**

Буферизации: Д1031 — по кольцу;

Дисциплина отказа: Д1001 — под указателем;

Дисциплина постановки на обслуживание: Д2П1 — приоритет по номеру прибора, Д2Б5 — приоритет по номеру источника;

**Виды отображения результатов работы программной модели:**

Динамическое отражение результатов: ОД1 — календарь событий, буфер и текущее состояние;

Отражение результатов после сбора статистики: ОР1 — сводная таблица результатов.

# Временная диаграмма

Изображение выглядит как текст

Автоматически созданное описание

# Вывод законов распределения

Равномерный закон распределения:

Изображение выглядит как снимок экрана

Автоматически созданное описание

В программе данное выражение записано следующим образом:

**((float)qrand() / (float)RAND\_MAX) \* (beta - alpha) + alpha;**

Пуассоновский закон распределения

Изображение выглядит как снимок экрана

Автоматически созданное описание

В программе данное выражение записано следующим образом:

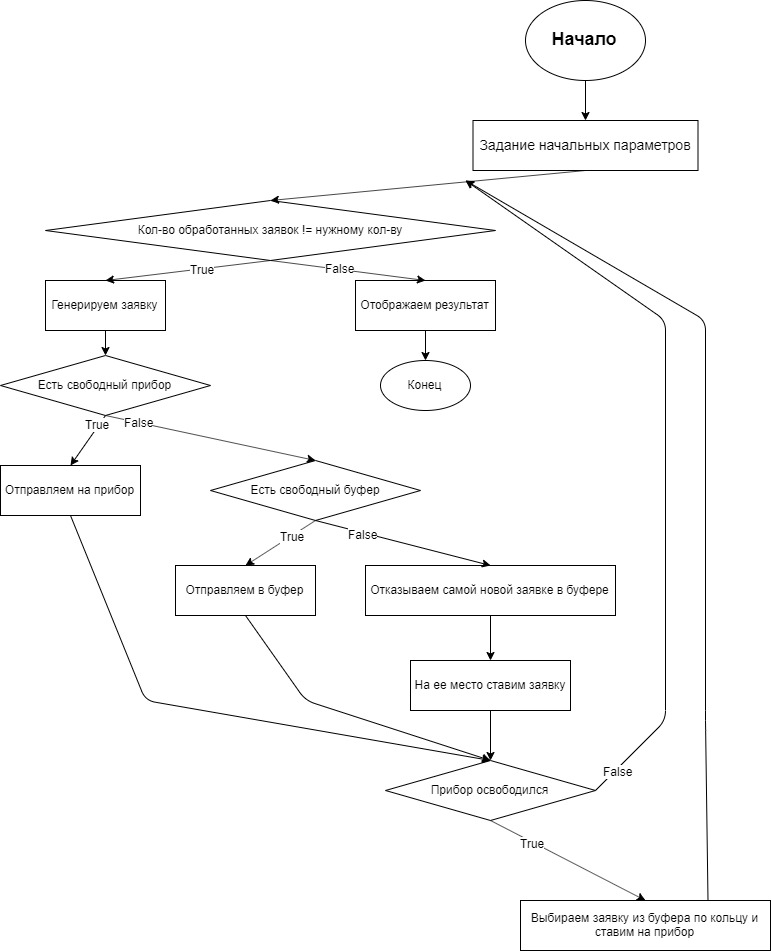
**(-1 / lambda) \* std::log(((float)qrand() / (float)RAND\_MAX));**

# Ограничения и требуемые характеристики

* Вероятность отказа должна составлять не более 10%.
* Загрузка приборов более 90%.
* Alpha: 1.0
* Beta: 1.1
* Lambda: 0.275

Время пребывания заявки в системе не ограничено, т.к. в зависимости от присланных данных, заявка может обрабатываться длительное время для получения верного результата.

# Обобщенная блок-схема



# Модульная структура

Разработка производилась в среде Qt Creator 5.13.2 на языке С++ с использованием графической библиотеки QT.

Приложение является объектно-ориентированным и содержит следующие классы:

* Request — содержит описание заявки. Реализует методы для получения этих описаний;
* Source — реализует методы создания заявки и генерации времени заявки;
* Buffer — реализует методы проверки свободного места в буфере, добавления заявки в буфер, выбора заявки из буфера;
* Device — реализует методы проверки свободных приборов, загрузки заявки на прибор, удаления заявки из прибора;
* Controller — реализует основной цикл работы системы;
* Analytics — реализует сбор аналитики для пошагового и автоматического отображения;
* Main Window — реализует графический интерфейс.

# Описание работы программы

При запуске программы появляется следующее окно:

Изображение выглядит как снимок экрана

Автоматически созданное описание

Мы находимся во вкладке Configure. В ней необходимо задать параметры для СМО и нажать кнопку Configure для сохранения параметров. Переключаясь между вкладками «Automode» и «Stepmode» мы можем наблюдать результаты работы нашей системы.

Пример отображения Automode:

Изображение выглядит как монитор, внутренний, сидит, электроника

Автоматически созданное описание

При нажатии на кнопку Refresh у нас отображаетя последний результат работы.

Пример отображения Stepmode:

Изображение выглядит как монитор, снимок экрана, внутренний, стена

Автоматически созданное описание

# Результаты работы

**Определение количества реализаций:**

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

Изображение выглядит как снимок экрана

Автоматически созданное описание

, где p—вероятность отказа заявкам в обслуживании,

=1.643 для α = 0.9,

= 0.1 — относительная точность.

По результатам работы программы получено, что в большинстве случаев для достижения заданной точности необходимо около 6000 заявок. Однако, в случаях, когда p мало (<0.05) для достижения точности в 10% может потребоваться существенно больше заявок (50000-60000).

# Анализ результатов:

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 | Размер буфера | Alpha | Beta | Lambda | Загруженность | P отк | T в сист |
| 20 | 1 | 20 | 0.5 | 1.5 | 0.8 | 99% | 95% | 0.6 |
| 15 | 2 | 15 | 0.8 | 1.3 | 0.5 | 99% | 92.5% | 0.45 |
| 10 | 3 | 10 | 1.0 | 1.1 | 0.4 | 93% | 49% | 0.55 |

Из таблицы видно, что в последнем случае мы получили необходимую загруженность приборов, но вероятность отказов не удовлетворяет нашим условиям.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Число источников | Число приборов | Размер буфера | Alpha | Beta | Lambda | Загруженность | P отк | T в сист |
| 10 | 3 | 10 | 1.0 | 1.1 | 0.3 | 94% | 19% | 0.189 |
| 10 | 3 | 10 | 1.0 | 1.1 | 0.275 | 91% | 7% | 0.12 |

Последний случай удовлетворяет всем условиям.

Если загруженность плохая, можно попробовать изменить три вещи:

* Увеличить количество источников;
* Заменить эти приборы на меньшее количество приборов с большей производительностью.

# Вывод

В ходе курсовой работы была написана система массового обслуживания на языке C++ с использованием графической библиотеки Qt. С помощью данной программы была проанализирована реальная система и подобрана максимально выгодная комплектация данной системы.