Программные системы и вычислительные методы
Правильная ссылка на статью:
Антонова П.В. — Принципы разработки систем массового обслуживания с ограниченной очередью на платформе .NET // Программные системы и вычислительные методы. – 2023. – № 2. DOI: 10.7256/2454-0714.2023.2.43403 EDN: HIXNHH URL: https://nbpublish.com/library_read_article.php?id=43403

Принципы разработки систем массового обслуживания с ограниченной очередью на платформе .NET

Антонова Полина Валерьевна

старший преподаватель кафедры интеллектуальных систем и управления информационными ресурсами Казанского национального исследовательского технологического университета

420015, Россия, республика Татарстан, г. Казань, ул. Карла Маркса, 68

valerevna.p@inbox.ru

Статья из рубрики "Автоматизированные системы управления технологическими процессами"

DOI:

10.7256/2454-0714.2023.2.43403

EDN:

HIXNHH

Дата направления статьи в редакцию:

21-06-2023

Дата публикации:

29-06-2023

Аннотация: Статья является результатом тщательного исследования, направленного на изучение принципов создания, моделирования и практического использования систем массового обслуживания (СМО) с ограниченным размером очереди, основанных на технологической платформе .NET и реализованных с применением языка программирования C#. В ходе изложения материала особое внимание уделяется как одноканальным, так и многоканальным системам. Во вступительной части статьи рассмотрены фундаментальные концепции теории массового обслуживания. Обсуждаются основные характеристики систем, такие как системы с фиксированной длиной очереди, одноканальные и многоканальные системы с вероятностью отказа в обслуживании, системы с неограниченным и ограниченным временем ожидания, замкнутые системы, а также многоканальные системы, в которых имеется взаимодействие между каналами. Представлены подробные примеры программного кода на языке C#, иллюстрирующие структуры классов, применяемые для моделирования как одноканальных, так и многоканальных СМО. Рассматриваются конкретные сценарии

использования представленных структур в рамках моделирования СМО. В статье освещается практическое применение систем массового обслуживания в решении реальных задач. На примерах из банковской сферы и управления трафиком в телекоммуникациях демонстрируется, как СМО могут способствовать оптимизации времени ожидания и эффективному управлению ресурсами. Предлагаются перспективы дальнейших исследований в области систем массового обслуживания. Учитывая важность СМО в различных отраслях, таких как банковское дело, телекоммуникации, логистика и многие другие, высокая актуальность темы обусловлена необходимостью поиска новых подходов и методов для повышения эффективности и оптимизации этих систем. В целом статья представляет собой ценный исследовательский материал для специалистов, занимающихся вопросами моделирования и практического применения систем массового обслуживания.

Ключевые слова:

системы массового обслуживания, моделирование, ограниченная очередь, одноканальные системы, многоканальные системы, Си Шарп, платформа NET, оптимизация, программные системы, вычислительные методы

Введение

Теория массового обслуживания играет ключевую роль в анализе и оптимизации процессов, связанных с обслуживанием потоков заявок в различных областях, таких как телекоммуникации, транспорт, здравоохранение и производство [1, 2]. Рассмотрим основные понятия теории массового обслуживания и типы систем обслуживания.

Системы с фиксированной очередью: в системах с фиксированной очередью количество мест в очереди ограничено. Задачи упорядочения касаются определения порядка, в котором заявки будут обслуживаться. Основные стратегии упорядочения включают FIFO (First In, First Out), LIFO (Last In, First Out) и приоритетное обслуживание.

Одноканальные и многоканальные системы с отказами: в одноканальных системах присутствует только один канал обслуживания. Если заявка приходит, когда канал занят, она может быть отклонена или поставлена в очередь. Многоканальные системы имеют несколько каналов обслуживания, что позволяет обрабатывать несколько заявок одновременно. Если все каналы заняты, заявка также может быть отклонена или поставлена в очередь.

Системы с неограниченным и ограниченным ожиданием обслуживания: в системах с неограниченным ожиданием заявки могут ожидать обслуживания неопределенно долго. В системах с ограниченным ожиданием существуют ограничения по длине очереди или времени ожидания. Если заявка не может быть обслужена в пределах этих ограничений, она отклоняется.

Замкнутые системы: в замкнутых системах заявки возвращаются в систему после обслуживания. Это часто встречается в сценариях, где ресурсы циркулируют в системе и могут быть повторно использованы.

Многоканальные системы с взаимодействием между каналами: в некоторых многоканальных системах каналы могут взаимодействовать между собой, динамически распределяя ресурсы и обмениваясь информацией для повышения эффективности

обслуживания. Это важно в сложных сетевых средах, где различные каналы могут обрабатывать разные типы заявок и необходима гибкая реакция на изменения в рабочей нагрузке.

Для анализа систем массового обслуживания и определения оптимальных стратегий обслуживания используются различные математические модели и методы, такие как марковские процессы, теория очередей и симуляционное моделирование. Марковские процессы позволяют описать случайные процессы с дискретными состояниями, теория очередей фокусируется на характеристиках потоков заявок и каналов обслуживания, а симуляционное моделирование позволяет изучать поведение системы в различных условиях.

Знание и понимание теории массового обслуживания имеют высокую практическую значимость. Например, в телекоммуникациях это помогает оптимизировать сетевые ресурсы, в здравоохранении – уменьшить время ожидания пациентов, а в производстве – повысить производительность и снизить издержки [3, 4].

Системы массового обслуживания с ограниченной очередью являются важным подклассом в теории массового обслуживания. Они характеризуются наличием верхнего предела на количество заявок, которые могут ожидать обслуживания в очереди. Как только этот предел достигнут, все последующие заявки будут отклонены до тех пор, пока место в очереди не освободится [5].

Представим аспекты систем массового обслуживания с ограниченной очередью:

– Очередь: очередь – это структура данных, которая хранит заявки, ожидающие обслуживания. В системах с ограниченной очередью количество заявок в очереди не может превышать определенный уровень, известный как вместимость очереди.

– Стратегии обслуживания: системы с ограниченной очередью могут использовать различные стратегии обслуживания, такие как FIFO (первым пришел, первым обслужен), LIFO (последним пришел, первым обслужен) или приоритетное обслуживание, где некоторые заявки могут иметь приоритет перед другими.

– Модель M/M/c/K: одна из наиболее известных моделей систем с ограниченной очередью – это модель M/M/c/K. В этой модели, интервалы времени между поступлением заявок и время обслуживания предполагается экспоненциальным (обозначается буквой 'M'), 'c' обозначает количество каналов обслуживания, а 'K' - общее количество мест в системе (количество мест в очереди плюс количество каналов).

– Анализ производительности: для анализа производительности систем с ограниченной очередью используются различные показатели, такие как вероятность блокировки (вероятность отказа новой заявке), среднее время ожидания в очереди и среднее число заявок в системе [6].

– Применение: системы с ограниченной очередью широко применяются в различных областях, таких как телекоммуникации (для управления вызовами), сетевые маршрутизаторы (для управления пакетами данных), здравоохранение (для управления потоком пациентов) и производство (для оптимизации производственных линий). Например, в телекоммуникациях системы с ограниченной очередью могут использоваться для управления входящими звонками в колл-центре. Когда все операторы заняты и очередь достигла своей максимальной вместимости, последующие звонки могут быть перенаправлены на автоматическую систему голосового ответа или получать сообщение

об отказе. В сфере здравоохранения системы с ограниченной очередью могут быть использованы в отделениях экстренной помощи, где пациенты могут быть приняты в соответствии с уровнем срочности их состояния. Когда очередь достигает своей максимальной вместимости, пациенты с менее срочными состояниями могут быть направлены в другие медицинские учреждения.

– Вызовы и стратегии оптимизации: одним из ключевых вызовов в системах с ограниченной очередью является балансировка между уровнем обслуживания и ресурсами. Слишком маленькая очередь может привести к частым отказам, тогда как слишком большая очередь может приводить к неэффективному использованию ресурсов. Оптимизация таких систем часто сводится к выбору подходящего размера очереди и числа каналов обслуживания, а также выбору эффективной стратегии обслуживания. Симуляционное моделирование и аналитические методы, такие как теория очередей, могут быть использованы для анализа и оптимизации производительности системы в различных сценариях.

Системы массового обслуживания с ограниченной очередью играют важную роль во многих областях, где необходимо эффективно управлять потоками заявок. Оптимизация таких систем является сложной задачей, которая требует анализа и использования специализированных инструментов и методов.

В статье представлены принципы реализации одноканальной и многоканальной СМО с ограниченной очередью на платформе .NET с применением языка программирования языка C#.

Актуальность применения платформы .NET для реализации систем массового обслуживания с ограниченной очередью на языке программирования C# можно рассмотреть с нескольких точек зрения:

– C# является высокопроизводительным языком программирования, а платформа .NET предоставляет современные инструменты и библиотеки для оптимизации работы с данными и ресурсами. Это позволяет создавать высокопроизводительные СМО, способные обрабатывать большое количество заявок с минимальными задержками.

– Платформа .NET и C# предоставляют встроенную поддержку многопоточности и параллельных вычислений [7, 8]. Это особенно актуально для СМО, где часто необходимо параллельно обрабатывать несколько заявок и координировать работу нескольких каналов обслуживания.

– .NET предоставляет широкий набор библиотек и инструментов, которые могут быть использованы при разработке СМО. Это включает в себя библиотеки для работы с математическими моделями, статистическим анализом, симуляционным моделированием и другими инструментами, необходимыми для эффективного проектирования систем массового обслуживания.

– Платформа .NET поддерживает кросс-платформенное развертывание, что позволяет разрабатывать приложения, которые могут работать на различных операционных системах без изменения кода. Это добавляет гибкости в развертывание и масштабирование СМО в различных средах.

– C# известен своей чистотой синтаксиса и строгой типизацией, что способствует написанию читаемого и поддерживаемого кода [9]. Это облегчает разработку сложных систем, таких как СМО, и сокращает затраты на их поддержку и развитие.

Принципы разработки одноканальной СМО с ограниченной очередью

Одноканальная система массового обслуживания с ограниченной очередью является важной моделью, использующейся в различных сферах, от телекоммуникаций до здравоохранения.

Для демонстрации реализации данного типа СМО с применением платформы .NET и языка программирования C# рассмотрим простой пример одноканальной СМО с максимальной вместимостью очереди равной 5.

Сначала создадим класс SingleServerQueue, который будет инкапсулировать логику СМО.

```
public class SingleServerQueue
{
    private Queue queue;
    private int maxQueueLength;
    private bool isServerBusy;
    public SingleServerQueue(int maxQueueLength)
    {
        this.queue = new Queue();
        this.maxQueueLength = maxQueueLength;
        this.isServerBusy = false;
    }
    public bool EnqueueCustomer(Customer customer)
    {
        if (queue.Count < maxQueueLength)
        {
            queue.Enqueue(customer);
            return true;
        }
        return false;
    }
    public void DequeueCustomer()
    {
        if (queue.Count > 0)
        {
            queue.Dequeue();
            if (queue.Count == 0)
            {
                isServerBusy = false;
            }
        }
    }
    //Дополнительные методы и свойства
}
```

Теперь, когда у нас есть основной класс, создадим симуляцию.

```
class Program
{
    static void Main(string[] args)
    {
```

```
□□□□□□□□SingleServerQueue□singleServerQueue□=□new□SingleServerQueue(5);□□□
□□□□□□□□for□(int□i□=□0;□i□<□10;□i++)□
□□□□□□□□{□
□□□□□□□□□□□□Customer□customer□=□new□Customer();□
□□□□□□□□□□□□bool□isEnqueued□=□singleServerQueue.EnqueueCustomer(customer)
□□□□□□□□□□□□if□(isEnqueued)□
□□□□□□□□□□□□{□
□□□□□□□□□□□□□□□□Console.WriteLine($"Customer□{i}□is□enqueued.");□
□□□□□□□□□□□□}□
□□□□□□□□□□□□else□
□□□□□□□□□□□□{□
□□□□□□□□□□□□□□□□Console.WriteLine($"Customer□{i}□is□rejected.");□
□□□□□□□□□□□□}□
□□□□□□□□}□
□□□□}□
}□
```

В этом примере кода создаем экземпляр одноканальной СМО с максимальной длиной очереди 5 и пытаемся добавить 10 клиентов.

После создания базовой симуляции, важным этапом является анализ и оптимизация СМО. В C# можно использовать различные библиотеки и инструменты для анализа производительности, такие как профилировщики и средства для работы со статистическими данными.

```
public□void□ProcessCustomers()
{□
□□□□while□(queue.Count□>□0)
□□□□{□
□□□□□□□□Customer□customer□=□queue.Peek();□
□□□□□□□□if□(IsServiceCompleted(customer))□
□□□□□□□□{□
□□□□□□□□□□□□DequeueCustomer();□
□□□□□□□□}□
□□□□}□
}□
private□bool□IsServiceCompleted(Customer□customer)□
{□
□□□□//□Логика□определения,□завершено□ли□обслуживание□клиента□
}□
```

В приведенном выше фрагменте кода добавляем методы для обработки клиентов в очереди и проверки завершения обслуживания. Это можно расширить с использованием статистических методов для анализа времени ожидания, эффективности обслуживания и других метрик.

Благодаря объектно-ориентированной природе C#, можно легко расширять и модифицировать СМО, добавляя новые функциональные возможности или изменяя существующие. Например, можно добавить поддержку приоритетов в очереди, внедрить алгоритмы планирования или интегрировать с внешними источниками данных.

При создании реального приложения СМО также важно предоставить интерфейс

пользователя. С использованием .NET возможно создание как консольных, так и графических пользовательских интерфейсов (GUI), что позволяет создавать удобные инструменты для мониторинга и управления СМО.

Принципы разработки многоканальной СМО с ограниченной очередью

В современных условиях, когда множество процессов требуют оптимизации и параллелизации, многоканальные системы массового обслуживания становятся все более актуальными. Они находят применение в различных отраслях: от телекоммуникаций до логистики. В данном фрагменте научной статьи рассматриваются принципы разработки многоканальной СМО с ограниченной очередью на платформе .NET с использованием языка программирования C#.

Основными компонентами СМО являются каналы обслуживания и очередь. В C# можно использовать классы и коллекции для моделирования этих компонентов. Например, очередь можно представить с помощью коллекции Queue, а каналы обслуживания – с помощью списка объектов каналов.

```
public□class□ServiceChannel
{
□□□□public□bool□IsBusy□{□get;□set;□}
□□□□//□другие□свойства□и□методы
}
public□class□MultiChannelQueueSystem
{
□□□□private□Queue□queue□=□new□Queue();
□□□□private□List□serviceChannels□=□new□List();
□□□□public□MultiChannelQueueSystem(int□numberOfChannels)
□□□□{
□□□□□□□□for□(int□i□=□0;□i□<□numberOfChannels;□i++)
□□□□□□□□{
□□□□□□□□□□□□serviceChannels.Add(new□ServiceChannel());
□□□□□□□□}
□□□□}□□□□
□□□□//□другие□методы
}
```

Обработка заявок в многоканальной СМО включает добавление заявок в очередь и их последующую обработку доступными каналами.

```
public□void□EnqueueCustomer(Customer□customer)
{
□□□□if□(queue.Count□<□MaxQueueLength)
□□□□{
□□□□□□□□queue.Enqueue(customer);
□□□□}
}
public□void□ProcessQueue()
{
□□□□foreach□(ServiceChannel□channel□in□serviceChannels)
□□□□{
□□□□□□□□if□(!channel.IsBusy□&&□queue.Count□>□0)
```

```
□□□□□□□□{
□□□□□□□□□□□□Customer□customer□=□queue.Dequeue();
□□□□□□□□□□□□//□Обработка□заявки□на□канале
□□□□□□□□}
□□□□}
}
```

После моделирования многоканальной СМО важно провести анализ ее работы. Это может включать в себя мониторинг времени ожидания в очереди, загрузки каналов и других параметров.

С использованием объектно-ориентированных подходов (ООП) подходов в C# можно создать гибкие и многократно используемые компоненты, что позволит с легкостью адаптировать систему к различным условиям и требованиям. Например, можно реализовать базовый класс для канала обслуживания и наследовать его для специальных типов каналов.

```
public□class□ServiceChannel
{
□□□□public□virtual□bool□IsBusy□{□get;□set;□}
□□□□//□другие□свойства□и□методы
}
public□class□PriorityServiceChannel□:□ServiceChannel
{
□□□□public□override□bool□IsBusy
□□□□{
□□□□□□□□get
□□□□□□□□{
□□□□□□□□□□□□//□Кастомная□логика□для□приоритетного□канала
□□□□□□□□}
□□□□□□□□set□{□/*□...□*/□}
□□□□}
}
```

Также можно добавить события и делегаты для обработки различных сценариев в системе, таких как оповещения при переполнении очереди.

Поскольку многоканальная СМО предполагает одновременное обслуживание нескольких заявок, важно эффективно использовать многопоточность [10]. В C# это можно сделать с помощью Task Parallel Library (TPL).

```
public□void□ProcessQueue()
{
□□□□Parallel.ForEach(serviceChannels,□(channel)□=>
□□□□{
□□□□□□□□if□(!channel.IsBusy□&&□queue.Count□>□0)
□□□□□□□□{
□□□□□□□□□□□□Customer□customer□=□queue.Dequeue();
□□□□□□□□□□□□//□Обработка□заявки□на□канале
□□□□□□□□}
□□□□});
}
```

При разработке СМО важно проводить тестирование и валидацию для проверки корректности работы системы. Это можно делать с использованием модульных тестов и интеграционного тестирования.

Разработка многоканальной системы массового обслуживания с ограниченной очередью на платформе .NET с использованием C# позволяет создавать эффективные и гибкие решения. Благодаря возможностям ООП, многопоточности и мощным библиотекам, C# является отличным выбором для реализации сложных СМО, способных адаптироваться к меняющимся требованиям и условиям.

Практическое использование

Системы массового обслуживания активно применяются в различных сферах, от телекоммуникаций до финансов [11, 12]. В данной части статьи мы рассмотрим примеры практической реализации одноканальной и многоканальной СМО с ограниченной очередью на платформе .NET с использованием C#.

Для моделирования одноканальной СМО с ограниченной очередью создадим класс SingleChannelQueueSystem.

```
public class SingleChannelQueueSystem
{
    private Queue queue = new Queue();
    private bool isChannelBusy = false;
    private readonly int maxQueueLength;
    public SingleChannelQueueSystem(int maxQueueLength)
    {
        this.maxQueueLength = maxQueueLength;
    }
    public void EnqueueCustomer(Customer customer)
    {
        if (queue.Count < maxQueueLength)
        {
            queue.Enqueue(customer);
        }
    }
    public void ProcessQueue()
    {
        if (!isChannelBusy && queue.Any())
        {
            var customer = queue.Dequeue();
            isChannelBusy = true;
            //Обработка клиента
            isChannelBusy = false;
        }
    }
}
```

Пример использования:

```
SingleChannelQueueSystem queueSystem = new SingleChannelQueueSystem(5);
queueSystem.EnqueueCustomer(new Customer());
```

```
queueSystem.EnqueueCustomer(new Customer());
queueSystem.ProcessQueue();
```

Для моделирования многоканальной СМО создадим класс MultiChannelQueueSystem.

```
public class MultiChannelQueueSystem
{
    private Queue queue = new Queue();
    private List serviceChannels;
    private readonly int maxQueueLength;
    public MultiChannelQueueSystem(int numberOfChannels, int maxQueueLength)
    {
        serviceChannels = new List(new bool[numberOfChannels]);
        this.maxQueueLength = maxQueueLength;
    }
    public void EnqueueCustomer(Customer customer)
    {
        if (queue.Count < maxQueueLength)
        {
            queue.Enqueue(customer);
        }
    }
    public void ProcessQueue()
    {
        for (int i = 0; i < serviceChannels.Count; i++)
        {
            if (!serviceChannels[i] && queue.Any())
            {
                var customer = queue.Dequeue();
                serviceChannels[i] = true;
                //Обработка клиента
                serviceChannels[i] = false;
            }
        }
    }
}
```

Пример использования:

```
MultiChannelQueueSystem queueSystem = new MultiChannelQueueSystem(3, 10);
queueSystem.EnqueueCustomer(new Customer());
queueSystem.EnqueueCustomer(new Customer());
queueSystem.EnqueueCustomer(new Customer());
queueSystem.ProcessQueue();
```

После создания базовых моделей СМО, важно анализировать их производительность и корректность работы. Использование инструментов профилирования, таких как Visual Studio Profiler, поможет выявить узкие места и оптимизировать код.

Для улучшения производительности многоканальной СМО можно воспользоваться параллелизацией. Это можно сделать с использованием Task Parallel Library (TPL) в C#.

```
public void ProcessQueueParallel()
```

```
{
□□□□Parallel.For(0,□serviceChannels.Count,□i□=>
□□□□{
□□□□□□□□if□(!serviceChannels[i]□&&□queue.Any())
□□□□□□□□{
□□□□□□□□□□□□var□customer□=□queue.Dequeue();
□□□□□□□□□□□□serviceChannels[i]□=□true;
□□□□□□□□□□□□//□Обработка□клиента
□□□□□□□□□□□□serviceChannels[i]□=□false;
□□□□□□□□}
□□□□});
}
```

Тестирование и валидация системы необходимы для обеспечения точности и надежности моделирования. Для этого можно использовать модульные тесты и инструменты тестирования, такие как MSTest, NUnit или xUnit.

Пример модульного теста:

```
[TestClass]
public□class□QueueSystemTests
{
□□□□[TestMethod]
□□□□public□void□TestSingleChannelQueueSystem()
□□□□{
□□□□□□□□var□queueSystem□=□new□SingleChannelQueueSystem(5);
□□□□□□□□queueSystem.EnqueueCustomer(new□Customer());
□□□□□□□□queueSystem.ProcessQueue();
□□□□□□□□//□Проверка□условий□(например,□корректное□количество□заявок□в□с
□□□□}
}
```

Одноканальные и многоканальные системы массового обслуживания могут быть применены в различных сферах, таких как телекоммуникации, банковское дело, логистика и т.д. В частности, они могут быть использованы для оптимизации времени ожидания клиентов, распределения ресурсов и улучшения общей производительности системы.

Заключение

Разработка систем массового обслуживания с ограниченной очередью представляет собой комплексный процесс, включающий моделирование, анализ, оптимизацию и создание пользовательского интерфейса. Применение современных технологий и инструментов, доступных в рамках платформы .NET, позволяет создавать эффективные и масштабируемые решения, способные справляться с реальными задачами в различных отраслях.

В статье освещены ключевые аспекты разработки, моделирования и практической реализации СМО с использованием технологий .NET и языка C#. Проведено детальное рассмотрение одноканальных и многоканальных систем, их особенностей, принципов функционирования и методов реализации.

Кроме рассмотренных аспектов существует ряд направлений для дальнейшего

исследования и развития систем массового обслуживания:

– Интеграция с искусственным интеллектом: с применением искусственного интеллекта можно улучшить процессы принятия решений в системе, например, предсказывать поток заявок и динамически изменять параметры системы для оптимизации обслуживания.

– Использование облачных технологий: интеграция с облачными платформами может обеспечить повышение доступности, масштабируемости и надежности системы, а также снижение затрат на инфраструктуру.

– Расширенная аналитика и визуализация: разработка инструментов аналитики и визуализации данных позволит администраторам и пользователям системы более эффективно мониторить и анализировать ход обслуживания и использовать эту информацию для оптимизации процессов.

– Обеспечение безопасности: безопасность данных и транзакций является ключевым аспектом в современных системах массового обслуживания [13, 14]. Разработка и интеграция механизмов защиты от несанкционированного доступа и вредоносного программного обеспечения является важной частью создания надежной системы.

Библиография

1. Harchol-Balter M. Performance modeling and design of computer systems: queueing theory in action / Cambridge University Press, 2013.
2. Шевцов А. Н., Щитов А. Н., Конорева Н. А. Моделирование телекоммуникационных систем с помощью СМО // Математика и ее приложения в современной науке и практике, 2015. С. 128-132.
3. Вишневский В. М., Дудин А. Н. Системы массового обслуживания с коррелированными входными потоками и их применение для моделирования телекоммуникационных сетей // Автоматика и телемеханика. 2017. № 8. С. 3-59.
4. Gross D., Harris C. Fundamentals of Queueing Theory / Wiley-Interscience, 1998.
5. Осипов Л. А. Имитационное моделирование систем массового обслуживания с ограниченной очередью // Наука и техника транспорта. 2010. № 4. С. 30-36.
6. Фурина К. О., Осечкина Т. А. Математическая модель одноканальной системы массового обслуживания с ограниченной очередью // Наука и современность. 2014. № 2. С. 103-110.
7. Гибадуллин Р.Ф. Потокобезопасные вызовы элементов управления в обогащенных клиентских приложениях // Программные системы и вычислительные методы. 2022. № 4. С. 1-19.
8. Гибадуллин Р.Ф., Викторов И.В. Неоднозначность результатов при использовании методов класса Parallel в рамках исполняющей среды .NET Framework // Программные системы и вычислительные методы. 2023. № 2. С. 1-14.
9. Albahari J. C# 10 in a Nutshell / O'Reilly Media, Inc., 2022.
10. Викторов И.В., Гибадуллин Р.Ф. Разработка синтаксического дерева для автоматизированного транслятора последовательного программного кода в параллельный код для многоядерных процессоров // Программные системы и вычислительные методы. 2023. № 1. С. 13-25.
11. Осипов Г. С. Исследование систем массового обслуживания с ожиданием в AnyLogic // Бюллетень науки и практики. 2016. № 10 (11). С. 139-151.
12. Осипов Г. С. Системы массового обслуживания с ограниченной длительностью ожидания // Бюллетень науки и практики. 2016. № 12 (13). С. 28-36.

13. Гибадуллин Р.Ф., Гашигуллин Д.А., Вершинин И.С. Разработка декоратора StegoStream для ассоциативной защиты байтового потока. Моделирование, оптимизация и информационные технологии. 2023. 11(2). URL: moitvivt.ru/ru/journal/pdf?id=1359.
14. Гибадуллин Р.Ф., Вершинин И.С., Глебов Е.Е. Разработка приложения для ассоциативной защиты файлов // Инженерный вестник Дона. 2023. № 6. URL: ivdon.ru/ru/magazine/archive/n6y2023/8462/.

Результаты процедуры рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.
Со списком рецензентов издательства можно ознакомиться здесь.

Предметом исследования в рецензируемой статье выступают Принципы разработки систем массового обслуживания с ограниченной очередью.
Методология исследования базируется на изучении отечественного и зарубежного опыта разработки систем массового обслуживания на языке C#.
Актуальность работы авторы связывают с тем, что системы массового обслуживания играет ключевую роль в анализе и оптимизации процессов, связанных с обслуживанием потоков заявок в различных областях, таких как телекоммуникации, транспорт, здравоохранение и производство.
Научная новизна рецензируемого исследования, к сожалению, не сформулирована автором, и после прочтения статьи не ясно в чем состоит приращение научного знания.
Структурно в статье выделены следующие разделы: Введение, Принципы разработки одноканальной системы массового обслуживания (СМО) с ограниченной очередью, Принципы разработки многоканальной СМО с ограниченной очередью, Практическое использование, Заключение, Библиография.
Авторы в начале статьи приводят основные понятия теории массового обслуживания и рассматривают типы систем массового обслуживания. Для демонстрации реализации данного СМО с применением платформы .NET и языка программирования C# в работе приводятся фрагменты программного кода на языке C#. В публикации рассматривается, как после создания базовой симуляции провести анализ и оптимизацию СМО с использованием различных библиотек и инструментов анализа производительности (профилировщиков и средств для работы со статистическими данными). Особое внимание уделено практической реализации СМО с ограниченной очередью на платформе .NET с использованием C#. В заключении рассмотрены перспективы развития систем массового обслуживания.
Библиографический список включает 14 источников – публикации российских и зарубежных авторов по теме статьи. В тексте имеются адресные отсылки на библиографические источники, что подтверждает наличие апелляции к оппонентам.
Из недостатков, следует отметить следующие. Во-первых, в названии статьи используется аббревиатура (акроним), что является не лучшим вариантом, поскольку в научных публикациях принято после первого употребления сокращений приводить полную расшифровку слов, а в названии статьи этого по понятным причинам сделать не удастся. Автору предлагается рассмотреть следующую редакцию названия статьи: «Принципы разработки систем массового обслуживания с ограниченной очередью на современных программных платформах». Во-вторых, в тексте статьи не указан такой общепринятый в современных научных публикациях элемент как методы исследования. В-третьих, в статье не сформулирована цель и не отражены элементы научной новизны.

В-четвертых, объем статьи представляется уместным сократить за счет исключения из текста публикации общеизвестных положений теории массового обслуживания, изложенных в начале статьи и повторяющих материал учебной литературы.

Рецензируемый материал соответствует направлению журнала «Программные системы и вычислительные методы», отражает результаты проведенной авторами работы, имеет практическое значение для разработки систем массового обслуживания с ограниченной очередью на платформе .NET, может вызвать интерес у читателей, однако, статья нуждается в доработке в соответствие с высказанными замечаниями.