

УДК 519.872

РАЗРАБОТКА И АНАЛИЗ МАТЕМАТИЧЕСКОЙ МОДЕЛИ ДИСПЕТЧЕРА ЗАДАЧ РЕКОНФИГУРИРУЕМОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

Мартышкин А.И.*ФГБОУ ВО «Пензенский государственный технологический университет», Пенза,
e-mail: Alexey314@yandex.ru*

В статье представлены результаты по разработке и исследованию математической модели диспетчера задач с общей очередью реконфигурируемой вычислительной системы. В статье проведен анализ характеристик аналитических и имитационных моделей диспетчеров задач. Результаты исследования диспетчеров задач с общей очередью показали свою эффективность. Данный вывод подтвержден, исходя из анализа результатов исследования загрузки диспетчера задач, времени реакции, средней длины очереди и т.д. В ходе проведенных исследований и разработок предложен макетный прототип диспетчера задач. В процессе разработки потребовалось исследовать способы и методы организации работы диспетчеров задач. Рассмотренный в статье диспетчер задач синтезирован в соответствии с известным принципом «разделение времени». В процессе написания статьи учтен опыт по реализации диспетчеров задач. Система соответствует дискретно-событийной модели и способна обрабатывать заявки в многопоточном режиме. Для проведения вычислительного эксперимента задаваемые параметры исследуемой реконфигурируемой вычислительной системы брались исходя из параметров существующих устройств (в части влияния трудоемкости задач в реконфигурируемой вычислительной системе на загрузку диспетчера задач), для того чтобы результаты экспериментов были приближены к реальным показателям. Полученные результаты помогают выполнить более детальный анализ применения подобных систем. Проведено исследование технических основ функционирования диспетчера задач в реконфигурируемых системах. В заключение приводятся основные выводы по проведенному исследованию.

Ключевые слова: аналитическое моделирование, диспетчер задач, имитационное моделирование, математическая модель, планирование процессов, приоритет, реконфигурируемая вычислительная система, система массового обслуживания

DEVELOPMENT AND ANALYSIS A MATHEMATICAL MODEL OF THE TASK MANAGER OF A RECONFIGURABLE COMPUTING SYSTEM

Martyshkin A.I.*Penza State Technological University, Penza, e-mail: Alexey314@yandex.ru*

The article presents the results of the development and research of a mathematical model of a task manager with a common queue of a reconfigurable computing system. The article analyzes the characteristics of analytical and simulation models of task controllers. The results of the study of dispatchers of tasks with a common queue showed how effective. This conclusion is confirmed based on the analysis of the results of the task manager load study, reaction time, average queue length, etc. In the course of research and development, a mock-up prototype of the task manager was proposed. During the development process, it was necessary to investigate the ways and methods of organizing the work of task managers. The task manager considered in the article is synthesized in accordance with the well-known principle of “time sharing”. In the process of writing the article, the experience of implementing task managers is taken into account. The system corresponds to a discrete-event model and is capable of processing applications in multithreaded mode. To conduct a computational experiment, the parameters of the reconfigurable computing system under study were taken based on the parameters of existing devices (in terms of the impact of the complexity of tasks in the reconfigurable computing system on the loading of the task manager), so that the experimental results were close to real indicators. The results obtained help to perform a more detailed analysis of the use of such systems. A study of the technical foundations of the functioning of the task manager in reconfigurable computing systems has been carried out. At the article concludes, the conclusions of the study are presented.

Keywords: analytical modeling, task manager, simulation modeling, mathematical model, process planning, priority, reconfigurable computing system, queuing system

С увеличением вычислительных мощностей большую популярность набирают реконфигурируемые (многопроцессорные, многоядерные, распределенные) вычислительные системы (РВС), где возрастает число обрабатываемых задач. Известны технические средства по поддержанию работы диспетчера задач (ДЗ) согласованной в течение всего периода обработки поступающих задач. В предлагаемой статье рассматриваются вопросы исследования особенностей

ДЗ РВС при помощи его математической модели. Сегодня существует несколько возможных архитектур ДЗ, среди которых разделение времени и разделение пространства [1, 2]. В статье исследуются упомянутые архитектурные подходы, выполняется их сравнение. Прототип ДЗ реализован программно, описаны этапы его разработки, выбраны и обоснованы используемые технологии, проведено тестирование и проверена корректность полученных результатов.

Исходя из вышеописанного, целью данной работы является разработка и исследование модели и прототипа ДЗ РВС. В процессе работы исследованы способы организации ДЗ в соответствии с архитектурой «разделение времени». Предлагаемый прототип по своему функционалу опрашивает очередь на предмет наличия ожидающих задач и обрабатывает их в многопоточном режиме.

Материалы и методы исследования

Дискретно-событийное моделирование (ДСМ) [3, 4] – подход к моделированию, предполагающий абстрагирование от природы событий и рассмотрение процессов, описываемых моделью как набором отдельных независимых событий. В ДСМ выделяют основные состояния, в которых может находиться система, и описывающие ситуации, которые вызывают смену одного состояния другим. ДСМ находит широкое применение в самых разнообразных сферах деятельности человека: начиная от прикладных задач, заканчивая классическими системами массового обслуживания (СМО) [5, 6].

Суть дискретно-событийного моделирования можно рассмотреть на примере классической задачи о «спящем парикмахере» [1]. Клиенты, приходящие в парикмахерскую, занимают место в очереди, затем свободный парикмахер по очереди приглашает клиентов на обслуживание. После стрижки клиент оплачивает проведенную работу. Целью такого исследования является оценка эффективности работы СМО. В процессе решения задач получают конкретные значения характеристик, описывающие качество обслуживания. Для подобной системы оценивают время оказания услуги. Клиенты, посещающие парикмахерскую во время ее работы, ожидают как можно скорее получить качественную услугу. Описанную систему можно представить в виде математической модели с некоторым числом объектов, причем клиенты, приходящие для исполнения своих потребностей, будут считаться задачами, ожидающими обработки, а сотрудник – обработчиком. Структура моделируемой системы должна соответствовать реальной модели массового обслуживания: заявки (клиенты) поступают в систему; становятся в очередь на обслуживание (парикмахерам); выходят из системы после завершения обработки. Существует ряд программ и программных комплексов для аналогичного моделирования [7, 8].

Ввиду поступления событий на обработку в единую (глобальную) очередь и их обслуживания независимо друг от друга, имеется возможность использования параллельной обработки данных заявок. Наличие такой возможности позволяет запускать обработку заявок не только в многопоточном режиме, но и организовать обработку несколькими системами. В статье при организации работы ДЗ и представления его математической модели используем способ событийного алгоритма [9], что обусловлено возможностью повышения скорости обработки задач за счет применения многопоточных инструментов.

Организация и функционирование ДЗ. Сегодня популярность получили высоконагруженные приложения и программы, для обеспечения приемлемой работы которых требуются программы с минимальным откликом и оптимальным распределением задач. Для соответствия данным требованиям и обеспечения эффективной работы вычислительных систем в режиме реального времени используются реконфигурируемые архитектуры, способные выполнять операции в многопоточном режиме. Разработка подобных систем и поддержание параллельного выполнения исходного кода сопряжены с управлением процессами и выделением свободных ресурсов путем диспетчеризации. Основным показателем эффективности работы ДЗ является скорость переключения между процессами и передачи управления конкретному потоку. Для выполнения таких задач используется предварительный планировщик и непосредственно ДЗ. При использовании современных РВС нагрузка на указанные механизмы значительно возрастает, вследствие чего могут возникать конфликты за программные ресурсы, которые необходимо решать. Классическая программная реализация ДЗ не всегда соответствует допустимому минимуму производительности системы. Поэтому данная проблема является актуальной для корректной работы высоконагруженных систем.

Задача диспетчеризации процессов в РВС состоит из назначения задачи и определения целевого процессора. Другим важным критерием эффективной работы ДЗ является сбалансированность нагрузки различных процессоров, выполняющих задачи.

В ходе многочисленных попыток реализации ДЗ сформировались некоторые способы организации работы ДЗ: с разделением времени и разделением пространства [1]. Подход в построении ДЗ с разделением пространства работает по принципу, описанному в [10].

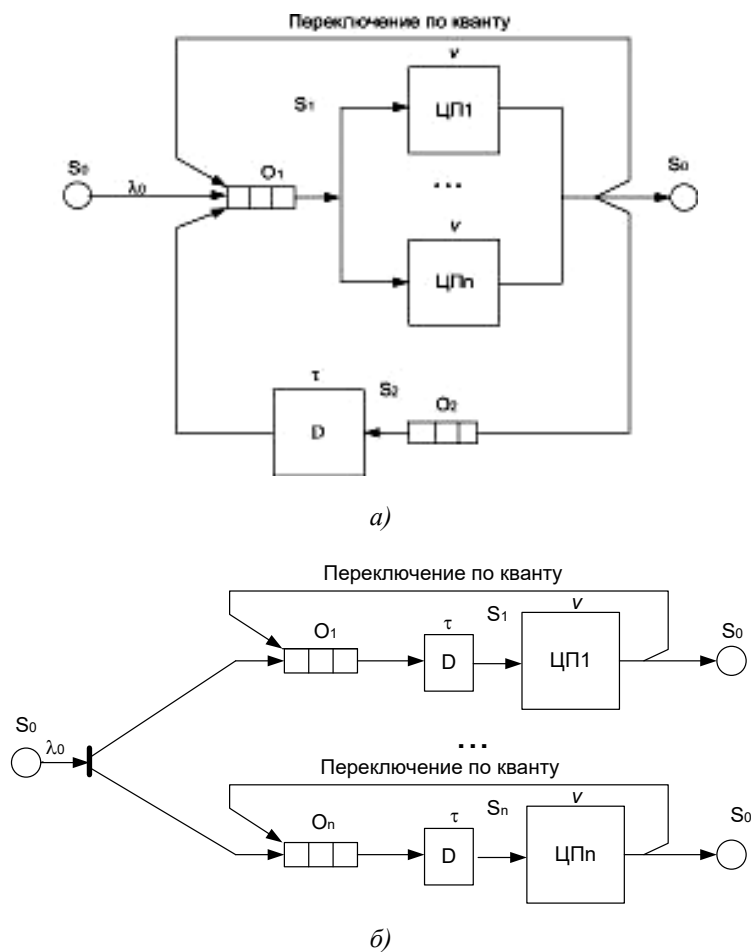


Рис. 1. Современные архитектуры ДЗ: принцип «разделение времени» (а) и «разделение пространства» (б) [1]

Для каждого процессорного узла выделяется место под хранение задач для обработки. В тот момент, когда процессор закончил обработку текущей задачи, он уведомляет ДЗ, который в свою очередь выделяет ему новую задачу. В случае возникновения прерывания процессор помещает задачу в свою локальную память и продолжает ее обработку позднее. Для работы системы необходим так называемый планировщик задач, который принимает решение, какой поток должен получить конкретную задачу. В ДЗ, построенных по принципу разделения времени, имеется глобальная очередь. Это может привести к перегрузке кэш-памяти в случае возникновения прерывания и необходимости переключения задач, исполняющихся в текущий момент. Когда задача отправляется на дополнительную обработку, происходит увеличение занимаемой памяти, кэш-память начинает выполнять лишние действия, происходит увеличение количества кэш-промахов. Несмотря на это, дан-

ный архитектурный подход предоставляет достаточную степень параллелизма и делает работу ДЗ приемлемо эффективной. Существует высокое разнообразие подходов планирования и диспетчеризации. Большинство из них основывается на правилах составления очереди готовых к выполнению задач и правилах, по которым конкретная задача будет выбрана для исполнения и отправлена в обработчик. Ограниченность ресурсов системы – основная проблема, которая должна быть решена при разработке ДЗ. Диспетчеризация потоков и их правильный конкурентный доступ к разделяемым ресурсам рассматриваются в качестве основных способов решения подобной проблемы. Основным критерий эффективности функционирования РВС – производительность. Для получения конкретных числовых значений возможных издержек в производительности принято использовать аналитические модели с конкретным типом архитектуры. Описывают аналогич-

ные модели разомкнутые сети массового обслуживания (РСМО). Для описания ДЗ с общей глобальной очередью используется многоканальная СМО, так как поступающая в систему задача может быть отправлена как из стороннего источника, так и после прерывания текущей задачи контекста и перенаправляется любому свободному в данный момент процессору. Если в системе имеется глобальный накопитель заявок, доступный всем процессорным узлам, модель описывается одноканальной СМО. ДЗ с локальными очередями представляет собой совокупность нескольких одноканальных СМО, каждая из которых иллюстрирует работу процессора с собственным ДЗ [10].

При использовании математической модели с временным разделением, пришедшая в систему задача вначале поступает в единую глобальную очередь и находится там, пока не найдется процессор, готовый ее обработать. ДЗ, спроектированный по методу разделения пространства, отправляет новую задачу, полученную обработчиком, в локальную очередь свободного процессора. Если в накопителе имеются задачи, осуществляется поиск задачи, подходящей в текущий момент для обработки. Возможна ситуация, что задач на данный момент нет, тогда система переходит в режим ожидания и находится там до появления необработанных заявок. Когда задача поступает в очередь, происходит сортировка в соответствии с приоритетом [11]. Заявка, поступившая в процессорный узел, начинает обслуживаться обработчиком. В случае успешного завершения обработки процессор отправляет уведомление с соответствующим кодом ДЗ и пользователю. Если задача не может быть обработана, процессор прерывает ее выполнение и отправляет в конец той локальной очереди, из которой она пришла.

В стандартных приложениях создаются некоторые потоки задач, а время, необходимое для их обработки, описывается экспоненциальной формулой. Данное предположение позволяет получить верхнюю границу временных значений. Чаще всего она будет отличаться от реальных, но эти значения можно использовать для приблизительных оценок.

В ДЗ с общей глобальной очередью это время описывается выражением

$$t = k \cdot (v + pw), \quad (1)$$

где v – время обработки заявки процессором; p – вероятность возникновения прерывания; w – время простоя кэш-памяти; k – среднее квантовое число для процессинга задачи.

В ДЗ с локальными очередями время, затрачиваемое на обработку, определяется выражением

$$t = k \cdot (v + \tau), \quad (2)$$

где τ – время, затрачиваемое ДЗ на переключение контекста.

Существуют различные дисциплины организации очереди в ДЗ РВС, например, с приоритетами [12]. В [10] проведено исследование особенностей работы ДЗ с архитектурой, построенной по принципу разделения времени. При анализе применялось имитационное моделирование системы. Для проверки корректности модели рассматривались указанные ниже критерии: скорость потока заявок с разной степенью нагрузки процессора составляла: $\lambda_0 = 0,03; 0,06; 0,09$ задач/мкс; время для переключения контекста – 10 мкс; вероятность успешной обработки – 5%; вероятность прерывания – 95%; время обработки заявки – 7 мс; емкость накопителя – 150 элементов.

В результате исследования выявлено, что ДЗ с такими характеристиками пригодны для использования в системах, подобных тем, что используются в настоящей статье. Учитывая время отклика, можно сказать, что рекомендуется использование процессоров, количество которых не превосходит 15. В случае если система должна обрабатывать более мощные заявки, следует задуматься об использовании более мощных аппаратных средств, которые могли бы улучшить качество и время работы ДЗ. Также были выведены способы получения приблизительных временных показателей, по которым можно оценить время нахождения заявки в накопителе. Данные значения подтверждены имитационным моделированием. Полученная погрешность моделирования составила значение не более 0,15, что свидетельствует о допустимом уровне погрешности на начальном этапе проектирования.

Стратегия с разделением времени. Моделируемая система не может в полной мере соответствовать построенной модели. Она обычно представляет собой упрощенный прототип реальной системы. В приведенной модели могут быть опущены некоторые свойства и параметры, упрощена сама модель. Все это может исказить результаты исследования модели в дальнейшем. Для проведения исследования использована базовая СМО [13]. Такой метод хорошо подходит для проведения испытания, так как основные исследуемые характеристики соответствуют реальным. Ввиду того, что мы не ожидаем в реальной системе столь огромного числа заявок, мы можем пренебречь этим отличием. Ключевым является

то, что следующие параметры у реальной и моделируемой систем не различаются: число процессоров от 2 до 18; частота поступления задач 1-го вида – от 7,2 до 72 задач/мс; частота поступления задач 2-го вида – от 0,416 до 3,56 задач/мс; частота поступления задач 3-го вида – от 0,23 до 2,3 задач/мс. В настоящее время технические возможности позволяют выполнять очень трудоемкие задачи за сравнительно приемлемое время. Подобные системы показывают высокие показатели перформанса и способны обрабатывать задачи из накопителя с низкой трудоемкостью. В известных публикациях подобные задачи называются короткими [1, 9]. Благодаря проведенному моделированию получены экспериментальные значения для некоторых характеристик ДЗ. С учетом дополнительной нагрузки на работу с кэшем и прочие временные затраты можно с верхней границей оценить время работы ДЗ в виде 0,017 мс. Для наглядного представления рассмотрим модель с общим накопителем и конечным числом источников, генерирующих заявки, предназначенные для обработки ДЗ. На рис. 2 приведена графовая модель системы, в которой выделены составные элементы: S0 – предварительный планировщик заявок; S1 – процессор; S2 – ДЗ с временным разделением.

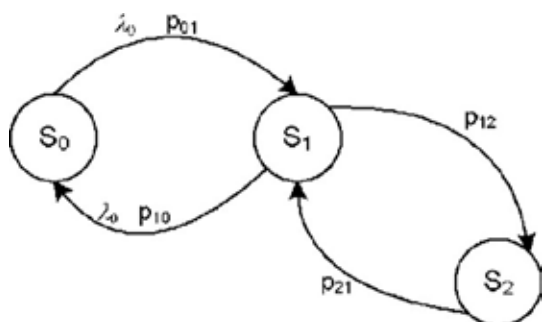


Рис. 2. Графовая модель системы с ДЗ с временным разделением

Значения вероятностей переходов (p) из различных состояний зависят от сложности, поступившей заявки. Между трудоемкостью и временем, которое заявка проводит в системе, имеется прямая пропорциональная зависимость: чем сложнее заявка, тем дольше она будет находиться на обработке. В свою очередь, вероятность того, что заявка будет отправлена на дополнительную обработку, также увеличивается, ввиду возрастания сложности задачи. При моделировании исследуемой системы подставлялись различные значения для возможных вероятностных значений. Расчеты описанного выше моделирования произво-

дились в специализированной программе расчета характеристик [6].

Имитационное моделирование. Для подтверждения корректности описанных моделей необходимо провести имитационное моделирование, подразумевающее построение системы, работающей в распределенном режиме. Модель предназначена для запуска обработки поступающих в накопитель задач. Для расчета времени, затрачиваемого на обработку задач, вычисляется единичное время, отводимое на обработку одной задачи. Как отмечалось ранее, модель будет представлена в виде нескольких процессоров с общим накопителем. При исследовании приняты параметры, соответствующие трудоемкости реальных систем (высокая трудоемкость: число процессоров менялось от 4 до 18, среднее время обработки на процессоре – 0,009 мс; средняя трудоемкость: число процессоров менялось от 4 до 18, среднее время обработки на процессоре – 0,068 мс; низкая трудоемкость: число процессоров менялось от 4 до 18, среднее время обработки на процессоре – 0,15 мс). Представленный диапазон значений интенсивности обеспечивает среднюю загрузку ПУ на уровне 65–67%. Проводилось исследование зависимостей загрузки ДЗ от количества процессоров. Модели демонстрировали загрузку до 25% при 16 процессорах в РВС.

На рис. 3 показано влияние трудоемкости задач в РВС на загрузку ДЗ с общей очередью: с понижением трудоемкости задач возрастает загрузка ДЗ в системе.

Разработка ДЗ. Система-прототип обработки поступающих в очередь заявок представляет собой класс, принимающий на вход функцию, которую будет использовать для обработки заявок, а также входные параметры, которые будут обрабатываться по указанному правилу. Данный класс обобщенно можно называть обработчик. Он используется только для обработки входных данных и возврата результата этой обработки. Возвращаемый результат обработки данных зависит от вызова функции. В случае если функция, с которой работает обработчик, возвращает информацию о том, что обработка выполнена успешно, получены какие-либо конечные результаты, то этот результат передается в наружные модули, которые в дальнейшем будут собирать статистику и передавать данные пользователю. В случае если функция вернула исключительный ответ на указанные входные параметры, выполняется процедура по повторной обработке текущей заявки, подразумевающая, что, если функция $F(x)$ прервалась в какой-то момент выполнения t , заявка будет отправлена очередь и ее обработка будет продолжена с этого момента t .

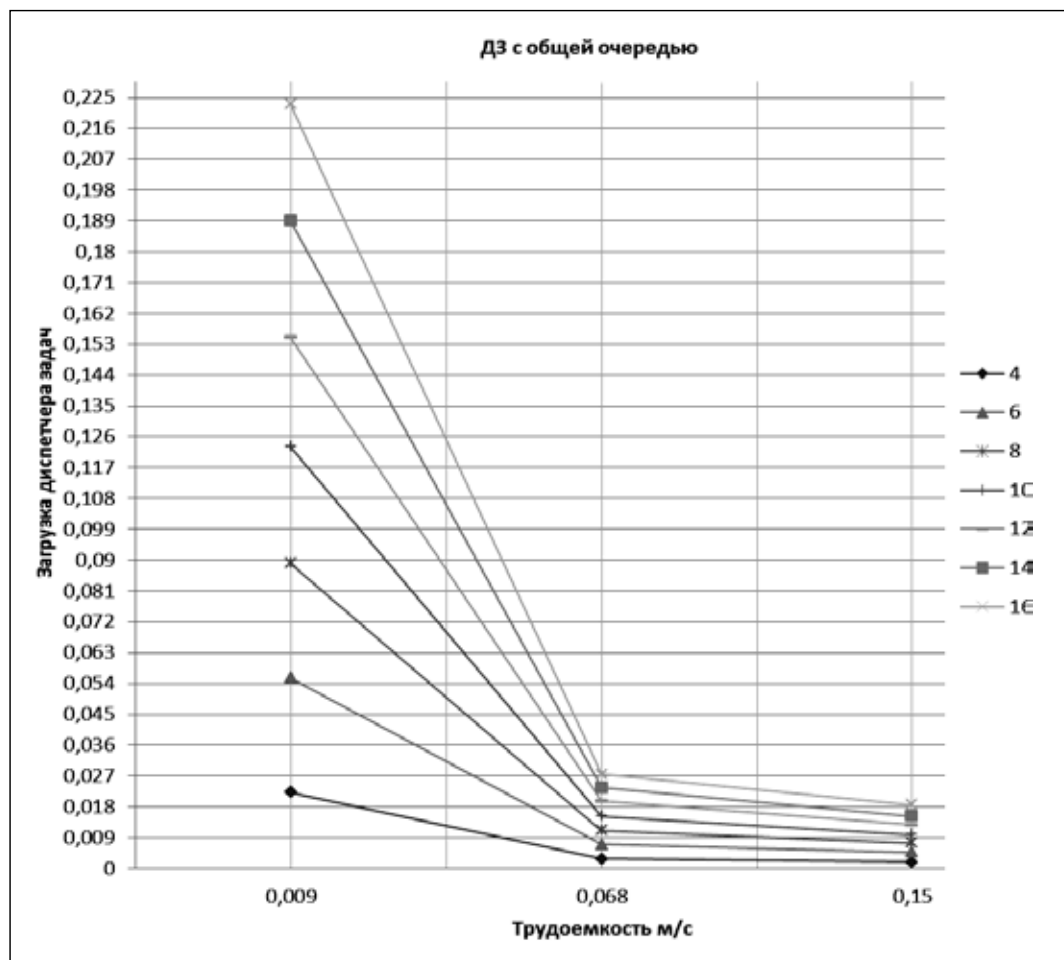


Рис. 3. Влияние трудоемкости задач в РВС на загрузку ДЗ с общей очередью

```

ПотокThread-0:Заявка Event(id=0, input=66) обработана успешно: SuccessResult(resultValue=66)
ПотокThread-10:Заявка Event(id=0, input=1) обработана успешно: SuccessResult(resultValue=1)
ПотокThread-3:Заявка Event(id=-1, input=32) обработана успешно: SuccessResult(resultValue=32)
ПотокThread-2:Заявка Event(id=0, input=11) обработана успешно: SuccessResult(resultValue=11)
ПотокThread-4:Заявка Event(id=-1, input=42) обработана успешно: SuccessResult(resultValue=42)
ПотокThread-11:Заявка обрабатывается
Создается альтернативная заявка
ПотокThread-11:Заявка Event(id=-1, input=11) обработана успешно: SuccessResult(resultValue=11)
ПотокThread-12:Заявка обрабатывается
Создается альтернативная заявка
ПотокThread-12:Заявка Event(id=-1, input=21) обработана успешно: SuccessResult(resultValue=21)
ПотокThread-8:Заявка Event(id=-1, input=41) обработана успешно: SuccessResult(resultValue=41)
ПотокThread-6:Заявка Event(id=-1, input=21) обработана успешно: SuccessResult(resultValue=21)
ПотокThread-7:Заявка Event(id=-1, input=31) обработана успешно: SuccessResult(resultValue=31)
ПотокThread-1:Заявка Event(id=0, input=22) обработана успешно: SuccessResult(resultValue=22)
ПотокThread-13:Заявка обрабатывается
Создается альтернативная заявка
ПотокThread-13:Заявка Event(id=-1, input=31) обработана успешно: SuccessResult(resultValue=31)
ПотокThread-14:Заявка обрабатывается
Создается альтернативная заявка
ПотокThread-14:Заявка Event(id=-1, input=41) обработана успешно: SuccessResult(resultValue=41)
ПотокThread-15:Заявка обрабатывается
    
```

Рис. 4. Фрагмент работы программы

Результаты исследования и их обсуждение

Результаты проведенных исследований с учетом характеристик и особенностей модели ДЗ на основе подхода с разделением времени говорят о целесообразности использования именно ее. Этот выбор обоснован в первую очередь отсутствием простоя процессоров и их равномерной загрузкой. Также данная модель функционирования ДЗ позволяет гибко организовать сбор статистики в дальнейшем использовании.

Рассмотрим экспериментальный вариант работы системы-прототипа. Для тестирования и сравнения ожидаемых результатов с фактическими выбрана тестовая функция: `if (param > 50) {return ОК; } else { return to Queue with (param + 10);}`.

В случае если входное число больше 50, считаем, что функция корректно обработала параметр. Иначе прибавляем 10 к исходному числу и перезапускаем обработку заявки в очереди. Запускаем программу с такими параметрами и проверяем результат работы системы-прототипа (рис. 4).

Из рис. 4 видно, что каждая заявка выполняется в отдельном потоке. Исходя из этого, корректность работы системы-прототипа можно считать допустимой.

Заключение

В статье проведен анализ характеристик аналитических и имитационных моделей ДЗ. Результаты исследования ДЗ с общей очередью показали свою эффективность. Данный вывод подтвержден, исходя из анализа графиков загрузки ДЗ и т.д.

В ходе выполнения работы предложен прототип ДЗ. В процессе разработки системы потребовалось исследовать способы организации работы ДЗ. Предлагаемый в статье ДЗ разработан в соответствии с широко известной архитектурой «разделение времени». Система соответствует дискретно-событийной модели и способна обрабатывать заявки в многопоточном режиме. Полученные результаты помогают выполнить более детальный анализ применения подобных систем. Кроме того, результаты будут полезны исследователям, занимаю-

щимся реконфигурируемыми вычислительными системами.

Исследование выполнено за счет гранта Российского научного фонда № 21-71-00110, <https://rscf.ru/project/21-71-00110>.

Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. СПб.: Питер, 2015. 1120 с.
2. Орлов С.А., Цилькер Б.Я. Организация ЭВМ и систем: учебник для вузов. 3-е изд. Стандарт третьего поколения. СПб.: Питер, 2014. 688 с.
3. Илюхина Н.А., Комаревцева О.О. Дискретно-событийное моделирование в управлении экономической системы муниципального образования // Современные наукоемкие технологии. 2015. № 7. С. 77–80.
4. Cassandras C.G. and Lafortune S. Introduction to Discrete Event Systems, Cham, Switzerland: Springer, 2021.
5. Клейнрок Л. Вычислительные системы с очередями. М.: Мир, 1979. 600 с.
6. Майоров С.А., Новиков Г.И., Алиев Т.И., Махарев Э.И., Тимченко Б.Д. Основы теории вычислительных систем: учебное пособие для вузов / Под ред. С.А. Майорова. М.: Высшая школа, 1978. 409 с.
7. Мартышкин А.И. Программа для расчета основных вероятностно-временных характеристик подсистемы планирования и диспетчеризации многопроцессорных систем // Свидетельство о регистрации программы для ЭВМ RU 2016619353, 18.08.2016. Заявка № 2016616801 от 27.06.2016.
8. Мартенс-Атюшев Д.С., Мартышкин А.И. Программный комплекс моделирования подсистемы «процессор – память» специализированной реконфигурируемой многопроцессорной системы // Свидетельство о регистрации программы для ЭВМ 2021619274, 08.06.2021. Заявка № 2021615965 от 23.04.2021.
9. Алиев Т.И. Основы моделирования дискретных систем. СПб: СПбГУ ИТМО, 2009. 363 с.
10. Мартышкин А.И. Математическое моделирование диспетчеров задач в многопроцессорных вычислительных системах на основе стохастических сетей массового обслуживания: автореф. дис. ... канд. техн. наук. Пенза, 2013. 23 с.
11. Ильяшенко А.С. Модели и метод исследования приоритетных систем массового обслуживания с вероятностным выталкивающим механизмом: автореф. дис. ... канд. физ.-мат. наук. Санкт-Петербург, 2014. 18 с.
12. Мартышкин А.И., Воронцов А.А., Валова О.О. Математическое моделирование диспетчеров задач с пространственным разделением с неоднородным потоком задач на обслуживание и ограниченной длиной очереди // XXI век: итоги прошлого и проблемы настоящего плюс. 2015. Т. 1. № 3 (25). С. 142–149.
13. Мартенс-Атюшев Д.С. Анализ задержек при проектировании специализированных многопроцессорных систем с применением теории массового обслуживания // Прикладная математика и информатика: современные исследования в области естественных и технических наук. Материалы VI Международной научно-практической конференции (школы-семинара) молодых ученых. 2020. С. 296–300.