

**Министерство науки и высшего образования Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**  
**УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**  
**МЕХАНИКИ И ОПТИКИ»**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**РАЗРАБОТКА И РЕАЛИЗАЦИЯ МЕТОДОВ ЭФФЕКТИВНОГО**  
**ВЗАИМОДЕЙСТВИЯ ПРОЦЕССОВ В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ**

Автор: Губарев Владимир Юрьевич \_\_\_\_\_

Направление подготовки: 09.04.04 Программная  
инженерия

Квалификация: Магистр

Руководитель: Косяков М.С., к.т.н. \_\_\_\_\_

**К защите допустить**

Руководитель ОП Парфенов В.Г., проф., д.т.н. \_\_\_\_\_

« \_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

Санкт-Петербург, 2020 г.

Студент Губарев В.Ю.

Группа Р42111 Факультет ИТиП

Направленность (профиль), специализация  
Информационно-вычислительные системы

ВКР принята « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

Оригинальность ВКР \_\_\_\_ %

ВКР выполнена с оценкой \_\_\_\_\_

Дата защиты « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

Секретарь ГЭК Болдырева Е.А. \_\_\_\_\_

Листов хранения \_\_\_\_\_

Демонстрационных материалов/Чертежей хранения \_\_\_\_\_

**Министерство науки и высшего образования Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**  
**УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**  
**МЕХАНИКИ И ОПТИКИ»**

**УТВЕРЖДАЮ**

Руководитель ОП  
проф., д.т.н. Парфенов В.Г. \_\_\_\_\_  
« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

**ЗАДАНИЕ**  
**НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**

Студент Губарев В.Ю.

Группа Р42111 Факультет ИТиП

Руководитель Косяков М.С., к.т.н., доцент ФПИиКТ

**1 Наименование темы:** Разработка и реализация методов эффективного взаимодействия процессов в распределенных системах

**Направление подготовки (специальность):** 09.04.04 Программная инженерия

**Направленность (профиль):** Информационно-вычислительные системы

**Квалификация:** Магистр

**2 Срок сдачи студентом законченной работы:** « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

**3 Техническое задание и исходные данные к работе**

Требуется разработать стилевой файл для системы L<sup>A</sup>T<sub>E</sub>X, позволяющий оформлять бакалаврские работы и магистерские диссертации на кафедре компьютерных технологий Университета ИТМО. Стилиевой файл должен генерировать титульную страницу пояснительной записки, задание, аннотацию и содержательную часть пояснительной записки. Первые три документа должны максимально близко соответствовать шаблонам документов, принятым в настоящий момент на кафедре, в то время как содержательная часть должна максимально близко соответствовать ГОСТ 7.0.11-2011 на диссертацию.

**4 Содержание выпускной работы (перечень подлежащих разработке вопросов)**

Пояснительная записка должна демонстрировать использование наиболее типичных конструкций, возникающих при составлении пояснительной записки (перечисления, рисунки, таблицы, листинги, псевдокод), при этом должна быть составлена так, что демонстрируется корректность работы стилевого файла. В частности, записка должна содержать не менее двух приложений (для демонстрации нумерации рисунков и таблиц по приложениям согласно ГОСТ) и не менее десяти элементов нумерованного перечисления первого уровня вложенности (для демонстрации корректности используемого при нумерации набора русских букв).

**5 Перечень графического материала (с указанием обязательного материала)**

Графические материалы и чертежи работой не предусмотрены

## 6 Исходные материалы и пособия

- а) ГОСТ 7.0.11-2011 «Диссертация и автореферат диссертации»;
- б) С.М. Львовский. Набор и верстка в системе  $\text{\LaTeX}$ ;
- в) предыдущий комплект стилевых файлов, использовавшийся на кафедре компьютерных технологий.

7 Дата выдачи задания « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

Руководитель ВКР \_\_\_\_\_

Задание принял к исполнению \_\_\_\_\_

« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

**Министерство науки и высшего образования Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**  
**УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**  
**МЕХАНИКИ И ОПТИКИ»**

**АННОТАЦИЯ**  
**ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

**Студент:** Губарев Владимир Юрьевич

**Наименование темы ВКР:** Разработка и реализация методов эффективного взаимодействия процессов в распределенных системах

**Наименование организации, в которой выполнена ВКР:** Университет ИТМО

**ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

1 Цель исследования: Разработка удобного стилевого файла  $\text{\LaTeX}$  для бакалавров и магистров кафедры компьютерных технологий.

2 Задачи, решаемые в ВКР:

- а) обеспечение соответствия титульной страницы, задания и аннотации шаблонам, принятым в настоящее время на кафедре;
- б) обеспечение соответствия содержательной части пояснительной записки требованиям ГОСТ 7.0.11-2011 «Диссертация и автореферат диссертации»;
- в) обеспечение относительного удобства в использовании — указание данных об авторе и научном руководителе один раз и в одном месте, автоматический подсчет числа тех или иных источников.

3 Число источников, использованных при составлении обзора: 0

4 Полное число источников, использованных в работе: 7

5 В том числе источников по годам:

Отечественных			Иностранных		
Последние 5 лет	От 5 до 10 лет	Более 10 лет	Последние 5 лет	От 5 до 10 лет	Более 10 лет
2	3	2	0	0	0

6 Использование информационных ресурсов Internet: да, число ресурсов: 4

7 Использование современных пакетов компьютерных программ и технологий:

Пакеты компьютерных программ и технологий	Раздел работы
Пакет <code>tabularx</code> для чуть более продвинутых таблиц	??, Приложения А, Б
Пакет <code>biblatex</code> и программное средство <code>biber</code>	Список использованных источников

8 Краткая характеристика полученных результатов

Получился, надо сказать, практически неплохой стилевик. В 2015–2018 годах его уже использовали некоторые бакалавры и магистры. Надеюсь на продолжение.

## 9 Гранты, полученные при выполнении работы

Автор разрабатывал этот стилевик исключительно за свой счет и на добровольных началах. Однако значительная его часть была бы невозможна, если бы автор не написал в свое время кандидатскую диссертацию в L<sup>A</sup>T<sub>E</sub>X, а также не отвечал за формирование кучи научно-технических отчетов по гранту, известному как «5-в-100», что происходило при государственной финансовой поддержке ведущих университетов Российской Федерации (субсидия 074-U01).

## 10 Наличие публикаций и выступлений на конференциях по теме работы

По теме этой работы я (к счастью!) ничего не публиковал. Однако покажу, как можно ссылаться на свои публикации из списка литературы:

Студент            Губарев В.Ю. \_\_\_\_\_

Руководитель    Косяков М.С.        \_\_\_\_\_

« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	6
ГЛАВА 1. Обзор методов межпроцессного взаимодействия .....	9
Выводы по главе 1 .....	9
ГЛАВА 2. Анализ методов межпроцессного взаимодействия .....	10
Выводы по главе 2 .....	10
ГЛАВА 3. Разработка и реализация методов эффективного взаимодействия процессов .....	11
Выводы по главе 3 .....	11
ГЛАВА 4. Проведение эксперимента и обработка результатов .....	12
4.1. Постановка эксперимента .....	12
4.1.1. Конфигурация экспериментального стенда .....	12
4.1.2. Конфигурация экспериментальной системы .....	12
4.1.3. Характер экспериментальной нагрузки .....	13
4.1.4. Время обслуживания заявок в процессах .....	13
4.2. Результаты экспериментов .....	15
4.2.1. Использование ТСР для передачи данных .....	15
4.2.2. Использование разделяемой памяти для передачи данных .....	16
Выводы по главе 4 .....	19
ЗАКЛЮЧЕНИЕ .....	21
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	22
ПРИЛОЖЕНИЕ А. Пример приложения .....	23
ПРИЛОЖЕНИЕ Б. Еще один пример приложения с невероятно длинным названием для тестирования переносов .....	24
ПРИЛОЖЕНИЕ В. Пример огромного листинга .....	25

## ВВЕДЕНИЕ

**Объектом исследования** являются методы межпроцессного взаимодействия.

**Предметом исследования** является временная задержка на передачу данных между процессами распределенной системы в пределах одного физического узла.

**Цель работы** – уменьшение временной задержки на передачу данных между процессами в пределах одного физического узла путем разработки и применения методов эффективного межпроцессного взаимодействия.

В рамках настоящей работы поставлены следующие **задачи**:

- рассмотреть существующие методы межпроцессного взаимодействия, доступные при взаимодействии процессов, находящихся на одном физическом узле;
- произвести анализ и отбор методов межпроцессного взаимодействия для реализации новых методов межпроцессного взаимодействия;
- разработать и реализовать эффективные методы межпроцессного взаимодействия;
- экспериментально оценить полученные методы межпроцессного взаимодействия.

**Актуальность исследования.**

Для некоторых систем эффективное межпроцессное взаимодействие является критически важной частью их работы. Требование по минимизации времени обслуживания заявок может напрямую следовать из области применения системы, как в случае с системами для алгоритмической торговли на финансовых рынках. Обслуживание заявок множеством логически связанных процессов может быть существенно ускорено при размещении таких процессов на одном физическом узле. Современные процессоры с количеством с десятками вычислительных ядер могут обеспечить такую конфигурацию нужными ресурсами. Это позволяет использовать более эффективные методы межпроцессного взаимодействия, а именно методы на основе разделяемой памяти [1]. Эффективные методы межпроцессного взаимодействия могут использоваться для связи виртуальных машин или контейнеров в пределах машины-хозяина [5, 6]. Для связи программных модулей, исполняющихся в разных процессах для обеспечения отказоустойчивости



за счет изоляции процессов на уровне ОС. Для высокопроизводительных вычислений, таких как анализ научных данных или прогнозирование погоды.

При разработке сложной многокомпонентной распределенной системы программисту необходимо сосредоточиться на логике и корректности работы самой системы. В то время как методы межпроцессного взаимодействия должны быть для него прозрачны. Этого можно достичь, используя единый унифицированный интерфейс для межпроцессного взаимодействия. Это упрощает разработку, снижает необходимость сложного управления ресурсами для межпроцессного взаимодействия. А также позволяет автоматически использовать наиболее подходящие методы межпроцессного взаимодействия для данных *пространственных конфигураций* (**TBD: может, убрать?**) процессов, что может повысить эффективность выполнения некоторых задач этой системой.

Таким образом, разработка и реализация эффективных методов межпроцессного взаимодействия и интерфейса для автоматического доступа к наиболее подходящим из них необходима и обоснована. Посредством этого интерфейса программист прозрачно для себя использует методы межпроцессного взаимодействия на основе разделяемой памяти при взаимодействии с локальными процессами без необходимости перекомпиляции программы. Но поскольку зачастую нельзя разместить всю систему на одном, даже очень производительном, сервере используется TCP при взаимодействии с процессами на других физических узлах.

**Методы исследования** включают в себя анализ существующих методов межпроцессного взаимодействия, экспериментальную оценку разработанных методов эффективного межпроцессного взаимодействия и методы математической статистики для обработки экспериментальных данных.

#### **Средства исследования:**

- язык программирования C++, компилятор *Clang 6.0.1*, стандартная библиотека C++ *libstdc++*;
- Библиотека Boost.Interprocess [2] для управления разделяемой памятью;
- система трассировки событий [7] на основе инструмента LTTng [4].

**Научная новизна** заключается в предложенных новых методах эффективного межпроцессного взаимодействия в пределах одного физического узла, которые не описаны в существующих исследованиях.

#### **Положения, выносимые на защиту**

**TBD.**

- а) shmem + mux + futex + lf;
- б) shmem + mux + futex + hsha;
- в) shmem + mux + busy wait.

### **Апробация результатов.**

Основные результаты работы были представлены на IX Конгрессе Молодых Ученых.

Предложенные методы межпроцессного взаимодействия на основе разделяемой памяти подготовлены к использованию в платформе для торговли на финансовых рынках Tbricks от компании Itiviti.

## ГЛАВА 1. ОБЗОР МЕТОДОВ МЕЖПРОЦЕССНОГО ВЗАИМОДЕЙСТВИЯ

Пример ссылок в рамках обзора: [**example-english**, **example-russian**, **unrestricted-jump-evco**, **nsga-ii-steady-state**]. Вне обзора: [**bellman**].

### Выводы по главе 1

В конце каждой главы желательно делать выводы. Вывод по данной главе — нумерация работает корректно, ура!

## **ГЛАВА 2. АНАЛИЗ МЕТОДОВ МЕЖПРОЦЕССНОГО ВЗАИМОДЕЙСТВИЯ**

### **Выводы по главе 2**

В конце каждой главы желательно делать выводы. Вывод по данной главе — нумерация работает корректно, ура!

## ГЛАВА 3. РАЗРАБОТКА И РЕАЛИЗАЦИЯ МЕТОДОВ ЭФФЕКТИВНОГО ВЗАИМОДЕЙСТВИЯ ПРОЦЕССОВ

Листинг 1 должен иметь номер 4.

Листинг 1 – Исходный код и флюид `algorithm`

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, world!");
    }
}
```

Рисунок 1 должен иметь номер 2.

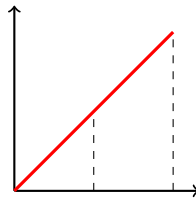


Рисунок 1 – Пример рисунка

Таблица 1 должна иметь номер 3.

Таблица 1 – Таблица умножения с помощью `tabularx` (фрагмент)

–	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
3	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51
4	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68

### Выводы по главе 3

В конце каждой главы желательно делать выводы. Вывод по данной главе — нумерация работает корректно, ура!

## **ГЛАВА 4. ПРОВЕДЕНИЕ ЭКСПЕРИМЕНТА И ОБРАБОТКА РЕЗУЛЬТАТОВ**

### **4.1. Постановка эксперимента**

#### **4.1.1. Конфигурация экспериментального стенда**

##### **4.1.1.1. Аппаратное обеспечение**

Процессор Intel(R) Xeon(R) Platinum 8168 CPU 2.7 GHz. Технология Hyper-Threading отключена для уменьшения влияния на время обработки заявок в системе [3].

Оперативная память: DDR4-2666 128 GiB.

##### **4.1.1.2. Программное обеспечение**

Операционная система Red Hat Enterprise Linux Server release 7.8 (Maipo).  
Ядро Linux 3.10.0-1127.el7.x86\_64

Компилятор C++ Clang 6.0.1.

Стандартная библиотека C++: libstdc++ 8.1.0.

Библиотека Boost.Interprocess 1.68.0.

#### **4.1.2. Конфигурация экспериментальной системы**

Система для проведения эксперимента состоит из двух процессов:

- Процесс-шлюз отвечает за преобразование заявок из формата внешнего мира во внутренний формат системы и обратно.
- Процесс-обработчик совершает некоторые преобразования над заявкой и отправляет результат за пределы системы через процесс-шлюз.

Процессы выполняются на двух процессорах, расположенных в разных разъемах на материнской плате физического узла.

Снаружи системы находится симулятор внешнего мира. Он генерирует поток заявок в систему и получает результат обработки заявки в системе. Схема взаимодействия процессов в эксперименте представлена на Рисунке 2.

В настоящей работе измеряется временная задержка на передачу данных между процессами внутри системы, а именно из процесса-шлюза в процесс-обработчик и обратно (сообщения типа №1 и №2 в запросе и ответе между процессом-шлюзом и процессом обработчиком на Рисунке 2).

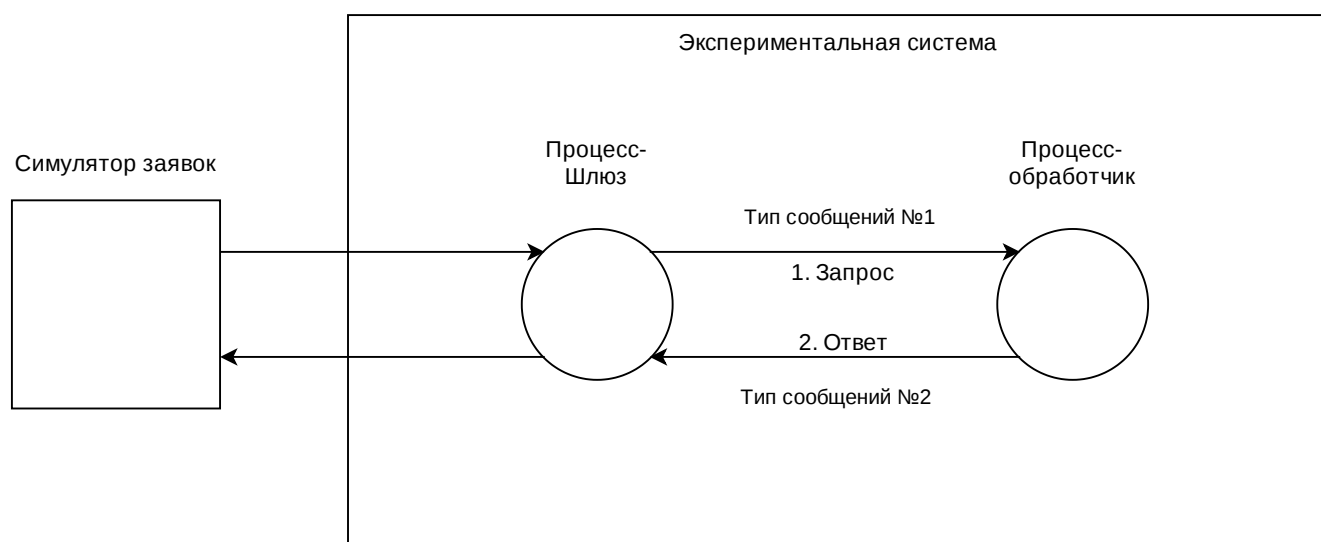


Рисунок 2 – Схема взаимодействия процессов в эксперименте

#### 4.1.3. Характер экспериментальной нагрузки

**TBD:** может, стоит пояснить, почему вместо 5мс получился такой разброс между сериями? Симулятор заявок каждые  $10 \pm 4$  мс создает и отправляет 4 заявки с интервалом  $50 \pm 26$  мкс между ними. Интервалы указаны с 95% доверительной вероятностью.

**TBD:** Подозрительный текст. Может, убрать пункт про варьирование размера сообщений типа №1 Размер сообщений типа №1 (см. Рисунок 2) выбирается в зависимости от приходящих в процесс-шлюз заявок с равной вероятностью между 85, 96 и 104 байтами. Размер сообщений типа №2 равен 530 байтам.

**TBD:** тоже стоит убрать. На моих гистограммах этого не видно. Нужны гораздо более искусственные замеры, чтобы обнаружить разницу. Различие в размерах сообщений, передаваемых между процессами, позволяет количественно оценить влияние непосредственно размера сообщения на временную задержку на его передачу.

#### 4.1.4. Время обслуживания заявок в процессах

**TBD:** разобраться с обработкой соединений и обслуживанием заявок. Должно быть одно слово для обозначения одной сущности.

На Рисунке 3 представлена гистограмма времени обслуживания заявок в процессе-обработчике. Время обслуживания заявок с 95% доверительной вероятностью укладывается в диапазон  $12 \pm 6$  мкс.

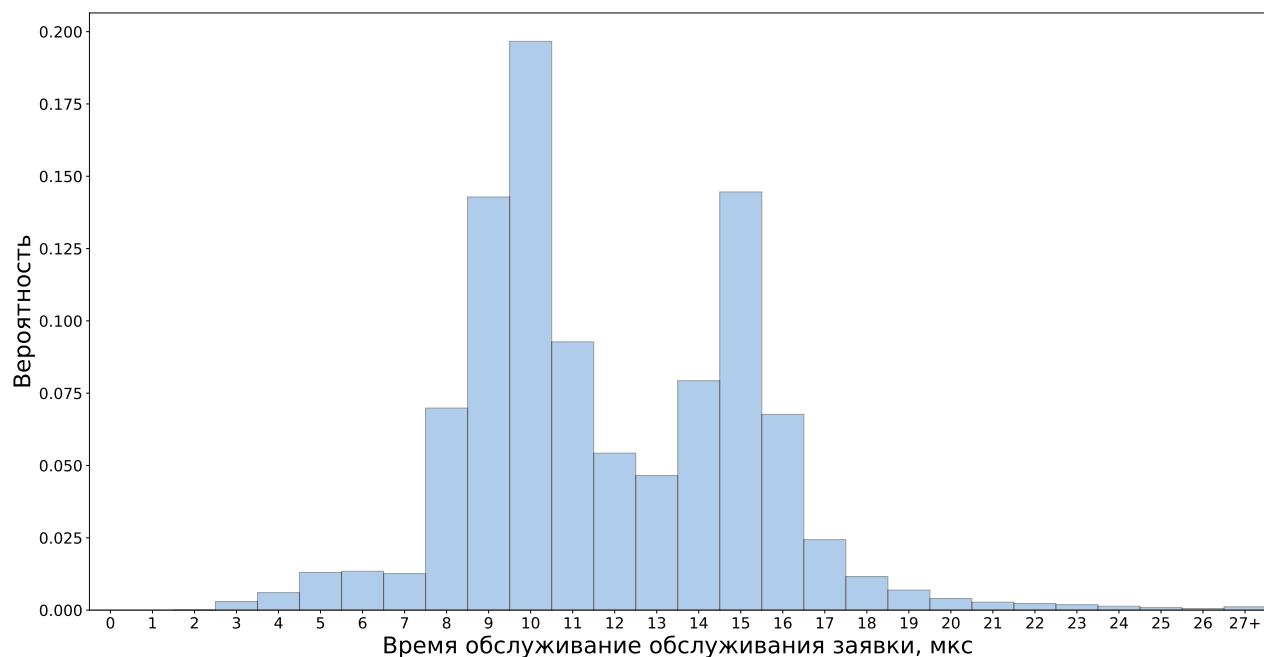


Рисунок 3 – Гистограмма времени обслуживания заявки в процессе-обработчике

На Рисунке 4 представлена гистограмма времени обслуживания заявок в процессе-шлюзе. Время обслуживания заявок с 95% доверительной вероятностью укладывается в диапазон  $18.5 \pm 5.5$  мкс.

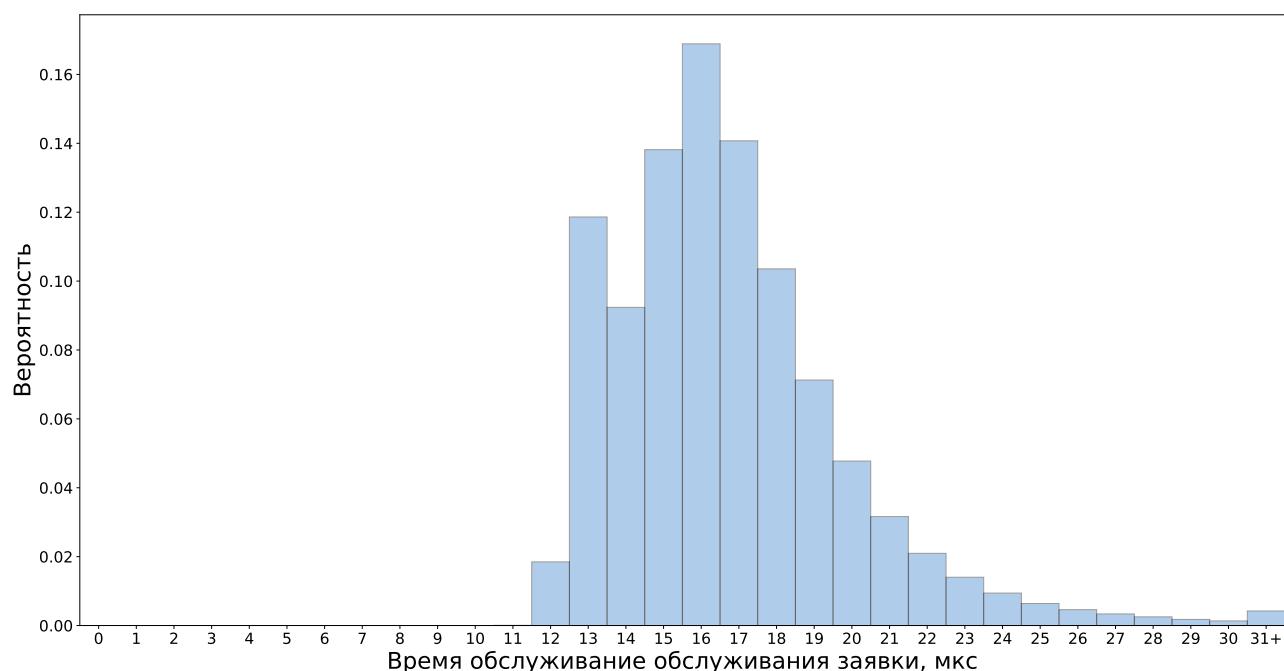


Рисунок 4 – Гистограмма времени обслуживания заявки в процессе-шлюзе



## 4.2. Результаты экспериментов

### 4.2.1. Использование ТСП для передачи данных

В качестве точки отсчета в настоящей работе выступает метод межпроцессного взаимодействия на основе ТСП, используемый посредством сокетов **TBD: ссылка на определение и на раздел диссертации ??**. Гистограмма временной задержки на передачу данных для данного метода приведена на Рисунке 5.

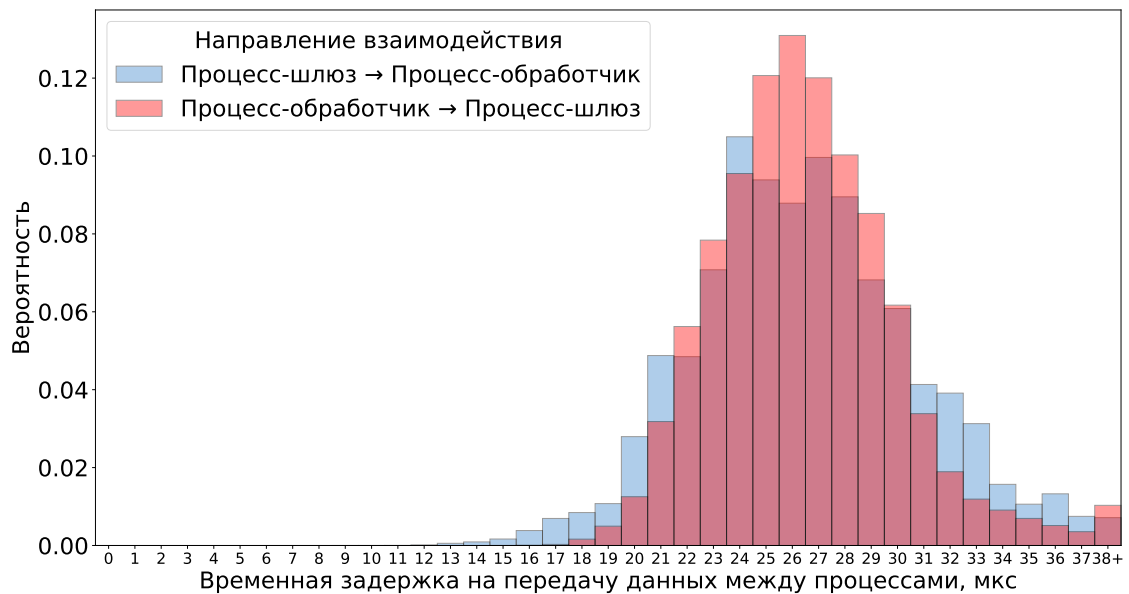


Рисунок 5 – Гистограмма временной задержки на передачу данных между процессами при использовании ТСП

В Таблице 2 приведены основные временные характеристики данного метода. Межпроцессное взаимодействие в обе стороны имеет похожие значения.

Таблица 2 – Основные показатели временной задержки на передачу данных для метода на основе ТСП

Направление взаимодействия	min(t), мкс	M(t) ± 95%, мкс	max(t), мс	δ между сериями, мс	δ между заявками, мкс
Процесс-шлюз → Процесс-обработчик	9	27.5 ± 8.5	2.1	10 ± 4	50 ± 26
Процесс-обработчик → Процесс-шлюз	13	28 ± 7	9.2	10 ± 4	87 ± 32

## 4.2.2. Использование разделяемой памяти для передачи данных

### 4.2.2.1. Использование ТСР для оповещения о появлении данных

В данном подразделе приведены данные об экспериментах с методом меж-процессного взаимодействия, описанным в Разделе ??.

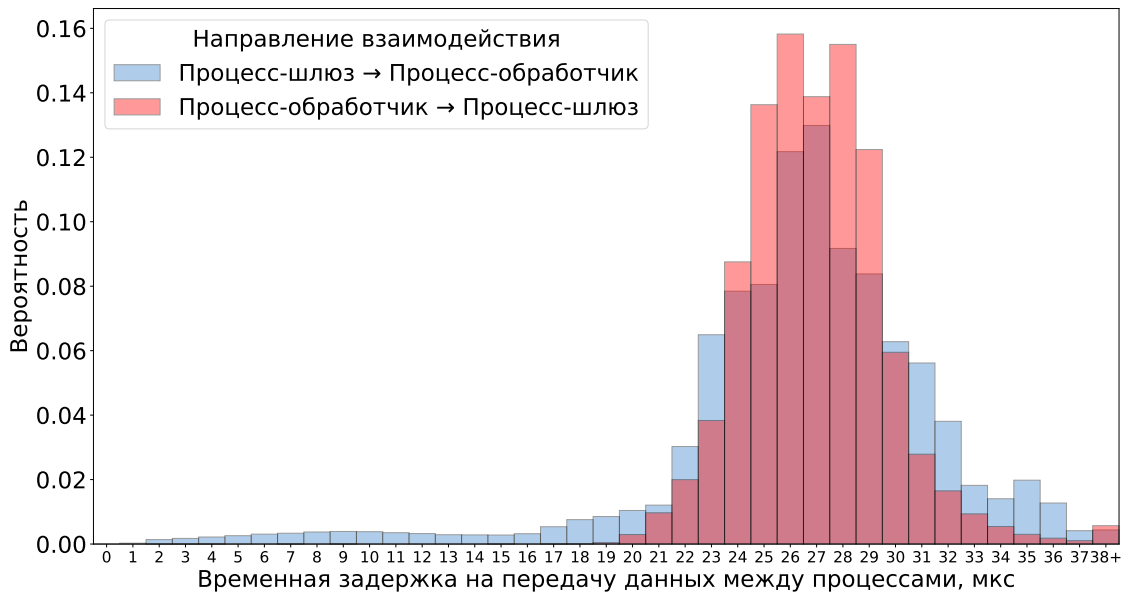


Рисунок 6 – Гистограмма временной задержки на передачу данных между процессами при использовании разделяемой памяти для передачи данных и ТСР для оповещения о появлении данных в ней

В Таблице 3 приведены основные временные характеристики данного метода. Межпроцессное взаимодействие в сторону процесса-обработчика работает быстрее, т.к. в среднем время обслуживания заявок в процессе-обработчике заметно меньше (см. Рисунок 3), чем скорость поступления новых заявок, что позволяет использовать оптимизацию, описанную в Разделе ?? TBD: сослаться на оптимизацию про отсутствие необходимости отправки новых сигналов, когда старые еще не были обработаны..

### 4.2.2.2. Использование мультиплексора в разделяемой памяти для оповещения о появлении данных

В данном подразделе приведены данные об экспериментах с методом меж-процессного взаимодействия, описанным в Разделе ??.

Таблица 3 – Основные показатели временной задержки на передачу данных для метода на основе TCP

Направление взаимодействия	min(t), мкс	M(t) ± 95%, мкс	max(t), мс	δ между сериями, мс	δ между заявками, мкс
Процесс-шлюз → Процесс-обработчик	1	22.5 ± 12.5	3	10 ± 4	50 ± 27
Процесс-обработчик → Процесс-шлюз	2	27.5 ± 5.5	9.8	10 ± 4	87 ± 32

**Блокирующие методы** В блокирующих методах поток мультиплексора событий использует примитив *futex* **Может, сослаться на определение futex?** для пассивного ожидания новых сигналов (см. Раздел ??).

**Диспетчеризация и обработка соединений по модели "Полусинхронный/Полуреактивный"** В данном подразделе приведены данные об экспериментах с методом межпроцессного взаимодействия, описанным в Разделе ??.

В Таблице 4 приведены основные временные характеристики данного метода. **TBD**

На Рисунке 7 приведена плотность вероятности временной задержки на передачу данных для данного метода.

Таблица 4 – Основные показатели временной задержки на передачу данных между процессами для метода, использующего разделяемую память для передачи данных, блокирующий мультиплексор в разделяемой памяти и модель "Полусинхронный/Полуреактивный" при обслуживании заявок

Направление взаимодействия	min(t), мкс	M(t) ± 95%, мкс	max(t), мс	δ между сериями, мс	δ между заявками, мкс
Процесс-шлюз → Процесс-обработчик	1	12.5 ± 5.5	6.9	10 ± 4	51 ± 28
Процесс-обработчик → Процесс-шлюз	3	11.5 ± 2.5	11.6	10 ± 4	91 ± 36

**Диспетчеризация и обработка соединений по модели "Лидер/Последователи"** В данном подразделе приведены данные об экс-

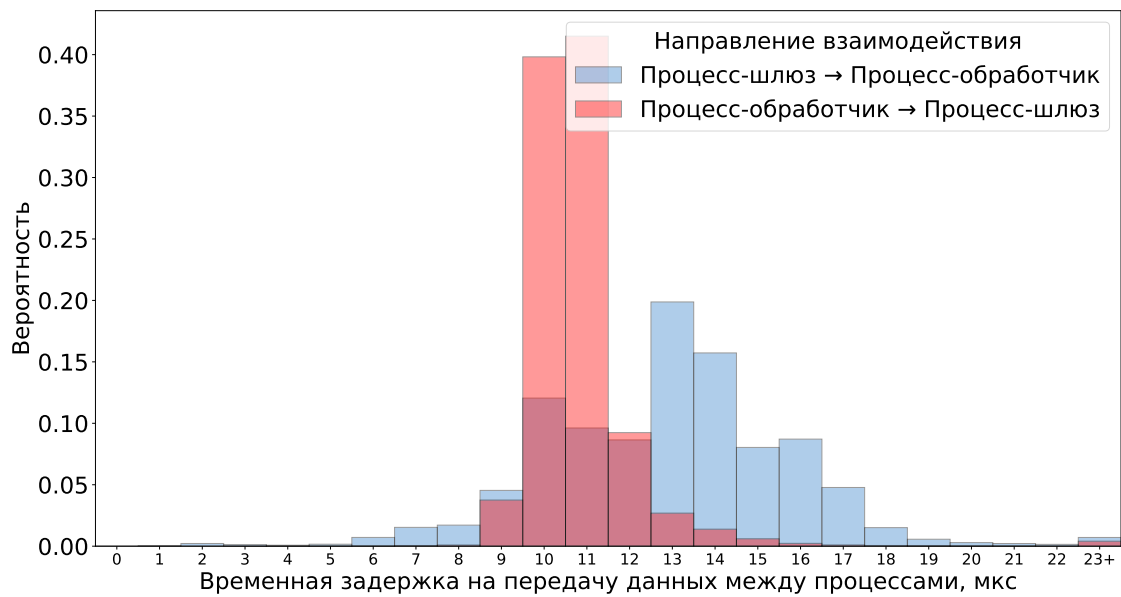


Рисунок 7 – Гистограмма временной задержки на передачу данных между процессами для метода, использующего разделяемую память для передачи данных, блокирующий мультиплексор в разделяемой памяти и модель "Полусинхронный/Полуреактивный" при обслуживании заявок

периментах с методом межпроцессного взаимодействия, описанным в Разделе ??.

В Таблице 5 приведены основные временные характеристики данного метода. **TBD**

На Рисунке 8 приведена плотность вероятности временной задержки на передачу данных для данного метода.

Таблица 5 – Основные показатели временной задержки на передачу данных между процессами для метода, использующего разделяемую память для передачи данных, неблокирующий мультиплексор в разделяемой памяти и модель "Лидер/Последователи" при обслуживании заявок

Направление взаимодействия	min(t), мкс	M(t) ± 95%, мкс	max(t), мс	δ между сериями, мс	δ между заявками, мкс
Процесс-шлюз → Процесс-обработчик	1	11.5 ± 6.5	2.4	8.4 ± 5.3	52 ± 29
Процесс-обработчик → Процесс-шлюз	2	9.5 ± 1.5	9.5	8.4 ± 5.3	90 ± 35

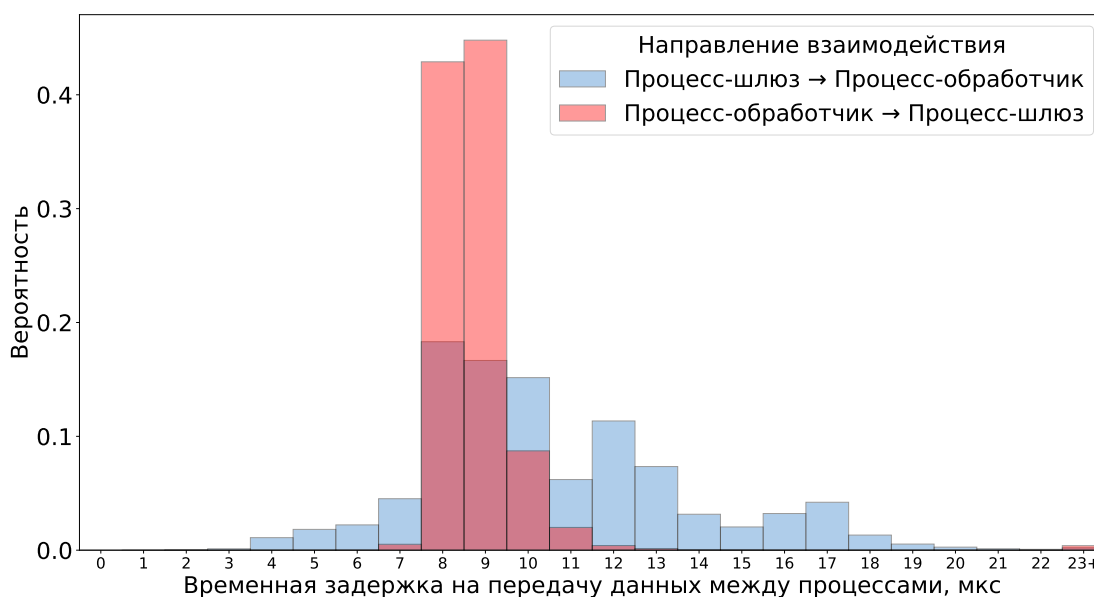


Рисунок 8 – Гистограмма временной задержки на передачу данных между процессами для метода, использующего разделяемую память для передачи данных, неблокирующий мультиплексор в разделяемой памяти и модель ”Лидер/Последователи“ при обслуживании заявок

**Неблокирующий метод** В неблокирующем методе поток мультиплексора событий метод активного ожидания новых сигналов (см. Раздел ??). В данном параграфе рассматривается исключительно модель обслуживания заявок ”Лидер/Последователи“ т.к. в параграфе выше она показала лучшие результаты по сравнению с моделью ”Полусинхронный/Полуреактивный“.

**Диспетчеризация и обработка соединений по модели ”Лидер/Последователи”** В данном подразделе приведены данные об экспериментах с методом межпроцессного взаимодействия, описанным в Разделе ??.

В Таблице 6 приведены основные временные характеристики данного метода. **TBD**

На Рисунке 9 приведена плотность вероятности временной задержки на передачу данных для данного метода.

## Выводы по главе 4

Вывод по очередной главе

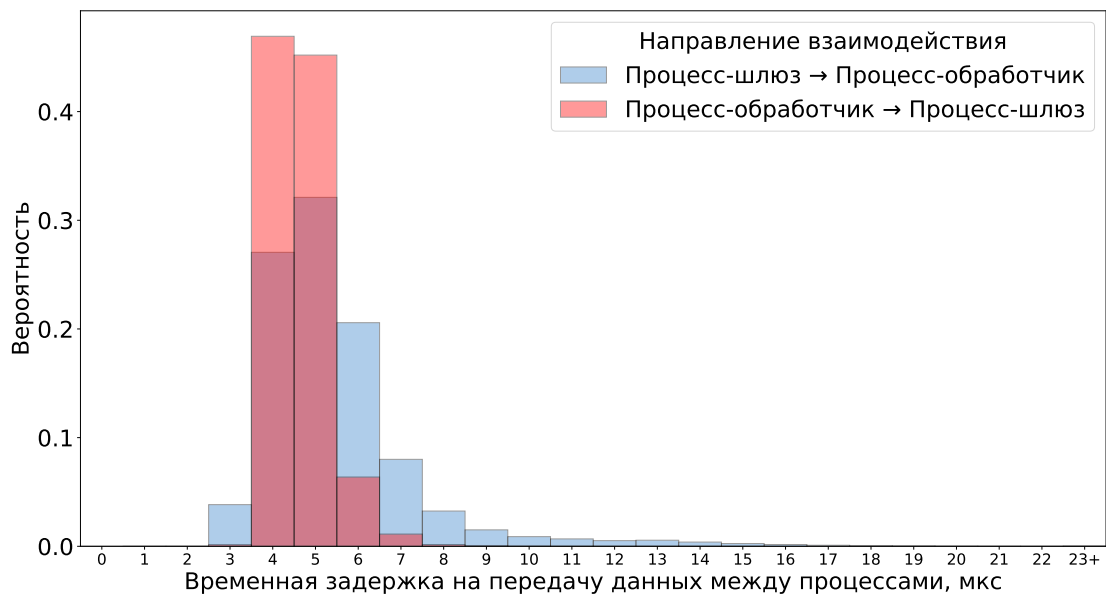


Рисунок 9 – Гистограмма временной задержки на передачу данных между процессами для метода, использующего разделяемую память для передачи данных, блокирующий мультиплексор в разделяемой памяти и модель ”Лидер/Последователи“ при обслуживании заявок

Таблица 6 – Основные показатели временной задержки на передачу данных между процессами для метода, использующего разделяемую память для передачи данных, блокирующий мультиплексор в разделяемой памяти и модель ”Лидер/Последователи“ при обслуживании заявок

Направление взаимодействия	min(t), мкс	M(t) ± 95%, мкс	max(t), мс	δ между сериями, мс	δ между заявками, мкс
Процесс-шлюз → Процесс-обработчик	1	7 ± 4	8.5	7.4 ± 5.8	58 ± 35
Процесс-обработчик → Процесс-шлюз	3	5 ± 1	0.03	7.4 ± 5.8	167 ± 113

## **ЗАКЛЮЧЕНИЕ**

В данном разделе размещается заключение.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Draft: Have you checked your IPC performance lately? [Электронный ресурс] / S. Smith [и др.]. — 2012. — URL: <https://www.semanticscholar.org/paper/Draft-%3A-Have-you-checked-your-IPC-performance-Smith-Madhavapeddy/1cb3b82e2ef6a95b576573b8af0f3ec6f7bc21d9> (дата обр. 25.03.2020).
- 2 *Gaztanaga I.* Chapter 18. Boost.Interprocess [Электронный ресурс]. — URL: [https://www.boost.org/doc/libs/1\\_63\\_0/doc/html/interprocess.html](https://www.boost.org/doc/libs/1_63_0/doc/html/interprocess.html) (дата обр. 25.03.2020).
- 3 *Khartchenko E.* Optimizing Computer Applications for Latency: Part 1: Configuring the Hardware [Электронный ресурс]. — 25.07.2017. — URL: <https://software.intel.com/en-us/articles/optimizing-computer-applications-for-latency-part-1-configuring-the-hardware> (дата обр. 01.03.2020).
- 4 LTTng: an open source tracing framework for Linux [Электронный ресурс]. — URL: <https://lttng.org/> (дата обр. 25.03.2020).
- 5 *Macdonell A. C.* Shared-Memory Optimizations for Virtual Machines : PhD dissertation / Macdonell A. Cameron. — Department of Computing Science : University of Alberta, 2011. — DOI: <https://doi.org/10.7939/R3Q715>.
- 6 *Xiaodi K.* Interprocess Communication Mechanisms With Inter-Virtual Machine Shared Memory : Master's dissertation / Xiaodi Ke. — Department of Computing Science : University of Alberta, 2011. — DOI: <https://doi.org/10.7939/R3PH6H>.
- 7 *Кузичкина А.* Исследование и разработка методов трассировки событий в параллельных и распределенных системах : Пояснительная записка выпускной квалификационной работы / Кузичкина Анастасия. — Факультет ПИиКТ : Университет ИТМО, 06.2017.



## **ПРИЛОЖЕНИЕ А. ПРИМЕР ПРИЛОЖЕНИЯ**

**ПРИЛОЖЕНИЕ Б. ЕЩЕ ОДИН ПРИМЕР ПРИЛОЖЕНИЯ С  
НЕИМОВЕРНО ДЛИННЮЩИМ НАЗВАНИЕМ ДЛЯ ТЕСТИРОВАНИЯ  
ПЕРЕНОСОВ**

**ПРИЛОЖЕНИЕ В. ПРИМЕР ОГРОМНОГО ЛИСТИНГА**