

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Астраханский государственный университет имени В. Н. Татищева»
(Астраханский государственный университет им. В. Н. Татищева)

СОГЛАСОВАНО
Руководитель ОПОП

_____ М.В. Коломина

«8» сентября 2022 г.

УТВЕРЖДАЮ
Зав. кафедрой ПМИ

_____ М.В. Коломина

«8» сентября 2022 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Параллельное программирование»

Составители

**Елизаров Р.А., ассистент ФИТиП, ИТМО
Корнеев Г.А., к.т.н., доцент ФИТиП, ИТМО
Гордеев И.И., к.ф.-м.н., доцент каф. ПМИ, АГУ**

Направление подготовки /
специальность

**01.03.02 Прикладная математика и
информатика**

Направленность (профиль) ОПОП

**Программирование и искусственный
интеллект**

Квалификация (степень)

бакалавр

Форма обучения

очная

Год приёма

2023

Курс

3

Семестр(ы)

5

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

1.1. Целями освоения дисциплины «Параллельное программирование» является формирование у студентов теоретических знаний и практических навыков программирования параллельных и распределенных систем.

1.2. Задачи освоения дисциплины:

- изучение языков программирования и алгоритмов и структур данных, методов работы с данными, систем контроля версий.
- изучение поисковых структур данных и их особенностей.
- приобретение практических навыков проектирования и реализации параллельных и распределенных программных систем,
- приобретение практических навыков проектирования, разработки и тестирования прикладного и системного программного обеспечения, проектирования баз данных

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

2.1. Учебная дисциплина «Параллельное программирование» относится к части, формируемой участниками образовательных отношений и осваивается в 5 семестре.

2.2. Для изучения данной учебной дисциплины (модуля) необходимы следующие знания, умения, навыки, формируемые предшествующими учебными дисциплинами (модулями):

Архитектура компьютера

Язык программирования C++

Технологии программирования

Знания: базовые конструкции построения алгоритмов, представление целочисленных и вещественных данных на компьютере.

Умения: использовать основные конструкции языков программирования, выбирать типы данных, соответствующие решаемой задаче.

Навыки: программирования на C++, чтения и записи двоичных кодов в шестнадцатеричном виде.

2.3. Последующие учебные дисциплины и (или) практики, для которых необходимы знания, умения, навыки, формируемые данной учебной дисциплиной:

- Проектирование программного обучения,
- Написание выпускной квалификационной работы,
- Производственная практика.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс освоения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ФГОС ВО и ОПОП ВО по данному направлению подготовки / специальности:

а) профессиональных (ПК):

- ПК-1. Способен создавать, отлаживать и оформлять программный код
- ПК-3. Способен обеспечивать заданный уровень производительности, надежности и безопасности при создании вариантов архитектуры программного средства
 - ПК-7. Способен создавать и оценивать варианты архитектуры программного средства и осуществлять выбор среди них
 - ПК-10. Разработка и реализация архитектуры программного обеспечения
 - ПК-12. Разработка компонентов системных программных продуктов и интеграция разработанного программного обеспечения
 - ПК-13. Способность разрабатывать, анализировать, реализовывать и внедрять ал-

горитмы и структуры данных в рамках разработки системного и прикладного программного обеспечения

- ПК-15. Способностью проектировать и реализовывать сложные программные системы и комплексы, в том числе высоконагруженные, распределенные и масштабируемые гетерогенные системы

Таблица 1 – Декомпозиция результатов обучения

Код и наименование компетенции	Планируемые результаты обучения по дисциплине (модулю)		
	Знать (1)	Уметь (2)	Владеть (3)
ПК-1. Способен создавать, отлаживать и оформлять программный код	ИПК-1.1.1. Языки программирования и алгоритмы и структур данных, методов работы с данными, систем контроля версий.	ИПК-1.2.1. Осуществлять формализацию и алгоритмизацию поставленных задач, создавать программный код с использованием языков программирования, работать с системой контроля версий, оформлять в соответствии с требованиями, проверять и отлаживать программный код.	ИПК-1.3.1. Способностью создания, отлаживания и оформления программного кода.
ПК-3. Способен обеспечивать заданный уровень производительности, надежности и безопасности при создании вариантов архитектуры программного средства	ИПК-3.1.1. Шаблоны проектирования ПО, используемые в промышленной разработке ПО, инструменты для профилирования разработанного кода под существующей нагрузкой сервиса.	ИПК-3.2.1. Читать код, соответствующего стилю кода в проекте.	ИПК-3.3.1. Навыками написания кода в рамках заданного в проекте стиля написания кода.
ПК-7. Способен создавать и оценивать варианты архитектуры программного средства и осуществлять выбор среди них	ИПК-7.1.1. Типы, слои и шаблоны проектирования программных компонентов, архитектуры программного средства.	ИПК-7.2.1. Определять качественные характеристики и осуществлять выбор типа и слоев программных компонентов.	ИПК-7.3.1. Навыками создания и оценивания вариантов архитектуры программного средства.
ПК-10. Разработка и реализация архитектуры программного обеспечения	ИПК-10.1.1. Поисковые структуры данных и их особенности.	ИПК-10.2.1. Оценивать характеристики поступающих данных для определения эффективной структуры данных.	ИПК-10.3.1. Балансировкой деревьев поиска.
ПК-12. Разработка компонентов системных программных продуктов и интеграция разработанного программного обеспечения	ИПК-12.1.1. Современные алгоритмы и структуры данных, основные приемы, используемые при проектировании алгоритмов и структур данных.	ИПК-12.2.1. Применять аппарат математической логики, теории типов и абстрактной алгебры для анализа программ и процессов в них.	ИПК-12.3.1. Навыками проектирования, разработки и тестирования прикладного и системного программного обеспечения, проектирования баз данных, создания информационных ресурсов в сети интернет.

ПК-13. Способность разрабатывать, анализировать, реализовывать и внедрять алгоритмы и структуры данных в рамках разработки системного и прикладного программного обеспечения	ИПК-13.1.1. Базовые структуры данных и операции над ними	ИПК-13.2.1. Решать типовые задачи дискретной математики, возникающие в учебной и профессиональной деятельности.	ИПК-13.3.1. Навыком проектирования алгоритма и его реализации средствами языка программирования.
ПК-15. Способностью проектировать и реализовывать сложные программные системы и комплексы, в том числе высоконагруженные, распределенные и масштабируемые гетерогенные системы	ИПК-15.1.1. Программные системы.	ИПК-15.2.1. Применять алгоритмы и структуры данных для эффективной реализации высоконагруженных систем.	ИПК-15.3.1. Навыками проектирования и реализации параллельных и распределенных программных систем.

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Объём дисциплины составляет 4 зачётных единицы, в том числе 68 часа, выделенных на контактную работу обучающихся с преподавателем (из них 34 часа – лекции, 34 часа – лабораторные работы) и 76 часов – на самостоятельную работу обучающихся.

Таблица 2 – Структура и содержание дисциплины

Раздел, тема дисциплины	Семестр	Контактная работа (в часах)			Самост. работа		Форма текущего контроля успеваемости, форма промежуточной аттестации [по семестрам]
		Л	ПЗ	ЛР	КР	СР	
Основы параллельного программирования	5	17		17		38	Лабораторная работа № 1, 2
Основные объекты в параллельном программировании	5	17		17		38	Лабораторная работа № 3, 4, 5
Итого		34		34		76	Экзамен

Таблица 3 – Матрица соотношения разделов, тем учебной дисциплины (модуля) и формируемых компетенций

Раздел, тема дисциплины (модуля)	Кол-во часов	Код компетенции							Общее количество компетенций
		ПК-1	ПК-3	ПК-7	ПК-10	ПК-12	ПК-13	ПК-15	
Основы параллельного программирования	72	+	+	+	+	+	+	+	7
Основные объекты в параллельном программировании	72	+	+	+	+	+	+	+	7
Итого	144								7

Краткое содержание каждой темы дисциплины (модуля)

№ раздела	Наименование раздела дисциплины	Содержание
1	Основы параллельного программирования	Алгоритмы взаимного исключения, Разделяемые регистры, Основные понятия параллельного программирования, Атомарный снимок состояния
2	Основные объекты в параллельном программировании	Модель памяти, Консенсус, Базовые структуры данных в многопоточности, Многопоточные (Thread-Safe) объекты, Применение JMM

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРЕПОДАВАНИЮ И ОСВОЕНИЮ ДИСЦИПЛИНЫ

5.1. Указания для преподавателей по организации и проведению учебных занятий по дисциплине (модулю)

Лекционные занятия

Основной формой реализации теоретического обучения является лекция, которая представляет собой систематическое, последовательное изложение преподавателем-лектором учебного материала теоретического характера. Цель лекции – организация целенаправленной познавательной деятельности студентов по овладению программным материалом учебной дисциплины.

Порядок подготовки лекционного занятия включает в себя выполнение следующих этапов:

- изучение требований программы дисциплины;
- определение целей и задач лекции;
- разработка плана проведения лекции;
- подбор литературы (ознакомление с методической литературой, публикациями периодической печати по теме лекционного занятия);
 - отбор необходимого и достаточного по содержанию учебного материала;
 - определение методов, приемов и средств поддержания интереса, внимания, стимулирования творческого мышления студентов;
- написание конспекта лекции.

Лекция должна включать следующие разделы:

- формулировку темы лекции;
- указание основных изучаемых разделов или вопросов и предполагаемых затрат времени на их изложение;
- изложение вводной части;
- изложение основной части лекции;
- краткие выводы по каждому из вопросов;
- заключение;
- рекомендации литературных источников по излагаемым вопросам.

Лабораторные занятия

Лабораторное занятие – целенаправленная форма организации педагогического процесса, направленная на углубление научно-теоретических знаний и овладение определенными методами работы, в процессе которых вырабатываются умения и навыки выполнения тех или иных учебных действий в данной сфере науки. Они развивают научное мышление и речь, позволяют проверить знания студентов и выступают как средства оперативной обратной связи.

Правильно организованные лабораторные занятия ориентированы на решение следующих задач:

- обобщение, систематизация, углубление, закрепление полученных на лекциях и в процессе самостоятельной работы теоретических знаний по дисциплине (предмету);
- формирование практических умений и навыков, необходимых в будущей профессиональной деятельности, реализация единства интеллектуальной и практической деятельности;
- выработка при решении поставленных задач таких профессионально значимых качеств, как самостоятельность, ответственность, точность, творческая инициатива.

Состав заданий для лабораторного занятия должен быть спланирован с расчетом, чтобы за отведенное время они могли быть качественно выполнены большинством учащихся.

Лабораторные занятия должны так быть организованы, чтобы студенты ощущали нарастание сложности выполнения заданий, испытывали бы положительные эмоции от переживания собственного успеха в учении, поисками правильных и точных решений.

Самостоятельная работа

Самостоятельная работа – это вид учебной деятельности, которую студент совершает в установленное время и в установленном объеме индивидуально или в группе, без непосредственной помощи преподавателя (но при его контроле), руководствуясь сформированными ранее представлениями о порядке и правильности выполнения действий.

В учебном процессе образовательного учреждения выделяются два вида самостоятельной работы:

- аудиторная – выполняется на учебных занятиях, под непосредственным руководством преподавателя и по его заданию (выполнение самостоятельных работ; выполнение контрольных и практических работ; решение задач);
- внеаудиторная – выполняется по заданию преподавателя, но без его непосредственного участия (подготовка к аудиторным занятиям; изучение учебного материала, вынесенного на самостоятельную проработку; выполнение домашних заданий разнообразного характера; выполнение индивидуальных заданий, направленных на развитие у студентов самостоятельности и инициативы; подготовка к контрольной работе). Внеаудиторные самостоятельные работы представляют собой логическое продолжение аудиторных занятий, проводятся по заданию преподавателя, который инструктирует студентов и устанавливает сроки выполнения задания.

5.2. Указания для обучающихся по освоению дисциплины (модулю)

Лекция

- Лекция – основной вид обучения в вузе.
- В лекции излагаются основные положения теории, ее понятия и законы, приводятся факты, показывающие связь теории с практикой.
- Накануне лекции необходимо повторить содержание предыдущей лекции (а также теорию по изучаемой теме в школьных учебниках геометрии, если эта тема была представлена в них), а затем посмотреть тему очередной лекции по программе (по плану лекций).
- Полезно вести записи (конспекты) лекций: для непонятных вопросов оставлять место при работе над темой лекции с учебными пособиями.
- Записи лекций следует вести в отдельной тетради, оставляя место для дополнений во время самостоятельной работы.
- При конспектировании лекций выделяйте главы и разделы, параграфы, подчеркивайте основное.

Лабораторное занятие

- Лабораторное занятие – наиболее активный вид учебных занятий в вузе. Он предполагает самостоятельную работу над лекциями и учебными пособиями.
- К каждому лабораторному занятию нужно готовиться. Подготовку следует начинать с повторения теории (по записям лекций или по учебному пособию). После этого нужно решать задачи из предложенного домашнего задания.

Организация самостоятельной работы

Самостоятельность в учебной работе способствует развитию заинтересованности студента в изучаемом материале, вырабатывает у него умение и потребность самостоятельно получать знания, что весьма важно для специалиста с высшим образованием. Самостоятельная работа студентов представлена в следующих формах:

- работа с учебной литературой и конспектом лекций с целью подготовки к лабораторным занятиям, составление конспектов тем, выносимых на самостоятельную проработку;
- систематическое выполнение домашних работ.

Таблица 4. Содержание самостоятельной работы обучающихся

Вопросы, выносимые на самостоятельное изучение	Кол-во часов	Форма работы
Основы параллельного программирования	38	Выполнение лабораторной работы № 1, 2
Основные объекты в параллельном программировании	38	Выполнение лабораторной работы № 3, 4, 5

5.3. Виды и формы письменных работ, предусмотренных при освоении дисциплины (модуля), выполняемые обучающимися самостоятельно

Лабораторная работа выполняется в рамках каждого раздела курса с целью усвоения прослушанного студентом теоретического материала.

Лабораторные работы должны быть сданы в период прочтения курса.

Сдача работы представляет собой предоставление отчета в свободной форме в письменном или электронном виде и, в случае необходимости, устные ответы на уточняющие вопросы по отдельным задачам.

6. ОБРАЗОВАТЕЛЬНЫЕ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

При реализации различных видов учебной работы по дисциплине «Параллельное программирование» могут использоваться электронное обучение и дистанционные образовательные технологии.

6.1. Образовательные технологии

Учебные занятия по дисциплине могут проводиться с применением информационно-телекоммуникационных сетей при опосредованном (на расстоянии) интерактивном взаимодействии обучающихся и преподавателя в режимах on-line или off-line в формах.

Таблица 5. Образовательные технологии, используемые при реализации учебных занятий

Раздел, тема дисциплины	Форма учебного занятия		
	Лекция	Практическое занятие, семинар	Лабораторная работа
Основы параллельного программирования	<i>Обзорная лекция</i>	<i>Не предусмотрено</i>	<i>Выполнение лабораторной работы</i>
Основные объекты в параллельном программировании	<i>Обзорная лекция</i>	<i>Не предусмотрено</i>	<i>Выполнение лабораторной работы</i>

6.2. Информационные технологии

При реализации различных видов учебной и внеучебной работы используются следующие информационные технологии:

- система управления обучением LMS Moodle;
- использование возможностей Интернета в учебном процессе (рассылка заданий, предоставление выполненных работ, ответы на вопросы, ознакомление обучающихся с оценками и т.д.);
- использование электронных учебников и различных сайтов (например, электронные библиотеки, журналы и т.д.) как источник информации;
- использование возможностей электронной почты;
- использование средств представления учебной информации (электронных учебных пособий, применение новых технологий для проведения занятий с использованием презентаций и т.д.);
- использование интерактивных средств взаимодействия участников образовательного процесса (технологии дистанционного или открытого обучения в глобальной сети);
- использование интегрированных образовательных сред, где главной составляющей являются не только применяемые технологии, но и содержательная часть, т.е. информационные ресурсы (доступ к мировым информационным ресурсам, на базе которых строится учебный процесс).

6.3. Программное обеспечение, современные профессиональные базы данных и информационные справочные системы

6.3.1. Программное обеспечение

Наименование программного обеспечения	Назначение
Adobe Reader	Программа для просмотра электронных документов
Платформа дистанционного обучения LMS Moodle	Виртуальная обучающая среда
Microsoft Office 2013, Microsoft Office Project 2013, Microsoft Office Visio 2013	Пакет офисных программ
7-zip	Архиватор
Microsoft Windows 7 Professional	Операционная система
Kaspersky Endpoint Security	Средство антивирусной защиты
Google Chrome	Браузер
OpenOffice	Пакет офисных программ

6.3.2. Современные профессиональные базы данных и информационные справочные системы

1. Электронная библиотека «Астраханский государственный университет» собственной генерации на платформе ЭБС «Электронный Читальный зал – БиблиоТех». <https://biblio.asu.edu.ru>
2. Электронно-библиотечная система (ЭБС) ООО «Политехресурс» «Консультант студента». www.studentlibrary.ru.
3. Электронная библиотечная система издательства ЮРАЙТ, раздел «Легендарные книги». www.biblio-online.ru
4. Электронный каталог Научной библиотеки АГУ на базе MARK SQL НПО «Информ-систем». <https://library.asu.edu.ru>

7. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

7.1. Паспорт фонда оценочных средств

При проведении текущего контроля и промежуточной аттестации по дисциплине «Параллельное программирование» проверяется сформированность у обучающихся компетенций, указанных в разделе 3 настоящей программы. Этапность формирования данных компетенций в процессе освоения образовательной программы определяется последовательным освоением дисциплин и прохождением практик, а в процессе освоения дисциплины – последовательным достижением результатов освоения содержательно связанных между собой разделов, тем.

Таблица 6 – Соответствие разделов, тем дисциплины, результатов обучения по дисциплине и оценочных средств

№ п/п	Контролируемые разделы, темы дисциплины (модуля)	Код контролируемой компетенции (компетенций)	Наименование оценочного средства
1	Основы параллельного программирования	ПК-1, ПК-3, ПК-7, ПК-10, ПК-12, ПК-13, ПК-15	лабораторная работа
2	Основные объекты в параллельном программировании	ПК-1, ПК-3, ПК-7, ПК-10, ПК-12, ПК-13, ПК-15	лабораторная работа

7.2. Описание показателей и критериев оценивания компетенций, описание шкал оценивания

Таблица 7 – Показатели оценивания результатов обучения в виде знаний

Шкала оценивания	Критерии оценивания
5 «отлично»	демонстрирует глубокое знание теоретического материала, умение обоснованно излагать свои мысли по обсуждаемым вопросам, способность полно, правильно и аргументированно

Шкала оценивания	Критерии оценивания
	но отвечать на вопросы, приводить примеры
4 «хорошо»	демонстрирует знание теоретического материала, его последовательное изложение, способность приводить примеры, допускает единичные ошибки, исправляемые после замечания преподавателя
3 «удовлетворительно»	демонстрирует неполное, фрагментарное знание теоретического материала, требующее наводящих вопросов преподавателя, допускает существенные ошибки в его изложении, затрудняется в приведении примеров и формулировке выводов
2 «неудовлетворительно»	демонстрирует существенные пробелы в знании теоретического материала, не способен его изложить и ответить на наводящие вопросы преподавателя, не может привести примеры

Таблица 8 – Показатели оценивания результатов обучения в виде умений и владений

Шкала оценивания	Критерии оценивания
5 «отлично»	демонстрирует способность применять знание теоретического материала при выполнении заданий, последовательно и правильно выполняет задания, умеет обоснованно излагать свои мысли и делать необходимые выводы
4 «хорошо»	демонстрирует способность применять знание теоретического материала при выполнении заданий, последовательно и правильно выполняет задания, умеет обоснованно излагать свои мысли и делать необходимые выводы, допускает единичные ошибки, исправляемые после замечания преподавателя
3 «удовлетворительно»	демонстрирует отдельные, несистематизированные навыки, испытывает затруднения и допускает ошибки при выполнении заданий, выполняет задание по подсказке преподавателя, затрудняется в формулировке выводов
2 «неудовлетворительно»	не способен правильно выполнить задания

7.3. Контрольные задания и иные материалы, необходимые для оценки результатов обучения по дисциплине (модулю)

Контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств:

Лабораторные работы

Лабораторная работа выполняется в рамках каждого раздела курса с целью усвоения прослушанного студентом теоретического материала.

Лабораторные работы должны быть сданы в период прочтения курса.

Сдача работы представляет собой предоставление отчета в свободной форме в письменном или электронном виде и, в случае необходимости, устные ответы на уточняющие вопросы по отдельным задачам.

Лабораторные работы

№ п/п	Номер раздела дисциплины	Наименование лабораторной работы	Трудоемкость, часов
1	1	Алгоритмы взаимного исключения	9
2	1	Атомарный снимок состояния	8
3	2	Консенсус	5
4	2	Базовые структуры данных в многопоточности	5
5	2	Применение JMM	5

Примеры заданий к лабораторной работе «Алгоритмы взаимного исключения»

1. Студенты должны реализовать программу, реализующую алгоритмы взаимного исключения: Петерсона, Лампорта

Примеры заданий к лабораторной работе «Атомарный снимок состояния»

1. Студенты должны реализовать программу для построения атомарного снимка состояния N регистров

Примеры заданий к лабораторной работе «Консенсус»

1. Студенты должны реализовать визуализацию метода достижения консенсуса

Примеры заданий к лабораторной работе «Базовые структуры данных в многопоточности»

1. Студенты должны реализовать программу, реализующую одну из базовых структур данных в условиях многопоточности

Примеры заданий к лабораторной работе «Применение JMM»

1. Студенты должны реализовать программу с использованием JMM

Порядок предоставления отчета по лабораторной работе

Отчет по лабораторной работе представляется в печатном виде в формате, предусмотренном шаблоном отчета по лабораторной работе. Время, отводимое на выполнение – 4 часа. Защита отчета проходит в форме доклада студента по выполненной работе и ответов на вопросы преподавателя.

Шаблон отчета по лабораторной работе

Отчет по лабораторной работе № _____

«Название лабораторной работы»

1. Цель и задачи лабораторной работы: _____
2. Методика проведения исследования: _____
3. Анализ погрешностей: _____
4. Результаты: _____
5. Выводы: _____

Требования к выполнению лабораторной работы

Отчеты по лабораторным работам должны быть отправлены на электронную почту преподавателя не позднее, чем через две недели после выдачи задания. Полученные выводы и графический материал должны быть информативными и корректными.

Перечень вопросов и заданий, выносимых на экзамен

Экзамен проводится в устной форме по билетам.

Порядок формирования экзаменационного билета: по одному вопросу из первой и второй части перечня вопросов к экзамену, дополнительный вопрос из любой части списка

Требования к ответу: полный развернутый ответ на каждый из вопросов

Возможность дополнительных вопросов: возможны уточнения в рамках лекционного материала

Перечень вопросов:

1. Логические часы Лампорта и векторные часы, их свойства
2. Часы с прямой зависимостью (и их свойства) и матричные часы.
3. Взаимное исключение в распределенной системе. Централизованный алгоритм.
4. Взаимное исключение в распределенной системе. Алгоритм Лампорта.
5. Взаимное исключение в распределенной системе. Алгоритм Рикарта и Агравалы.
6. Взаимное исключение в распределенной системе. Алгоритм обедающих философов.
7. Взаимное исключение в распределенной системе. Алгоритм на основе токена.
8. Взаимное исключение в распределенной системе. Алгоритмы основе кворума (простое большинство, рушащиеся стены).
9. Согласованное глобальное состояние (согласованный срез). Алгоритм Чанди-Лампорта. Запоминание сообщений на стороне отправителя.
10. Согласованное глобальное состояние (согласованный срез). Алгоритм Чанди-Лампорта. Запоминание сообщений на стороне получателя.
11. Глобальные свойства. Стабильные и нестабильные предикаты. Слабый конъюнктивный предикат. Централизованный алгоритм.
12. Слабый конъюнктивный предикат. Распределенный алгоритм.

13. Диффундирующие вычисления. Останов. Алгоритм Дейксты и Шолтена.
 14. Локально-стабильные предикаты, согласованные интервалы, барьерная синхронизация (3 алгоритма). Применение для определения взаимной блокировки.
 15. Упорядочивание сообщений. Определения, иерархия порядков. Алгоритм для FIFO.
 16. Упорядочивание сообщений. Определения, иерархия порядков. Алгоритм для причинно-согласованного порядка.
 17. Упорядочивание сообщений. Определения, иерархия порядков. Алгоритм для синхронного порядка.

Экзаменационный билет № 1

Вопрос 1. Взаимное исключение в распределенной системе. Алгоритм Лампорта.

Вопрос 2. Общий порядок (total order). Алгоритм Скина.

Таблица 9 – Примеры оценочных средств с ключами правильных ответов

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
ПК-1. Способен создавать, отлаживать и оформлять программный код				
1.	Задание закрытого типа	<p><i>Выберите верный ответ.</i></p> <p>Что такое поток в программировании?</p> <p>а. абстракция, созданная для выполнения параллельных задач в рамках одного процесса. б. абстракция, созданная для выполнения барьерных задач в рамках одной распределенной системы. в. абстракция, созданная для выполнения распределенных задач в рамках одного процесса. г. абстракция, созданная для выполнения удаленных процедур в распределенной системе.</p>	а	1-3
2.		<p><i>Выберите верный ответ.</i></p> <p>Многопоточное приложение на 2 ядрах для обработки 10 элементов массива работает дольше, чем однопоточное. Причина?</p> <p>а. необходимо использовать минимум 10 ядер б. необходимо использовать многопоточные приложения для обработки больших данных, на малых данных они работают неэффективно за счет времени на создания и синхронизацию потоков в. нужно перерабатывать алгоритм г. оно всегда будет работать дольше, так как занимает больше ресурсов процессора</p>	б	1-3
3.		<p><i>Выберите верные ответы.</i></p> <p>Какие языки программирования обладают встроенной поддержкой многопроцессности?</p>	б, в	1-3

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
		а. С б. Java в. C# г. TBB		
4.		<i>Выберите верный ответ.</i> К какой модели программирования относится модель, в которой все исполнители (потoki/процессоры) запускают одну программу, а разделение логики между ними основано на уникальных идентификаторах исполнителей? а. MPI б. Loop Parallelism в. SPMD г. Master/Worker	в	1-3
5.		<i>Выберите верный ответ.</i> Для чего предназначен шаблон Loop Parallelism? а. коммуникация внутри распределенной системы из вычислительных узлов, соединенных между собой по замкнутой цепи б. формирование очереди задач при конвейерных вычислениях в. распараллеливание циклов в программе путем распределения отдельных итераций между потоками г. балансировка нагрузки на процессоры при распределенных вычислениях	в	1-3
6.	Задание открытого типа	Опишите алгоритм для FIFO-порядка.	Алгоритм FIFO основан на нумерации сообщений: <ul style="list-style-type: none"> • Для каждой пары процессов нумеруем отправляемые сообщения натуральными числами. • Отправитель знает номер отправленного сообщения, получатель знает номер следующего ожидаемого сообщения. • К каждому сообщению прикрепляется номер. Если получатель получил "неожиданное" сообщение, то он складывает его в очередь и ждёт. Обработывает только когда подойдёт нужный номер.	5-8
7.		Дайте определение <i>диффундирующих вычислений</i> .	Диффундирующее вычисление — это вычисление, которое происходит следующим образом: <ul style="list-style-type: none"> • Каждый процесс бывает либо <i>активным</i>, либо <i>пассивным</i> • Если процесс получает сообщение, он становится активным • Только активный процесс может 	5-8

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			<p>посылать сообщения</p> <ul style="list-style-type: none"> Активный процесс может в любой момент по своему желанию стать пассивным <p>Исходно есть ровно один активный процесс — инициатор</p>	
8.		<p>Дайте определение <i>распределённому алгоритму для WCP</i>.</p>	<p>Распределенный алгоритм для WCP – алгоритм для поиска наименьшего (проще говоря, самого левого) согласованного среза в котором выполняется слабый конъюнктивный предикат.</p> <p>В распределенном алгоритме используются векторные часы, как и в централизованном (они вообще похожи, рекомендуется сначала понять централизованный).</p> <p>В дополнение к каждому из NN процессов заведем еще NN координаторов (вместо одного на всех), где каждый процесс связан со своим координатором. Каждый процесс отправляет сообщения либо другим процессам, либо своему координатору (каждый раз, когда выполняется локальный предикат и увеличились векторные часы). Каждый координатор отправляет сообщения только другим координаторам.</p>	8-10
9.		<p>В чем заключается основная идея алгоритма Дейкстры и Шолтена.</p>	<p>Алгоритм Дейкстры и Шолтена решает задачу останова диффундирующего вычисления в распределённой системе.</p> <p>Основная идея: выстроить процессы в дерево: кто кого активизировал. Процесс добавляется в дерево (становится "красным"), когда становится активным, а удаляется (становится "зелёным"), когда он и все его потомки стали пассивными и там нет сообщений (т.е. поддерево закончило вычисления). Исходно дерево содержит только инициатора, а когда вычисление остановится, станет пустым (о чём узнает инициатор).</p> <p>Каждый процесс будет требовать подтверждения на каждое своё сообщение, чтобы можно было учесть сообщения в пути.</p>	5-8
10.		<p>Дайте определение термину <i>кворум простого большинства</i>.</p>	<p>Кворум простого большинства — пример кворума, у которого любой элемент обладает мощностью строго больше половины.</p> <p>Если у нас есть 5 процессов P1..P5, то элементом кворума, как один из вариантов, будет подмножество {P1, P2, P3}. Таким образом, если P4 захочет попасть в критическую секцию, то ему будет достаточно получить подтверждение от {P1, P2, P3}.</p>	5-8
11.	<p>Задание комбинированного типа</p>	<p><i>Верно ли утверждение:</i> Многопоточное приложение на 2 ядрах для обработки 10 элементов массива работает дольше, чем однопоточное. Причина в том, что необходимо использовать минимум 10 ядер. Ответ обоснуйте.</p>	<p>Утверждение неверно, причина может заключаться в том, что необходимо использовать многопоточные приложения для обработки больших данных, на малых данных они работают неэффективно за счет времени на создания и синхронизацию потоков.</p>	3-5

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
ПК-3. Способен обеспечивать заданный уровень производительности, надежности и безопасности при создании вариантов архитектуры программного средства				
12.	Задание закрытого типа	<p><i>Выберите верный ответ.</i></p> <p>Подождать поток в C# можно...</p> <p>а. с использованием метода StopThread б. с использованием метода WaitThread в. с использованием метода StopProcess г. с использованием метода Join</p>	г	1-3
13.		<p><i>Выберите верный ответ.</i></p> <p>Если в системе 4 ядра, то сколько потоков необходимо создать для эффективной работы многопоточного приложения?</p> <p>а. 4 б. 2 в. 8 г. 100</p>	4	1-3
14.		<p><i>Выберите верный ответ.</i></p> <p>Какой из стандартов параллельного программирования предназначен для использования в системах с общей памятью?</p> <p>а. OpenMP б. MPI в. OpenMPI г. MPICH</p>	а	1-3
15.		<p><i>Выберите верные ответы.</i></p> <p>Какие из реализаций стандарта параллельного программирования предназначены для использования в системах с распределенной памятью?</p> <p>а. OpenMP б. MPI в. OpenMPI г. MPICH</p>	а, г	1-3
16.		<p><i>Выберите верный ответ.</i></p> <p>Как можно создать поток в C#?</p> <p>а. с помощью System.Distributed.ParallelObject б. с помощью System.Distributed.ThreadObject в. с помощью System.Threading.Thread г. с помощью Thread.Distributed.Process</p>	г	1-3

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
17.	Задание открытого типа	Сформулируйте задачу взаимной блокировки.	Задача: есть процессы, они могут друг друга ждать (в этот момент они ничего не могут делать и не меняют своё внутреннее состояние). Мы хотим запустить алгоритм, который будет нам сообщать о появлении взаимных блокировок, когда процессы ждут друг друга по циклу. Практическое применение — поиск ошибок в программах (динамические анализаторы распределённого кода, базы данных и SQL-запросы).	5-8
18.		Дайте определение <i>синхронному порядку сообщений</i> .	В системе есть синхронный порядок сообщений, если всем сообщениям можно сопоставить время $T(m)$ (число) так, что верно: 1. Для любого сообщения $m: T(recv(m)) = T(snd(m))$ (обозначается $T(m)$) 2. Для любых двух событий $e \rightarrow f$ верно $T(e) < T(f)$ (обратное может быть неверно), т.е. T является логическими часами.	5-8
19.		Запишите 4 гарантии порядка сообщений в иерархии порядков сообщений.	В распределённых системах могут быть разные гарантии порядка доставки отправленных сообщений. Более того: иногда программисты могут неявно предполагать тот или иной порядок и очень удивляться, когда он нарушается. Мы рассматриваем четыре гарантии порядка сообщений, от более слабых к более сильным: 1. Асинхронная передачи: никаких гарантий, только exactly-once delivery 2. FIFO (First In First Out order) 3. Причинно-согласованный порядок (causally consistent ordering, от слова cause, а не casual) Синхронный порядок	5-8
20.		Как можно уменьшит количество посылок о выполнении предиката?	Если в каком-то процессе выполнялся предикат, а потом он получил сообщение, то не надо ещё раз высылать сообщение координатору. Доказательство: если бы было решение, где новое состояние процесса после получения сообщения было границей искомого среза, то мы можем сдвинуть это состояние назад. Предикат всё ещё будет выполняться, а срез менее согласованным не станет, т.к. получения сообщений можно выкидывать безболезненно.	5-8
21.		Имеется следующий код на OpenMP: double a = 10.0; #pragma omp parallel private(a) { ... } Можно ли средствами OpenMP сделать так, чтобы локальные копии переменной a в каждом потоке имели то же значение, что и исходная переменная a ?	Да. Нужно заменить параметр private на параметр firstprivate.	3-5

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
выбор среди них				
22.	Задание закрытого типа	<p><i>Выберите верный ответ.</i></p> <p>Что такое распределенная система?</p> <p>а. комплекс приложений для Іс, которые распределяют потоки по процессорам суперкомпьютера б. система финансового распределения задач в экономическом кластере между его узлами в. система распределения задач исполнителям в бизнес информатике г. комплекс, представляющий собой многокомпонентные приложения на узлах, взаимодействующих по сети</p>	г	1-3
23.		<p><i>Выберите верный ответ.</i></p> <p>Зачем необходимо ожидание рабочих потоков при решении параллельных задач?</p> <p>а. необходимо синхронизовать параллельные действия в распределенной системе для распределения задач между процессорами б. необходимо закрыть семафор, чтобы синхронизовать данные в. поскольку потоки могут остаться в операционной системе и забивать процессорное время г. поскольку необходимо дождаться промежуточных результатов и сделать редукцию результата</p>	г	1-3
24.		<p><i>Выберите верный ответ.</i></p> <p>Барьерная синхронизация – это...</p> <p>а. это средство в С# для создания потоков б. нужен для разделения заданий и сбора результатов в. нужен для создания потоков в многокомпонентной системе г. нужен для ожидания потоков в распределенной системе</p>	б	1-3
25.		<p><i>Выберите верный ответ.</i></p> <p>При программировании сервиса на С# используется...</p> <p>а. System.Threading.ThreadService б. специальный именованный семафор в. удаленные процедуры для работы с потоками</p>	б	1-3

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
		г. атрибут класса [ServiceContract]		
26.		<p>Выберите верный ответ.</p> <p>Если нужно вызвать удаленную процедуру, возвращающую курсы валют на бирже со стороннего сервиса, нужно использовать технологию...</p> <p>а. удаленное ожидание с <code>map-reduce</code> б. семафор в. веб-сервис г. канал</p>	в	1-3
27.	Задание открытого типа	Можно ли использовать директиву <code>OpenMP #pragma omp for</code> в функции <code>f()</code> , в которой не содержится директивы <code>#pragma omp parallel</code> ?	Да. Будет ли цикл выполняться несколькими потоками, зависит от того, в каком контексте будет вызвана функция <code>f()</code> : из параллельной секции или из последовательного кода	3-5
28.		Дайте определение <i>барьерной синхронизации</i> .	Интервал $[G, H]$ ($G \subseteq H$) называется барьерно-синхронизированным, если для любых событий $e \in G$ и $f \in H$ верно, что $e \rightarrow f$.	5-8
29.		Вы написали программу с использованием OpenMP. При запуске на 4-ядерной системе оказалось, что загрузка процессора равна 25%. Какова наиболее вероятная причина?	отсутствие ключа <code>/openmp (/Qopenmp)</code> при сборке программы	3-5
30.		Дайте определение понятию <i>слабый конъюнктивный предикат</i> .	Слабый конъюнктивный предикат (WCP) — предикат, имеющий вид конъюнкции локальных предикатов над состоянием каждого процесса. Слабый конъюнктивный предикат P истинен, если он истинен на хотя бы одном согласованном срезе	5-8
31.		Дайте определения стабильным и локальным предикатам.	Предикат PP является стабильным , если для любых согласованных срезов $G \leq H \leq N$ из $P(G)P(G)$ следует $P(H)P(H)$. Предикат PP является локальным , если он зависит только от состояния одного конкретного процесса. Например, если у процесса есть состояние (переменная) x , то локальным предикатом может быть $(x < 0)$.	5-8
ПК-10. Разработка и реализация архитектуры программного обеспечения				
32.	Задание закрытого типа	<p>Выберите верный ответ.</p> <p>Что такое суперкомпьютер?</p> <p>а. персональный компьютер, пользователь которого характеризует его как "супер"; б. компьютер с максимальными характеристиками производительности на данный момент, в его состав могут входить сотни тысяч процессоров; в. компьютер будущего. Его создание только планируется;</p>	б	1-3

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
		г. четырнадцатиядерный 164-х битный компьютер		
33.		<p><i>Выберите верный ответ.</i></p> <p>Проблема гонки данных это...</p> <p>а. проблема гонки обедающих философов за едой б. доступ к общим данным в программе из конкурирующих процессов/потоков в. доступ к общему семафору в ОС из конкурирующих процессов/потоков г. доступ к ядру процессора за данными в распределенной системе</p>	б	1-3
34.		<p><i>Выберите верный ответ.</i></p> <p>Технология программирования OpenMP расширяет язык программирования за счет:</p> <p>а. новых ключевых слов; б. новых библиотечных функций и переменных окружения; в. новых директив и специальных комментариев; г. новых переменных.</p>	б, в	1-3
35.		<p><i>Выберите верный ответ.</i></p> <p>Проблема гонки данных это...</p> <p>а. проблема гонки обедающих философов за едой б. доступ к общим данным в программе из конкурирующих процессов/потоков в. доступ к общему семафору в ОС из конкурирующих процессов/потоков г. доступ к ядру процессора за данными в распределенной системе</p>	б	1-3
36.		<p><i>Выберите верный ответ.</i></p> <p>Что такое удаленный вызов процедуры?</p> <p>а. процедура вызова потока для синхронизации удаленности б. вызов процедуры, работающей на другом узле с возвратом результата в. передача данных между 1с формами в многопоточном режиме г. вызов процедуры через некоторое время</p>	б	1-3
37.	Задание	Дайте определение <i>централизо-</i>	Централизованный алгоритм для WCP –	5-8

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
	открытого типа	<i>ванному алгоритму для WSP.</i>	алгоритм для поиска наименьшего (проще говоря, самого левого) согласованного среза в котором выполняется слабый конъюнктивный предикат. Если есть хотя бы один согласованный срез, в котором выполняется слабый конъюнктивный предикат, то такой срез существует и единственен (см. слабый конъюнктивный предикат). В централизованном алгоритме используются векторные часы. Срез задается набором векторных часов для всех процессов или просто вектором, в котором соответствующая компонента показывает время для соответствующего потока.	
38.		Дайте определение понятию <i>глобальные свойства систем.</i>	Глобальные свойства системы — это предикаты от согласованных срезов системы (состояния процессов плюс сообщения в пути), которые бывают двух видов: <ul style="list-style-type: none"> • стабильные предикаты; • нестабильные предикаты. Говорить "предикат сейчас верен" или "предикат сейчас неверен" некорректно, потому что нет "сейчас". Можно говорить "существует/не существует согласованный срез", но если у нас система ещё работает, то "не существует" мы можем заявить лишь на каком-то префиксе срезов системе.	5-8
39.		Дайте описание алгоритму Чанди-Ломпорта.	Алгоритм Чанди-Лампорта получения согласованного среза с запоминанием на стороне получателя с учетом FIFO канала связи. В алгоритме мы ставим в соответствие каждому процессу белый или красный цвет. Изначально все процессы белые. Когда все процессы станут красными, мы получим согласованный срез. Есть инициатор (observer) с которого всё начинается. Он становится красным, запоминает свое состояние и посылает маркер всем другим процессам, с которыми у него есть канал. Только после этого он может снова обрабатывать полученные сообщения или же отправлять. Если процесс получил маркер и еще белый, то он обязан стать красным, записать свое локальное состояние и отправить остальным процессам маркеры. Иначе ничего делать не надо. Таким образом, если процесс только что стал красным (и записал своё состояние в согласованный срез), то все сообщения, которые он уже обработал, были посланы остальными процессами в моменты, которые учтены в согласованном срезе из-за FIFO. Будем называть сообщения красными, если они отправлены от красного процесса, аналогично для белого процесса. С технической точки зрения, при отправке сообщения или же маркера, будем добавлять цвет передаваемого сообщения.	8-10

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
40.		Как можно уменьшит количество посылки о выполнении предиката?	Если в каком-то процессе выполнялся предикат, а потом он получил сообщение, то не надо ещё раз высылать сообщение координатору. Доказательство: если бы было решение, где новое состояние процесса после получения сообщения было границей искомого среза, то мы можем сдвинуть это состояние назад. Предикат всё ещё будет выполняться, а срез менее согласованным не станет, т.к. получения сообщений можно выкидывать безболезненно.	5-8
41.		В чем заключается основная идея алгоритма Дейкстры и Шолтена.	Алгоритм Дейкстры и Шолтена решает задачу останова диффундирующего вычисления в распределённой системе. Основная идея: выстроить процессы в дерево: кто кого активизировал. Процесс добавляется в дерево (становится "красным"), когда становится активным, а удаляется (становится "зелёным"), когда он и все его потомки стали пассивными и там нет сообщений (т.е. поддерево закончило вычисления). Исходно дерево содержит только инициатора, а когда вычисление остановится, станет пустым (о чём узнает инициатор). Каждый процесс будет требовать подтверждения на каждое своё сообщение, чтобы можно было учесть сообщения в пути.	5-8
ПК-12. Разработка компонентов системных программных продуктов и интеграция разработанного программного обеспечения				
42.	Задание закрытого типа	<i>Выберите верный ответ.</i> Map - reduce подход ... а. нужен для разделения заданий и сбора результатов б. нужен для ожидания потоков в распределенной системе в. это средство в C# для создания потоков г. нужен для создания потоков в многокомпонентной системе	а	1-3
43.		<i>Выберите верный ответ.</i> Узел распределенной системы – это... а. ERP для сетевых потоков б. бизнес приложение для распределения задач между семафорами на мультипроцессорной системе в. связь потоков на процессоре г. устройство, на котором работает один из компонентов распределенной системы	г	1-3
44.		<i>Выберите верный ответ.</i> MPI – это ... а. многопоточные интерфейсы	г	1-3

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)																									
		инициализации общих данных б. UML диаграмма в. мульти-процессорные интерфейсы синхронизации сетевых сервисов WCF г. мехнология для программирования вычислительных систем для суперкомпьютеров																											
45.		<i>Выберите верный ответ.</i> Если нужно вызвать удаленную процедуру, возвращающую курсы валют на бирже со стороннего сервиса, нужно использовать технологию... а. удаленное ожидание с <code>task.Wait</code> б. семафор в. веб-сервис г. канал	в	1-3																									
46.		<i>Выберите верный ответ.</i> Проблема обедающих философов может быть решена... а. добавлением синхронизации с еще одной вилкой б. добавлением процесс-официанта в. добавлением семафора у первого философа г. добавлением философа и двух вилок	б	1-3																									
47.	Задание открытого типа	Сформулируйте теорему <i>нижняя граница для срезов</i> .	Заметим, что если есть два согласованных среза G_1G_1 и G_2G_2 , то срез $G_1 \cap G_2 G_1 \cap G_2$ тоже согласован и, более того, $(G_1 \cap G_2) \leq G_1, G_2 (G_1 \cap G_2) \leq G_1, G_2$. Доказательство: рассмотрим, какие сообщения могут пересылаться между различными частями системы: <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>откуда\куда</th> <th>$G_1 \cap G_2$</th> <th>$G_1 \cap G_2$</th> <th>$G_1 \cap G_2$</th> <th>$G_1 \cap G_2$</th> </tr> </thead> <tbody> <tr> <td>$G_1 \cap G_2$</td> <td>*</td> <td>*</td> <td>*</td> <td>*</td> </tr> <tr> <td>$G_1 \cap G_2$</td> <td>-</td> <td>*</td> <td>-</td> <td>*</td> </tr> <tr> <td>$G_1 \cap G_2$</td> <td>-</td> <td>-</td> <td>*</td> <td>*</td> </tr> <tr> <td>$G_1 \cap G_2$</td> <td>-</td> <td>-</td> <td>-</td> <td>*</td> </tr> </tbody> </table> Заметим, что не существует сообщений, которые бы пересылались из $E \setminus (G_1 \cap G_2) E \setminus (G_1 \cap G_2)$ в $G_1 \cap G_2 G_1 \cap G_2$, что и требовалось.	откуда\куда	$G_1 \cap G_2$	*	*	*	*	$G_1 \cap G_2$	-	*	-	*	$G_1 \cap G_2$	-	-	*	*	$G_1 \cap G_2$	-	-	-	*	8-10				
откуда\куда	$G_1 \cap G_2$	$G_1 \cap G_2$	$G_1 \cap G_2$	$G_1 \cap G_2$																									
$G_1 \cap G_2$	*	*	*	*																									
$G_1 \cap G_2$	-	*	-	*																									
$G_1 \cap G_2$	-	-	*	*																									
$G_1 \cap G_2$	-	-	-	*																									
48.		Дайте определение понятию согласованный срез.	Мотивация: если у распределенной системы нет «глобального состояния», то как запомнить её состояние на диске, чтобы можно было продолжить работу после восстановления с диска? Пусть EE — множество событий с полным порядком (\ll) в рамках каждого процесса. Согласованный срез G — подмножество E такое, что $\forall f \in E, \forall g \in G : f \rightarrow g \Rightarrow f \in G.$	5-8																									
49.		Дайте определение понятию срез.	Мотивация: если у распределенной системы	5-8																									

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			нет «глобального состояния», то как запомнить её состояние на диске, чтобы можно было продолжить работу после восстановления с диска? Пусть E — множество событий с полным порядком (\ll) в рамках каждого процесса. Срез F — подмножество E такое, что $\forall f \in E, \forall g \in G : f \rightarrow g \Rightarrow f \in G$.	
50.		Дайте определение термину <i>кворум рушащейся стенки</i> .	Кворум рушащейся стенки (Crumbling walls quorum) — пример кворума с подмножествами размера порядка $O(\sqrt{n})$ вместо $O(n)$ как у кворума простого большинства. Структура: Процессы упорядочены в линии, по возможности равной длины. Минимальный элемент кворума — объединение всех процессов какой-то одной линии плюс по одному представителю из каждой линии ниже.	5-8
51.		Дайте определение термину <i>кворум простого большинства</i> .	Кворум простого большинства — пример кворума, у которого любой элемент обладает мощностью строго больше половины. Если у нас есть 5 процессов P1..P5, то элементом кворума, как один из вариантов, будет подмножество {P1, P2, P3}. Таким образом, если P4 захочет попасть в критическую секцию, то ему будет достаточно получить подтверждение от {P1, P2, P3}.	5-8
ПК-13. Способность разрабатывать, анализировать, реализовывать и внедрять алгоритмы и структуры данных в рамках разработки системного и прикладного программного обеспечения				
52.	Задание закрытого типа	Верно ли утверждение: OpenMP ориентирован в первую очередь на написание программ для векторно-конвейерных компьютеров	неверно	1-3
53.		Верно ли утверждение: большинство конструкций OpenMP реализуется с помощью спецкомментариев	верно	1-3
54.		<i>Выберите верный ответ.</i> Для распределения итераций цикла между потоками необходимо использовать следующую директиву OpenMP: а. #pragma omp parallel in б. #pragma omp parallel for в. #pragma parallel for	б	1-3
55.		<i>Выберите верный ответ.</i> Зачем нужен семафор? а. для синхронизации деделок б. для решения задачи бесконечного ожидания в. для доступа к квантованию времени г. для синхронизации процессов и потоков	г	1-3
56.		<i>Выберите верный ответ.</i>	в	1-3

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
		<p>Преимущества вызова удаленных сервисов для ERP систем?</p> <p>а. ERP системы не могут предоставить нужные возможности вызова удаленных сервисов б. обработка данных баз данных в. делегирование сложных функций, требующих математики, многопоточности или сложных действий, что сложно написать на базе модельного языка ERP системы г. необходимо написание семафоров и каналов для моделирования действий, спроектированных на UML диаграмме Activity</p>		
57.	Задание открытого типа	Опишите алгоритм Лампорта.	<p>Алгоритм Лампорта -- алгоритм взаимного исключения, работающий в случае, если все сообщения идут FIFO и использующий логические часы Лампорта (т.е. с каждым сообщением посылается ещё и временная метка).</p> <p>Каждый поток поддерживает очередь запросов на вход в критическую секцию. Приоритет – <временная метка, номер потока> (т.е. при равенстве временных меток берем тот поток, чей номер меньше, иначе возможна взаимная блокировка, если процессы будут общаться идеально симметрично).</p> <p>Когда поток хочет войти в критическую секцию, он:</p> <p>Добавляет свой запрос в свою очередь (т.е. временную метку и номер потока) Посылает всем потокам запрос (req) Ждет от них ответа (ok) Получив все ответы, ждет, когда он станет первым в своей очереди, и входит в критическую секцию Выйдя из критической секции, удаляет свой запрос из своей очереди, посылает всем сообщение о том, что вышел (rel) Действия вне критической секции:</p> <p>При получении запроса от другого потока, запрос добавляется в очередь и запрашивающему потоку посылается ответ При получении release от другого потока, его запрос удаляется из очереди</p>	8-10
58.		Опишите алгоритм Рикарта-Агравалы.	<p>Алгоритм Рикарта-Агравалы — алгоритм взаимного исключения, являющийся оптимизацией алгоритма Лампорта.</p> <p>Мы объединяем запросы rel и ok в один: вместо отправки ok сразу посылает только если не хотим входить в критическую секцию или сразу как только выйдем из секции.</p> <p>Когда процесс P_i</p>	8-10

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			хочет войти в критический участок, то рассылает всем сообщение req с текущей временной меткой. Когда процесс P _k получает от P _j запрос войти в критический участок: если он сам не посылал запрос, то посылает отклик; если он послал свой запрос, он сравнивает временные метки этих двух запросов и посылает отклик только если у его собственного запроса метка позже (на самом деле, при равенстве меток, нужно проверять, у кого больше номер, т.е. для отсутствия блокировок нужно ввести приоритет на потоках); в остальных случаях процесс P _k задерживает отправку отклика. Процесс может войти в критический участок только после получения откликов ото всех других узлов сети. После выхода из него он рассылает задержанные отклики на все ожидающие запросы.	
59.		Сформулируйте <i>обобщенную</i> задачу <i>обедаящих философов</i> .	В случае с философами у нас есть N процессов и граф конфликтов между ними из N рёбер (одно ребро — один общий для двух процессов ресурс). Процессу для работы надо собрать все конфликтующие с другими потоками ресурсы. А взаимное исключение — это полный граф конфликтов, т.е есть вилка для каждой пары философов. Так что алгоритм для философов можно использовать и для задачи взаимного исключения.	5-8
60.		Дайте определение понятию <i>кворум</i> .	Кворум - это семейство Q подмножеств множества процессов 2^P причем: 1. Q замкнуто относительно взятия надмножества, т.е. если $Q \ni A \subseteq B$, то $B \in Q$ 2. Любые два элемента (т.е подмножества относительно множества процессов) этого семейства имеют непустое пересечение.	5-8
61.		Дайте определение термину <i>кворум рушащейся стенки</i> .	Кворум рушащейся стенки (Crumbling walls quorum) — пример кворума с подмножествами размера порядка $O(\sqrt{n})$ вместо $O(n)$ как у кворума простого большинства. Структура: Процессы упорядочены в линии, по возможности равной длины. Минимальный элемент кворума — объединение всех процессов какой-то одной линии плюс по одному представителю из каждой линии ниже.	5-8
ПК-15. Способностью проектировать и реализовывать сложные программные системы и комплексы, в том числе высоконагруженные, распределенные и масштабируемые гетерогенные системы				
62.	Задание закрыто-	Выберите <i>верный</i> ответ.	а	1-3

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
	го типа	Проблема «обедающих философов» – это... а. моделирует дедлок б. моделирует алгоритм выбора лидера в. моделирует семафор г. моделирует гонку данных		
63.		<i>Выберите верный ответ.</i> Проблема «обедающих философов» – это... а. моделирует дедлок б. моделирует алгоритм выбора лидера в. моделирует семафор г. моделирует гонку данных	а	1-3
64.		<i>Выберите верный ответ.</i> Что такое WCF? а. UML диаграмма коммуникаций б. технология работы с семафорами в. технология web сервисов в C#.net г. технология web сервисов в IC	в	1-3
65.		<i>Выберите верный ответ.</i> Барьерная синхронизация – это... а. это средство в C# для создания потоков б. нужен для разделения заданий и сбора результатов в. нужен для создания потоков в многокомпонентной системе г. нужен для ожидания потоков в распределенной системе	б	1-3
66.		<i>Выберите верный ответ.</i> При программировании сервиса на C# используется... а. System.Threading.ThreadService б. специальный именованный семафор в. удаленные процедуры для работы с потоками г. атрибут класса [ServiceContract]	б	1-3
67.	Задание открытого типа	Дайте определение понятию <i>кворум</i> .	Кворум - это семейство Q подмножеств множества процессов 2^P причем: 1. Q замкнуто относительно взятия надмножества, т.е. если $Q \ni A \subseteq B$, то $B \in Q$ 2. Любые два элемента (т.е подмножества относительно множества процессов) этого семейства имеют непустое пересечение.	5-8

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
68.		Дайте определения стабильным и локальным предикатам.	Предикат РР является стабильным , если для любых согласованных срезов $G \leq HG \leq H$ из $P(G)P(G)$ следует $P(H)P(H)$. Предикат РР является локальным , если он зависит только от состояния одного конкретного процесса. Например, если у процесса есть состояние (переменная) x , то локальным предикатом может быть $(x < 0)$.	5-8
69.		Сформулируйте задачу <i>обедающих философов</i> .	За круглым столом сидят шесть философов (процессы). На столе шесть тарелок с рисом и шесть вилок. В каждый момент философ может либо хотеть есть (тогда ему необходимо две вилки), либо хотеть думать (для этого ему нужен только он сам). Философы не разговаривают между собой (только в начале, когда договариваются об алгоритме). Необходимо, чтобы все философы поели и не возникло драки за вилки.	5-8
70.		Опишите <i>централизованный алгоритм</i> взаимного исключения.	Централизованный алгоритм взаимного исключения: Центральный управляющий узел ведет очередь запросов на вход. Процесс, желающий войти в критический участок, посылает ему сообщение "запрос" и ждет, пока не получит сообщение "разрешение" (когда подходит его очередь). Когда процесс выходит из критического участка, то посылает управляющему узлу сообщение "освобождение". Всего нужно послать три сообщения для каждого случая доступа к критическому ресурсу. Не масштабируется из-за необходимости иметь выделенного координатора.	5-8
71.		В чем заключается отличие векторных часов от матричных?	Матричные часы – это обобщение векторных часов. В отличие от векторных часов: Каждый процесс хранит вектор векторов (матрицу) целых чисел; При передаче сообщения передается вся имеющаяся матрица; В случае любого события, i -ый поток увеличивает на единицу свою компоненту матрицы, т.е $M[i][i]$; При приеме сообщения для всех строк, кроме строки, соответствующей номеру получившего потока ($myId$), берется покомпонентный максимум. А для $M[myId][j]$ берется максимум из $M[myId][j]$ и $W[srcId][j]$, где W - матрица, которая пришла, $srcId$ - номер потока, от которого пришла матрица W . Используя матричные часы, мы можем оценить нижнюю границу того, что знает другой поток.	8-10

Полный комплект оценочных материалов по дисциплине (модулю) (фонд оценочных средств) хранится в электронном виде на кафедре, утверждающей рабочую программу дисциплины (модуля), и в Центре мониторинга и аудита качества обучения.

7.4. Методические материалы, определяющие процедуры оценивания результатов обучения по дисциплине (модулю)

Таблица 10 – Технологическая карта рейтинговых баллов по дисциплине (модулю)

№ п/п	Контролируемые мероприятия	Количество мероприятий / баллы	Максимальное количество баллов	Срок представления
Основной блок				
1.	Выполнение лабораторных работ	5/8	30	Сроки указаны в Moodle
Всего			40	-
Блок бонусов				
2.	Посещение занятий		10	
Всего			10	-
Дополнительный блок**				
3.	Экзамен			
Всего			50	-
ИТОГО			100	-

Таблица 11 – Шкала перевода рейтинговых баллов в итоговую оценку за семестр по дисциплине

Сумма баллов	Оценка по 4-балльной шкале
90–100	5 (отлично)
85–89	4 (хорошо)
75–84	
70–74	
65–69	
60–64	3 (удовлетворительно)
Ниже 60	2 (неудовлетворительно)

При реализации дисциплины в зависимости от уровня подготовленности обучающихся могут быть использованы иные формы, методы контроля и оценочные средства, исходя из конкретной ситуации.

8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

8.1. Основная литература

1. Hopcroft J. E., Motwani R., Ullman J. D. Introduction to Automata Theory, Languages, and Computation (3rd Edition). — Addison-Wesley, Boston, MA, USA, 2006. — 750 с.
2. Шень А. Программирование: теоремы и задачи. — М.: МЦНМО, 2014. — 296 с.
3. Шень А., Верещагин Н. Языки и исчисления. — М.: МЦНМО, 2012. — 240 с.
4. Верещагин, Н. К. Колмогоровская сложность и алгоритмическая случайность [Электронный ресурс] / Н. К. Верещагин, В. А. Успенский, А. Шень. — Электрон. дан. — СПб: Лань, 2013. — 575 с. — Режим доступа: <https://e.lanbook.com/book/56395> — Загл. с экрана.

8.2. Дополнительная литература

Кривцова, И. Е. Основы дискретной математики. Часть 1. Учебное пособие [Электронный ресурс] / И. Е. Кривцова, И. С. Лебедев, А. В. Настека. — Электрон. дан. — СПб: ИТМО, 2016. — 92 с. — Режим доступа:

http://books.ifmo.ru/book/1869/osnovy_diskretnoy_matematiki_chast_1_uchebnoe_posobie.htm

— Загл. с экрана.

1. Ганичева, А. В. Дискретная математика : учебное пособие для вузов / А. В. Ганичева, А. В. Ганичев. — 2-е изд., стер. — Санкт-Петербург : Лань, 2024. — 160 с. — ISBN 978-5-507-49204-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/382370>
2. Шевелев, Ю. П. Дискретная математика : учебное пособие для вузов / Ю. П. Шевелев. — 5-е изд., стер. — Санкт-Петербург : Лань, 2024. — 592 с. — ISBN 978-5-507-49681-5. —

Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/399194>

8.3. Интернет-ресурсы, необходимые для освоения дисциплины

1. Вики-конспекты. — http://neerc.ifmo.ru/wiki/index.php?title=Заглавная_страница
2. Электронный каталог Научной библиотеки АГУ на базе MARK SQL НПО «Информ-систем»: <https://library.asu.edu.ru>
3. Корпоративный проект Ассоциации региональных библиотечных консорциумов (АРБИКОН) «Межрегиональная аналитическая роспись статей» (МАРС): <http://mars.arbicon.ru>
4. Единое окно доступа к образовательным ресурсам <http://window.edu.ru>

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Для проведения **лекционных занятий**:

1. Используется аудитория, оборудованная необходимым количеством столов, стульев, доской маркерной и электронной.
2. Аудитория должна иметь следующие нормы освещенности
 - СНиП 23-05-95 «Естественное и искусственное освещение» норма освещенности аудиторий ВУЗов 400 Лк.
 - СанПиН 2.2.1/2.1.1.1278-03 «Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий» пункт 3.3.3. «Общее освещение в помещениях общественных зданий должно быть равномерным».
3. Электронная доска должна быть подключена к сети Интернет.

Для проведения **лабораторных занятий**:

1. Лабораторные занятия проводятся с группами или подгруппами не более 15 человек.
2. Аудитория должна быть оснащена необходимым количеством столов, стульев, доской маркерной и электронной.
4. Аудитория должна иметь следующие нормы освещенности
 - СНиП 23-05-95 «Естественное и искусственное освещение» норма освещенности аудиторий ВУЗов 400 Лк.
 - СанПиН 2.2.1/2.1.1.1278-03 «Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий» пункт 3.3.3. «Общее освещение в помещениях общественных зданий должно быть равномерным».
5. В аудитории должно быть не менее 15 компьютеров, находящихся в исправном состоянии.
6. Расположение компьютеров в аудитории должно позволять преподавателю подойти к рабочему месту студента.
7. Компьютеры должны быть соединены локальной сетью со скоростью не менее 1 Гбит/с и подключены к сети Интернет.
8. Компьютеры должны обладать минимальными характеристиками:
 - Объем оперативной памяти 16 Гб
 - Накопитель SDD 500 Гб
 - Процессор 12th Gen Intel(R) Core(TM) i3-12100
 - Видеоадаптер Intel(R) UHD Graphics 730

Рабочая программа дисциплины (модуля) при необходимости может быть адаптирована для обучения (в том числе с применением дистанционных образовательных технологий) лиц с ограниченными возможностями здоровья, инвалидов. Для этого требуется заявление обучающихся, являющихся лицами с ограниченными возможностями здоровья, инвалидами, или их законных представителей и рекомендации психолого-медико-педагогической комиссии. Для инвалидов содержание рабочей программы дисциплины (модуля) может опреде-

ляться также в соответствии с индивидуальной программой реабилитации инвалида (при наличии).