ЦЕНТРАЛЬНЫЙ УНИВЕРСИТЕТ

УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«24» июня 2025 г.
Протокол №2

Рабочая программа дисциплины (модуля)
«Многопоточная синхронизация»

Направление подготовки: *02.03.01 Математика и компьютерные науки*

Направленность (профиль) подготовки: *Математика и компьютерные науки*

Квалификация (степень) выпускника: *бакалавр*

Форма обучения: *очная*

Срок освоения программы: *4 года*

Год набора: *2025*

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 0215B0C000D1B3FEB44A37EE631E0C27D4
Владелец: Ивашкевич Евгений Васильевич
Действителен: с 13.01.2026 до 13.04.2027

Электронный документ

Москва
2025

Содержание

1. **Краткая характеристика дисциплины (модуля)** 3
2. **Перечень планируемых результатов обучения** 4
3. **Тематический план** 5
4. **Содержание дисциплины (модуля)** 5
5. **Учебно-методическое обеспечение** 6
6. **Материально-техническое обеспечение** 6
7. **Методические и оценочные материалы** 8

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Многопоточная синхронизация» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по специальности 02.03.01 Математика и компьютерные науки, профиль Математика и компьютерные науки, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 807 от 23.08.2017 года.

Изучение дисциплины (модуля) «Многопоточная синхронизация» позволяет студентам овладеть ключевыми навыками разработки эффективных и масштабируемых программ, что особенно важно в условиях современного многопроцессорного оборудования. Это знание помогает избежать распространенных ошибок, связанных с многопоточностью, и обеспечивает создание надежного программного обеспечения, что является важным аспектом в будущей профессиональной деятельности.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки бакалавриата по направлению 02.03.01 Математика и компьютерные науки, профиль Математика и компьютерные науки и входит в обязательную часть Блока 1.

Дисциплина (модуль) доступна для изучения на 2 или 3 курсе в 4 или 6 семестре на выбор.

Цель изучения дисциплины (модуля): заключается в формировании понимания проблемы синхронизации и гонки данных в многопоточных приложениях, навыков исследования реализации примитивов синхронизации и их применения для разработки надежных многопоточных приложений и сервисов.

Задачи изучения дисциплины (модуля):

— изучить основные концепции многопоточности, включая механизмы синхронизации, такие как мьютексы, семафоры и условные переменные;

— освоить методы предотвращения проблем, возникающие в параллельном программировании;

— развить навыки анализа и применения паттернов синхронизации в реальных приложениях, а также развитие умений по отладке и оптимизации многопоточного кода.

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

— распространенные проблемы гонки данных;

— устройство примитивов синхронизации;

уметь:

— применять механизмы синхронизации;

— реализовывать структуры данных и алгоритмы, используемые в многопоточном программировании;

владеть:

— навыком реализации надежных многопоточных программ.

2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

Компетенция	Содержание компетенции	Индикатор компетенции	Перечень планируемых результатов обучения по дисциплине (модулю)
ОПК-3.	Способен самостоятельно представлять научные результаты, составлять научные документы и отчеты	ОПК-3.1.	Знает основные требования к научным документам и отчетам; стандарты оформления и представления научных результатов
		ОПК-3.2.	Умеет представлять научные результаты, составлять научные документы и отчеты
		ОПК-3.3.	Имеет практический опыт выступлений и научной аргументации в профессиональной деятельности
ОПК-4.	Способен находить, анализировать, реализовывать программно и использовать на практике математические алгоритмы, в том числе с применением современных вычислительных систем	ОПК-4.1.	Знает базовые основы современного математического аппарата, связанного с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности
		ОПК-4.2.	Умеет использовать этот математический аппарат в профессиональной деятельности
		ОПК-4.3.	Имеет практический опыт применения современного математического аппарата, связанного с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в Электронный документ различных областях человеческой деятельности

3. Тематический план

№п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Контактная работа		Контроль	Самостоятельная работа	
		Лекции	Семинары (практические занятия)			
1	Проблема гонки данных и простая синхронизация с помощью мьютекса	7	7		31	Домашние задания, Контрольная работа
2	Архитектурные аспекты многопоточной синхронизации	8	8		32	Домашние задания, Контрольная работа
3	Классические механизмы синхронизации	8	8		32	Домашние задания, Контрольная работа
4	Механизмы синхронизации для высоконагруженных систем	7	7		31	Домашние задания, Контрольная работа
	Экзамен			4		
	Итого:	***30***	***30***	***4***	***126***	
Объем дисциплины (модуля) (в ак. ч.)		***190***				
Объем дисциплины (модуля) (в зач. ед.)		***5***				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Проблема гонки данных и простая синхронизация с помощью мьютекса	Проблема гонки данных. Ошибки в параллельной разработке. CAS-операции и атомики. Мьютексы.
2	Архитектурные аспекты многопоточной синхронизации	Модель памяти. Профилирование и тестирование
3	Классические механизмы синхронизации	Условные переменные. Семафоры.
4	Механизмы синхронизации для высоконагруженных систем	Неблокирующие структуры данных. Lock-Free и Wait-Free алгоритмы. Асинхронный ввод-вывод. Green Threads. Корутины. Структурные шаблоны разработки.

5. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый студент в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Студентам обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

Основная литература:

1. Уильямс, Э. C++. Практика многопоточного программирования : практическое руководство / Э. Уильямс. - Санкт-Петербург : Питер, 2020. - 640 с. - (Серия «Для профессионалов»). - ISBN 978-5-4461-0831-2. - Текст : электронный. - URL: https://znanium.com/catalog/product/1733519.

2. Цукалос, М. Golang для профи: работа с сетью, многопоточность, структуры данных и машинное обучение с Go : практическое руководство / М. Цукалос. - Санкт-Петербург : Питер, 2021. - 720 с. - (Серия «Для профессионалов»). - ISBN 978-5-4461-1617-1. - Текст : электронный. - URL: https://znanium.ru/catalog/product/1733707.

3. Уильямс, Э. Параллельное программирование на C++ в действии. Практика разработки многопоточных программ : практическое руководство / Э. Уильямс ; пер. с англ. А. А. Слинкина. - 2-е изд. - Москва : ДМК Пресс, 2023. - 673 с. - ISBN 978-5-89818-319-6. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2102606.

4. Клири, С. Конкурентность в C#. Асинхронное, параллельное и многопоточное программирование : практическое руководство / С. Клири. - 2-е межд. изд. - Санкт-Петербург : Питер, 2020. - 272 с. - (Серия «Для профессионалов»). - ISBN 978-5-4461-1572-3. - Текст : электронный. - URL: https://znanium.com/catalog/product/1756127.

Дополнительная литература:

1. Maurice Herlihy, Nir Shavit. The Art of Multiprocessor Programming. 2008 - ISBN 978-0-12-370591-4.

2. Allen B. Downey. The Little Book of Semaphores. Second Edition. - 2016 - 291 p. - ISBN 978-1441418685.

3. Огнева, М. В. Программирование на языке C++: практический курс : учебник для вузов / М. В. Огнева, Е. В. Кудрина, А. А. Казачкова. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 342 с. — (Высшее образование). — ISBN 978-5-534-18949-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/563618.

4. Уильямс Э. Параллельное программирование на C++ в действии. Практика разработки многопоточных программ. Пер. с англ. Слинкин А. А. – М.: ДМК Пресс, 2012. – 672с.: ил. – ISBN 978-5-94074-448-1.

6. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и

индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных:

— столами и стульями;
— компьютерной техникой;
— механическими калькуляторами;
— специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

№	Наименование портала (издания, курса, документа)	Ссылка
1.	Научная электронная библиотека elibrary.ru библиотека	https://elibrary.ru/defaultx.asp
2.	База данных для IT-специалистов	https://habr.com
3.	База данных ScienceDirect	https://www.sciencedirect.com
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
5.	Федеральный портал «Российское образование"	https://www.edu.ru/
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
7.	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
8.	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

Наименование ПО	Производство	Лицензионное / свободно распространяемое
Операционные системы:		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
Браузеры:		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
Офисные приложения:		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
Программное обеспечение для планирования и учета времени:		

Toggle app	зарубежное	свободно распространяемое
Системы управления проектами:		
Microsoft Imagine (Project)	зарубежное	лицензионное
Системы управления базами данных:		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
Системы резервного копирования (backup):		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
Справочно-правовые системы:		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
Средства антивирусной защиты:		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
Среды разработки:		
Visual Studio Code	зарубежное	свободно распространяемое
Bash (Unix shell)	зарубежное	свободно распространяемое
Anaconda	зарубежное	свободно распространяемое
Robotic Operating System	зарубежное	свободно распространяемое
CopelliaSim	зарубежное	свободно распространяемое
Google Colaboratory	зарубежное	свободно распространяемое
Пакеты программных средств и библиотек:		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
Системы управления библиографической информацией:		
Zotero	зарубежное	свободно распространяемое
Сервисы и службы:		
Bind	зарубежное	свободно распространяемое
Docker	зарубежное	свободно распространяемое

7. Методические и оценочные материалы

Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Многопоточная синхронизация» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, домашние задания и контрольные работы, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Лекция – систематическое, последовательное, монологическое изложение преподавателем учебного материала, как правило, теоретического характера.

В процессе лекций рекомендуется вести конспект лекций: кратко и схематично фиксировать основные идеи, выводы и обобщения лекции; выделять важные мысли, ключевые слова и термины. Необходимо отметить вопросы или материалы, которые вызывают затруднения, и попытаться найти ответы в рекомендованной литературе. Если разобраться в материале не удается, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

Участие в семинаре (аудиторная работа) – активная работа студента на семинаре, его ответы на вопросы преподавателя и участие в дискуссии.

Для успешного участия в семинаре студентам рекомендуется заранее ознакомиться с темой обсуждения, прочитать необходимые материалы и подготовить вопросы. Важно активно слушать и вовлекаться в дискуссию, высказывая свои мнения и аргументируя их.

При ответах на вопросы преподавателя стоит быть уверенным, четким и логичным, опираясь на изученный материал. Также полезно поддерживать диалог с однокурсниками, чтобы обогатить обсуждение и расширить свои знания.

Домашнее задание – набор задач по темам недели.

При работе над домашними заданиями важно внимательно ознакомиться с требованиями и сроками выполнения. Рекомендуется разбивать задания на этапы, чтобы избежать перегрузки и лучше усвоить материал. Использовать различные источники информации, включая учебники и онлайн-ресурсы, для более глубокого понимания темы.

Контрольная работа – письменная работа с набором задач, которые нужно решить за ограниченное время.

Цель контрольной работы - получить специальные знания по одной или нескольким темам дисциплины (модуля) и продемонстрировать навыки их практического применения.

Самостоятельная работа – работа студентов, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы студенты взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи студента включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов планов и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских заданий и другое.

Система оценивания результатов обучения по дисциплине (модулю)

Критерии получения уровня и оценивания сформированности компетенций по дисциплине (модулю) «Многопоточная синхронизация»

Оценивание уровня учебных достижений, обучающихся по дисциплине (модулю), осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме ***экзамена***, при этом проводится оценка компетенций, сформированных по дисциплине.

Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом:

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину. Он самостоятельно и логически последовательно отвечает на все вопросы, акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать
9	Отлично	
8	Отлично	

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
		теоретические аспекты дисциплины (модуля) с практическими задачами.
7	Хорошо	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
6	Хорошо	
5	Удовлетворительно	Студент обладает базовыми знаниями по дисциплине (модулю), но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
4	Удовлетворительно	
3	Не сдан	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	
1	Не сдан	

Дисциплина (модуль) «Многопоточная синхронизация» оценивается следующим образом:

Активность	Вес	Количество	Описание
Домашние задания	20%	5	Набор задач по темам недели
Контрольные работы	40%	4	Письменная работа с набором задач, которые нужно решить за ограниченное время
Экзамен	40%	1	Письменная или устная работа над заданием, направленным на проверку полученных знаний и навыков по дисциплине (модулю)

Формула расчёта итоговой оценки по дисциплине (модулю) «Многопоточная синхронизация»: «0,2 × среднее за домашние задания + 0,4 × среднее за контрольные работы + 0,4 × экзамен».

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание: Проблема гонки данных

1. Напишите краткое эссе (300-500 слов), в котором объясните, что такое гонка данных, как она возникает и какие последствия могут быть у гонок данных в многопоточных приложениях. Приведите примеры ситуаций, когда гонка данных может произойти.
2. Создайте простую программу на языке C или Python, которая демонстрирует проблему гонки данных. Программа должна использовать несколько потоков, которые одновременно увеличивают общее значение переменной. Запустите программу несколько раз и проанализируйте результаты. Объясните, как гонка данных влияет на итоговое значение.
3. На основе программы, написанной в предыдущем задании, исправьте проблему гонки данных, добавив соответствующие механизмы синхронизации (например, мьютексы или условные переменные). Объясните, как ваши изменения устраняют гонку данных и обеспечивают корректный результат.
4. Предоставьте фрагмент кода (на любом языке программирования), в котором присутствует гонка данных. Попросите однокурсников проанализировать код и выявить проблему гонки данных. Затем напишите комментарии к коду, объясняющие, как и почему возникает гонка данных.
5. Напишите отчет (500-700 слов), в котором исследуются различные методы предотвращения гонок данных, такие как мьютексы, семафоры, условные переменные и атомарные операции. Обсудите преимущества и недостатки каждого метода, а также ситуации, в которых они наиболее эффективны.

Домашнее задание: Профилирование и тестирование

1. Напишите краткое эссе (300-500 слов) о том, что такое профилирование программного обеспечения. Объясните, почему профилирование важно для разработки программ, и какие основные метрики (например, время выполнения, использование памяти) следует учитывать при профилировании.
2. Выберите один из инструментов профилирования (например, gprof для C/C++, cProfile для Python, или VisualVM для Java). Напишите программу, которая выполняет вычисления (например, факториал или сортировку), и используйте выбранный инструмент для профилирования этой программы. Сравните результаты с разными входными данными и проанализируйте, какие части кода требуют оптимизации.
3. Выберите функцию из вашего предыдущего проекта или создайте новую. Напишите набор юнит-тестов для этой функции, используя библиотеку тестирования (например, unittest для Python, JUnit для Java или pytest для C). Объясните, какие сценарии тестирования вы выбрали и почему они важны.
4. Создайте простое веб-приложение (например, на Flask или Express), которое выполняет определенные операции (например, CRUD). Затем проведите нагрузочное тестирование с помощью инструмента (например, Apache JMeter или Locust) и проанализируйте результаты. Напишите отчет о том, как приложение справляется с нагрузкой и какие узкие места были выявлены.
5. Напишите отчет (500-700 слов), в котором сравниваются различные методы тестирования (модульное, интеграционное, системное и приемочное тестирование). Обсудите, какие преимущества и недостатки есть у каждого метода, и в каких ситуациях их следует применять. Приведите примеры из практики.

Домашнее задание: Неблокирующие структуры данных

1. Напишите краткое эссе (300-500 слов), в котором объясните, что такое неблокирующие структуры данных. Опишите их преимущества по сравнению с блокирующими структурами данных и приведите примеры, где они могут быть полезны в многопоточных приложениях.
2. Реализуйте неблокирующий стек на языке программирования по вашему выбору (например, C++, Java или Python) с использованием атомарных операций. Объясните, как ваша реализация обеспечивает безопасность при доступе из нескольких потоков. Протестируйте стек, добавляя и удаляя элементы из нескольких потоков одновременно.
3. Сравните производительность неблокирующей и блокирующей структуры данных (например, стека или очереди) в условиях многопоточности. Напишите программу, в которой используются обе структуры данных для выполнения одинаковых операций, и измерьте время выполнения. Проанализируйте результаты и сделайте выводы о том, какая структура данных более эффективна в вашем тесте.
4. Создайте неблокирующую очередь с использованием алгоритма "lock-free" (например, алгоритм Michael-Scott). Объясните, как ваша реализация обрабатывает ситуации гонки данных и обеспечивает корректность при одновременном доступе из нескольких потоков. Протестируйте очередь, добавляя и извлекая элементы из нескольких потоков.
5. Напишите отчет (500-700 слов), в котором исследуются различные неблокирующие алгоритмы и структуры данных (например, неблокирующие списки, деревья, хеш-таблицы). Обсудите их преимущества и недостатки, а также ситуации, в которых они могут быть более эффективными по сравнению с блокирующими аналогами. Приведите примеры из практики.

Примерные задания для контрольных работ

Контрольная работа 1: Архитектурные аспекты многопоточной синхронизации

1. Определите и объясните следующие термины:
- Взаимное исключение
- Гонки данных
- Мьютексы
- Семафоры
- Условные переменные
2. Опишите проблему философов-ужина и объясните, как она иллюстрирует проблемы многопоточной синхронизации. Предложите один из способов решения этой проблемы и опишите его преимущества и недостатки.
3. Сравните мьютексы и семафоры. В каких ситуациях лучше использовать каждый из этих механизмов? Приведите примеры, когда один метод может быть предпочтительнее другого.
4. Напишите код на языке программирования по вашему выбору, который реализует мьютекс для защиты общего ресурса. Объясните, как ваш код обеспечивает взаимное исключение и предотвращает гонки данных.
5. Опишите, что такое гонка данных и как она может возникнуть в многопоточных приложениях. Приведите пример кода, в котором возникает гонка данных, и объясните, как ее можно исправить с помощью синхронизации.
6. Представьте, что вы разрабатываете многопоточную систему для обработки запросов пользователей в веб-приложении. Опишите архитектурные аспекты, которые необходимо учитывать для обеспечения корректной синхронизации и взаимодействия потоков. Укажите, какие механизмы синхронизации вы бы использовали и почему.
7. Опишите, как использование различных методов синхронизации может повлиять на производительность многопоточной системы. Обсудите компромиссы между

безопасностью и производительностью, а также приведите примеры, когда производительность может быть критически важна.

Контрольная работа 2: Механизмы синхронизации для высоконагруженных систем

1. Опишите, что такое высоконагруженные системы и какие специфические требования к синхронизации возникают в таких системах. Приведите примеры высоконагруженных приложений и объясните, почему эффективная синхронизация критична для их производительности.
2. Сравните различные механизмы синхронизации, такие как мьютексы, семафоры, условные переменные и барьеры. Обсудите их преимущества и недостатки в контексте высоконагруженных систем. Когда и почему вы бы предпочли один механизм другому?
3. Объясните, что такое "узкое место" (bottleneck) в контексте многопоточных приложений. Приведите примеры, как неправильное использование механизмов синхронизации может привести к узким местам, и предложите способы их устранения.
4. Исследуйте неблокирующие структуры данных (например, lock-free и wait-free алгоритмы) и их применение в высоконагруженных системах. Напишите краткий обзор (300-500 слов) о том, как такие структуры данных могут улучшить производительность и уменьшить задержки.
5. Напишите код на языке программирования по вашему выбору для реализации семафора. Объясните, как ваша реализация может быть использована для управления доступом к ресурсу в высоконагруженной системе и как она справляется с возможными проблемами, такими как starvation.
6. Проведите эксперимент, в котором сравните производительность многопоточной программы, использующей блокирующие и неблокирующие механизмы синхронизации. Опишите методологию тестирования, результаты и сделайте выводы о том, какой подход лучше подходит для высоконагруженных систем.
7. Опишите, как механизмы синхронизации применяются в распределенных системах. Обсудите проблемы, такие как согласованность данных и задержки сети, и предложите подходы для эффективного управления синхронизацией в таких системах.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1	Укажите основную проблему, возникающую при одновременном доступе нескольких потоков к разделяемым данным.	проблема гонки данных / race condition	ОПК-3
2	Назовите механизм синхронизации, который предотвращает одновременный доступ к критической секции.	мьютекс	ОПК-3
3	Укажите операцию, которая позволяет атомарно сравнить и обменять значение в памяти.	CAS-операция	ОПК-3
4	Назовите тип ошибки в параллельной разработке, связанный с неправильной синхронизацией.	deadlock	ОПК-3
5	Укажите модель, описывающую поведение памяти в многопоточных программах.	модель памяти	ОПК-3
6	Назовите инструмент для анализа производительности многопоточных программ.	профилирование	ОПК-3
7	Укажите метод тестирования многопоточных приложений на наличие ошибок синхронизации.	тестирование	ОПК-3
8	Назовите механизм синхронизации, позволяющий потокам ожидать определенного условия.	условная переменная	ОПК-3

9	Укажите классический механизм синхронизации для управления доступом к ресурсам.	семафор	ОПК-3
10	Назовите алгоритм, обеспечивающий синхронизацию без блокировок.	Lock-Free алгоритм	ОПК-4
11	Укажите механизм для асинхронной обработки ввода-вывода в высоконагруженных системах.	асинхронный ввод-вывод	ОПК-4
12	Назовите легковесные потоки, управляемые пользовательским пространством.	Green Threads	ОПК-4
13	Укажите функцию, позволяющую приостанавливать и возобновлять выполнение в многопоточной среде.	корутина	ОПК-4
14	Назовите шаблон разработки для организации конкурентного доступа к данным.	структурный шаблон	ОПК-4
15	Укажите тип структуры данных, обеспечивающей неблокирующий доступ.	неблокирующая структура данных	ОПК-4
16	Назовите алгоритм, гарантирующий прогресс для всех потоков без ожидания.	Wait-Free алгоритм	ОПК-4
17	Укажите проблему, возникающую при циклической зависимости блокировок.	deadlock	ОПК-4
18	Назовите механизм для атомарных операций в многопоточной синхронизации.	атомик	ОПК-4
19	Укажите условие, при котором потоки могут бесконечно ожидать ресурса.	starvation	ОПК-4
20	Назовите аспект многопоточной синхронизации, связанный с видимостью изменений в памяти.	модель памяти	ОПК-4