

УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«07» марта 2024 г.
Протокол №1

**Рабочая программа дисциплины (модуля)
«Многопоточная синхронизация»**

Направление подготовки: 02.03.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Искусственный интеллект

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Срок освоения программы: 4 года

Год набора: 2024

**Москва
2024**

Содержание

1. Краткая характеристика дисциплины (модуля)	3
2. Перечень планируемых результатов обучения	4
3. Тематический план	4
4. Содержание дисциплины (модуля)	6
5. Учебно-методическое обеспечение	7
6. Материально-техническое обеспечение	7
7. Методические и оценочные материалы	9

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 Искусственный интеллект, 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 Искусственный интеллект и входит в часть Блока 1, формируемую участниками образовательных отношений, как дисциплина по выбору.

Дисциплина (модуль) изучается на 2 или 3 курсе в 4 или 5 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):

- освоение знания и понимания распространенных проблем гонки данных, устройства примитивов синхронизации;
- формирование умений применения механизмов синхронизации, реализации структуры данных и алгоритмов, используемых в многопоточном программировании;
- формирование навыков 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)
УК-1.	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1.	Знает методы поиска и анализа информации в области искусственного интеллекта, основные принципы критической оценки источников информации и их релевантности
		УК-1.2.	Умеет критически оценивать источники информации и синтезировать данные из различных источников для решения задач, применять системный подход к анализу и решению комплексных проблем
		УК-1.3.	Имеет практический опыт работы с современными инструментами и технологиями для обработки информации, формулировании и структурировании задач на основе полученной информации
ОПК-1.	Способен консультировать и использовать фундаментальные знания в области математического анализа, комплексного и функционального анализа алгебры, аналитической геометрии, дифференциальной геометрии и топологии, дифференциальных уравнений, дискретной математики и математической логики, теории вероятностей, математической статистики и случайных процессов, численных методов, теоретической механики в профессиональной деятельности	ОПК-1.1.	Знает основные концепции и теории в области математического анализа и смежных дисциплин; методы и подходы, используемые в различных областях математики
		ОПК-1.2.	Умеет применять математические методы для решения профессиональных задач
		ОПК-1.3.	Имеет практический опыт разработки и реализации математических моделей в профессиональной деятельности
ОПК-6	Способен разрабатывать алгоритмы и компьютерные программы, пригодные	ОПК-6.1.	Знает алгоритмы разработки, компьютерные программы, а также алгоритмы вычислительной математики в области

	для практического применения		искусственного интеллекта
		ОПК-6.2.	Умеет разрабатывать математические программные продукты и комплексы с использованием современных технологий программирования в области искусственного интеллекта
		ОПК-6.3.	Имеет практический опыт разработки интеллектуальных информационных систем для визуализации результатов исследований в области искусственного интеллекта
ПК-1.	Способен формулировать задачи с математической точностью, обосновывать утверждения строго и анализировать полученные результаты в области математики и компьютерных наук	ПК-1.1.	Знает методы и подходы к формулированию задач, а также основные принципы математического доказательства и анализа результатов
		ПК-1.2.	Умеет корректно ставить и формулировать математические задачи, применять строгие методы доказательства и анализировать полученные результаты
		ПК-1.3.	Имеет опыт работы с задачами в области математики и компьютерных наук, включая применение математических методов для решения практических задач

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		32	Проекты
2	Архитектурные аспекты многопоточной синхронизации	7	7		32	Проекты
3	Классические механизмы синхронизации	7	7		32	Проекты
4	Механизмы синхронизации для высоконагруженных систем	7	7	2	32	Проекты Защита проекта
	<i>Зачет с оценкой</i>			4		
	Итого:	28	28	6	128	
	<i>Объем дисциплины (модуля) (в ак. ч.)</i>	190				
	<i>Объем дисциплины (модуля) (в зач. ед.)</i>	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. С++. 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 С++ 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 С++: 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 С++ 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, проекты, 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 удастся, 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.

Проект – исследовательская работа по курсу и презентация результатов.

Для успешной подготовки к проекту рекомендуется: четко определить цели и задачи проекта; составить план работы, разбив проект на этапы с указанием сроков выполнения каждого из них; использовать разнообразные источники информации и инструменты для исследования темы; регулярно проверять прогресс и вносить коррективы в план, если это необходимо.

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 *зачета с оценкой*, 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	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	Зачтено	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
9	Отлично	Зачтено	
8	Отлично	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
7	Хорошо	Зачтено	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	Зачтено	
5	Удовлетворительно	Зачтено	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	Зачтено	
3	Не сдан	Не зачтено	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	Не зачтено	
1	Не сдан	Не зачтено	

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

Активность	Вес	Количество	Описание
Проекты	30%	3	Исследовательская работа по курсу и презентация результатов.

Активность	Вес	Количество	Описание
Защита проекта	40%	1	Презентация итогового проекта по дисциплине (модулю)
Зачет с оценкой	30%	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»: $\langle 0,3 \times \text{среднее за проекты} + 0,4 \times \text{среднее за защиту проекта} + 0,3 \times \text{зачет с оценкой} \rangle$.

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные задания для проектов

Проект 1. «Гонки данных и корректная синхронизация: банк транзакций»

Тема: Проблема гонок/проблема гонок данных; ошибки параллельной разработки; CAS/атомики; мьютексы.

Задание

Реализовать многопоточное приложение-симулятор банковских счетов:

- Есть счетов (балансов) и поток транзакций (переводы между счетами). **N**
- Параллельно выполняются работники, каждый оформляет денежные переводы. **T**
- Требования:

1. Гарантировать **корректность балансов** ((суммарный баланс системы постоянный, нет «пропавших»/«созданных» денег).

2. **2 реализации синхронизации:** Предусмотреть как минимум Транспорт:

- вариант А: мьютексы (например, блокировка на счет/шард/таблицу);
- вариант Б: атомики/CAS (где применяется; термин гибрид).
- 3. **филд проверяет инварианты** и режим «стресс-тест» (большие Н/Т/операций).

4. Собрать **метрики производительности** : пропускная способность, задержка (p50/p95), время выполнения, CPU.

Можно делать на C++/Rust/Go/Java (на выбор).

Этапы подготовки проекта

1. Постановка моделей данных и инвариантов

- формальное описание того, что считается «корректно» (инварианты), и где могут быть гонки.

2. Наивная (сломанная) версия

- версия без синхронизации или с заведомой ошибкой (для мобильных гонок).

3. Вариант с мьютексами

- выбрать переключатель: глобальная блокировка, блокировка для каждой учетной записи, упорядочивание блокировок (выход из взаимоблокировки).

4. Вариант с CAS/атомиками

- Проработать, что реально можно сделать lock-free (например, написание/зачисление через CAS-циклы, либо очередь транзакций lock-free + защищенные балансы).

5. Тестирование и стресс

- юнит-тесты на корректность, стресс-тесты на гонках, ошибки.

6. Варианты профилирования и сравнения

- Сравните А и В по метрикам и сделайте выводы.

7. Отчет + воспроизводимость

- README: как собрать/запустить, параметры, ожидаемые результаты.

Критерии измерения (10 баллов)

1. **Корректность (инварианты, отсутствие потерь денег, отсутствие гонки данных)** — 0–4
2. **Две реализации синхронизации + осмысленный выбор стратегии** — 0–2
3. **Тесты/стресс/демонстрация найденной гонки в наивной версии** — 0–2 — 0–2
4. **Профилирование и сравнение (графики/таблицы, выводы)** — 0–2

Критерии защиты

- Показать на защите:
 1. запуск наивной версии, где периодически «ломается» инвариант;
 2. запуск корректных версий и проверка инвариантов;
 3. мёртвый, **какая именно гонка была** и как устранена; и как адаптировано;
 4. почему выбранная CAS-логика корректна (где точка линеаризации/почему нет АВА-проблем, если актуально);
 5. Результаты профилирования: где узкое место, как меняется масштабирование при росте потоков.

Проект 2. «Производитель–Потребитель: условные переменные и семафоры + ограниченный буфер»

Тема: условные переменные; семафоры; синхронизация потоков.

Задание

Реализовать систему обработки задач:

- Несколько продюсеров-потоков генерируют задачи (например, строки текста/числа/файлов).
- Несколько потребительских потоков обрабатывают задачи (например, парсинг, хэширование, компрессия).
- Междурез — **ограниченный буфер** фиксированного размера (фиксированного размера (очередь)).
- Требования:
 1. **двумя способами** :Реализовать ограниченный буфер :
 - вариант А: **mutex + condition_variable**, `(две условия: ,); **not_empty not_full**
 - вариант Б: семафоры (например, и) + мьютекс на структуру данных. **itemsslots**
 2. Поддержите : правильное завершение Consumer'ов, когда производителя закончат.
 3. Измерьте производительность разных размеров буфера и количества потоков.

Этапы подготовки проекта

1. Спецификация интерфейса очереди: **pushpop**, сем, , семантика блокировок. **close**
2. Реализация А (condvar): корректная работа с «ложными пробуждениями», while-проверки.
3. Реализация В (семафор): корректное комбинирование семафоров и мьютекса.
4. Тесты:
 - финансовые (FIFO/не теряем задачи),
 - стресс (много потоков, небольшой буфер),
 - завершить (закрыть).
5. Бенчмарки: подобрать пейзажи, собрать таблицы/графики.
6. Итоговый отчет: когда какая реализация лучше и почему.

Критерии измерения (10 баллов)

1. — 0–4
2. **Две реализации (condvar и семафоры) с одинаковой внешней семантикой** — 0–2
3. **Корректное завершение (закрытие/остановка) и обработка краевых случаев** — 0 — 0–2
4. **Бенчмарки и выводы (влияние размера буфера, кол-ва потоков)** — 0–2

Критерии защиты

- В защиту соблюдения:

1. работа с маленьким буфером (1–2 элемента) без зависимостей;
2. корректное завершение всех потоков;
3. объяснить, почему в condvar-версии условия проверки делаются в **while** ;, а не ; **if**
4. объяснить, почему семафоры правильно ограничивают очередь; **items/slots**
5. производительность и сделать инженерный вывод.

Проект 3. «Неблокирующая структура данных: lock-free стек/очередь + нагрузка»

Тема: неблокирующая/неблокирующая структура данных; без блокировки/без ожидания; КАС; (опционально) корутины/зеленые нитки.

Задание

Реализовать и учитывать неблокирующую структуру данных по выбору:

- Международный 1: **Бесблокировочный стек (Treiber)**
- Единица 2: **Очередь без блокировки (Майкл – Скотт)** (сложнее/сложный)

Требования:

1. Реализовать структуру на атомике/CAS (без мьютекса в критическом пути).
2. **АВА** Предусмотреть проблему (любой приемлемый способ): помеченные указатели, указатели опасностей, восстановление по эпохам, запрет остановки узлов в тестовом режиме (минимум) + объяснение запрета.
3. Делаем тест «много производителей/потребителей» + проверка корректности (не теряем элементы, порядок — согласно выбранной стадии).
4. Сравнить с рейтингом мьютекса (базовый уровень).

Этапы подготовки проекта

1. точка линеаризации Выбор структуры (стек/очередь) и описание линейризуемости (где точка линеаризации).
2. Замок РеалиРеализация lock-free алгоритма (CAS-циклы).
3. Управление памятью/АВА (минимально работоспособный подход + обоснование).
4. Тестирование: функциональное + стресс + рандомизированные тесты.
5. Бенчмарки: сравнение lock-free и мьютекса при разных потоках/нагрузках.
6. Документация: ограничения, которые гарантируются (без блокировки или без ожидания), где возможна деградация.

Критерии измерения (10 баллов)

1. **Корректность (линейризуемость на тестах, отсутствие потерь/дубликатов)** — 0–4
2. **Реальная неблокирующая выдача (CAS, без скрытой блокировки блокировки)** — 0 — 0–2
3. **Решение/обсуждение АВА и освобождение памяти** — 0–2
4. **Сравнение с базовым уровнем мьютекса + выводы по производительности** — 0–2 — 0–2

Критерии защиты

- Показать:

1. тесты, которые «ломают» наивную доставку (если есть) и передают в вашу страну;
2. объяснение точки линеаризации и почему операции корректны при гонках;
3. что такое АВА именно в вашем коде и насколько вы ограничены в риске;
4. результаты бенчмарков (когда lock-free быстрее/медленнее);
5. Какие гарантии дают вашу структуру (без блокировки, но не без ожидания) и что это означает.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Что из следующего является основным источником гонки данных? а) Неправильное использование мьютексов б) Одновременный доступ нескольких потоков к общему ресурсу в) Использование атомарных операций г) Неправильная реализация CAS-операций	б	ОПК-6
2.	Какой механизм синхронизации позволяет потокам ожидать изменения состояния переменной? а) Семафор б) Условная переменная в) Мьютекс г) CAS-операция	б	ОПК-6
3.	Какой из следующих алгоритмов не блокирует потоки и предоставляет доступ к данным? а) Мьютекс б) Семафор в) Lock-Free алгоритм г) Wait-Free алгоритм	в	ОПК-1
4.	Как называется метод, который позволяет избежать гонок данных при параллельном доступе к ресурсам?	Мьютекс	ОПК-1
5.	Определение порядка операций в многопоточной среде	Модель памяти	ОПК-6
6.	Какой механизм синхронизации используется для ограничения количества потоков, получающих доступ к ресурсу?	Семафор	ОПК-1
7.	Как называется подход, который использует асинхронный ввод-вывод для повышения производительности?	Асинхронный ввод-вывод	ПК-1
8.	Какой тип ошибок часто возникает в результате неправильного использования мьютексов?	Deadlock (взаимная блокировка)	УК-1
9.	Какой механизм синхронизации позволяет потокам сигнализировать друг другу о том, что они могут продолжить выполнение?	Условная переменная	ПК-1
10.	Как называются алгоритмы, которые гарантируют, что каждый поток завершит свою работу за конечное время?	Wait-Free алгоритмы	ПК-1
11.	Какой термин описывает состояние, когда один поток удерживает ресурс, ожидая другой ресурс, удерживаемый другим потоком?	Взаимная блокировка (deadlock)	УК-1
12.	Какой метод используется для профилирования многопоточных приложений?	Профилер производительности	ОПК-1