

**УТВЕРЖДЕНА**

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

**Рабочая программа дисциплины (модуля)  
«Архитектура компьютера и операционные системы 2»**

**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:** 2025

**Москва  
2025**

## Содержание

<b>1. Краткая характеристика дисциплины (модуля)</b> .....	<b>3</b>
<b>2. Перечень планируемых результатов обучения</b> .....	<b>5</b>
<b>3. Тематический план</b> .....	<b>7</b>
<b>4. Содержание дисциплины (модуля)</b> .....	<b>7</b>
<b>5. Учебно-методическое обеспечение</b> .....	<b>8</b>
<b>6. Материально-техническое обеспечение</b> .....	<b>8</b>
<b>7. Методические и оценочные материалы</b> .....	<b>10</b>

## 1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Архитектура компьютера и операционные системы 2» 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) «Архитектура компьютера и операционные системы 2» позволяет студентам понять, как функционируют компьютерные системы, включая взаимодействие аппаратного и программного обеспечения, что критично для оптимизации производительности и разработки эффективных приложений. Это знание поможет в профессиональной деятельности эффективно управлять ресурсами, обеспечивая стабильность и безопасность вычислительных систем.

### 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 вариативную часть Блока 1, формируемую участниками образовательных отношений.

Дисциплина (модуль) является выборной и доступна для изучения на 3 или 4 курсе в 4, 5, 6, 7 или 8 семестре на выбор. Доступна после успешном освоения дисциплины (модуля) «Архитектура компьютеров и операционные системы»

**Цель изучения дисциплины (модуля):** заключается в формировании у студентов фундаментальных знаний об устройстве работы компьютера и операционных систем, которые необходимы для эффективной разработки программного обеспечения, а также для понимания более сложных аспектов компьютерных наук, таких как распределенные системы, виртуализация и кибербезопасность.

### Задачи изучения дисциплины (модуля):

— формирование знания о устройстве командной системы центрального процессора, устройстве операционных систем, взаимодействие прикладных программ с операционными системами и основы сетевого взаимодействия;

— формирование умения вручную управлять памятью на языке системного программирования Си и языках ассемблера;

— формирование умения эффективно взаимодействовать с операционной системой при реализации прикладных программ;

— формирование умения создавать программы без использования стандартной и сторонних библиотек;

— формирование умения реализовывать низкоуровневые сервисы и их взаимодействие между собой.

### В результате освоения дисциплины (модуля) обучающийся должен:

#### **знать:**

— устройство командной системы центрального процессора, основные типы команд и их назначение;

— архитектуру операционных систем, включая управление процессами, памятью и вводом-выводом;

— принципы взаимодействия прикладных программ с операционной системой и базовые механизмы системных вызовов;

— основы сетевого взаимодействия на уровне операционных систем и протоколов передачи данных;

#### **уметь:**

- вручную управлять памятью (выделять, освобождать, адресовать) с использованием языка С и ассемблера;
- эффективно взаимодействовать с операционной системой при разработке прикладных программ, используя системные вызовы и API;
- создавать программы без использования стандартных или сторонних библиотек, реализуя базовые функции самостоятельно;
- реализовывать низкоуровневые сервисы и обеспечивать их взаимодействие между собой на уровне операционной системы;

***владеть:***

- навыками программирования на языках системного уровня (Си и ассемблер) для реализации системных и прикладных задач;
- методами отладки и анализа работы программ, взаимодействующих с операционной системой и аппаратным обеспечением;
- технологиями разработки и интеграции низкоуровневых сервисов и драйверов устройств;
- способностью проектировать и реализовывать собственные механизмы управления ресурсами и взаимодействия процессов.

## 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 разработки, 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 для практического применения	ОПК-6.1.	Знает алгоритмы разработки, компьютерные программы, а также алгоритмы вычислительной математики в области 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.	Обладает базовыми знаниями, полученными в области математических наук, программирования и информационных технологий
		ПК-1.2.	Умеет находить, формулировать и решать стандартные задачи в собственной научно-исследовательской деятельности в математике и информатике
		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)
		<i>Очная форма</i>					
		Контактная работа			Контроль	Самостоятельная работа	
Лекции	Семинары	Консультации					
1	Базовые системные вызовы UNIX	7	7	2		30	Домашние задания, Контрольная работа
2	Процессы	8	8	2		30	Домашние задания, Контрольная работа
3	Межпроцессорное взаимодействие	8	8	2		30	Домашние задания, Контрольная работа
4	Средства реализации высоконагруженных систем	7	7			30	Домашние задания, Контрольная работа
	<i>Зачет с оценкой</i>				4		
	<b>Итого:</b>	<b>30</b>	<b>30</b>	<b>6</b>	<b>4</b>	<b>120</b>	
	<b>Объем дисциплины (модуля) (в ак. ч.)</b>	<b>190</b>					
	<b>Объем дисциплины (модуля) (в зач. ед.)</b>	<b>5</b>					

### 4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Базовые системные вызовы UNIX	Файловые дескрипторы. Атрибуты файлов. Отображение файлов на память. Файловые системы FUSE
2	Процессы	Запуск и завершение процессов. Выполнение внешних программ. Каналы между процессами. Лимиты и изоляция процессов
3	Межпроцессорное взаимодействие	Сигналы. Сигналы – продвинутый уровень. Локальные и TCP сокеты. Основы сетевого взаимодействия
4	Средства реализации высоконагруженных систем	Мультиплексирование ввода-вывода. Многопоточность в POSIX. BPF и eBPF

## 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. Таненбаум, Э. С. Компьютерные сети / Э. С. Таненбаум, Д. Уэзеролл. - 5-е изд. - Санкт-Петербург : Питер, 2021. - 960 с. - ISBN 978-5-4461-1248-7. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2141427>.

2. Букатов, А. А. Компьютерные сети: расширенный начальный курс : учебник для вузов / А. А. Букатов, С. А. Гуда. - Санкт-Петербург : Питер, 2020. - 496 с. - (Серия «Учебник для вузов»). - ISBN 978-5-4461-1338-5. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1756115>.

3. Рабчевский, А. Н. Компьютерные сети и системы связи. Вводный курс : учебное пособие для вузов / А. Н. Рабчевский. — Москва : Издательство Юрайт, 2025. — 207 с. — (Высшее образование). — ISBN 978-5-534-21489-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/572633>.

4. Компьютерные сети : учебник и практикум для вузов / под научной редакцией А. М. Нечаева, А. Е. Трубина, А. Ю. Анисимова. — Москва : Издательство Юрайт, 2025. — 515 с. — (Высшее образование). — ISBN 978-5-534-21452-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/572239>.

### *Дополнительная литература:*

1. Амини Камран. Экстремальный Си. Параллелизм, ООП и продвинутые возможности. — СПб.: Питер, 2021. — 752 с. — ISBN 978-5-4461-1694-2

2. Керриск Майкл. Linux API. Исчерпывающее руководство. — СПб.: Питер, 2019. — 1248 с. — ISBN 978-5-4461-0985-2

3. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. — СПб.: Питер, 2022. — 1120 с. — ISBN 978-5-4461-1155-8

## 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	<a href="https://elibrary.ru/defaultx.asp">https://elibrary.ru/defaultx.asp</a>
2.	База данных для IT-специалистов	<a href="https://habr.com">https://habr.com</a>
3.	База данных ScienceDirect	<a href="https://www.sciencedirect.com">https://www.sciencedirect.com</a>
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	<a href="https://minobrnauki.gov.ru/">https://minobrnauki.gov.ru/</a>
5.	Федеральный портал «Российское образование»	<a href="https://www.edu.ru/">https://www.edu.ru/</a>
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	<a href="http://window.edu.ru/">http://window.edu.ru/</a>
7.	Единая коллекция цифровых образовательных ресурсов	<a href="http://school-collection.edu.ru/">http://school-collection.edu.ru/</a>
8.	Федеральный центр информационно - образовательных ресурсов	<a href="http://fcior.edu.ru/">http://fcior.edu.ru/</a>

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
<b>Операционные системы:</b>		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
<b>Браузеры:</b>		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
<b>Офисные приложения:</b>		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
<b>Программное обеспечение для планирования и учета времени:</b>		
Toggle app	зарубежное	свободно распространяемое
<b>Системы управления проектами:</b>		
Microsoft Imagine (Project)	зарубежное	лицензионное
<b>Системы управления базами данных:</b>		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
<b>Системы резервного копирования (backup):</b>		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
<b>Справочно-правовые системы:</b>		

КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
<b>Средства антивирусной защиты:</b>		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
<b>Среды разработки:</b>		
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
<b>Пакеты программных средств и библиотек:</b>		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
<b>Системы управления библиографической информацией:</b>		
Zotero	зарубежное	свободно распространяемое
<b>Сервисы и службы:</b>		
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) «Архитектура компьютера и операционные системы 2» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, консультации, 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 удастся, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

*Семинар* — это форма учебной деятельности, проводимая в учебном заведении под руководством преподавателя, где студенты активно участвуют в обсуждениях, практических заданиях и других формах взаимодействия.

Для успешной подготовки к семинару рекомендуется заранее ознакомиться с темой занятия и основными материалами, чтобы иметь возможность активно участвовать в обсуждении. Также полезно подготовить вопросы и идеи для обсуждения, что поможет глубже понять материал и продемонстрировать заинтересованность.

*Консультации* – структурированные встречи, на которых преподаватели предоставляют индивидуальную или групповую помощь в освоении учебного материала, обсуждении вопросов и решении проблем, возникающих в процессе обучения.

Консультации могут включать разъяснение сложных тем, подготовку к экзаменам и помощь в выполнении проектных работ, что способствует более глубокому пониманию

предмета и улучшению академической успеваемости.

*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) «Архитектура компьютера и операционные системы 2»**

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 оценками*, 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
9	Отлично	Зачтено	
8	Отлично	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			задачи. Студент хорошо знаком с 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) «Архитектура компьютера и операционные системы 2» оценивается следующим образом:

Активность	Вес	Количество	Описание
Домашние задания	20%	10	Набор задач по темам недели

Активность	Вес	Количество	Описание
Контрольные работы	30%	3	Письменная работа с набором задач, которые нужно решить за ограниченное время
Зачет с оценкой	50%	1	Письменная или устная работа над заданием, направленным на проверку полученных знаний и навыков по курсу

**Формула расчёта итоговой оценки по дисциплине (модулю) «Архитектура компьютера и операционные системы 2»:** « $0,2 \times$  среднее за домашние задания +  $0,3 \times$  среднее за контрольные работы +  $0,5 \times$  зачет с оценкой».

### Текущий контроль успеваемости обучающихся по дисциплине (модулю)

#### Примерные домашние задания

##### Домашнее задание 1: Мультиплексирование ввода-вывода

1. Напишите программу, которая использует `select()` для мониторинга нескольких файловых дескрипторов (например, `stdin` и `pipe`) и обрабатывает ввод асинхронно.
2. Создайте простой `echo`-сервер с `select()`: сервер принимает соединения от нескольких клиентов и эхом возвращает сообщения без блокировок.
3. Реализуйте программу, которая использует `poll()` для мониторинга TCP-сокета и файла, выводя события (`POLLIN`, `POLLOUT`).
4. Напишите программу, которая сравнивает производительность `select()` и `poll()` на 100 дескрипторах, измеряя время отклика.
5. Создайте многоклиентный чат-сервер с `select()`: клиенты подключаются, отправляют сообщения, сервер рассылает всем.

##### Домашнее задание 2: Многопоточность в POSIX

1. Напишите программу на C с POSIX threads (`pthread`), которая создает два потока: один считает сумму массива, другой - произведение.
2. Создайте программу, которая использует `pthread_mutex` для синхронизации доступа к общему счетчику между двумя потоками.
3. Реализуйте `producer-consumer` модель с `pthread_cond`: один поток производит данные, другой потребляет, используя `condition variables`.
4. Напишите многопоточный сервер: главный поток принимает соединения, рабочие потоки обрабатывают клиентов с помощью `thread pool`.
5. Создайте программу, которая демонстрирует `deadlock`: два потока пытаются захватить два `mutex` в разном порядке. Объясните и исправьте.

##### Домашнее задание 3: BPF и eBPF

1. Установите `bpftool` и напишите простой BPF-программу, которая фильтрует сетевые пакеты по порту (например, блокирует порт 80) с помощью `tc`.
2. Создайте eBPF-программу, которая отслеживает системные вызовы (например, `open`) с помощью `tracerepoint` и выводит логи в `/sys/kernel/debug/tracing/trace`.
3. Напишите eBPF-программу для мониторинга производительности: отслеживайте время выполнения функции с помощью `kprobe` и сохраняйте в `map`.

4. Реализуйте eBPF-фильтр для сокетов: программа, которая блокирует соединения на определенный IP-адрес с помощью SO\_ATTACH\_BPF.

5. Создайте комплексную eBPF-программу: комбинируйте tracerpoint и map для подсчета количества вызовов exesce() по PID. Анализируйте результаты с bpftool.

### Примерные задания для контрольных работ

#### Контрольная работа 1

1. Объясните, что такое файловый дескриптор в UNIX. Приведите пример системного вызова, который возвращает файловый дескриптор, и опишите, как его использовать для чтения файла.

2. Напишите фрагмент кода на C, который открывает файл "test.txt" с помощью open(), проверяет атрибуты файла с помощью stat() и выводит размер файла и права доступа в восьмеричном формате.

3. Опишите принцип работы mmap() для отображения файла на память. Напишите код, который отображает файл на память только для чтения, читает первые 10 байт и корректно освобождает память.

4. Что такое FUSE? Напишите простой пример кода FUSE, реализующий операцию getattr для виртуального файла, который всегда возвращает фиксированный размер 100 байт.

5. Сравните производительность чтения файла через read() и через mmap(). Приведите аргументы за и против каждого метода, основываясь на типичных сценариях использования.

6. Объясните разницу между fork() и exesce() в контексте создания процессов. Напишите код, который создает дочерний процесс с помощью fork() и заменяет его образ на программу "ls" с помощью exesce1().

7. Опишите, как работает системный вызов wait(). Напишите программу, которая создает два дочерних процесса, каждый из которых выполняет sleep(2), и родитель ждет их завершения, выводя PID и код выхода каждого.

8. Что такое pipe в UNIX? Напишите код, который создает pipe, в родительском процессе пишет в него строку "Hello", а в дочернем читает и выводит ее на экран.

9. Объясните понятие лимитов процессов (resource limits). Напишите код, который получает текущий лимит на количество открытых файлов (RLIMIT\_NOFILE) с помощью getrlimit() и выводит его.

10. Опишите механизм изоляции процессов с помощью chroot(). Приведите пример сценария, где это полезно, и напишите код, который изменяет корневую директорию процесса на "/tmp/jail" и выполняет команду "ls".

#### Контрольная работа 2

1. Объясните, что такое сигнал в UNIX. Напишите код, который устанавливает обработчик для сигнала SIGINT с помощью signal(), который выводит "Interrupted" и завершает программу.

2. Опишите разницу между signal() и sigaction(). Напишите пример использования sigaction() для обработки SIGUSR1 с флагом SA\_RESTART.

3. Что такое Unix domain socket? Напишите простой клиент-серверный код: сервер создает локальный сокет, принимает соединение и читает сообщение от клиента.

4. Объясните основы TCP-сокетов. Напишите код TCP-сервера, который слушает порт 8080, принимает соединение, читает строку от клиента и отправляет ее обратно (эхо).

5. Опишите функцию getaddrinfo(). Напишите код, который использует getaddrinfo() для разрешения имени "localhost" и устанавливает TCP-соединение к порту 80.

6. Объясните принцип работы `select()` для мультиплексирования ввода-вывода. Напишите код, который использует `select()` для мониторинга `stdin` и TCP-сокета, выводя данные при их готовности.

7. Что такое `deadlock` в многопоточности? Напишите код с двумя потоками и двумя `mutex`, который демонстрирует `deadlock`, и предложите способ его избежать.

8. Опишите `producer-consumer` модель с использованием `POSIX threads`. Напишите код, реализующий эту модель с `pthread_mutex` и `pthread_cond` для буфера размером 10 элементов.

9. Что такое `BPF` и `eBPF`? Напишите простой пример `eBPF`-программы, которая отслеживает системный вызов `open` с помощью `tracerepoint` и логирует `PID` процесса.

10. Сравните `select()` и `poll()` для мультиплексирования. Напишите код, который использует `poll()` для мониторинга нескольких дескрипторов и обрабатывает события `POLLIN`.

### Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1	Укажите основной способ идентификации открытых файлов в UNIX-системах.	файловый дескриптор	УК-1
2	Назовите атрибут файла, определяющий его размер в байтах.	размер файла	УК-1
3	Укажите механизм отображения содержимого файла в адресное пространство процесса.	<code>mmap</code>	УК-1
4	Назовите технологию для создания пользовательских файловых систем в UNIX.	<code>FUSE</code>	УК-1
5	Укажите системный вызов для чтения атрибутов файла в UNIX.	<code>stat</code>	УК-1
6	Назовите математическую модель для оценки лимитов ресурсов процесса.	ограничения (limits)	ОПК-1
7	Укажите алгоритм завершения дочернего процесса в UNIX.	<code>wait</code>	ОПК-1
8	Назовите метод выполнения внешней программы из процесса.	<code>exec</code>	ОПК-1
9	Укажите механизм передачи данных между процессами через <code>pipe</code> .	канал ( <code>pipe</code> )	ОПК-1
10	Назовите математический подход к моделированию изоляции процессов.	пространства имен ( <code>namespaces</code> )	ОПК-1
11	Укажите тип сигнала для асинхронного уведомления процессов в UNIX.	сигнал ( <code>signal</code> )	ОПК-6
12	Назовите механизм обработки сигналов на продвинутом уровне.	<code>sigaction</code>	ОПК-6
13	Укажите тип сокетов для локального межпроцессного взаимодействия.	unix-сокеты	ОПК-6
14	Назовите протокол для TCP-сокетов в сетевом взаимодействии.	TCP	ОПК-6
15	Укажите базовый механизм сетевого взаимодействия между процессами.	сокеты	ОПК-6
16	Назовите метод мультиплексирования ввода-вывода для высоконагруженных систем.	<code>select/poll/epoll</code>	ПК-1
17	Укажите механизм создания потоков в POSIX-системах.	<code>pthread</code>	ПК-1

18	Назовите инструмент для фильтрации пакетов и трассировки в ядре.	BPF	ПК-1
19	Укажите расширенную версию BPF для динамического анализа.	eBPF	ПК-1
20	Назовите математическую модель для анализа производительности многопоточных систем.	модель очередей (queueing theory)	ПК-1