ЦЕНТРАЛЬНЫЙ УНИВЕРСИТЕТ

УТВЕРЖДЕНА

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

Рабочая программа дисциплины (модуля)
«Архитектура операционных систем»

Направление подготовки: *02.03.01 Математика и компьютерные науки*

Направленность (профиль) подготовки: *Программа двух дипломов НИУ ВШЭ и ЦУ «Прикладная математика и информатика»*

Квалификация (степень) выпускника: *бакалавр*

Форма обучения: *очная*

Срок освоения программы: *4 года*

Год набора: *2024*

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

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

Электронный документ

Москва
2024

Содержание

1. **Краткая характеристика дисциплины (модуля)** 3
2. **Перечень планируемых результатов обучения** 4
3. **Тематический план** 6
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) «Архитектура операционных систем» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по направлению 02.03.01 Математика и компьютерные науки, профиль «Программа двух дипломов НИУ ВШЭ и ЦУ «Прикладная математика и информатика», утвержденный приказом Министерства науки и высшего образования Российской Федерации № 807 от 23.08.2017 года.

Изучение дисциплины (модуля) «Архитектура операционных систем» важно для понимания принципов взаимодействия аппаратного и программного обеспечения, что позволяет эффективно разрабатывать и оптимизировать системы. Это знание способствует созданию надежных, производительных и безопасных вычислительных сред, необходимых в современных технологиях.

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 «Программа двух дипломов НИУ ВШЭ и ЦУ «Прикладная математика и информатика» и входит в вариативную часть Блока 1, формируемую участниками образовательных отношений.

Дисциплина (модуль) является выборной и доступна для изучения на 3 курсе в 5 или 6 семестре на выбор.

Цель изучения дисциплины (модуля): формирование глубокого понимания внутреннего устройства и принципов работы операционных систем для эффективного проектирования и управления вычислительными ресурсами.

Задачи изучения дисциплины (модуля):

— изучить основные компоненты архитектуры операционных систем, включая ядро, процессы, управление памятью, файловые системы и механизмы ввода-вывода, для понимания их взаимосвязи и роли в обеспечении надежной работы программного обеспечения;

— освоить методы и алгоритмы управления системными ресурсами, такие как планирование процессов, синхронизация и распределение памяти, с целью развития навыков анализа производительности и эффективности ОС;

— развить способности к проектированию и модификации операционных систем для различных платформ и сценариев применения, включая высоконагруженные системы и встроенные устройства, с акцентом на практическое применение в разработке и администрировании.

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

— взаимодействие операционной системы с оборудованием;

— алгоритмы планирования задач;

— принципы устройства файловых систем;

уметь:

— вести разработку на эмуляторе компьютера без ОС;

владеть:

— навыком реализиации простой операционной системы с базовым функционалом.

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 реализация математических моделей 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 различных областях человеческой 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
		ОПК-6.2.	Умеет разрабатывать математические программные продукты и комплексы с использованием современных технологий программирования
		ОПК-6.3.	Имеет практический опыт разработки интеллектуальных информационных систем для визуализации результатов исследований
ПК-3.	Способен использовать методы математического и алгоритмического моделирования при решении теоретических и прикладных задач	ПК-3.1.	Знает основные методы математического и алгоритмического моделирования, а также их применение для решения теоретических и прикладных задач
		ПК-3.2.	Умеет разрабатывать и применять математические модели и алгоритмы для решения различных задач, анализируя полученные результаты
		ПК-3.3.	Имеет практический опыт использования методов математического и алгоритмического моделирования в реальных проектах или исследованиях

3. Тематический план

№п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Аудиторная работа		Контроль	Самостоятельная работа	
		Лекции	Семинары			
1	Инструменты для разработки ядра ОС. Процесс загрузки ОС	4	4		18	Кейс
2	Прерывания процессора	4	4		18	Кейс
3	Реализация виртуальной памяти	5	4		20	Кейс
4	Планирование процессов	4	4		18	Кейс
5	Работа с внешними носителями данных	5	4		19	Кейс
6	Базовые принципы организации файловых систем	4	4		19	Кейс
7	Запуск пользовательских процессов	4	4		18	Проект
	Зачет с оценкой			2		Коллоквиум
	Итого:	***30***	***28***	***2***	***130***	
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	Инструменты для разработки ядра ОС. Процесс загрузки ОС	Инструментарий для разработки ядра; процесс начальной загрузки
2	Прерывания процессора	Прерывания и обработка исключений процессора. Режимы процессора и переключение контекста
3	Реализация виртуальной памяти	Виртуальная память: страничная организация и MMU. Алгоритмы выделения памяти: buddy, slab, kmalloc. Синхронизация ядра: спинлоки, мьютексы, отключение прерываний. Создание и завершение процессов; системные вызовы fork/exec
4	Планирование процессов	Планирование процессов: FCFS, RR, приоритеты, MLFQ. Межпроцессное взаимодействие: сигналы, каналы, очереди, shared-memory
5	Работа с внешними носителями данных	Блочные устройства, контроллеры дисков и DMA. Символьные устройства: консоль, UART, таймеры
6	Базовые принципы организации файловых систем	Основы файловых систем: inodes, каталоги, буферный кэш. Проектирование файловых систем: структуры данных и операции
7	Запуск пользовательских процессов	Загрузчик ELF и запуск пользовательских процессов. Пользовательский режим: загрузка ELF, системный вызов, отладка ядра

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. Операционные системы и СУБД ЛИНТЕР : учебное пособие / сост. А. В. Калач, А. Н. Перегудов, В. В. Здольник ; ФКОУ ВО Воронежский институт ФСИН России. - Иваново : Издательско-полиграфический комплекс «ПресСто», 2023. - 152 с. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2158318.

2. Таненбаум, Э. Современные операционные системы : научно-популярное издание / Э. Таненбаум, Х. Бос. - 4-е изд. - Санкт-Петербург : Питер, 2021. - 1120 с. - (Серия «Классика computer science»). - ISBN 978-5-4461-9883-2. - Текст : электронный. - URL: https://znanium.ru/catalog/product/1857039.

Дополнительная литература:

1. Новожилов О. П. Архитектура ЭВМ и вычислительных систем : учебник для вузов / О. П. Новожилов. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2025. — 505 с. — (Высшее образование). — ISBN 978-5-534-20365-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/568920.

2. Толстобров А. П. Архитектура ЭВМ : учебник для вузов / А. П. Толстобров. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 162 с. — (Высшее образование). — ISBN 978-5-534-16839-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/566711.

3. Гостев И. М. Операционные системы : учебник и практикум для вузов / И. М. Гостев. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2025. — 164 с. — (Высшее образование). — ISBN 978-5-534-04520-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/561557.

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) «Архитектура операционных систем» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, кейсы, коллоквиум, проект, 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.

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.

Кейс – практическая работа студентов над реальными или смоделированными задачами, что позволяет студенту применять теоретические знания на практике.

Студент самостоятельно разрабатывает стратегию решения поставленной задачи, что способствует развитию навыков критического мышления и самостоятельного принятия решений. Такой подход помогает подготовить будущих специалистов к реальным вызовам

в их профессиональной деятельности.

Коллоквиум – устные ответы на вопросы, список которых известен студенту заранее.

В процессе подготовки к коллоквиуму необходимо проанализировать учебные материалы, ознакомившись с лекциями, учебниками и дополнительными источниками, акцентируя внимание на ключевых темах. Рекомендуется создать структурированные конспекты, выделяя основные идеи, термины и формулы.

Проект – исследовательская работа по курсу и презентация результатов.

Для успешной подготовки к проекту: четко определите цели и задачи проекта, распределите роли и обязанности между участниками, а также установите сроки выполнения каждой части работы. Регулярно проводите встречи для обсуждения прогресса и решения возникающих вопросов.

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 операционных систем»

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, если иное не указано в силлабусе дисциплины:

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
9	Отлично	Зачтено	
8	Отлично	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			исследований, а также решает сложные 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 операционных систем» оценивается следующим образом:

Активность	Вес	Количество	Описание
Кейс	40%	4	Практическая работа студентов над реальными или смоделированными задачами, что позволяет студенту применять теоретические знания на практике
Проект	40%	1	Исследовательская работа по дисциплине (модулю) и презентация результатов
Зачёт с оценкой	20%	1	Коллоквиум - устные ответы на вопросы, список которых известен студенту заранее

Формула расчёта итоговой оценки по дисциплине (модулю) «Архитектура операционных систем»: «0,4 × среднее за кейсы + 0,4 × проект + 0,2 × зачёт с оценкой».

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные задания для кейсов

Кейс 1: Прерывания процессора (Прерывания и обработка исключений процессора. Режимы процессора и переключение контекста)

1. Объясните разницу между аппаратными и программными прерываниями. Напишите фрагмент кода на C с использованием signal() для обработки программного прерывания SIGUSR1, который выводит сообщение "Signal received" и увеличивает счетчик.
2. Опишите процесс переключения контекста при прерывании. Напишите простой пример кода, который устанавливает обработчик прерывания для SIGINT с помощью sigaction(), сохраняющий и восстанавливающий состояние (например, переменную).
3. Что такое режимы процессора (user mode vs kernel mode)? Напишите код, который вызывает системный вызов getpid() и объясняет, как происходит переход в kernel mode при его выполнении.
4. Реализуйте обработчик исключения для деления на ноль (SIGFPE). Напишите программу, которая пытается выполнить деление на ноль, перехватывает сигнал и выводит "Division by zero caught".
5. Опишите механизм маскирования прерываний. Напишите код с использованием sigprocmask(), который блокирует SIGINT на 5 секунд, а затем разблокирует его.

Кейс 2: Планирование процессов (Планирование процессов: FCFS, RR, приоритеты, MLFQ. Межпроцессное взаимодействие: сигналы, каналы, очереди, shared-memory)

1. Объясните алгоритм планирования FCFS. Напишите код, который создает три дочерних процесса с помощью fork(), каждый из которых выполняет sleep() на разное время, и родитель ждет их завершения в порядке FCFS.
2. Опишите Round Robin (RR) планирование. Напишите программу, имитирующую RR с использованием alarm() для таймера, где процессы выполняют работу в циклах и переключаются каждые 2 секунды.
3. Что такое приоритетное планирование? Напишите код с использованием nice() для установки приоритетов двум процессам и демонстрации их выполнения (например, один процесс с высоким приоритетом завершается быстрее).
4. Реализуйте межпроцессное взаимодействие с помощью каналов (pipes). Напишите код, где родитель пишет строку "Hello from parent" в канал, а дочерний процесс читает и выводит ее, затем дочерний пишет "Hello from child", а родитель читает и выводит.
5. Опишите использование shared memory для IPC. Напишите программу с использованием shmget() и shmat(), где два процесса (родитель и потомок) обмениваются значением переменной через разделяемую память, увеличивая ее по очереди.

Кейс 3: Реализация виртуальной памяти (Виртуальная память: страничная организация и MMU. Алгоритмы выделения памяти: buddy, slab, kmalloc. Синхронизация ядра: спинлоки, мьютексы, отключение прерываний. Создание и завершение процессов; системные вызовы fork/exec)

1. Объясните страничную организацию виртуальной памяти и роль MMU. Напишите код, который вызывает malloc() для выделения 1 МБ памяти, проверяет адрес и освобождает его с помощью free(), демонстрируя работу с виртуальными адресами.
2. Опишите алгоритм выделения памяти buddy. Напишите простой пример кода, имитирующего buddy-аллокатор для выделения блоков размера 2^n (используя массивы или структуры для симуляции).
3. Что такое slab-аллокатор? Напишите код с использованием kmalloc() (в контексте ядра, но симулируйте в user-space с malloc()), который выделяет и освобождает память для структур, объясняя преимущества slab над общим аллокатором.
4. Объясните синхронизацию ядра с помощью спинлоков и мьютексов. Напишите код с использованием pthread_mutex_lock() (для симуляции мьютекса в user-space), где два потока пытаются увеличить общую переменную, демонстрируя синхронизацию.
5. Опишите системные вызовы fork() и exec(). Напишите код, где родительский процесс создает дочерний с помощью fork(), дочерний вызывает execl() для запуска программы /bin/ls, а родитель ждет завершения с помощью wait().

Примерные задания для коллоквиума

Задание 1: Объясните роль inodes в файловой системе. Нарисуйте простую диаграмму структуры inode (включая поля для метаданных, указателей на блоки данных) и напишите псевдокод функции, которая по пути к файлу находит его inode и выводит размер файла.

Задание 2: Опишите, как каталоги организуют файлы и подкаталоги. Напишите код на C, который рекурсивно обходит каталог с помощью opendir() и readdir(), выводя имена файлов и их типы (файл или каталог).

Задание 3: Что такое буферный кэш и зачем он нужен в файловой системе? Объясните механизм кэширования страниц (page cache) и напишите простой пример кода на C с использованием open(), read() и close(), демонстрирующий чтение файла с возможным попаданием в кэш.

Задание 4: Объясните основные операции файловой системы (create, read, write, delete). Напишите псевдокод алгоритма создания нового файла, включая выделение inode, обновление каталога и запись данных в блоки.

Задание 5: Опишите структуру данных inode: какие поля она содержит и как они используются. Напишите код на C, который с помощью stat() получает информацию о файле (размер, права доступа, время модификации) и выводит ее.

Задание 6: Как каталоги связаны с inodes? Объясните концепцию жестких и символических ссылок. Напишите код на C, который создает жесткую ссылку на файл с помощью link() и проверяет, что два имени указывают на один inode (сравнивая номера inodes).

Задание 7: Что такое буферный кэш в контексте операций ввода-вывода? Опишите алгоритм write-back кэширования. Напишите простой пример кода, имитирующего буферный кэш: структура данных для кэша и функции для чтения/записи с проверкой попадания.

Задание 8: Объясните проектирование структуры данных для хранения каталогов (например, линейный список vs. хэш-таблица). Нарисуйте диаграмму B-дерева для каталога с 10 файлами и напишите псевдокод поиска файла по имени в такой структуре.

Задание 9: Опишите операцию чтения файла в файловой системе. Напишите код на C, который открывает файл, читает его содержимое блоками по 1024 байта с помощью read(), и выводит общее количество прочитанных байтов, демонстрируя взаимодействие с буферным кэшем.

Задание 10: Как работает выделение блоков данных для файлов (bitmap vs. linked list)? Объясните преимущества и недостатки. Напишите псевдокод функции выделения свободного блока с использованием bitmap (массив битов).

Задание 11: Объясните роль каталогов в иерархии файловой системы. Напишите код на C, который создает новый каталог с помощью mkdir(), добавляет в него файл с помощью open() и write(), затем удаляет файл с помощью unlink().

Задание 12: Что такое inode table и как она связана с каталогами? Опишите процесс разрешения пути (path resolution). Напишите псевдокод алгоритма, который по абсолютному пути (например, "/home/user/file.txt") находит inode файла, начиная от корневого каталога.

Задание 13: Опишите механизм буферного кэша для предотвращения потери данных (fsync). Напишите код на C, который записывает данные в файл с помощью write(), затем вызывает fsync() для принудительной записи на диск и измеряет время операции.

Задание 14: Объясните проектирование операций удаления файла: что происходит с inode и блоками данных. Напишите код на C, который удаляет файл с помощью remove() и проверяет, что inode освобожден (с помощью stat() после удаления).

Задание 15: Как буферный кэш взаимодействует с виртуальной памятью? Опишите концепцию mapped files. Напишите простой пример кода на C с использованием mmap() для отображения файла в память, чтения и записи данных напрямую через указатель.

Примерное описание задания и критерии оценивания к проекту

Описание проекта:

Итоговый проект направлен на комплексную проверку и закрепление знаний, полученных в ходе изучения ключевых аспектов архитектуры операционных систем. Студентам предлагается разработать прототип или модель компонентов операционной системы, отражающих основные механизмы и принципы, изученные в дисциплине. Проект должен продемонстрировать понимание и умение применять технологии разработки ядра, обработку прерываний, управление виртуальной памятью, планирование процессов, работу с внешними носителями данных, основы организации файловых систем и запуск пользовательских процессов.

В рамках проекта студентам необходимо:

- Создать минимальную среду или модуль, реализующий процесс загрузки ОС с использованием загрузчика и инициализации ядра.
- Реализовать обработку аппаратных и программных прерываний с использованием таблицы векторов прерываний (IDT).
- Организовать механизм виртуальной памяти с таблицами страниц, обработкой page fault и базовым управлением памятью.
- Внедрить простой планировщик процессов, обеспечивающий переключение контекста и управление состояниями процессов.
- Обеспечить взаимодействие с внешним устройством хранения данных, реализовав базовые операции ввода-вывода с учетом асинхронности и прерываний.
- Спроектировать структуру файловой системы с каталогами, файлами и метаданными, реализовать алгоритмы размещения и управление свободным пространством.
- Обеспечить запуск пользовательских процессов с корректной инициализацией и мониторингом.

Проект может быть выполнен в форме программного модуля, эмулятора, расширения существующей ОС или учебного симулятора. Важно продемонстрировать интеграцию изученных тем и понимание взаимодействия компонентов ОС.

Критерии оценивания:

1. Понимание и применение концепций загрузки ОС
2. Обработка прерываний
3. Механизмы виртуальной памяти

4. Планирование процессов
5. Работа с внешними носителями данных
6. Организация файловой системы
7. Запуск и управление пользовательскими процессами
8. Качество реализации и интеграция компонентов
9. Демонстрация и защита проекта

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1	Укажите инструмент для разработки ядра ОС.	Компилятор / compiler, gcc	УК-1
2	Укажите метод синтеза процесса загрузки ОС.	BIOS / bios, firmware	УК-1
3	Укажите практический опыт работы с отладчиком ядра.	GDB / gdb, debugger	УК-1
4	Укажите концепцию дискретной математики для обработки прерываний.	Вектор / vector, таблица	ОПК-1
5	Укажите применение математической логики в синхронизации.	Мьютекс / mutex, lock	ОПК-1
6	Укажите практический опыт моделирования планирования процессов.	Симулятор / simulator, эмулятор	ОПК-1
7	Укажите базовый математический аппарат для страничной организации.	Адрес / address, mapping	ОПК-4
8	Укажите умение применять аппарат в оценке сложности buddy.	Логарифм / logarithm, log	ОПК-4
9	Укажите опыт реализации алгоритма FCFS.	Очередь / queue, fifo	ОПК-4
10	Укажите опыт оценки качества MLFQ.	Приоритет / priority, level	ОПК-4
11	Укажите алгоритм для системного вызова fork.	Клонирование / cloning, copy	ОПК-6
12	Укажите умение разрабатывать программу для DMA.	Драйвер / driver, module	ОПК-6
13	Укажите умение применять код в UART.	Передача / transmission, serial	ОПК-6
14	Укажите опыт визуализации процессов в ядре.	Граф / graph, tree	ОПК-6
15	Укажите опыт симуляции межпроцессного взаимодействия.	Канал /pipe, channel	ОПК-6
16	Укажите метод моделирования виртуальной памяти.	MMU / mmu, unit	ПК-3
17	Укажите умение применять модель в планировании RR.	Время / time, slice	ПК-3
18	Укажите умение моделировать shared-memory.	Сегмент / segment, block	ПК-3
19	Укажите опыт реализации модели для inodes.	Узел / node, index	ПК-3
20	Укажите опыт расчёта сложности для ELF-загрузчика.	Секция /section, header	ПК-3