

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»**

Институт информационных технологий, математики и механики

УТВЕРЖДЕНО
решением Ученого совета ННГУ
протокол № 10 от 27.08.2025

Рабочая программа дисциплины

Архитектуры и операционные системы: аспекты производительности

Уровень высшего образования
Бакалавриат

Направление подготовки / специальность
02.03.02 - Фундаментальная информатика и информационные технологии

Направленность образовательной программы
Системное программирование

Форма обучения
очная

г. Нижний Новгород

2025 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.ДВ.02.01 Архитектуры и операционные системы: аспекты производительности относится к части, формируемой участниками образовательных отношений образовательной программы.

2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции		Наименование оценочного средства	
	Индикатор достижения компетенции (код, содержание индикатора)	Результаты обучения по дисциплине	Для текущего контроля успеваемости	Для промежуточной аттестации
ПК-Опер_1: Осуществляет управление архитектурой изолированной (неинтегрированной) программной системы	<p>ПК-Опер_1.1: Выявляет и согласовывает требования к программной системе с точки зрения архитектуры</p> <p>ПК-Опер_1.2: Осуществляет выбор и моделирование архитектурного решения для реализации программной системы</p> <p>ПК-Опер_1.3: Разрабатывает разделы по архитектуре проектных и эксплуатационных документов программной системы</p> <p>ПК-Опер_1.4: Контролирует реализацию и испытания программной системы с точки зрения архитектуры</p> <p>ПК-Опер_1.5: Осуществляет сопровождение эксплуатации программной системы с точки зрения архитектуры</p>	<p>ПК-Опер_1.1:</p> <p>ПК-Опер_1.1. У-1. Способен выявлять несоответствия требований заказчика к программной системе с точки зрения архитектуры</p> <p>ПК-Опер_1.1. У-2. Способен описывать требования к программной системе с точки зрения архитектуры</p> <p>ПК-Опер_1.1. У-3. Умеет проверять требования на соответствие архитектуре программной системы</p> <p>ПК-Опер_1.1. У-4. Умеет выявлять требования к архитектуре программной системы путем проведения интервью с заинтересованными сторонами</p> <p>ПК-Опер_1.1. У-5. Умеет формулировать архитектурные требования к программной системе</p> <p>ПК-Опер_1.1. З-1. Знает методы управления требованиями.</p> <p>ПК-Опер_1.1. З-2. Знает методы моделирования архитектуры программной системы</p> <p>ПК-Опер_1.1. З-3. Знает методы проектирования архитектуры программной</p>	Контрольная работа	Зачёт: Контрольные вопросы

		<p>системы</p> <p>ПК-Опер_1.2: ПК-Опер_1.2. У-1. Способен выбрать оптимальное архитектурное решение с учетом особенностей программной системы и принципов её организации</p> <p>ПК-Опер_1.2. У-2. Способен определить архитектуру системы, ее, бизнес-процессов, структуру данных и отдельных компонентов программной системы и методы их интеграции.</p> <p>ПК-Опер_1.2. У-3. Способен определить перечень элементов архитектуры, которые должны быть защищены от угроз безопасности информации, связанных с нарушением конфиденциальности, целостности и доступности.</p> <p>ПК-Опер_1.2. У-4. Способен моделировать архитектурное решение для изолированной программной системы</p> <p>ПК-Опер_1.2. У-5. Умеет проектировать бизнес-архитектуру программных систем с применением лучших практик, шаблонов и стилей архитектурного проектирования</p> <p>ПК-Опер_1.2. У-6. Умеет проектировать архитектуру интегрированной программной системы с учетом устойчивости к воздействиям внутреннего и внешнего нарушителя (хакер, неосторожный пользователь, программист, поставщик компонентов) на любую из подсистем и с использованием методов и шаблонов конструктивной (встроенной) безопасности</p> <p>ПК-Опер_1.2. З-1. Знает методы моделирования архитектуры программных систем и критерии сравнения</p>		
--	--	---	--	--

		<p>архитектурных решений ПК-Опер_1.2. 3-2. Знает протоколы взаимодействия программных систем ПК-Опер_1.2. 3-3. Знает нормативные правовые акты, организационно-распорядительные документы и методические рекомендации, определяющие требования к безопасности программного обеспечения ПК-Опер_1.2. 3-4. Знает методики определения актуальных угроз безопасности персональных данных при их обработке в информационных системах персональных данных ПК-Опер_1.2. 3-5. Знает лучшие практики и шаблоны создания конструктивно-безопасных интегрированных информационных систем</p> <p>ПК-Опер_1.3: ПК-Опер_1.3. У-1. Способен описывать технические и организационные меры, обеспечивающие сохранение и восстановление программного обеспечения ПК-Опер_1.3. У-2. Умеет проектировать и моделировать архитектурные элементы программных систем и их взаимосвязи. ПК-Опер_1.3. У-3. Умеет формировать технические и организационные меры для защиты программной системы от несанкционированного доступа к элементам конфигурации ПК-Опер_1.3. 3-1. Знает методы моделирования и технического описания архитектуры программных систем ПК-Опер_1.3. 3-2. Знает нормативные правовые акты, организационно-распорядительные документы</p>		
--	--	--	--	--

		<p><i>и методические рекомендации, определяющие требования к безопасности программного обеспечения</i></p> <p><i>ПК-Опер_1.3. 3-3. Знает методики определения актуальных угроз безопасности персональных данных при их обработке в информационных системах персональных данных</i></p> <p><i>ПК-Опер_1.4:</i></p> <p><i>ПК-Опер_1.4. У-1. Способен проверять соответствие реализации программной системы выбранному архитектурному решению</i></p> <p><i>ПК-Опер_1.4. У-2. Способен проверять результаты испытаний программной системы на соответствие архитектуре и архитектурным решениям</i></p> <p><i>ПК-Опер_1.4. У-3. Умеет проверять характеристики реализованной программной системы на соответствие архитектурным требованиям</i></p> <p><i>ПК-Опер_1.4. У-4. Умеет формулировать рекомендации по изменению реализованной программной системы для обеспечения соответствия ее архитектурным требованиям</i></p> <p><i>ПК-Опер_1.4. 3-1. Знает способы определения характеристик работающей программной системы</i></p> <p><i>ПК-Опер_1.4. 3-2. Знает методы параметризации архитектуры программных систем</i></p> <p><i>ПК-Опер_1.4. 3-3. Знает основы процесса управления изменениями программных систем</i></p> <p><i>ПК-Опер_1.5:</i></p> <p><i>ПК-Опер_1.5. У-1. Способен проверять запросы на изменения программной системы на реализуемость с</i></p>		
--	--	--	--	--

		<p>точки зрения архитектуры программной системы</p> <p>ПК-Опер_1.5. У-2. Способен согласовывать запросы на изменения программной системы с точки зрения архитектуры</p> <p>ПК-Опер_1.5. У-3. Умеет взаимодействовать с авторами запросов на изменения программной системы для уточнения содержания запросов</p> <p>ПК-Опер_1.5. У-4. Умеет выявлять несоответствия и изменять запросы на изменения программной системы для обеспечения их соответствия выбранной архитектуре</p> <p>ПК-Опер_1.5. З-1. Знает основы процесса управления изменениями программных систем</p> <p>ПК-Опер_1.5. З-2. Знает методы обеспечения устойчивости функционирования программной системы</p> <p>ПК-Опер_1.5. З-3. Знает методы обеспечения надежности архитектуры программной системы</p>		
<p>ПК-ТОП_11: Способен разрабатывать, оптимизировать и отлаживать системное программное обеспечение</p>	<p>ПК-ТОП_11.1: Разрабатывает низкоуровневый код для встраиваемых систем и драйверов</p> <p>ПК-ТОП_11.2: Оптимизирует код под ограниченные ресурсы</p> <p>ПК-ТОП_11.3: Работает с ОС, загрузчиками и аппаратурой</p> <p>ПК-ТОП_11.4: Отлаживает системное ПО без ОС</p>	<p>ПК-ТОП_11.1: ПК-ТОП_11.1. З-1. Знает архитектуру современных процессоров, особенности работы с оборудованием процессора</p> <p>ПК-ТОП_11.1. З-2. Знает принципы взаимодействия ПО с аппаратурой</p> <p>ПК-ТОП_11.1. У-1. Умеет разрабатывать низкоуровневый код для встроенного программного обеспечения и драйверов</p> <p>ПК-ТОП_11.1. У-2. Умеет разрабатывать драйверы для аппаратных устройств</p> <p>ПК-ТОП_11.2:</p>	<p>Контрольная работа</p>	<p>Зачёт: Контрольные вопросы</p>

		<p>ПК-ТОП_11.2. 3-1. Знает методы оптимизации (кэш, память, тактовая частота процессора)</p> <p>ПК-ТОП_11.2. 3-2. Знает инструменты профилирования</p> <p>ПК-ТОП_11.5. 3-2. Знает архитектуру компиляторов, оптимизации для выбранных архитектур процессоров</p> <p>ПК-ТОП_11.2. У-1. Умеет анализировать бенчмарки, выявлять узкие места</p> <p>ПК-ТОП_11.2. У-2. Умеет разрабатывать код, оптимизированный для выбранной аппаратной архитектуры</p> <p>ПК-ТОП_11.3:</p> <p>ПК-ТОП_11.3. 3-1. Знает архитектуру ядра Linux.</p> <p>ПК-ТОП_11.3. 3-2. Знает принципы работы встраиваемых операционных систем</p> <p>ПК-ТОП_11.3. У-1. Умеет портировать код между различными вариантами загрузчиков и ОС</p> <p>ПК-ТОП_11.3. У-2. Умеет выполнять разработку на стыке программного обеспечения и оборудования</p> <p>ПК-ТОП_11.4:</p> <p>ПК-ТОП_11.4. 3-1. Знает методы анализа bare-metal сбоев</p> <p>ПК-ТОП_11.4. 3-2. Знает форматы бинарных файлов, работу с памятью</p> <p>ПК-ТОП_11.4. У-1. Умеет отлаживать код через специализированные отладчики системного программного обеспечения</p> <p>ПК-ТОП_11.4. У-2. Умеет анализировать дампы памяти, дизассемблированный код</p>		
ПК-Ф1: Способен	ПК-Ф1.1: Применяет	ПК-Ф1.1:	Контрольная	

<p>планировать и организовывать аналитические работы с использованием технологий больших данных</p>	<p>методы машинного обучения и статистического анализа. Знает типы анализа больших данных, виды аналитики; теоретические и прикладные основы анализа больших данных; содержание этапов жизненного цикла больших данных ПК-Ф1.2: Обеспечивает соответствие результатов анализа бизнес-задачам заказчика. Умеет планировать и проводить аналитические работы с использованием технологий больших данных ПК-Ф1.3: Подготавливает отчеты и визуализации для презентации результатов. Умеет проводить анализ больших данных</p>	<p>ПК-Ф1.1. 3-1. Знает основные алгоритмы машинного обучения ПК-Ф1.1. 3-2. Знает методы статистического анализа данных ПК-Ф1.1. 3-3. Знает критерии выбора алгоритмов для различных задач ПК-Ф1.1. У-1. Умеет реализовывать алгоритмы машинного обучения ПК-Ф1.1. У-2. Умеет интерпретировать результаты статистического анализа ПК-Ф1.2: ПК-Ф1.2. 3-1. Знает методы перевода бизнес-требований в аналитические задачи ПК-Ф1.2. 3-2. Знает ключевые бизнес-метрики в предметной области ПК-Ф1.2. 3-3. Знает принципы интерпретации результатов для бизнес-пользователей ПК-Ф1.2. У-1. Умеет адаптировать аналитические модели под бизнес-потребности ПК-Ф1.2. У-2. Умеет оценивать экономический эффект от аналитических решений ПК-Ф1.3: ПК-Ф1.3 3-1. Знает принципы эффективной визуализации данных ПК-Ф1.3 3-2. Знает инструменты создания аналитических отчетов ПК-Ф1.3 3-3. Знает методы сторителлинга на основе данных ПК-Ф1.3 У-1. Умеет выбирать оптимальные типы визуализации ППК-ДА3.3 У-2. Умеет создавать интерактивные дашборды</p>	<p>работа</p>	<p>Зачёт: Контрольные вопросы</p>
---	--	--	---------------	--

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

	очная
Общая трудоемкость, з.е.	2
Часов по учебному плану	72
в том числе	
аудиторные занятия (контактная работа):	
- занятия лекционного типа	12
- занятия семинарского типа (практические занятия / лабораторные работы)	12
- КСР	1
самостоятельная работа	47
Промежуточная аттестация	0 Зачёт

3.2. Содержание дисциплины

(структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий)

Наименование разделов и тем дисциплины	Всего (часы)	в том числе			
		Контактная работа (работа во взаимодействии с преподавателем), часы из них			Самостоятельная работа обучающегося, часы
		Занятия лекционного типа	Занятия семинарского типа (практические занятия/лабораторные работы), часы	Всего	
0 Ф 0	0 Ф 0	0 Ф 0	0 Ф 0	0 Ф 0	
Взаимодействие ядер в многоядерной системе.	7	1	2	3	4
Кеширование в многоядерных и многопроцессорных системах. Когерентность кешей.	11	2	2	4	7
Поддержка многопоточности на уровне микроархитектуры CPU.	3	1	0	1	2
Планирование в многоядерных архитектурах. Алгоритм планирования MS Windows.	8	2	0	2	6
Аспекты производительности виртуальной памяти со страничной организацией.	7	1	2	3	4
Особенности производительности в системах архитектуры ccNUMA.	4	1	1	2	2
Программирование без блокировок (Lock-free).	23	3	4	7	16
Мониторинг работы компонентов CPU.	8	1	1	2	6
Аттестация	0				
КСР	1			1	
Итого	72	12	12	25	47

Содержание разделов и тем дисциплины

Цели и задачи изучения дисциплины.

Целями освоения дисциплины являются.

- Формирование у обучающихся системного понимания принципов организации и функционирования современных многоядерных и многопроцессорных вычислительных систем.
- Изучение микроархитектурных механизмов повышения производительности процессоров, включая многопоточность, когерентность кешей, виртуальную память и NUMA-архитектуры.
- Освоение принципов взаимодействия аппаратного и программного обеспечения в многопоточных и высокопроизводительных вычислительных системах.
- Изучение методов разработки эффективного параллельного и lock-free программного обеспечения с учетом особенностей современных моделей памяти.

Задачами освоения дисциплины являются.

- Освоить принципы организации кеш-памяти в многоядерных системах и механизмы обеспечения когерентности кешей. Изучить протоколы когерентности MSI, MESI, MOESI и MESIF, а также особенности их реализации.
- Изучить методы планирования потоков и процессов в многоядерных системах, включая алгоритмы планировщика MS Windows.
- Освоить принципы организации виртуальной памяти, работу TLB и использование больших страниц памяти.
- Изучить атомарные операции и методы разработки lock-free структур данных. Рассмотреть механизм Read-Copy-Update (RCU) и области его применения.
- Освоить методы мониторинга производительности CPU с использованием аппаратных счетчиков и специализированных инструментов анализа.

Содержание разделов и тем дисциплины.

1. Взаимодействие ядер в многоядерной системе

Архитектура современных многоядерных CPU. Shared memory и distributed memory. Межъядерные соединения. Синхронизация между ядрами. Межпроцессорные прерывания.

2. Кеширование в многоядерных системах

Архитектура кешей (private/shared, Inclusive/exclusive/non-inclusive). Когерентность кешей. Snooping и directory-based protocols. Протокол MSI. Протокол MESI. Протоколы MOESI и MESIF. Оптимизация межъядерного трафика.

3. Поддержка многопоточности в микроархитектуре CPU

Процессы и потоки. Причины простоев конвейера. Fine-grained multithreading. Simultaneous Multithreading (SMT), Hyper-Threading. Работа ROB, scheduler, execution units в SMP-микроархитектуре. Производительность SMT.

4. Планирование в многоядерных архитектурах

Привязка потока (Affinity). Миграция потоков. NUMA-aware scheduling. Алгоритм работы планировщика MS Windows.

5. Производительность виртуальной памяти

Страничная организация памяти. Многоуровневые таблицы страниц. Translation Lookaside Buffer (TLB). Большие страницы (Huge Pages). Принцип локальности, рабочий набор, предвыборка данных из памяти. Архитектура ccNUMA

Основы NUMA. Локальная и удаленная память, домены NUMA. Архитектура ccNUMA (Cache-coherent NUMA). Межсокетное взаимодействие (пример Intel UPI). Производительность NUMA (латентность и пропускная способность доступа к памяти). Оптимизация программ с учетом NUMA.

7. Программирование без блокировок (Lock-Free)

Основы lock-free программирования. Программирование wait-free, lock-free, obstruction-free. Модели памяти: sequential consistency, relaxed memory model, семантика acquire/release, memory fences.

Атомарные операции. ABA problem. Принципы работы Read-Copy-Update (RCU). Lock-free версии структур данных: стек, очередь.

8. Мониторинг работы CPU

Аппаратные счетчики производительности. Архитектурные и микроархитектурные события. Счетчики вне ядра. Инструменты мониторинга. Roofline model.

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа обучающихся включает в себя подготовку к контрольным вопросам и заданиям для текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведенным в п. 5.

Для обеспечения самостоятельной работы обучающихся используются:

Электронные курсы, созданные в системе электронного обучения ННГУ:

Архитектура и организация ЭВМ, <https://e-learning.unn.ru/course/view.php?id=2435>.

Иные учебно-методические материалы:

А.В. Кудин, А.В. Линёв. Архитектура и операционные системы параллельных вычислительных систем. Нижний Новгород, 2007. <http://www.unn.ru/books/resources.html>

С.Бастраков, В.Гергель, А.Горшков, Е.Козин, А.Линев, И.Мееров, А.Сиднев, А.Сысоев. Введение в принципы функционирования и применения современных мультиядерных архитектур (на примере Intel Xeon Phi). <http://www.intuit.ru/studies/courses/10611/1095/info>

В.Гуров, В.Чуканов. Архитектура и организация ЭВМ. <http://www.intuit.ru/studies/courses/60/60/info>

В.Гуров. Архитектура микропроцессоров. <http://www.intuit.ru/studies/courses/604/460/info>

Д.Северов. Архитектура вычислительных систем и язык ассемблера. <http://www.intuit.ru/studies/courses/535/391/info>

5. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)

5.1 Типовые задания, необходимые для оценки результатов обучения при проведении текущего контроля успеваемости с указанием критериев их оценивания:

5.1.1 Типовые задания (оценочное средство - Контрольная работа) для оценки сформированности компетенции ПК-Опер_1:

Какие способы межъядерного взаимодействия используются в современных CPU?

Что такое shared memory architecture?

Что такое interconnect и какие его типы существуют?

Что такое Network-on-Chip (NoC)?

Что такое межпроцессорное прерывание (IPI)?

Какие механизмы синхронизации используются между ядрами?

Какие уровни кеш-памяти существуют в современных процессорах?

Какие существуют подходы к обеспечению когерентности кешей?

Как работает протокол MESI?

Как работает протокол MOESI?

5.1.2 Типовые задания (оценочное средство - Контрольная работа) для оценки сформированности компетенции ПК-ТОП_11:

Что такое аппаратная многопоточность?

Как многопоточность влияет на IPC?

Что такое thread affinity?

Какие особенности имеет планирование в NUMA-системах?

Как работает планировщик MS Windows?

Почему локальность данных важна для планирования?

Что такое page walk? Что такое TLB?

Что такое большие страницы (Huge Pages)? Какие преимущества они дают?

Что такое ccNUMA?

Что такое Intel UPI?

Что такое NUMA-aware scheduling?

5.1.3 Типовые задания (оценочное средство - Контрольная работа) для оценки сформированности компетенции ПК-Ф1:

Что такое lock-free программирование? Чем lock-free отличается от wait-free?

Что такое memory ordering? Что такое relaxed memory model?

Для чего используются memory fences?

Что такое acquire/release semantics?

Какие существуют способы решения ABA problem?

Что такое Read-Copy-Update (RCU)? Как работает механизм grace period?

Что такое аппаратные счетчики производительности? Какие события могут измеряться внутри ядра?

Что такое PMU (Performance Monitoring Unit)?

Какие инструменты используются для профилирования CPU?

Что такое roofline model?

Какие метрики важны для многопоточных приложений?

Критерии оценивания (оценочное средство - Контрольная работа)

Оценка	Критерии оценивания
превосходно	Студент получил верный ответ во всех заданиях. При этом студент продемонстрировал знание дополнительного материала.
отлично	Студент получил верный ответ во всех заданиях.
очень хорошо	Студент получил верный ответ в большинстве заданий.
хорошо	Студент решил большую часть задач с незначительными недочетами
удовлетворительно	Студент решил большую часть задач с существенными недочетами
неудовлетворительно	Студент допускает грубые ошибки в решении стандартных задач.
плохо	Отсутствие знаний материала, отсутствует способность решения стандартных задач.

5.2. Описание шкал оценивания результатов обучения по дисциплине при промежуточной аттестации

Шкала оценивания сформированности компетенций

Уровень сформированности компетенций (индикатора достижения компетенций)	плохо	неудовлетворительно	удовлетворительно	хорошо	очень хорошо	отлично	превосходно
	не зачтено			зачтено			
<u>Знания</u>	Отсутствие знаний теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от	Уровень знаний ниже минимальных требований. Имели место грубые ошибки	Минимально допустимый уровень знаний. Допущено много негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько несуществе	Уровень знаний в объеме, соответствующем программе подготовки. Ошибок нет.	Уровень знаний в объеме, превышающем программу подготовки.

	ответа			ошибок	нных ошибок		
<u>Умения</u>	Отсутствие минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки	Продемонстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме	Продемонстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания в полном объеме, но некоторые с недочетами	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания в полном объеме, но некоторые с недочетами	Продемонстрированы все основные умения. Решены все основные задачи с отдельными и несущественными недочетами, выполнены все задания в полном объеме	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов
<u>Навыки</u>	Отсутствие базовых навыков. Невозможность оценить наличие навыков вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки	Имеется минимальный набор навыков для решения стандартных задач с некоторым и недочетами	Продемонстрированы базовые навыки при решении стандартных задач с некоторым и недочетами	Продемонстрированы базовые навыки при решении стандартных задач без ошибок и недочетов	Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов	Продемонстрирован творческий подход к решению нестандартных задач

Шкала оценивания при промежуточной аттестации

Оценка		Уровень подготовки
зачтено	превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно», продемонстрированы знания, умения, владения по соответствующим компетенциям на уровне выше предусмотренного программой
	отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично».
	очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо»
	хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо».
	удовлетворительно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»
не зачтено	неудовлетворительно	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно».
	плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

5.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения на промежуточной аттестации с указанием критериев их оценивания:

5.3.1 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-Опер_1

Какие способы межъядерного взаимодействия используются в современных CPU?

Что такое shared memory architecture?

Что такое interconnect и какие его типы существуют?

Что такое Network-on-Chip (NoC)?

Что такое межпроцессорное прерывание (IPI)?

Какие механизмы синхронизации используются между ядрами?

Какие уровни кеш-памяти существуют в современных процессорах?

Какие существуют подходы к обеспечению когерентности кешей?

Как работает протокол MESI?

Как работает протокол MOESI?

Что такое аппаратная многопоточность?

Как многопоточность влияет на IPC?

5.3.2 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-ТОП_11

Что такое thread affinity?

Какие особенности имеет планирование в NUMA-системах?

Как работает планировщик MS Windows?

Почему локальность данных важна для планирования?

Что такое page walk? Что такое TLB?

Что такое большие страницы (Huge Pages)? Какие преимущества они дают?

Что такое ccNUMA?

Что такое Intel UPI?

Что такое NUMA-aware scheduling?

5.3.3 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-Ф1

Что такое lock-free программирование? Чем lock-free отличается от wait-free?

Что такое memory ordering? Что такое relaxed memory model?

Для чего используются memory fences?

Что такое acquire/release semantics?

Какие существуют способы решения ABA problem?

Что такое Read-Copy-Update (RCU)? Как работает механизм grace period?

Что такое аппаратные счетчики производительности? Какие события могут измеряться внутри ядра?

Что такое PMU (Performance Monitoring Unit)?

Какие инструменты используются для профилирования CPU?

Что такое roofline model?

Какие метрики важны для многопоточных приложений?

Критерии оценивания (оценочное средство - Контрольные вопросы)

Оценка	Критерии оценивания
зачтено	Выполнены все или большая часть этапов решения задачи или задача решена с незначительными недочетами. Результаты работы представлены преподавателю в срок.
не зачтено	Выполнены не все практические задания или выполнены не в полном объеме (представлено не полное описание этапов выполнения заданий, получен неверный ответ, результаты работы не представлены преподавателю).

6. Учебно-методическое и информационное обеспечение дисциплины (модуля)

Основная литература:

1. Максимов Николай Вениаминович. Архитектура ЭВМ и вычислительных систем : учеб. для студентов учреждений сред. проф. образования, обучающихся по группе специальностей "Информатика и вычисл. техника". - 3-е изд., перераб. и доп. - М. : Форум, 2010. - 512 с. : ил. - (Профессиональное образование). - На обл. и корешке кн. загл.: Архитектура ЭВМ и вычислительные системы. - ISBN 978-5-91134-374-3 : 169.95. - Текст : непосредственный., 14 экз.

Дополнительная литература:

1. Паттерсон Дэвид. Архитектура компьютера и проектирование компьютерных систем = Computer Organization and Design. - 4-е изд. - СПб. : Питер, 2012. - 784 с. : ил. - (Классика Computer Science). - ISBN 978-5-459-00291-1 : 500.00. - Текст : непосредственный., 1 экз.
2. Гуров Валерий Валентинович. Архитектура микропроцессоров : учеб. пособие. - М. : Интернет-Университет Информационных Технологий : Бином. Лаборатория знаний, 2010. - 272 с. : ил., табл. - (Основы информационных технологий). - ISBN 978-5-9963-0267-3 : 246.40. - Текст : непосредственный., 1 экз.
3. Кудин А. В. Архитектура и операционные системы параллельных вычислительных систем : учебно-методические материалы по программе повышения квалификации «информационные технологии и компьютерное моделирование в прикладной математике» / Кудин А. В., Линёв А. В. - Нижний Новгород : ННГУ им. Н. И. Лобачевского, 2007. - 73 с. - Библиогр.: доступна в карточке книги, на сайте ЭБС Лань. - Книга из коллекции ННГУ им. Н. И. Лобачевского - Информатика. - Текст : электронный., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=730170&idb=0>.
4. Введение в принципы функционирования и применения современных мультиядерных архитектур (на примере Intel Xeon Phi) / Гергель В.П., Мееров И.Б., Бахраков С.И., Горшков А.В., Козинев Е.А., Линев А.В., Сиднев А.А., Сысоев А.В. - Москва : ИНТУИТ, 2016. - Текст : электронный., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=662900&idb=0>.

Программное обеспечение и Интернет-ресурсы (в соответствии с содержанием дисциплины):

1. Операционные системы семейства Microsoft Windows, лицензия по подписке Microsoft Imagine.
2. Браузер Google Chrome, предоставляется бесплатно на условиях лицензионных соглашений на программное обеспечение с открытым исходным кодом.
3. Среда разработки семейства Microsoft Visual Studio <https://visualstudio.microsoft.com/ru/>
4. ПО виртуализации Oracle VM VirtualBox (распространяется бесплатно в соответствии с условиями лицензии GPL версии 3).
5. ОС семейства Linux с установленным компилятором языка C (есть несколько бесплатных дистрибутивов).
6. Симулятор микроархитектур gem5 (распространяется бесплатно).

Интернет-ресурсы:

- <http://e-learning.unn.ru/>
- <https://e-lib.unn.ru>
- <https://intuit.ru/>

7. Материально-техническое обеспечение дисциплины (модуля)

Учебные аудитории для проведения учебных занятий, предусмотренных образовательной программой, оснащены мультимедийным оборудованием (проектор, экран), техническими средствами обучения, компьютерами.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду.

Программа составлена в соответствии с требованиями ОС ННГУ по направлению подготовки/специальности 02.03.02 - Фундаментальная информатика и информационные технологии.

Автор(ы): Линев Алексей Владимирович.

Заведующий кафедрой: Мееров Иосиф Борисович, кандидат технических наук.

Программа одобрена на заседании методической комиссии от 25.06.2025, протокол № Протокол №11.