

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Агафонов Александр Викторович

Должность: директор филиала

Дата подписания: 19.06.2025 15:33:27

Уникальный программный ключ:

23E608A9C8D11E158157408000000000

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
ЧЕБОКСАРСКИЙ ИНСТИТУТ ФИЛИАЛ) МОСКОВСКОГО ПОЛИТЕХНИЧЕСКОГО УНИВЕРСИТЕТА

Кафедра Информационных технологий
и систем управления



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Параллельное программирование»

(наименование дисциплины)

Направление подготовки	09.03.01 «Информатика и вычислительная техника» (код и наименование направления подготовки)
Направленность (профиль) подготовки	«Программное обеспечение вычислительной техники и автоматизированных систем» (наименование профиля подготовки)
Квалификация выпускника	бакалавр
Форма обучения	очная, заочная
Год начала обучения	2025

Рабочая программа дисциплины разработана в соответствии с:

- Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.01 – Информатика и вычислительная техника, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 929 от 19 сентября 2017 г. зарегистрированный в Минюсте 10 октября 2017 года, рег. номер 48489 (далее – ФГОС ВО);

- Учебным планом (очной, заочной форм обучения) по направлению подготовки 09.03.01 «Информатика и вычислительная техника».

Рабочая программа дисциплины включает в себя оценочные материалы для проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине (п.8 Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины)

Автор Пикина Наталия Евгеньевна, кандидат педагогических наук,
доцент кафедры информационных технологий и систем управления

(указать ФИО, ученую степень, ученое звание или должность)

Программа одобрена на заседании кафедры Информационных технологий и систем управления (протокол № 8 от 12.04.2025г).

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы (Цели освоения дисциплины)

1.1. Целями освоения дисциплины «Параллельное программирование» являются:

- ознакомление обучающихся с архитектурой виртуальных машин;
- с организацией управления и взаимодействия процессов;
- организацией управления в многопользовательских и многозадачных операционных системах;
- ознакомление обучающихся с администрированием современных операционных систем;
- ознакомление обучающихся с основами взаимодействия процессов по сети.

Задачами освоения дисциплины Параллельное программирование являются:

- владение навыками установки современных операционных систем ;
- научиться получению основных данных о текущем состоянии операционной системы с помощью консольных команд;
- овладение обучающимися базовыми навыками разработки сетевых приложений на сокетах;
- научиться получать информацию о состоянии операционной системы с выводом отчета на сценарных языках программирования встроенных в систему;
- научиться разрабатывать сценарии управления на встроенных в систему сценарных языках;
- научиться создавать группы пользователей и пользователей с различными ролями;
- приобретение обучающимися первичных навыков удаленного администрирования операционных систем.

В настоящее время нельзя назвать область человеческой деятельности, в которой в той или иной степени не использовалось программное обеспечение, для корректной работы которого необходимы Параллельное программирование.

1.2. Области профессиональной деятельности и (или) сферы профессиональной деятельности, в которых выпускники, освоившие программу, могут осуществлять профессиональную деятельность:

Об Связь, информационные и коммуникационные технологии (в сфере проектирования, разработки, внедрения и эксплуатации средств вычислительной техники и информационных систем, управления их жизненным циклом).

1.3. К основным задачам изучения дисциплины относится подготовка обучающихся к выполнению трудовых функций в соответствии с профессиональными стандартами:

Код и наименование профессионального стандарта	Обобщенные трудовые функции			Трудовые функции		
	код	наименование	уровень квалификации	наименование	код	уровень (подуровень) квалификации
06.001 Программист Профессиональный стандарт «Программист», утвержденный приказом Министерством труда и социальной защиты Российской Федерации от 18.11.2013 № 679н (зарегистрирован в Министерстве труда и социальной защиты Российской Федерации от 18 декабря 2013 г. №30635)	D	Разработка требований и проектирование программного обеспечения	6	Анализ требований к программному обеспечению	D/01.6	6
			6	Разработка технических спецификаций на программные компоненты и их взаимодействие	D/02.6	
			6	Проектирование программного обеспечения	D/03.6	
06.028 Системный программист Профессиональный стандарт «Системный программист», утв. Приказом Министерством труда и социальной защиты РФ от 29 сентября 2020 года N 678н	A	Разработка компонентов системных программных продуктов	6	Разработка драйверов устройств	A/01.6	6
			6	Разработка компиляторов, загрузчиков, сборщиков	A/02.6	6
				Разработка системных утилит	A/03.6	6
				Создание инструментальных средств программирования	A/04.6	6
06.015 Специалист по информационным системам Профессиональный стандарт «Специалист по информационным системам», утв. приказом Министерством труда и социальной защиты РФ 18 ноября 2014 г. №896н	C	Выполнение работ и управление работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы	6	Разработка модели бизнес-процессов заказчика	C/08.6	6
				Выявление требований к ИС	C/11.6	6
				Анализ требований	C/12.6	6
				Разработка архитектуры ИС	C/14.6	6
				Проектирование и дизайн ИС	C/16.6	6
				Разработка баз данных ИС	C/17.6	6
				Организационное и технологическое обеспечение кодирования на	C/18.6	6

Код и наименование профессионального стандарта	Обобщенные трудовые функции			Трудовые функции		
	код	наименование	уровень квалификации	наименование	код	уровень (подуровень) квалификации
				языках программирования		
				Организационное и технологическое обеспечение модульного тестирования ИС (верификации)	C/19.6	6
				Создание пользовательской документации к ИС	C/22.6	6
				Организация репозитория хранения данных о создании (модификации) и вводе ИС в эксплуатацию	C/40.6	6

1.4. Компетенции обучающегося, формируемые в результате освоения дисциплины

Наименование категории (группы) компетенций	Код и наименование компетенций	Код и наименование индикатора достижения компетенции	Перечень планируемых результатов обучения
Разработка, отладка, проверка работоспособности, модификация программного обеспечения	ПК-2 Способен разрабатывать требования и проектировать программное обеспечение	ПК-2.1 Выполняет анализ требований к программному обеспечению	<p><i>на уровне знаний:</i> знать основные результаты теории формальных языков;</p> <p><i>на уровне умений:</i> уметь создавать распознаватели, интерпретаторы и трансляторы информационных потоков; уметь находить и устранять проблем взаимодействия вычислительных процессов;</p> <p><i>на уровне навыков:</i> навыками программного построения распознавателей,</p>

			интерпретаторов и трансляторов информационных потоков
		ПК-2.2 Разрабатывает технические спецификации на программные компоненты и их взаимодействие	<p><i>на уровне знаний:</i> знать формальные модели основных вычислительных процессов, методы анализа вычислительных процессов;</p> <p><i>на уровне умений:</i> уметь моделировать работу алгоритмов взаимодействия процессов и ресурсов в работе вычислительной системы.</p> <p><i>на уровне навыков:</i> навыками использования стандартных инструментальных средств построения распознавателей и трансляторов в системном программировании</p>
		ПК-2.3 Проектирует программное обеспечение	<p><i>на уровне знаний:</i> теоретические основы методов проектирования и способы описания языков программирования, стандарты, используемые для языков программирования</p> <p><i>на уровне умений:</i> уметь разрабатывать алгоритмы, реализующие методы синтаксического анализа и перевода для наиболее часто используемых классов формальных грамматик</p> <p><i>на уровне навыков:</i> навыками практического анализа проблем взаимодействия вычислительных процессов.</p>

2. Место дисциплины в структуре ОПОП

Дисциплина Б1.Д(М).В.ДВ.6.1 «Параллельное программирование» реализуется в рамках вариативной части Блока 1 «Элективные дисциплины (модули)» программы бакалавриата.

Дисциплина «Параллельное программирование» преподаётся обучающимся по очной форме обучения – в 5-м семестре, по заочной форме – в 7-м семестре.

Дисциплина «Параллельное программирование» является промежуточным этапом формирования компетенций ПК-2 процессе освоения ОПОП.

Дисциплина «Параллельное программирование» основывается на знаниях, умениях и навыках, приобретенных при изучении дисциплин Учебная практика: ознакомительная практика, Информационные сети и коммуникации, Проектная деятельность, Учебная практика: научно-исследовательская работа (получение первичных навыков научно-исследовательской работы); Учебная практика: технологическая (проектно-технологическая) практика, Базы данных, Интеллектуальные системы, Основы систем искусственного интеллекта и является предшествующей для изучения дисциплин Системное программное обеспечение, Теория вычислительных процессов и языков программирования, Эргономика, Технический дизайн, Производственная практика: технологическая (проектно-технологическая) практика, Производственная практика: научно-исследовательская работа, Цифровая обработка сигналов, Архитектура вычислительных систем, Производственная практика: преддипломная практика, Государственная итоговая аттестация: подготовка к сдаче и сдача государственного экзамена, Государственная итоговая аттестация: выполнение и защита выпускной квалификационной работы.

Формой промежуточной аттестации знаний обучаемых по очной форме обучения является экзамен в 5-м семестре, по заочной форме – в 7-м семестре.

3. Объем дисциплины

очная форма обучения:

Вид учебной работы по дисциплине	Всего в з.е. и часах	Семестр 5 в часах
Общая трудоёмкость дисциплины	4 з.е. - 144 ак.час	144 ак.час
<i>Контактная работа - Аудиторные занятия</i>	49	49
<i>Лекции</i>	16	16
<i>Лабораторные занятия</i>	32	32
<i>Семинары, практические занятия</i>	-	-
<i>Консультация</i>	1	1
Самостоятельная работа	59	59
Курсовая работа (курсовой проект)	-	-
Вид промежуточной аттестации	Экзамен – 36 часов	Экзамен – 36 часов

заочная форма обучения:

Вид учебной работы по дисциплине	Всего в з.е. и часах	Семестр 7 в часах
Общая трудоёмкость дисциплины	4 з.е. - 144 ак.час	144 ак.час
<i>Контактная работа - Аудиторные занятия</i>	13	13
<i>Лекции</i>	6	6
<i>Лабораторные занятия</i>	6	6
<i>Семинары, практические занятия</i>	-	-
<i>Консультация</i>	1	1
<i>Самостоятельная работа</i>	122	122
Курсовая работа (курсовой проект)	-	-
Вид промежуточной аттестации	Экзамен – 9 часов	Экзамен – 9 часов

4. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий

4.1 Учебно-тематический план

Очная форма обучения

Тема (раздел)	Количество часов				Код индикатора достижений компетенции
	контактная работа			самостоятельная работа	
	лекции	лабораторные занятия	семинары и практические занятия		
Тема 1. Общие понятия дисциплины.	4	8	-	12	ПК-2.1 ПК-2.2 ПК-2.3
Тема 2. Системы с общей памятью. Низкоуровневое параллельное программирование..	4	8	-	12	ПК-2.1 ПК-2.2 ПК-2.3
Тема 3. Системы с общей памятью. OpenMP.	4	8	-	12	ПК-2.1 ПК-2.2 ПК-2.3
Тема 4. Message Passing Interface.	2	4	-	12	ПК-2.1 ПК-2.2 ПК-2.3
Тема 5. Модель акторов.	2	4	-	11	ПК-2.1 ПК-2.2 ПК-2.3
Консультации	1			-	ПК-2.1 ПК-2.2 ПК-2.3
Контроль (экзамен)	36				ПК-2.1 ПК-2.2 ПК-2.3
ИТОГО	49			59	

Заочная форма обучения

Тема (раздел)	Количество часов				Код индикатора достижений компетенции
	контактная работа			самостоятельная работа	
	лекции	лабораторные занятия	семинары и практические занятия		
Тема 1. Общие понятия дисциплины.	-	2	-	26	ПК-2.1 ПК-2.2 ПК-2.3
Тема 2. Системы с общей памятью. Низкоуровневое параллельное программирование..	-	2	-	24	ПК-2.1 ПК-2.2 ПК-2.3
Тема 3. Системы с общей памятью. OpenMP.	2	-	-	24	ПК-2.1 ПК-2.2 ПК-2.3
Тема 4. Message Passing Interface.	2	-	-	24	ПК-2.1 ПК-2.2 ПК-2.3
Тема 5. Модель акторов.	2	2	-	24	ПК-2.1 ПК-2.2 ПК-2.3
Консультации	1			-	ПК-2.1 ПК-2.2 ПК-2.3
Контроль (экзамен)	9				ПК-2.1 ПК-2.2 ПК-2.3
ИТОГО	13			122	

4.2. Содержание дисциплины

Тема 1. Общие понятия дисциплины

Общее представление о параллельном программировании и его значении в современных вычислениях.

Преимущества и проблемы параллельных вычислений, включая синхронизацию, гонки данных и дедлоки.

Введение в параллельные архитектуры: многозадачность, многопоточность, многоядерные процессоры.

Основные модели параллельного вычисления: SISD, SIMD, MIMD, SPMD.

Применение параллельного программирования в науке, промышленности и обработке больших данных.

Тема 2. Системы с общей памятью. Низкоуровневое параллельное программирование

Особенности параллельного программирования в системах с общей памятью.

Принципы многозадачности и многопоточности.

Введение в низкоуровневые аспекты параллельного программирования, использование потоков и синхронизации.

Проблемы, связанные с доступом к общей памяти: гонки данных, дедлоки и способы их предотвращения.

Основные средства синхронизации: мьютексы, семафоры, барьеры.

Тема 3. Системы с общей памятью. OpenMP

Обзор OpenMP как инструмента для параллельного программирования в системах с общей памятью.

Основные директивы и конструкции OpenMP: параллельные регионы, разделяемые и приватные переменные, циклы.

Основные функции OpenMP для многозадачных вычислений.

Оптимизация программ с использованием OpenMP, включая минимизацию синхронизации и уменьшение накладных расходов.

Тема 4. Message Passing Interface

Введение в MPI как стандарт для параллельного программирования в распределенных системах.

Основные функции MPI для обмена сообщениями между процессами: отправка и получение данных, синхронизация процессов.

Использование блоковых и неблоковых операций, ошибок и обработки сбоев в MPI.

Примеры практического использования MPI для параллельных вычислений и обработки больших данных.

Тема 5. Модель акторов

Общее представление о модели акторов как параллельной вычислительной модели.

Принципы работы с акторными системами: процессы как независимые акторы, взаимодействующие через сообщения.

Преимущества модели акторов для разработки масштабируемых и отказоустойчивых систем.

Применение модели акторов в современных распределенных и многозадачных приложениях.

Сравнение с другими моделями параллельного программирования.

5. Учебно-методическое обеспечение самостоятельной работы студентов

Самостоятельная работа проводится с целью: систематизации и закрепления полученных теоретических знаний и практических умений обучающихся; углубления и расширения теоретических знаний студентов; формирования умений использовать нормативную, правовую, справочную документацию, учебную и специальную литературу; развития познавательных способностей и активности обучающихся: творческой инициативы, самостоятельности, ответственности, организованности; формирование

самостоятельности мышления, способностей к саморазвитию, совершенствованию и самоорганизации; формирования профессиональных компетенций; развитию исследовательских умений студентов.

Формы и виды самостоятельной работы студентов: чтение основной и дополнительной литературы – самостоятельное изучение материала по рекомендуемым литературным источникам; работа с библиотечным каталогом, самостоятельный подбор необходимой литературы; работа со словарем, справочником; поиск необходимой информации в сети Интернет; конспектирование источников; реферирование источников; составление аннотаций к прочитанным литературным источникам; составление рецензий и отзывов на прочитанный материал; составление обзора публикаций по теме; составление и разработка терминологического словаря; составление хронологической таблицы; составление библиографии (библиографической картотеки); подготовка к различным формам текущей и промежуточной аттестации (к тестированию, контрольной работе, экзамену); выполнение домашних контрольных работ; самостоятельное выполнение практических заданий репродуктивного типа (ответы на вопросы, задачи, тесты; выполнение творческих заданий).

Технология организации самостоятельной работы обучающихся включает использование информационных и материально-технических ресурсов образовательного учреждения: библиотеку с читальным залом, компьютерные классы с возможностью работы в Интернет; аудитории (классы) для консультационной деятельности.

Перед выполнением обучающимися внеаудиторной самостоятельной работы преподаватель проводит консультирование по выполнению задания, который включает цель задания, его содержания, сроки выполнения, ориентировочный объем работы, основные требования к результатам работы, критерии оценки. Во время выполнения обучающимися внеаудиторной самостоятельной работы и при необходимости преподаватель может проводить индивидуальные и групповые консультации.

Контроль самостоятельной работы студентов предусматривает: соотнесение содержания контроля с целями обучения; объективность контроля; валидность контроля (соответствие предъявляемых заданий тому, что предполагается проверить); дифференциацию контрольно-измерительных материалов.

Формы контроля самостоятельной работы: просмотр и проверка выполнения самостоятельной работы преподавателем; организация самопроверки, взаимопроверки выполненного задания в группе; обсуждение результатов выполненной работы на занятии; проведение письменного опроса; проведение устного опроса; организация и проведение индивидуального собеседования; организация и проведение собеседования с группой.

Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Тема 1. Общие понятия дисциплины.	<ol style="list-style-type: none"> 1. Определение параллельного программирования и его значение в современном программировании. 2. Основные принципы параллельного программирования: разделение задач, синхронизация, управление потоками. 3. Различие между параллелизмом и конкуренцией. 4. Параллельные вычисления: преимущества и недостатки. 5. Архитектура параллельных систем: многопроцессорные и многопоточные системы. 6. Основные модели параллельного программирования: данные и процессы. 7. Примеры применения параллельного программирования в реальных задачах. 8. Влияние параллельного программирования на производительность приложений. 9. Будущее параллельного программирования: новые технологии и подходы. 10. Сравнение параллельного программирования с другими парадигмами (например, распределенное программирование). 	<p>Изучение литературы по основам параллельного программирования. Проведение анализа примеров применения параллельного программирования в науке и бизнесе. Просмотр и конспектирование материалов о современных подходах к параллельным вычислениям.</p>
Тема 2. Системы с общей памятью. Низкоуровневое параллельное программирование..	<ol style="list-style-type: none"> 1. Определение систем с общей памятью и их архитектура. 2. Основные концепции низкоуровневого параллельного программирования. 3. Синхронизация потоков: mutexes, семафоры и другие механизмы. 4. Проблемы, возникающие в системах с общей памятью: гонки, взаимные блокировки. 5. Примеры низкоуровневого параллельного программирования на C/C++. 6. Параллелизм в системах с общей памятью: подходы и техники. 7. Использование атомарных операций для управления данными. 8. Производительность систем с общей памятью: анализ и оптимизация. 9. Роль операционных систем в управлении потоками и памятью. 10. Будущее низкоуровневого параллельного программирования: новые технологии и подходы. 	<p>Исследование примеров использования синхронизации в многопоточных приложениях. Выполнение практических задач по параллельному программированию на C/C++. Просмотр материалов о механизмах синхронизации.</p>
Тема 3. Системы с общей памятью. OpenMP.	<ol style="list-style-type: none"> 1. Определение OpenMP и его назначение в параллельном программировании. 2. Основные конструкции OpenMP: директивы, функции и переменные среды. 3. Параллелизация циклов с помощью OpenMP: примеры и техники. 4. Управление потоками и синхронизация в OpenMP. 5. Реализация задач параллельного программирования с использованием OpenMP. 6. Оптимизация производительности приложений с OpenMP. 7. Примеры успешного применения OpenMP в научных и инженерных задачах. 8. Сравнение OpenMP с другими параллельными библиотеками (например, Pthreads). 	<p>Выполнение практических задач на параллелизацию с использованием OpenMP. Исследование примеров приложений, использующих OpenMP. Просмотр материалов о параллельном программировании с OpenMP.</p>

	<p>9. Ограничения и проблемы, связанные с использованием OpenMP.</p> <p>10. Будущее OpenMP: новые функции и развитие стандартов.</p>	
Тема 4. Message Passing Interface.	<p>1. Определение Message Passing Interface (MPI) и его назначение.</p> <p>2. Основные концепции и принципы работы MPI.</p> <p>3. Директивы MPI: отправка и получение сообщений.</p> <p>4. Коллективные операции в MPI: broadcast, scatter, gather и другие.</p> <p>5. Примеры реализации параллельных приложений с использованием MPI.</p> <p>6. Производительность MPI: оценка и оптимизация.</p> <p>7. Сравнение MPI с другими подходами к параллельному программированию.</p> <p>8. Ограничения и сложности, связанные с использованием MPI.</p> <p>9. Применение MPI в высокопроизводительных вычислениях и научных исследованиях.</p> <p>10. Будущее MPI: новые подходы и расширения.</p>	<p>Выполнение практических задач на параллельное программирование с использованием MPI.</p> <p>Исследование примеров успешного применения MPI в научных и инженерных задачах. Просмотр материалов о параллельном программировании с MPI.</p>
Тема 5. Модель акторов.	<p>1. Определение модели акторов и её назначение в параллельном программировании.</p> <p>2. Основные концепции модели акторов: актеры, сообщения, асинхронность.</p> <p>3. Преимущества и недостатки модели акторов по сравнению с другими парадигмами.</p> <p>4. Реализация модели акторов в различных языках программирования (например, Erlang, Akka).</p> <p>5. Примеры приложений, использующих модель акторов.</p> <p>6. Сравнение модели акторов с традиционными потоками и процессами.</p> <p>7. Проблемы и вызовы, связанные с использованием модели акторов.</p> <p>8. Применение модели акторов в распределённых системах.</p> <p>9. Будущее модели акторов: новые подходы и технологии.</p> <p>10. Сравнение различных реализаций модели акторов.</p>	<p>Выполнение практических задач на реализацию модели акторов. Исследование примеров успешного применения модели акторов в реальных проектах. Просмотр и обсуждение материалов о параллельном программировании с использованием модели акторов.</p>

Шкала оценивания

Шкала оценивания	Критерии оценивания
«Отлично»	Обучающийся глубоко и содержательно раскрывает тему самостоятельной работы, не допустив ошибок. Ответ носит развернутый и исчерпывающий характер
«Хорошо»	Обучающийся в целом раскрывает тему самостоятельной работы, однако ответ хотя бы на один из них не носит развернутого и исчерпывающего характера
«Удовлетворительно»	Обучающийся в целом раскрывает тему самостоятельной работы и допускает ряд неточностей, фрагментарно раскрывает содержание теоретических вопросов или их раскрывает содержательно, но допуская значительные неточности.

«Неудовлетворительно»	Обучающийся не владеет выбранной темой самостоятельной работы
-----------------------	---

6. Оценочные материалы (фонд оценочных средств) для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины

6.1. Паспорт фонда оценочных средств

№	Контролируемые разделы (темы) дисциплины	Код и наименование компетенции	Индикатор достижения компетенции	Наименование оценочного средства
1.	Тема 1. Общие понятия дисциплины.	ПК-2. Способен разрабатывать требования и проектировать программное обеспечение	ПК-2.1 Выполняет анализ требований к программному обеспечению ПК-2.2. Разрабатывает технические спецификации на программные компоненты и их взаимодействие ПК-2.3 Проектирует программное обеспечение	Опрос, тест, доклад, реферат, экзамен
2.	Тема 2. Системы с общей памятью. Низкоуровневое параллельное программирование..	ПК-2. Способен разрабатывать требования и проектировать программное обеспечение	ПК-2.1 Выполняет анализ требований к программному обеспечению ПК-2.2. Разрабатывает технические спецификации на программные компоненты и их взаимодействие ПК-2.3 Проектирует программное обеспечение	Опрос, тест, доклад, реферат, экзамен
3.	Тема 3. Системы с общей памятью. OpenMP.	ПК-2. Способен разрабатывать требования и проектировать программное обеспечение	ПК-2.1 Выполняет анализ требований к программному обеспечению ПК-2.2. Разрабатывает технические спецификации на программные компоненты и их	Опрос, тест, доклад, реферат, экзамен

			взаимодействие ПК-2.3 Проектирует программное обеспечение	
4.	Тема 4. Message Passing Interface.	ПК-2. Способен разрабатывать требования и проектировать программное обеспечение	ПК-2.1 Выполняет анализ требований к программному обеспечению ПК-2.2. Разрабатывает технические спецификации на программные компоненты и их взаимодействие ПК-2.3 Проектирует программное обеспечение	Опрос, тест, доклад, реферат, экзамен
5.	Тема 5. Модель акторов.	ПК-2. Способен разрабатывать требования и проектировать программное обеспечение	ПК-2.1 Выполняет анализ требований к программному обеспечению ПК-2.2. Разрабатывает технические спецификации на программные компоненты и их взаимодействие ПК-2.3 Проектирует программное обеспечение	Опрос, тест, доклад, реферат, экзамен

Этапы формирования компетенций в процессе освоения ОПОП прямо связаны с местом дисциплин в образовательной программе. Каждый этап формирования компетенции, характеризуется определенными знаниями, умениями и навыками и (или) опытом профессиональной деятельности, которые оцениваются в процессе текущего контроля успеваемости, промежуточной аттестации по дисциплине (практике) и в процессе итоговой аттестации.

Дисциплина «Параллельное программирование» является промежуточным этапом комплекса дисциплин, в ходе изучения которых у студентов формируются компетенции ПК-2.

Формирование компетенции ПК-2 начинается с изучения дисциплины Учебная практика: ознакомительная практика, Информационные сети и коммуникации, «Проектная деятельность», Учебная практика: научно-исследовательская работа (получение первичных навыков научно-исследовательской работы), Учебная практика: технологическая (проектно-технологическая) практика, «Базы данных», «Интеллектуальные системы», «Основы систем искусственного интеллекта».

Завершается работа по формированию у студентов указанных компетенций в ходе предмета «Архитектура вычислительных систем», Производственная практика: преддипломная практика.

Итоговая оценка сформированности компетенций ПК-2 определяется в период Государственная итоговая аттестация: подготовка к сдаче и сдача государственного экзамена, Государственная итоговая аттестация: выполнение и защита выпускной квалификационной работы.

В процессе изучения дисциплины, компетенции также формируются поэтапно.

Основными этапами формирования ПК-2 при изучении дисциплины Б1.Д(М).В.ДВ.6.1 «Параллельное программирование» является последовательное изучение содержательно связанных между собой тем учебных занятий. Изучение каждой темы предполагает овладение студентами необходимыми дескрипторами (составляющими) компетенций. Для оценки уровня сформированности компетенций в процессе изучения дисциплины предусмотрено проведение текущего контроля успеваемости по темам (разделам) дисциплины и промежуточной аттестации по дисциплине – экзамен.

6.2. Контрольные задания и материалы, необходимые для оценки знаний, умений и навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

6.2.1. Контрольные вопросы по темам (разделам) для опроса на занятиях

Тема (раздел)	Вопросы
Тема 1. Общие понятия дисциплины.	ПК-2 1. Понятие параллельное программирование 2. Архитектура ВС. Классификация по Флинну. 3. Архитектура однопроцессорной машины. 4. Архитектура с разделяемой памятью. 5. Архитектура с распределенной памятью. 6. Пути достижения параллелизма. 7. Параллелизм на уровне команд, потоков, процессов. 8. Анализ эффективности параллельных вычислений. 9. Закон Амдала. 10. Проблемы разработки параллельных приложений. 11. Проблема гонки данных. 12. Проблемы синхронизации. 13. Проблемы кешируемой памяти. 14. Модели параллельных приложений.
Тема 2. Системы с общей памятью. Низкоуровневое параллельное программирование..	ПК-2 1. Понятие потока. 2. Структура потока. 3. Состояния потоков. 4. Что такое POSIX.

	<ol style="list-style-type: none"> 5. Создание, контроль и завершение выполнения потоков. 6. Отличие между join и detach. 7. Передача параметров в поток 8. Приостановление потока. 9. Приоритеты потоков. 10. Локальное хранилище потока. 11. Пул потоков. 12. Понятие критическая секция. 13. Средства синхронизации. 14. Средства для взаимного исключения. 15. Monitor. 16. Mutex. 17. Сигнальные сообщения. 18. Семафоры. 19. Атомарные операции 20. Конкурентные коллекции 21. Работа с задачами
<p>Тема 3. Системы с общей памятью. OpenMP.</p>	<p>ПК-2</p> <ol style="list-style-type: none"> 1. Понятие Fork-Join параллелизма. 2. Директивы и функции. 3. Параллельные и последовательные области. 4. Модель данных. 5. Распределение работы. 6. Синхронизация. 7. Барьер 8. Критические секции 9. Гонка данных. Директива atomic. 10. Замки (locks). 11. Согласованный образ памяти. Директива flush. 12. Использование OpenMP
<p>Тема 4. Message Passing Interface.</p>	<p>ПК-2</p> <ol style="list-style-type: none"> 1. Основные понятия MPI. 2. Структура программы MPI. 3. Передача сообщений между двумя процессами. 4. Основные типы операций передачи данных. 5. Неблокирующий обмен. 6. Блокирующий обмен. 7. Выполнение операций приема и передачи одной функцией 8. Базовые типы данных. 9. Пользовательские типы данных. 10. Коллективные операции. Коммуникаторы. 11. Виртуальные топологии
<p>Тема 5. Модель акторов.</p>	<p>ПК-2</p> <ol style="list-style-type: none"> 1. Основная концепция. 2. Строение актора. 3. Создание экземпляров или систем акторов 4. Управление необработанными сообщениями 5. Поведение и состояние актора 6. Akka. ActorSystem. 7. ActorRef, почтовый ящик и актор 8. Akka. Диспечеры. 9. Akka. Акторы и сеть

Шкала оценивания ответов на вопросы

Шкала оценивания	Критерии оценивания
«Отлично»	Обучающийся глубоко и содержательно раскрывает ответ на каждый теоретический вопрос, не допустив ошибок. Ответ носит развернутый и исчерпывающий характер.
«Хорошо»	Обучающийся в целом раскрывает теоретические вопросы, однако ответ хотя бы на один из них не носит развернутого и исчерпывающего характера.
«Удовлетворительно»	Обучающийся в целом раскрывает теоретические вопросы и допускает ряд неточностей, фрагментарно раскрывает содержание теоретических вопросов или их раскрывает содержательно, но допуская значительные неточности.
«Неудовлетворительно»	Обучающийся не знает ответов на поставленные теоретические вопросы.

6.2.2. Темы для докладов

Тема (раздел)	Вопросы
Тема 1. Общие понятия дисциплины.	<p>ПК-2</p> <ol style="list-style-type: none"> 1. Введение в параллельное программирование: зачем оно нужно и какие задачи решает. 2. Основные концепции параллельного программирования: многозадачность, многопоточность, синхронизация. 3. Отличие параллельного программирования от многозадачности и асинхронного программирования. 4. Преимущества и недостатки параллельных вычислений. 5. Основные подходы к параллельному программированию: data parallelism, task parallelism. 6. Архитектуры параллельных вычислений: многопроцессорные системы, многоядерные процессоры. 7. Стандартные модели и библиотеки для параллельного программирования. 8. Проблемы, связанные с параллельным программированием: гонки, deadlock, привязка потоков.
Тема 2. Системы с общей памятью. Низкоуровневое параллельное программирование..	<p>ПК-2</p> <ol style="list-style-type: none"> 1. Преимущества и недостатки систем с общей памятью в параллельном программировании. 2. Как работает система с общей памятью: архитектура и принципы взаимодействия. 3. Особенности многозадачности и многопоточности в системах с общей памятью. 4. Основы низкоуровневого параллельного программирования: работа с потоками, синхронизация, блокировки. 5. Параллельное программирование на уровне ядра операционной системы. 6. Работа с разделяемыми ресурсами в многозадачных средах. 7. Параллельные структуры данных и алгоритмы для систем с общей памятью.

	8. Примеры реализации низкоуровневого параллельного кода на C/C++.
Тема 3. Системы с общей памятью. OpenMP.	<p>ПК-2</p> <ol style="list-style-type: none"> 1. Введение в OpenMP: основные концепции и структура. 2. Принципы работы OpenMP: директивы, потоки, синхронизация. 3. Модели параллелизма в OpenMP: data parallelism, task parallelism. 4. Использование директив OpenMP для параллелизации циклов и функций. 5. Как эффективно использовать разделяемую память и синхронизацию в OpenMP. 6. Оптимизация параллельных программ с использованием OpenMP. 7. Оценка производительности программ с OpenMP: профилирование и измерения. 8. Проблемы и ограничения при использовании OpenMP: эффективность, масштабируемость, производительность.
Тема 4. Message Passing Interface.	<p>ПК-2</p> <ol style="list-style-type: none"> 1. Введение в Message Passing Interface (MPI): что это такое и для чего используется. 2. Основные концепции MPI: процессы, сообщения, буферы, коммуникация. 3. Различие между системами с общей памятью и MPI: когда и почему использовать MPI. 4. Основы работы с MPI: синхронная и асинхронная коммуникация. 5. Как использовать MPI для параллельных вычислений: передача данных между процессами. 6. Основные функции и библиотеки MPI для организации обмена сообщениями. 7. Пример реализации параллельных программ с использованием MPI. 8. Проблемы и решения в MPI: производительность, маршрутизация сообщений, синхронизация процессов.
Тема 5. Модель акторов.	<p>ПК-2</p> <ol style="list-style-type: none"> 1. Введение в модель акторов: что это такое и как она используется в параллельных системах. 2. Основные принципы модели акторов: независимые сущности, обмен сообщениями, асинхронность. 3. Моделирование параллельных вычислений с использованием акторов. 4. Как модель акторов помогает избежать общих проблем параллельного программирования: гонок, deadlock. 5. Преимущества и ограничения модели акторов в контексте многозадачных систем. 6. Реализация акторов в различных языках программирования (например, Erlang, Akka в Scala). 7. Пример реализации модели акторов в многозадачной системе. 8. Применение модели акторов в распределенных и облачных вычислениях.

Шкала оценивания

Шкала оценивания	Критерии оценивания
------------------	---------------------

«Отлично»	Обучающийся глубоко и содержательно раскрывает тему доклада, не допустив ошибок. Ответ носит развернутый и исчерпывающий характер.
«Хорошо»	Обучающийся в целом раскрывает тему доклада, однако ответ хотя бы на один из них не носит развернутого и исчерпывающего характера.
«Удовлетворительно»	Обучающийся в целом раскрывает тему доклада и допускает ряд неточностей, фрагментарно раскрывает содержание теоретических вопросов или их раскрывает содержательно, но допуская значительные неточности.
«Неудовлетворительно»	Обучающийся не владеет выбранной темой

6.2.3. Оценочные средства остаточных знаний (тест)

ПК-2.

1. Что такое параллельное программирование?

- 1) Программирование, где задачи выполняются последовательно
- 2) Программирование, где задачи выполняются одновременно на нескольких процессорах
- 3) Программирование, где данные обрабатываются поочередно
- 4) Программирование с использованием единственного процессора

2. Какой из этих методов параллельного программирования относится к разделению задач?

- 1) Разделение данных
- 2) Создание многозадачных процессов
- 3) Использование однопоточности
- 4) Синхронизация потоков

3. Какое из утверждений верно относительно многозадачности?

- 1) Многозадачность — это выполнение нескольких задач одновременно на одном процессоре
- 2) Многозадачность не требует использования дополнительных процессоров
- 3) Многозадачность не влияет на производительность системы
- 4) Многозадачность используется для уменьшения времени выполнения одной задачи

4. Что такое поток в контексте параллельного программирования?

- 1) Независимая единица выполнения внутри программы
- 2) Совокупность всех выполняемых задач в программе
- 3) Объект для хранения данных в программе
- 4) Процесс выполнения программы на отдельном устройстве

5. Какой из следующих подходов является примером параллельной обработки данных?

- 1) Многозадачность
- 2) Многопроцессорная обработка данных
- 3) Оптимизация одного потока

4) Синхронизация потоков

6. Что такое синхронизация в параллельном программировании?

- 1) Процесс распределения задач между процессами
- 2) Процесс управления доступом к общим ресурсам
- 3) Процесс разделения задач на более мелкие
- 4) Процесс запуска нескольких потоков одновременно

7. Какую задачу решает алгоритм "мастера-работника"?

- 1) Разделение данных между процессами
- 2) Синхронизация потоков
- 3) Управление очередями задач
- 4) Обработка ошибок параллельных процессов

8. Что такое race condition в параллельном программировании?

- 1) Ситуация, когда два потока одновременно пытаются получить доступ к общим данным, что вызывает ошибку
- 2) Ситуация, когда один поток обрабатывает несколько задач
- 3) Ситуация, когда программа завершает выполнение
- 4) Ситуация, когда потоки синхронизированы

9. Что такое deadlock?

- 1) Ситуация, когда два или более потока ожидают друг друга, и ни один из них не может продолжить выполнение
- 2) Ситуация, когда один поток завершает выполнение задачи
- 3) Ситуация, когда потоки синхронизированы
- 4) Ситуация, когда программа работает без ошибок

10. Какая модель параллельного программирования используется для разделения задачи на подзадачи?

- 1) Модель "мастера-работника"
- 2) Модель "параллельных потоков"
- 3) Модель "синхронизации потоков"
- 4) Модель "параллельных данных"

11. Что такое многозадачность на уровне процессов?

- 1) Запуск нескольких потоков в одном процессе
- 2) Одновременное выполнение нескольких процессов, каждый из которых может иметь свои потоки
- 3) Синхронизация работы потоков внутри одного процесса
- 4) Разделение ресурсов между потоками

12. Что такое распределенные вычисления?

- 1) Вычисления, выполняемые одновременно на разных машинах
- 2) Вычисления, выполняемые в одном процессе

- 3) Вычисления, выполняемые на одном процессоре
- 4) Вычисления, выполняемые на одном компьютере с несколькими ядрами

13. Какой из этих подходов позволяет уменьшить время выполнения задачи с помощью параллельных вычислений?

- 1) Разделение задачи на несколько потоков
- 2) Использование единственного процесса
- 3) Ограничение использования процессоров
- 4) Одновременное выполнение всех задач в одном потоке

14. Что такое параллельный алгоритм?

- 1) Алгоритм, который выполняется на одном процессе
- 2) Алгоритм, который разбивает задачу на несколько частей и выполняет их одновременно
- 3) Алгоритм, который использует только один поток
- 4) Алгоритм, который выполняет задачи по очереди

15. Что такое критическая секция в параллельном программировании?

- 1) Место в коде, где доступ к общим ресурсам должен быть ограничен
- 2) Место, где выполняются все параллельные вычисления
- 3) Место, где процессы могут быть синхронизированы
- 4) Место, где выполняется только один поток

16. Что такое алгоритм разделяй и властвуй?

- 1) Алгоритм, который решает задачи, деля их на подзадачи и решая их параллельно
- 2) Алгоритм, который решает задачи поочередно
- 3) Алгоритм, который использует только один поток для выполнения задачи
- 4) Алгоритм, который разделяет задачи по процессам

17. Что такое балансировка нагрузки в параллельном программировании?

- 1) Разделение задачи между процессами, чтобы не один процесс не был перегружен
- 2) Использование одного процесса для выполнения всех задач
- 3) Объединение всех процессов в один поток
- 4) Синхронизация потоков для равномерного выполнения

18. Какой подход используется для решения проблемы гонки потоков?

- 1) Использование мьютексов для синхронизации доступа к данным
- 2) Использование одного потока для обработки всех данных
- 3) Использование нескольких процессов для выполнения задачи
- 4) Использование одного процессора для всех задач

19. Что такое блокировка (lock) в параллельном программировании?

- 1) Метод синхронизации доступа к разделяемым ресурсам между потоками
- 2) Метод завершения выполнения программы
- 3) Метод разделения данных между потоками
- 4) Метод преобразования данных

20. Что такое асинхронные вычисления?

- 1) Вычисления, которые выполняются без ожидания результата другого процесса
- 2) Вычисления, которые выполняются синхронно с другими процессами
- 3) Вычисления, которые выполняются поочередно
- 4) Вычисления, которые используют один поток для выполнения задач

21. Что такое модель памяти в параллельном программировании?

- 1) Способ представления и доступа к данным в многозадачной среде
- 2) Способ распределения задач между процессами
- 3) Способ синхронизации потоков
- 4) Способ разбиения данных на подзадачи

22. Что такое спинлок в параллельном программировании?

- 1) Механизм синхронизации, при котором поток постоянно проверяет наличие блокировки
- 2) Механизм, который завершает выполнение потока
- 3) Механизм для параллельного выполнения всех задач
- 4) Механизм для синхронизации входных данных

23. Что такое фабрика потоков (thread pool)?

- 1) Набор потоков, готовых к выполнению задач по мере их поступления
- 2) Один поток, выполняющий задачи последовательно
- 3) Набор процессов, которые выполняют параллельные вычисления
- 4) Механизм синхронизации потоков

24. Что такое шардирование в распределенных системах?

- 1) Разделение данных на части, которые могут быть обработаны на различных машинах
- 2) Разделение задач на подзадачи внутри одного потока
- 3) Разделение потока на несколько параллельных потоков
- 4) Разделение вычислений по процессам

25. Что такое стратегия разделения работы в параллельных вычислениях?

- 1) Стратегия, при которой задачи разделяются между процессами для оптимальной производительности

- 2) Стратегия использования одного процессора для всех вычислений
- 3) Стратегия выполнения задач поочередно
- 4) Стратегия синхронизации потоков

26. Что такое синхронизация на основе событий?

- 1) Механизм, при котором потоки синхронизируются через ожидание событий
- 2) Механизм для блокировки доступа к ресурсам
- 3) Механизм для синхронизации по времени
- 4) Механизм для ускорения выполнения задач

27. Что такое разделяемая память в параллельных вычислениях?

- 1) Память, доступная для всех потоков или процессов в системе
- 2) Память, используемая только одним потоком
- 3) Память, доступная только для одного процесса
- 4) Память, разделенная между процессами по очереди

28. Какой тип параллелизма используется в многозадачных системах?

- 1) Параллелизм на уровне процессов
- 2) Параллелизм на уровне потоков
- 3) Параллелизм на уровне памяти
- 4) Параллелизм на уровне данных

29. Что такое межпроцессорная синхронизация?

- 1) Синхронизация процессов на разных процессорах
- 2) Синхронизация потоков внутри одного процесса
- 3) Синхронизация потоков на одном процессоре
- 4) Синхронизация потоков между несколькими потоками

30. Что такое барьер синхронизации в параллельном программировании?

- 1) Техника, при которой все потоки ждут, пока не достигнут общей точки перед продолжением выполнения
- 2) Техника, при которой потоки выполняются без синхронизации
- 3) Техника, при которой поток завершает выполнение задачи
- 4) Техника синхронизации на основе времени

31. Что такое динамическая балансировка нагрузки?

- 1) Распределение задач между процессами по мере их поступления
- 2) Распределение задач между процессами заранее
- 3) Синхронизация задач для равномерного распределения
- 4) Использование одного процесса для всех вычислений

32. Что такое параллельное решение задач с использованием очередей?

1) Использование очередей для упорядочивания задач, которые будут выполняться параллельно

- 2) Использование одной очереди для выполнения задач последовательно
- 3) Использование очередей для синхронизации потоков
- 4) Использование одной очереди для обработки данных

33. Что такое распределенная память?

1) Память, которая используется в распределенных вычислительных системах

- 2) Память, которая доступна только одному процессу
- 3) Память, используемая для синхронизации потоков
- 4) Память, которая доступна только для одного потока

34. Что такое масштабируемость в параллельных системах?

1) Способность системы работать эффективно при увеличении числа процессов или потоков

- 2) Способность системы работать только с одним процессом
- 3) Способность системы выполнять задачи последовательно
- 4) Способность системы уменьшать время выполнения задачи

35. Какой тип параллельности используется в многозадачных системах с несколькими процессорами?

- 1) Параллельность на уровне данных
- 2) Параллельность на уровне потоков
- 3) Параллельность на уровне процессов
- 4) Параллельность на уровне синхронизации

36. Что такое оптимизация параллельных вычислений?

1) Процесс улучшения производительности при выполнении параллельных вычислений

- 2) Процесс разделения задачи на несколько потоков
- 3) Процесс синхронизации потоков для уменьшения времени выполнения
- 4) Процесс использования одного процессора для всех задач

37. Что такое блокировка потока?

- 1) Ситуация, когда поток ожидает освобождения ресурса
- 2) Ситуация, когда поток завершает выполнение
- 3) Ситуация, когда потоки выполняются одновременно
- 4) Ситуация, когда поток синхронизирован с другими

38. Что такое параллельная сортировка?

1) Алгоритм сортировки, который использует несколько потоков для упорядочивания данных

- 2) Алгоритм сортировки, который работает только на одном потоке
- 3) Алгоритм сортировки, который использует многозадачность

4) Алгоритм сортировки, который обрабатывает данные поочередно

39. Что такое использование несколько потоков для обработки больших данных?

- 1) Разделение данных между потоками для ускорения их обработки
- 2) Использование одного потока для обработки всех данных
- 3) Использование многозадачности для обработки данных
- 4) Использование одного процесса для всех данных

40. Что такое ускорение параллельных вычислений?

- 1) Уменьшение времени выполнения задачи за счет параллельной обработки
- 2) Уменьшение времени выполнения задачи за счет использования одного потока
- 3) Уменьшение времени выполнения задачи за счет использования одного процессора
- 4) Уменьшение времени выполнения задачи за счет синхронизации

41. Какой из этих алгоритмов наиболее эффективен при выполнении параллельных вычислений?

- 1) Алгоритм с минимальной степенью параллельности
- 2) Алгоритм с максимальной степенью параллельности
- 3) Алгоритм, выполняющий задачи последовательно
- 4) Алгоритм, использующий один поток для выполнения задач

42. Что такое балансировка потоков?

- 1) Распределение нагрузки между несколькими потоками для оптимальной работы
- 2) Синхронизация потоков для завершения задач
- 3) Использование одного потока для выполнения задач
- 4) Запуск всех потоков в одинаковое время

43. Что такое параллельное исполнение на GPU?

- 1) Использование графических процессоров для выполнения параллельных вычислений
- 2) Использование центральных процессоров для выполнения параллельных вычислений
- 3) Использование многозадачности для ускорения вычислений
- 4) Использование нескольких потоков для обработки данных

44. Что такое адаптивное параллельное программирование?

- 1) Программирование, которое подстраивает выполнение задачи в зависимости от изменений в системе
- 2) Программирование, которое выполняет задачи по очереди
- 3) Программирование, которое использует только один процесс

4) Программирование, которое использует одно ядро процессора

45. Что такое анализ производительности параллельных систем?

- 1) Оценка эффективности параллельных вычислений
- 2) Оценка скорости выполнения задачи на одном процессе
- 3) Оценка качества синхронизации потоков
- 4) Оценка использования памяти

Ключ к тесту:

1.2	2.1	3.1	4.1	5.2	6.2	7.1	8.1	9.1
10.1	11.2	12.1	13.1	14.2	15.1	16.1	17.1	18.1
19.1	20.1	21.1	22.1	23.1	24.1	25.1	26.1	27.1
28.1	29.1	30.1	31.1	32.1	33.1	34.1	35.1	36.1
37.1	38.1	39.1	40.1	41.2	42.1	43.1	44.1	45.1

Шкала оценивания результатов тестирования

% верных решений (ответов)	Шкала оценивания
85 - 100	отлично
70 - 84	хорошо
50- 69	удовлетворительно
0 - 49	неудовлетворительно

6.2.4. Примеры задач при разборе конкретных ситуаций

Тема 1. Общие понятия дисциплины.

ПК-2.

1. Написать программу на C/C++, которая реализует простой алгоритм параллельной сортировки (например, сортировка слиянием) с использованием потоков. Описать, как программа делит задачи между потоками и как происходит их синхронизация.

2. Провести эксперимент с различными алгоритмами (например, сортировка пузырьком и быстрая сортировка) для оценки их производительности в однопоточном и многопоточном режимах. Подготовить отчет с графиками, показывающими время выполнения в зависимости от количества данных.

3. Проанализировать задачу параллельного вычисления чисел Фибоначчи с использованием рекурсии и потоков. Написать программу, которая вычисляет n-е число Фибоначчи параллельно, и сравнить производительность с последовательным решением.

Тема 2. Системы с общей памятью. Низкоуровневое параллельное программирование..

ПК-2.

1. Реализовать программу на C/C++, которая использует мьютексы для предотвращения гонки данных в многопоточном приложении. Описать, как мьютексы обеспечивают синхронизацию потоков.

2. Написать программу, которая моделирует работу банка с несколькими счетами, позволяя нескольким потокам одновременно выполнять операции. Использовать механизмы синхронизации для обеспечения корректности операций.

3. Провести эксперимент по измерению времени выполнения задач в системах с общей памятью при использовании различных механизмов синхронизации (мьютексы, семафоры, атомарные операции). Подготовить отчет с выводами о производительности.

Тема 3. Системы с общей памятью. OpenMP.

ПК-2.

1. Написать программу на C/C++, использующую OpenMP для параллелизации вычисления суммы элементов массива. Описать, как директивы OpenMP влияют на разделение работы между потоками.

2. Создать программу, которая использует OpenMP для распараллеливания цикла, вычисляющего факториал числа. Объяснить, какие преимущества это дает по сравнению с последовательным вычислением.

3. Провести анализ производительности программы с использованием OpenMP для различных размеров входных данных. Подготовить отчет, сравнивающий время выполнения в зависимости от количества потоков и размера массива.

Тема 4. Message Passing Interface.

ПК-2.

1. Написать программу на C/C++, использующую MPI для распараллеливания задачи суммирования массива. Описать, как данные передаются между процессами и как результат собирается.

2. Создать программу, которая использует MPI для решения задачи "Параллельной сортировки". Описать, как осуществляется обмен сообщениями между процессами и как достигается синхронизация.

3. Провести эксперимент по сравнению производительности программ, использующих MPI и OpenMP для решения одной и той же задачи. Подготовить отчет с выводами о том, в каких ситуациях лучше использовать каждую из технологий.

Тема 5. Модель акторов.

ПК-2.

1. Написать программу на языке, поддерживающем модель акторов (например, Erlang или Akka на Scala), для реализации простой системы обмена сообщениями между актерами. Описать, как актеры взаимодействуют друг с другом и как достигается асинхронность.

2. Создать приложение, использующее модель акторов для обработки запросов в веб-сервере. Объяснить, как модель акторов позволяет масштабировать сервер и управлять состоянием.

3. Провести анализ производительности системы на основе модели акторов по сравнению с традиционными потоками. Подготовить отчет, в котором будет описано, как модель акторов справляется с высокой нагрузкой и какие есть ограничения.

Шкала оценивания

Шкала оценивания	Критерии оценивания
«Отлично»	обучающийся ясно изложил условие задачи, решение обосновал
«Хорошо»	обучающийся ясно изложил условие задачи, но в обосновании решения имеются сомнения;
«Удовлетворительно»	обучающийся изложил решение задачи, но обосновал его формулировками обыденного мышления;
«Неудовлетворительно»	обучающийся не уяснил условие задачи, решение не обосновал либо не сдал работу на проверку (в случае проведения решения задач в письменной форме).

6.2.5. Темы для рефератов

Тема (раздел)	Вопросы
Тема 1. Общие понятия дисциплины.	<p>ПК-2</p> <ol style="list-style-type: none"> 1. Определение параллельного программирования и его значение в современных вычислениях. 2. Основные принципы параллелизма: разделение задач, синхронизация и управление потоками. 3. Различие между параллелизмом и конкуренцией: понятия и примеры. 4. Архитектуры параллельных вычислительных систем: многопроцессорные и многопоточные системы. 5. Преимущества и недостатки параллельного программирования. 6. Примеры алгоритмов, использующих параллельное программирование: сортировка, поиск и т.д. 7. Роль операционных систем в управлении параллельными задачами. 8. Будущее параллельного программирования: новые тенденции и технологии. 9. Сравнение параллельного программирования с другими парадигмами (например, распределенное программирование). 10. Влияние параллельного программирования на производительность приложений.
Тема 2. Системы с общей памятью. Низкоуровневое параллельное программирование..	<p>ПК-2</p> <ol style="list-style-type: none"> 1. Определение систем с общей памятью: архитектура и принципы работы. 2. Основные концепции низкоуровневого параллельного программирования: синхронизация, блокировки, мьютексы. 3. Проблемы, возникающие в системах с общей памятью: гонки, взаимные блокировки. 4. Примеры низкоуровневого параллельного программирования на C/C++. 5. Влияние архитектуры процессоров на производительность

	<p>параллельных приложений.</p> <p>6. Использование атомарных операций для управления данными в системах с общей памятью.</p> <p>7. Дебаггинг и тестирование параллельных программ: методы и инструменты.</p> <p>8. Оптимизация производительности приложений в системах с общей памятью.</p> <p>9. Сравнение низкоуровневого и высокоуровневого параллельного программирования.</p> <p>10. Будущее низкоуровневого параллельного программирования: новые технологии и подходы.</p>
Тема 3. Системы с общей памятью. OpenMP.	<p>ПК-2</p> <p>1. Определение OpenMP и его роль в параллельном программировании.</p> <p>2. Основные конструкции OpenMP: директивы, функции и переменные среды.</p> <p>3. Примеры параллелизации циклов с использованием OpenMP.</p> <p>4. Синхронизация потоков и управление ресурсами в OpenMP.</p> <p>5. Преимущества и недостатки использования OpenMP в разработке приложений.</p> <p>6. Оптимизация производительности приложений с OpenMP: примеры и рекомендации.</p> <p>7. Сравнение OpenMP с другими библиотеками параллельного программирования (например, Pthreads).</p> <p>8. Примеры успешного применения OpenMP в научных и инженерных задачах.</p> <p>9. Ограничения и сложности, связанные с использованием OpenMP.</p> <p>10. Будущее OpenMP: развитие стандартов и новые функции.</p>
Тема 4. Message Passing Interface.	<p>ПК-2</p> <p>1. Определение Message Passing Interface (MPI) и его назначение.</p> <p>2. Основные концепции и принципы работы MPI: отправка и получение сообщений.</p> <p>3. Коллективные операции в MPI: broadcast, scatter, gather и другие.</p> <p>4. Примеры реализации параллельных приложений с использованием MPI.</p> <p>5. Производительность MPI: оценка и оптимизация.</p> <p>6. Сравнение MPI с другими подходами к параллельному программированию.</p> <p>7. Примеры применения MPI в высокопроизводительных вычислениях и научных исследованиях.</p> <p>8. Проблемы, возникающие при использовании MPI: отладка и тестирование.</p> <p>9. Будущее MPI: новые подходы и расширения.</p> <p>10. Как MPI влияет на архитектуру параллельных систем.</p>
Тема 5. Модель акторов.	<p>ПК-2</p> <p>1. Определение модели акторов и её значение в параллельном программировании.</p> <p>2. Основные концепции модели акторов: актеры, сообщения, асинхронность.</p> <p>3. Преимущества и недостатки модели акторов по сравнению с другими парадигмами.</p>

	<p>4. Реализация модели акторов в различных языках программирования (например, Erlang, Akka).</p> <p>5. Примеры успешного применения модели акторов в реальных проектах.</p> <p>6. Сравнение модели акторов с традиционными потоками и процессами.</p> <p>7. Проблемы и вызовы, связанные с использованием модели акторов.</p> <p>8. Применение модели акторов в распределённых системах.</p> <p>9. Будущее модели акторов: новые подходы и технологии.</p> <p>10. Сравнение различных реализаций модели акторов и их применение в параллельных системах.</p>
--	---

Шкала оценивания

Шкала оценивания	Критерии оценивания
«Отлично»	Обучающийся глубоко и содержательно раскрывает тему самостоятельной работы, не допустив ошибок. Ответ носит развернутый и исчерпывающий характер.
«Хорошо»	Обучающийся в целом раскрывает тему самостоятельной работы, однако ответ хотя бы на один из них не носит развернутого и исчерпывающего характера.
«Удовлетворительно»	Обучающийся в целом раскрывает тему самостоятельной работы и допускает ряд неточностей, фрагментарно раскрывает содержание теоретических вопросов или их раскрывает содержательно, но допуская значительные неточности.
«Неудовлетворительно»	Обучающийся не владеет выбранной темой самостоятельной работы

6.3. ОЦЕНОЧНЫЕ СРЕДСТВА ПРОМЕЖУТОЧНОГО КОНТРОЛЯ

Вопросы для подготовки к промежуточной аттестации по итогам освоения дисциплины Параллельное программирование:

ПК-2.

1. Понятие параллельное программирование
2. Архитектура ВС. Классификация по Флинну.
3. Архитектура однопроцессорной машины.
4. Архитектура с разделяемой памятью.
5. Архитектура с распределенной памятью.
6. Пути достижения параллелизма.
7. Параллелизм на уровне команд, потоков, процессов.
8. Анализ эффективности параллельных вычислений.
9. Закон Амдала.
10. Проблемы разработки параллельных приложений.
11. Проблема гонки данных.

12. Проблемы синхронизации.
13. Проблемы кешируемой памяти.
14. Модели параллельных приложений.
15. Понятие потока.
16. Структура потока.
17. Состояния потоков.
18. Что такое POSIX.
19. Создание, контроль и завершение выполнения потоков.
20. Отличие между join и detach.
21. Передача параметров в поток
22. Приостановление потока.
23. Приоритеты потоков.
24. Локальное хранилище потока.
25. Пул потоков.
26. Понятие критическая секция.
27. Средства синхронизации.
28. Средства для взаимного исключения.
29. Monitor.
30. Mutex.
31. Сигнальные сообщения.
32. Семафоры.
33. Атомарные операции
34. Конкурентные коллекции
35. Работа с задачами
36. OpenMP. Понятие Fork-Join параллелизма.
37. OpenMP. Директивы и функции.
38. OpenMP. Параллельные и последовательные области.
39. OpenMP. Модель данных.
40. OpenMP. Распределение работы.
41. OpenMP. Синхронизация.
42. OpenMP. Барьер.
43. OpenMP. Критические секции.
44. OpenMP. Гонка данных. Директива atomic.
45. OpenMP. Замки (locks).
46. OpenMP. Согласованный образ памяти. Директива flush.
47. Использование OpenMP.
48. Основные понятия MPI.
49. Структура программы MPI.
50. MPI. Передача сообщений между двумя процессами.
51. MPI. Основные типы операций передачи данных.
52. MPI. Неблокирующий обмен.
53. MPI. Блокирующий обмен.
54. MPI. Выполнение операций приема и передачи одной функцией.
55. MPI. Базовые типы данных.
56. MPI. Пользовательские типы данных.

57. MPI. Коллективные операции. Коммуникаторы.
58. MPI. Виртуальные топологии.
59. Модель акторов. Основная концепция.
60. Модель акторов. Строение актора
61. Модель акторов. Создание экземпляров или систем акторов.
62. Модель акторов. Поведение и состояние актора.
63. Модель акторов. Akka. ActorSystem.
64. Модель акторов. ActorRef, почтовый ящик и актор
65. Модель акторов. Akka. Диспечеры.
66. Модель акторов. Akka. Акторы и сеть
67. Различия между параллелизмом и конкурентностью.
68. Что такое ложное пробуждение (spurious wakeup) в контексте многопоточности?
69. Принципы работы с Condition Variables (условными переменными).
70. Что такое deadlock (взаимная блокировка) и методы её предотвращения?
71. Роль memory barriers (барьеров памяти) в многопоточных программах.
72. Особенности программирования на CUDA для параллельных вычислений на GPU.
73. Как работает модель памяти Java и как это влияет на многопоточность?
74. Применение lock-free и wait-free алгоритмов в параллельном программировании.
75. Использование профилировщиков и отладчиков для анализа параллельных программ.

6.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Основной целью проведения промежуточной аттестации является определение степени достижения целей по учебной дисциплине или ее разделам. Осуществляется это проверкой и оценкой уровня теоретической знаний, полученных обучающимися, умения применять их в решении практических задач, степени овладения обучающимися практическими навыками и умениями в объеме требований рабочей программы по дисциплине, а также их умение самостоятельно работать с учебной литературой.

Организация проведения промежуточной аттестации регламентирована «Положением об организации образовательного процесса в федеральном государственном автономном образовательном учреждении «Московский политехнический университет»

6.4.1. Показатели оценивания компетенций на различных этапах их формирования, достижение обучающимися планируемых результатов обучения по дисциплине

Код и наименование компетенции ПК-2. Способен разрабатывать требования и проектировать программное обеспечение				
Этап (уровень)	Критерии оценивания			
	неудовлетворительно	удовлетворительно	хорошо	отлично
знать	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие следующих знаний: Основные характеристики и предназначение языков программирования низкого уровня	Обучающийся демонстрирует неполное соответствие следующих знаний: Основные характеристики и предназначение языков программирования низкого уровня	Обучающийся демонстрирует частичное соответствие следующих знаний: Основные характеристики и предназначение языков программирования низкого уровня	Обучающийся демонстрирует полное соответствие следующих знаний: Основные характеристики и предназначение языков программирования низкого уровня
уметь	Обучающийся не умеет или в недостаточной степени умеет Использовать понимание архитектуры виртуальной машины языка низкого	Обучающийся демонстрирует неполное соответствие следующих умений: Использовать понимание архитектуры виртуальной машины языка низкого	Обучающийся демонстрирует частичное соответствие следующих умений: Использовать понимание архитектуры виртуальной машины языка низкого	Обучающийся демонстрирует полное соответствие следующих умений Использовать понимание архитектуры виртуальной машины языка низкого
владеть	Обучающийся не владеет или в недостаточной степени владеет: Базовыми навыками разработки ПО на языке низкого уровня	Обучающийся владеет в неполном и проявляет недостаточность владения: Базовыми навыками разработки ПО на языке низкого уровня	Обучающимся допускаются незначительные ошибки, неточности, затруднения, частично владеет Базовыми навыками разработки ПО на языке низкого уровня	Обучающийся свободно применяет полученные навыки, в полном объеме владеет Базовыми навыками разработки ПО на языке низкого уровня

6.4.2. Методика оценивания результатов промежуточной аттестации

Показателями оценивания компетенций на этапе промежуточной аттестации по дисциплине «Параллельное программирование» являются результаты обучения по дисциплине.

Оценочный лист результатов обучения по дисциплине

Код компетенции	Знания	Умения	Навыки	Уровень сформированности
-----------------	--------	--------	--------	--------------------------

и				компетенции на данном этапе / оценка
ПК-2. Способен разрабатывать требования и проектировать программное обеспечение	на уровне знаний: знать основные характеристики и предназначение языков программирования низкого уровня	на уровне умений: уметь использовать понимание архитектуры виртуальной машины языка низкого уровня	на уровне навыков: базовыми навыками разработки ПО на языке низкого уровня	
Оценка по дисциплине (среднее арифметическое)				

Оценка по дисциплине зависит от уровня сформированности компетенций, закрепленных за дисциплиной, и представляет собой среднее арифметическое от выставленных оценок по отдельным результатам обучения (знания, умения, навыки).

Оценка «отлично» выставляется, если среднее арифметическое находится в интервале от 4,5 до 5,0.

Оценка «хорошо» выставляется, если среднее арифметическое находится в интервале от 3,5 до 4,4.

Оценка «удовлетворительно» выставляется, если среднее арифметическое находится в интервале от 2,5 до 3,4.

Оценка «неудовлетворительно» выставляется, если среднее арифметическое находится в интервале от 0 до 2,4.

Промежуточная аттестация обучающихся в форме экзамена проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по дисциплине «Параллельное программирование», при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине проводится преподавателем, ведущим занятия по дисциплине методом экспертной оценки. По итогам промежуточной аттестации по дисциплине выставляется оценка «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

Шкала оценивания	Описание
Отлично	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.

Хорошо	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует неполное, правильное соответствие знаний, умений, навыков приведенным в таблицах показателей, либо если при этом были допущены 2-3 несущественные ошибки.
Удовлетворительно	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, в котором освещена основная, наиболее важная часть материала, но при этом допущена одна значительная ошибка или неточность.
Неудовлетворительно	Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

7. Электронная информационно-образовательная среда

Каждый обучающийся в течение всего периода обучения обеспечивается индивидуальным неограниченным доступом к электронной информационно-образовательной среде Чебоксарского института (филиала) Московского политехнического университета из любой точки, в которой имеется доступ к информационно-телекоммуникационной сети «Интернет» (далее – сеть «Интернет»), как на территории филиала, так и вне ее.

Электронная информационно-образовательная среда – совокупность информационных и телекоммуникационных технологий, соответствующих технологических средств, обеспечивающих освоение обучающимися образовательных программ в полном объеме независимо от места нахождения обучающихся.

Электронная информационно-образовательная среда обеспечивает:

а) доступ к учебным планам, рабочим программам дисциплин (модулей), практик, электронным учебным изданиям и электронным образовательным ресурсам, указанным в рабочих программах дисциплин (модулей), практик;

б) формирование электронного портфолио обучающегося, в том числе сохранение его работ и оценок за эти работы;

в) фиксацию хода образовательного процесса, результатов промежуточной аттестации и результатов освоения программы бакалавриата;

г) проведение учебных занятий, процедур оценки результатов обучения, реализация которых предусмотрена с применением электронного обучения, дистанционных образовательных технологий;

д) взаимодействие между участниками образовательного процесса, в том числе синхронное и (или) асинхронное взаимодействия посредством сети "Интернет".

Функционирование электронной информационно-образовательной среды обеспечивается соответствующими средствами информационно-коммуникационных технологий и квалификацией работников, ее использующих и поддерживающих.

Функционирование электронной информационно-образовательной среды соответствует законодательству Российской Федерации.

Основными составляющими ЭИОС филиала являются:

а) сайт института в сети Интернет, расположенный по адресу www.polytech21.ru, <https://chebpolytech.ru/> который обеспечивает:

- доступ обучающихся к учебным планам, рабочим программам дисциплин, практик, к изданиям электронных библиотечных систем, электронным информационным и образовательным ресурсам, указанных в рабочих программах (разделы сайта «Сведения об образовательной организации»);

- информирование обучающихся обо всех изменениях учебного процесса (новостная лента сайта, лента анонсов);

- взаимодействие между участниками образовательного процесса (подразделы сайта «Задать вопрос директору»);

б) официальные электронные адреса подразделений и сотрудников института с Яндекс-доменом @polytech21.ru (список контактных данных подразделений Филиала размещен на официальном сайте Филиала в разделе «Контакты», списки контактных официальных электронных данных преподавателей размещены в подразделах «Кафедры») обеспечивают взаимодействие между участниками образовательного процесса;

в) личный кабинет обучающегося (портфолио) (вход в личный кабинет размещен на официальном сайте Филиала в разделе «Студенту» подразделе «Электронная информационно-образовательная среда») включает в себя портфолио студента, электронные ведомости, рейтинг студентов и обеспечивает:

- фиксацию хода образовательного процесса, результатов промежуточной аттестации и результатов освоения образовательных программ обучающимися,

- формирование электронного портфолио обучающегося, в том числе с сохранение работ обучающегося, рецензий и оценок на эти работы,

г) электронные библиотеки, включающие электронные каталоги, полнотекстовые документы и обеспечивающие доступ к учебно-методическим материалам, выпускным квалификационным работам и т.д.:

Чебоксарского института (филиала) - «ИРБИС»

д) электронно-библиотечные системы (ЭБС), включающие электронный каталог и полнотекстовые документы:

- ЭБС «ЛАНЬ» -<https://e.lanbook.com/>

- Образовательная платформа Юрайт - <https://urait.ru>

- IPR SMART -<https://www.iprbookshop.ru/>

- е) платформа цифрового образования Политеха -<https://lms.mospolytech.ru/>

- ж) система «Антиплагиат» -<https://www.antiplagiat.ru/>

з) система электронного документооборота DIRECTUM Standard — обеспечивает документооборот между Филиалом и Университетом;

и) система «IC Управление ВУЗом Электронный деканат» (Московский политехнический университет) обеспечивает фиксацию хода образовательного процесса, результатов промежуточной аттестации и результатов освоения образовательных программ обучающимися;

к) система «POLYTECH systems» обеспечивает информационное, документальное автоматизированное сопровождение образовательного процесса;

л) система «Абитуриент» обеспечивает документальное автоматизированное сопровождение работы приемной комиссии.

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература:

1. Малявко, А. А. Параллельное программирование на основе технологий openmp, cuda, opencl, mpi : учебник для вузов / А. А. Малявко. — 3-е изд., испр. и доп. — Москва : Издательство Юрайт, 2025. — 135 с. — (Высшее образование). — ISBN 978-5-534-14116-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/562821>.

2. Раубер, Т. Параллельное программирование для мультитядерных и кластерных систем : руководство / Т. Раубер, Г. Рюнгер ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2024. — 632 с. — ISBN 978-5-93700-261-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/456770>. — Режим доступа: для авториз. пользователей.

3. Кубенский, А. А. Функциональное программирование : учебник и практикум для вузов / А. А. Кубенский. — Москва : Издательство Юрайт, 2025. — 348 с. — (Высшее образование). — ISBN 978-5-9916-9242-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/561074>.

4. Паронджанов, В. Д. Алгоритмические языки и программирование: ДРАКОН : учебник для вузов / В. Д. Паронджанов. — Москва : Издательство Юрайт, 2025. — 436 с. — (Высшее образование). — ISBN 978-5-534-13146-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/567153>.

Дополнительная литература:

1. Зыков, С. В. Программирование : учебник и практикум для вузов / С. В. Зыков. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 285 с. — (Высшее образование). — ISBN 978-5-534-16031-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/560815>.

Периодика:

Научный периодический журнал «Вестник Южно-Уральского государственного университета. Серия «Компьютерные технологии,

управление, радиоэлектроника»: Научный рецензируемый журнал.
<https://vestnik.susu.ru/ctcr> - Текст: электронный.

9. Профессиональные базы данных и информационно-справочные системы

Профессиональная база данных и информационно-справочные системы	Информация о праве собственности (реквизиты договора)
<p>Университетская информационная система РОССИЯ https://uisrussia.msu.ru/</p>	<p>Тематическая электронная библиотека и база для прикладных исследований в области экономики, управления, социологии, лингвистики, философии, филологии, международных отношений, права. свободный доступ</p>
<p>научная электронная библиотека Elibrary http://elibrary.ru/</p>	<p>Научная электронная библиотека eLIBRARY.RU - это крупнейший российский информационно-аналитический портал в области науки, технологии, медицины и образования, содержащий рефераты и полные тексты более 26 млн научных статей и публикаций, в том числе электронные версии более 5600 российских научно-технических журналов, из которых более 4800 журналов в открытом доступе свободный доступ</p>
<p>сайт Института научной информации по общественным наукам РАН. http://www.inion.ru</p>	<p>Библиографические базы данных ИНИОН РАН по социальным и гуманитарным наукам ведутся с начала 1980-х годов. Общий объём массивов составляет более 3 млн. 500 тыс. записей (данные на 1 января 2012 г.). Ежегодный прирост — около 100 тыс. записей. В базы данных включаются аннотированные описания книг и статей из журналов и сборников на 140 языках, поступивших в Фундаментальную библиотеку ИНИОН РАН. Описания статей и книг в базах данных снабжены шифром хранения и ссылками на полные тексты источников из Научной электронной библиотеки.</p>
<p>Федеральный портал «Российское образование» [Электронный ресурс] – http://www.edu.ru</p>	<p>Федеральный портал «Российское образование» – уникальный интернет-ресурс в сфере образования и науки. Ежедневно публикует самые актуальные новости, анонсы событий, информационные материалы для широкого круга читателей. Ежедневно на портале размещаются эксклюзивные материалы, интервью с ведущими специалистами – педагогами, психологами, учеными, репортажи и аналитические статьи. Читатели получают доступ к нормативно-правовой базе сферы образования, они могут пользоваться самыми различными полезными сервисами – такими, как онлайн-тестирование, опросы по актуальным темам и т.д.</p>
<p>Информационные технологии – периодическое научно-техническое издание в области информационных технологий,</p>	<p>Издательство выпускает теоретические и прикладные научно-технические журналы, обеспечивающие научной, производственной, обзорно-аналитической и образовательной информацией руководящих работников и специалистов промышленных предприятий, научных академических и отраслевых организаций, а также учебных</p>

автоматизированных систем и использования информатики в различных приложениях novtex.ru	заведений в области приоритетных направлений развития науки и технологий.
Ассоциация инженерного образования России http://www.ac-raee.ru/	Совершенствование образования и инженерной деятельности во всех их проявлениях, относящихся к учебному, научному и технологическому направлениям, включая процессы преподавания, консультирования, исследования, разработки инженерных решений, оказания широкого спектра образовательных услуг, обеспечения связей с общественностью, производством, наукой и интеграции в международное научно-образовательное пространство. свободный доступ

10. Программное обеспечение (лицензионное и свободно распространяемое), используемое при осуществлении образовательного процесса

Адрес (местоположение) объекта подтверждающего наличие ПО	Программное обеспечение	Информация о праве собственности (реквизиты договора, номер лицензии и т.д.)
№2026 Учебная аудитория для проведения учебных занятий всех видов, предусмотренных программой среднего профессионального образования/бакалавриата/специалитета/ магистратуры, оснащенная оборудованием и техническими средствами обучения, состав которых определяется в рабочих программах дисциплин (модулей) Компьютерный класс Лаборатория микропроцессоров Лаборатория информационных технологий	Kaspersky Endpoint Security для бизнеса – Расширенный Russian Edition. 150-249 Node 2 year Educational Renewal License	Сублицензионный договор № ППИ-126/2023 от 14.12.2023
	Windows 7 OLPNLAcdmс	договор №Д03 от 30.05.2012) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)
	AdobeReader	свободно распространяемое программное обеспечение (бессрочная лицензия)
	Гарант- справочно-правовая система	Договор №С-002-2025 от 09.01.2025
	Yandex браузер	свободно распространяемое программное обеспечение (бессрочная лицензия)
	Microsoft Office Standard 2007(Microsoft DreamSpark Premium Electronic Software Delivery Academic(Microsoft Open License	номер лицензии-42661846 от 30.08.2007) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)
	Zoom	свободно распространяемое программное обеспечение (бессрочная лицензия)

	AIMP	отечественное свободно распространяемое программное обеспечение (бессрочная лицензия)
№ 2196 Учебная аудитория для проведения учебных занятий всех видов, предусмотренных программой бакалавриата/специалитета/ магистратуры, оснащенная оборудованием и техническими средствами обучения, состав которых определяется в рабочих программах дисциплин (модулей)	Windows 7 OLPNLAcdmс	договор №Д03 от 30.05.2012) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)
	Kaspersky Endpoint Security для бизнеса – Расширенный Russian Edition. 150-249 Node 2 year Educational Renewal License	Сублицензионный договор № ППИ-126/2023 от 14.12.2023
	Google Chrome	Свободное распространяемое программное обеспечение (бессрочная лицензия)
	Zoom	свободно распространяемое программное обеспечение (бессрочная лицензия)
	Microsoft Office Standard 2007(Microsoft DreamSpark Premium Electronic Software Delivery Academic(Microsoft Open License	номер лицензии-42661846 от 30.08.2007) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)
№ 1126 Помещение для самостоятельной работы обучающихся	Kaspersky Endpoint Security для бизнеса – Расширенный Russian Edition. 150-249 Node 2 year Educational Renewal License	Сублицензионный договор № ППИ-126/2023 от 14.12.2023
	Windows 7 OLPNLAcdmс	договор №Д03 от 30.05.2012) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)
	AdobeReader	свободно распространяемое программное обеспечение (бессрочная лицензия)
	Гарант- справочно-правовая система	Договор №С-002-2025 от 09.01.2025
	Yandex браузер	свободно распространяемое программное обеспечение (бессрочная лицензия)
	Microsoft Office Standard 2007(Microsoft	номер лицензии-42661846 от 30.08.2007) с допсоглашениями от

	DreamSpark Premium Electronic Software Delivery Academic(Microsoft Open License	29.04.14 и 01.09.16 (бессрочная лицензия)
	Zoom	свободно распространяемое программное обеспечение (бессрочная лицензия)
	AIMP	отечественное свободно распространяемое программное обеспечение (бессрочная лицензия)

11. Материально-техническое обеспечение дисциплины

Тип и номер помещения	Перечень основного оборудования и технических средств обучения
Учебная аудитория для проведения учебных занятий всех видов, предусмотренных программой среднего профессионального образования/бакалавриата/специалитета/ магистратуры/бакалавриата/специалитета/ магистратуры,, оснащенная оборудованием и техническими средствами обучения, состав которых определяется в рабочих программах дисциплин (модулей) Кабинет систем управления ООО «НПО «Каскад-ГРУП» № 219б (г. Чебоксары, ул. К.Маркса, д.60)	<u>Оборудование:</u> комплект мебели для учебного процесса; доска учебная; стенды <u>Технические средства обучения:</u> мультимедийное оборудование (проектор, экран)
Учебная аудитория для проведения учебных занятий всех видов, предусмотренных программой среднего профессионального образования/бакалавриата/специалитета/ магистратуры, оснащенная оборудованием и техническими средствами обучения, состав которых определяется в рабочих программах дисциплин (модулей) Лаборатория информационных технологий	<u>Оборудование:</u> комплект мебели для учебного процесса; доска учебная; автоматизированные рабочие места, автоматизированное рабочее место преподавателя, проектор и экран; маркерная доска; <u>Технические средства обучения:</u> компьютерная техника (процессор Core i3, оперативная память объемом не менее 8 Гб); сервер в лаборатории (8-ядерный процессор с частотой не менее 3 ГГц, оперативная память объемом не менее 16 Гб, жесткие диски общим объемом не менее 1 Тб)

<p>№2066 (г. Чебоксары, ул. К.Маркса, д.60)</p>	
<p>Помещение для самостоятельной работы обучающихся № 1126 (г. Чебоксары, ул. К.Маркса. 60)</p>	<p><u>Оборудование:</u> комплект мебели для учебного процесса; <u>Технические средства обучения:</u> компьютерная техника с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Филиала</p>

Тип и номер помещения	Перечень основного оборудования и технических средств обучения
<p>Учебная аудитория для проведения учебных занятий всех видов, предусмотренных программой среднего профессионального образования/бакалавриата/ специалитета/ магистратуры, оснащенная оборудованием и техническими средствами обучения, состав которых определяется в рабочих программах дисциплин (модулей) Компьютерный класс Лаборатория микропроцессоров Лаборатория информационных технологий №2026 (г. Чебоксары, ул. К.Маркса. 60)</p>	<p><u>Оборудование:</u> комплект мебели для учебного процесса; доска учебная; стенды <u>Технические средства обучения:</u> компьютерная техника; мультимедийное оборудование (проектор, экран)</p>
<p>Учебная аудитория для проведения учебных занятий всех видов, предусмотренных программой среднего профессионального образования/бакалавриата/ специалитета/ магистратуры/бакалавриата/ специалитета/ магистратуры,, оснащенная оборудованием и техническими средствами обучения, состав которых определяется в рабочих программах дисциплин (модулей) Кабинет систем управления ООО «НПО «Каскад-ГРУП» № 2196 (г. Чебоксары, ул. К.Маркса, д.60)</p>	<p><u>орудование:</u> комплект мебели для учебного процесса; доска учебная; стенды <u>Технические средства обучения:</u> мультимедийное оборудование (проектор, экран)</p>
<p>Помещение для самостоятельной работы обучающихся № 1126 (г. Чебоксары, ул. К.Маркса. 60)</p>	<p><u>Оборудование:</u> комплект мебели для учебного процесса; <u>Технические средства обучения:</u> компьютерная техника с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Филиала</p>

12. Методические указания для обучающегося по освоению дисциплины

Методические указания для занятий лекционного типа

В ходе лекционных занятий обучающемуся необходимо вести конспектирование учебного материала, обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации.

Необходимо задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций. Целесообразно дорабатывать свой конспект лекции, делая в нем соответствующие записи из основной и дополнительной литературы, рекомендованной преподавателем и предусмотренной учебной программой дисциплины.

Методические указания для занятий лабораторного типа.

Выполнению лабораторных работ предшествует проверка знаний студентов – их теоретической готовности к выполнению задания. Проверка знаний проводится в форме, которую определяет преподаватель дисциплины (тестирование, опрос).

При проведении лабораторных занятий выделяют следующие разделы:

- общие положения (перечень лабораторных или практических занятий);
- ообщие требования к выполнению работ;
- инструкция по каждой работе;
- справочные материалы и т. д.

Лабораторные занятия позволяют развивать у обучающегося творческое теоретическое мышление, умение самостоятельно изучать литературу, анализировать практику; учат четко формулировать мысль, вести дискуссию, то есть имеют исключительно важное значение в развитии самостоятельного мышления.

Готовясь к докладу или выступлению в рамках интерактивной формы, при необходимости, следует обратиться за помощью к преподавателю.

Методические указания к самостоятельной работе.

Самостоятельная работа обучающегося является основным средством овладения учебным материалом во время, свободное от обязательных учебных занятий. Самостоятельная работа обучающегося над усвоением учебного материала по учебной дисциплине может выполняться в библиотеке университета, учебных кабинетах, компьютерных классах, а также в домашних условиях. Содержание и количество самостоятельной работы обучающегося определяется учебной программой дисциплины, методическими материалами, практическими заданиями и указаниями преподавателя.

Самостоятельная работа в аудиторное время может включать:

- 1) конспектирование (составление тезисов) лекций;
- 2) выполнение контрольных работ;
- 3) решение задач;
- 4) работу со справочной и методической литературой;
- 5) работу с нормативными правовыми актами;
- 6) выступления с докладами, сообщениями на семинарских занятиях;
- 7) защиту выполненных работ;
- 8) участие в оперативном (текущем) опросе по отдельным темам изучаемой дисциплины;
- 9) участие в собеседованиях, деловых (ролевых) играх, дискуссиях, круглых столах, конференциях;
- 10) участие в тестировании и др.

Самостоятельная работа во внеаудиторное время может состоять из:

- 1) повторения лекционного материала;
- 2) подготовки к практическим занятиям;
- 3) изучения учебной и научной литературы;
- 4) изучения нормативных правовых актов (в т.ч. в электронных базах данных);
- 5) решения задач, и иных практических заданий
- 6) подготовки к контрольным работам, тестированию и т.д.;
- 7) подготовки к практическим занятиям устных докладов (сообщений);
- 8) подготовки рефератов, эссе и иных индивидуальных письменных работ по заданию преподавателя;
- 9) выполнения курсовых работ, предусмотренных учебным планом;
- 10) выполнения выпускных квалификационных работ и др.
- 11) выделения наиболее сложных и проблемных вопросов по изучаемой теме, получение разъяснений и рекомендаций по данным вопросам с преподавателями на консультациях.
- 12) проведения самоконтроля путем ответов на вопросы текущего контроля знаний, решения представленных в учебно-методических материалах кафедры задач, тестов, написания рефератов и эссе по отдельным вопросам изучаемой темы.

Текущий контроль осуществляется в форме устных, тестовых опросов, докладов, творческих заданий.

В случае пропусков занятий, наличия индивидуального графика обучения и для закрепления практических навыков студентам могут быть выданы типовые индивидуальные задания, которые должны быть сданы в установленный преподавателем срок.

13. Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья

Обучение по дисциплине «Параллельное программирование» инвалидов и лиц с ограниченными возможностями здоровья (далее ОВЗ)

осуществляется преподавателем с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.

Для студентов с нарушениями опорно-двигательной функции и с ОВЗ по слуху предусматривается сопровождение лекций и практических занятий мультимедийными средствами, раздаточным материалом.

Для студентов с ОВЗ по зрению предусматривается применение технических средств усиления остаточного зрения, а также предусмотрена возможность разработки аудиоматериалов.

По дисциплине «Параллельное программирование» обучение инвалидов и лиц с ограниченными возможностями здоровья может осуществляться как в аудитории, так и с использованием электронной информационно-образовательной среды, образовательного портала и электронной почты.

ЛИСТ ДОПОЛНЕНИЙ И ИЗМЕНЕНИЙ

рабочей программы дисциплины

Рабочая программа дисциплины рассмотрена, обсуждена и одобрена для исполнения в 202__-202__ учебном году на заседании кафедры, протокол № от « » 202 г.

Внесены дополнения и изменения _____

Рабочая программа дисциплины рассмотрена, обсуждена и одобрена для исполнения в 202__-202__ учебном году на заседании кафедры, протокол № от « » 202 г.

Внесены дополнения и изменения _____

Рабочая программа дисциплины рассмотрена, обсуждена и одобрена для исполнения в 202__-202__ учебном году на заседании кафедры, протокол № от « » 202 г.

Внесены дополнения и изменения _____

Рабочая программа дисциплины рассмотрена, обсуждена и одобрена для исполнения в 202__-202__ учебном году на заседании кафедры, протокол № от « » 202 г.

Внесены дополнения и изменения _____

