Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 23.10.2023 16:02:46
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

Декан факультета информационных технологий

/Д. Г. Демидов/

«28» 04 2022 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Параллельное программирование»

Направление подготовки
09.03.03 Прикладная информатика
Образовательная программа (профиль подготовки)
«Большие и открытые данные»

Квалификация (степень) выпускника
бакалавр

Форма обучения
очная

Москва 2022

Программа дисциплины «Параллельное программирование» составлена в соответствии с требованиями ФГОС ВО и учебным планом по направлению **09.03.03 «Прикладная информатика»** и профилю подготовки **«Большие и открытые данные».**

Программу составил



_____/В.Г. Евтихов/

Программа дисциплины утверждена на заседании кафедры «Прикладная информатика»
«____» августа 2022 г. протокол № _____

Заведующий кафедрой
доцент, к.э.н.  /С.В.Суворов/

Программа согласована с руководителем образовательной программы по направлению подготовки **09.03.03 «Прикладная информатика»** по профилю подготовки **«Большие и открытые данные».**



_____ /С.В.Суворов/
«____» августа 2022 г.

Программа утверждена на заседании учебно-методической комиссии факультета Информационных технологий

Председатель комиссии  / Д. Г. Демидов/

«____»_____2022 г. Протокол:

1. Цели освоения дисциплины

Целями дисциплины являются получение представления о современных подходах, используемых при создании высокопроизводительных программно-аппаратных комплексов, освоение современных методов высокопроизводительных вычислений, и, в частности, параллельного программирования.

Уделить основное внимание существующим и перспективным архитектурам высокопроизводительных вычислительных систем и передовым программным технологиям, обеспечивающим высокую производительность создаваемых программ.

Задачами дисциплины являются:

- Овладение технологиями, используемыми при разработке высокопроизводительных современных приложений.
- Получение навыков, необходимых для построения высокопроизводительных параллельных программ с использованием различных технологий.
- Ознакомление с новыми перспективными технологиями разработки высокопроизводительных программ.
- Освоение навыков разработки высокопроизводительных информационных подсистем.
- Подготовиться разработке реальных высокопроизводительных приложений.

2. Место дисциплины в структуре ООП бакалавриата

Входными требованиями, необходимыми для освоения дисциплины, является наличие у обучающихся компетенций, сформированных при изучении дисциплины «Программирование».

На данную дисциплину опираются дисциплины «Проектирование интернет-приложений», «Теоретические основы информатики».

3. Перечень планируемых результатов обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

В результате освоения дисциплины у обучающихся формируются следующие компетенции и должны быть достигнуты следующие результаты обучения как этап формирования соответствующих компетенций:

Код компетенции	В результате освоения образовательной программы обучающийся должен обладать	Перечень планируемых результатов обучения по дисциплине
ПК-7	способностью к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения	**знать:** основные понятия, идеи, методы, связанные с дисциплинами фундаментальной математики, информатики, математического моделирования; краткую историю эволюции вычислительных систем; технологии программирования, основы архитектуры операционных систем; задачи и методы исследования и обеспечения качества и надежности программных компонентов. **уметь:** систематизировать методы фундаментальной математики для построения математических моделей в элементарных прикладных задачах, описывать основные этапы построения алгоритмов; разрабатывать и отлаживать эффективные алгоритмы и программы с использованием современных технологий программирования; формулировать требования к создаваемым программным комплексам. **владеть:** методологией математического моделирования, навыками сбора и работы с математическими источниками информации, теоретическими основами построения алгоритмов; навыками работы с инструментами системного анализа; комбинаторным, теоретико-множественным и вероятностным подходами к постановке и решению задач; навыками программирования в современных средах.

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет **4** зачетных единиц, т.е. 144 академических часа (из них 90 часов– самостоятельная работа студентов). Все они осваиваются на четвертом курсе в 7 семестре.

Лекции – 1 час в неделю (18 часов), лабораторные занятия – 2 часа в неделю (36 часов) форма контроля – Зачет.

Содержание разделов дисциплины

Седьмой семестр

Прикладные задачи, для решения которых требуется использование вычислительных систем высокой и сверхвысокой производительности. Увеличение производительности компьютеров.

Основные инженерные и научные области-потребители суперкомпьютерных ресурсов. Основные классы ресурсоемких (с точки зрения вычислений) задач. Способы увеличения производительности компьютеров.

Архитектура современных и перспективных высокопроизводительных компьютеров

Классификация вычислительных систем . Детализация архитектур по достижимой степени параллелизма. Векторно-конвейерные компьютеры. Вычислительные системы с распределенной памятью (мультикомпьютеры). Параллельные компьютеры с общей памятью (мультипроцессоры). Кластеры. Системы распределенных вычислений (grid-системы) и метакомпьютинг. Возможные будущие системы: квантовые вычислительные устройства.

Разработка алгоритмов для параллельных вычислений

Возможное повышение производительности при параллельных вычислениях. Закон Амдала. Граф алгоритма, параллельная форма алгоритма, информационная структура программ. Графы конкретных алгоритмов. Проектирование параллельных алгоритмов (инженерный подход). Технологические подходы к программированию для параллельных вычислительных систем.

Разработка программ в системах с общей памятью. Технология OpenMp.

POSIX интерфейс для организации нитей (Pthreads). OpenMP – высокоуровневая надстройка над Pthreads. Параллельные и последовательные области. Директивы pragma и функции исполняющей среды.

Разработка программ в системах с распределенной памятью. Интерфейс передачи сообщений MPI.

Основные понятия и определения. Операции передачи данных. Понятие коммуникаторов. Типы данных. Виртуальные топологии. Сообщения MPI. Группы коммуникаторов и процессов. Определение времени выполнения программы. Режимы передачи данных. Синхронизация вычислений.

Программирование на графических процессорах.

Разница между CPU и GPU в параллельных расчетах. API CUDA. Фундаментальные понятия технологии CUDA. CUDA с аппаратной точки зрения. CUDA с программной точки зрения. Управление производительностью.

5. Образовательные технологии

Методика преподавания дисциплины «Параллельное программирование» и реализация компетентностного подхода в изложении и восприятии материала предусматривает использование следующих активных и интерактивных форм проведения групповых, индивидуальных, аудиторных занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся:

– подготовка к лекциям и к выполнению практических работ;
– выполнение лабораторных работ;
– использование интерактивных форм проведения занятий;

Удельный вес занятий, проводимых в интерактивных формах, определен образовательной программой, особенностью контингента обучающихся и содержанием дисциплины «Параллельное программирование» и в целом по дисциплине составляет 25% аудиторных занятий. Занятия лекционного типа составляют примерно 30% от объема аудиторных занятий.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточных аттестаций:

В седьмом семестре

Проверка домашних заданий
Проверка готовности студентов
Проверка выполнения лабораторных работ.

6.1. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

6.1.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.

В результате освоения дисциплины формируются следующие компетенции:

Код компетенции	В результате освоения образовательной программы обучающийся должен обладать
ПК-7	способностью к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения

В процессе освоения образовательной программы данные компетенции, в том числе их отдельные компоненты, формируются поэтапно в ходе освоения обучающимися дисциплин, практик в соответствии с учебным планом и календарным графиком учебного процесса.

6.1.2. Описание показателей и критериев оценивания компетенций, формируемых по итогам освоения дисциплины, описание шкал оценивания

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине.

ПК-7 способностью к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения				
Показатель	**Критерии оценивания**			
	2	**3**	**4**	**5**
знать: основные понятия, идеи, методы, связанные с дисциплинами фундаментальной математики, информатики, математического моделирования; краткую историю эволюции вычислительных систем; технологии программирования, основы архитектуры операционных систем; задачи и методы исследования и обеспечения качества и надежности программных компонентов	Обучающийся демонстрирует полное незнание основные понятия, идеи, методы, связанные с дисциплинами фундаментальной математики, информатики, математического моделирования; краткую историю эволюции вычислительных систем; технологии программирования, основы архитектуры операционных систем; задачи и методы исследования и обеспечения качества и надежности программных компонентов.	Обучающийся способен сформулировать основные понятия, идеи, методы, связанные с дисциплинами фундаментальной математики, информатики, математического моделирования; краткую историю эволюции вычислительных систем; технологии программирования, основы архитектуры операционных систем; задачи и методы исследования и обеспечения качества и надежности программных компонентов	Обучающийся способен сформулировать несколько основных понятий, идеи, методы, связанные с дисциплинами фундаментальной математики, информатики, математического моделирования; краткую историю эволюции вычислительных систем; технологии программирования, основы архитектуры операционных систем; задачи и методы исследования и обеспечения качества и надежности программных компонентов	Обучающийся способен сформулировать несколько задач, основные понятия, идеи, методы, связанные с дисциплинами фундаментальной математики, информатики, математического моделирования; краткую историю эволюции вычислительных систем; технологии программирования, основы архитектуры операционных систем; задачи и методы исследования и обеспечения качества и надежности программных компонентов.

уметь: систематизировать методы фундаментальной математики для построения математических моделей в элементарных прикладных задачах, описывать основные этапы построения алгоритмов; разрабатывать и отлаживать эффективные алгоритмы и программы с использованием современных технологий программирования; формулировать требования к создаваемым программным комплексам.	Обучающийся демонстрирует полную неспособность систематизировать методы фундаментальной математики для построения математических моделей в элементарных прикладных задачах, описывать основные этапы построения алгоритмов; разрабатывать и отлаживать эффективные алгоритмы и программы с использованием современных технологий программирования; формулировать требования к создаваемым программным комплексам	Обучающийся способен систематизировать методы фундаментальной математики для построения математических моделей в элементарных прикладных задачах, описывать основные этапы построения алгоритмов; разрабатывать и отлаживать эффективные алгоритмы и программы с использованием современных технологий программирования; формулировать требования к создаваемым программным комплексам	Обучающийся способен создать и отладить программу с систематизировать методы фундаментальной математики для построения математических моделей в элементарных прикладных задачах, описывать основные этапы построения алгоритмов; разрабатывать и отлаживать эффективные алгоритмы и программы с использованием современных технологий программирования; формулировать требования к создаваемым программным комплексам.	Обучающийся способен создать и отладить программу с использованием технологий OpenMP и MPI; систематизировать методы фундаментальной математики для построения математических моделей в элементарных прикладных задачах, описывать основные этапы построения алгоритмов; разрабатывать и отлаживать эффективные алгоритмы и программы с использованием современных технологий программирования; формулировать требования к создаваемым программным комплексам

владеть: методологией математического моделирования, навыками сбора и работы с математическими источниками информации, теоретическими основами построения алгоритмов; навыками работы с инструментами системного анализа; комбинаторным, теоретико-множественным и вероятностным подходами к постановке и решению задач; навыками программирования в современных средах	Обучающийся демонстрирует полное незнание методологий математического моделирования, навыками сбора и работы с математическими источниками информации, теоретическими основами построения алгоритмов; навыками работы с инструментами системного анализа; комбинаторным, теоретико-множественным и вероятностным подходами к постановке и решению задач; навыками программирования в современных средах .	Обучающийся способен владеть методологией математического моделирования, навыками сбора и работы с математическими источниками информации, теоретическими основами построения алгоритмов; навыками работы с инструментами системного анализа; комбинаторным, теоретико-множественным и вероятностным подходами к постановке и решению задач; навыками программирования в современных средах	Обучающийся владеет в не полном объеме методологией математического моделирования, навыками сбора и работы с математическими источниками информации, теоретическими основами построения алгоритмов; навыками работы с инструментами системного анализа; комбинаторным, теоретико-множественным и вероятностным подходами к постановке и решению задач; навыками программирования в современных средах.	Обучающийся владеет в полном объеме методологией математического моделирования, навыками сбора и работы с математическими источниками информации, теоретическими основами построения алгоритмов; навыками работы с инструментами системного анализа; комбинаторным, теоретико-множественным и вероятностным подходами к постановке и решению задач; навыками программирования в современных средах.

Шкалы оценивания результатов промежуточной аттестации и их описание:

Форма промежуточной аттестации: зачет.

Промежуточная аттестация обучающихся в форме зачёта проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине (модулю) проводится преподавателем, ведущим занятия по дисциплине (модулю) методом экспертной оценки. По итогам промежуточной аттестации по дисциплине (модулю) выставляется оценка «зачтено» или «не зачтено».

К промежуточной аттестации допускаются только студенты, выполнившие все виды учебной работы, предусмотренные рабочей программой по дисциплине «Численные методы» (указывается что именно – прошли промежуточный контроль, выполнили лабораторные работы, выступили с докладом и т.д.)

Шкала оценивания	**Описание**
Зачтено	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.
Не зачтено	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.

Фонды оценочных средств представлены в приложении 2 к рабочей программе.

7. Учебно-методическое и информационное обеспечение дисциплины.

а) основная литература:

1. Белова, И.М. Параллельное программирование**.** [Электронный ресурс] / И.М. Белова, А.А. Рассказов. — Электрон. дан. — М. : МГИУ, 2012. — 101 с. — Режим доступа: http://e.lanbook.com/book/51752

2. Биллиг В. А «Параллельные вычисления и многопоточное программирование». [Электронный ресурс], Издательство: Национальный Открытый Университет «ИНТУИТ», 2016 г. — Режим доступа: http://www.knigafund.ru/books/177948

3. Антонов А. С. «Параллельное программирование с использованием технологии MPI» , [Электронный ресурс] , Издательство: Интернет-Университет Информационных Технологий, 2008 г. — Режим доступа: http://www.knigafund.ru/books/176041

4. Николаев Е. И «Параллельные вычисления». [Электронный ресурс] , Издательство: СКФУ, 2016г. — Режим доступа: http://www.knigafund.ru/books/206142/ г.

б) дополнительная литература:

1. Гергель «Теория и практика параллельных вычислений», [Электронный ресурс] Издательство: Интернет-Университет Информационных Технологий, 2007 г. — Режим доступа: http://www.knigafund.ru/books/178098/

в) программное обеспечение и интернет-ресурсы:

1. ОС Linux
2. Браузер Mozilla Firefox
3. Браузер Google Chrome
4. http://www.parallel.ru
5. http://openmp.org

9. Материально-техническое обеспечение дисциплины.

- Компьютерные классы с ОС Linux в аудиториях АВ1201, АВ1202, АВ1209.
- Лекционная аудитория с проектором АВ1310.

9. Методические рекомендации для самостоятельной работы студентов

Изучение дисциплины осуществляется в строгом соответствии с целевой установкой в тесной взаимосвязи учебным планом. Основой

теоретической подготовки студентов являются лекции. При рассмотрении учебного материалы рекомендуется делать акцент на практические примеры, демонстрировать их реальную работу с помощью проектора. Полезно также обратить внимание студентов на организацию работы кластера при использовании MPI и организацию доступа на компьютеры, участвующие в работе параллельной программы, также следует обратить внимание на организацию работы компьютера при использовании Cuda.

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, дорабатывают конспекты лекций, готовятся к экзамену, а также самостоятельно изучают отдельные темы учебной программы.

Важным обстоятельством является привлечение внимания студентов к обсуждаемой проблеме, стимулирование интереса к ней и организация активного обсуждения, как структуры проблемы, так и составляющих ее наиболее актуальных тем. Для повышения эффективности проведения занятия требуется предварительная подготовка всех его участников. В этой связи рекомендуется заблаговременно (не менее, чем за неделю) оповестить студентов о теме занятия, дать перечень литературы по теме.

При проведении практического занятия преподаватель выполняет, в основном, функции ведущего – направляет студентов в правильное русло решения задач, рассматривает оптимальность предложенных решений, корректирует возможные ошибки.

Активная работа студента на практическом занятии учитывается при определении итоговой оценки его знаний по дисциплине на экзамене.

Проведение лабораторных работ осуществляется в виде разработки небольшими студенческими группами новых программ, использующих параллельное программирование. Студентам необходимо выполнить 3 проекта и написать документацию к ним.

Для сдачи проекта необходимо ответить на 4 из 5 вопросов по проекту. Проект рекомендуется оценивать по 10 балльной шкале. Результат 9-10 баллов формирует компетенции на продвинутом уровне. Результат 4-8 формирует компетенции на базовом уровне. Компетенции считаются неосвоенными при количестве баллов менее 4.

Самостоятельная работа по дисциплине «Веб-технологии» предполагает: выполнение студентами домашних заданий. Домашние задания являются, как правило, продолжением практических занятий и содействуют овладению практическими навыками по основным разделам дисциплины. Самостоятельная работа студентов предполагает изучение теоретического и практического материала по актуальным вопросам

дисциплины. Рекомендуется самостоятельное изучение учебной и научной литературы, использование справочной литературы и др..

При выдаче заданий на самостоятельную работу используется дифференцированный подход к студентам. Перед выполнением студентами самостоятельной внеаудиторной работы преподаватель проводит инструктаж по выполнению задания, который включает: цель здания, его содержание, сроки выполнения, ориентировочный объем работы, основные требования к результатам работы, критерии оценки. В процессе инструктажа преподаватель предупреждает студентов о возможных типичных ошибках, встречающихся при выполнении задания. Инструктаж проводится преподавателем за счет объема времени, отведенного на изучение дисциплины.

Текущий контроль осуществляется на практических занятиях, промежуточный контроль осуществляется на экзамене в письменной (устной) форме.

Самостоятельная работа осуществляется индивидуально.

Контроль самостоятельной работы организуется в двух формах:

- самоконтроль и самооценка студента;
- контроль со стороны преподавателей (текущий и промежуточный).

Текущий контроль осуществляется на практических занятиях, промежуточный контроль осуществляется на зачете, экзамене в письменной (устной) форме.

Критериями оценки результатов самостоятельной работы студента являются:

- уровень освоения студентом учебного материала;
- умения студента использовать теоретические знания при выполнении практических задач;
- сформированность умений;
- оформление материала в соответствии с требованиями.

10. Методические рекомендации для преподавателя

Приложение 1.

Структура и содержание дисциплины «Параллельное программирование» по направлению подготовки 09.03.03 «Прикладная информатика»
(бакалавр)

n/n	Раздел	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость в часах	Виды самостоятельной работы студентов	Формы аттестации									
				Л	П/С	Лаб	СРС	КСР	К.Р.	К.П.	РГР	Реферат	К/р	Э	З
	Седьмой семестр														
1	Прикладные задачи, для решения которых требуется использование вычислительных систем высокой и сверхвысокой производительности. Увеличение производительности компьютеров. Основные инженерные и научные области-потребители суперкомпьютерных ресурсов. Основные классы ресурсоемких (с точки зрения вычислений) задач. Способы увеличения производительности	7	1	3		6	15	+							

	компьютеров. Знакомство с программированием с использованием «нитей»														
2	**Архитектура современных и перспективных высокопроизводительных компьютеров** Классификация вычислительных систем . Детализация архитектур по достижимой степени параллелизма. Векторно-конвейерные компьютеры. Вычислительные системы с распределенной памятью (мультикомпьютеры). Параллельные компьютеры с общей памятью (мультипроцессоры). Кластеры. Системы распределенных вычислений (grid-системы) и метакомпьютинг. Возможные будущие системы: квантовые вычислительные устройства. Разработка программ с использованием «нитей»	7	2-4	3		6	15	+							
3	**Разработка алгоритмов для параллельных вычислений** Возможное повышение производительности	7	5-6	3		6	15	+							

	при параллельных вычислениях. Закон Амдала. Граф алгоритма, параллельная форма алгоритма, информационная структура программ. Графы конкретных алгоритмов. Проектирование параллельных алгоритмов (инженерный подход). Технологические подходы к программированию для параллельных вычислительных систем. Знакомство с Openmp на лабораторных работах														
4	**Разработка программ в системах с общей памятью. Технология OpenMp.** POSIX интерфейс для организации нитей (Pthreads). OpenMP – высокоуровневая надстройка над Pthreads. Параллельные и	7	7-9	3		6	15						+		

	последовательные области. Директивы pragma и функции исполняющей среды. Проект 1														
5	Разработка программ в системах с распределенной памятью. Интерфейс передачи сообщений MPI. Основные понятия и определения. Операции передачи данных. Понятие коммуникаторов. Типы данных. Виртуальные топологии. Сообщения MPI. Группы коммуникаторов и процессов. Определение времени выполнения программы. Режимы передачи данных. Синхронизация вычислений. Проект 2	**7**	10-13	3		6	15	+							
6	**Программирование на графических процессорах.** Разница между CPU и GPU в параллельных расчетах. API CUDA. Фундаментальные понятия технологии CUDA. CUDA с аппаратной точки зрения. CUDA с программной	7	13-17	3		6	15	+							

	точки зрения. Управление производительностью. Проект 3														
	Форма аттестации		**18-21**											**Э**	
	Всего часов по дисциплине в седьмом семестре			18		36	90	+	+				+		

Приложение 2

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Направление подготовки:

09.03.03 «Прикладная информатика»

Форма обучения: очная

Вид профессиональной деятельности: (В соответствии с ФГОС ВО)

Кафедра: Прикладная информатика

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

«Параллельное программирование»

Состав:

1. Паспорт фонда оценочных средств
2. Перечень оценочных средств
3. Оценочные средства

Москва 2022 г.

Паспорт фонда оценочных средств по дисциплине «Параллельное программирование» по направлению подготовки 09.03.03 «Прикладная информатика» (бакалавр)

«Параллельное программирование»					
ФГОС ВО 09.03.03 «Прикладная информатика» (уровень бакалавриата)					
В процессе освоения данной дисциплины студент формирует и демонстрирует следующие **профессиональные компетенции**:					
Компетенции		**Перечень компонентов**	**Технология формирования компетенций**	**Форма оценочного средства**	**Степени уровней освоения компетенций**
Индекс	**Формулировка**				
ПК-7	способностью к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения	**знать:** основные понятия, идеи, методы, связанные с дисциплинами фундаментальной математики, информатики, математического моделирования; краткую историю эволюции вычислительных систем; технологии программирования, основы архитектуры операционных систем; задачи и методы исследования и обеспечения качества и надежности программных компонентов. **уметь:** систематизировать методы фундаментальной математики для построения	лекции, лабораторные работы, практические занятия	экзамен (Экз)	**пороговый уровень:** Обучающийся способен сформулировать 1 задачу, в которой применяются высокопроизводительные вычислительные системы и способен сформулировать основные принципы и основы синтаксиса OpenMP. Обучающийся способен создать и отладить простейшую программу с использованием технологии OpenMP **базовый уровень:** Обучающийся способен сформулировать несколько задач, в которых применяются высокопроизводительные

		математических моделей в элементарных прикладных задачах, описывать основные этапы построения алгоритмов; разрабатывать и отлаживать эффективные алгоритмы и программы с использованием современных технологий программирования; формулировать требования к создаваемым программным комплексам. **владеть:** методологией математического моделирования, навыками сбора и работы с математическими источниками информации, теоретическими основами построения алгоритмов; навыками работы с инструментами системного анализа; комбинаторным, теоретико-множественным и вероятностным подходами к постановке и решению задач; навыками программирования в современных средах			вычислительные системы, способен сформулировать основные принципы и основы синтаксиса OpenMP и MPI. Обучающийся способен создать и отладить программу с использованием технологий OpenMP и MPI. **повышенный уровень:** Обучающийся способен сформулировать несколько задач, в которых применяются высокопроизводительные вычислительные системы, способен сформулировать основные принципы и основы синтаксиса OpenMP и MPI а также основы программирования на графических процессорах. Обучающийся способен создать и отладить программу с использованием технологий OpenMP и MPI; разработать и отладить программу с использованием программирования на графических процессорах с использованием технологии CUDA

Примерный перечень вопросов к экзамену по курсу «Параллельное программирование»:

1. Что такое параллельное программирование и суперкомпьютеры. Области в которых может возникать потребность в параллельных вычислениях. Особенности параллельных вычислений.
2. Увеличение производительности при параллельных вычислениях. К каким областям задач может быть применено распараллеливание к каким нет. Закон Амдала.
3. Классификация параллельных компьютеров и систем. Классификация Флинна (SISD, SIMD, MISD, MIMD) и другие классификации.
4. MIMD параллельные системы.
5. SIMD системы.
6. Кластерные компьютеры и их особенности.
7. Векторно-конвейерные компьютеры. Компьютер Cray.
8. Массивно-параллельные компьютеры.
9. Параллельные компьютеры с общей памятью.
10. Производительность параллельных компьютеров.
11. Особенности построения параллельных алгоритмов. Системы автоматического распараллеливания и обычные схемы.
12. Граф алгоритма и параллельные вычисления.
13. Внутренний параллелизм алгоритмов.
14. Разделение на подзадачи. Установление связей между отдельными подзадачами.
15. Балансировка загрузки узлов. Зависимость выбора алгоритма от структуры вычислительной сети.
16. Технология программирования OpenMP.
17. Распараллеливание циклов в OpenMP.
18. Современные средства "неавтоматического" параллельного программирования. MPI. Блокируемая и неблокируемая передача сообщений в MPI.
19. MPI. Коллективные взаимодействия процессов. Понятия групп и коммуникаторов, управление ими.
20. Виртуальные топологии в MPI.
21. Системы автоматического распараллеливания. Особенности программирования задач в таких системах.
22. T-система
23. Система распараллеливания CUDA.
24. Параллельные вычисления на графических процессорах