Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Артеменков Михаил Николаевич
Должность: Ректор
Дата подписания: 09.11.2026 18:40:57
Уникальный программный ключ:
a63b74166e8f4fc17bd1e0462725f79e393bfdf9

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Смоленский государственный университет»

Кафедра прикладной математики и информатики

«Утверждаю»
Проректор по учебно-методической
деятельности
_____ Ю.А. Устименко
«25» декабря 2025 г.

Рабочая программа дисциплины
Б1.В.03 Параллельное программирование прикладных задач

Направление подготовки **01.04.02 Прикладная математика и информатика**
Направленность (профиль) **Математическое и программное обеспечение цифровых радиотехнических систем и комплексов**
Форма обучения – очная
Курс – 2
Семестр – 3
Всего зачетных единиц – 3, часов – 108
Форма отчетности: экзамен – 3 семестр

Программу разработал: доктор технических наук, доцент В.И. Мунерман

Одобрена на заседании кафедры
«20» ноября 2025 г., протокол № 5

Заведующий кафедрой _____ С.В. Козлов

Смоленск
2025

1. Место дисциплины в структуре ОП

Дисциплина «Параллельное программирование прикладных задач» относится к части, формируемой участниками образовательных отношений, основной образовательной программы по направлению подготовки 01.04.02 Прикладная математика и информатика.

Потребность решения сложных прикладных задач с большим объемом вычислений и принципиальная ограниченность максимального быстродействия "классических" - по схеме фон Неймана - ЭВМ привели к появлению многопроцессорных вычислительных систем (МВС). Использование таких средств вычислительной техники позволяет существенно увеличивать производительность ЭВМ при любом существующем уровне развития компьютерного оборудования. При этом, однако, необходимо "параллельное" обобщение традиционной – последовательной – технологии решения задач на ЭВМ. Так, численные методы в случае МВС должны проектироваться как системы параллельных и взаимодействующих между собой процессов, допускающих исполнение на независимых процессорах. Применяемые алгоритмические языки и системное программное обеспечение должны обеспечивать создание параллельных программ, организовывать синхронизацию и взаимоисключение асинхронных процессов и т.п. Предметом рассмотрения настоящего курса и является изучение перечисленного круга вопросов. Цель курса состоит в изложении математических моделей и методов параллельного программирования для многопроцессорных вычислительных систем.

Для изучения дисциплины «Параллельное программирование прикладных задач» необходимы знания, умения и навыки, полученные при изучении таких дисциплин, как «Взаимодействие в распределенных программных системах», «Web программирование» / «Технологии разработки Web-сервисов».

В результате изучения данной дисциплины у обучающихся формируются знания, умения, навыки и компетенции, необходимые для прохождение технологической (проектно-технологической) и преддипломной практик, выполнения и защиты выпускной квалификационной работы.

2. Планируемые результаты обучения по дисциплине

Компетенция	Индикаторы достижения
ПК-1. Способен к проведению научных исследований, включая постановку задачи исследования, моделирование объектов и процессов в целях оптимизации их параметров, проведение экспериментальных исследований и подготовку отчетов и публикаций	**Знает**: принципы подготовки и проведения научных исследований, математические и информационные модели, методы моделирования сигналов, процессов и явлений, лежащих в основе принципов действия радиотехнических систем и комплексов, методы разработки эффективных алгоритмов решения научно-исследовательских задач, способы организации и проведения экспериментальных исследований. **Умеет**: планировать порядок проведения научных исследований, формулировать и решать задачи, использовать математический аппарат и численные методы для анализа, синтеза и моделирования радиотехнических систем и комплексов, применять алгоритмы решения исследовательских задач с использованием современных языков программирования, самостоятельно проводить экспериментальные исследования. **Владеет**: навыками выбора теоретических и экспериментальных методов исследования, математическим аппаратом для решения задач теоретической и прикладной радиотехники, методами исследования и моделирования объектов радиотехники,

	навыками разработки стратегии и методологии исследования радиотехнических систем и комплексов, навыками проведения исследования с применением современных средств и методов.
ПК-2. Способен проектировать программное и информационное обеспечение, вычислительные модели и модели данных для реализации элементов новых (или известных) программных продуктов в области цифровых радиотехнических систем и комплексов	**Знает:** структуру программного обеспечения, основные требования к его проектированию, методики разработки и анализа блок-схем алгоритмов, современные языки программирования, технологии программирования. **Умеет:** проектировать программное и информационное обеспечение в области цифровых радиотехнических систем и комплексов, формулировать требования к программным продуктам, создавать блок-схемы алгоритмов функционирования разрабатываемых программных продуктов, оценивать их вычислительную сложность. **Владеет:** методами и приемами формализации и алгоритмизации поставленных проектных задач по созданию программного обеспечения.
ПК-3. Способен разрабатывать программное обеспечение цифровых радиотехнических систем и комплексов	**Знает:** архитектуру аппаратной платформы, синтаксис, принципы и особенности программирования (в том числе кроссплатформенного, распределенного и параллельного программирования), стандартные библиотеки выбранного языка программирования, стандарты реализации интерфейсов устройств, технологии разработки и отладки программных продуктов, принципы информационного построения сетевого взаимодействия, методики тестирования программного обеспечения. **Умеет:** использовать методы и приемы формализации и алгоритмизации поставленных задач, применять выбранный язык программирования для написания программного кода, осуществлять отладку программного обеспечения. **Владеет:** основными методами и приемами разработки и отладки программных продуктов.

3. Содержание дисциплины

1. **Принципы разработки параллельных алгоритмов и программ.** Оценка эффективности параллельных вычислений. Показатель эффекта распараллеливания (ускорение). Эффективность использования вычислительной системы. Способы оценки показателей. Основные характеристики вычислительной системы, влияющие на величины ускорения и эффективности (архитектура, количество процессоров, топология каналов передачи данных).

2. **Оценка коммуникационной трудоемкости параллельных алгоритмов**. Характеристики топологий сети передачи данных. Алгоритмы маршрутизации. Методы передачи данных. Анализ трудоемкости основных операций передачи данных. Передача данных между двумя процессорами сети. Одиночная и множественная рассылка сообщений. Операция циклического сдвига. Методы логического представления топологии коммуникационной среды. Отображение кольцевой топологии и топологии решетки на гиперкуб.

3. **Уровни распараллеливания вычислений**. Распараллеливание вычислений на уровне команд, выражений, программных модулей, отдельно выполняемых заданий.

4. **Этапы построения параллельных алгоритмов и программ**. Выявление функциональной независимости отдельных фрагментов алгоритма (параллелизм команд). Геометрическое разделение вычислений (параллелизм данных). Иерархическая декомпозиция обработки данных. Выбор параллельного алгоритма. Реализация алгоритма в виде параллельной программы. Построение исполняемой программы для параллельной вычислительной системы.

5. **Технологические аспекты распараллеливания**. Декомпозиция алгоритма на параллельно исполняемые фрагменты вычислений. Распределение заданий по процессорам и балансировка. Синхронизация и взаимоисключение. Организация взаимодействия.

6. **Параллельные численные алгоритмы**. Организация параллельного исполнения рекурсивных вычислений. Проблема рекурсивной зависимости этапов обработки данных. Способы разбиения матриц (горизонтальная, вертикальная, блочные схемы). Методы вычисления произведения матриц с использованием разных схем разбиения матриц.

4. Тематический план

№ п/п	Разделы и темы	Всего часов	Формы занятий		
			лекции	лабораторные занятия	самостоятельная работа
1	Принципы разработки параллельных алгоритмов и программ.	12	2	2	8
2	Оценка коммуникационной трудоемкости параллельных алгоритмов.	13	2	2	9
3	Уровни распараллеливания вычислений	13	2	2	9
4	Этапы построения параллельных алгоритмов и программ	13	2	2	9
5	Технологические аспекты распараллеливания	13	2	2	9
6	Параллельные численные алгоритмы	17	4	4	9
	Экзамен	27	–	–	27
	ИТОГО	108	14	14	53+27

5. Виды учебной деятельности

Занятия лекционного типа

1. **Принципы разработки параллельных алгоритмов и программ**. Оценка эффективности параллельных вычислений. Показатель эффекта распараллеливания (ускорение). Эффективность использования вычислительной системы. Способы оценки показателей. Основные характеристики вычислительной системы, влияющие на величины ускорения и эффективности (архитектура, количество процессоров, топология каналов передачи данных).

2. **Оценка коммуникационной трудоемкости параллельных алгоритмов.** Характеристики топологий сети передачи данных. Алгоритмы маршрутизации. Методы передачи данных. Анализ трудоемкости основных операций передачи данных. Передача данных между двумя процессорами сети. Одиночная и множественная рассылка сообщений. Операция циклического сдвига. Методы логического представления

топологии коммуникационной среды. Отображение кольцевой топологии и топологии решетки на гиперкуб.

3. **Уровни распараллеливания вычислений.** Распараллеливание вычислений на уровне команд, выражений, программных модулей, отдельно выполняемых заданий.

4. **Этапы построения параллельных алгоритмов и программ.** Выявление функциональной независимости отдельных фрагментов алгоритма (параллелизм команд). Геометрическое разделение вычислений (параллелизм данных). Иерархическая декомпозиция обработки данных. Выбор параллельного алгоритма. Реализация алгоритма в виде параллельной программы. Построение исполняемой программы для параллельной вычислительной системы.

5. **Технологические аспекты распараллеливания.** Декомпозиция алгоритма на параллельно исполняемые фрагменты вычислений. Распределение заданий по процессорам и балансировка. Синхронизация и взаимоисключение. Организация взаимодействия.

6. **Параллельные численные алгоритмы.** Организация параллельного исполнения рекурсивных вычислений. Проблема рекурсивной зависимости этапов обработки данных.

7. **Способы разбиения матриц.** Способы разбиения матриц (горизонтальная, вертикальная, блочные схемы). Методы вычисления произведения матриц с использованием разных схем разбиения матриц.

Занятия семинарского типа - лабораторные работы

Лабораторная работа №1. *Уровни распараллеливания вычислений*

Вопросы:

1. Распараллеливание вычислений на уровне команд.
2. Распараллеливание вычислений на уровне выражений.
3. Распараллеливание вычислений на уровне программных модулей.
4. Распараллеливание вычислений на уровне отдельно выполняемых заданий.

Задание:

1. Для заданной задачи умножения матриц $A_{n\times n}\times B_{n\times n}$ ($n = 1000$) выделите и опишите: уровень инструкций (ILP) – какие операции можно выполнять параллельно на уровне CPU; уровень данных (DLP) – как разбить матрицы на блоки для параллельной обработки; уровень задач (TLP) – какие подзадачи (например, вычисление строк результата) можно запускать независимо. Обоснуйте выбор для каждого уровня.

2. Рассмотреть задачу сортировки массива из 1 млн элементов. Для каждого уровня распараллеливания (ILP, DLP, TLP): предложите конкретный метод реализации; оцените потенциальное ускорение (по закону Амдала при 8 ядрах); укажите основные ограничения (зависимости данных, накладные расходы).

3. Выберите популярное ПО с поддержкой параллелизма (например, видеоредактор, браузер, СУБД). Для него: определите, на каких уровнях распараллеливания оно работает; приведите 1-2 примера конкретных параллельных фрагментов кода/процессов; оцените, насколько эффективно используется аппаратный параллелизм (многоядерность, SIMD).

Лабораторная работа №2. *Этапы построения параллельных алгоритмов и программ*

Вопросы:

1. Выявление функциональной независимости отдельных фрагментов алгоритма (параллелизм команд).
2. Геометрическое разделение вычислений (параллелизм данных).
3. Иерархическая декомпозиция обработки данных.
4. Выбор параллельного алгоритма.

5. Реализация алгоритма в виде параллельной программы.
6. Построение исполняемой программы для параллельной вычислительной системы.

Задание:

1. Для задачи численного интегрирования $\int e^{-x^2} dx$ от 0 до 1 методом прямоугольников (10 000 интервалов): выполните декомпозицию на 4 независимые подзадачи; опишите, как будут распределяться данные между потоками; укажите точки синхронизации.

2. Разработайте схему обмена данными для параллельного алгоритма решения СЛАУ методом Гаусса (матрица 100 × 100, 4 потока). Ответьте: какие данные передаются между потоками на каждом шаге; как избежать гонок данных; какой тип коммуникаций (точка-точка, коллективные операции) оптимален.

3. Реализуйте параллельный подсчёт числа π методом Монте Карло (1 млн точек) с использованием 2, 4 и 8 потоков. Для каждого случая: измерьте время выполнения; постройте график ускорения Sp=T1/Tp; объясните отклонения от линейного ускорения.

Лабораторная работа №3. *Технологические аспекты распараллеливания. Синхронизация и взаимоисключение. Организация взаимодействия.*

Вопросы:

1. Декомпозиция алгоритма на параллельно исполняемые фрагменты вычислений.
2. Распределение заданий по процессорам и балансировка.

Задание:

1. Дан код на C++ с общим счётчиком counter++ в 4 потоках. продемонстрируйте, как возникает гонка данных (приведите пример некорректного результата); исправьте код, используя мьютекс; оцените накладные расходы на синхронизацию (замерьте время для 10 000 операций).

2. Реализуйте алгоритм параллельного сокращения (суммирование массива из 1024 элементов) с использованием барьеров. Разбейте массив на 8 блоков; каждый поток суммирует свой блок; используйте барьер для ожидания завершения всех потоков перед финальным сложением. Проверьте корректность результата.

3. Создайте потокобезопасную очередь на основе связанного списка. Реализуйте: методы push() и pop() с использованием мьютексов; тест с 3 потоками-производителями и 2 потоками-потребителями; проверку отсутствия мёртвых блокировок.

Лабораторная работа №4. *Технологические аспекты распараллеливания*

Вопросы:

1. Синхронизация и взаимоисключение переменных и процессов.
2. Организация взаимодействия процессов.

Задание:

1. Для задачи обработки потока изображений (100 кадров/сек) сравните: многопоточность (threads); асинхронное программирование (async/await); GPU-вычисления (CUDA/OpenCL). Укажите: преимущества и недостатки каждой модели; ожидаемое ускорение; сложность реализации.

2. Дано: параллельный цикл по массиву из 10 000 элементов с лёгкой операцией (умножение на 2). Измерьте время выполнения при гранулярности 1, 10, 100 и 1000 элементов на поток; постройте график зависимости времени от гранулярности; определите оптимальную гранулярность для 4 потоков.

3. Используйте инструмент профилирования (например, perf для Linux или Visual Studio Profiler) для анализа параллельной программы (например, сортировки слиянием). Выявите «горячие точки» (узкие места); предложите 2–3 способа оптимизации; оцените потенциальный прирост производительности.

Лабораторная работа №5-6. *Параллельные численные алгоритмы*

Вопросы:

1. Организация параллельного исполнения рекурсивных вычислений.
2. Проблема рекурсивной зависимости этапов обработки данных.

Задание:

1. Реализуйте умножение квадратных матриц 1000×1000 с разбиением на блоки 100×100. Используйте 4 потока для независимых блоков; измерьте время выполнения; сравните с последовательным алгоритмом и вычислите ускорение.

2. Решите систему Ax=b (матрица 500×500, разреженная) методом Якоби с 4 потоками. Распределите строки матрицы между потоками; организуйте обмен граничными значениями после каждой итерации; остановитесь при достижении точности 10^{-6}. Оцените число итераций и время.

3. Вычислите производную функции f(x)=sin(x) на отрезке [0,2π] с шагом h = 0,01 (200 точек). Разделите отрезок на 4 части; каждый поток вычисляет производную на своём участке по формуле (f(x+h)−f(x))/h; соберите результаты и постройте график. Сравните с аналитическим решением f′(x)=cos(x).

Лабораторная работа №7. *Параллельные численные алгоритмы*

Вопросы:

1. Способы разбиения матриц (горизонтальная, вертикальная, блочные схемы).
2. Методы вычисления произведения матриц с использованием разных схем разбиения матриц.

Задание:

1. Решите двумерное уравнение теплопроводности ∂t/∂u=α(∂x2/∂2u+∂y2/∂2u) на квадратной области [0, 1]×[0, 1] с нулевыми граничными условиями и начальным условием u(x,y,0)=sin(πx)sin(πy).

2. Реализуйте параллельный вариант алгоритма Кули-Тьюки для вычисления БПФ комплексного вектора длины $N=2^{14}$.

3. Смоделируйте движение N=10000 частиц под действием гравитационных сил (в 2D).

Самостоятельная работа

Самостоятельная работа направлена на углубление и закрепление знаний студентов и развитии практических умений. Она заключается в работе с лекционными материалами, поиске и обзоре литературы и электронных источников, информации по заданным темам курса, в изучении тем, вынесенных на самостоятельную проработку и самостоятельном выполнении заданий.

Темы для самостоятельного изучения

1. Основы параллелизма и модели исполнения.
2. Синхронизация и примитивы взаимоисключения.
3. OpenMP: параллелизация циклов и секций.
4. MPI: обмен сообщениями в распределённых системах.
5. CUDA: программирование для GPU.
6. Асинхронное программирование и event loops.
7. Параллельные структуры данных.
8. Профилирование и отладка параллельных программ.
9. Паттерны параллельного программирования.
10. Современные фреймворки и экосистемы.

Задания для самостоятельной работы

Задание 1. Реализовать таймаут для асинхронной операции.

Задание 2. Реализовать lock-free очередь на CAS.
Задание 3. Разработать OpenMP-программу и найти «горячие точки».
Задание 4. Реализовать quicksort с параллельным разделением.
Задание 5. Запустить Ray-задачу на кластере из 2 узлов.

6. Критерии оценивания результатов освоения дисциплины (модуля)

6.1. Оценочные средства и критерии оценивания для текущей аттестации

1. Задания для лабораторных занятий

Образец задания для лабораторной работы

Разработать параллельную программу для операции разделения в алгоритме внешней сортировки.

Критерии оценивания задания для лабораторной работы:

№ п/п	Критерий	Балл
1.	Задание выполнено в срок (на соответствующем занятии)	1
2.	Задание выполнено студентом самостоятельно (без помощи и подсказок преподавателя)	1
3.	Результат строго соответствует заданию, служит прямым ответом на соответствующий вопрос задания	1
4.	Представленный результат проходит проверку (тестирование) на разных наборах входных данных	1
5.	Студент дает правильные пояснения о том, как выполнялось задание и каковы его результаты	1
6.	Работа оформлена в соответствии с требованиями (структура, форматирование, подписи, комментарии в коде и т. п.)	1
7.	Алгоритм/решение выстроено последовательно, блоки кода или расчётов логически разделены и обоснованы	1
8.	Использованы оптимальные методы и алгоритмы; решение не содержит избыточных или неоправданно сложных шагов	1
9.	В работе (особенно в коде или расчётах) присутствуют понятные пояснения, облегчающие восприятие	1
10.	Студент уверенно отвечает на дополнительные вопросы по заданию, демонстрирует понимание смежных аспектов задачи	1
	Итоговая (суммарная) оценка	Max - 10

2. Теоретические вопросы

Теоретические вопросы предлагаются студенту во время сдачи лабораторного задания и направлены на проверку знания студентом основных теоретических положений дисциплины.

Примеры теоретических вопросов:

1. Что такое алгоритм внешней сортировки?
2. Какие способы реализации операции разделения в алгоритме внешней сортировки вам известны?
3. Как реализовать операции разделения в алгоритме внешней сортировки для параллельной программы?

Критерии оценки ответов на теоретические вопросы:

Показатель	Количество баллов
1) Полнота раскрытия темы (ответ охватывает все ключевые аспекты вопроса, не упускает существенные детали)	1
2) Точность терминологии (корректно использованы понятия и термины дисциплины; отсутствуют смысловые искажения)	1
3) Логичность и структура (ответ выстроен последовательно: есть введение, основная часть, выводы; мысли связаны между собой)	1
4) Глубина понимания (показано осмысление материала: не просто пересказ, а объяснение причинно-следственных связей, принципов, механизмов)	1
5) Аргументированность (положения подкреплены примерами, ссылками на источники, формулы, теоремы или экспериментальные данные)	1
6) Самостоятельность изложения (ответ не сводится к дословному воспроизведению учебника/лекции; присутствует индивидуальная интерпретация)	1
7) Актуальность и современность (упомянуты современные исследования, технологии или дискуссии по теме (если применимо))	1
8) Корректность примеров (приведённые иллюстрации точно соответствуют теории, не содержат ошибок и не вводят в заблуждение)	1
9) Ясность и доступность (текст понятен без дополнительных уточнений; сложные моменты разъяснены; нет избыточной перегруженности)	1
10) Грамотность и оформление (отсутствуют орфографические, пунктуационные и стилистические ошибки; формулы, схемы, цитаты оформлены по правилам)	1
Итоговая (суммарная) оценка	Max - 10

3. Выполнение заданий для самостоятельной работы (перечень заданий приведен в разделе 5 «Самостоятельная работа»).

Критерии оценивания выполнения заданий:

№ п/п	Критерий	Балл
1.	Задание выполнено в срок (на соответствующем занятии)	1
2.	Задание выполнено студентом самостоятельно (без помощи и подсказок преподавателя)	1
3.	Результат строго соответствует заданию, служит прямым ответом на соответствующий вопрос задания	1
4.	Представленный результат проходит проверку (тестирование) на разных наборах входных данных	1
5.	Студент дает правильные пояснения о том, как выполнялось задание и каковы его результаты	1
6.	Работа оформлена в соответствии с требованиями (структура, форматирование, подписи, комментарии в коде и т. п.)	1
7.	Алгоритм/решение выстроено последовательно, блоки кода или расчётов логически разделены и обоснованы	1
8.	Использованы оптимальные методы и алгоритмы; решение не содержит избыточных или неоправданно сложных шагов	1
9.	В работе (особенно в коде или расчётах) присутствуют	1

	понятные пояснения, облегчающие восприятие	
10.	Студент уверенно отвечает на дополнительные вопросы по заданию, демонстрирует понимание смежных аспектов задачи	1
	Итоговая (суммарная) оценка	Max - 10

4. Проверочная работа

Образец проверочной работы

1. Выполнить сравнение времени последовательного и параллельного (алгоритм Кэннона) умножения матриц (9 потоков).
2. Сравнить расчетное и фактическое ускорения.

Критерии оценивания проверочной работы:

№ п/п	Критерий	Балл
1.	Задание выполнено в срок (на соответствующем занятии)	1
2.	Задание выполнено студентом самостоятельно (без помощи и подсказок преподавателя)	1
3.	Результат строго соответствует заданию, служит прямым ответом на соответствующий вопрос задания	1
4.	Представленный результат проходит проверку (тестирование) на разных наборах входных данных	1
5.	Студент дает правильные пояснения о том, как выполнялось задание и каковы его результаты	1
6.	Работа оформлена в соответствии с требованиями (структура, форматирование, подписи, комментарии в коде и т. п.)	1
7.	Алгоритм/решение выстроено последовательно, блоки кода или расчётов логически разделены и обоснованы	1
8.	Использованы оптимальные методы и алгоритмы; решение не содержит избыточных или неоправданно сложных шагов	1
9.	В работе (особенно в коде или расчётах) присутствуют понятные пояснения, облегчающие восприятие	1
10.	Студент уверенно отвечает на дополнительные вопросы по заданию, демонстрирует понимание смежных аспектов задачи	1
	Итоговая (суммарная) оценка	Max - 10

6.2 Оценочные средства и критерии оценивания для промежуточной аттестации

Форма промежуточной аттестации – **экзамен**.

Для определения уровня сформированности компетенций применяется процедура независимой оценки. Независимая оценка уровня сформированности компетенций осуществляется комиссией (с возможностью привлечения внешних экспертов со стороны работодателей). В состав комиссии входят 3 преподавателя: заведующий кафедрой – председатель комиссии и независимый эксперт, преподаватель, читающий дисциплину «Параллельное программирование прикладных задач», преподаватель кафедры – независимый эксперт. Экзамен проходит в традиционной форме – по билетам. В билет включены 2 теоретических вопроса и 1 практическое задание. Ответ студента оценивается

коллегиально, путем голосования. Каждый член комиссии имеет один голос. При расхождении оценок у преподавателей, решающим является голос заведующего кафедрой.

Вопросы для подготовки к экзамену (Экзаменационные вопросы)

1. Оценка эффективности параллельных вычислений. Показатель эффекта распараллеливания (ускорение). Эффективность использования вычислительной системы.
2. Основные характеристики вычислительной системы, влияющие на величины ускорения и эффективности (архитектура, количество процессоров, топология каналов передачи данных).
3. Алгоритмы маршрутизации.
4. Методы передачи данных.
5. Анализ трудоемкости основных операций передачи данных.
6. Передача данных между двумя процессорами сети.
7. Одиночная и множественная рассылка сообщений.
8. Операция циклического сдвига.
9. Методы логического представления топологии коммуникационной среды.
10. Отображение кольцевой топологии и топологии решетки на гиперкуб.
11. Распараллеливание вычислений на уровне команд, выражений, программных модулей, отдельно выполняемых заданий.
12. Выбор параллельного алгоритма.
13. Реализация алгоритма в виде параллельной программы.
14. Построение исполняемой программы для параллельной вычислительной системы.
15. Параллельное исполнение машинной программы.
16. Выбор оптимального алгоритма для конкретной вычислительной системы,
17. Нахождение наилучшей топологии вычислительной системы для решения определенной задачи, распараллеливание существующего алгоритма.
18. Декомпозиция алгоритма на параллельно исполняемые фрагменты вычислений.
19. Распределение заданий по процессорам и балансировка.
20. Синхронизация и взаимоисключение. Организация взаимодействия.
21. Автоматическая векторизация и распараллеливание.
22. Проблемно-ориентированные компиляторы.
23. Общая характеристика стандарта OpenMP.
24. Создание параллельных областей.
25. Разделение вычислительной нагрузки между потоками.
26. Работа с данными. Синхронизация.
27. Выявление функциональной независимости отдельных фрагментов алгоритма (параллелизм команд).
28. Геометрическое разделение вычислений (параллелизм данных). Иерархическая декомпозиция обработки данных.
29. Проблема рекурсивной зависимости этапов обработки данных. Каскадная схема. Метод Оутса.
30. Способы разбиения матриц (горизонтальная, вертикальная, блочные схемы).
31. Методы вычисления произведения матриц с использованием разных схем разбиения матриц. Обеспечение предельно допустимого параллелизма.
32. Ленточный алгоритм.
33. Алгоритм Фокса.
34. Алгоритм Кэннона.
35. Симметричной горизонтальное распределение данных.
36. Параллельная реализация операции Join.

Практические задания (примеры)

1. Вычислить сумму элементов массива через #pragma omp parallel for. Замерить время, рассчитать ускорение.

2. Реализовать потокобезопасный счётчик с мьютексами. Проверить корректность при 4-8 потоках.

3. Распределить блоки матриц между процессами, использовать MPI_Send/MPI_Recv. Оценить масштабируемость.

4. Выполнить 10+ запросов через asyncio, сравнить с последовательным вариантом. Обработать ошибки.

5. Обработать список файлов (подсчёт слов) через пул потоков. Сравнить производительность для 2, 4, 8 потоков.

6. Реализовать редукцию на GPU для поиска max в массиве. Сравнить скорость с CPU-версией.

7. Моделировать обмен данными между потоками через семафоры. Протестировать с буфером размером 5, 10, 20 элементов.

Образец экзаменационного билета

1. Работа с данными. Синхронизация.
2. Проблема рекурсивной зависимости этапов обработки данных. Каскадная схема. Метод Оутса.
3. Разработать программу, параллельно реализующую операцию Join.

Критерии и показатели оценивания ответа студента по уровням

Оценка 10/ "Отлично" выставляется студенту, который демонстрирует при выполнении практического задания систематизированные, глубокие и полные знания по всем разделам учебной дисциплины, а также по основным вопросам, выходящим за ее пределы; точное использование научной терминологии, стилистически грамотное, логически правильное изложение ответа на вопросы; безупречное владение инструментарием учебной дисциплины, умение его эффективно использовать в постановке и решении научных и профессиональных задач; обладает способностью самостоятельно и творчески решать сложные проблемы в нестандартной ситуации; демонстрирует полное и глубокое усвоение содержания основной и дополнительной литературы, рекомендованной рабочей программой дисциплины; демонстрирует умение ориентироваться в теориях, концепциях и направлениях по изучаемой дисциплине и давать им критическую оценку, использовать научные достижения других дисциплин.

Оценка 9/"Отлично" выставляется студенту, который демонстрирует при выполнении заданий систематизированные, глубокие и полные знания по всем разделам учебной дисциплины; точное использование научной терминологии, грамотное, логически правильное изложение ответа на вопросы; владение инструментарием учебной дисциплины, умение его эффективно использовать в постановке и решении научных и профессиональных задач; обладает способностью самостоятельно и творчески решать сложные задачи в нестандартной ситуации в рамках учебной дисциплины; демонстрирует полное усвоение содержания основной и дополнительной литературы, рекомендованной рабочей программой дисциплины; демонстрирует умение ориентироваться в теориях, концепциях и направлениях по изучаемой дисциплине и давать им аналитическую оценку.

Оценка 8/"Отлично" выставляется студенту, который демонстрирует при выполнении заданий систематизированные, глубокие и полные знания по всем поставленным вопросам в объеме учебной дисциплины; использование научной терминологии, грамотное, логически правильное изложение ответа на вопросы; умение делать обоснованные выводы и обобщения; владение инструментарием учебной

дисциплины, умение его использовать в постановке и решении научных и профессиональных задач; обладает способностью самостоятельно решать сложные задачи в рамках учебной дисциплины; демонстрирует полное усвоение содержания основной и дополнительной литературы, рекомендованной рабочей программой дисциплины; демонстрирует умение ориентироваться в теориях, концепциях и направлениях по изучаемой дисциплине и давать им аналитическую оценку.

Оценка 7/"Хорошо" выставляется студенту, который демонстрирует при выполнении заданий систематизированные, глубокие знания по всем поставленным вопросам в объеме учебной дисциплины; использование научной терминологии, грамотное, логически правильное изложение ответа на вопросы; умение делать обоснованные выводы и обобщения; владение инструментарием учебной дисциплины, умение его использовать в постановке и решении научных и профессиональных задач; обладает способностью свободно решать типовые задачи в рамках учебной дисциплины; демонстрирует усвоение содержания основной и дополнительной литературы, рекомендованной рабочей программой дисциплины; демонстрирует умение ориентироваться в основных теориях, концепциях и направлениях по изучаемой дисциплине и давать им аналитическую оценку.

Оценка 6/"Хорошо" выставляется студенту, который демонстрирует при выполнении заданий достаточно полные и систематизированные знания в объеме учебной дисциплины; использование необходимой научной терминологии, грамотное, логически правильное изложение ответа на вопросы; умение делать обобщенные и обоснованные выводы; владение инструментарием учебной дисциплины, умение его использовать в решении научных и профессиональных задач; обладает способностью решать типовые задачи в рамках учебной дисциплины; демонстрирует усвоение содержания основной и дополнительной литературы, рекомендованной рабочей программой дисциплины; демонстрирует умение ориентироваться в базовых теориях, концепциях и направлениях по изучаемой дисциплине и давать им сравнительную оценку.

Оценка 5/"Удовлетворительно" выставляется студенту, который демонстрирует при выполнении заданий достаточные знания в объеме учебной дисциплины; минимальное использование необходимой научной терминологии, достаточно грамотное изложение ответа на вопросы, но с нарушением логической последовательности; умение делать выводы с допущением несущественных ошибок; недостаточное владение инструментарием учебной дисциплины, умение его использовать только в решении стандартных (типовых) задач; умение решать стандартные (типовые) задачи в рамках учебной дисциплины с минимальной долей самостоятельности под руководством преподавателя; демонстрирует усвоение содержания основной литературы, рекомендованной рабочей программой дисциплины; умение ориентироваться в основных теориях, концепциях и направлениях по изучаемой дисциплине и давать им оценку; умение под руководством преподавателя решать стандартные (типовые) задачи, допустимый уровень культуры исполнения заданий.

Оценка 4/"Удовлетворительно" выставляется студенту, который демонстрирует при выполнении заданий достаточный объем знаний в рамках учебной дисциплины; научная терминология используется редко, при ответе на вопросы демонстрируется нарушение логической последовательности; владение инструментарием учебной дисциплины только при решении стандартных (типовых) задач; низкий уровень культуры исполнения заданий.

Оценка 3/"Неудовлетворительно" выставляется студенту, который демонстрирует при выполнении заданий недостаточный объем знаний в рамках учебной дисциплины; научная терминология не используется; в ответах на вопросы

демонстрируется существенные логические ошибки; слабое владение инструментарием учебной дисциплины, некомпетентность при решении стандартных (типовых) задач.

Оценка 2/"Неудовлетворительно" выставляется студенту, который демонстрирует при выполнении заданий фрагментарные знания в рамках учебной дисциплины; научная терминология не используется или используется неграмотно; в ответах на вопросы демонстрируется грубые логические ошибки; невладение инструментарием учебной дисциплины, некомпетентность при решении стандартных (типовых) задач.

Оценка 1/"Неудовлетворительно" выставляется студенту, который демонстрирует отсутствие приращения знаний и компетентности в рамках учебной дисциплины.

Таблица перевода оценок из 10-ти бальной системы в 5-ти бальную систему

Оценка по 10-ти бальной системе	Оценка по 5-ти бальной системе
10, 9, 8	5 «отлично»
7, 6	4 «хорошо»
5, 4	3 «удовлетворительно»
3, 2, 1	2 «неудовлетворительно»

7. Перечень основной и дополнительной учебной литературы

7.1 Основная литература

1. Гордеев С. И. Организация баз данных: учебник для вузов / С. И. Гордеев, В. Н. Волошина. — 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2025. — 691 с. — (Высшее образование). — ISBN 978-5-534-21115-3. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/559377

2. Парфенов Ю. П. Постреляционные хранилища данных: учебное пособие для вузов / Ю. П. Парфенов; под научной редакцией Н. В. Папуловской. — Москва: Издательство Юрайт, 2025. — 97 с. — (Высшее образование). — ISBN 978-5-534-21173-3. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/559502.

3. 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.

4. 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.

5. Бабичев С.Л. Распределенные системы: учебник для вузов / С. Л. Бабичев, К. А. Коньков. — Москва: Издательство Юрайт, 2025. — 507 с. — (Высшее образование). — ISBN 978-5-534-11380-8. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/566315.

7.2 Дополнительная литература

1. Бройдо, В. Л. Вычислительные системы, сети и телекоммуникации: учеб. пособие: / В. Л. Бройдо, О. П. Ильина. - 3-е изд. - СПб. : Питер, 2008.

2. Максимов, Н. В. Компьютерные сети: учеб. пособие : / Н. В. Максимов, И. И. Попов. - 3-е изд., перераб. и доп. - М. : Форум, 2008.

3. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы : учеб. пособие / В. Г. Олифер, Н. А. Олифер. - 4-е изд. - СПб. : Питер, 2011.

4. Гергель В.П., Стронгин Р.Г. Основы параллельных вычислений для многопроцессорных вычислительных систем. - Н.Новгород, ННГУ, 2001.
5. Богачев К.Ю. Основы параллельного программирования. - М.: БИНОМ. Лаборатория знаний, 2003.
6. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. - СПб.: БХВ-Петербург, 2002.
7. Немнюгин С., Стесик О. Параллельное программирование для многопроцессорных вычислительных систем - СПб.: БХВ-Петербург, 2002.
8. Гергель В.П. (2007). Теория и практика параллельных вычислений. – М.: Интернет-Университет, БИНОМ. Лаборатория знаний.
9. Голицына, О. Л. Программное обеспечение : учеб. пособие / О. Л. Голицына, Т. Л. Партыка, И. И. Попов. - 3-е изд., перераб. и доп. - М. : Форум, 2010.
10. В.Г. Олифер, Н.А. Олифер. Компьютерные сети. Принципы, технологии, протоколы. С-Пб.: Питер, 2001. –672 с.
11. Робачевский. Операционная система Unix. С-Пб.: BHV-Санкт-Петербург, 1997.
12. ОС Unix: Руководство системного администратора. Киев: BHV, 1997.
13. "ATM: технология высокоскоростных сетей", А.Н. Назаров, М.В. Симонов. "Радио и связь", 1997.
14. М.В. Ратынский "Основы сотовой связи". Москва, <Радио и связь>, 1998.
15. Н.Н. Слепов "Синхронные цифровые сети SDH". Москва, <Эко-трэндз>, 1997.
16. Шиндер Д. Основы компьютерных сетей. Диал-ка, 2002,304с.
17. Олифер Н. Сетевые операционные системы. Учебник. Питер, 2001, 544 с.

7.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Автоматически реализуемые свойства (Руководство по программированию на C#). – http://msdn.microsoft.com/ru-ru/library/bb384054.aspx
2. Oliver Sturm Functional Programming in C#: Classic Programming Techniques for Modern Projects. – Wiley. 2011. – 290 c.
3. Практическое руководство. Создание и использование библиотек DLL на языке C# (Руководство по программированию на C#). – http://msdn.microsoft.com/ru-ru/library/3707x96z(v=vs.90).aspx
4. Guido Gybels "Использование ассемблера в Дельфи". Электронная книга. – http://delphiexpert.ru/view_download.php?id=7
5. Доступ к COM серверам Microsoft Office. – http://www.delphisources.ru/pages/faq/base/access_to_microsoft_com_servers.html
6. Интеграция C# с Microsoft Office. – http://easyprog.ru/index.php?option=com_content&task=view&id=818&Itemid=54
7. Работа с базами данных в Delphi с помощью ADO. – http://www.delphisources.ru/pages/articles/2006_year/art_27_11_06/art_27_11_06.html
8. Zeeshan Amjad Using Unmanaged code and assembler in C#. – http://www.codeproject.com/Articles/1392/Using-Unmanaged-code-and-assembler-in-C

8. Материально-техническое обеспечение

Учебная аудитория для проведения занятий лекционного типа, оснащенная стандартной учебной мебелью, интерактивной доской, мультимедиапроектором, ноутбуком и колонками.

Учебная аудитория для проведения занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации - компьютерная аудитория с выходом в Интернет.

Помещение для самостоятельной работы – компьютерный класс с доступом к сети «Интернет» и ЭИОС СмолГУ.

9. Программное обеспечение

Kaspersky Endpoint Security для бизнеса Стандартный АО «Лаборатория Касперского», лицензия 1FB6-161215-133553-1-6231.

Microsoft Open License, лицензия 49463448 в составе: Microsoft Windows Professional 7 Russian; Microsoft Office 2010 Russian.

Microsoft Visual Studio.

Microsoft SQL Server Express