

**Федеральное государственное автономное образовательное
учреждение высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»**

УТВЕРЖДЕНО
Проректор по учебной работе

А.А. Воронов

	Рабочая программа дисциплины (модуля)
по дисциплине:	Введение в распараллеливание алгоритмов и программ
по направлению:	Информатика и вычислительная техника
профиль подготовки:	Программная инженерия
	передовая инженерная школа радиолокации, радионавигации и программной инженерии
	кафедра информатики и вычислительной математики
курс:	2
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 3 (осенний) - Дифференцированный зачет

Аудиторных часов: 30 всего, в том числе:

лекции: 30 час.

семинары: 0 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 15 час.

Всего часов: 45, всего зач. ед.: 1

Программу составил: В.Е. Карпов, канд. физ.-мат. наук, доцент

Программа обсуждена на заседании кафедры информатики и вычислительной математики 07.02.2024

Аннотация

В настоящем курсе студенты познакомятся с написанием параллельных программ, направленных на решение вычислительных задач. Будут рассмотрены основные подходы к организации параллельных вычислений: динамическое, потоковое, статическое планирование. Обучающиеся освоят классические алгоритмы, такие как решение жестких систем ОДУ большой размерности, решение краевых линейных и нелинейных задач, конечно-разностные методы.

1. Цели и задачи

Цель дисциплины

Дать студентам необходимый запас фундаментальных понятий, моделей, формальных методов, навыков, знаний, посильное понимание существа дела для того, чтобы они могли, используя сформированный в курсе знания и навыки, получить базовые знания для освоения специальности, для изучения специализированных дисциплин или конкретной области применения информатики, а также для развития информационной культуры. Также целью является освоение студентами знаний в области применения современных высокопроизводительных комплексов различной архитектуры в научных исследованиях и прикладных областях, в частности — в математическом моделировании и обработке больших массивов данных.

Задачи дисциплины

- Формирование основных знаний в области применения высокопроизводительных вычислительных комплексов различной архитектуры на основе курсов информатики, операционных систем, языков программирования и курсов вычислительной математики для обеспечения технологических основ математического моделирования в современных инновационных сферах деятельности;
- обучение студентов принципам создания эффективных параллельных алгоритмов и программ, анализа существующих программ и алгоритмов на параллельность; знакомство с основными методами и принципами параллельного программирования, основными технологиями параллельного программирования;
- формирование подходов к выполнению исследований студентами в области параллельных вычислений и математического моделирования с использованием современных технологий, и программных средств параллельного программирования в рамках выпускных работ на степень бакалавра.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-4 Способен осуществлять сбор и обработку научно-технической и (или) технологической информации для решения фундаментальных и прикладных задач	ОПК-4.1 Владеет методами научного поиска и интеллектуального анализа информации при решении задач профессиональной деятельности

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны знать:

- Историю эволюции вычислительных систем и историческую необходимость использования параллельных вычислений;
- основы архитектуры параллельных вычислительных комплексов;
- основные технологические этапы разработки параллельных программ;
- принципы асимптотического анализа алгоритмов;
- методы декомпозиции последовательных алгоритмов;
- способы эквивалентных и неэквивалентных преобразований последовательных программ, позволяющих использовать их на параллельных вычислительных комплексах;
- основные идеи при реализации численных алгоритмов, позволяющих избежать случая низкой эффективности распараллеливания.

уметь:

- Оценивать асимптотическую сложность используемых алгоритмов и выбирать оптимальные алгоритмы для современных программ;
- анализировать последовательные программы для выявления возможности их распараллеливания;
- оценивать эффективность работы распараллеленных программ;
- выбирать эффективные численные методы для поставленных задач математического моделирования.

владеть:

- Приемами распараллеливания алгоритмов и программ;
- средствами и технологиями разработки приложений, обеспечивающих проведение параллельного вычислительного эксперимента.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Проблемы эволюции вычислительных систем. Архитектурный и программный параллелизм. Парадигмы последовательного и параллельного программирования	6			3
2	Элементы асимптотического анализа алгоритмов	4			3
3	Декомпозиция алгоритмов на уровне операций	4			3
4	Укрупнение параллельных ярусов	6			2
5	Параллельность циклов	6			2
6	Основные подходы к организации размещения задач на процессорах	4			2
Итого часов		30			15
Подготовка к экзамену		0 час.			
Общая трудоёмкость		45 час., 1 зач.ед.			

4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 3 (Осенний)

1. Проблемы эволюции вычислительных систем. Архитектурный и программный параллелизм. Парадигмы последовательного и параллельного программирования

Архитектурный и программный параллелизм. Проблемы использования параллельных систем. Непереносимость алгоритмов. Ошибки округления. Зависимость от архитектуры, языка, компилятора, ОС. Расширенная квалификация Флинна. Примеры SISD, SIMD, MISD, MIMD машин. Модели параллельного программирования. Этапы параллельного решения проблем: decomposition, assignment, orchestration, mapping. Задачи, решаемые на каждом этапе.

2. Элементы асимптотического анализа алгоритмов

Элементы асимптотического анализа алгоритмов. Основные предположения. Вычислительная модель RAM. Терминология и обозначения. Асимптотические отношения. Наилучший последовательный алгоритм. Пример асимптотического анализа сложности последовательного алгоритма выбора элемента из множества. Рекуррентные соотношения. Основная теорема асимптотического анализа. Вычислительные модели PRAM. Ускорение при распараллеливании. Стоимость параллельного алгоритма. Оптимальность алгоритма по стоимости. Пример асимптотического анализа сложности параллельного алгоритма выбора элемента из множества. Ограниченность асимптотического анализа.

3. Декомпозиция алгоритмов на уровне операций

Декомпозиция алгоритмов на уровне операций. Понятие о графе алгоритма. Строго параллельные формы графа, каноническая параллельная форма. Соотнесение строго параллельных форм с выполнением алгоритма на конкретных архитектурных решениях. Ярусы параллельной формы, их ширина и высота. Концепция неограниченного параллелизма. Определение максимально возможного ускорения по ярусно-параллельной форме алгоритма.

4. Укрупнение параллельных ярусов

Укрупнение параллельных ярусов. Декомпозиция алгоритмов и программ на уровне действий и операторов. Условия Бернштейна и их нарушение. Истинная или потоковая зависимость, антизависимость, зависимость по выходным данным. Графы зависимостей. Связь зависимостей операторов с возможностью одновременного выполнения.

5. Параллельность циклов

Параллельность циклов. Простые циклы: расстояние зависимости; зависимости, связанные и несвязанные с циклом. Вложенные циклы. Вектора зависимости и направлений. Их использование для определения возможности распараллеливания циклов. Способы устранения зависимостей: loop distribution, code replication, loop alignment, приватизация переменных, индукция и редукция. Декомпозиция на уровне блоков операторов, -блоки.

6. Основные подходы к организации размещения задач на процессорах

Assignment. Основные подходы к организации размещения задач на процессорах. Динамическое, потоковое, статическое планирование, work pool, pipeline, competition, divide & conquer. Их недостатки и достоинства. Проблемы балансировки загрузки процессоров. Гомогенные и гетерогенные вычислительные системы.

5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Необходимое оборудование для лекций: аудитория, компьютер (ноутбук) и мультимедийное оборудование (проектор, звуковая система).

6. Перечень рекомендуемой литературы

Основная литература

1. Численные методы, алгоритмы и программы. Введение в распараллеливание [Текст] : учеб. пособие для вузов / В. Е. Карпов, А. И. Лобанов. — М. : Физматкнига, 2014. — 192 с.

Дополнительная литература

1. Вычислительная математика и структура алгоритмов [Текст] : 10 лекций о том, почему трудно решать задачи на вычислительных системах параллельной архитектуры и что надо знать дополнительно, чтобы успешно преодолевать эти трудности : учебник для вузов / В. В. Воеводин ; Моск. гос. ун-т им. М. В. Ломоносова. — 2-е изд., стереотип. — М. : Изд-во Моск. ун-та, 2010. — 168 с.

7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

Не используются

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)

На лекциях используются мультимедийные технологии, включая демонстрацию презентаций. В процессе самостоятельной работы обучающихся возможно использование любые среды программирования.

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Студент, изучающий дисциплину «Введение в распараллеливание алгоритмов и программ», должен с одной стороны, овладеть общим понятийным аппаратом, а с другой стороны, должен научиться применять теоретические знания на практике.

В результате изучения дисциплины студент должен знать основные определения, понятия, аксиомы.

Успешное освоение курса требует напряжённой самостоятельной работы студента. В программе курса приведено минимально необходимое время для работы студента над темой. Самостоятельная работа включает в себя:

- чтение и конспектирование рекомендованной литературы;
- проработку учебного материала (по конспектам лекций, учебной и научной литературе), подготовку ответов на вопросы, предназначенных для самостоятельного изучения, доказательство отдельных утверждений, свойств;
- подготовку к дифференцированному зачету.

Руководство и контроль за самостоятельной работой студента осуществляется в форме индивидуальных консультаций.

Важно добиться понимания изучаемого материала, а не механического его запоминания. При затруднении изучения отдельных тем, вопросов, следует обращаться за консультациями к лектору.

ПРИЛОЖЕНИЕ

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Информатика и вычислительная техника
профиль подготовки:	Программная инженерия передовая инженерная школа радиолокации, радионавигации и программной инженерии кафедра информатики и вычислительной математики
курс:	2
квалификация:	бакалавр
Семестр, формы промежуточной аттестации: 3 (осенний) - Дифференцированный зачет	
Разработчик:	В.Е. Карпов, канд. физ.-мат. наук, доцент

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-4 Способен осуществлять сбор и обработку научно-технической и (или) технологической информации для решения фундаментальных и прикладных задач	ОПК-4.1 Владеет методами научного поиска и интеллектуального анализа информации при решении задач профессиональной деятельности

2. Показатели оценивания компетенций

В результате изучения дисциплины «Введение в распараллеливание алгоритмов и программ» обучающийся должен:

знать:

- Историю эволюции вычислительных систем и историческую необходимость использования параллельных вычислений;
- основы архитектуры параллельных вычислительных комплексов;
- основные технологические этапы разработки параллельных программ;
- принципы асимптотического анализа алгоритмов;
- методы декомпозиции последовательных алгоритмов;
- способы эквивалентных и неэквивалентных преобразований последовательных программ, позволяющих использовать их на параллельных вычислительных комплексах;
- основные идеи при реализации численных алгоритмов, позволяющих избежать случая низкой эффективности распараллеливания.

уметь:

- Оценивать асимптотическую сложность используемых алгоритмов и выбирать оптимальные алгоритмы для современных программ;
- анализировать последовательные программы для выявления возможности их распараллеливания;
- оценивать эффективность работы распараллеленных программ;
- выбирать эффективные численные методы для поставленных задач математического моделирования.

владеть:

- Приемами распараллеливания алгоритмов и программ;
- средствами и технологиями разработки приложений, обеспечивающих проведение параллельного вычислительного эксперимента.

3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

Типовые контрольные задания для текущего контроля

- Какие из технологических этапов присущи только парадигме параллельного программирования: построение математической модели, декомпозиция, аранжировка, написание программы?
- Чем декомпозиция по данным отличается от декомпозиции по вычислениям?
- В чем заключаются достоинства и недостатки статического и динамического способов назначения задач виртуальным исполнителям?
- Пусть имеется два последовательных алгоритма решения одной и той же задачи — 1 и 2. Алгоритм 1 является оптимальным по поведению, алгоритм 2 не является оптимальным. Что можно сказать о реальных временах решения при заданном параметре масштаба N : время работы алгоритма 1 всегда будет меньше времени работы алгоритма 2, времена будут одинаковыми, ничего сказать нельзя?
- Используя основную теорему асимптотического анализа, оцените асимптотическое поведение алгоритма сортировки слиянием. Является ли этот алгоритм оптимальным?
- Какие схемы разрешения конфликта при разрешенной одновременной записи различными исполнителями в одну ячейку памяти в модели вычислительной системы PRAM вы знаете?

- При вычислении теоретического значения ускорения для параллельных алгоритмов решения некоторой задачи на 4-х исполнителях были получены следующие значения: 2, 3.9, 5. Какие из значений являются корректными? Почему?
- При вычислении реального значения ускорения для параллельных алгоритмов решения некоторой задачи на 4-х исполнителях были получены следующие значения: 2, 3.9, 5. Какие из значений являются корректными? Почему?
- Что означает высказывание, «некоторый параллельный алгоритм является оптимальным по стоимости»?
- Перечислите основные свойства графа алгоритма, реализованного программой.
- Сколько строгих параллельных форм может существовать у графа алгоритма: 1, ограниченное количество, но больше 1; бесконечно много?
- Докажите единственность канонической строгой параллельной формы графа алгоритма.
- Для графа алгоритма, содержащего 500 вершин, высота некоторой строгой параллельной формы равна 25. Что можно сказать о максимальном ускорении, которое может быть получено при реализации алгоритма на параллельной вычислительной системе в модели PRAM — оно будет всегда меньше 20, строго равно 20, может быть больше 20?
- Для программного текста, реализованного на Фортране 77

```
b(1) = a(1) * 2
```

```
do i = 2, 5
```

```
    b(i) = b(i-1) + a(i) * 2
```

```
enddo
```

постройте граф алгоритма, каноническую строгую параллельную форму и оцените максимально возможное ускорение при реализации алгоритма на параллельной вычислительной системе.

4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

1. Максимальная пиковая производительность наиболее мощных современных параллельных вычислительных систем измеряется: в единицах EFLOPs, в десятках PFLOPs, в единицах PFLOPs, в сотнях TFLOPs?
2. Производительность компьютера, достигнутая при выполнении некоторой программы, выражена в TFLOPs. Это значение говорит о: среднем количестве операций над вещественными данными, представленными в форме с фиксированной запятой, выполненных за секунду в процессе обработки данной программы; общем числе команд, выполненных за время работы программы; средней скорости выполнения данным компьютером арифметических операций над вещественными числами, представленными в форме с плавающей запятой; средней скорости выполнения данным компьютером арифметических операций над вещественными числами, представленными в форме с плавающей запятой, достигнутой при выполнении данной программы; высокой реальной производительности данного компьютера.
3. Умножение двух квадратных плотных вещественных матриц компьютер выполнил за 5 сек с производительностью 50 GFLOPs. Какого размера были матрицы: 500*500, 1000*1000, 2000*2000, 5000*5000, 7000*7000, верного ответа нет?
4. Сколько кризисов software насчитывается за всю историю развития электронных вычислительных систем: 4, 3, 2, 1, ни одного?
5. В каком компьютере функциональные устройства сочетали одновременно принципы конвейерной и параллельной обработки: IBM 704, IBM STRETCH, CDC 6600, CDC 7600, ILLIAC IV, ATLAS, верного ответа нет?
6. Отметьте правильные утверждения на тему машинного представления чисел в современных ЭВМ: все существовавшие до сих пор ЭВМ используют в качестве базовой двоичную систему счисления; машинное эpsilon в основном определяется длиной мантиссы в представлении вещественных чисел; мантисса числа в двоичном представлении - та же, что и мантисса его десятичного логарифма; машинные ноль и эpsilon не могут быть получены с помощью фортран-программы, их следует найти в документации к компьютеру; машинное сложение коммутативно; машинное сложение ассоциативно; машинное умножение коммутативно; машинное умножение ассоциативно?
7. Кто из перечисленных ниже людей внёс наибольший вклад в развитие параллельной вычислительной техники: Джон Грей; Сеймур Крэй; Стивен Крейн; Кристиан Рэй; Френсис Дрейк?

8. Действительные числа в машинном представлении: всегда хранятся точно; всегда хранятся с ненулевой ошибкой округления; хранились на всех существовавших вычислительных системах в двоичном представлении; имеют относительную ошибку округления не более машинного нуля; имеют абсолютную ошибку округления не более машинного эпсилон; иногда хранятся точно?
9. Конвейерное ФУ деления состоит из пяти ступеней, срабатывающих за 2, 5, 3, 1 и 1 такт соответственно. Чему равно наименьшее число тактов, за которое можно обработать 40 пар аргументов на данном устройстве: 1, 5, 12, 40, 207, 212, 480, верного ответа нет?
10. В конвейерном устройстве есть 7 ступеней, срабатывающих за одну единицу времени каждая. За сколько единиц времени это устройство обработает 7 пар аргументов: 1, 3, 7, 8, 13, 14, верного ответа нет?
11. Есть два конвейерных ФУ: сложение (4 ступени) и умножение (6 ступеней), все ступени срабатывают за один такт. За сколько тактов будет выполнена векторная операция $A_i = B_i + C_i * s$, $i=1,2,...,60$, с использованием данных устройств в режиме с зацеплением ФУ: 60, 69, 70, 128, 130, 240, 600?
12. Есть два конвейерных ФУ: сложение (4 ступени) и умножение (6 ступеней), все ступени срабатывают за один такт. За сколько тактов будет выполнена векторная операция $A_i = B_i + C_i * s$, $i=1,2,...,60$, с использованием данных устройств без зацепления ФУ: 60, 69, 70, 128, 130, 240, 600?
13. Архитектура компьютеров. Отметьте правильные утверждения: в SMP-компьютерах все процессоры равноправны; архитектуры NUMA и ccNUMA позволяют сохранить единое адресное пространство для параллельной программы; кэш-память явилась причиной возникновения архитектуры NUMA; поиск команд, которые можно выполнять параллельно, в суперскалярных процессорах происходит во время работы программы; параллелизм в классических VLIW-компьютерах выделяется компилятором.
14. Отметьте правильные утверждения про компьютеры: классификация Флинна содержит 3 типа компьютеров; классификация Флинна содержит 4 типа компьютеров; классификация Флинна содержит 6 типов компьютеров; одним из признаков векторно-конвейерного компьютера является многопроцессорность; одним из признаков векторно-конвейерного компьютера является наличие хотя бы одного конвейера; одним из признаков векторно-конвейерного компьютера является наличие векторных регистров; концепция неограниченного параллелизма при развитии компьютерной техники в отдалённом будущем может стать реальностью.
15. Какие из технологических этапов присущи только парадигме параллельного программирования: построение математической модели, декомпозиция, аранжировка, написание программы?
16. Чем декомпозиция по данным отличается от декомпозиции по вычислениям?
17. В чем заключаются достоинства и недостатки статического и динамического способов назначения задач виртуальным исполнителям?
18. Каковы основные цели этапа назначения: сокращение загрузки исполнителей, балансировка загрузки исполнителей, сокращение обменов данными между исполнителями, равномерный обмен данными между исполнителями, сокращение накладных расходов на назначение?
19. Верно ли следующее утверждение: модель передачи сообщений в параллельном программировании применима только на системах с распределенной памятью? Обоснуйте свой ответ.
20. Приведите пример программного кода с внутренним параллелизмом, который, с Вашей точки зрения, нельзя распараллелить автоматически (если такие существуют).
21. В чем заключаются достоинства и недостатки различных подмоделей программирования в модели общей памяти?
22. На каком этапе в параллельной программе задачи назначаются реальным физическим исполнителям: на этапе декомпозиции, на этапе назначения, на этапе аранжировки, на этапе отображения?
23. Современная парадигма параллельного программирования включает: 5, 6, 7, 8, 9, 10 этапов.
24. Перечислите три основных принципа асимптотического анализа алгоритмов.
25. Пусть $T1(n)$ и $T2(n)$ – времена работы последовательных алгоритмов 1 и 2 соответственно. Если $T1(n) = O(T2(n))$, то алгоритм 1: лучше алгоритма 2, не хуже алгоритма 2, хуже алгоритма 2, не хуже алгоритма 2, ничего нельзя сказать.
26. Сформулируйте понятие оптимальности последовательного алгоритма.

27. Какие из нижеперечисленных формулировок относятся к модели вычислительной системы RAM: время доступа к памяти одинаково для всех ячеек, независимо от того рассматривается операция чтения или операция записи; время выполнения всех операций на процессоре считается одинаковым; время выполнения основных операций на процессоре есть $\Theta(1)$?

28. Пусть два последовательных алгоритма решения одной задачи являются оптимальными по поведению. Что можно сказать о реальных временах решения при заданном параметре масштаба N : времена будут одинаковыми, ничего сказать нельзя?

29. Пусть имеется два последовательных алгоритма решения одной и той же задачи — 1 и 2. Алгоритм 1 является оптимальным по поведению, алгоритм 2 не является оптимальным. Что можно сказать о реальных временах решения при заданном параметре масштаба N : время работы алгоритма 1 всегда будет меньше времени работы алгоритма 2, времена будут одинаковыми, ничего сказать нельзя?

30. Используя основную теорему асимптотического анализа, оцените асимптотическое поведение алгоритма сортировки слиянием. Является ли этот алгоритм оптимальным?

31. Какие схемы разрешения конфликта при разрешенной одновременной записи различными исполнителями в одну ячейку памяти в модели вычислительной системы PRAM вы знаете?

32. При вычислении теоретического значения ускорения для параллельных алгоритмов решения некоторой задачи на 4-х исполнителях были получены следующие значения: 2, 3.9, 5. Какие из значений являются корректными? Почему?

33. При вычислении реального значения ускорения для параллельных алгоритмов решения некоторой задачи на 4-х исполнителях были получены следующие значения: 2, 3.9, 5. Какие из значений являются корректными? Почему?

34. Что означает высказывание, «некоторый параллельный алгоритм является оптимальным по стоимости»?

35. Перечислите основные свойства графа алгоритма, реализованного программой.

36. Сколько строгих параллельных форм может существовать у графа алгоритма: 1, ограниченное количество, но больше 1; бесконечно много?

37. Докажите единственность канонической строгой параллельной формы графа алгоритма.

38. Для графа алгоритма, содержащего 500 вершин, высота некоторой строгой параллельной формы равна 25. Что можно сказать о максимальном ускорении, которое может быть получено при реализации алгоритма на параллельной вычислительной системе в модели PRAM — оно будет всегда меньше 20, строго равно 20, может быть больше 20?

39. Для программного текста, реализованного на Фортране 77

```
b(1) = a(1) * 2
```

```
do i = 2, 5
```

```
    b(i) = b(i-1) + a(i) * 2
```

```
enddo
```

постройте граф алгоритма, каноническую строгую параллельную форму и оцените максимально возможное ускорение при реализации алгоритма на параллельной вычислительной системе.

40. Объясните понятия interleaving, недетерминированный и детерминированный набор активностей.

41. Приведите пример набора активностей, для которого нарушены все условия Бернштейна, но который, тем не менее, является детерминированным.

42. Если в программе встречаются два оператора, приведенных в динамическом порядке следования,

```
a = b+c;
```

```
d = e+f;
```

то можно ли их исполнять параллельно? Обоснуйте свое утверждение.

43. Перечислите виды зависимостей между операторами (блоками операторов), которые Вам известны.

44. Какие из видов зависимостей по данным могут являться принципиальными препятствиями к распараллеливанию: истинная зависимость, зависимость по выходным данным, антагонизм?

45. Постройте граф зависимостей по данным между операторами для фрагмента текста программы, расположенного в динамическом порядке следования:

```
S1: x = e + 2z
```

```
S2: y = 2f + x
```

```
S3: z = z + y
```

S4: $y = z + x$

46. Постройте граф зависимостей по данным между операторами для фрагмента текста программы, расположенного в динамическом порядке следования:

S1: $x = x + 2y$

S2: $y = 2f + x$

S3: $z = z + z$

S4: $y = z + x$

47. Для цикла

do $i = 1, u$

S1: $a(i) = d(i) + 5 * i$

S2: $c(i) = a(i-1) * 2$

enddo

определите расстояние зависимости, тип зависимости и возможность распараллеливания.

48. Для цикла

do $i = 1, u$

S1: $a(i) = d(i) + 5 * i$

S2: $c(i) = a(i+1) * 2$

enddo

определите расстояние зависимости, тип зависимости и возможность распараллеливания.

49. Для цикла

do $i = 1, u$

S1: $a(i) = d(i) + 5 * i$

S2: $c(i) = a(i) * 2$

enddo

определите расстояние зависимости, тип зависимости и возможность распараллеливания.

50. С чем связаны основные проблемы определения вектора расстояний при анализе вложенных циклов на зависимость по данным?

51. Для цикла

do $j1 = 1, u1$

do $j2 = 1, u2$

S1: $a(i, j) = b(i, j) * 2$

S2: $c(i, j) = a(i, j - 1) + 1$

enddo

enddo

определите вектор расстояний, вектор направлений и тип зависимости.

52. Для цикла

do $j1 = 1, u1$

do $j2 = 1, u2$

S1: $a(i, j) = b(i, j) * 2$

S2: $c(i, j) = a(i, j + 2) + 1$

enddo

enddo

определите вектор расстояний, вектор направлений и тип зависимости.

53. Для цикла

do $j1 = 1, u1$

do $j2 = 1, u2$

S1: $a(i, j) = b(i, j) * 2$

S2: $c(i, j) = a(i - 1, j + 2) + 1$

enddo

enddo

определите вектор расстояний, вектор направлений и тип зависимости.

54. Для цикла

do $j1 = 1, u1$

do $j2 = 1, u2$

```

S1: a(i, j) = b(i, j) * 2
S2: c(i, j) = a(i + 1, j + 3) + 1
    enddo
enddo

```

определите вектор расстояний, вектор направлений и тип зависимости.

55. Для цикла

```

do j1 = 1,u1
  do j2 = 1,u2
S1: a(i, j) = b(i, j) * 2
S2: c(i, j) = a(i, j + 2) + 1
    enddo
  enddo

```

определите вектор расстояний, вектор направлений и тип зависимости.

56. Для цикла

```

do j1 = 1,u1
  do j2 = 1,u2
S1: a(i, j) = b(i, j) * 2
S2: c(i, j) = a(i, j) + 1
    enddo
  enddo

```

определите вектор расстояний, вектор направлений и тип зависимости.

57. Для цикла

```

do j1 = 1,u1
  do j2 = 1,u2
S1: a(i, j) = b(i, j) * 2
S2: c(i, j) = a(i + 2, j - 3) + 1
    enddo
  enddo

```

определите вектор расстояний, вектор направлений и тип зависимости.

58. Пусть вектор направлений для двумерного цикла имеет вид («=»,«=»). Какие из следующих утверждений являются верными: цикл может быть распараллелен по внешнему индексу без всяких ограничений, по внешнему индексу при условии дублирования входных данных, по внутреннему индексу при условии барьерной синхронизации между итерациями внешнего цикла, по двум индексам без ограничений, не может быть распараллелен?

59. Пусть вектор направлений для двумерного цикла имеет вид («<»,«<»). Какие из следующих утверждений являются верными: цикл может быть распараллелен по внешнему индексу без всяких ограничений, по внешнему индексу при условии дублирования входных данных, по внутреннему индексу при условии барьерной синхронизации между итерациями внешнего цикла, по двум индексам без ограничений, не может быть распараллелен?

60. Пусть вектор направлений для двумерного цикла имеет вид («<»,«<»). Какие из следующих утверждений являются верными: цикл может быть распараллелен по внешнему индексу без всяких ограничений, по внешнему индексу при условии дублирования входных данных, по внутреннему индексу при условии барьерной синхронизации между итерациями внешнего цикла, по двум индексам без ограничений, не может быть распараллелен?

61. Пусть вектор направлений для двумерного цикла имеет вид («=»,«<»). Какие из следующих утверждений являются верными: цикл может быть распараллелен по внешнему индексу без всяких ограничений, по внешнему индексу при условии дублирования входных данных, по внутреннему индексу при условии барьерной синхронизации между итерациями внешнего цикла, по двум индексам без ограничений, не может быть распараллелен?

62. Пусть вектор направлений для двумерного цикла имеет вид («>»,«=»). Какие из следующих утверждений являются верными: цикл может быть распараллелен по внешнему индексу без всяких ограничений, по внешнему индексу при условии дублирования входных данных, по внутреннему индексу при условии барьерной синхронизации между итерациями внешнего цикла, по двум индексам без ограничений, не может быть распараллелен?

63. Пусть вектор направлений для двумерного цикла имеет вид ($\langle \Rightarrow, \langle \Rightarrow \rangle$). Какие из следующих утверждений являются верными: цикл может быть распараллелен по внешнему индексу безо всяких ограничений, по внешнему индексу при условии дублирования входных данных, по внутреннему индексу при условии барьерной синхронизации между итерациями внешнего цикла, по двум индексам без ограничений, не может быть распараллелен?
64. Пусть вектор направлений для двумерного цикла имеет вид ($\langle \Rightarrow, \langle \Rightarrow \rangle$). Какие из следующих утверждений являются верными: цикл может быть распараллелен по внешнему индексу безо всяких ограничений, по внешнему индексу при условии дублирования входных данных, по внутреннему индексу при условии барьерной синхронизации между итерациями внешнего цикла, по двум индексам без ограничений, не может быть распараллелен?
65. Пусть вектор направлений для двумерного цикла имеет вид ($\langle \Leftarrow, \langle \Rightarrow \rangle$). Какие из следующих утверждений являются верными: цикл может быть распараллелен по внешнему индексу безо всяких ограничений, по внешнему индексу при условии дублирования входных данных, по внутреннему индексу при условии барьерной синхронизации между итерациями внешнего цикла, по двум индексам без ограничений, не может быть распараллелен?
66. Пусть вектор направлений для двумерного цикла имеет вид ($\langle \Rightarrow, \langle \Leftarrow \rangle$). Какие из следующих утверждений являются верными: цикл может быть распараллелен по внешнему индексу безо всяких ограничений, по внешнему индексу при условии дублирования входных данных, по внутреннему индексу при условии барьерной синхронизации между итерациями внешнего цикла, по двум индексам без ограничений, не может быть распараллелен?
67. Пусть для некоторого многомерного цикла вектор направлений имеет вид ($\langle \Rightarrow, \langle \Leftarrow, \dots, \langle \Rightarrow \rangle$). Что можно сказать про тип зависимости, обнаруживаемой между операторами после раскрытия цикла: это — истинная зависимость, это — антизависимость, ничего сказать нельзя?
68. Пусть для некоторого многомерного цикла вектор направлений имеет вид ($\langle \Rightarrow, \langle \Leftarrow, \dots, \langle \Rightarrow \rangle$). Что можно сказать про тип зависимости, обнаруживаемой между операторами после раскрытия цикла: это — истинная зависимость, это — антизависимость, ничего сказать нельзя?
69. Пусть для некоторого многомерного цикла вектор направлений имеет вид ($\langle \Leftarrow, \langle \Rightarrow, \dots, \langle \Rightarrow \rangle$). Что можно сказать про тип зависимости, обнаруживаемой между операторами после раскрытия цикла: это — истинная зависимость, это — антизависимость, ничего сказать нельзя?
70. Эквивалентное преобразование программы — это: преобразование динамического порядка следования операторов, сохраняющее результат при любых входных данных; преобразование, сохраняющее граф алгоритма; преобразование, сохраняющее зависимости по данным между операторами, ничего из вышеперечисленного.
71. Почему знание о возможности эквивалентного изменения порядка вложенности во вложенных циклах для распараллеливания программ является важным?
72. Перечислите дополнительные условия в теореме о достаточности приемов разделения цикла, выравнивания цикла и допустимой перестановки операторов в теле цикла для устранения зависимостей, связанных с циклом.
73. Что означает выражение: в цикле нет рекурсивных зависимостей по данным?
74. Справедливо ли утверждение, если между операторами цикла существует истинная зависимость — цикл по итерациям распараллелить невозможно? Обоснуйте ответ.
75. Справедливо ли утверждение, если между операторами невложенного цикла существует истинная зависимость с расстоянием зависимости равным 1 — цикл по итерациям распараллелить принципиально невозможно? Обоснуйте ответ.
76. Какие приемы распараллеливания циклов при наличии зависимости операторов цикла по скалярным переменным являются эквивалентным преобразованием программ: приватизация, «ликвидация» индукционных переменных, редукция?
77. Справедливо ли следующее утверждение: распараллеливание с помощью приема редукции всегда является неэквивалентным преобразованием программы? Обоснуйте свой ответ.
78. Справедливо ли следующее утверждение: распараллеливание с помощью приема «ликвидации» индукционных переменных всегда является неэквивалентным преобразованием программы? Обоснуйте свой ответ.
79. Можно ли устранить в последовательной программе истинные зависимости по данным для получения корректной параллельной версии?
80. Какие схемы распределения работ на этапе назначения между виртуальными исполнителями Вам известны?
81. Система ОДУ не является жесткой и решается явным методом Рунге-Кутты. Будет ли реализованная для такого уравнения последовательная программа эффективно распараллеливаться?

82. Жесткая система обыкновенных дифференциальных уравнений решается W-методом с заменой точного обращения матрицы приближенным по методу Шульца. Какая версия метода допускает более эффективную параллельную реализацию – с несколькими итерациями для вычисления матрицы или с учетом нескольких последовательных степеней невязки?

Билет 1

1. Решаются две близкие задачи – уравнение диффузии и уравнение диффузии с конвекцией. Используются схемы, в которых диффузионный оператор аппроксимируется на верхнем слое по времени, а конвективные члены – на нижнем слое по времени. Можно ли в каждом из этих случаев использовать алгоритм редукции для решения системы линейных уравнений с трехдиагональной матрицей?

2. Решаются две близкие задачи – уравнение диффузии и уравнение диффузии с конвекцией. Используется неявная схема. Можно ли в каждом из этих случаев использовать алгоритм редукции для решения системы линейных уравнений с трехдиагональной матрицей?

Билет 2

1. При реализации параллельной пристрелки реализовано две версии метода. При первой реализации сначала решаются нелинейные уравнения, данные сохраняются, а затем решаются уравнения в вариациях, во второй реализации все эти уравнения решаются одновременно. Какая реализация более эффективна с точки зрения распараллеливания?

2. Есть ли смысл при реализации метода параллельной стрельбы использовать распараллеливание решения системы линейных уравнений в процессе итераций по пристрелочным параметрам?

Билет 3

1. В каком случае метод параллельной стрельбы будет более эффективным по сравнению с методом стрельбы?

2. Будет ли однократно диагонально неявный метод, основанный на методе Розенброка, более эффективным с точки зрения распараллеливания по сравнению со всеми остальными методами того же класса?

Критерии оценивания

Оценка отлично 10 баллов - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины, проявляющему интерес к данной предметной области, продемонстрировавшему умение уверенно и творчески применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично 9 баллов - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично 8 баллов - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений, с некоторыми недочетами.

Оценка хорошо 7 баллов - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но недостаточно грамотно обосновывает полученные результаты.

Оценка хорошо 6 баллов - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности.

Оценка хорошо 5 баллов - выставляется студенту, если он в основном знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач достаточно большое количество неточностей.

Оценка удовлетворительно 4 балла - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он освоил основные разделы учебной программы, необходимые для дальнейшего обучения, и может применять полученные знания по образцу в стандартной ситуации.

Оценка удовлетворительно 3 балла - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, допускающему ошибки в формулировках базовых понятий, нарушения логической последовательности в изложении программного материала, слабо владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и с трудом применяет полученные знания даже в стандартной ситуации.

Оценка неудовлетворительно 2 балла - выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных принципов и не умеет использовать полученные знания при решении типовых задач.

Оценка неудовлетворительно 1 балл - выставляется студенту, который не знает основного содержания учебной программы дисциплины, допускает грубейшие ошибки в формулировках базовых понятий дисциплины и вообще не имеет навыков решения типовых практических задач.

5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Во время проведения дифференциального зачета обучающиеся могут пользоваться программой дисциплины, а также справочной литературой, вычислительной техникой, конспектами лекций.

Дифференциальный зачет проводится путем организации специального опроса, проводимого в устной форме.