

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



«УТВЕРЖДАЮ

**Проректор по учебной работе
и экономическому развитию**

_____ **Д.А. Зубцов**

Рабочая программа дисциплины (модуля)

по дисциплине: Практикум по параллельному программированию
по направлению:
профиль подготовки:

кафедра информатики и вычислительной математики

курс: 4

квалификация:

Семестр, формы промежуточной аттестации: 8(Весенний) - Дифференцированный зачет

Аудиторных часов: 30 всего, в том числе:
лекции: 0 час.
практические и семинарские занятия: 0 час.
лабораторные занятия: 30 час.

Самостоятельная работа: 6 час.

Всего часов: 36, всего зач. ед.: 1

Программу составили:

В.Е. Карпов, канд. физ.-мат. наук, доцент, доцент
А.И. Лобанов, д-р физ.-мат. наук, профессор, доцент

Программа обсуждена на заседании кафедры

15 января 2016 г.

СОГЛАСОВАНО:

Заведующий кафедрой

И.Б. Петров

Начальник учебного управления

И.Р. Гарайшина

1. Цели и задачи

Цель дисциплины

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. Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Практикум по параллельному программированию» включает в себя разделы, которые могут быть отнесены к вариативной части цикла Б.3 кода УЦ ООП.

Дисциплина «Практикум по параллельному программированию» базируется на дисциплинах:

Дифференциальные уравнения;
Математический анализ;
Информатика.

Дисциплина «Практикум по параллельному программированию» предшествует изучению дисциплин:

Проектирование интерактивных систем;
Управление проектами разработки программ;
Методы анализа данных и распознавания.

3. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Освоение дисциплины направлено на формирование следующих общекультурных, общепрофессиональных и профессиональных компетенций:

- способность применять теорию и методы математики для построения качественных и количественных моделей объектов и процессов в естественнонаучной сфере деятельности (ОПК-2);
- способность понимать ключевые аспекты и концепции в области их специализации (ОПК-3);
- способность применять полученные знания для анализа систем, процессов и методов (ОПК-4);
- способность выбирать и применять подходящее оборудование, инструменты и методы исследований для решения задач в избранной предметной области (ПК-3);
- способность критически оценивать применимость применяемых методик и методов (ПК-4).

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;
- способы организации работы пользователей на современных многопроцессорных вычислительных системах и языки управления заданиями для них.

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;
- самостоятельно разрабатывать и запускать параллельные приложения на современных вычислительных комплексах.

владеть:

- приемами распараллеливания алгоритмов и программ;
- средствами и технологиями разработки приложений, обеспечивающих проведение параллельного вычислительного эксперимента;
- навыками отладки и запуска параллельных приложений для проведения 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	Виды учебных занятий, включая самостоятельную работу				
		Лекции	Практические и семинарские занятия	Лаборат. работы	Задания, курсовые работы	Самост. работа
1	Проблемы эволюции вычислительных систем. Парадигмы последовательного и параллельного программирования.			4		1
2	Элементы асимптотического анализа алгоритмов			2		1
3	Архитектура современных параллельных кластеров. Язык управления заданиями PBS.			2		
4	Среда программирования MPI. Общие принципы построения. Понятие сообществ исполнителей.			2		

5	Простейшие MPI программы. Запуск одной и той же программы на нескольких исполнителях. Передача информации по кругу.			2		
6	Ярусно-параллельные формы.			2		
7	Реализация последовательных и параллельных программ расчета значений элементарных функций.			4		
8	Укрупнение ярусов. Параллельность циклов			6		
9	Анализ и распараллеливание программ, содержащих циклы.			4		
10	Статическое и динамическое распределение работ между исполнителями. Решение одномерного уравнения теплопроводности			2		4
Итого часов				30		6
Подготовка к экзамену		0 час.				
Общая трудоёмкость		36 час., 1 зач.ед.				

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

Семестр: 8 (Весенний)

1. Проблемы эволюции вычислительных систем. Парадигмы последовательного и параллельного программирования.

Три кризиса в развитии математического обеспечения. Архитектурный и программный параллелизм. Проблемы использования параллельных систем. Парадигма последовательного программирования. Модели последовательного программирования. Парадигма параллельного программирования. Этапы декомпозиции, назначения, оркестрирования, отображения. Задачи, решаемые на каждом этапе. Модели параллельного программирования.

2. Элементы асимптотического анализа алгоритмов

Основные предположения. Вычислительная модель RAM. Терминология и обозначения. Асимптотические отношения. Оптимальный по поведению последовательный алгоритм. Пример асимптотического анализа сложности последовательного алгоритма выбора элемента из множества. Рекуррентные соотношения. Основная теорема асимптотического анализа. Расширенная квалификация Флинна. Примеры SISD, SIMD, MISD, MIMD машин. 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. Архитектура современных параллельных кластеров. Язык управления заданиями PBS.

Архитектура современных параллельных кластеров. Язык управления заданиями PBS.

4. Среда программирования MPI. Общие принципы построения. Понятие сообществ исполнителей.

Среда программирования MPI. Общие принципы построения. Понятие сообществ исполнителей.

5. Простейшие MPI программы. Запуск одной и той же программы на нескольких исполнителях. Передача информации по кругу.

Простейшие MPI программы. Запуск одной и той же программы на нескольких исполнителях. Передача информации по кругу.

6. Ярусно-параллельные формы.

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.

7. Реализация последовательных и параллельных программ расчета значений элементарных функций.

Групповые операции в MPI. Широковещательная рассылка и редукция. Барьерная синхронизация.

8. Укрупнение ярусов. Параллельность циклов

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. 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.

9. Анализ и распараллеливание программ, содержащих циклы.

Анализ и распараллеливание программ, содержащих циклы.

10. Статическое и динамическое распределение работ между исполнителями. Решение одномерного уравнения теплопроводности

Статическое и динамическое распределение работ между исполнителями. Решение одномерного уравнения теплопроводности

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

Учебная аудитория, оснащенная мультимедиапроектором и экраном для чтения лекций.
Учебный компьютерный класс с выходом на ЭВМ (кластер) с установленной средой программирования MPI и компилятором, поддерживающим OpenMP.

6. Перечень основной и дополнительной литературы, необходимой для освоения дисциплины (модуля)

Основная литература

1. Карпов В. Е., Лобанов А.И. Численные методы, алгоритмы и программы. Введение в распараллеливание. — М., Физматкнига, 2014. — 192 с.
2. Антонов А.С. Технологии параллельного программирования MPI и OpenMP. — М., Издательство Московского университета, 2012. — 344с.

Дополнительная литература

1. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. — СПб.: БХВ-Петербург, 2002. — 608 с.
2. Гергель В П. Высокопроизводительные вычисления для многопроцессорных многоядерных систем. – М., Изд-во МГУ, 2010. – 544 с.
3. H. Jordan, G. Alaghband. Fundamentals of Parallel Processing. — Pearson Education, Inc., 2003. — 536 p.
4. Selim G. Aki. The Design and Analysis of Parallel Algorithms. — Prentice-Hall, Inc., 1989. — 401 p.
5. Claudia Leopold. Parallel and Distributed Computing. — John Wiley & Sons, Inc., 2001. — 260 p.
6. Миллер Р., Боксер Л. Последовательные и параллельные алгоритмы. — М.: Бинوم. Лаборатория знаний, 2006. — с. 406.

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

<http://www.intuit.ru>, <http://www.parallels.ru>, <http://www.hpc-education.ru>

8. Методические указания для обучающихся по освоению дисциплины

Студент, изучающий курс «Практикум по параллельному программированию» 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, методы доказательств, методы исследования асимптотического поведения алгоритмов, методы анализа программ и алгоритмов на параллельность, уметь выбирать правильный численный метод для параллельного решения задач математического моделирования, уметь использовать среды программирования MPI и OpenMP.

Успешное освоение курса требует напряжённой 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 учебной и научной литературе), подготовку ответов на вопросы, предназначенных для самостоятельного изучения, доказательство отдельных утверждений, свойств;
- решение задач, предлагаемых студентам на лабораторных работах;
- подготовку к лабораторным работам и зачету.

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

Показателем владения материалом служит умение решать задачи. Для формирования умения применять теоретические знания на практике студенту необходимо решать как можно больше задач. При решении задач каждое действие необходимо аргументировать, ссылаясь на известные теоретические сведения.

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

9. Фонд оценочных средств для проведения промежуточной аттестации по итогам обучения

Приложение

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ
ПО ДИСЦИПЛИНЕ**

по направлению:

профиль подготовки:

кафедра информатики и вычислительной математи

курс:

4

квалификация:

Семестр, формы промежуточной аттестации: 8(Весенний) - Дифференцированный зачет

Разработчики:

В.Е. Карпов, канд. физ.-мат. наук, доцент, доцент

А.И. Лобанов, д-р физ.-мат. наук, профессор, доцент

1. Компетенции, формируемые в процессе изучения дисциплины

Освоение дисциплины направлено на формирование у обучающегося следующих общекультурных (ОК), общепрофессиональных (ОПК) и профессиональных (ПК) компетенций:

способность применять теорию и методы математики для построения качественных и количественных моделей объектов и процессов в естественнонаучной сфере деятельности (ОПК-2);

способность понимать ключевые аспекты и концепции в области их специализации (ОПК-3);

способность применять полученные знания для анализа систем, процессов и методов (ОПК-4);

способность выбирать и применять подходящее оборудование, инструменты и методы исследований для решения задач в избранной предметной области (ПК-3);

способность критически оценивать применимость применяемых методик и методов (ПК-4).

2. Показатели оценивания компетенций

В результате изучения дисциплины «Практикум по параллельному программированию» 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;
- способы организации работы пользователей на современных многопроцессорных вычислительных системах и языки управления заданиями для них.

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;
- самостоятельно разрабатывать и запускать параллельные приложения на современных вычислительных комплексах.

владеть:

- приемами распараллеливания алгоритмов и программ;
- средствами и технологиями разработки приложений, обеспечивающих проведение параллельного вычислительного эксперимента;
- навыками отладки и запуска параллельных приложений для проведения вычислительного эксперимента.

3. Перечень типовых контрольных заданий, используемых для оценки знаний, умений, навыков

Промежуточная аттестация по дисциплине «Практикум по параллельному программированию» осуществляется в форме дифференцированного зачета. Зачет проводится в письменной форме.

Перечень контрольных вопросов:

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 эpsilon; 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 $A_i = V_i + C_i * s$, $i=1,2,\dots,60$, 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 $A_i = V_i + C_i * s$, $i=1,2,\dots,60$, 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 $T_1(n)$ и $T_2(n)$ – времена работы последовательных алгоритмов 1 и 2 соответственно. Если $T_1(n) = O(T_2(n))$, 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 принципиальным препятствиям к распараллеливанию: истинная зависимость, зависимость по выходным данным, антизависимость?

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 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?

- операторами после раскрытия цикла: это — истинная зависимость, это — антизависимость, ничего сказать нельзя?
68. Пусть для некоторого многомерного цикла вектор направлений имеет вид ($\langle\langle\rangle\rangle, \langle\langle\rangle\rangle, \dots, \langle\langle\rangle\rangle$). 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 ($\langle\langle\rangle\rangle, \langle\langle\rangle\rangle, \dots, \langle\langle\rangle\rangle$). 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?

Примеры контрольных заданий:

Контрольное задание состоит из 10 слегка измененных контрольных вопросов, охватывающих все разделы дисциплины.

4. Критерии оценивания

За каждый контрольный вопрос из контрольного задания студент получает от 0 до 4 баллов в зависимости от полноты представленного ответа (решения). Количество набранных баллов определяет оценку за зачет:

Оценка	Набранные баллы
отлично (10)	более 36
отлично (9)	от 32 до 36 включительно
отлично (8)	от 28 до 32 включительно
хорошо (7)	от 24 до 28 включительно
хорошо (6)	от 20 до 24 включительно
хорошо (5)	от 16 до 20 включительно
удовлетворительно (4)	от 12 до 16 включительно
удовлетворительно (3)	от 8 до 12 включительно
неудовлетворительно (2)	от 4 до 8 включительно
неудовлетворительно (1)	не более 4

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

Время проведения письменного зачета составляет 2 академических часа.

Во время проведения зачета обучающиеся могут пользоваться программой дисциплины.