

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
имени В.Ф. Уткина

Кафедра электронных вычислительных машин

«СОГЛАСОВАНО»


Декан факультета ВТ

Д.А. Перепелкин
« » 2020 г.



«УТВЕРЖДАЮ»

Проректор по РОПиМД
А.В. Корячко
« » 2020 г.

Заведующий кафедрой ЭВМ

Б.В. Костров
« » 2020 г.

РАБОЧАЯ ПРОГРАММА
дисциплины

Б1.В.ДВ.02.01 «Параллельное программирование»

Направления подготовки:

02.03.03 «Математическое обеспечение и администрирование информационных систем»

ОПОП академического бакалавриата
«Математическое обеспечение
и администрирование информационных систем»

Квалификация выпускника – бакалавр

Форма обучения – очно-заочная

Рязань 2020 г.

ЛИСТ СОГЛАСОВАНИЙ

Рабочая программа составлена с учетом требований федерального государственного образовательного стандарта высшего образования по направлению подготовки (специальности) 02.03.03 Математическое обеспечение и администрирование информационных систем (уровень бакалавриата), утвержденным приказом Минобрнауки России от 23.08.2017г. № 809.

Программу составил
к.т.н., доц. кафедры
«Электронные вычислительные машины»



С.И. Елесина

Программа рассмотрена и одобрена на заседании кафедры ЭВМ
«14» 06 2020 г., протокол № 10

Заведующий кафедрой
«Электронные вычислительные машины»,
д.т.н., проф. кафедры ЭВМ



Б.В. Костров

1 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Рабочая программа по дисциплине «Параллельное программирование» является составной частью основной профессиональной образовательной программы (ОПОП) по направлению подготовки бакалавров 02.03.03 – «Математическое обеспечение и администрирование информационных систем», разработанной в соответствии с Федеральным государственным образовательным стандартом высшего образования по направлению подготовки 02.03.03 – «Математическое обеспечение и администрирование информационных систем» (уровень бакалавриата), утвержденным приказом Минобрнауки России от 23.08.2017 г. № 809.

Цель дисциплины: формирование у студентов теоретических знаний и навыков разработки и отладки параллельных программ для различных современных архитектур.

Задачами данного курса являются:

- освоение студентами базовых знаний (понятий, концепций, методов и моделей) в области параллельного программирования;
- приобретение теоретических знаний и практических умений и навыков в области параллельного программирования.

Перечень планируемых результатов обучения по дисциплине

Выпускник, освоивший программу бакалавриата, должен обладать следующими компетенциями:

Код компетенции	Содержание компетенций	Перечень планируемых результатов обучения по дисциплине
ПК-1	Способен проектировать программное обеспечение с использованием современных инструментальных средств	<p>ПК-1.1. Знает основные принципы и приемы построения архитектуры программного обеспечения; методы и средства проектирования программного обеспечения, в том числе баз данных.</p> <p>ПК-1.2. Умеет использовать существующие типовые решения и шаблоны проектирования программного обеспечения, применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов.</p> <p>ПК-1.3. Имеет практический опыт построения архитектуры программного обеспечения, проектирования структур данных, баз данных и программных интерфейсов с использованием современных инструментальных средств</p>

ПК-6	Способен использовать современные методы разработки и реализации конкретных алгоритмов математических моделей на базе языков программирования и пакетов прикладных программ моделирования	ПК-6.1. Знает современные методы разработки и реализации алгоритмов математических моделей на базе языков и пакетов прикладных программ моделирования. ПК-6.2. Умеет разрабатывать и реализовывать алгоритмы математических моделей на базе языков и пакетов прикладных программ моделирования. ПК-6.3. Имеет практический опыт разработки и реализации алгоритмов их на базе языков и пакетов прикладных программ моделирования.
------	---	---

2 Место дисциплины в структуре образовательной программы

Дисциплина «Параллельное программирование» относится к дисциплине по выбору Б1.В.ДВ.02.01 основной образовательной программы подготовки бакалавров направления 02.03.03 «Математическое обеспечение и администрирование информационных систем».

Дисциплина изучается по очно-заочной форме обучения на 3 курсе в 6 семестре.

Дисциплина базируется на следующих дисциплинах: «Основы алгоритмизации и объектно-ориентированное программирование» и «Архитектура ЭВМ и вычислительных систем», «Сети и телекоммуникации».

Знания, полученные в результате освоения дисциплины будут полезны обучающемуся при изучении дисциплин: «Операционные системы и оболочки», «Программирование распределенных систем».

Материал дисциплины «Параллельное программирование» формирует методологические и организационные основы для выполнения обучающимися практик и выпускной квалификационной работы.

3 Объём дисциплины и виды учебной работы (часы)

Объём дисциплины в зачетных единицах (ЗЕ): 3 ЗЕ, 108 часов.

Вид учебной работы	Всего часов
Общая трудоемкость дисциплины	108
1. Контактная работа обучающихся с преподавателем (всего), в том числе:	48,25
лекции	16
практические занятия	16
лабораторные работы	16
консультации	-
иная контактная работа (промежуточная аттестация)	0,25
2. Самостоятельная работа обучающихся (всего), в том числе:	51
курсовой проект (работа)	-
иная самостоятельная работа	51
3. Контроль	8,75
Вид промежуточной аттестации обучающегося	зачет

4 Содержание дисциплины, структурированное по темам с указанием отведенного на них количества академических часов и видов учебных занятий

4.1 Содержание дисциплины, структурированное по темам

Тема 1. Общая характеристика параллельных вычислительных систем.

. Классификации архитектур вычислительных систем: классификация Флинна, классификация Ванга и Бриггса и др. Мультипроцессоры. Мультикомпьютеры.

Эволюция и тенденции развития архитектур современных вычислительных систем: многопоточность и многоядерность.

Тема 2. Параллельные вычисления

Уровни параллелизма. Степень гранулярности. Метрики параллельных вычислений. Профиль параллелизма программы. Основные метрики параллелизма: ускорение, эффективность, избыточность и т.д. Закономерности параллельных вычислений. Закон Амдала. Закон Густафсона. Закон Сана-Ная. Метрика Карпа-Флэтта. Моделирование и анализ параллельных вычислений.

Тема 3. Технология параллельного программирования OpenMP

Введение. Основные понятия. Параллельные и последовательные области. Модель данных. Распределение работы. Синхронизация. Параллельные алгоритмы сортировки данных. Параллельные методы матричного умножения.

Тема 4. Технология параллельного программирования для графических процессоров CUDA

Гибридная модель вычислений. Архитектура графического процессора (GPU). Обзор средств программирования для GPU. Программная модель CUDA. Понятие потока, блока, сети блоков. Функция-ядро. Иерархия памяти. Регистры и локальная память. Глобальная память. CUDA-потоки. Разделяемая память. Константная память. Текстурная память. .

Тема 5. Перспективные направления в параллельном программировании

Сопряженные процессоры. Грид-вычисления. Транзакционная память. Модель MapReduce. Развивающие языки.

4.2. Разделы дисциплины и трудоемкость по видам учебных занятий (в академических часах).

Название раздела	Общая трудоемкость, всего часов	Контактная работа обучающихся с преподавателем						Самостоятельная работа обучающихся	Контроль
		Всего	Лекции	Практические занятия	Лабораторные работы	Консультации	Иные виды контактной работы		
Тема 1. Общая характеристика параллельных вычислительных систем	6	2	2	0	0	-	-	3	1
Тема 2. Параллельные вычисления	7	2	2	0	0	-	-	3	2
Тема 3. Технология	43	20	4	8	8	-	-	21	2

Название раздела	Общая трудоемкость, всего часов	Контактная работа обучающихся с преподавателем						Самостоятельная работа обучающихся	Контроль
		Всего	Лекции	Практические занятия	Лабораторные работы	Консультации	Иные виды контактной работы		
параллельного программирования OpenMP									
Тема 4. Технология параллельного программирования для графических процессоров CUDA	45,75	22	6	8	8	-	-	21	2,75
Тема 5. Перспективные направления в параллельном программировании	6	2	2	0	0	-	-	3	1
Промежуточная аттестация	0,25	0,25	-	-	-	-	0,25	-	-
Итого	108	48,35	16	16	16	-	0,25	51	8,75

Виды лабораторных, практических и самостоятельных работ

Тема	Вид работы	Наименование и содержание работы	Трудоемкость, часов
Тема 1. Вычислительные системы. Основные понятия и классификация	Самостоятельная работа	Изучение конспекта лекций	3
		Подготовка к зачету	1
Тема 2. Параллельные вычисления	Самостоятельная работа	Изучение конспекта лекций Подготовка к зачету	3 2
Технология параллельного программирования OpenMP	Практическая работа	Изучение классов переменных OpenMP. Параллельные и последовательные области. Директивы parallel, single, master	2
		Конструкции распределения работы. Исследование опции schedule. Решение задач	4
		Синхронизация в OpenMP.	2
	Лабораторная работа	Сортировка массивов	4
		Работа с матрицами	4
	Самостоятельная работа	Изучение конспекта лекций Изучение методических указаний, подготовка к практическим занятиям	

Тема	Вид работы	Наименование и содержание работы	Трудо-емкость, часов
		и лабораторным работам	21
		Подготовка к зачету	2
Технология параллельного программирования для графических процессоров CUDA	Практическая работа	Начало работы. Модель программирования. Типы памяти в Cuda. Расширения языка.	2
		Реализация на CUDA базовой операции над массивами – REDUCE	2
		Распараллеливание с помощью блоков, нитей, блоков и нитей. Решение задач.	4
	Лабораторная работа	Работа с разделяемой памятью. Атомарные операции. Вычисление гистограммы на CPU. Решение задач.	4
		Сортировка массивов	4
	Самостоятельная работа	Изучение конспекта лекций Изучение методических указаний, подготовка к практическим занятиям и лабораторным работам Подготовка к зачету	21 2,75
Перспективные направления в параллельном программировании	Самостоятельная работа	Изучение конспекта лекций	3
		Подготовка к зачету	1

5 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

1) Левин М.П. Параллельное программирование с использованием OpenMP [Электронный ресурс] / М.П. Левин. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 133 с. — 978-5-94774-857-4. — Режим доступа: <http://www.iprbookshop.ru/52216.html> - ЭБС "IPRbooks", по паролю (дата обращения: 24.06.2019)

2) Параллельные вычисления на GPU. Архитектура и программная модель CUDA [Электронный ресурс] : учебное пособие / А.В. Боресков [и др.]. — Электрон. текстовые данные. — М. : Московский государственный университет имени М.В. Ломоносова, 2015. — 336 с. — 978-5-19-011058-6. — Режим доступа: <http://www.iprbookshop.ru/54647.html> - ЭБС "IPRbooks", по паролю (дата обращения: 24.06.2019).

6 Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Фонд оценочных средств приведен в Приложении к рабочей программе дисциплины (см. документ «Оценочные материалы по дисциплине «Параллельное программирование»).

7 Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная учебная литература:

- 1) Цилькер, Б.Я. Организация ЭВМ и систем : Учеб.для вузов. - СПб.:Питер, 2006. - 667с. (198 экз.)
- 2) Мищенко В.К. Архитектура высокопроизводительных вычислительных систем [Электронный ресурс] : учебное пособие / В.К. Мищенко. — Электрон. текстовые данные. — Новосибирск: Новосибирский государственный технический университет, 2013. — 40 с. — 978-5-7782-2365-3. — Режим доступа: <http://www.iprbookshop.ru/44898.html> - ЭБС "IPRbooks", по паролю (дата обращения: 24.06.2019)
- 3) Барский А.Б. Архитектура параллельных вычислительных систем [Электронный ресурс] / А.Б. Барский. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 297 с. — 978-5-94774-546-7. — Режим доступа: <http://www.iprbookshop.ru/73821.html> - ЭБС "IPRbooks", по паролю (дата обращения: 24.06.2019)
- 4) Параллельные вычисления общего назначения на графических процессорах [Электронный ресурс] : учебное пособие / К.А. Некрасов [и др.]. — Электрон. текстовые данные. — Екатеринбург: Уральский федеральный университет, 2016. — 104 с. — 978-5-7996-1722-6. — Режим доступа: <http://www.iprbookshop.ru/69657.html> - ЭБС "IPRbooks", по паролю (дата обращения: 24.06.2019)
- 5) Гергель В.П. Теория и практика параллельных вычислений [Электронный ресурс] / В.П. Гергель. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 480 с. — 978-5-94774-645-7. — Режим доступа: <http://www.iprbookshop.ru/57385.html> - ЭБС "IPRbooks", по паролю (дата обращения: 24.06.2019)
- 6) Intel Parallel Programming Professional (Introduction) [Электронный ресурс] / В.П. Гергель [и др.]. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 568 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/52170.html> - ЭБС "IPRbooks", по паролю (дата обращения: 24.06.2019)

Дополнительная учебная литература:

- 1) Калачев А.В. Многоядерные процессоры [Электронный ресурс] / А.В. Калачев. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 351 с. — 978-5-9963-0349-6. — Режим доступа: <http://www.iprbookshop.ru/62816.html> - ЭБС "IPRbooks", по паролю (дата обращения: 24.06.2019)
- 2) Николаев Е.И. Параллельные вычисления [Электронный ресурс] : учебное пособие / Е.И. Николаев. — Электрон. текстовые данные. — Ставрополь: Северо-Кавказский федеральный университет, 2016. — 185 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/66086.html> - ЭБС "IPRbooks", по паролю (дата обращения: 24.06.2019)
- 3) Федотов И.Е. Приемы параллельного программирования [Электронный ресурс] : учебное пособие / И.Е. Федотов. — Электрон. текстовые данные. — М. : Российский новый университет, 2009. — 184 с. — 978-5-89789-048-4. — Режим доступа: <http://www.iprbookshop.ru/21300.html> - ЭБС "IPRbooks", по паролю (дата обращения: 24.06.2019)

4) Модель программирования CUDA [Электронный ресурс] : учебник / В.В. Коробицын [и др.]. — Электрон. текстовые данные. — Омск: Омский государственный университет им. Ф.М. Достоевского, 2012. — 256 с. — 978-5-7779-1489-7. — Режим доступа: <http://www.iprbookshop.ru/24903.html> - ЭБС "IPRbooks", по паролю (дата обращения: 24.06.2019)

8 Перечень ресурсов информационно-телекоммуникационной сети «интернет», необходимых для освоения дисциплины

Обучающимся предоставлена возможность индивидуального доступа к следующим электронно-библиотечным системам:

Электронно-библиотечная система «IPRbooks», режим доступа – с любого компьютера РГРТУ без пароля, из сети интернет по паролю. – URL: <https://iprbookshop.ru/>.

9 Методические указания для обучающихся по освоению дисциплины

Указания в рамках лекций

Во время лекции студент должен вести краткий конспект.

Работа с конспектом лекций предполагает просмотр конспекта в тот же день после занятий. При этом необходимо пометить материалы конспекта, которые вызывают затруднения для понимания. При этом обучающийся должен стараться найти ответы на затруднительные вопросы, используя рекомендуемую литературу. Если ему самостоятельно не удалось разобраться в материале, необходимо сформулировать вопросы и обратиться за помощью к преподавателю на консультации или ближайшей лекции.

Обучающимся необходимо регулярно отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам.

Указания в рамках лабораторных работ

Лабораторные работы составляют важную часть профессиональной подготовки студентов. Основная цель проведения лабораторных работ – формирование у студентов аналитического и творческого мышления путем приобретения практических навыков.

Методические указания к лабораторным работам по дисциплине наряду с рабочей программой и графиком учебного процесса относятся к методическим документам, определяющим уровень организации и качества образовательного процесса. Содержание лабораторных работ фиксируется в рабочей программе дисциплины.

При подготовке к лабораторным работам необходимо просмотреть конспекты лекций и методические указания, рекомендованную литературу по данной теме, а так же подготовиться к ответу на контрольные вопросы.

В ходе выполнения индивидуального задания лабораторной работы студент готовит отчет о работе (с помощью офисного пакета LibreOffice или в другом редакторе доступном студенту, схемы алгоритмов выполняются с помощью векторного графического редактора Microsoft Office Visio). В отчете отражаются цель работы, индивидуальное задание, постановка задачи, разработка алгоритма решения поставленной задачи и программа. Должны быть подготовлены тестовые наборы данных и отражены результаты работы программы.

За 10 минут до окончания занятия преподаватель проверяет объем выполненной за занятие работы и отмечает результат в рабочем журнале. Оставшиеся невыполненными пункты задания студент обязан доделать самостоятельно.

После проверки отчета преподаватель может проводить устный или письменный опрос студентов для контроля усвоения ими основных теоретических и практических знаний по теме занятия (студенты должны знать смысл полученных ими результатов и ответы на контрольные вопросы). По результатам проверки отчета и опроса выставляется оценка за лабораторную работу.

Указания в рамках практических занятий

Перед выполнением практического занятия необходимо внимательно ознакомиться с заданием. Желательно заранее выполнить подготовку проекта в инструментальной среде, чтобы на практическом занятии осталось время для сдачи работы.

Перед сдачей работы рекомендуется ознакомиться со списком вопросов изучаемой темы и попытаться самостоятельно на них ответить, используя конспект лекций и рекомендуемую литературу. Таким образом, вы сможете сэкономить свое время и время преподавателя.

Кроме чтения учебной литературы из обязательного списка рекомендуется активно использовать информационные ресурсы сети Интернет по изучаемой теме. Ответы на многие вопросы, связанные с разработкой параллельных программ вы можете получить в сети Интернет, посещая соответствующие информационные ресурсы.

Указания в рамках подготовки к промежуточной аттестации

При подготовке к зачету в дополнение к изучению конспектов лекций, учебных пособий, слайдов и другого раздаточного материала предусмотренного рабочей программой дисциплины, необходимо пользоваться учебной литературой, рекомендованной к настоящей рабочей программе. При подготовке к зачету нужно изучить теорию: определения всех понятий и подходы к оцениванию до состояния понимания материала и самостоятельно решить по нескольким типовым задач из каждой темы (в том случае если тема предусматривает решение задач). При решении задач всегда необходимо уметь качественно интерпретировать итог решения.

Указания в рамках самостоятельной работы студентов

Самостоятельная работа студентов по дисциплине играет важную роль в ходе всего учебного процесса. Методические материалы и рекомендации для обеспечения самостоятельной работы студентов готовятся преподавателем и выдаются студентам в виде раздаточных материалов или оформляются в виде электронного ресурса используемого в рамках системы дистанционного обучения ФГБОУ ВО «РГРТУ».

Самостоятельное изучение тем учебной дисциплины способствует:

- закреплению знаний, умений и навыков, полученных в ходе аудиторных занятий;
- углублению и расширению знаний по отдельным вопросам и темам дисциплины;
- освоению умений прикладного и практического использования полученных знаний.

Самостоятельная работа как вид учебной работы может использоваться на лекциях и практических и лабораторных работах, а также иметь самостоятельное значение – внеаудиторная самостоятельная работа обучающихся – при подготовке к лекциям, практическим и лабораторным занятиям, а также к экзамену.

Основными видами самостоятельной работы по дисциплине являются:

- самостоятельное изучение отдельных вопросов и тем дисциплины;
- подготовка к выполнению лабораторных работ и практическим занятиям;

- подготовка к защите лабораторных работ и практических занятий, оформление отчета.

Рекомендации по работе с литературой

Теоретический материал курса становится более понятным, когда дополнительно к прослушиванию лекции и изучению конспекта, изучается дополнительная рекомендованная литература. Литературу по курсу рекомендуется изучать в библиотеке, с использованием доступной электронной библиотечной системы или с помощью сети Интернет (источники, которые могут быть использованы без нарушения авторских прав).

10 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

При проведении занятий по дисциплине используются следующие информационные технологии:

- удаленные информационные коммуникации между студентами и преподавателем, ведущим лекционные и практические занятия, посредством информационной образовательной среды ФГБОУ ВО «РГРТУ», позволяющие осуществлять оперативный контроль графика выполнения и содержания образовательного процесса, решение организационных вопросов, консультирование;
- доступ к современным профессиональным базам данных (в том числе международным реферативным базам данных научных изданий) и информационным справочным системам;
- проведение аудиторных занятий с использованием презентаций и раздаточных материалов в электронном виде;
- выполнение студентами различных видов учебных работ с использованием лицензионного программного обеспечения, установленного на рабочих местах студента в компьютерных классах и в помещениях для самостоятельной работы, а также для выполнения самостоятельной работы в домашних условиях.

Обучающимся по данной дисциплине предоставляется доступ к дистанционным курсам, расположенным в системе дистанционного обучения ФГБОУ ВО «РГРТУ»: «Параллельное программирование» [Электронный ресурс]. URL: <http://cdo.rsreu.ru/course/view.php?id=1283>

Система дистанционного обучения ФГБОУ ВО «РГРТУ» доступна как из внутренней информационной системы организации, так и из глобальной сети Интернет.

Перечень лицензионного программного обеспечения:

- 1) Операционная система Windows 7 (лицензия DreamSpark Membership ID 700102019);
- 2) Open Office (лицензия Apache License, Version 2.0);
- 3) Интегрированная среда разработки Visual Studio C++ 13 (лицензия DreamSpark Membership ID 700102019);
- 4) Программно-аппаратная архитектура параллельных вычислений CUDA (free software, EULA for the NVIDIA CUDA Toolkit, NVIDIA Software License Agreement + CUDA Supplement to Software License Agreement).

Перечень профессиональных баз данных (в том числе международным реферативным базам данных научных изданий) и информационных справочных систем:

- 1) Информационно-правовой портал ГАРАНТ.РУ [Электронный ресурс]. – URL: <http://www.garant.ru>. – Режим доступа: свободный доступ (дата обращения 02.02.2017).
- 2) Справочная правовая система КонсультантПлюс [Электронный ресурс]. – URL: <http://www.consultant.ru/online/>. – Режим доступа: свободный доступ (будние дни – 20.00 - 24.00, выходные и праздничные дни – круглосуточно) (дата обращения 02.02.2017).

11 Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Для освоения дисциплины необходимы:

- 1) для проведения лекционных занятий необходима аудитория с достаточным количеством посадочных мест, соответствующая необходимым противопожарным нормам и санитарно-гигиеническим требованиям;
- 2) для проведения практических занятий необходим класс персональных компьютеров с видеокартами с поддержкой CUDA, с инсталлированными операционными системами Microsoft Windows 7 (или выше), и установленным лицензионным программным обеспечением Open Office, Visual Studio 13, CUDA.
- 3) для проведения лекций аудитория должна быть оснащена проекционным оборудованием.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Рязанский государственный радиотехнический университет
имени В.Ф. Уткина»

КАФЕДРА ЭЛЕКТРОННЫХ ВЫЧИСЛИТЕЛЬНЫХ МАШИН

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

по дисциплине

Б1.В.ДВ.02.01 «Параллельное программирование»

Направления подготовки:

02.03.03 «Математическое обеспечение и администрирование информационных систем»

ОПОП академического бакалавриата
«Математическое обеспечение
и администрирование информационных систем»

Квалификация выпускника – бакалавр

Форма обучения – очно-заочная

Рязань, 2020 г.

Общие положения

Оценочные материалы – это совокупность учебно-методических материалов (практических заданий, описаний форм и процедур проверки), предназначенных для оценки качества освоения обучающимися данной дисциплины как части ОПОП.

Цель – оценить соответствие знаний, умений и владений, приобретенных обучающимся в процессе изучения дисциплины, целям и требованиям ОПОП в ходе проведения промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности общепрофессиональных и профессиональных компетенций.

Контроль знаний обучающихся проводится в форме промежуточной аттестации. Промежуточная аттестация проводится в форме зачета. Форма проведения зачета – тестирование, письменный опрос по теоретическим вопросам и выполнение практических заданий.

Описание показателей и критериев оценивания компетенций

Сформированность каждой компетенции (или ее части) в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

- пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;
- продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;
- эталонный уровень характеризуется максимально возможной выраженностью компетенций и является важным качественным ориентиром для самосовершенствования.

Уровень освоения компетенций, формируемых дисциплиной:

Описание критериев и шкалы оценивания тестирования:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 85 до 100%
2 балла (продвинутый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 70 до 84%
1 балл (пороговый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 50 до 69%
0 баллов	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 0 до 49%

Описание критериев и шкалы оценивания теоретического вопроса:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	выставляется студенту, который дал полный ответ на вопрос, показал глубокие систематизированные знания, смог привести примеры, ответил на дополнительные вопросы преподавателя
2 балла (продвинутый уровень)	выставляется студенту, который дал полный ответ на вопрос, но на некоторые дополнительные вопросы преподавателя ответил только с помощью наводящих вопросов
1 балл	выставляется студенту, который дал неполный ответ на вопрос

<i>(пороговый уровень)</i>	в билете и смог ответить на дополнительные вопросы только с помощью преподавателя
0 баллов	выставляется студенту, который не смог ответить на вопрос

Описание критериев и шкалы оценивания практического задания:

Шкала оценивания	Критерий
3 балла <i>(эталонный уровень)</i>	Задача решена верно
2 балла <i>(продвинутый уровень)</i>	Задача решена верно, но имеются неточности в логике решения
1 балл <i>(пороговый уровень)</i>	Задача решена верно, с дополнительными наводящими вопросами преподавателя
0 баллов	Задача не решена

На промежуточную аттестацию (зачет) выносятся тест, ва теоретических вопроса и 2 задачи. Максимально студент может набрать 15 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную форму по системе «зачтено» и «не зачтено».

Оценки «зачтено» заслуживает обучающийся, продемонстрировавший полное знание материала изученной дисциплины, усвоивший основную литературу, рекомендованную рабочей программой дисциплины; показавший систематический характер знаний по дисциплине, ответивший на все вопросы билета или допустивший погрешности в ответах на вопросы, но обладающий необходимыми знаниями для их устранения под руководством преподавателя.

Дополнительным условием получения оценки «зачтено» могут стать успехи при выполнении лабораторных работ, систематическая активная работа на лабораторных работах.

Оценка «зачтено» выставляется студенту, набравшему 8 и более баллов при промежуточной аттестации.

Оценки «не зачтено» заслуживает обучающийся, продемонстрировавший серьезные пробелы в знаниях основного материала изученной дисциплины, не ответивший на все вопросы билета и дополнительные вопросы. Как правило, оценка «не зачтено» ставится обучающимся, которые не могут продолжить обучение по образовательной программе без дополнительных занятий по соответствующей дисциплине (формирования и развития компетенций, закрепленных за данной дисциплиной).

Оценка «не зачтено» выставляется студенту, набравшему менее 8 баллов при промежуточной аттестации.

Паспорт оценочных материалов по дисциплине

Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или её части)	Вид, метод, форма оценочного мероприятия
Тема 1. Общая характеристика параллельных вычислительных систем	ПК-1, ПК-6	Зачет
Тема 2. Параллельные вычисления	ПК-1, ПК-6	Зачет
Тема 3. Технология параллельного программирования OpenMP	ПК-1, ПК-6	Зачет
Тема 4. Технология параллельного программирования для графических процессоров CUDA	ПК-1, ПК-6	Зачет
Тема 5. Перспективные направления в параллельном программировании	ПК-1, ПК-6	Зачет

ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ

Промежуточная аттестация в форме зачета

Код компетенции	Результаты освоения ОПОП Содержание компетенций
ПК-1	Способен проектировать программное обеспечение с использованием современных инструментальных средств
ПК-6	Способен использовать современные методы разработки и реализации конкретных алгоритмов математических моделей на базе языков программирования и пакетов прикладных программ моделирования

Типовые тестовые вопросы:

Вопрос 1

Вычислительная система называется многопроцессорной, если:

- а) она содержит несколько процессоров, работающих с общей оперативной памятью, и управляется разными операционными системами;
- б) содержит несколько процессоров;
- +в) она содержит несколько процессоров, работающих с общей оперативной памятью, и управляется одной общей операционной системой;
- г) она содержит несколько процессоров, работающих с отдельной оперативной памятью, и управляется одной общей операционной системой.

Вопрос 2

Вычислительная система, которая содержит несколько ЭВМ, каждая из которых имеет свою ОП и работает под управлением своей операционной системы:

- а) многопроцессорной;
- б) однородной;
- в) специализированной;
- +г) многомашинной.

Вопрос 3

Представителями какого класса являются классические фон-неймановские ВМ

- +а) SISD;
- б) MISD;
- в) SIMD;
- г) MIMD.

Вопрос 4

Вычислительная машина какой архитектуры позволяют выполнять одну арифметическую операцию сразу над многими данными — элементами вектора.

- а) SISD;
- б) MISD;
- +в) SIMD;
- г) MIMD.

Вопрос 5

В сильно связанных системах класса MIMD:

- а) вся память распределена между процессорными элементами;
- +б) мультипроцессоры симметричные.
- в) существуют кластерные системы;
- +г) общая память данных и команд;

Вопрос 6

Что такое многоядерный процессор?

- а) это процессор, содержащий два и более вычислительных ядра на нескольких процессорных кристаллах;
- б) это центральный процессор, содержащий четыре и более вычислительных ядра на одном процессорном кристалле или в одном корпусе;
- в) это центральный процессор, содержащий от двух до восьми вычислительных ядра на одном процессорном кристалле или в одном корпусе.
- +г) это центральный процессор, содержащий два и более вычислительных ядра на одном процессорном кристалле или в одном корпусе.

Вопрос 7

В вычислительных системах с общей памятью...

- а) значение, записанное в память одним из процессоров недоступно для другого процессора;
- б) память физически распределена по различным частям системы, но логически разделяема;
- +в) значение, записанное в память одним из процессоров, напрямую доступно для другого процессора;
- г) каждый процессор имеет свою локальную память с локальным адресным пространством.

Вопрос 8

Под симметричным доступом к памяти понимается следующее (допускается множественный выбор):

- +а) равные права всех процессоров на доступ к памяти;
- +б) одна и та же адресация для всех элементов памяти;
- в) разное время доступа всех процессоров системы;
- +г) равное время доступа всех процессоров системы;

Вопрос 9

Какой уровень параллелизма осуществляется путем одновременного выполнения нескольких независимых заданий на разных процессорах, которые, практически, не взаимодействуя друг с другом. Этот уровень реализуется на многопроцессорных и многомашинных ВС.

- а) уровень потоков;
- +б) уровень заданий;
- в) микроуровень;
- г) уровень команд

Вопрос 10

Гранулярность это...

- а) объем вычислений, выполненный в параллельной задаче;
- б) мера отношения объема коммуникаций к объему вычислений, выполненных в параллельной задаче;
- в) объем коммуникаций;
- +г) мера отношения объема вычислений, выполненных в параллельной задаче, к объему коммуникаций.

Вопрос 11

Какой уровень параллелизма обеспечивается операционной системой.

- а) среднезернистый параллелизм;
- +б) крупнозернистый параллелизм;
- в) мелкозернистый параллелизм;
- г) среднезернистый и крупнозернистый параллелизм.

Вопрос 12

Отношение времени, затрачиваемого на проведение вычислений на однопроцессорной ВС, ко времени решения той же задачи на параллельной n-процессорной системе):

- а) ускорение; +
- б) индекс параллелизма;
- в) эффективность;
- г) утилизация.

Вопрос 13

Избыточность это ...

- а) целесообразность наращивания числа процессоров через ту долю ускорения, достигнутого за счет параллельных вычислений, которая приходится на один процессор;

- б) средняя скорость параллельных вычислений, выраженная через количество выполненных операций;
- +в) отношение объема параллельных вычислений к объему эквивалентных последовательных вычислений;
- г) показатель эффективной скорости вычислений.

Вопрос 14

Количество операций, выполняемых на данной вычислительной системе в единицу времени это:

- а) быстродействие вычислительной системы;
- б) масштабируемость вычислительной системы;
- +в) производительность вычислительной системы;
- г) реконфигурируемость вычислительной системы.

Вопрос 15

Назовите характеристики симметричных мультипроцессорных систем.

- а) не все процессоры способны выполнять одинаковые функции;
- +б) 2+ процессоров сопоставимой производительности;
- +в) любой из процессоров может обслуживать внешние прерывания;
- +г) разное время доступа к памяти любого процессора.

Вопрос 16

Назовите преимущества SMP-систем.

- +а) отказ одного из компонентов не ведет к отказу системы;
- +б) производительность;
- в) число процессоров в системе изменить нельзя;
- г) все перечисленное.

Вопрос 17

Назовите основные причины появления систем с массовой параллельной обработкой.

- +а) необходимость построения ВС с гигантской производительностью;
- б) стремление раздвинуть границы производства ВС в большом диапазоне производительности и стоимости;
- в) нет верного ответа.

Вопрос 18

По каким особенностям вычислительную систему причисляют к классу MPP?

- а) система плохо масштабируется;
- +б) вычислительный узел обладает всеми средствами для независимого функционирования;
- в) один узел содержит сетевой адаптер, используемый для объединения остальных узлов.
- +г) работа системы координируется главной ВМ (хост-компьютером).

Вопрос 19

Что общего у кластерной вычислительной системы с MPP?

- а) узлы кластерной системы — это не специализированные устройства, приспособленные под использование в ВС;
- +б) тот же принцип распределенной памяти;
- +в) большой потенциал для масштабирования системы;
- г) в единую систему могут объединяться узлы разного типа.

Вопрос 20

Благодаря чему группа взаимосвязанных узлов становится кластером?

- а) благодаря межузловым связям;
- +б) благодаря связующему слою;
- в) благодаря сетевым адаптерам;
- г) благодаря подключению к шине.

Вопрос 21

Неоднородный вычислительный кластер – это:

- а) кластер, узлы которого имеют различные процессоры только по производительности;
- б) кластер, узлы которого имеют одинаковые по производительности и по архитектуре процессоры;
- в) кластер, узлы которого имеют различные процессоры только по архитектуре;
- г) кластер, узлы которого имеют различные по производительности и/или по архитектуре процессоры.

Вопрос 22

С чем в современных условиях связан переход к активному распараллеливанию вычислений? Допускается множественный выбор.

- +а) с увеличением выделяемой теплоты (рост температуры);
- б) со сложностью изготовления;
- +в) с нецелесообразностью наращивать потребление;
- +г) с нецелесообразностью увеличивать частоту.

Вопрос 23

Может ли программа быть полностью распараллеленной?

- а) программа всегда не может быть распараллелена;
- +б) программа может быть распараллелена, но не полностью;
- в) программа всегда может быть распараллелена.

Вопрос 24

Что такое OpenMP?

- а) технология организации параллельных вычислений в системах с распределённой памятью;
- +б) технология организации параллельных вычислений в системах с общей памятью;
- в) технология организации последовательных вычислений на графическом процессоре;
- г) технология организации последовательных вычислений на мобильном процессоре.

Вопрос 25

Какие преимущества даёт программисту использование OpenMP?

- а) предлагает встроенную реализацию популярных паттернов;
- б) предлагает статический анализ кода;
- +в) предлагает модель инкрементального программирования;
- г) предлагает средства рефакторинга.

Вопрос 26

Каким образом осуществляется программирование с помощью OpenMP?

- а) с помощью новых встроенных функций;
- +б) с помощью новых встроенных функций и директив;
- в) с помощью новых встроенных функций, директив и типов.

Вопрос 27

Что такое поток? В чём заключается разница между master и slave потоками?

- а) поток – это экземпляр программы, все потоки равны;
- б) поток – это экземпляр программы, master поток управляет slave потоками;
- в) поток – это обособленная копия набора команд вместе со своим стеком, все потоки равны;
- +г) поток – это обособленная копия набора команд вместе со своим стеком, master поток управляет slave потоками.

Вопрос 28

Директива parallel в технологии OpenMP:

- а) запускает цикл на нескольких параллельных потоках;
- +б) запускает код в нескольких параллельных потоках;
- в) обозначает критическую секцию;
- г) обозначает барьер синхронизации параллельных потоков.

Вопрос 29

Какое количество одновременно работающих потоков в OpenMP устанавливается по умолчанию; можно установить максимальным?

- а) оба равны количеству процессоров;
- б) равное количеству процессоров, не ограничено;
- в) оба равны количеству вычислительных ядер;
- +г) равное количеству вычислительных ядер, не ограничено;

Вопрос 30

Выберите все объекты OpenMP, которые позволяют управлять настройками количества потоков.

- а) опция threadprivate;
- +б) опция num_threads;
- +в) функция omp_set_num_thread;
- г) директива single.

Вопрос 31

Чем модификатор firstprivate отличается от модификатора lastprivate?

а) `firstprivate` задаёт значение переменной только в потоке-мастере, а `lastprivate` – только в подчинённом потоке;

+б) `firstprivate` задаёт значение переменной при входе в поток, а `lastprivate` – на выходе из потока;

в) `firstprivate` обнуляет значение переменной только в потоке-мастере, а `lastprivate` – только в подчинённом потоке;

г) ничем не отличаются;

Вопрос 32

Выберите верные утверждения об опции `reduction`.

а) опция распараллеливает циклы;

б) опция распараллеливает условные операторы;

+в) опция выполняет указанную операцию на выходе из параллельной области;

г) опция выполняет очистку переменных на выходе из параллельной области.

Вопрос 33

Какие функции выполняет критическая секция?

+а) впускает не более одного потока одновременно;

б) впускает только мастер поток;

в) включает режим высокоточных вычислений;

г) является параллельным аналогом `try-catch`.

Вопрос 34

Можно ли в OpenMP регулировать необходимость распараллеливания программы по условию?

а) нет, нельзя;

б) нет, нельзя, реализовать вручную невозможно;

в) да, можно, но нужно реализовать вручную;

+г) да, можно, существуют встроенные средства.

Вопрос 35

Какие существуют способы настройки OpenMP вне директив?

а) иных способов не существует;

б) иных встроенных способов не существует, но можно реализовать самому;

+в) существуют встроенные функции;

г) существуют встроенные классы.

Вопрос 36

Укажите способы, которыми участок кода параллельной программы может быть выполнен только одним потоком.

+а) директива `single`;

+б) директива `if`;

+в) директива `master`;

г) функция `omp_set_num_threads`.

Вопрос 37

Можно ли распараллелить цикл? Какой цикл? Какими способами? Выберите верные утверждения.

- а) циклы распараллеливать нельзя;
- б) можно распараллелить любой цикл встроенными средствами;
- в) можно распараллелить любой цикл собственной реализацией;
- +г) можно распараллелить любой цикл собственной реализацией, если итерации цикла не зависят друг от друга;
- +д) можно распараллелить цикл for встроенными средствами.

Вопрос 38

Какие существуют режимы планирования распараллеливания итераций цикла?

- а) deferred – отложенный режим (зависит от пришедших данных);
- +б) static – статическое распределение по количеству итераций;
- в) lazy – ленивое распределение (равномерно-случайное);
- +г) guided – управляемое распределение (в зависимости от нагрузки).

Вопрос 39

Упорядочены ли распараллеленные итерации цикла? Можно ли управлять последовательностью их выполнения?

- а) да, да;
- б) да, нет;
- +в) нет, да;
- г) нет, нет.

Вопрос 40

Как работает барьер?

- а) впускает только первый поток;
- б) впускает только мастер поток;
- в) впускает первый поток, ждёт завершения, впускает следующий;
- +г) ждёт, пока все потоки подойдут к нему, после впускает всех одновременно.

Вопрос 41

Каким образом осуществляется синхронизация памяти в OpenMP?

- а) синхронизация памяти не предусмотрена;
- б) память синхронизируется автоматически;
- в) существует встроенная функция;
- +г) существует специальная директива.

Вопрос 42

Какое поведение определяет опция `copyin`?

- а) копирует значения всех указанных переменных в параллельную область;
- +б) копирует значения всех `threadprivate` переменных в параллельную область со значениями из нити мастера;
- в) создаёт внутри параллельных областей `threadprivate` переменные со значениями по умолчанию;
- г) возвращает из параллельной области значения указанных переменных.

Вопрос 43

В чём сложность организации параллельных вычислений? Выберите все правильные варианты.

- +а) циклы имеют зависимости между итерациями;
- б) невозможно создать локальные копии переменных;
- +в) потоки пытаются изменять одни и те же области памяти;
- г) высокая стоимость создания потока. +

Вопрос 44

Всегда ли распараллеливание программ даёт выигрыш в скорости вычислений?

- а) да, всегда даёт выигрыш;
- б) даёт выигрыш только для небольших задач;
- +в) даёт выигрыш только для больших задач;
- г) не даёт выигрыш.

Вопрос 45

Укажите основные отличия архитектур CPU и GPU.

- +а) CPU заточены под скорость выполнения, GPU – под параллельное выполнение;
- б) ядра CPU проще и дешевле, ядра GPU – дороже и сложнее;
- +в) ядра CPU сложнее и больше в размерах, ядра GPU – простые, маленькие, заточены под параллельное выполнение;
- г) только размеры ядер.

Вопрос 46

Какие основные особенности архитектуры современных GPU позволили им значительно увеличить вычислительную мощность?

- +а) малые размеры ядер;
- +б) простота исполнения ядер;
- в) высокая тактовая частота;
- г) применение конвейеризации.

Вопрос 47

Выберите верные утверждения о программной архитектуре CUDA.

- +а) CUDA Host API – это функции, выполняющиеся только на CPU;
- б) программа компилируется и выполняется полностью на GPU;
- в) допускается одновременное использование Driver API и Runtime API;
- +г) Runtime API – это высокоуровневая реализация с помощью Driver API;

Вопрос 48

Что такое устройство в терминах CUDA?

- а) ядро GPU;
- +б) графический ускоритель;
- в) устройство, управляющее процессом выполнения и компиляции;

Вопрос 49

Что такое хост?

- а) ядро GPU;

- б) графический ускоритель;
- +в) устройство, управляющее процессом выполнения и компиляции;

Вопрос 50

Что такое ядро (kernel)?

- а) ядро GPU;
- б) графический ускоритель;
- в) CPU;
- +г) программа на CUDA.

Вопрос 51

Как осуществляется компиляция и выполнение программы, написанной с помощью технологии CUDA?

- а) программа компилируется и выполняется полностью на GPU;
- +б) программа компилируется на CPU, выполняется на GPU;
- в) часть программы компилируется на CPU, часть – GPU, выполнение всегда на GPU.

Вопрос 52

Что такое в терминах технологии CUDA блок, тред?

- +а) тред – единица выполнения программы, блок – группа тредов с общей памятью;
- б) тред – исполнительный блок, блок – группа исполняющих блоков;
- в) тред – единица выполнения программы, блок – группа тредов с неразделяемой памятью.

Вопрос 53

На скольких одновременно гридах может выполняться ядро?

- а) 1; +
- б) 4;
- в) все сразу;
- г) устанавливается вручную.

Вопрос 54

Что будет, если для выполнения ядра не будет достаточного количества вычислительных ресурсов?

- а) произойдет ошибка времени выполнения;
- б) ядра, на которых не хватило ресурсов, будут проигнорированы;
- +в) ядра, на которых не хватило ресурсов, будут поставлены в очередь.

Вопрос 55

Какие существуют ограничения на размеры гридов, блоков?

- а) ограничений нет, это логические понятия;
- б) ограничения заданные статично и не изменяются в версиях архитектуры;
- +в) каждое поколение GPU имеет собственные ограничения.

Вопрос 56

Почему ограничения на размер грида/блока важно учитывать при разработке программ?

- а) потому что превышение установленного уровня может вызвать отказ устройства;
- б) потому что превышение установленного уровня на порядок снижает производительность GPU;
- +в) потому что превышение установленного уровня может приводить к небольшому снижению производительности и ошибкам устройства.

Вопрос 57

Выберите все основные объекты программного интерфейса CUDA.

- +а) встроенные типы;
- +б) встроенные функции;
- +в) встроенные переменные;
- г) встроенные директивы;
- +д) атрибуты переменных;
- +е) атрибуты функций;
- ж) атрибуты директив.

Вопрос 58

Выберите верные утверждения касательно модификаторов `__global__`, `__host__`, `__device__`.

- +а) `__global__` - модификатор функции, делающий её ядром;
- б) `__global__` - модификатор переменной, делающий её видимой на устройстве и хосте;
- +в) `__device__` может быть как модификатором функции, там и модификатором переменной;
- г) функция с модификатором `__host__` компилируется на GPU, а выполняется на CPU;
- +д) функция с модификатором `__device__` выполняется на GPU и может вызываться там же.

Вопрос 59

Выберите участок кода, который определяет и запускает ядро.

- а) `void Kernel(...) { } Kernel(param1, ..., paramN);`
- б) `__device__ void Kernel(...) { } Kernel(param1, ..., paramN);`
- +в) `__global__ void Kernel(...) { } Kernel<<<k, l>>>(param1, ..., paramN);`
- г) `__device__ int Kernel(...) { } Kernel<<<k, l>>>(param1, ..., paramN);`

Вопрос 60

Какие параметры используются при запуске ядра?

- а) `grid`, `block` – количество блоков в гриде и потоков в блоке;
- +б) `grid`, `block`, `mem`, `stream` - количество блоков в гриде, потоков в блоке, объём дополнительной `shared`-памяти, поток вызова;
- в) `mem`, `stream` - объём дополнительной `shared`-памяти и поток вызова;
- г) `dev`, `mem`, `block` – номер устройства, количество выделяемой памяти на блок, количество потоков в блоке.

Вопрос 61

Выберите необходимые шаги, которые должны быть выполнены любой программой, выполняющей расчёты на GPU с помощью CUDA.

- +а) выделение памяти на устройстве;
- б) выделение памяти на хосте;
- +в) копирование данных на устройство;
- +г) копирование данных на хост;
- +д) вызов ядра;
- +е) проверка наличия ошибок выполнения;
- +ж) освобождение памяти на устройстве;
- з) освобождение памяти на хосте.

Вопрос 62

Какой из перечисленных способов распараллеливания наиболее оптимальный в большинстве случаев?

- а) по гридам
- б) по блокам;
- в) по тредам;
- +г) по блокам и тредам;
- д) всё вместе.

Вопрос 63

Какие существуют модификаторы переменных в CUDA?

- +а) `__device__` - переменная находится на устройстве;
- б) `__host__` - переменная находится на хосте;
- +в) `__const__` - переменная является константой;
- +г) `__shared__` - переменная разделена между тредами в блоке;
- д) `__local__` - переменная является локальной у треда.

Вопрос 64

Какая операция не является операцией по работе с глобальной памятью CUDA.

- а) `cudaMalloc`;
- б) `cudaFree`;
- в) `cudaMemcpy`;
- +г) `cudaMemclr`.

Вопрос 65

В чём преимущества shared памяти?

- а) дешевизна;
- б) объём;
- +в) скорость.

Вопрос 66

Каким образом можно спроектировать на плоскость текущее положение треда в блоке?

- а) $\text{int threadId} = \text{blockIdx.x} + \text{threadIdx.x}$;
- +б) $\text{int threadId} = \text{blockIdx.x} * \text{blockDim.x} + \text{threadIdx.x}$;
- в) $\text{int threadId} = \text{blockIdx.x} * \text{blockDim.y} + \text{threadIdx.x} * \text{threadDim.y}$;
- г) $\text{int threadId} = \text{blockDim.x} + \text{threadIdx.y}$.

Типовые теоретические вопросы:

- 1) Чем различаются многомашинные и многопроцессорные системы
- 2) В чем заключается классификация архитектур ВС Флинна
- 3) Какие ВС относятся к классам SISD и SIMD?
- 4) Какие ВС относятся к классам MISD и MIMD?
- 5) Тенденции развития современных ВС
- 1) Уровни параллелизма. Чем обеспечивается каждый уровень?
- 2) Что такое степень гранулярности. Чем она обеспечивается?
- 3) Метрики параллелизма.
- 4) Закон Амдала
- 5) Закон Густавсона
- 6) Моделирование и анализ параллельных вычислений.
- 7) Симметричные мультипроцессорные системы (SMP)
- 8) Системы с массовой параллельной обработкой (MPP)
- 9) Кластеры вычислительных машин
- 10) Параллельные и последовательные области в OpenMP.
 - 11) Модель данных в OpenMP.
 - 12) Распределение работы в OpenMP.
 - 13) Синхронизация в OpenMP.
 - 14) Параллельные алгоритмы сортировки данных.
 - 15) Параллельные методы матричного умножения.
 - 16) Гибридная модель вычислений.
 - 17) Архитектура графического процессора (GPU).
 - 18) Обзор средств программирования для GPU.
 - 19) Программная модель CUDA.
 - 20) Понятие потока, блока, сети блоков. Функция-ядро.
 - 21) Иерархия памяти. Регистры и локальная память.
 - 22) Глобальная память. CUDA-потоки. Разделяемая память.
 - 23) Константная память. Текстурированная память.
 - 24) Процессоры Cell.
 - 25) Грид-вычисления.
 - 26) Транзакционная память.
 - 27) Модель MapReduce.
 - 28) Развивающиеся языки: Chapel, Fortress, X10.

Типовые задачи:

Выполняются 2 задания одно с использованием технологии OpenMP, второе – с использованием технологии CUDA.

Технология OpenMP

- 1) В параллельной области программы вывести номер исполняющейся в данный момент нити.
- 2) С помощью опции reduction произвести суммирование элементов вектора, длина которого равна количеству нитей, создаваемых при старте параллельной области (значение по умолчанию либо устанавливаемое вами).

- 3) Создать вектор из четырёх элементов. Заполнить его случайными значениями. С помощью директивы `firstprivate` передать его в параллельную область. Вывести на экран элемент по индексу, равному номеру исполняющейся нити. Заменить значение этого элемента номером исполняющейся нити. С помощью `lastprivate` передать изменённый вектор в последовательную область. Показать на экране элементы изменённого вектора.
- 4) Произвести параллельное суммирование значений строк матрицы. Полученные значения записать в вектор.
- 5) Произвести параллельное умножение элементов вектора на коэффициент, равный номеру исполняющейся нити, увеличенному на единицу.
- 6) Создать вектор длиной более ста элементов и кратной количеству параллельных нитей, заполнить его случайными числами не больше тысячи по модулю. Произвести параллельную редукцию.
- 7) Параллельно заполнить вектор из тысячи элементов числами от 1 до 1000. В решении применить цикл `for` и директиву для его распараллеливания.

Технология CUDA

- 1) Создать матрицу размера $[10 \times 10]$. В первый элемент каждой строки записать номер исполняющего блока, остальные элементы равномерно заполнить числами от 0 до 8.
- 2) Произвести параллельное суммирование значений строк матрицы. Полученные значения записать в вектор.
- 3) Произвести параллельное умножение элементов вектора на коэффициент, равный номеру треда, увеличенному на единицу.
- 4) Создать вектор длиной более ста элементов и кратной количеству блоков, заполнить его случайными числами не больше тысячи по модулю. Произвести параллельную редукцию.
- 5) Создать вектор из тысячи целых чисел, заполнить его значениями $[1; 10]$. Параллельно посчитать гистограмму распределений чисел. В решении использовать буферный вектор из десяти элементов.
- 6) Распараллелить любой простой расчёт в цикле `while` при условии, что все итерации цикла независимы друг от друга.
- 7) Произвести параллельное суммирование элементов матрицы $[100 \times 100]$.

Составил
к.т.н., доц. кафедры ЭВМ

Елесина С.И.

Зав. кафедрой ЭВМ
д.т.н., проф.

Костров Б.В.