


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»

Директор института  
магистратуры и аспирантуры  
 О.А. Бодров  
«\_\_» \_\_\_\_\_ 2020 г.

«УТВЕРЖДАЮ»

  
Проректор по РОИиМД  
А.В. Корячко  
«\_\_» \_\_\_\_\_ 2020 г.

Заведующий кафедрой  
 Б.В. Костров  
«\_\_» \_\_\_\_\_ 2020 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**Б1.О.04 «Вычислительные системы»**

Направление подготовки  
09.04.01 «Информатика и вычислительная техника»

Направленность подготовки  
«Вычислительные машины, системы, комплексы и сети»

Уровень подготовки  
магистратура

Квалификация выпускника – магистр

Форма обучения – очная

Рязань 2020 г.

## ЛИСТ СОГЛАСОВАНИЙ

Рабочая программа составлена с учетом требований федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.04.01 «Информатика и вычислительная техника» (уровень магистратуры), утвержденным приказом Минобрнауки России от 19.09.2017 г. № 918.

Разработчик  
доцент кафедры ЭВМ



С.И. Елесина

Программа рассмотрена и одобрена на заседании кафедры ЭВМ  
«11» 06 2020 г., протокол № 10

Заведующий кафедрой ЭВМ



Б.В. Костров

## 1. Цель и задачи освоения дисциплины

*Целью дисциплины* «Вычислительные системы» является изучение современных высокопроизводительных вычислительных систем и их компонентов, а также получение знаний и навыков в использовании этих систем.

*Основными задачами дисциплины* являются:

- познакомить обучающихся с современными вычислительными системами, их архитектурами, компонентами и классами решаемых ими задач;
- дать представление об основных структурных, функциональных и алгоритмических решениях направленных на повышение производительности вычислений;
- дать начальное представление о программировании таких систем;
- дать возможность на примере программирования многопроцессорных систем с общей памятью и графических процессоров, получить практический опыт эксплуатации вычислительных систем повышенной производительности.

## 2. Место дисциплины в структуре образовательной программы

Дисциплина Б1.О.04 «Вычислительные системы» является обязательной, относится к базовой части блока № 1 дисциплин основных профессиональных образовательных программ академической магистратуры ОПОП – «Вычислительные машины, системы, комплексы и сети» по направлению подготовки 09.04.01 «Информатика и вычислительная техника» ФГБОУ ВО «РГРТУ».

Дисциплина изучается по очной форме обучения на 1 курсе в 1 семестре.

Для изучения дисциплины обучаемый должен

*знать:*

- архитектуру современных компьютеров;
- архитектурные особенности современных процессоров;
- принципы организации и функционирования памяти компьютера;
- основы сетевых технологий;
- основы организации операционных систем;
- методы проектирования программных средств;

*уметь:*

- проектировать базовые программные компоненты;

*владеть:*

- базовыми навыками программирования на языке C/C++.

*Взаимосвязь с другими дисциплинами.* Дисциплина «Вычислительные системы» логически связана со следующими дисциплинами: «Технологии разработки программного обеспечения», «Интернет-технологии».

Знания, полученные в результате освоения дисциплины будут полезны обучающемуся при изучении дисциплин: «Методы и алгоритмы обработки изображений», «Системы технического зрения», «Программно-аппаратные средства высокоскоростной обработки информации».

Материал дисциплины «Вычислительные системы» формирует методологические и

организационные основы для выполнения обучающимися практик и выпускной квалификационной работы.

### 3. Компетенции обучающегося, формируемые в результате освоения дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций в соответствии с ФГОС ВО.

#### Общепрофессиональные компетенции выпускников и индикаторы их достижения

Код компетенции	Результаты освоения ОПОП Содержание компетенций	Перечень планируемых результатов обучения по дисциплине
<b>ОПК-2</b>	Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач;	<p>ОПК-2.1. Знать: современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы для решения профессиональных задач</p> <p>ОПК-2.2. Уметь: обосновывать выбор современных информационно-коммуникационных и интеллектуальных технологий, разрабатывать оригинальные программные средства для решения профессиональных задач</p> <p>ОПК-2.3. Владеть: методами разработки оригинальных программных средств, в том числе с использованием современных информационно-коммуникационных и интеллектуальных технологий, для решения профессиональных задач</p>
<b>ОПК-6</b>	Способен разрабатывать компоненты программно-аппаратных комплексов обработки информации и автоматизированного проектирования;	<p>ОПК-6.1. Знать: аппаратные средства и платформы инфраструктуры информационных технологий, виды, назначение, архитектуру, методы разработки и администрирования программно-аппаратных комплексов объекта профессиональной деятельности</p> <p>ОПК-6.2. Уметь: анализировать техническое задание, разрабатывать и оптимизировать программный код для решения задач обработки информации и автоматизированного проектирования</p> <p>ОПК-6.3. Владеть: методами составления технической документации по использованию и настройке компонентов программно-аппаратного комплекса</p>

#### 4. Структура и содержание дисциплины

**4.1 Объем дисциплины в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся**

Общая трудоемкость (объем) дисциплины составляет 5 зачетных единиц (ЗЕ), 180 часа.

Вид учебной работы	Всего часов	1 семестр
Общая трудоемкость дисциплины	180	180
1. Контактная работа обучающихся с преподавателем (всего), в том числе:	58,65	58,65
лекции	24	24
лабораторные работы	-	-
практические занятия	32	32
консультации	2	2
контактная внеаудиторная работа	-	-
иная контактная работа	0,65	0,65
2. Иные формы работы	-	-
3. Самостоятельная работа	61,3	61,3
4. Контроль	44,35	44,35
5. КП	15,7	15,7
Вид промежуточной аттестации обучающегося	экзамен, КП	экзамен, КП

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

Тема	Общая трудоемкость, всего часов	Контактная работа обучающихся с преподавателем			Самостоятельная работа
		Всего	Лекции	Практические занятия	
Тема 1. Тенденции развития современных вычислительных систем. Технология OpenMP	24	12	4	8	12
Тема 2. Вычислительные системы. Основные понятия и классификация	13	1	1	0	12
Тема 3. Параллельные вычисления	13	1	1	0	12
Тема 4. Вычислительные системы класса SISD. Внутрипроцессорный параллелизм	13	1	1	0	12
Тема 5. Организация памяти вычислительных систем	16	3	3	0	13
Тема 6. Вычислительные системы класса SIMD	18	4	4	0	14
Тема 7. Гетерогенные вычисления	46	21	5	16	25
Тема 8. Вычислительные системы класса MIMD	19	5	5	0	14
Курсовое проектирование	18	0	0	0	18
<b>Всего:</b>	<b>180</b>	<b>48</b>	<b>24</b>	<b>24</b>	<b>132</b>

### **4.3. Содержание дисциплины, структурированное по темам с указанием отведенного на них количества академических часов и видов учебных занятий**

**Тема 1.** Тенденции развития современных вычислительных систем. Технология OpenMP.

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

Технология OpenMP. Основные понятия. Классы переменных. Последовательные и параллельные области. Конструкции распределения работы. Синхронизация в OpenMP.

#### **Тема 2. Вычислительные системы. Основные понятия и классификация.**

Понятие системы. Вычислительные системы: определение и преимущества ВС. Многомашинные и многопроцессорные ВС. Классификация ВС. Классификации архитектур вычислительных систем: классификация Флинна, классификация Ванга и Бриггса и др.

#### **Тема 3. Параллельные вычисления**

Уровни параллелизма. Степень гранулярности. Метрики параллельных вычислений. Основные метрики параллелизма: ускорение, эффективность, избыточность и т.д. Закономерности параллельных вычислений. Закон Амдала. Закон Густафсона. Закон Сана-Ная.

**Тема 4. Вычислительные системы класса SISD. Внутрипроцессорный параллелизм**

Общая характеристика систем класса SISD. Конвейеризация вычислений. Конвейер команд. Конфликты в конвейере команд. Суперконвейерные процессоры. Суперскалярность и внеочередное выполнение команд. Суперскалярные процессоры. Архитектуры VLIW и EPIC. Особенности микропроцессоров семейства Itanium (IA-64). Гиперпоточковая технология.

#### **Тема 5. Организация памяти вычислительных систем**

Модели архитектур памяти ВС. Физически разделяемая память UMA с шинной организацией и коммутацией. Физически разделенная разделяемая память: NUMA (ccNUMA, nccNUMA); COMA; DSM. Распределенная память NORMA. Мультипроцессорная когерентность кэш-памяти. Проблема когерентности. Программные и аппаратные средства решения проблемы когерентности. Возможности для поддержания когерентности. Разделяемая кэш-память. Некэшируемые данные. Широковещательная запись. Протоколы наблюдения. Протокол сквозной записи. Протокол обратной записи. Протокол однократной записи. Протокол MESI. Протокол на основе справочника. Полный справочник. Ограниченный справочник. Сцепленный справочник.

## **Тема 6. Вычислительные системы класса SIMD**

Общая характеристика систем класса SIMD.

Векторные вычислительные системы. Понятие векторного процессора. Архитектуры векторной обработки. Структура векторного процессора. Структура векторной вычислительной системы.

Матричные вычислительные системы. Контроллер массива процессорных элементов. Массив процессорных элементов. Структура процессорного элемента.

Ассоциативные вычислительные системы. Ассоциативная память. Ассоциативные процессоры. Ассоциативные многопроцессорные системы.

Вычислительные системы с систолической структурой. Классификация систолических структур. Топология систолических структур. Структура процессорного элемента.

Процессоры потоков данных.

SIMD-расширение в SISD-системе. Технологии MMX, SSE, AVX.

## **Тема 7. Гетерогенные вычисления**

Вычисления общего назначения на графических процессорах (GPU). Архитектура графического процессора. Гибридная модель вычислений. Средства программирования для GPU.

Архитектура параллельных вычислений CUDA. Программная модель CUDA. Понятие потока, блока, сети блоков. Функция-ядро. Иерархия памяти. Регистры и локальная память. Глобальная память. CUDA-потоки. Разделяемая память. Константная память. Текстуриная память. .

OpenCL – открытый язык параллельных вычислений для гетерогенных ВС. Архитектура OpenCL: модель платформы; модель исполнения; модель памяти; модель программирования. Типы данных и функции OpenCL.

## **Тема 8. Вычислительные системы класса MIMD**

Общая характеристика систем класса MIMD. MIMD-системы с разделяемой памятью. Симметричные мультипроцессорные системы (SMP). Параллельные векторные системы. Вычислительные системы с неоднородным доступом к памяти. MIMD-системы с распределенной памятью. Системы с массовой параллельной обработкой (MPP). Кластерные вычислительные системы. Технология MPI. Кластеры больших SMP-систем. Суперкомпьютеры.

### Виды практических и самостоятельных работ

Тема	Вид работы	Наименование и содержание работы	Трудоемкость, часов
Тема 1. Тенденции развития современных вычислительных систем. Технология OpenMP	Самостоятельная работа	Изучение методических указаний, подготовка к практическим занятиям	4
		Изучение конспекта лекций	4
		Подготовка к экзамену и консультации	4
	Практическая работа	Технология OpenMP. Введение. Основные понятия. Классы переменных OpenMP.	2
		Параллельные и последовательные области. Директивы parallel, single, master	2
		Конструкции распределения работы. Исследование опции schedule.	2
		Синхронизация в OpenMP.	2
Тема 2. Вычислительные системы. Основные понятия и классификация	Самостоятельная работа	Изучение конспекта лекций и дополнительного теоретического материала	5
		Подготовка к экзамену и консультации	7
Тема 3. Параллельные вычисления	Самостоятельная работа	Изучение конспекта лекций и дополнительного теоретического материала	5
		Подготовка к экзамену и консультации	7
Тема 4. Вычислительные системы класса SISD. Внутри-процессорный параллелизм	Самостоятельная работа	Изучение конспекта лекций и дополнительного теоретического материала	6
		Подготовка к экзамену и консультации	6
Тема 5. Организация памяти вычислительных систем	Самостоятельная работа	Изучение конспекта лекций	5
		Подготовка к экзамену и консультации	8
Тема 6. Вычислительные системы класса SIMD	Самостоятельная работа	Изучение конспекта лекций	6
		Подготовка к экзамену и консультации	8
Тема 7. Гетерогенные вычисления	Практическая работа	Основные понятия и определения CUDA. Модель программирования. Расширения языка CUDA C. Типы памяти в CUDA. Работа с глобальной памятью. Распараллеливание с помощью блоков,	2



Тема	Вид работы	Наименование и содержание работы	Трудоемкость, часов
		нитей, блоков и нитей.	
		Работа с разделяемой памятью. Атомарные операции. Вычисление гистограммы на CPU. Решение задач.	2
		Реализация на CUDA базовой операции над массивами – REDUCE	2
		Разработка приложения с отображением входного и выходного изображений	2
		Генерация изображений с помощью CUDA	2
		Введение в OpenCL. Архитектура OpenCL. Создание и конфигурирование OpenCL проекта в Visual Studio.	2
		Язык OpenCL C: типы данных; квалификаторы; встроенные функции. Функции OpenCL	2
		Объекты памяти BUFFER и IMAGE. Работа с Image в OpenCL. Обработка изображения с использованием OpenCL	2
	Самостоятельная работа	Изучение конспекта лекций	6
		Изучение методических указаний, подготовка к практическим занятиям	8
Подготовка к экзамену и консультации		11	
Тема 8. Вычислительные системы класса MIMD	Самостоятельная работа	Изучение конспекта лекций	6
		Подготовка к экзамену и консультации	8
Курсовое проектирование	Самостоятельная работа	<p>Постановка задачи;</p> <p>Разработка последовательного алгоритма и его программная реализация на языке программирования C/C++;</p> <p>Разработка параллельных алгоритмов и их программная реализация с использованием технологий OpenMP и Cuda (или OpenCL).</p> <p>Исследование и сравнительный анализ последовательной и параллельных реализаций.</p> <p>Оформление пояснительной записки</p>	18

## Курсовое проектирование

Цель курсового проектирования: получить практический опыт программирования многопроцессорных систем с общей памятью и графических процессоров с использованием технологий OpenMP и CUDA (или OpenCL).

Тема курсового проекта: «Параллельные вычисления с использованием технологий OpenMP и CUDA (OpenCL)»

Основные направления для исследований:

1. Сортировка данных
2. Обработка графов
3. Параллельные методы матричного умножения
4. Параллельные методы решения систем линейных уравнений
5. Обработка изображений.
6. Корреляционное совмещение изображений.

Перечень индивидуальных заданий:

- 1) Параллельные алгоритмы на графах (алгоритмы Флойда и Прима)
- 2) Параллельные методы матричного умножения. Блочные алгоритмы умножения матриц.
- 3) Параллельные методы сортировки. Алгоритмы пузырьковой сортировки и Шелла.
- 4) Параллельные методы решения систем линейных уравнений. Метод Гаусса.
- 5) Реализация алгоритмов контрастирования изображений.
- 6) Параллельные методы решения систем линейных уравнений. Метод сопряженных градиентов.
- 7) Реализация алгоритмов выделения контуров изображений.
- 8) Реализация алгоритма кластеризации изображения (k-means)
- 9) Корреляционное совмещение изображений методом поэтапного сканирования.
- 10) Параллельные методы сортировки. Алгоритм быстрой сортировки (алгоритм Хоара).
- 11) Распознавание образов с использованием инвариантных моментов Ну.
- 12) Реализация алгоритмов фильтрации изображений.
- 13) Параллельные методы матричного умножения. Алгоритм умножения матриц, основанный на ленточном разделении данных.
- 14) Реализация параллельного алгоритма обнаружения границ Canny.
- 15) Корреляционное совмещение изображений с использованием пирамиды изображений
- 16) Реализация алгоритмов морфологических операций.
- 17) Быстрое преобразование Фурье
- 18) Корреляционное совмещение изображений методом мультистарт) (локальный метод – МДМ)

- 19) Корреляционное совмещение изображений методом мультистарт (локальный метод – Хука-Дживса)
- 20) Корреляционное совмещение изображений методом случайного поиска

## **5 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 «Вычислительные системы»).

## **6 Учебно-методическое обеспечение дисциплины**

### **6.1. 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>
- 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>
- 4) Архитектуры и топологии многопроцессорных вычислительных систем [Электронный ресурс] / А.В. Богданов [и др.]. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 135 с. — 5-9556-0018-3. — Режим доступа: <http://www.iprbookshop.ru/52189.html>
- 5) 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>
- 6) Ершова Н.Ю. Организация вычислительных систем [Электронный ресурс] / Н.Ю. Ершова, А.В. Соловьев. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 224 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/73687.html>
- 7) 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>
- 8) 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>

9) 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>

10) 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>

## **6.2. Дополнительная учебная литература:**

1) Бохан К.А. Вычислительные машины и системы : учеб. пособие / РГРТУ. - Рязань, 2013. - 95с. (23 экз.)

2) Елесина С.И. Обработка информации с использованием технологии OpenCL : метод. указ. / С. И. Елесина, Е. Р. Муратов ; РГРТУ. - Рязань, 2015. - 40с. (23 экз.)

3) 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>

4) 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>

5) 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>

6) Куликов И.М. Технологии разработки программного обеспечения для математического моделирования физических процессов. Часть 1. Использование суперкомпьютеров, оснащенных графическими ускорителями [Электронный ресурс] : учебное пособие / И.М. Куликов. — Электрон. текстовые данные. — Новосибирск: Новосибирский государственный технический университет, 2013. — 40 с. — 978-5-7782-2195-6. — Режим доступа: <http://www.iprbookshop.ru/45044.html>

7) 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>

## **7 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 РГПТУ без пароля, из сети интернет по паролю. – URL: <https://iprbookshop.ru/>.

## **8. 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 практических занятий**

Практические (семинарские) занятия составляют важную часть профессиональной подготовки студентов. Основная цель проведения практических (семинарских) занятий – формирование у студентов аналитического и творческого мышления путем приобретения практических навыков.

Методические указания к практическим занятиям по дисциплине наряду с рабочей программой и графиком учебного процесса относятся к методическим документам, определяющим уровень организации и качества образовательного процесса. Содержание практических занятий фиксируется в рабочей программе дисциплины в разделе 4.

Важнейшей составляющей любой формы практических занятий являются упражнения (задания). Основа в упражнении – пример, который разбирается с позиций теории, развитой в лекции. Как правило, основное внимание уделяется формированию конкретных умений, навыков, что и определяет содержание деятельности студентов – решение задач, графические работы, уточнение категорий и понятий науки, являющихся предпосылкой правильного мышления и речи.

Практические занятия выполняют следующие задачи:

- стимулируют регулярное изучение рекомендуемой литературы, а также внимательное отношение к лекционному курсу;
- закрепляют знания, полученные в процессе лекционного обучения и самостоятельной работы над литературой;
- расширяют объем профессионально значимых знаний, умений и навыков;
- позволяют проверить правильность ранее полученных знаний;
- прививают навыки самостоятельного мышления, устного выступления;
- способствуют свободному оперированию терминологией;
- представляют преподавателю возможность систематически контролировать уровень самостоятельной работы студентов.

При подготовке к практическим занятиям 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 практического занятия 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 отчет заносятся результаты выполнения каждого пункта задания (анализ задачи, найденные пути решения, поясняющие схемы, диаграммы, графики, таблицы, расчеты, ответы на вопросы пунктов задания, выводы по проделанной работе и т.д.). Примерный образец оформления отчета предоставляется студентам в виде раздаточных материалов или прилагается к рабочей программе дисциплины.

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

#### **Указания в рамках подготовки к промежуточной аттестации**

При подготовке к экзамену 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 экзамену нужно изучить теорию: определения всех понятий и подходы к оцениванию до состояния понимания материала и самостоятельно решить по несколько типовых 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 выполнения курсового проекта**

Целью курсового проектирования являются разработка и отладка приложений с использованием технологий OpenMP и Cuda или OpenCL.

Тема курсового проекта: «Параллельные вычисления с использованием технологий OpenMP и CUDA (OpenCL)».

Курсовой проект (КП) представляет собой самостоятельную работу по заданной теме. Работа предполагает:

- домашнюю внеаудиторную подготовку;
- консультации по КП;
- предъявление промежуточных результатов для проверки и контроля хода курсового проектирования;
- написание и оформление пояснительной записки;

– сдачу и защиту КП в сроки согласно учебному графику.

Основные этапы работы над курсовым проектом:

1) Постановка задачи: выбираются исходные данные, их типы и структуры, указываются ограничения на исходные данные; выполняется детализация функций программы; определяются формы представления результатов работы программы. Разработанную постановку задачи необходимо согласовать с преподавателем.

2) Разработка алгоритмов и программ: выделение модулей программы; разработка последовательного алгоритма и его программная реализация на языке программирования C/C++; разработка параллельных алгоритмов и их программная реализация с использованием технологий OpenMP и Cuda (или OpenCL).

3) Сравнительный анализ последовательной и параллельных реализаций: перечисление технических характеристик ВС; результаты вычислительных экспериментов; вычисление ускорения; выводы.

4) Разработка технической документации.

Оформление пояснительной записки к КП выполняется с помощью офисного пакета LibreOffice или в другом редакторе доступном студенту, схемы алгоритмов выполняются с помощью векторного графического редактора Microsoft Office Visio.

### **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, а также иметь самостоятельное значение – внеаудиторная самостоятельная работа обучающихся – при подготовке к лекциям, практическим занятиям, а также к экзамену.

Основными видами самостоятельной работы по дисциплине являются:

- самостоятельное изучение отдельных вопросов и тем дисциплины;
- выполнение практического задания;
- подготовка к защите практического задания, 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).

### **9. 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 дистанционному курсу, расположенного в системе дистанционного обучения ФГБОУ ВПО «РГРТУ»:

- 1) Вычислительные системы [Электронный ресурс]. URL: <http://cdo.rsreu.ru/course/view.php?id=1283> .

Система дистанционного обучения ФГБОУ ВПО «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) LibreOffice 4.4 – лицензия LGPLv3;
- 3) Microsoft Office Visio (лицензия Microsoft DreamSpark Membership ID 700102019).
- 4) Интегрированная среда разработки Visual Studio C++ 13 (лицензия DreamSpark Membership ID 700102019);



- 5) Программно-аппаратная архитектура параллельных вычислений CUDA (free software, EULA for the NVIDIA CUDA Toolkit, NVIDIA Software License Agreement + CUDA Supplement to Software License Agreement).

### **10 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 LibreOffice, Microsoft Office Visio, 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.О.04 «Вычислительные системы»**

Направление подготовки  
09.04.01 «Информатика и вычислительная техника»

Направленность подготовки  
«Вычислительные машины, системы, комплексы и сети»

Уровень подготовки  
магистратура

Квалификация выпускника – магистр

Форма обучения – очная, заочная

Рязань, 2020 г.

## 1 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 компетенций.

Контроль знаний обучающихся проводится в форме промежуточной аттестации.

Промежуточная аттестация проводится в форме экзамена и защиты курсового проекта. Форма проведения экзамена - тестирование, письменный опрос по теоретическим вопросам и выполнение практических заданий.

## 2 ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

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

- 1) пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;
- 2) продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;
- 3) 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:

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

#### Описание критериев и шкалы оценивания теоретического вопроса:

<i><b>Шкала оценивания</b></i>	<b>Критерий</b>
<i>3 балла (эталонный уровень)</i>	выставляется студенту, который дал полный ответ на вопрос, показал глубокие систематизированные знания, смог привести примеры, ответил на дополнительные вопросы преподавателя
<i>2 балла (продвинутый уровень)</i>	выставляется студенту, который дал полный ответ на вопрос, но на некоторые дополнительные вопросы преподавателя ответил только с помощью наводящих вопросов
<i>1 балл (пороговый уровень)</i>	выставляется студенту, который дал неполный ответ на вопрос в билете и смог ответить на дополнительные вопросы только с помощью преподавателя
<i>0 баллов</i>	выставляется студенту, который не смог ответить на вопрос

### Описание критериев и шкалы оценивания практического задания:

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

### Описание критериев и шкалы оценивания курсового проекта

<b>Шкала оценивания</b>	<b>Критерий</b>
Оценка «отлично» (эталонный уровень)	Курсовой проект (КП) выполнен в полном объеме, нет замечаний по разработке алгоритмов и программ, дана оценка полученных результатов, проект выполнен самостоятельно, пояснительная записка к КП оформлена аккуратно, соблюдались сроки сдачи и защиты КП, при защите КП студент ответил на все предложенные вопросы
Оценка «хорошо» (продвинутый уровень)	Курсовой проект выполнен в полном объеме, присутствуют незначительные замечания по разработке алгоритмов и программ, дана оценка полученных результатов, проект выполнен самостоятельно, пояснительная записка к КП оформлена аккуратно, соблюдались сроки сдачи и защиты курсового проекта, при защите КП студент ответил не на все предложенные вопросы (правильных ответов не менее 75%)
Оценка «удовлетворительно» (пороговый уровень)	Курсовой проект выполнен в полном объеме, присутствуют ошибки при разработке алгоритмов и программ, оценка полученных результатов не является полной, КП выполнен самостоятельно, по оформлению пояснительной записки к КП имеются замечания, частично соблюдались сроки сдачи и защиты КП, при защите курсового проекта студент ответил не на все предложенные вопросы (правильных ответов не менее 50%)
Оценка «неудовлетворительно»	Курсовой проект выполнен не в полном объеме, присутствуют грубые ошибки при разработке алгоритмов и программ, отсутствует оценка полученных результатов, КП выполнен не самостоятельно, по оформлению пояснительной записки к КП имеются замечания, не соблюдались сроки сдачи и защиты курсового проекта, при защите курсового проекта студент ответил не на все предложенные вопросы (правильных ответов менее 50%)

**На промежуточную аттестацию выносятся тест, два 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 «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно».

**Оценка «отлично»** выставляется студенту, который набрал в сумме 15 баллов (выполнил все задания на эталонном уровне). Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

**Оценка «хорошо»** выставляется студенту, который набрал в сумме от 10 до 14 баллов при условии выполнения всех заданий на уровне не ниже продвинутого. Обязательным

условием является выполнение всех предусмотренных в течение семестра практических заданий.

**Оценка «удовлетворительно»** выставляется студенту, который набрал в сумме от 5 до 9 баллов при условии выполнения всех заданий на уровне не ниже порогового. Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

**Оценка «неудовлетворительно»** выставляется студенту, который набрал в сумме менее 5 баллов или не выполнил все предусмотренные в течение семестра практические задания.

<b>Код компетенции</b>	<b>Результаты освоения ОПОП Содержание компетенций</b>
<b>ОПК-2</b>	Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач;
<b>ОПК-6</b>	Способен разрабатывать компоненты программно-аппаратных комплексов обработки информации и автоматизированного проектирования.

### 3 ПАСПОРТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ПО ДИСЦИПЛИНЕ

<i>Контролируемые разделы (темы) дисциплины</i>	<i>Код контролируемой компетенции (или её части)</i>	<b>Вид, метод, форма оценочного мероприятия</b>
Тема 1. Тенденции развития современных вычислительных систем. Технология OpenMP	ОПК-2, ОПК-6	Экзамен, курсовой проект
Тема 2. Вычислительные системы. Основные понятия и классификация	ОПК-6	Экзамен
Тема 3. Параллельные вычисления	ОПК-2, ОПК-6	Экзамен, курсовой проект
Тема 4. Вычислительные системы класса SISD. Внутрипроцессорный параллелизм	ОПК-6	Экзамен
Тема 5. Организация памяти вычислительных систем	ОПК-6	Экзамен
Тема 6. Вычислительные системы класса SIMD	ОПК-6	Экзамен
Тема 7. Гетерогенные вычисления	ОПК-2, ОПК-6	Экзамен, курсовой проект
Тема 8. Вычислительные системы класса MIMD	ОПК-6	Экзамен

## 4 ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ

### 4.1. Промежуточная аттестация в форме экзамена

#### 4.1.1. Компетенция ОПК-6

Код компетенции	Результаты освоения ОПОП Содержание компетенций
ОПК-6	Способен разрабатывать компоненты программно-аппаратных комплексов обработки информации и автоматизированного проектирования.

#### Типовые тестовые вопросы:

##### Вопрос 1

*Для каких целей используются ВС (допускается множественный выбор):*

- +а) для повышения надежности средств ВТ;
- б) для создания системы автоматизации программирования;
- в) для объединения нескольких компьютеров;
- +г) для достижения сверхвысокой производительности.

##### Вопрос 2

*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 3

*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 4

*В каких системах отдельные ЭВМ находятся на значительных расстояниях и обмениваются информацией по каналам связи через специальную аппаратуру в последовательном коде?*

- а) неоднородных;
- б) универсальных;
- в) децентрализованных;
- +г) распределенных.

##### Вопрос 5

*Представителями какого класса являются классические фон-неймановские ВМ*

- +а) SISD;
- б) MISD;
- в) SIMD;

г) MIMD.

### Вопрос 6

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

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

### Вопрос 7

*Укажите особенности процессоров с архитектурой CISC*

- а) устройство управления с жесткой логикой
- +б) микропрограммное устройство управления
- +в) небольшое количество регистров общего назначения
- г) большой регистровый файл
- +д) большое количество различных форматов команд и данных
- е) все команды, кроме «Чтение» и «Запись», имеют формат «регистр-регистр

### Вопрос 8

*Укажите особенности процессоров с архитектурой RISC*

- +а) устройство управления с жесткой логикой
- б) микропрограммное устройство управления
- в) небольшое количество регистров общего назначения
- г+) большой регистровый файл
- д) большое количество различных форматов команд
- +е) все команды, кроме «Чтение» и «Запись», имеют формат «регистр-регистр

### Вопрос 9

*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 10

*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 11

*Главные требования к суперконвейерным процессорам:*

- а) возможность реализации операции в каждой ступени конвейера наиболее сложными техническими средствами;
- б) разность задержки во всех ступенях;

- +в) возможность реализации операции в каждой ступени конвейера наиболее простыми техническими средствами;
- +г) одинаковость задержки во всех ступенях.

### **Вопрос 12**

*Укажите недостатки суперконвейерных процессоров*

- +а) возрастает вероятность конфликтов;
- +б) дороже обходится ошибка предсказания перехода;
- в) необходимость синхронного продвижения команд в каждом из конвейеров;
- + г) усложняется логика взаимодействия ступеней конвейера.

### **Вопрос 135**

*Как называется механизм, когда команды отправляются на исполнение не в той последовательности, в которой они располагаются в коде программы?*

- а) «последовательное исполнение»;
- б) «параллельное исполнение»;
- +в) «внеочередное исполнение»;
- г) «поочередное исполнение».

### **Вопрос 14**

*За счет чего достигается одновременное выполнение более чем одной скалярной команды?*

- а) за счет включения в состав ЦП нескольких связанных функциональных блоков;
- б) за счет включения в состав ЦП одного функционального блока;
- в) за счет исключения из состава ЦП функциональных блоков;
- +г) за счет включения в состав ЦП нескольких самостоятельных функционал. блоков.

### **Вопрос 15**

*Гиперпотокковая технология это –*

- +а) пример реализации идеи параллельной многопоточности;
- б) минимальная загрузка функциональных блоков процессора;
- в) технология формирования таких команд, которые процессор мог бы выполнять последовательно;
- г) технология, которая организует поступление взаимозависимых команд.

### **Вопрос 16**

*Укажите совместно используемые ресурсы при гиперпотокковой технологии.*

- а) логические блоки;
- б) потоки команд;
- +в) функциональные (исполнительные) блоки;
- г) ресурсы вычислительного ядра.

### **Вопрос 17**

*Что используют для поддержания двух полностью независимых потоков?*

- а) кэш-память;
- б) отдельные ресурсы;
- в) совместно используемые ресурсы;
- +г) дублированные ресурсы.

### **Вопрос 18**

*В каких ВС каждый процессор имеет свою локальную память с локальным адресным пространством?*



- +а) с распределенной памятью;
- б) с общей памятью;
- в) с гибридной памятью;
- г) с разделяемой памятью.

### **Вопрос 19**

***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 20**

***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 21**

***Как называются системы, в которых в качестве оперативной памяти используется только локальная кэш-память процессоров?***

- а) cc-NUMA-системы;
- б) ncc-NUMA-системы;
- +в) СОМА-системы;
- г) UMA

### **Вопрос 22**

***В какой архитектуре каждый процессор обладает локальной памятью, но в отличие от NUMA не имеет возможности физического доступа к локальной памяти других процессоров?***

- а) ccNUMA;
- б) СОМА;
- +в) DSM;
- г) NORMA.

### **Вопрос 23**

***Что приводит к так называемой проблеме когерентности?***

- а) несколько уровней кэш-памяти;
- +б) оснащение каждого процессора локальной кэш-памятью;
- в) обращение процессоров к общей памяти;
- г) все перечисленное верно.

### **Вопрос 24**

***При какой записи новая информация одновременно заносится как в кэш, так и в основную память?***

- а) запись с аннулированием;

- б) записи с трансляцией;
- +в) сквозная запись;
- г) обратная запись.

### **Вопрос 25**

*Какой вариант записи предполагает, что любая запись в локальный кэш немедленно дублируется и во всех остальных кэшах, содержащих копию измененного блока?*

- +а) запись с обновлением;
- б) записи с трансляцией;
- в) обратная запись;
- г) сквозная запись.

### **Вопрос 26**

*При какой записи каждый запрос на запись в конкретную кэш-память направляется также и всем остальным кэшам системы?*

- а) обратная запись;
- б) запись с обновлением;
- в) запись с аннулированием;
- +г) широковещательная запись.

### **Вопрос 27**

*Какой протокол предполагает сбор и отслеживание информации о содержимом всех локальных КЭШей?*

- а) протокол Dragon;
- +б) протокол на основе справочника;
- в) протокол MESI;
- г) протокол Berkeley.

### **Вопрос 28**

*На что возлагается ответственность за поддержание когерентности всех кэшей многопроцессорной системы?*

- +а) на контроллеры КЭШей;
- б) на процессор;
- в) оба варианта верны;

Далее см. п.п. 4.1.3

### **Типовые практические задания:**

*См. п.п. 4.1.3*

### **Типовые теоретические вопросы:**

- 1) Что такое вычислительная система.
- 2) Преимущества вычислительных систем
- 3) Чем различаются многомашинные и многопроцессорные системы
- 4) В чем заключается классификация архитектур ВС Флинна
- 5) Какие ВС относятся к классам SISD и SIMD?
- 6) Какие ВС относятся к классам MISD и MIMD?
- 7) Тенденции развития современных ВС
- 8) Пути повышения производительности в системах класса SISD
- 9) Конвейеризация вычислений
- 10) Суперскалярность и внеочередное выполнение команд

- 11) Гиперпоточность
- 12) VLIW-архитектура
- 13) Векторные ВС
- 14) Матричные ВС
- 15) Какие существуют модели архитектур памяти ВС.
- 16) Особенности физически разделяемой памяти (UMA)
- 17) Особенности архитектуры NUMA
- 18) Особенности архитектуры COMA и DSM
- 19) Особенности архитектуры DSM
- 20) Особенности архитектуры NORMA
- 21) Проблема когерентности кэш-памяти
- 22) Программные способы решения проблемы когерентности
- 23) Аппаратные способы решения проблемы когерентности

#### 4.1.2. Компетенция ОПК-2

Код компетенции	Результаты освоения ОПОП Содержание компетенций
ОПК-2	Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач.

#### Типовые тестовые вопросы:

##### Вопрос 1

*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 2

*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 3

*Какой уровень параллелизма обеспечивается операционной системой.*

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

**Вопрос 4**

*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 5**

*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 6**

*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 8**

*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 9**

*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 10**

*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 11**

*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 12**

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 13**

*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 14**

*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.

*Далее см. п.п. 4.1.3*

**Типовые практические задания:**

*См. п.п. 4.1.3*

**Типовые теоретические вопросы:**

- 1) Какие тенденции развития современных ВС.
- 2) Уровни параллелизма. Чем обеспечивается каждый уровень?
- 3) Что такое степень гранулярности. Чем она обеспечивается?
- 4) Метрики параллелизма.
- 5) Закон Амдала
- 6) Закон Густавсона
- 7) Параллельные вычисления в SISD системе (MMX, SSE и AVX технологии)
- 8) Симметричные мультипроцессорные системы (SMP)
- 9) Системы с массовой параллельной обработкой (MPP)
- 10) Кластеры вычислительных машин
- 11) Гетерогенные ВС
- 12) Архитектура GPU

### 4.1.3. Технологии параллельного программирования OpenMP, Cuda и OpenCL

Код компетенции	Результаты освоения ОПОП Содержание компетенций
ОПК-2	Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач;
ОПК-6	Способен разрабатывать компоненты программно-аппаратных комплексов обработки информации и автоматизированного проектирования.

#### Типовые тестовые вопросы:

#### Технология OpenMP

##### Вопрос 1

*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 2

*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 3

*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 4

*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 5

*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 6**

**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 7**

**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 8**

**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 9**

**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 10**

**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 11**

**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 12**

**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 13**

*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 14**

*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 15**

*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 16**

*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 17**

*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 18**

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

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



**Вопрос 19****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 20****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 21****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.

**Технология Cuda****Вопрос 1****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 2****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 3****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 4****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 5

**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 6

**Что такое устройство в терминах CUDA?**

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

### Вопрос 7

**Что такое хост?**

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

### Вопрос 8

**Что такое ядро (kernel)?**

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

### Вопрос 9

**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 10

**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 11

**На скольких одновременно гридах может выполняться ядро?**

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

### Вопрос 12

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

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

### Вопрос 13

**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 14

**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 15

**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 16

**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 17

**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 18

**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 19

**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 20

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 21

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 22

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

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

### Вопрос 23

**В чём преимущества shared памяти?**

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

### Вопрос 24

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

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

## Технология OpenCL

### Вопрос 1

*Что такое OpenCL? Для чего была создана данная технология?*

- a) проприетарный стандарт для вычислений на гомогенных системах;
- б) открытый стандарт для вычислений на гомогенных системах;
- в) проприетарный стандарт для вычислений на гетерогенных системах;
- +г) открытый стандарт для вычислений на гетерогенных системах.

### Вопрос 2

*В чём отличие OpenCL от других технологий выполнения вычислений на GPU?*

- +а) открытость;
- +б) аппаратная независимость;
- в) возможность компиляции на ГПУ;

### Вопрос 3

*Укажите основные модели архитектуры OpenCL.*

- +а) модель платформы;
- б) модель процессоров;
- +в) модель памяти;
- +г) модель исполнения;
- д) модель компиляции;
- +е) модель программирования.

### Вопрос 4

*Какие типы устройств присутствуют в модели платформы OpenCL?*

- +а) хост;
- б) ГПУ;
- +в) устройство;
- г) процессор;
- д) память.

### Вопрос 5

*Что может быть устройством в OpenCL?*

- а) только GPU;
- б) только CPU;
- в) CPU или GPU;
- +г) любое устройство, поддерживающее OpenCL.

### Вопрос 6

*Каким образом происходит работа программы в OpenCL?*

- а) компиляция и выполнение на хосте;
- б) компиляция на хосте, исполнение на устройстве;
- в) компиляция и выполнение на устройстве;
- +г) компиляция на хосте, исполнение как на хосте, так и на устройстве.

### Вопрос 7

**Что такое рабочий элемент, рабочая группа?**

- а) вычислительное ядро, группа вычислительных ядер;
- б) устройство; группа устройств;
- +в) экземпляр исполняющейся программы, подмножество исполняющихся программ;
- г) вычислительная система, группа вычислительных систем.

**Вопрос 8**

**Соотнесите понятия рабочий элемент и рабочая группа в OpenCL с таковыми понятиями CUDA.**

- +а) ядро, блок;
- б) тред, блок;
- в) ядро, грид;
- г) блок, грид.

**Вопрос 9**

**Что такое процессорный элемент, вычислительная единица?**

- +а) вычислительное ядро, вычислительный процессор;
- б) вычислительное устройство; группа вычислительных устройств;
- в) экземпляр исполняющейся программы, подмножество исполняющихся программ;
- г) вычислительная система, группа вычислительных систем.

**Вопрос 10**

**Соотнесите понятия процессорный элемент, вычислительная единица в OpenCL с таковыми понятиями CUDA.**

- +а) функциональный блок, мультипроцессор;
- б) тред, блок;
- в) ядро, мультипроцессор;
- г) блок, грид.

**Вопрос 11**

**Укажите способы идентификации рабочего элемента.**

- +а) глобальный индекс;
- +б) комбинация номера группы и локального индекса;
- в) комбинация номера группы и локального индекса в нескольких измерениях;
- г) комбинация глобального индекса.

**Вопрос 12**

**Гарантируется ли параллельное выполнение нескольких рабочих групп?**

- +а) не гарантируется;
- б) гарантируется при указании специального флага компиляции;
- в) гарантируется при использовании специальных функций;
- г) гарантируется всегда.

**Вопрос 13**

**Что такое NDRange?**

- а) одномерное пространство индексов;
- +б) многомерное пространство индексов;
- в) одномерное пространство рабочих элементов;
- г) многомерное пространство рабочих элементов.

**Вопрос 14**

**К какой модели относится контекст OpenCL?**

- а) модели платформы;
- +б) модели исполнения;
- в) модели памяти;
- г) модели программирования.

### Вопрос 15

*Для чего нужен контекст OpenCL и что он содержит?*

- +а) необходим для выполнения, так как содержит устройства, ядра, программные объекты и объекты памяти;
- б) необходим для компиляции, так как содержит программные объекты и объекты памяти;
- в) необходим для выполнения, так как содержит устройства и ядра;
- г) необходим для компиляции, так как содержит устройства и ядра;

### Вопрос 16

*Каким образом выполняются очереди команд? Выберите верные утверждения.*

- +а) в разных контекстах – параллельно;
- б) в одном контексте – параллельно;
- в) в разных контекстах – конкурируя между собой;
- +г) в одном контексте – конкурируя между собой;

### Вопрос 17

*Выберите типы объектов памяти OpenCL.*

- а) объекты векторов;
- +б) объекты буферов;
- +в) объекты изображений;
- г) объекты матриц.

### Вопрос 18

*Выберите основные типы памяти OpenCL.*

- а) память хоста;
- б) память устройства;
- +в) глобальная память;
- +г) локальная память;
- +д) константная память;
- е) динамическая память;
- +ж) частная память;
- з) общая память.

### Вопрос 19

*Чем локальная память отличается от частной?*

- а) локальная – это память устройства, частная – память процессорного элемента;
- +б) локальная – это память рабочей группы, частная – память рабочего элемента;
- в) локальная – это память устройства, частная – память рабочей группы;
- г) локальная – это память контекста, частная – память хоста.

### Вопрос 20

*В чём отличие программной модели OpenCL от модели программирования CUDA?*

- а) поддерживается параллелизм команд;
- б) не поддерживается параллелизм команд;
- +в) поддерживается параллелизм данных;
- г) не поддерживается параллелизм данных.

**Вопрос 21**

**Укажите основные способы синхронизации выполнения в OpenCL.**

- а) синхронизация рабочих групп между собой;
- +б) синхронизация рабочих элементов в рабочей группе;
- +в) синхронизация очередей команд в одном контексте;
- г) синхронизация очередей команд в разных контекстах.

**Вопрос 22**

Для чего нужны модификаторы (квалификаторы) `__global`, `__local`, `__constant`, `__private`?

- +а) это квалификаторы адресного пространства (памяти);
- б) это квалификаторы функций ядра;
- в) это квалификаторы всех функций OpenCL;
- г) это квалификаторы управления ходом выполнения.

**Типовые практические задания:**

Выполняются 2 задания одно с использованием технологии OpenMP, второе – с использованием технологии Cuda или OpenCL.

**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 [10x10]. В первый элемент каждой строки записать номер исполняющегося 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 [100x100].

### *Технология OpenCL*

- 1) Создать матрицу размера [10x10]. В первый элемент каждой строки записать номер рабочей группы, 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 исполняющего элемента, 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 [100x100].

#### 4.2. Промежуточная аттестация в форме курсового проекта

Код компетенции	Результаты освоения ОПОП Содержание компетенций
ОПК-2	Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач;
ОПК-6	Способен разрабатывать компоненты программно-аппаратных комплексов обработки информации и автоматизированного проектирования.

### **Типовое задание для курсовой работы по дисциплине:**

Главной целью написания курсового проекта является проверка усвоения студентами знаний в области использования вычислительных систем для достижения сверхвысокой производительности, получение практического опыта программирования многопроцессорных систем с общей памятью и графических процессоров с использованием технологий OpenMP и CUDA (или OpenCL).

Тема курсового проекта: «Параллельные вычисления с использованием технологий OpenMP и CUDA (OpenCL)»

#### ***Основные направления для исследований:***

- 1) Сортировка данных
- 2) Обработка графов
- 3) Параллельные методы матричного умножения
- 4) Параллельные методы решения систем линейных уравнений
- 5) Обработка изображений.
- 6) Корреляционное совмещение изображений.

#### ***Перечень индивидуальных заданий:***

- 1) Параллельные алгоритмы на графах (алгоритмы Флойда и Прима)
- 2) Параллельные методы матричного умножения. Блочные алгоритмы умножения матриц.
- 3) Параллельные методы сортировки. Алгоритмы пузырьковой сортировки и Шелла.
- 4) Параллельные методы решения систем линейных уравнений. Метод Гаусса.
- 5) Реализация алгоритмов контрастирования изображений.
- 6) Параллельные методы решения систем линейных уравнений. Метод сопряженных градиентов.
- 7) Реализация алгоритмов выделения контуров изображений.
- 8) Реализация алгоритма кластеризации изображения (k-means)
- 9) Корреляционное совмещение изображений методом поэтапного сканирования.
- 10) Корреляционное совмещение изображений с использованием пирамиды изображений.
- 11) Параллельные методы сортировки. Алгоритм быстрой сортировки (алгоритм Хоара).
- 12) Распознавание образов с использованием инвариантных моментов Ну.
- 13) Реализация алгоритмов фильтрации изображений.
- 14) Параллельные методы матричного умножения. Алгоритм умножения матриц, основанный на ленточном разделении данных.
- 15) Реализация параллельного алгоритма обнаружения границ Canny.
- 16) Реализация алгоритмов морфологических операций.
- 17) Быстрое преобразование Фурье
- 18) Корреляционное совмещение изображений методом мультистарт) (локальный метод – МДМ)
- 19) Корреляционное совмещение изображений методом мультистарт (локальный метод – Хука-Дживса)
- 20) Корреляционное совмещение изображений методом случайного поиска

Индивидуальные задания выбираются студентами и фиксируются преподавателем. Дублирование тем для индивидуального исследования в пределах одной учебной группы не допускается.

Защита курсового проекта назначается по итогам проверки предоставленной пояснительной записки, оформленной в соответствии с требованиями и разработанного приложения. Защита осуществляется в форме ответов на вопросы преподавателя.

#### **Требования к содержанию пояснительной записки:**

Титульный лист

Задание

Содержание

Введение

1. Теоретический материал по индивидуальному заданию
2. Постановка задачи
3. Разработка алгоритмов и программ
4. Сравнительный анализ последовательной и параллельных реализаций
5. Разработка технической документации
6. Список использованных источников

Приложение

#### **Типовые вопросы на защите курсового проекта:**

1. *Что представляет собой последовательный алгоритм для решения вашей задачи. По каким критериям он подходит для распараллеливания?*
2. *Какие директивы OpenMP использовались при разработке параллельной программы?*
3. *Какие виды синхронизации используются в вашей программе и зачем?*
4. *Сколько потоков используется в вашей программе на OpenMP? Можно ли изменить количество потоков?*
5. *Какие режимы планирования распараллеливания итераций цикла вы использовали?*
6. *Какие шаги, должна выполнить программа, которая выполняет расчёты на GPU с помощью Cuda.*
7. *Какие шаги, должна выполнить программа, которая выполняет расчёты на GPU с помощью OpenCL.*
8. *Что такое ускорение?*
9. *Какое ускорение получено при реализации параллельной программы с помощью технологии OpenMP?*
10. *Какое ускорение получено при реализации параллельной программы с помощью технологии Cuda (OpenCL)?*
11. *При каких значениях параметра метода целесообразно использовать параллельные вычисления и почему?*