

**МИНОБРНАУКИ РОССИИ**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**  
**ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**  
**(ФГБОУ ВО «ВГУ»)**



**УТВЕРЖДАЮ**  
Заведующий кафедрой  
Кургалин Сергей Дмитриевич  
Кафедра цифровых технологий  
28.02.2022

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

Б1.В.01 Параллельные и GRID-технологии

**1. Код и наименование направления подготовки/специальности:**

02.04.01 Математика и компьютерные науки

**2. Профиль подготовки/специализация:**

Компьютерное моделирование и искусственный интеллект

**3. Квалификация (степень) выпускника:**

Магистратура

**4. Форма обучения:**

Очная

**5. Кафедра, отвечающая за реализацию дисциплины:**

Кафедра цифровых технологий

**6. Составители программы:**

Борзунов Сергей Викторович, кандидат физико-математических наук, доцент

**7. Рекомендована:**

НМС ФКН (протокол № 3 от 25.02.2022)

**8. Учебный год:**

2023-2024

Семестры: 3

## 9. Цели и задачи учебной дисциплины:

Целью дисциплины является приобретение обучающимися знаний, навыков, опыта и профессиональных компетенций в области параллельной обработки информации, технологий распределённых вычислений и обработки данных. Задачей дисциплины является приобретение обучающимися практических навыков работы с распределёнными GRID-системами.

## 10. Место учебной дисциплины в структуре ООП:

Часть блока Б1, формируемая участниками образовательных отношений. Для успешного освоения дисциплины необходимо предварительное изучение следующих разделов компьютерных наук: архитектура ЭВМ, сетевые технологии.

## 11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями выпускников) и индикаторами их достижения:

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ПК-9	Способен использовать современные методы разработки и реализации конкретных алгоритмов математических моделей на базе языков программирования и пакетов прикладных программ моделирования	ПК-9.1	Владеет современными методами разработки и реализации алгоритмов математических моделей на базе языков и пакетов прикладных программ моделирования	Знает математическую формулировку базовых алгоритмов анализа данных
		ПК-9.2	Умеет разрабатывать и реализовывать алгоритмы математических моделей на базе языков и пакетов прикладных программ моделирования	Умеет реализовывать математически сложные алгоритмы на высокоуровневых языках программирования
		ПК-9.3	Имеет практический опыт разработки и реализации алгоритмов на базе языков и пакетов прикладных программ моделирования	Владеет навыками адаптации известных алгоритмов для решения практических задач и реализации в программных комплексах

## 12. Объем дисциплины в зачетных единицах/час:

3/108

Форма промежуточной аттестации: зачет

## 13. Трудоемкость по видам учебной работы

Вид учебной работы		Трудоемкость	
		Всего	По семестрам
			3 семестр
Аудиторные занятия		28	28
в том числе:	лекции	14	14
	практические		
	лабораторные	14	14
Самостоятельная работа		80	80
в том числе: курсовая работа (проект)			
Форма промежуточной аттестации (зачет)			
Итого:		108	108

### 13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
1	Технологии параллельного программирования	Параллельное программирование с использованием MPI. Структура MPI- программы. Сообщения, их передача и прием. Синхронное и асинхронное взаимодействие. Коллективный обмен данными. Виды коллективного обмена, барьеры, широковещательная рассылка данных. Система программирования OpenMP.	<a href="https://edu.vsu.ru/course/view.php?id=3853">https://edu.vsu.ru/course/view.php?id=3853</a>
2	Кластерные системы и инфраструктура грид	Общие принципы построения кластерных систем. Введение в архитектуры и средства программирования многопроцессорных вычислительных систем. Кластерные системы управления пакетной обработкой. СПО Torque. Понятие распределённых вычислений. Определение грид- инфраструктуры. Основные функциональные подсистемы глобального грида. Базовые функции, физическая структура грида. Знакомство с ПО ARC Nordugrid. Grid- сертификаты, переменные окружения. Запуск задач в ARC. Задания без входных данных. Задания с внешними данными и файлами.	<a href="https://edu.vsu.ru/course/view.php?id=3853">https://edu.vsu.ru/course/view.php?id=3853</a>
3	Реализация грид-технологий в проектах EGEE и NorduGrid	Проект NorduGrid. Проект EGEE. Российский сегмент RDIG. Промежуточное программное обеспечение ARC. Мониторинг внешних ресурсов в рамках NorduGrid.	<a href="https://edu.vsu.ru/course/view.php?id=3853">https://edu.vsu.ru/course/view.php?id=3853</a>

### 13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела)	Лекционные занятия	Практические занятия	Лабораторные занятия	Самостоятельная работа	Всего
1	Технологии параллельного программирования	6		6	30	42
2	Кластерные системы и инфраструктура грид	4		4	34	42
3	Реализация грид-технологий в проектах EGEE и NorduGrid	4		4	16	24
	Итого	14		14	80	108

### 14. Методические указания для обучающихся по освоению дисциплины

Освоение дисциплины складывается из аудиторной работы (учебной деятельности, выполняемой под руководством преподавателя) и внеаудиторной работы (учебной деятельности, реализуемой обучающимся самостоятельно).

Аудиторная работа состоит из работы на лекциях и выполнения практических (или лабораторных) заданий в объёме, предусмотренном учебным планом. Лекция представляет собой последовательное и систематическое изложение учебного материала, направленное на знакомство обучающихся с основными понятиями и теоретическими положениями изучаемой дисциплины. Лекционные занятия формируют базу для лабораторных занятий, на которых полученные теоретические знания применяются для решения конкретных практических задач. Обучающимся для успешного освоения дисциплины рекомендуется вести конспект лекций и лабораторных занятий.

Самостоятельная работа предполагает углублённое изучение отдельных разделов дисциплины с использованием литературы, рекомендованной преподавателем, а также конспектов лекций, презентационным материалом (при наличии) и конспектов лабораторных занятий. В качестве плана для самостоятельной работы может быть использован раздел 13.1 настоящей рабочей программы, в котором зафиксированы разделы дисциплины и их содержание. В разделе 13.2 рабочей программы определяется количество часов, отводимое на самостоятельную работу по каждому разделу дисциплины. Большее количество часов на самостоятельную работу отводится на наиболее трудные разделы дисциплины. Для самостоятельного изучения отдельных разделов дисциплины используется перечень литературы и других ресурсов, перечисленных в пунктах 15 и 16 настоящей рабочей программы.

Успешность освоения дисциплины определяется систематичностью и глубиной аудиторной и внеаудиторной работы обучающегося.

При использовании дистанционных образовательных технологий и электронного обучения выполнять все указания преподавателей, вовремя подключаться к online занятиям, ответственно подходить к заданиям для самостоятельной работы.

### 15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины

№ п/п	Источник
1	Кареева, Е. Д. Основы многопоточного и параллельного программирования : учебное пособие / Е.Д. Кареева ; Красноярск : СФУ, 2016 .— 355 с. : ил. — Библиогр. в кн.— <a href="http://biblioclub.ru/">http://biblioclub.ru/</a> .— ISBN 978-5-7638-3385-0 .— <URL: <a href="http://biblioclub.ru/index.php?page=book&amp;id=497217">http://biblioclub.ru/index.php?page=book&amp;id=497217</a> >.
2	Туральчук, К. А. Параллельное программирование с помощью языка C / К.А. Туральчук .— 2-е изд., испр. — Москва : Национальный Открытый Университет «ИНТУИТ», 2016 .— 190 с. : ил. — <a href="http://biblioclub.ru/">http://biblioclub.ru/</a> .— <URL: <a href="http://biblioclub.ru/index.php?page=book&amp;id=429098">http://biblioclub.ru/index.php?page=book&amp;id=429098</a> >.

б) дополнительная литература:

№ п/п	Источник
1	<b>Уильямс, Э.</b> Параллельное программирование на C++ в действии. Практика разработки многопоточных программ [Электронный ресурс] : учебное пособие. — Электрон. дан. — М. : ДМК Пресс, 2012. — 672 с. — <URL: <a href="http://e.lanbook.com/books/element.php?pl1_id=4813">http://e.lanbook.com/books/element.php?pl1_id=4813</a> >

в) информационные электронно-образовательные ресурсы:

№ п/п	Источник
1	Электронный университет ВГУ <a href="https://edu.vsu.ru">https://edu.vsu.ru</a>
2	ЭБС «Лань» <a href="https://e.lanbook.com/">https://e.lanbook.com/</a>
3	«Университетская библиотека online» <a href="https://biblioclub.ru/">https://biblioclub.ru/</a>
4	«Консультант студента» <a href="http://www.studmedlib.ru/">http://www.studmedlib.ru/</a>
5	Электронный университет ВГУ: <a href="https://edu.vsu.ru/">https://edu.vsu.ru/</a>

### 16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п	Источник
1	<b>Уильямс, Э.</b> Параллельное программирование на C++ в действии. Практика разработки многопоточных программ [Электронный ресурс] : учебное пособие. — Электрон. дан. — М.: ДМК Пресс, 2012. — 672 с. — <URL: <a href="http://e.lanbook.com/books/element.php?pl1_id=4813">http://e.lanbook.com/books/element.php?pl1_id=4813</a> >

## **17. Образовательные технологии, используемые при реализации учебной дисциплины, включая дистанционные образовательные технологии (ДОТ), электронное обучение (ЭО), смешанное обучение):**

При реализации дисциплины могут использоваться технологии электронного обучения и дистанционные образовательные технологии на базе портала edu.vsu.ru, а также другие доступные ресурсы сети Интернет.

## **18. Материально-техническое обеспечение дисциплины:**

Аудитория для лекционных занятий: мультимедиа-проектор, экран для проектора, компьютер с выходом в сеть «Интернет». Специализированная мебель (столы ученические, стулья, доска). Программное обеспечение: LibreOffice v.5-7, программа для просмотра файлов формата pdf, браузер.

Аудитория для лабораторных занятий: компьютеры с выходом в сеть «Интернет» и доступом к электронным библиотечным системам, специализированная мебель (столы ученические, стулья, доска). Программное обеспечение: LibreOffice v.5-7, программа для просмотра файлов формата pdf, браузер.

## **19. Оценочные средства для проведения текущей и промежуточной аттестаций**

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

№ п/п	Разделы дисциплины (модули)	Код компетенции	Код индикатора	Оценочные средства для текущей аттестации
1	Разделы 1-3	ПК-9	ПК-9.1	Письменный опрос; лабораторные работы 1 – 7
2	Разделы 1-3	ПК-9	ПК-9.2	Письменный опрос; лабораторные работы 1 – 7
3	Разделы 1-3	ПК-9	ПК-9.3	Письменный опрос; лабораторные работы 1 – 7

Промежуточная аттестация форма контроля - Зачет

Оценочные средства для промежуточной аттестации Перечень вопросов к зачету

## **20 Типовые оценочные средства и методические материалы, определяющие процедуры оценивания**

### **20.1 Текущий контроль успеваемости**

Текущий контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств:

- письменный опрос;
- лабораторная работа.

#### **Перечень вопросов для письменного опроса**

1. Получение сертификата для обеспечения аутентификации в Globus.
2. Кластерные системы управления пакетной обработкой.
3. Установка прав доступа к открытому и закрытому ключам пользователя.

4. Системы распределённых вычислений. Определение Grid-инфраструктуры.
5. Синтаксис XRSL.
6. Базовые функции Grid.
7. Наблюдение за процессом выполнения задачи.
8. Физическая структура Grid.
9. Запуск простого задания без входных данных.
10. Архитектура сервисов распределённых систем и технологии ее реализации.
11. Grid-сертификаты, переменные окружения.
12. Архитектуры и средства программирования многопроцессорных вычислительных систем.
13. Жизненный цикл задания в Grid.
14. Создайте скрипт, выводящий на экран строку «Argument=...», где вместо троеточия подставляется аргумент командной строки.
15. Средства управления заданиями и их мониторинг в Grid.
16. Переадресация задачи кластерному программному обеспечению (TORQUE).
17. Архитектуры и средства программирования многопроцессорных вычислительных систем.
18. Кластерные системы управления пакетной обработкой.
19. Запуск задания от имени другого пользователя.

Технология проведения: обучающемуся выдается случайным образом билет, состоящий из двух вопросов. Полный и логически обоснованный ответ на каждый вопрос оценивается в 25 баллов.

### **Перечень лабораторных работ**

1. Табулирование функций.
2. Основные функции MPI. Процедуры Bcast, Scatter, Scatterv.
3. Основные функции MPI. Процедуры Gather, Gatherv, Alltoall.
4. Основные функции MPI. Процедуры Reduce, AllReduce, Scan.
5. Основные функции MPI. Процедуры Bcast, Scatter, Scatterv.
6. Grid-сертификаты, переменные окружения.
7. Запуск задач в ARC.

### **Типовое задание для лабораторной работы**

#### **Лабораторная работа № 1**

#### **«Табулирование функций»**

**Цель работы:** изучение и исследование параллельных методов табулирования функций.

**Требования к выполнению работы:** выполнение лабораторной работы предусматривает написание программы, реализующей изучаемые методы, и проверку её работы на контрольном примере.

**Отчёт о работе** проводится в виде собеседования и заключается в демонстрации работы программы, объяснении принципов работы алгоритма и ответов на дополнительные вопросы.

**Критерии оценки:** для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

**Задание:** написать программу, реализующую параллельное вычисление таблицы значений функций. Проверить работу программы на контрольном примере. Исследовать сходимость методов и провести сравнение по числу шагов, потребовавшихся для достижения указанной точности.

## 20.2 Промежуточная аттестация

Промежуточная аттестация по дисциплине осуществляется с помощью следующих оценочных средств:

- собеседование по вопросам из перечня.

### Перечень вопросов к зачету

1. Получение сертификата для обеспечения аутентификации в Globus.
2. Кластерные системы управления пакетной обработкой.
3. Установка прав доступа к открытому и закрытому ключам пользователя.
4. Системы распределённых вычислений. Определение Grid-инфраструктуры.
5. Синтаксис XRSL.
6. Базовые функции Grid.
7. Наблюдение за процессом выполнения задачи.
8. Физическая структура Grid.
9. Запуск простого задания без входных данных.
10. Архитектура сервисов распределённых систем и технологии ее реализации.
11. Grid-сертификаты, переменные окружения.
12. Архитектуры и средства программирования многопроцессорных вычислительных систем.
13. Жизненный цикл задания в Grid.
14. Создайте скрипт, выводящий на экран строку «Argument=...», где вместо троеточия подставляется аргумент командной строки.
15. Средства управления заданиями и их мониторинг в Grid.
16. Переадресация задачи кластерному программному обеспечению (TORQUE).
17. Архитектуры и средства программирования многопроцессорных вычислительных систем.
18. Кластерные системы управления пакетной обработкой.
19. Запуск задания от имени другого пользователя.

Технология проведения: обучающемуся выдается случайным образом билет, состоящий из двух вопросов.

**Приведённые ниже задания рекомендуется использовать при проведении диагностических работ для оценки остаточных знаний по дисциплине**

### Тестовые задания - 1 балл

1. Компиляция программы в рамках стандарта MPI может быть выполнена с помощью команды:  
a) mpirun  
**b) mpifort**  
c) mpicxx  
d) mpicc
2. Запуск программы в рамках стандарта MPI может быть выполнен с помощью команды:  
a) mpirun  
**b) mpiexec**  
c) mpicc  
d) mpifort

3. Получить значение ранга процесса можно с помощью подпрограммы MPI:

- a) MPI\_Comm\_size
- b) **MPI\_Comm\_rank**
- c) MPI\_Init
- d) MPI\_Finalize

4. Получить значение размера коммуникатора можно с помощью подпрограммы MPI:

- a) **MPI\_Comm\_size**
- b) MPI\_Comm\_rank
- c) MPI\_Init
- d) MPI\_Finalize

5. Стандартный коммуникатор, содержащий все запущенные процессы MPI-программы, называется

- a) **MPI\_COMM\_WORLD**
- b) MPI\_COMM\_NULL
- c) MPI\_ANY\_TAG
- d) MPI\_ANY\_SOURCE

6. Первым по порядку вызовом подпрограммы MPI может быть вызов:

- a) MPI\_Initialized
- b) **MPI\_Init**
- c) MPI\_Comm\_rank
- d) MPI\_Comm\_size

7. В коллективной операции обмена данных принимают участие:

- a) не более двух процессов
- b) ровно два процесса
- c) **все процессы, принадлежащие указанному коммуникатору**
- d) не менее двух процессов

8. Для выполнения широковещательной рассылки сообщений применяется:

- a) **MPI\_Bcast**
- b) MPI\_Bsend
- c) MPI\_Send
- d)

9. К классу коллективных операций MPI относятся:

- a) **сбор данных**
- b) **распределение данных**
- c) **операция редукции данных**
- d) обмен по готовности

10. В рамках операции редукции можно выполнить

- a) **суммирование элементов массива**
- b) умножение двух матриц
- c) **определение максимального элемента**
- d) **перемножение элементов массива**

11. Операция широковещательной рассылки

- a) **пересылает одни и те же данные всем процессам**
- b) выполняет частичную редукцию
- c) пересылает всем остальным процессам разные фрагменты данных
- d)

12. Подпрограмма MPI\_Gather:

- a) выполняет широковещательную рассылку
- b) **выполняет сбор данных**
- c) **является операцией коллективного обмена данными**
- d)



13. Подпрограмма MPI\_Scatter:

- a) выполняет распределение данных между процессами
- b) выполняет широковещательную рассылку
- c) является операцией синхронизации
- d)

14. Какие из следующих операций выполняют сбор данных на одном процессе

- a) **MPI\_Gather**
- b) **MPI\_Gatherv**
- c) MPI\_Allgather
- d)

15. Какие их перечисленных операций относятся к классу векторных

- a) MPI\_Alltoall
- b) **MPI\_Alltoallv**
- c) MPI\_Allgather
- d) **MPI\_Allgatherv**

16. Подпрограмма MPI\_Alltoall:

- a) выполняет операцию частичной редукции
- b) выполняет операцию полной редукции
- c) выполняет операцию синхронизации
- d) **выполняет передачу данных по схеме «каждый-всем»**

17. Подпрограмма MPI\_Scan:

- a) **выполняет операцию частичной редукции**
- b) выполняет операцию полной редукции
- c) выполняет операцию синхронизации
- d) выполняет передачу данных по схеме «точка-точка»

18. В рамках стандарта MPI создание производного типа в MPI выполняется функцией:

- a) MPI\_Comm\_create
- b) MPI\_Type\_create
- c) **MPI\_Type\_create\_struct**
- d)

19. В рамках стандарта MPI создание структурного типа в MPI выполняется функцией

- a) **MPI\_Type\_struct**
- b) MPI\_Type\_create
- c) MPI\_Comm\_create
- d)

20. Коммуникатором в MPI называется:

- a) **группа процессов, наделенная общим контекстом обмена данными**
- b) подпрограмма, выполняющая передачу или прием сообщения
- c) подпрограмма для обработки сигналов, поступающих от операционной системы
- d)

21. Для создания коммуникатора предусмотрена команда

- a) MPI\_Comm\_size
- b) **MPI\_Comm\_create**
- c) MPI\_Comm\_rank
- d)

22. Создать группу коммуникаторов можно с помощью команды

- a) **MPI\_Group\_excl**
- b) **MPI\_Group\_incl**
- c) **MPI\_Group\_difference**
- d) MPI\_Group\_free

23. Подпрограмма MPI\_Comm\_size:

- a) является операцией двухточечного обмена
- b) **возвращает количество процессов в указанном коммуникаторе**
- c) задает количество процессов в указанном коммуникаторе
- d) проверяет возможность создания коммуникатора с указанным размером

24. Определить количество процессов в коммуникаторе можно с помощью:

- a) **MPI\_Comm\_size**
- b) MPI\_Comm\_rank
- c) MPI\_Comm\_create
- d)

25. В MPI существуют следующие типы двухточечных обменов:

- a) **стандартный обмен**
- b) **обмен по готовности**
- c) **синхронный обмен**
- d) **обмен с буферизацией**

26. Стандартная блокирующая двухточечная передача выполняется

- a) **MPI\_Send**
- b) MPI\_Ssend
- c) MPI\_Bsend
- d) MPI\_Rsend

27. Двухточечная передача с буферизацией выполняется функцией

- a) MPI\_Send
- b) MPI\_Ssend
- c) **MPI\_Bsend**
- d) MPI\_Rsend

28. Неблокирующий вариант операций передачи сообщений существует для:

- a) **стандартной операции обмена данными**
- b) **обмена с буферизацией**
- c) **обмена «по готовности»**
- d)

29. Неблокирующий прием в MPI выполняется :

- a) **MPI\_Irecv**
- b) MPI\_Recv
- c) MPI\_Reduce\_Scatter
- d)

30. Подпрограмма MPI\_Wait предназначена для:

- a) **блокирующей проверки выполнения обмена**
- b) неблокирующей проверки выполнения обмена
- c) приостановки выполнения программы на заданный период времени
- d)

31. Подпрограмма MPI\_Test предназначена для:

- a) блокирующей проверки выполнения обмена
- b) **неблокирующей проверки выполнения обмена**
- c) приостановки выполнения программы на заданный период времени
- d)

32. Подпрограмма MPI\_Testall выполняет проверку:

- a) завершения любой операции обмена из некоторого множества
- b) **завершения всех обменов**
- c) завершения одного обмена
- d)

33. Подпрограмма MPI\_Testany выполняет проверку:

- a) **завершения любой операции обмена из некоторого множества**
- b) завершения всех обменов
- c) завершения одного обмена
- d)

34. Неблокирующая передача с буферизацией выполняется :

- a) **MPI\_IbSend**
- b) MPI\_Bsend
- c) MPI\_Test
- d) MPI\_Testall

35. Средством разработки многопоточных программ является:

- a) **OpenMP**
- b) PVM
- c) MPI
- d)

36. Под "гонкой данных" понимается ситуация, когда:

- a) не хватает долговременной памяти для размещения данных
- b) **несколько потоков работают с разделяемыми данными, и конечный результат зависит от порядка доступа потоков к данным**
- c) несколько потоков работают с данными, локализованными для каждого из потоков, и конечный результат собирается в одну переменную
- d) не хватает оперативной памяти для размещения данных

37. Какие существуют подходы к использованию ресурсов параллелизма в разрабатываемой программе

- a) **автоматическое распараллеливание последовательной версии программы средствами компилятора**
- b) **использование специализированных языков для параллельного программирования**
- c) **использование библиотек, предоставляющих возможности параллельного исполнения кода**
- d) **программирование с использованием особых расширений языка — средств распараллеливания**

38. Какие виды зависимости операторов в программе существуют

- a) **зависимость по выходу**
- b) **поточковая зависимость**
- c) **антизависимость**
- d)

39. Какие ключевые слова OpenMP можно использовать после директивы parallel

- a) **firstprivate**
- b) **lastprivate**
- c) **reduction**
- d) runtime

40. Какие ключевые слова OpenMP можно использовать после директивы single

- a) **private**
- b) **firstprivate**
- c) **nowait**
- d) **copyprivate**

41. Произвольный замок в OpenMP может находиться в состоянии:

- a) **неинициализированном**
- b) **разблокированном**
- c) **заблокированном**
- d) **распределенном**

42. Какая из приведенных в лекции топологий (при одинаковом количестве процессоров) обладает

наименьшим диаметром:

- a) топология гиперкуб
- b) топология линейка
- c) топология полный граф**
- d) топология трехмерной решетки

43. Какое сокращение соответствует стандартной одноядерной архитектуре в рамках классификации Флинна?

- a) SISD**
- b) SIMD
- c) MISD
- d) MIMD

44. Какая архитектура в рамках классификации Флинна наиболее распространена среди суперкомпьютеров?

- a) SISD
- b) SIMD
- c) MISD
- d) MIMD**

45. Нарушению какого условия Бернштейна соответствует наличие зависимости по выходным данным?

- a) нарушению первого условия Бернштейна**
- b) нарушению первого условия Бернштейна
- c) нарушению первого условия Бернштейна
- d) ни одно из условий Бернштейна не нарушено

#### **Задания с коротким ответом — 2 балла**

46. Какой аббревиатурой в классификации Флинна обозначается класс вычислительных систем, поддерживающих работу одного потока команд, одного потока данных?

47. Какой аббревиатурой в классификации Флинна обозначается класс вычислительных систем, поддерживающих работу одного потока команд, нескольких потоков данных?

48. Какой аббревиатурой в классификации Флинна обозначается класс вычислительных систем, поддерживающих работу нескольких потоков команд, одного потока данных?

49. Какой аббревиатурой в классификации Флинна обозначается класс вычислительных систем, поддерживающих работу нескольких потоков команд, нескольких потоков данных?

50. В системах с распределенной памятью каждый вычислительный узел имеет доступ только к принадлежащему ему участку памяти — локальной памяти. В этом случае для межпроцессорного обмена данными предусматривается возможность отправки и приема сообщений по коммуникационной сети, объединяющей вычислительную систему. Как называется соответствующая модель программирования?

51. Как называется связный ациклический граф?

52. Какой аббревиатурой принято обозначать модель вычислительной системы – машину с произвольным доступом к памяти?

53. Какой аббревиатурой принято обозначать модель вычислительной системы – параллельную машину с произвольным доступом к памяти?

54. Как называется величина, равная отношению  $T1(N)/Tp(N)$ , где  $p$  – количество вычислительных узлов?

55. Как называется величина, равная отношению  $T1(N)/(p \cdot Tp(N))$ ?

56. Величина, равная произведению времени параллельного решения задачи и числа используемых процессоров называется «... вычислений». Заполните пропуск.

57. Как называется упорядоченный по времени набор векторов, образованных всеми данными в памяти вычислительной системы?

58. Какую аббревиатуру используют для обозначения отдельного устройства вычислительной системы, способное выполнять графический рендеринг, а также высокопроизводительные вычисления наряду с центральным процессором?

**Ответы:**

46. SISD

47. SIMD
48. MISD
49. SISD
50. модель передачи сообщений
51. дерево
52. RAM
53. PRAM
54. ускорение
55. эффективность
56. стоимость
57. история содержимого памяти
58. GPU

### 3) задания с развернутым ответом

59. Время работы последовательной версии некоторого алгоритма  $A$  равно  $T_1(N) = 2N \log_2(N)$  т, где  $N$  — размер входных данных,  $t$  — время выполнения одной вычислительной операции. В предположении, что алгоритм допускает максимальное распараллеливание, т.е. время работы на вычислительной системе с  $p$  процессорами равно  $T_p(N) = T_1(N) / p$ , вычислите время работы алгоритма  $A$  в следующем случае:  $N = 32$ ,  $p = 4$ .

60. Время работы последовательной версии некоторого алгоритма  $A$  равно  $T_1(N) = 2N \log_2(N)$  т, где  $N$  — размер входных данных,  $t$  — время выполнения одной вычислительной операции. В предположении, что алгоритм допускает максимальное распараллеливание, т.е. время работы на вычислительной системе с  $p$  процессорами равно  $T_p(N) = T_1(N) / p$ , вычислите время работы алгоритма  $A$  в следующем случае:  $N = 32$ ,  $p = 16$ .

61. Время работы последовательной версии некоторого алгоритма  $B$  равно  $T_1(N) = 2^N t$ , где  $N$  — размер входных данных,  $t$  — время выполнения одной вычислительной операции. В предположении, что алгоритм допускает максимальное распараллеливание, т.е. время работы на вычислительной системе с  $p$  процессорами равно  $T_p(N) = T_1(N) / p$ , вычислите время работы алгоритма  $B$  в следующем случае:  $N = 32$ ,  $p = 4$ .

62. Время работы последовательной версии некоторого алгоритма  $B$  равно  $T_1(N) = 2^N t$ , где  $N$  — размер входных данных,  $t$  — время выполнения одной вычислительной операции. В предположении, что алгоритм допускает максимальное распараллеливание, т.е. время работы на вычислительной системе с  $p$  процессорами равно  $T_p(N) = T_1(N) / p$ , вычислите время работы алгоритма  $B$  в следующем случае:  $N = 32$ ,  $p = 16$ .

63. Пусть доля последовательных вычислений  $f$  в некоторой программе равна  $1/10$ . Вычислите максимальное ускорение программы  $(S_p)_{\max}$  на вычислительной системе с  $p$  процессорами с учетом закона Амдала.

64. Пусть доля последовательных вычислений  $f$  в некоторой программе равна  $1/100$ . Вычислите максимальное ускорение программы  $S^\infty$  с учетом закона Амдала.

65. Покажите, что условия Бернштейна не являются необходимыми условиями коммутативности двух операторов

66. На экзамене по курсу параллельного программирования студент утверждает, что в рамках коммуникатора любой процесс может отправить с помощью функции `MPI_Send()` сообщение любому процессу, в том числе и самому себе. Прав ли студент?

67. Пусть организуется попарный обмен сообщениями в рамках коммуникатора `MPI_COMM_WORLD`, и количество процессов в параллельной программе равно  $N$ . Запишите множество рангов, доступных для использования в качестве рангов процесса-получателя в `MPI_Send()`;

68. Пусть организуется попарный обмен сообщениями в рамках коммуникатора `MPI_COMM_WORLD`, и количество процессов в параллельной программе равно  $N$ . Запишите

множество рангов, доступных для использования в качестве рангов процесса-отправителя в MPI\_Recv().

69. Установите, допускает ли следующий участок кода распараллеливание с помощью директивы среды OpenMP. Если такое распараллеливание возможно, то приведите соответствующую директиву, если невозможно, то объясните, почему.

```
for(int i=0;i<(int)log2(N);i++)
{
  list[i] = 2.0*i*log2(i+1);
  if ((i>=IMIN) && (i<=IMAX)) temp[i] = list[i];
}
```

70. Установите, допускает ли следующий участок кода распараллеливание с помощью директивы среды OpenMP. Если такое распараллеливание возможно, то приведите соответствующую директиву, если невозможно, то объясните, почему.

```
double res = 0.0;
for(int i=0;i<N;i++)
res += list[i]*list[i];
```

71. Установите, допускает ли следующий участок кода распараллеливание с помощью директивы среды OpenMP. Если такое распараллеливание возможно, то приведите соответствующую директиву, если невозможно, то объясните, почему.

```
for(int i=0;i<N;i++)
{
  list[i] = 1.5*exp(i);
  if ( list[i] <= LMAX )
  break;
}
```

## Ответы

59.  $T_p(N) = 80t$

60.  $T_p(N) = 20t$

61.  $T_p(N) = 2^{(30)} t$

62.  $T_p(N) = 2^{(28)} t$

63.  $10p/(p+9)$

64.  $S^\infty = 100$

65. Рассмотрим следующую пару операторов ( $o_1, o_2$ ):  $o_1 : a = a + 1$ ;  $o_2 : a = a - 1$ ;

Ясно, что все три условия Бернштейна в данном случае нарушены. Тем не менее, результат работы программ, содержащих последовательности ( $o_1, o_2$ ) и ( $o_2, o_1$ ), будет одинаков для всех возможных значений переменной  $a$ . Таким образом, условия Бернштейна не являются необходимыми условиями коммутативности двух операторов.

66. Да, в среде MPI процесс коммуникатора может отправить сообщение любому процессу, принадлежащему тому же коммуникатору, в том числе и себе. Тем не менее, указанной возможности в программе желательно избегать, т.к. в данном случае возможно появление ситуации «тупика».

67.  $\text{rank} \in \{0, 1, \dots, N - 1\} \cup \{\text{MPI\_PROC\_NULL}\}$

68.  $\text{rank} \in \{0, 1, \dots, N - 1\} \cup \{\text{MPI\_PROC\_NULL}\} \cup \{\text{MPI\_ANY\_SOURCE}\}$

69. цикл допускает распараллеливание, соответствующая директива:

```
#pragma omp parallel for
```

70. цикл допускает распараллеливание, соответствующая директива:

```
#pragma omp parallel for reduction(+:res)
```

71. наличие оператора break не позволяет воспользоваться директивами распараллеливания;

Для оценивания результатов обучения на зачете используются оценки: «зачтено» и «не зачтено».

Критерии оценивания	Шкала оценок
Средний балл по итогам текущих аттестаций в сумме с баллами на итоговом собеседовании больше 49. Сданы все лабораторные работы.	Зачтено
Средний балл по итогам текущих аттестаций в сумме с баллами на итоговом собеседовании меньше 50 или не сдана хотя бы одна лабораторная работа.	Не зачтено