

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

УТВЕРЖДАЮ

Заведующий кафедрой
цифровых технологий



/ Кургалин С.Д.

28.02.2022 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Б1.В.09 ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ

1. Код и наименование направления подготовки/специальности:

02.03.01 Математика и компьютерные науки

2. Профиль подготовки/специализация:

распределенные системы и искусственный интеллект

3. Квалификация выпускника: бакалавр

4. Форма обучения: очная

5. Кафедра, отвечающая за реализацию дисциплины: 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 Семестр: 3

9. Цели и задачи учебной дисциплины

Дать представление об основных направлениях в развитии высокопроизводительных вычислительных систем, дать обзор средств параллельного программирования, рассмотреть идеи параллельного программирования с помощью интерфейса передачи сообщений, изучить модели функционирования параллельных программ.

10. Место учебной дисциплины в структуре ООП:

Дисциплина относится к вариативной части блока Б1. Для успешного освоения дисциплины необходимо предварительное изучение математического анализа и программирования.

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

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ПК-1	Способен демонстрировать базовые знания математических и естественных наук, основ программирования и информационных технологий.	ПК-1.1	Обладает базовыми знаниями, полученными в области математических и (или) естественных наук, программирования и информационных технологий	Знать: основные технологии распределённых вычислений и обработки данных.
ПК-1	Способен демонстрировать базовые знания математических и естественных наук, основ программирования и информационных технологий.	ПК-1.2	Умеет находить, формулировать и решать стандартные задачи в собственной научно-исследовательской деятельности в математике и информатике	Уметь: применять технологии параллельных вычислений при решении задач профессиональной деятельности.
ПК-1	Способен демонстрировать базовые знания математических и естественных наук, основ программирования и информационных технологий.	ПК-1.3	Имеет практический опыт научно-исследовательской деятельности в математике и информатике	Владеть: навыками использования параллельных алгоритмов для решения практических задач.
ПК-3	Способен создавать и исследовать новые математические модели в естественных науках, промышленности и бизнесе, с учетом возможностей современных информационных технологий и программирования и компьютерной техники.	ПК-3.1	Знает основные методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов, их	Знать: основы проектирования систем распределённых вычислений.

			сопровождения, администрирования и развития (эволюции)	
ПК-3	Способен создавать и исследовать новые математические модели в естественных науках, промышленности и бизнесе, с учетом возможностей современных информационных технологий и программирования и компьютерной техники.	ПК-3.2	Умеет использовать методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного продукта.	Уметь: использовать средства программирования параллельных вычислений с учетом особенностей их реализации.
ПК-3	Способен создавать и исследовать новые математические модели в естественных науках, промышленности и бизнесе, с учетом возможностей современных информационных технологий и программирования и компьютерной техники.	ПК-3.3	Имеет практический опыт применения указанных выше методов и технологий	Владеть: практическими навыками решения вычислительных задач с помощью технологий параллельного программирования.
ПК-4	Способен использовать современные методы разработки и реализации конкретных алгоритмов математических моделей на базе языков программирования и пакетов прикладных программ моделирования.	ПК-4.1	Знает современные методы разработки и реализации алгоритмов математических моделей на базе языков и пакетов прикладных программ моделирования	Знать: методы и средства параллельной обработки информации.
ПК-4	Способен использовать современные методы разработки и реализации конкретных алгоритмов математических моделей на базе языков программирования и пакетов прикладных программ моделирования.	ПК-4.2	Умеет разрабатывать и реализовывать алгоритмы математических моделей на базе языков и пакетов прикладных программ моделирования	Уметь: реализовывать параллельные алгоритмы на базе прикладных программ.
ПК-4	Способен использовать современные методы разработки и реализации конкретных алгоритмов математических моделей на базе	ПК-4.3	Имеет практический опыт разработки и реализации алгоритмов на базе языков и пакетов прикладных программ моделирования	Владеть: программными средствами для реализации параллельных вычислений.

	языков программирования и пакетов прикладных программ моделирования.			
ПК-5	Способен участвовать в разработке технической документации программных продуктов и программных комплексов	ПК-5.1	Знает основные стандарты, нормы и правила разработки технической документации программных продуктов и программных комплексов	Знать: основы норм разработки технической документации для систем, использующих распределённые вычисления.
ПК-5	Способен участвовать в разработке технической документации программных продуктов и программных комплексов	ПК-5.2	Умеет использовать их при подготовке технической документации программных продуктов	Уметь: использовать основные нормы разработки технической документации при проектировании систем, использующих параллельные вычисления.
ПК-5	Способен участвовать в разработке технической документации программных продуктов и программных комплексов	ПК-5.3	Имеет практический опыт подготовки технической документации	Владеть: навыками подготовки технической документации для систем, использующих параллельные вычисления.

12. Объем дисциплины в зачетных единицах/час. — 3/108.

Форма промежуточной аттестации 3 семестр – экзамен.

13. Трудоемкость по видам учебной работы

Вид учебной работы		Трудоемкость	
		Всего	По семестрам
			3 сем.
Аудиторные занятия		48	48
в том числе:	лекции	16	16
	практические	16	16
	лабораторные	16	16
Самостоятельная работа		24	24
в том числе: курсовая работа (проект)			
Форма промежуточной аттестации (экзамен)		36	36
Итого:		108	108

13.1. Содержание дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК*
1. Лекции			
1.1	Введение в высокопроизводительные вычисления	Введение. Роль и значение высокопроизводительных вычислений в современном мире. Производительность вычислительных систем. Закон Амдала.	https://edu.vsu.ru/course/view.php?id=2538
1.2	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. Коммуникаторы и топологии. Производные типы данных. Компиляция и отладка MPI-программ. Система программирования OpenMP. Распределенные вычисления с использованием GRID технологий.	https://edu.vsu.ru/course/view.php?id=2538
1.3	Параллельные алгоритмы	Информационный граф алгоритма. Показатели эффективности параллельного алгоритма. Умножение матрицы на вектор. Матричное умножение. Сортировка. Обработка графов. Задача Дирихле для уравнения Пуассона.	https://edu.vsu.ru/course/view.php?id=2538
1.4	Архитектура параллельных вычислительных систем	Пути повышения производительности процессоров: CISC- и RISC-процессоры. Конвейеризация. Суперскалярная, VLIW, векторная архитектуры. Кэш-память. Многопроцессорные архитектуры с общей и распределенной памятью. Классификация Флинна. Транспьютеры.	https://edu.vsu.ru/course/view.php?id=2538
1.5	Взаимодействующие процессы	Понятия процесса и ресурса. Примитивы создания и завершения процессов. Принципы организации взаимодействия процессов. Задача взаимного исключения. Семафоры. Использование графических способов представления при проектировании и анализе параллельных программ.	https://edu.vsu.ru/course/view.php?id=2538
1.6	Информационно-вычислительные сети	Классификация вычислительных сетей. Характеристики топологии сети. Механизмы передачи данных. Протокол передачи данных. Стандарты на реализации коммуникационной среды. Мониторный узел. Перегрузка сети. Структура распределенных ОС. Проблема тупиков.	https://edu.vsu.ru/course/view.php?id=2538
2. Практические занятия			
2.1	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. Коммуникаторы и топологии. Производные типы данных. Компиляция и отладка MPI-программ. Система программирования OpenMP. Распределенные	https://edu.vsu.ru/course/view.php?id=2538

		вычисления с использованием GRID технологий.	
2.2	Параллельные алгоритмы	Информационный граф алгоритма. Показатели эффективности параллельного алгоритма. Умножение матрицы на вектор. Матричное умножение. Сортировка. Обработка графов. Задача Дирихле для уравнения Пуассона.	https://edu.vsu.ru/course/view.php?id=2538
2.3	Архитектура параллельных вычислительных систем	Пути повышения производительности процессоров: CISC- и RISC-процессоры. Конвейеризация. Суперскалярная, VLIW, векторная архитектуры. Кэш-память. Многопроцессорные архитектуры с общей и распределенной памятью. Классификация Флинна. Транспьютеры.	https://edu.vsu.ru/course/view.php?id=2538
2.4	Взаимодействующие процессы	Понятия процесса и ресурса. Примитивы создания и завершения процессов. Принципы организации взаимодействия процессов. Задача взаимного исключения. Семафоры. Использование графических способов представления при проектировании и анализе параллельных программ.	https://edu.vsu.ru/course/view.php?id=2538
2.5	Информационно-вычислительные сети	Классификация вычислительных сетей. Характеристики топологии сети. Механизмы передачи данных. Протокол передачи данных. Стандарты на реализации коммуникационной среды. Мониторный узел. Перегрузка сети. Структура распределенных ОС. Проблема тупиков.	https://edu.vsu.ru/course/view.php?id=2538

13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (количество часов)				Всего
		Лекции	Практические	Лабораторные	Самостоятельная работа	
1	Введение в высокопроизводительные вычисления	1	0	0	2	3
2	Технологии параллельного программирования	2	2	2	4	10
3	Параллельные алгоритмы	4	4	4	6	18
4	Архитектура параллельных вычислительных систем	4	4	4	4	16
5	Взаимодействующие процессы	2	2	2	4	10
6	Информационно-вычислительные сети	3	4	4	4	15
	Итого:	16	16	16	24	72

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 практических (или лабораторных) занятий, на которых полученные теоретические знания применяются для решения конкретных практических задач. Обучающимся для успешного освоения дисциплины рекомендуется вести конспект лекций и практических (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 практических (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. Больше 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

а) основная литература:

№ п/п	Источник
1	Стуколов, С. В. Параллельное программирование. Практикум : учебное пособие / С. В. Стуколов. — Кемерово : КемГУ, 2020. — 273 с. — ISBN 978-5-8353-2723-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/173547
2	Борзунов, С.В. Суперкомпьютерные вычисления: практический подход / С.В. Борзунов, С.Д. Кургалин. – СПб. : БХВ, 2019. – 256 с. – (Серия: Учебная литература для вузов). – ISBN 978-5-9909805-2-5. – URL: https://elibrary.ru/item.asp?id=36628828

б) дополнительная литература:

№ п/п	Источник
1	Филатов, А. С. Параллельное программирование : учебное пособие / А. С. Филатов. — Москва : РТУ МИРЭА, 2021. — 46 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/218429
2	Парфенов, Д. В. Параллельные и распределенные вычисления : учебное пособие / Д. В. Парфенов, Д. А. Петрусевич. — Москва : РТУ МИРЭА, 2022. — 92 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/265658

в) информационные электронно-образовательные ресурсы (официальные ресурсы интернет)*:

№ п/п	Ресурс
1	ЗНБ ВГУ: https://lib.vsu.ru/
2	Электронно-библиотечная система "Университетская библиотека online": http://biblioclub.ru/
3	Электронно-библиотечная система "Лань": https://e.lanbook.com/

16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п	Источник
1	Стуколов, С. В. Параллельное программирование. Практикум : учебное пособие / С. В. Стуколов.

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.

Компьютерный класс: специализированная мебель, персональные компьютеры, мультимедийный проектор, экран.

ПО: ОС Windows v.7, 8, 10, Интерпретатор языка CPython, дистрибутив Anaconda, IDE PyCharm, редактор Jupyter.

19. Оценочные средства для проведения текущей и промежуточной аттестаций

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

№ п/п	Наименование раздела дисциплины (модуля)	Компетенции	Индикаторы достижения компетенции	Оценочные средства
1.	Разделы 1-6	ПК-1	ПК-1.1 ПК-1.2 ПК-1.3	Письменный опрос Лабораторные работы Лабораторные работы
2.	Разделы 1-6	ПК-3	ПК-3.1 ПК-3.2 ПК-3.3	Письменный опрос Лабораторные работы Лабораторные работы
3.	Разделы 1-6	ПК-4	ПК-4.1 ПК-4.2 ПК-4.3	Письменный опрос Лабораторные работы Лабораторные работы
4.	Разделы 1-6	ПК-5	ПК-5.1 ПК-5.2 ПК-5.3	Письменный опрос Лабораторные работы Лабораторные работы
Промежуточная аттестация форма контроля – экзамен				Перечень вопросов для письменного опроса

20. Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

20.1. Текущий контроль успеваемости

Контроль успеваемости по дисциплине осуществляется с помощью лабораторных работ.

Перечень лабораторных работ

1. Табулирование функций.
2. Компиляция и отладка MPI-программ.
3. Основные функции MPI. Процедуры Send, Recv.
4. Основные функции MPI. Процедуры Bcast, Scatter, Scatterv.
5. Основные функции MPI. Процедуры Gather, Gatherv, Alltoall..
6. Основные функции MPI. Процедуры Reduce, AllReduce, Scan
7. Основные функции MPI. Процедуры Bcast, Scatter, Scatterv.
8. Базовые параллельные алгоритмы. Задача о сумме.
9. Базовые параллельные алгоритмы. Задача о префиксе.

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 функции с использованием OpenMP. 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: перечень вопросов для письменного опроса.

Перечень вопросов для письменного опроса

Раздел 1. Введение в высокопроизводительные вычисления

1. Современные концепции и средства параллельного программирования.
2. Принципы программирования для ЭВМ с общей и распределенной памятью.
3. Роль и значение высокопроизводительных вычислений в современном мире.
4. Производительность вычислительных систем.
5. Закон Амдала.

Раздел 2. Технологии параллельного программирования

1. Структура MPI-программы.
2. Сообщения, их передача и прием.
3. Синхронное и асинхронное взаимодействие.
4. Коллективный обмен данными.
5. Виды коллективного обмена, барьеры, широковещательная рассылка данных. Коммуникаторы и топологии.
6. Производные типы данных.
7. Компиляция и отладка MPI-программ.
8. Система программирования OpenMP.

Раздел 3. Параллельные алгоритмы

1. Информационный граф алгоритма.
2. Показатели эффективности параллельного алгоритма.
3. Умножение матрицы на вектор.
4. Матричное умножение.
5. Сортировка.

Раздел 4. Архитектура параллельных вычислительных систем

1. Пути повышения производительности процессоров: CISC- и RISC-процессоры.
2. Конвейеризация.
3. Многопроцессорные архитектуры с общей и распределенной памятью.
4. Классификация Флинна.

Раздел 5. Взаимодействующие процессы

1. Понятия процесса и ресурса.
2. Примитивы создания и завершения процессов.
3. Принципы организации взаимодействия процессов. Задача взаимного исключения.
4. Семафоры.

Раздел 6. Информационно-вычислительные сети

1. Классификация вычислительных сетей.
2. Характеристики топологии сети.
3. Механизмы передачи данных.
4. Проблема тупиков.

Требования к студентам при проведении промежуточной аттестации

Для оценивания результатов обучения на экзамене с оценкой используется 4-балльная шкала: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
Дан полный, развёрнутый ответ на поставленный вопрос (вопросы), обучающийся свободно оперирует основными понятиями дисциплины, ориентируется в предметной области. Изложение материала не содержит ошибок, отличается последовательностью, грамотностью, логической стройностью.	Повышенный уровень	Отлично
Дан развёрнутый ответ на поставленный вопрос (вопросы), обучающийся свободно оперирует основными понятиями дисциплины, ориентируется в предметной области. Материал изложен в целом последовательно и грамотно, отсутствуют грубые ошибки, однако имеются отдельные неточности в	Базовый уровень	Хорошо

определениях, вычислениях, доказательствах, изложениях положений теории.		
Ответ на поставленный вопрос (вопросы) содержит изложение только базового теоретического материала, имеются ошибки в определениях, вычислениях, доказательствах, формулировках положений теории. Нарушена логическая последовательность в изложении материала.	Пороговый уровень	Удовлетворительно
Ответ на поставленный вопрос (вопросы) отсутствует, либо содержит грубые ошибки в определениях, вычислениях, доказательствах, формулировках положений теории. Обучающийся не владеет основными понятиями дисциплины. Отсутствует логическая последовательность в изложении материала.	–	Неудовлетворительно

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

Задания с выбором ответа

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 коммутатору**
- d) не менее двух процессов

8. Для выполнения широковещательной рассылки сообщений применяется:

- a) **MPI_Bcast**
- b) MPI_Bsend
- c) MPI_Send

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

12. Подпрограмма MPI_Gather:

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

13. Подпрограмма MPI_Scatter:

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

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

- a) **MPI_Gather**
- b) **MPI_Gatherv**
- c) MPI_Allgather

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**

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

- a) **MPI_Type_struct**
- b) MPI_Type_create
- c) MPI_Comm_create

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

21. Для создания коммуникатора предусмотрена команда

- a) MPI_Comm_size
- b) **MPI_Comm_create**
- c) MPI_Comm_rank

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

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»

29. Неблокирующий прием в MPI выполняется :

- a) **MPI_Irecv**
- b) MPI_Recv
- c) MPI_Reduce_Scatter

30. Подпрограмма MPI_Wait предназначена для:

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

31. Подпрограмма MPI_Test предназначена для:

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

32. Подпрограмма MPI_Testall выполняет проверку:

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

33. Подпрограмма MPI_Testany выполняет проверку:

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

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

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**

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 кратким ответом

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

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

3. Какой аббревиатурой в классификации Флинна обозначается класс вычислительных систем, поддерживающих работу нескольких потоков команд, одного потока данных?
4. Какой аббревиатурой в классификации Флинна обозначается класс вычислительных систем, поддерживающих работу нескольких потоков команд, нескольких потоков данных?
5. 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?
6. Как называется связный ациклический граф?
7. Какой аббревиатурой принято обозначать модель вычислительной системы – машину с произвольным доступом к памяти?
8. Какой аббревиатурой принято обозначать модель вычислительной системы – параллельную машину с произвольным доступом к памяти?
9. Как называется величина, равная отношению $T1(N)/Tp(N)$, где p – количество вычислительных узлов?
10. Как называется величина, равная отношению $T1(N)/(p \cdot Tp(N))$?
11. Величина, равная произведению времени параллельного решения задачи и числа используемых процессоров называется «... вычислений». Заполните пропуск.
12. Как называется упорядоченный по времени набор векторов, образованных всеми данными в памяти вычислительной системы?
13. Какую аббревиатуру используют для обозначения отдельного устройства вычислительной системы, способное выполнять графический рендеринг, а также высокопроизводительные вычисления наряду с центральным процессором?

Ответы:

1. SISD
2. SIMD
3. MISD
4. SISD
5. модель передачи сообщений
6. дерево
7. RAM
8. PRAM
9. ускорение
10. эффективность
11. стоимость
12. история содержимого памяти
13. GPU

Задания с развернутым ответом

1. Время работы последовательной версии некоторого алгоритма A равно $T1(N) = 2N \log_2(N)$ т, где N — размер входных данных, т — время выполнения одной вычислительной операции. В предположении, что алгоритм допускает максимальное распараллеливание, т.е. время работы на вычислительной системе с p процессорами равно $Tp(N) = T1(N) / p$, вычислите время работы алгоритма A в следующем случае: $N = 32$, $p = 4$.
2. Время работы последовательной версии некоторого алгоритма A равно $T1(N) = 2N \log_2(N)$ т, где N — размер входных данных, т — время выполнения одной вычислительной операции. В предположении, что алгоритм допускает максимальное распараллеливание, т.е. время работы на вычислительной системе с p процессорами равно $Tp(N) = T1(N) / p$, вычислите время работы алгоритма A в следующем случае: $N = 32$, $p = 16$.
3. Время работы последовательной версии некоторого алгоритма B равно $T1(N) = 2^N$ т, где N — размер входных данных, т — 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: $N = 32$, $p = 4$.

4. 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: $N = 32$, $p = 16$.

5. 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.

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

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

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

9. Пусть организуется попарный обмен сообщениями в рамках коммутатора `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()`;

10. Пусть организуется попарный обмен сообщениями в рамках коммутатора `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()`.

11. 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];
}
```

12. 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];
```

13. 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

1. $Tp(N) = 80t$

2. $Tp(N) = 20t$

3. $Tp(N) = 2^{30}t$

4. $Tp(N) = 2^{28}t$

5. $10p/(p+9)$

6. $S_{\infty} = 100$

7. Рассмотрим следующую пару операторов ($o1, o2$): $o1 : a = a + 1$; $o2 : a = a - 1$;

Ясно, что все три условия Бернштейна в данном случае нарушены. Тем не менее, результат работы программ, содержащих последовательности ($o1, o2$) и ($o2, o1$), будет одинаков для всех возможных значений переменной a . Таким образом, условия Бернштейна не являются необходимыми условиями коммутативности двух операторов.

8. 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».

9. $rank \in \{0, 1, \dots, N - 1\} \cup \{MPI_PROC_NULL\}$

10. $rank \in \{0, 1, \dots, N - 1\} \cup \{MPI_PROC_NULL\} \cup \{MPI_ANY_SOURCE\}$

11. цикл допускает распараллеливание, соответствующая директива:

```
#pragma omp parallel for
```

12. цикл допускает распараллеливание, соответствующая директива:

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

13. наличие оператора break не позволяет воспользоваться директивами распараллеливания.

Критерии оценивания	Баллы
Имеется верная последовательность всех этапов решения, обоснованно получен верный ответ.	3
Получен неверный ответ из-за вычислительной ошибки, при этом имеется верная последовательность всех этапов решения.	2
Получен верный ответ, однако имеются пропуски одного или двух этапов решения ИЛИ Решение не завершено, однако верно выполнен хотя бы один из этапов решения.	1
Решение не соответствует ни одному из критериев, перечисленных выше.	0