

Белорусский государственный университет

**УТВЕРЖДАЮ**

Проректор по учебной работе  
и образовательным инновациям

  
О.Г. Прохоренко

01 декабря 2023 г.

Регистрационный № УД –12628/уч.

*Распределенное и параллельные системы*

**Учебная программа учреждения высшего образования  
по учебной дисциплине для специальности:**

1-31 03 07 Прикладная информатика (по направлениям)

2023 г.

Учебная программа составлена на основе ОСВО 1-31 03 07-2021; типового учебного плана №G31-1-030 тип. от 01.07.2021, учебного плана G31-1-034уч. 23.07.2021.

**СОСТАВИТЕЛИ:**

**О. Г. Казанцева** – старший преподаватель кафедры многопроцессорных систем и сетей факультета прикладной математики и информатики Белорусского государственного университета;

**О. М. Кондратьева** – старший преподаватель кафедры многопроцессорных систем и сетей факультета прикладной математики и информатики Белорусского государственного университета;

**РЕЦЕНЗЕНТЫ:**

**Н.В. Лапицкая** – заведующая кафедрой программного обеспечения информационных систем Учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат технических наук, доцент;

**Е.А. Левчук** – доцент кафедры технологий программирования факультета прикладной математики и информатики, кандидат технических наук.

**РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:**

Кафедрой многопроцессорных систем и сетей  
(протокол № 5 от 20.11.2023);

Научно-методическим советом БГУ  
(протокол № 3 от 30.11.2023)

Заведующий кафедрой



С. В. Марков

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

### Цели и задачи учебной дисциплины

**Цель** преподавания учебной дисциплины «Распределённые и параллельные системы» – изучение принципов и технологий распределенных вычислений, знакомство с различными классами распределенных систем и получение практических навыков разработки распределенных приложений.

Учебная дисциплина «Распределённые и параллельные системы» входит в цикл дисциплин подготовки специалиста в области разработки высокоэффективных вычислений на современных вычислительных системах. Она представляет собой комплекс знаний, умений и навыков, позволяющих овладеть основами параллельных вычислений на современных многопроцессорных компьютерах с общей и распределенной памятью, а также практическими навыками разработки распределенных программ, программ для многопроцессорных систем.

Учебная программа предусматривает изучение вопросов построения и организации распределенных и параллельных систем. Содержание лекций ориентировано на подготовку студентов к практическому использованию изложенного материала, формирование у них широкого кругозора в области информационных технологий.

### Задачи учебной дисциплины

1. Сформировать такие фундаментальные понятия как распределенная система, распределенные вычисления.
2. Изучить методы распараллеливания вычислений с использованием мощных вычислительных систем с распределенной и общей памятью.
3. Изучить методы проектирования распределенных систем.
4. Уметь разрабатывать многопоточные приложения.
5. Уметь проектировать распределённые системы, обеспечивая базовые гарантии, которые обеспечивают доступность и надёжность распределённых систем.

### Место учебной дисциплины

В системе подготовки специалиста с высшим образованием для специальности 1-31 03 07 «Информатика» учебная дисциплина **относится к модулю «Прикладное программирование»** компонента учреждения высшего образования.

Учебная программа составлена с учетом межпредметных **связей** и программ по дисциплинам.

Основой для изучения учебной дисциплины являются дисциплины государственного компонента «Основы и методологии программирования» и «Промышленное программирование» модуля «Программирование», дисциплины государственного компонента «Архитектура компьютеров», «Операционные системы», «Компьютерные сети» модуля «Информатика и

компьютерные системы», дисциплины компоненты учреждения высшего образования «Web-программирование» модуля «Прикладное программирование».

Знания, полученные в учебной дисциплине, могут быть использованы при подготовке курсовой и выпускной квалификационной работ, и являются необходимыми для успешной работы в сфере информационных технологий.

### **Требования к компетенциям**

Освоение учебной дисциплины «Распределённые и параллельные системы» должно обеспечить формирование следующей **специальной компетенции:**

СК-5. Использовать программные средства и технологии для создания прикладного программного обеспечения.

В результате освоения учебной дисциплины студент должен:

#### **знать:**

- архитектуру распределённых и параллельных приложений;
- особенности различных классов распределённых систем и используемые при их реализации подходы;
- общие принципы проектирования распределённых и параллельных приложений;
- знать основные техники распараллеливания программ;
- общие принципы построения распределённых систем, модели программирования и методы решения типовых задач;
- проблематику распределённых вычислений, области применения и виды распределённых систем.

#### **уметь:**

- выбирать технологии проектирования в соответствии с задачей и имеющимся оборудованием и обосновывать выбор метода решения поставленной задачи;
- проектировать параллельные и распределённые приложения.

#### **владеть:**

- основными приемами проектирования и технологиями разработки параллельных и распределённых приложений;
- способами оценки эффективности созданных приложений.

### **Структура учебной дисциплины**

Дисциплина изучается в 6-м семестре. Всего на изучение учебной дисциплины «Распределённые и параллельные системы» отведено:

– в очной форме получения высшего образования: 216 часов, в том числе 102 аудиторных часа, из них: лекции – 68 часа, лабораторные занятия – 30 часов, УСП – 4 часа.

Трудоемкость учебной дисциплины составляет 6 зачетных единиц.

Форма промежуточной аттестации – зачет, экзамен.

# СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

## Раздел 1. Распределённые системы

### Тема 1.1 Основные понятия

Определения распределенных систем. Области применения и примеры. Типы распределенных систем, их отличия и особенности. Архитектурные стили распределенных систем (многоуровневая архитектура, объектно-ориентированные и сервис-ориентированные архитектуры, ресурсные архитектуры, архитектура публикация-подписка). Системная (программная) архитектура распределенных систем (централизованные организации, гибридные архитектуры). Примеры архитектур.

### Тема 1.2. Взаимодействие между процессами в распределенных системах. Протоколы

Возможные разновидности взаимодействий. Передача сообщений между парой процессов. Схема «запрос-ответ» и удаленные вызовы процедур (RPC).

Многоуровневые протоколы. Протокол HTTP. Эволюция протокола HTTP. Веб-сервисы, архитектура REST, сравнение с RPC.

Групповые взаимодействия. Надежная рассылка сообщений в группе. Масштабируемые подходы к распространению информации.

Непрямое взаимодействие. Очередь сообщений. Издатель-подписчик. Распределенная общая память. Пространство кортежей.

### Тема 1.3. Именованное пространство и поиск

Виды имен, схемы именования, разрешение имени. Domain Name System (DNS). Иерархический подход со структурированным пространством имен. Distributed Hash Table (DHT). Децентрализованный подход с плоским пространством имен. Именованное пространство на основе атрибутов.

### Тема 1.4. Масштабирование

Масштабируемость и масштабирование. Техники масштабирования и связанные задачи. Репликация stateless-сервиса и балансировка нагрузки. Кэширование. Разбиение stateful-сервиса по данным (шардинг).

### Тема 1.5. Репликация данных и согласованность. Координация

Репликация данных и согласованность. Репликация данных, известные подходы к реализации. Согласованность, связь с репликацией, известные модели.

Координация. Время, часы и порядок событий. Физические часы. Логические часы. Часы Лэмпорта. Векторные часы. Применение логических часов. Упорядоченная рассылка. Обнаружение конфликтов при репликации данных.

### Тема 1.6. Параллельная обработка

Параллельные вычисления. Параллельная обработка запросов. Параллельная обработка данных.

### Тема 1.7. Безопасность в распределенных системах

Требования к безопасности в РС. Базовые техники и механизмы. Шифрование. Аутентификация. Цифровая подпись. Управление ключами. Авторизация.

### **Тема 1.8. Отказоустойчивость**

Отказоустойчивость. Обнаружение отказов. Сбои и отказы. Типы отказов и модели РС. Детектор отказов. Реализации детектора отказов.

Консенсус и связанные задачи. Примеры практических задач. Возможные подходы и фундаментальные примитивы. Задача консенсуса. Алгоритм консенсуса Raft. Сервисы для координации.

Произвольные (византийские) отказы. Устойчивость к произвольным отказам. Задача о византийских генералах. Протокол Practical Byzantine Fault Tolerance (PBFT). Консенсус в сети Bitcoin.

## **Раздел 2. Параллельные вычисления**

### **Тема 2.1. Введение в параллельные вычисления**

Параллельная обработка данных. Большие задачи. Способы создания параллельных алгоритмов и программ, сложность. Средства разработки параллельных программ.

Параллельное программирование как способ повышения эффективности программы. Многопоточные программы как параллельные программы: общие проблемы и общие решения. Параллельность и конкурентность.

### **Тема 2.2. Параллельные программы**

Модели создания и функционирования потоков: делегирования, с равноправными узлами, конвейер, изготовитель-потребитель.

Классические задачи параллельного программирования.

Исследование распараллеливания: проведение экспериментов, анализ эффективности использования параллелизма, теоретические оценки.

### **Тема 2.3. Как распараллеливать программу**

Что такое распараллеливание?

Методология проектирования параллельных программ: разделение вычислений на независимые части, выделение информационных зависимостей, масштабирование набора подзадач, распределение подзадач между вычислительными элементами.

Декомпозиция задачи на подзадачи: требования, параллелизм по данным, функциональный параллелизм.

Параллелизм, асинхронность, зависимости и недетерминизм.

### **Тема 2.4. Технологии параллельного программирования**

Средства разработки параллельных программ: коммуникационные интерфейсы, параллельные языки и расширения языков, специализированные библиотеки, средства автоматического распараллеливания, инструментальные системы, специализированные прикладные пакеты.

Операционные системы. Поддержка разработки параллельных программ, использующих модель общей памяти. Встроенные потоки Windows и Unix: достоинства и недостатки.

Механизмы синхронизации. Ошибки параллельного программирования.

### **Тема 2.5. Разработка многопоточных приложений**

Поддержка многопоточности, встроенная в языки высокого уровня C++ и Java.

Управление потоками. Разделение данных между потоками. Синхронизация параллельных операций.

Потокобезопасные структуры данных.

Поддержка многопоточности на разных уровнях абстракции.

### **Тема 2.6. Шаблоны параллельного программирования**

Использование шаблонов как методика разработки параллельных программ.

Распространенные шаблоны параллельного программирования: MapReduce, Fork-Join, Producer-Consumer, Pipeline, Task Parallelism, Master-Worker.

### **Тема 2.7. Программный интерфейс OpenMP**

Назначение и компоненты OpenMP. Особенности модели памяти OpenMP.

Конструкции для создания потоков, распределения работы между потоками, управления работой с данными, синхронизации потоков. Процедуры библиотеки поддержки времени выполнения. Переменные окружения.

Поддержка шаблонов параллельного программирования.

## **Раздел 3. Инструменты для разработки приложений**

### **Тема 3.1. Версионный контроль**

Отслеживание изменений кода. Эволюция и знаковые систем контроля версий. Обзор и введение Git. Инспекция кода (code review). Рабочие процессы при версионировании (workflow).

### **Тема 3.2. Журналирование и тестирование**

Журналирование событий. Уровни журналирования, форматы сообщений, направления вывода сообщений.

Модульное тестирование. Особенности тестирования многопоточный приложений.

### **Тема 3.3. Практики DevOps. Непрерывная интеграция (CI). Непрерывная поставка и развертывание (CD)**

Типовые проблемы при совместной разработке. Конвейер (pipeline) и основные этапы. Лучшие практики. Популярные инструменты. Виды

артефактов. Введение в Docker: запуск приложений в контейнерах, построение образов. Введение в Jenkins.

Определение, цели и задачи. Конвейер поставки. Основные трудности. Требования к приложениям. Версионирование артефактов. Решение проблем с изменением окружений. Обновление и откат приложений. Развёртывание приложений и доступ через сервисы.

## УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Очная форма получения высшего образования с применением дистанционных образовательных технологий (ДОТ)

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов УСП	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
	2	3	4	5	6		8	9
<b>1</b>	<b>Распределённые системы</b>	<b>30</b>			<b>12</b>			
1.1	Основные понятия	4			2			Отчёт
1.2	Взаимодействие между процессами в распределённых системах. Протоколы	4			2			Отчёт. Коллоквиум
1.3	Именованье и поиск	2						Контрольная работа №1
1.4	Масштабирование	4			2			Отчёт
1.5	Репликация данных и согласованность. Координация	4			2			Отчёт
1.6	Параллельная обработка	2						Дискуссия, тест
1.7	Безопасность в распределённых системах	4			2			Дискуссия. Отчёт.
1.8	Отказоустойчивость	6			2			Отчёт. Контрольная работа №2
<b>2</b>	<b>Параллельные вычисления</b>	<b>34</b>			<b>16</b>		<b>2</b>	
2.1	Введение в параллельные вычисления	2			2			Отчёт
2.2	Параллельные программы	6			2			Отчёт
	Как распараллеливать программу	6			2			Отчёт.

2.3								Контрольная работа №3
2.4	Технологии параллельного программирования	4			2			Отчёт
2.5	Разработка многопоточных приложений	8			4			Отчёт. Коллоквиум
2.6	Шаблоны параллельного программирования	4			2			Отчёт. Контрольная работа №4
2.7	Программный интерфейс OpenMP	4			2		2	Отчёт. Собеседование
<b>3</b>	<b>Инструменты для разработки приложений</b>	<b>4</b>			<b>2</b>		<b>2</b>	Отчёт
3.1	Версионный контроль						1	Собеседование
3.2	Журналирование и тестирование	2			2			Отчёт
3.3	Практики DevOps. Непрерывная интеграция (CI). Непрерывная поставка и развертывание (CD)	2					1	Отчёт. Собеседование

## ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

### Перечень основной литературы

1. Стин ван М. Распределенные системы / пер. с англ. В. А. Яроцкого / М. Стин ван, Э. С. Таненбаум – М.: ДМК Пресс, 2021. – 583 с.
2. Таненбаум, Э. С. Современные операционные системы / Э. Таненбаум, Х. Бос ; [пер. с англ.: А. Леонтьева, М. Малышева, Н. Вильчинский]. – 4-е изд. – Санкт-Петербург [и др.] : Питер, 2020. – 1119 с. - <https://ibooks.ru/bookshelf/364626/reading>.
3. Гергель, В. П. Теория и практика параллельных вычислений : учеб. пособие / В. П. Гергель. – Москва : БИНОМ. Лаборатория знаний : Интернет-Университет Информационных Технологий, 2017. – 423 с. – <https://ibooks.ru/reading.php?short=1&productid=363224>.
4. Уильямс, Энтони. С++. Практика многопоточного программирования = С++. Concurrency in Action / Энтони Уильямс ; [пер. с англ. Н. Вильчинский]. – 2-е изд. – Санкт-Петербург [и др.] : Питер, 2020. – 640 с. – URL: <https://ibooks.ru/bookshelf/371682>.
5. Роби, Р. Параллельные и высокопроизводительные вычисления / Роберт Роби и Джулиана Замора ; [пер. с англ. А. В. Логунова]. – Москва : ДМК Пресс, 2022. – 799 с.

### Перечень дополнительной литературы

1. Coulouris G. Distributed Systems: Concepts and Design / G. Coulouris, J. Dollimore, T. Kindberg. – Изд. 3-е. – Addison-Wesley, 2012. – 1047 с.
2. Jeff Dean, Designs, Lessons and Advice from Building Large Distributed Systems, The 3rd ACM SIGOPS International Workshop on Large Scale Distributed Systems and Middleware (2009), Keynote #3, [Электронный ресурс]. – Режим доступа: <https://research.cs.cornell.edu/ladis2009/talks/dean-keynote-ladis2009.pdf>. – Дата доступа: 02.09.2023.
3. Lesson: Concurrency (The Java&trade; Tutorials > Essential Java Classes) [Электронный ресурс]. – Режим доступа: <https://docs.oracle.com/javase/tutorial/essential/concurrency/> . – Дата доступа: 02.09.2023.
4. Бёрнс, Б. Распределенные системы. Паттерны проектирования = Designing Distributed Systems. Patterns and Paradigms for Scalable, Reliable Services / Брендан Бёрнс ; [пер. с англ. К. Русецкого]. – Санкт-Петербург [и др.] : Питер, 2019. – 222 с. – URL: <https://ibooks.ru/bookshelf/361843/reading>.
5. Блинов, И. Н. Java from EPAM : учеб.-метод. пособие / И. Н. Блинов, В. С. Романчик. – Минск: Четыре четверти, 2021. – 560 с.
6. Галовиц, Я. С++17 STL. Стандартная библиотека шаблонов / Я. Галовиц. – СПб.: Питер, 2018. – 432 с.
7. Гетц, Б. Java Concurrency на практике / Б. Гетц, Т. Пайерлс, Дж. Блох, Дж. Боубер, Д. Холмс, Д. Ли. – СПб.: Питер, 2020. – 464 с.
8. Гримм, Р. Параллельное программирование на современном языке С++ / Р. Гримм. – М.: ДМК Пресс, 2022. – 616 с.

9. Клеппман М. Высоконагруженные приложения. Программирование, масштабирование, поддержка – СПб.: Питер, 2018. – 640 с.

10. Компьютерные сети: принципы, технологии, протоколы: учеб. пособие для вузов / В. Г. Олифер, Н. А. Олифер. – Изд. 4-е. – СПб.: Питер, 2015. – 943 с.

11. Страуструп, Б. Язык программирования C++. Краткий курс / Б. Страуструп. – СПб.: ООО «Диалектика», 2019. – 320 с.

12. Террелл, Р. Конкурентность и параллелизм на платформе .NET. Паттерны эффективного проектирования / Р. Террелл. – СПб.: Питер, 2019. – 624 с.

13. Федотов, И.Е. Параллельное программирование. Модели и приемы / И.Е.Федотов. – М.: СОЛОН-Пресс, 2018. – 390 с.

14. Фоккинг, У. Распределенные алгоритмы. Интуитивный подход / У. Фоккинг. – СПб.: Питер, 2017. – 272 с.

15. Хьюз, К. Параллельное и распределенное программирование с использованием C++ / К. Хьюз, Т. Хьюз. – М.: Вильямс, 2004. – 672 с.

#### Электронные ресурсы

16. Образовательный портал БГУ [Электронный ресурс]. – Режим доступа: <https://edufpmi.bsu.by/course/view.php?id=945>. – Дата доступа: 02.09.2023.

#### Перечень рекомендуемых средств диагностики и методика формирования итоговой отметки

Объектом диагностики компетенций студентов являются знания, умения, полученные ими в результате изучения учебной дисциплины. Выявление учебных достижений студентов осуществляется с помощью мероприятий текущего контроля и промежуточной аттестации.

Для диагностики компетенций в рамках учебной дисциплины рекомендуется использовать следующие формы:

1. Устная форма: собеседование, дискуссия, коллоквиум.
2. Письменная форма: контрольные работы.
3. Устно-письменная форма: отчёты по лабораторным работам с их устной защитой.
4. Техническая форма: электронные тесты, лабораторные работы, выполняемые на компьютере (оцениваются исходя из читаемости и оптимизированности программного кода, а также путём модульного тестирования программного кода).

В качестве рекомендуемых технических средств диагностики используется Образовательный портал <https://edufpmi.bsu.by>.

Формой промежуточной аттестации по дисциплине «Распределенные и параллельные системы» учебными планами предусмотрены **зачет** и **экзамен**.

**Зачет по дисциплине** проходит в устной и/или письменной форме в виде *защиты отчётов по лабораторным работам*. В случае успешной

защиты отчётов **по всем лабораторным работам** допускается получение зачета без проведения дополнительного опроса (это решение находится в компетенции преподавателей, которые проводят учебные занятия по данной дисциплине). При этом явка обучающегося на зачет является обязательной.

Итоговая отметка формируется на основе Положения о рейтинговой системе оценки знаний обучающихся по учебной дисциплине в БГУ (Приказ ректора БГУ от 31.03.2020 № 189-ОД).

Рейтинговая система предусматривает использование весовых коэффициентов в ходе проведения контрольных мероприятий текущей аттестации.

Примерные весовые коэффициенты, определяющие вклад текущей аттестации в отметку при прохождении промежуточной аттестации:

Формирование отметки за текущую аттестацию:

- отчет по лабораторным работам с их устной защитой – 35 %;
- выполнение теста – 10%;
- контрольные работы – 45 %;
- коллоквиум – 10 %.

Итоговая отметка по дисциплине рассчитывается на основе отметки текущей аттестации (рейтинговой системы оценки знаний) – 40% и экзаменационной отметки – 60%.

### **Примерная тематика лабораторных занятий**

Занятие 1. Модель распределенной системы. Проектирование архитектуры распределенного приложения.

Занятие 2. Журналирование и тестирование приложений.

Занятие 3. Протокол HTTP. Веб-серверы и веб-сервисы.

Занятие 4. Высоконагруженные сервисы и балансировка нагрузки.

Занятие 5. Репликация сервисов с состоянием, которое может изменяться в результате запросов клиентов.

Занятие 6. Методы обеспечения отказоустойчивости распределенных систем.

Занятие 7. Обеспечение безопасности распределенных систем.

Занятие 8. Поддержка многопоточности в Java. Многопоточная программа как параллельная программа. Разделение задачи на независимые части. Проведение вычислительных экспериментов.

Занятие 9. Поддержка многопоточности в Java. Модели создания и функционирования потоков. Модель делегирования.

Занятие 10. Поддержка многопоточности в Java. Модели создания и функционирования потоков. Модель производитель-потребитель.

Занятие 11. Поддержка многопоточности в Java. Этапы разработки параллельных алгоритмов. Методика проектирования Яна Фостера. Java.util.concurrent и Fork-Join Framework.

Занятие 12. Поддержка многопоточности в C++. Создание/завершение потоков. Совместное использование данных. Средства для работы на уровне заданий.

Занятие 13. Поддержка многопоточности в C++. Потокбезопасные структуры данных.

Занятие 14. Шаблоны параллельного программирования.

Занятие 15. Технология OpenMP. Распараллеливание циклов, параллельные секции, ограничения.

### **Примерный перечень заданий для управляемой самостоятельной работы студентов**

#### **Тема № 2.7. Технология OpenMP. Поддержка шаблонов параллельного программирования. (2 часа)**

##### **Перечень вопросов:**

1. Шаблон Loop Parallelism.
2. Шаблон Fork-Join.
3. Шаблон SPMD.
4. Шаблон Divide and Conquer.

##### **Примерный перечень заданий:**

1. Реализовать конкретный шаблон проектирования для некоторой задачи.
2. Выполнить вычислительные эксперименты для определения эффективности реализации.

Форма контроля: отчет, собеседование.

#### **Тема № 3.1. Версионный контроль (1 часа)**

##### **Перечень вопросов:**

1. Отслеживание изменений кода. Эволюция и знаковые систем контроля версий.
2. Обзор и введение Git. Инспекция кода (code review).
3. Рабочие процессы при версионировании (workflow).

##### **Примерный перечень заданий:**

1. Создать репозиторий для командной разработки проекта.
2. Выполнить инспекцию кода.

Форма контроля: собеседование.

#### **Тема № 3.3. Практики DevOps. Непрерывная интеграция (CI). Непрерывная поставка и развертывание (CD) (1 час)**

##### **Перечень вопросов:**

1. Типовые проблемы при совместной разработке.
2. Конвейер (pipeline) и основные этапы.
3. Популярные инструменты. Виды артефактов.

4. Введение в Docker: запуск приложений в контейнерах, построение образов.

#### **Примерный перечень заданий:**

1. Разработать скрипты для CI/CD.
2. Выполнить развертывание разрабатываемого распределенного приложения.

Форма контроля: Отчёт по лабораторной работе. Собеседование.

#### *Рекомендуемая тематика контрольных работ и коллоквиума:*

1. Контрольная работа № 1 «Именование и поиск: иерархический подход, децентрализованный подход, именование на основе атрибутов».
2. Контрольная работа № 2 «Отказоустойчивость».
3. Контрольная работа № 3 «Методика проектирования параллельных программ».
4. Контрольная работа № 4. «Реализация конкретного шаблона параллельного программирования на языке высокого уровня для некоторой задачи».
5. Коллоквиум (темы 1.2, 2.5) «Методы взаимодействия между процессами и потоками».

#### **Описание инновационных подходов и методов к преподаванию учебной дисциплины**

При организации образовательного процесса используются следующие методы:

– **метод учебной дискуссии**, который предполагает участие студентов в целенаправленном обмене мнениями, идеями для предъявления и/или согласования существующих позиций по определенной проблеме. Использование метода обеспечивает появление нового уровня понимания изучаемой темы, применение знаний (теорий, концепций) при решении проблем, определение способов их решения.

– **метод группового обучения**, который представляет собой форму организации учебно-познавательной деятельности обучающихся, предполагающую функционирование разных типов малых групп, работающих как над общими, так и специфическими учебными заданиями.

В качестве технических средств для организации работы в рамках учебной дисциплины рекомендуется использовать Образовательный портал БГУ (<https://edufpmi.bsu.by>) – инструмент с эффективной функциональностью контроля, тренинга и самостоятельной работы, систему контроля версий Git – распределенная система контроля версий, которая позволяет разработчикам отслеживать изменения в коде с течением времени. Она хранит все версии проекта в репозитории, предоставляя пользователям возможность переходить от одной версии к другой и позволяет нескольким программистам работать над одним проектом одновременно.

Для контроля самостоятельности выполнения работ рекомендуется использовать автоматические системы определения несанкционированных материалов, функционирующие в Insight Runner.

### **Методические рекомендации по организации самостоятельной работы обучающихся**

Для организации самостоятельной работы студентов по учебной дисциплине следует использовать современные информационные ресурсы: разместить на образовательном портале комплекс учебных и учебно-методических материалов (учебно-программные материалы, учебное издание для теоретического изучения дисциплины, методические указания к лабораторным занятиям, материалы текущего контроля и текущей аттестации, позволяющие определить соответствие учебной деятельности обучающихся требованиям образовательного стандарта высшего образования и учебно-программной документации, в т.ч. вопросы для подготовки к экзамену, задания, тесты, вопросы для самоконтроля, список рекомендуемой литературы, информационных ресурсов и др.).

При составлении общих и индивидуальных заданий по учебной дисциплине необходимо предусмотреть возрастание их сложности: от заданий, формирующих достаточные знания по изученному учебному материалу на уровне узнавания, к заданиям, формирующим компетенции на уровне воспроизведения, и далее к заданиям, формирующим компетенции на уровне применения полученных знаний.

Таким образом, задания по учебной дисциплине рекомендуется делить на три модуля:

- задания, формирующие достаточные знания по изученному учебному материалу на уровне узнавания;
- задания, формирующие компетенции на уровне воспроизведения;
- задания, формирующие компетенции на уровне применения полученных знаний.

### **Примерный перечень вопросов к экзамену**

1. Понятие распределенных систем, свойства распределенных систем, области применения.
2. Типы распределенных систем, их отличия и особенности.
3. Архитектурные стили распределенных систем.
4. Системная (программная) архитектура распределенных систем.
5. Разновидности взаимодействий между процессами в распределенных системах.
6. Передача сообщений между парой процессов.
7. Схема «запрос-ответ» и удаленные вызовы процедур (RPC).

8. Использование потоков в нераспределенных системах. Потоки в распределенных системах.
9. Многопоточные клиенты, многопоточные серверы.
10. Виртуализация и Распределённые системы, типы виртуализации.
11. Сетевые протоколы связи. Эволюция протокола HTTP.
12. Удаленный вызов процедуры (RPC).
13. Веб сервисы, архитектурный стиль REST, сравнение с RPC.
14. Групповые взаимодействия и рассылка.
15. Обмен сообщениями с сокетами. Интерфейс передачи сообщений (MPI).
16. Модель очереди сообщений. Брокеры сообщений.
17. Реализация системы наименования в распределенных системах. Виды имен, схемы именования, разрешение имени.
18. Система доменных имен в интернете (DNS).
19. Децентрализованный подход с плоским пространством имен. Распределённые хеш-таблицы (DHT).
20. Масштабирование распределенных систем. Техники масштабирования.
21. Репликация данных, известные подходы к реализации.
22. Согласованность, связь с репликацией, известные модели.
23. Координация. Время, часы и порядок событий. Физические часы, логические часы.
24. Отказоустойчивость, обнаружение отказов. Типы отказов и модели распределенных систем.
25. Отказоустойчивость. Консенсус и связанные задачи.
26. Отказоустойчивость. Устойчивость к произвольным (византийским) отказам.
27. Надежная связь клиент-сервер, надежное групповое общение.
28. Параллельная обработка: параллельные вычисления, параллельная обработка запросов, параллельная обработка данных.
29. Безопасность в распределенных системах. Требования к безопасности в распределенных системах, базовые техники и механизмы.
30. Параллельное программирование как способ повышения эффективности программы.
31. Многопоточная программа как параллельная программа.
32. Модель делегирования: подход 1 и подход 2.
33. Модель с равноправными узлами: подход 1 и подход 2.
34. Модель Конвейер.
35. Модель Производитель-Потребитель.
36. Классические задачи параллельного программирования.
37. Анализ эффективности использования параллелизма.
38. Методика Яна Фостера.

39. Способы декомпозиции задачи.
40. Классификация средств разработки параллельных программ
41. Механизмы синхронизации.
42. Ошибки параллельного программирования.
43. Способы завершения потоков.
44. Классические задачи синхронизации.
45. Механизмы синхронизации в Java.
46. Механизмы синхронизации в C++.
47. Потокобезопасные структуры данных в Java.
48. Потокобезопасные структуры данных в C++.
49. Синхронизация потоков ввода-вывода.
50. Реализация модели делегирования в Java.
51. Реализация модели делегирования в C++.
52. Реализация модели Производитель-Потребитель в Java.
53. Реализация модели Производитель-Потребитель в C++.
54. Классификация средств поддержки многопоточности в Java.
55. Классификация средств поддержки многопоточности в C++.
56. Шаблоны параллельного программирования.
57. Программный интерфейс OpenMP.
58. Поддержка шаблонов параллельного программирования в OpenMP.

## ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УВО

Название учебной дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)

**ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ ПО  
ИЗУЧАЕМОЙ УЧЕБНОЙ ДИСЦИПЛИНЕ**

на \_\_\_\_ / \_\_\_\_ учебный год

№ п/п	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры  
\_\_\_\_\_ (протокол № \_\_\_\_ от \_\_\_\_\_ 202\_ г.)

Заведующий кафедрой  
\_\_\_\_\_

УТВЕРЖДАЮ  
Декан факультета  
\_\_\_\_\_