

Белорусский государственный университет

УТВЕРЖДАЮ

Проректор по учебной работе и
образовательным инновациям

О.Г. Прохоренко

«01» декабря 2023 г.

Регистрационный № УД-13560/уч.

РАЗРАБОТКА ПАРАЛЛЕЛЬНЫХ ПРИЛОЖЕНИЙ

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

1-31 03 04 Информатика

2023 г.

Учебная программа составлена на основе образовательного стандарта высшего образования ОСВО 1-31 03 04-2021, типового учебного плана №G31-1-029тип. от 30.06.2021 и учебного плана БГУ №G 31-1-031/уч. 30.06.2021.

СОСТАВИТЕЛЬ:

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

РЕЦЕНЗЕНТ:

В.К. Чугунов – руководитель отдела поддержки эксплуатации продуктов и контроля качества ООО «Атлантконсалтсофт».

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 учебной дисциплины – подготовка студентов к самостоятельному решению алгоритмических задач с использованием многопроцессорных вычислительных систем.

Учебная дисциплина «Разработка параллельных приложений» представляет собой комплекс знаний, умений и навыков, позволяющих овладеть основами проектирования параллельных приложений для современных многопроцессорных компьютеров с общей памятью.

Задачи учебной дисциплины

- Понимать преимущества и ограничения параллельного программирования.
- Владеть практическими навыками параллельного программирования на C++ и Java.
- Создавать параллельный программный компонент на основе известного алгоритма.
- Улучшать производительность последовательного приложения путем внедрения параллелизма.

Место учебной дисциплины

В системе подготовки специалиста с высшим образованием для специальности 1-31 03 04 «Информатика» учебная дисциплина относится к **циклу** дисциплин специализации.

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

Основой для изучения учебной дисциплины являются следующие учебные дисциплины: «Промышленное программирование», «Дискретная математика и математическая логика», «Операционные системы».

Требования к компетенциям

Освоение учебной дисциплины «Разработка параллельных приложений» должно обеспечить формирование следующей **универсальной компетенции**:

УК-2. Решать стандартные задачи профессиональной деятельности на основе применения информационно-коммуникационных технологий.

В результате освоения учебной дисциплины студент должен:

знать:

- проблематику параллельных вычислений;
- архитектуру параллельных приложений;
- общие принципы проектирования параллельных приложений;
- основные техники распараллеливания программ;
- модели программирования и методы решения типовых задач.

уметь:

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

владеть:

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

Структура учебной дисциплины

Дисциплина изучается в 6-м семестре. Всего на изучение учебной дисциплины «Разработка параллельных приложений» отведено:

– в очной форме получения высшего образования: 108 часов, в том числе 68 аудиторных часа, из них: лекции – 34 часа, лабораторные занятия – 30 часов, УСП – 4 часа.

Трудоемкость учебной дисциплины составляет 3 зачетные единицы.

Форма промежуточной аттестации – зачет.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Тема 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. 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 3. 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 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 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. Методы синхронизации потоков.

Поддержка многопоточности на разных уровнях абстракции.

Тема 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 7. Технология OpenMP

Понятие параллельной программы. Отличительные особенности технологии.

Конструкции OpenMP: директивы, функции, переменные окружения. Создание потоков, распределение работы между потоками, управление работой с данными, синхронизация потоков.

Проведение вычислительных экспериментов.

Поддержка шаблонов параллельного программирования: SPMD, параллелизм циклов, разделяй и властвуй.

Поддержка шаблонов параллельного программирования: SPMD, параллелизм циклов, разделяй и властвуй.

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 УСР	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	2	3	4	5	6	7	8	9
		34			30		4	
1	Введение в параллельные вычисления	2						Экспресс- опрос
2	Как распараллеливать программу	4			2			Дискуссия. Отчет
3	Параллельные программы	4			4			Дискуссия. Отчет
4	Технологии параллельного программирования	4			4			Собеседован ие. Контрольная работа 1
5	Разработка многопоточных приложений	12			12			Отчет. Контрольная работа 2
6	Шаблоны параллельного программирования	4			4			Отчет
7	Технология OpenMP	4			4		4	Отчет. Собеседован ие

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Перечень основной литературы

1. 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>.
2. Страуструп, Б. Язык программирования C++. Краткий курс / Б. Страуструп. – СПб.: ООО «Диалектика», 2019. – 320 с.
3. Уильямс, Энтони. C++. Практика многопоточного программирования = C++. 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>.
4. Хорстманн, Кей С. Java. Библиотека профессионала, том 1. Основы / Кей С. Хорстманн ; [пер. с англ.]. – 11-е изд. – СПб.: ООО «Диалектика», 2019. – 864 с.
5. 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>.
6. 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. Хьюз, К. Параллельное и распределенное программирование с использованием C++ / К. Хьюз, Т. Хьюз. – М.: Вильямс, 2004. – 672 с.
2. Lesson: Concurrency (The Java™ Tutorials > Essential Java Classes) [Электронный ресурс]. – Режим доступа: <https://docs.oracle.com/javase/tutorial/essential/concurrency/>. – Дата доступа: 02.09.2023.
3. Галовиц, Я. C++17 STL. Стандартная библиотека шаблонов / Я. Галовиц. – СПб.: Питер, 2018. – 432 с.
4. Гетц, Б. Java Concurrency на практике / Б. Гетц, Т. Пайерлс, Дж. Блох, Дж. Боббер, Д. Холмс, Д. Ли. – СПб.: Питер, 2020. – 464 с.
5. Гримм, Р. Параллельное программирование на современном языке C++ / Р. Гримм. – М.: ДМК Пресс, 2022. – 616 с.
6. 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>.
7. Блинов, И. Н. Java from EPAM : учеб.-метод. пособие / И. Н. Блинов, В. С. Романчик. – Минск: Четыре четверти, 2021. – 560 с.
8. Федотов, И.Е. Параллельное программирование. Модели и приемы / И.Е.Федотов. – М.: СОЛОН-Пресс, 2018. – 390 с.

9. Herlihy, M. The Art of Multiprocessor Programming / M. Herlihy, N. Shavit, V. Luchangco, M. Spear – Elsevier Inc., 2021. – 562 p.

10. Mattson, T.G. The OpenMP common core: making OpenMP simple again / T.G. Mattson, Y. He, A.E. Koniges. – Cambridge, Massachusetts: The MIT Press, 2019. – 320 p.

Электронные ресурсы

1. Образовательный портал БГУ [Электронный ресурс]. – Режим доступа: <https://edufpmi.bsu.by/course/view.php?id=232>. – Дата доступа: 02.09.2023.

Перечень рекомендуемых средств диагностики и методика формирования итоговой отметки

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

1. Устная форма: собеседование, экспресс-опрос, дискуссия.
2. Письменная форма: контрольная работа.
3. Устно-письменная форма: отчет по лабораторным работам с их устной защитой и оцениванием на основе модульно-рейтинговой системы.

Формой промежуточной аттестации по дисциплине «Разработка параллельных приложений» учебным планом предусмотрен зачет.

При формировании итоговой отметки используется рейтинговая система оценки знаний студента, дающая возможность проследить и оценить динамику процесса достижения целей обучения.

Рейтинговая система предусматривает использование весовых коэффициентов для текущего контроля знаний и текущей аттестации студентов по дисциплине.

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 – 70 %;
- контрольные работы – 30%.

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

Примерная тематика лабораторных занятий

Лабораторная работа 1. Создание и запуск потоков.

Лабораторная работа 2. Разделение задачи на независимые части. Проведение вычислительных экспериментов.

Лабораторная работа 3. Совместное использование данных.

Лабораторная работа 4. Этапы разработки параллельных алгоритмов. Методология проектирования Яна Фостера.

Лабораторная работа 5. Завершение потока.

Лабораторная работа 6. Учебный пример от Райнера Гримма «Вычисление суммы элементов вектора».

Лабораторная работа 7. Совместное использование данных от Энтони Уильямса.

Лабораторная работа 8. Модели создания и функционирования потоков. Модель делегирования.

Лабораторная работа 9. Модель «изготовитель-потребитель».

Лабораторная работа 10. Средства для работы на уровне заданий.

Лабораторная работа 11. Потокбезопасные структуры данных.

Лабораторная работа 12. Методы синхронизации потоков.

Лабораторная работа 13. Шаблоны параллельного программирования.

Лабораторная работа 14. Технология OpenMP: распараллеливание циклов.

Лабораторная работа 15. Технология OpenMP: поддержка шаблонов проектирования.

Примерный перечень заданий для управляемой самостоятельной работы студентов

Тема 7. Технология OpenMP. (4 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.

Рекомендуемая тематика контрольных работ

1. Контрольная работа 1 «Методика проектирования параллельных программ».

2. Контрольная работа 2 «Реализация конкретного шаблона параллельного программирования на языке высокого уровня для некоторой задачи».

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.

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 образовательных стандартов высшего образования и учебно-программной документации, в том числе вопросы для подготовки к зачету, задания, тесты, вопросы для самоконтроля и др., список рекомендуемой литературы, информационных ресурсов и др.).

Примерный перечень вопросов к зачету

1. Параллельное программирование как способ повышения эффективности программы.
2. Многопоточная программа как параллельная программа.
3. Модель делегирования: подход 1 и подход 2.
4. Модель с равноправными узлами: подход 1 и подход 2.
5. Модель Конвейер.
6. Модель Производитель-Потребитель.
7. Классические задачи параллельного программирования.
8. Анализ эффективности использования параллелизма.
9. Методика Яна Фостера.
10. Способы декомпозиции задачи.
11. Классификация средств разработки параллельных программ

12. Механизмы синхронизации.
13. Ошибки параллельного программирования.
14. Способы завершения потоков.
15. Классические задачи синхронизации.
16. Механизмы синхронизации в Java.
17. Механизмы синхронизации в C++.
18. Потокобезопасные структуры данных в Java.
19. Потокобезопасные структуры данных в C++.
20. Синхронизация потоков ввода-вывода.
21. Реализация модели делегирования в Java.
22. Реализация модели делегирования в C++.
23. Реализация модели Производитель-Потребитель в Java.
24. Реализация модели Производитель-Потребитель в C++.
25. Классификация средств поддержки многопоточности в Java.
26. Классификация средств поддержки многопоточности в C++.
27. Шаблоны параллельного программирования.
28. Программный интерфейс OpenMP.
29. 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)
Учебная дисциплина не требует согласования			

**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 _____ 20_ г.)

Заведующий кафедрой

УТВЕРЖДАЮ
Декан факультета
