

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ЛУГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ ВЛАДИМИРА ДАЛЯ»**

Факультет компьютерных систем и информационных технологий

Кафедра информатики и программной инженерии

УТВЕРЖДАЮ

**Декан факультета компьютерных
систем и информационных технологий**

Кочевский А. А.

« 19 » 04 2023 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

«Программирование для параллельных вычислений»

по направлению подготовки 09.03.04 Программная инженерия

профиль подготовки «Разработка программно-информационных систем»

Луганск – 2023

Лист согласования РПУД

Рабочая программа учебной дисциплины «Программирование для параллельных вычислений» по направлению подготовки 09.03.04 Программная инженерия. – 13 с.

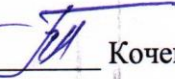
Рабочая программа учебной дисциплины «Программирование для параллельных вычислений» составлена на основе федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.04 Программная инженерия, утвержденного приказом Министерства образования и науки Российской Федерации от 19 сентября 2017 года № 920 (с изменениями и дополнениями), зарегистрированного в Министерстве юстиции Российской Федерации 16 октября 2017 года № 48546, учебного плана по направлению подготовки 09.03.04 Программная инженерия, (профиль «Разработка программно-информационных систем») и Положения о рабочей программе учебной дисциплины в ФГБОУ ВО «ЛГУ им. В. Даля».

СОСТАВИТЕЛЬ

канд. техн. наук, доцент, доцент кафедры информатики и программной инженерии
Письменский А.В.

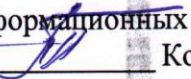
Рабочая программа дисциплины утверждена на заседании кафедры информатики и программной инженерии

18 апреля 2023 г., протокол № 17

Заведующий кафедрой информатики и программной инженерии  Кочевский А.А.

Переутверждена: «__» _____ 20__ г., протокол № _____

Согласована:

Декан факультета компьютерных систем и информационных технологий  Кочевский А. А.

Рекомендована на заседании учебно-методической комиссии факультета компьютерных систем и информационных технологий

19 апреля 2023 г., протокол № 8

Председатель учебно-методической комиссии факультета компьютерных систем и информационных технологий 

Ветрова Н. Н.

Структура и содержание дисциплины

1. Цели и задачи дисциплины, ее место в учебном процессе

Цель изучения дисциплины – разработка параллельных и распределенных систем, а также их программного или аппаратного воплощения. Использование многоядерных процессоров или многопроцессорных систем для осуществления многопоточности и выполнения несколько процессов параллельно.

Задачи: изучение основных принципов разработки параллельных и распределенных систем для различных применений: автоматизация процессов распознавания образов, адаптивное управление, аппроксимация функционалов, прогнозирование, создание экспертных систем, организация ассоциативной памяти и многие другие приложения. Изучение стандартной библиотека C++ Thread Library для реализации высокоуровневых абстракций.

При освоении программы у обучающихся формируется информационно-коммуникационная компетентность – знания, умения и навыки по объектно-ориентированному программированию, необходимые для изучения других общеобразовательных предметов, для их использования в ходе изучения специальных дисциплин профессионального цикла, в практической деятельности и повседневной жизни.

2. Место дисциплины в структуре ОПОП ВО

Дисциплина «Программирование для параллельных вычислений» входит в блок дисциплин обязательной части учебного плана.

Необходимыми условиями для освоения дисциплины являются: знания и принципы построения современных кроссплатформенных приложений и знаний ООП, языка программирования C++. Изложены основные принципы построения распределенных систем, основные стандарты проектирования программных систем, пакеты программ для объектно-ориентированного программирования.

Дисциплина реализуется кафедрой информатики и программной инженерии. Основывается на базе дисциплин предыдущего уровня образования и является логическим продолжением содержания дисциплин естественно-научного цикла.

Содержание дисциплины является логическим продолжением содержания дисциплин информатика, программирование и служит основой для освоения дисциплин: разработка клиент-серверных приложений, технология NET.

3. Требования к результатам освоения содержания дисциплины

Студенты, завершившие изучение дисциплины «Программирование для параллельных вычислений», должны

знать:

- организация параллелизма за счет нескольких процессов;
- организация параллелизма за счет нескольких потоков;

- основные причины для использования параллелизма в приложении;

- использование библиотеки многопоточности для с++;
- особенности использования платформенно-зависимых средств ;
- базовые функции и операции управления потоками;
- методы управления данными в потоке;
- способы передачи владения потока;
- методы идентификация потоков;
- проблемы разделения данных между потоками;
- способы защиты разделяемых данных с помощью мьютексов;
- методы организации потокобезопасной очереди на базе условных

переменных

- методы доступа к результатам асинхронных операций;
- основные концепции библиотек работы со временем;
- основы функционального программирования с применением

будущих результатов;

- атомарные типы и операции;
- методы синхронизация операций и принудительного

упорядочения;

- методы упорядочения, захвата-освобождения.

уметь:

- о проектировать параллельные и распределенные системы;

- о использовать библиотеку C++ Thread Library для реализации многопоточности;

- о выполнять защиту разделяемых данных с помощью условных переменных;

- о выполнять одновременный доступ к ресурсам;

- о выполнять защиту разделяемых данных;

Перечисленные результаты образования являются основой для формирования следующих компетенций (в соответствии с ФГОС ВО и требованиями к результатам освоения основной профессиональной образовательной программы (ОПОП ВО):

профессиональных:

ПК-3.1 Знать: методы и приемы формализации задач; языки формализации функциональных спецификаций; методы и приемы алгоритмизации поставленных задач; нотации и программные продукты для графического отображения алгоритмов; алгоритмы решения типовых задач, области и способы их применения

ПК-3.2 Уметь: использовать и применять: методы и приемы формализации задач; методы и приемы алгоритмизации поставленных задач; программные продукты для графического отображения алгоритмов; стандартные алгоритмы в

- ПК-3.3 Владеть: приемами составления формализованных описаний решений поставленных задач в соответствии с требованиями технического задания или других нормативных документов; приемами разработки алгоритмов решения поставленных задач в соответствии с требованиями технического задания или других нормативных документов; приемами оценки и согласования сроков выполнения поставленных задач

4. Структура и содержание дисциплины

4.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем часов (з.е.)		
	Очная форма	Очно-заочная форма	Заочная форма
Объем учебной дисциплины (всего)	324 (9 з.е.)		324 (9 з.е.)
Обязательная аудиторная учебная нагрузка дисциплины (всего) в том числе:	140		24
Лекции	56		12
Семинарские занятия	-		-
Практические занятия			-
Лабораторные работы	84		12
Курсовая работа (курсовой проект)	-		-
Индивидуальное задание	36		36
Самостоятельная работа студента (всего)	184		300
Форма аттестации	экзамен		экзамен

4.2. Содержание разделов дисциплины

Семестр 7

Тема 1. ВВЕДЕНИЕ В ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ.

Параллелизм в вычислительных системах. Подходы к организации параллелизма. Параллелизм за счет нескольких процессов. Параллелизм за счет нескольких потоков.

Тема 2. ЗАЧЕМ НУЖЕН ПАРАЛЛЕЛИЗМ.

Применение параллелизма для разделения обязанностей. Применение параллелизма для повышения производительности. Когда параллелизм вреден.

Тема 3. ПАРАЛЛЕЛИЗМ И МНОГОПОТОЧНОСТЬ В C++.

История многопоточности в C++. Поддержка параллелизма в новом стандарте. Эффективность библиотеки многопоточности для C++. Платформенно-зависимые средства.

Тема 4. УПРАВЛЕНИЕ ПОТОКАМИ.

Базовые операции управления потоками. Запуск потока. Ожидание завершения потока. Ожидание в случае исключения. Запуск потоков в фоновом режиме.

Тема 5. УПРАВЛЕНИЕ ДАННЫМИ В ПОТОКЕ.

Передача аргументов функции потока. Ссылки на r-значения. Семантика перемещения. Ссылки на r-значения и шаблоны функций.

Тема 6. ВЛАДЕНИЯ ПОТОКОМ.

Передача владения потоком. Задание количества потоков во время выполнения. Идентификация потоков.

Тема 7. РАЗДЕЛЕНИЕ ДАННЫХ МЕЖДУ ПОТОКАМИ.

Проблемы разделения данных между потоками. Гонки. Устранение проблематичных состояний гонки.

Тема 8. ЗАЩИТА РАЗДЕЛЯЕМЫХ ДАННЫХ С ПОМОЩЬЮ МЬЮТЕКСОВ.

Использование мьютексов в C++. Структурирование кода для защиты разделяемых данных. Выявление состояний гонки, внутренне присущих интерфейсам.

Тема 9. ВЗАИМОБЛОКИРОВКА: ПРОБЛЕМА И РЕШЕНИЕ.

Дополнительные рекомендации, как избежать взаимоблокировок. Гибкая блокировка с помощью `std::unique_lock`. Передача владения мьютексом между контекстами. Выбор правильной гранулярности блокировки.

Тема 10. ДРУГИЕ СРЕДСТВА ЗАЩИТЫ РАЗДЕЛЯЕМЫХ ДАННЫХ.

Защита разделяемых данных во время инициализации. Защита редко обновляемых структур данных. Рекурсивная блокировка.

Семестр 8

Тема 11. Синхронизация параллельных операций

Ожидание события или иного условия. Ожидание условия с помощью условных переменных. Потокобезопасная очередь на базе условных переменных

Тема 12. Ожидание одноразовых событий с помощью механизма будущих результатов

Возврат значения из фоновой задачи. Ассоциирование задачи с будущим результатом. Использование `std::promise`. Сохранение исключения в будущем результате. Ожидание в нескольких потоках.

Тема 13. Ожидание с ограничением по времени

Часы. Временные интервалы. Моменты времени. Функции, принимающие таймаут.

Тема 14. Применение синхронизации операций для упрощения кода

Функциональное программирование с применением будущих результатов. Синхронизация операций с помощью передачи сообщений.

Тема 15. Модель памяти C++ и атомарные операции

Объекты и ячейки памяти. Объекты, ячейки памяти и параллелизм. Порядок модификации. Атомарные операции и типы в C++.

Тема 16. Синхронизация операций и принудительное упорядочение

Отношение синхронизируется. Отношение происходит-раньше. Упорядочение доступа к памяти для атомарных операций. Последовательности освобождений. Барьеры. Упорядочение неатомарных операций с помощью атомарных

4.3. Лекции

№ п/п	Название темы	Объем часов		
		Очная форма	Очно- заочная форма	Заочная форма
1.	Введение в параллельное программирование	3		2
2.	Зачем нужен параллелизм	3		

3.	Параллелизм и многопоточность в C++	3		2
4.	Управление потоками	3		
5.	Управление данными в потоке	3		2
6.	Владения потоком	3		
7.	Разделение данных между потоками	3		
8.	Защита разделяемых данных с помощью мьютексов	3		2
9.	Выявление состояний гонки	3		
10.	Взаимоблокировка: проблема и решение	3		
11.	Передача владения мьютексом между контекстами	3		2
12.	Другие средства защиты разделяемых данных	3		
13.	Защита редко обновляемых структур данных	3		
14.	Рекурсивная блокировка	3		
15.	Синхронизация параллельных операций	3		
16.	Ожидание одноразовых событий с помощью механизма будущих результатов	3		
17.	Ожидание с ограничением по времени	2		
18.	Применение синхронизации операций для упрощения кода	2		
19.	Модель памяти C++ и атомарные операции	2		2
20.	Синхронизация операций и принудительное упорядочение	2		
Итого:		56		12

4.4. Практические (семинарские) занятия

Не предусмотрены

4.5. Лабораторные работы

№ п/п	Название темы	Объем часов		
		Очная форма	Очно- заочная форма	Заочная форма
1.	Введение в параллельное программирование	5		2
2.	Управление потоками	5		
3.	Управление данными в потоке	5		
4.	Ссылки на r-значения	5		2
5.	Передача владения потоком	5		
6.	Задание количества потоков во время выполнения	5		
7.	Идентификация потоков	5		
8.	Защита разделяемых данных с помощью мьютексов	5		2
9.	Одновременный доступ к ресурсам	5		
10.	Синхронизация действия, выполняемые в разных потоках	5		
11.	Моделирование одноразовых событий с помощью будущего результата	5		2
12.	Основные концепции библиотеки Chrono (C++)	5		
13.	Алгоритма быстрой сортировки Quicksort	6		
14.	Атомарные типы и операции	6		2

15.	Синхронизация операций и принудительное упорядочение modification order	6		
16.	Зависимости по данным, упорядочение захват-освобождение и семантика memory_order_consume	6		2
Итого:		84		12

4.6. Самостоятельная работа студентов

№ п/п	Название темы	Вид СРС	Объем часов	
			Очная форма	Заочная форма
1.	Подходы к организации параллелизма.	Написание реферата, создание презентации по теме	7	12
2.	Когда параллелизм вреден.	Написание реферата, создание презентации по теме	7	12
3.	Платформенно-зависимые средства	Написание реферата, создание презентации по теме	7	12
4.	Запуск потоков в фоновом режиме	Подготовка к лабораторным работам	7	12
5.	Ссылки на г-значения и шаблоны функций	Подготовка к лабораторным работам	7	12
6.	Идентификация потоков	Подготовка к лабораторным работам	7	12
7.	Устранение проблематичных состояний гонки.	Подготовка к лабораторным работам	7	12
8.	Структурирование кода для защиты разделяемых данных	Подготовка к лабораторным работам	7	12
9.	Гибкая блокировка с помощью std::unique_lock	Подготовка к лабораторным работам	7	12
10.	Защита редко обновляемых структур данных	Подготовка к лабораторным работам	7	12
11.	Быстрая сортировка в духе ФП	Написание реферата, создание презентации по теме	7	12
12.	Параллельная реализация Quicksort в духе ФП	Написание реферата, создание презентации по теме	7	12
13.	Стандартные атомарные типы	Написание реферата, создание презентации по теме	7	12
14.	Операции над std::atomic_flag	Подготовка к лабораторным работам	7	12
15.	Операции над std::atomic<bool>	Подготовка к лабораторным работам	7	12
16.	Операции над std::atomic<T*>: арифметика указателей	Подготовка к лабораторным работам	7	12
17.	Операции над стандартными атомарными целочисленными типами	Подготовка к лабораторным работам	8	12
18.	Основной шаблон класса std::atomic<>	Подготовка к лабораторным работам	8	12
19.	Свободные функции для атомарных операций	Подготовка к лабораторным работам	8	12

20.	Последовательно согласованное упорядочение	Подготовка к лабораторным работам	8	12
21.	Не последовательно согласованное упорядочение доступа к памяти	Подготовка к лабораторным работам	8	12
22.	Ослабленное упорядочение	Подготовка к лабораторным работам	8	12
23.	Механизм ослабленного упорядочения	Подготовка к лабораторным работам	8	12
24.	Упорядочение захват-освобождение	Подготовка к лабораторным работам	8	12
25.	Транзитивная синхронизация с помощью упорядочения захват-освобождение	Написание реферата, создание презентации по теме	8	12
Итого:			184	300

4.7. Курсовые работы/проекты

Курсовые работы (проекты) не предусмотрены.

5. Образовательные технологии

Преподавание дисциплины ведется с применением следующих видов образовательных технологий:

- традиционные объяснительно-иллюстративные технологии, которые обеспечивают доступность учебного материала для большинства студентов, системность, отработанность организационных форм и привычных методов, относительно малые затраты времени;

- технологии проблемного обучения, направленные на развитие познавательной активности, творческой самостоятельности студентов и предполагающие последовательное и целенаправленное выдвижение перед студентом познавательных задач, разрешение которых позволяет студентам активно усваивать знания (используются поисковые методы; постановка познавательных задач);

- технологии развивающего обучения, позволяющие ориентировать учебный процесс на потенциальные возможности студентов, их реализацию и развитие;

- технологии концентрированного обучения, суть которых состоит в создании максимально близкой к естественным психологическим особенностям человеческого восприятия структуры учебного процесса и которые дают возможность глубокого и системного изучения содержания учебных дисциплин за счет объединения занятий в тематические блоки;

- технологии модульного обучения, дающие возможность обеспечения гибкости процесса обучения, адаптации его к индивидуальным потребностям и особенностям обучающихся (применяются, как правило, при самостоятельном обучении студентов по индивидуальному учебному плану);

- технологии дифференцированного обучения, обеспечивающие возможность создания оптимальных условий для развития интересов и способностей студентов, в том числе и студентов с особыми

образовательными потребностями, что позволяет реализовать в культурно-образовательном пространстве университета идею создания равных возможностей для получения образования

– технологии активного (контекстного) обучения, с помощью которых осуществляется моделирование предметного, проблемного и социального содержания будущей профессиональной деятельности студентов (используются активные и интерактивные методы обучения) и т.д.

Максимальная эффективность педагогического процесса достигается путем конструирования оптимального комплекса педагогических технологий и (или) их элементов на личностно-ориентированной, деятельностной, диалогической основе и использования необходимых современных средств обучения.

6. Формы контроля освоения дисциплины

Текущая аттестация студентов производится в дискретные временные интервалы лектором и преподавателем(ями), ведущими лабораторные работы и практические занятия по дисциплине в следующих формах (например):

- доклады, сообщения;
- лабораторные работы;
- защита лабораторных работ (тестирование).

Фонды оценочных средств, включающие типовые задания, контрольные работы, тесты и методы контроля, позволяющие оценить результаты текущей и промежуточной аттестации обучающихся по данной дисциплине, помещаются в приложении к рабочей программе в соответствии с «Положением о фонде оценочных средств».

Форма аттестации по результатам освоения дисциплины проходит в форме письменного экзамена (включает в себя ответ на теоретические вопросы и решение задач). Студенты, выполнившие 75% текущих и контрольных мероприятий на «отлично», а остальные 25 % на «хорошо», имеют право на получение итоговой отличной оценки.

В экзаменационную ведомость и зачетную книжку выставляются оценки по шкале, приведенной в таблице.

Национальная шкала	Характеристика знания предмета и ответов
отлично (5)	Студент глубоко и в полном объеме владеет программным материалом. Грамотно, исчерпывающе и логично его излагает в устной или письменной форме. При этом знает рекомендованную литературу, проявляет творческий подход в ответах на вопросы и правильно обосновывает принятые решения, хорошо владеет умениями и навыками при выполнении практических задач.
хорошо (4)	Студент знает программный материал, грамотно и по сути излагает его в устной или письменной форме, допуская незначительные неточности в утверждениях, трактовках, определениях и категориях или незначительное количество ошибок. При этом владеет необходимыми умениями и навыками при выполнении практических задач.

удовлетворительно (3)	Студент знает только основной программный материал, допускает неточности, недостаточно чёткие формулировки, непоследовательность в ответах, излагаемых в устной или письменной форме. При этом недостаточно владеет умениями и навыками при выполнении практических задач. Допускает до 30% ошибок в излагаемых ответах.
неудовлетворительно (2)	Студент не знает значительной части программного материала. При этом допускает принципиальные ошибки в доказательствах, в трактовке понятий и категорий, проявляет низкую культуру знаний, не владеет основными умениями и навыками при выполнении практических задач. Студент отказывается от ответов на дополнительные вопросы.

7. Учебно-методическое и программно-информационное обеспечение дисциплины:

1. Энтони Уильямс. Параллельное программирование на C++ в действии. Практика разработки многопоточных программ. Пер. с англ. Слинкин А. А. – М.: ДМК Пресс, 2018. – 672с.: ил.

2. Богачёв К. Ю. Б73 Основы параллельного программирования [Электронный ресурс]: учебное пособие / К. Ю. Богачёв. – 2-е изд. (эл.). – М.: БИНОМ. Л

3. Антонов А.С. Параллельное программирование с использованием технологии OpenMP: Учебное пособие. – М.: Изд-во МГУ, 2009. – 77 с.

4. OpenMP Architecture Review Board (<http://www.openmp.org/>).

5. The Community of OpenMP Users, Researchers, Tool Developers and Providers (<http://www.compunity.org/>).

6. OpenMP Application Program Interface Version 3.0 May 2018 (<http://www.openmp.org/mp-documents/spec30.pdf>).

7. Что такое OpenMP? (http://parallel.ru/tech/tech_dev/openmp.html).

8. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002. – 608 с.

9. Barbara Chapman, Gabriele Jost, Ruud van der Pas. Using OpenMP: portable shared memory parallel programming (Scientific and Engineering Computation). Cambridge, Massachusetts: The MIT Press., 2008. – 353 pp.

10. Антонов А.С. Введение в параллельные вычисления. Методическое пособие.-М.: Изд-во Физического факультета МГУ, 2002. – 70 с.

11. Антонов А.С. Параллельное программирование с использованием технологии MPI: Учебное пособие. – М.: Изд-во МГУ, 2014. – 71 с.

в) методические указания:

1. Методические указания к лабораторным работам по дисциплине «Параллельные и распределенные вычисления» для студентов всех направлений подготовки. Луганск: Изд-во ЛНУ им. В.Даля, 2018. – 31 с.

2. Конспект лекций по дисциплине “Параллельные и распределенные вычисления” (для студентов по направлению «Прикладная информатика» 09.03.03) Часть I / Сост. Письменский А.В. – Луганск: изд-во ЛГУ им. В.И.Даля, 2018.– 116 с.

3. Конспект лекций по дисциплине «Параллельное программирование» (для студентов по направлению «Программная инженерия» 09.04.04) / Сост. Письменский А.В. – Луганск: ГОУ ВО ЛНР ЛГУ В.И.Даля, 2022.– 116 с.

в) Интернет-ресурсы:

1. Atomic Ptr Plus Project Home, <http://atomic-ptr-plus.sourceforge.net/>.
2. Boost C++ library collection, <http://www.boost.org>.
3. C++0x/C++11 Support in GCC, <http://gcc.gnu.org/projects/cxxOx.html>.
4. C++11—The Recently Approved New ISO C++ Standard, <http://www.research.att.com/~bs/C++OxFAQ.html>.
5. Erlang Programming Language, <http://www.erlang.org/>.
6. GNU General Public License, <http://www.gnu.org/licenses/gpl.html>.
Haskell Programming Language, <http://www.haskell.org/>

Информационный ресурс библиотеки образовательной организации

Научная библиотека имени А. Н. Коняева – <http://biblio.dahluniver.ru/>

8. Материально-техническое обеспечение дисциплины

Лекционные занятия: комплект электронных презентаций/слайдов; аудитория, оснащенная презентационной техникой (проектор, экран, компьютер/ноутбук).

Лабораторные работы: компьютерный класс, презентационная техника (проектор, экран, компьютер/ноутбук), пакеты ПО общего назначения.

Рабочее место преподавателя, оснащенное компьютером с доступом в Интернет, рабочие места студентов, оснащенные компьютерами с доступом в Интернет, предназначенные для работы в электронной образовательной среде.

Программное обеспечение:

Функциональное назначение	Бесплатное программное обеспечение	Ссылки
Офисный пакет	Libre Office 6.3.1	https://www.libreoffice.org/ https://ru.wikipedia.org/wiki/LibreOffice
Операционная система	UBUNTU 19.04	https://ubuntu.com/ https://ru.wikipedia.org/wiki/Ubuntu
Браузер	Firefox Mozilla	http://www.mozilla.org/ru/firefox/fx
Браузер	Opera	http://www.opera.com
Архиватор	7Zip	http://www.7-zip.org/
Редактор PDF	PDFCreator	http://www.pdfforge.org/pdfcreator
Математический редактор	Mathcad Prime 6.0	https://www.mathcad.com/ru/
Интегрированная среда разработки	Visual Studio 2019	https://visualstudio.microsoft.com/ru/free-developer-offers/
Серверная платформа и программная среда	Open Server Panel	https://ospanel.io/download/
Интегрированная среда разработки	CODESYS V3	https://owen.ru/product/codesys_v3
Графическая среда имитационного моделирования	Simulink	https://matlab.ru/products/Simulink