

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Галунин Сергей Александрович  
Должность: проректор по учебной работе  
Дата подписания: 23.12.2025 13:42:26  
Уникальный программный ключ:  
08ef34338325bdb0ac5a47baa5472ce36cc3fc3b

Приложение к ОПОП  
«Разработка программно-  
информационных систем»



**СПбГЭТУ «ЛЭТИ»**  
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

МИНОБРНАУКИ РОССИИ

федеральное государственное автономное образовательное учреждение высшего образования  
**«Санкт-Петербургский государственный электротехнический университет  
«ЛЭТИ» им. В.И.Ульянова (Ленина)»  
(СПбГЭТУ «ЛЭТИ»)**

---

**РАБОЧАЯ ПРОГРАММА**

дисциплины

**«ПАРАЛЛЕЛЬНЫЕ СИСТЕМЫ»**

для подготовки бакалавров

по направлению

09.03.04 «Программная инженерия»

по профилю

**«Разработка программно-информационных систем»**

Санкт-Петербург

2025

## **ЛИСТ СОГЛАСОВАНИЯ**

Разработчики:

старший преподаватель Калишенко Е.Л.

Рабочая программа рассмотрена и одобрена на заседании кафедры МОЭВМ  
20.01.2025, протокол № 1

Рабочая программа рассмотрена и одобрена учебно-методической комиссией  
ФКТИ, 28.01.2025, протокол № 1

Согласовано в ИС ИОТ

Начальник ОМОЛА Загороднюк О.В.

## 1 СТРУКТУРА ДИСЦИПЛИНЫ

Обеспечивающий факультет	ФКТИ
Обеспечивающая кафедра	МОЭВМ

Общая трудоемкость (ЗЕТ)	3
Курс	4
Семестр	7

### **Виды занятий**

Электронные лекции (акад. часов)	34
Электронные практические (академ. часов) (академ. часов)	34
Иная контактная работа (академ. часов)	1
Все контактные часы (академ. часов)	1
Самостоятельная работа, включая часы на контроль (академ. часов)	39
Всего (академ. часов)	108

### **Вид промежуточной аттестации**

Дифф. зачет (курс)	4
--------------------	---

## **2 АННОТАЦИЯ ДИСЦИПЛИНЫ**

### **«ПАРАЛЛЕЛЬНЫЕ СИСТЕМЫ»**

Дисциплина посвящена изучению современных методов и средств обеспечения высокопроизводительных вычислений. Начиная с проектирования параллельных программ, заканчивая анализом производительности и автоматизированным поиском ошибок. В курсе рассматриваются современные библиотеки программ, стандарты и методы параллельного программирования как в многоядерной среде, так и в распределённых системах. Большое внимание уделено теоретическим основам современных алгоритмов неблокирующей синхронизации и алгоритмам без ожидания.

### **SUBJECT SUMMARY**

#### **«PARALLEL SYSTEMS»**

Discipline is devoted to studying the modern methods and means of high-performance computing. It covers questions from parallel programs architecture to performance analysis and automatic error detection. Set of libraries, standards and methods of parallel programming both for multithreaded and distributed systems are considered. Great attention is devoted to theory and development principles of lock-free and wait-free algorithms.

## **3 ОБЩИЕ ПОЛОЖЕНИЯ**

### **3.1 Цели и задачи дисциплины**

1. Целью изучения дисциплины является овладение теоретическими знаниями современных методов и средств обеспечения высокопроизводительных вычислений в параллельных системах, а также умениями и навыками их применения.

2. Задачами изучения дисциплины являются:

- приобретение знаний, умений и навыков применения современных библиотек программ, стандартов и методов параллельного программирования как в многоядерной среде, так и в распределённых системах;
- ориентация в проектировании параллельных программ, анализе производительности и автоматизированном поиске ошибок.

3. Обучающиеся получают знания:

- о принципах построения и исполнения параллельных приложений;
- о видах примитивов синхронизации и алгоритмах их реализации;
- об алгоритмических подходах к уменьшению задержек при синхронизации на различных структурах данных;
- методов поиска и профилактики ошибок параллельного программирования.

4. Обучающиеся приобретут умения:

- правильно выбирать средства реализации высокопроизводительных приложений;
- проектировать высокопроизводительные приложения с применением шаблонов параллельного программирования;
- использовать средства асинхронной коммуникации приложений;
- использовать инструменты цикла разработки высокопроизводительных приложений.

5. У обучающихся будут сформированы навыки:

- применения на практике методов и средств распределения задач по вычислительным элементам;
- выбора шаблонов параллельного программирования для решения конкретных задач.

### **3.2 Место дисциплины в структуре ОПОП**

Дисциплина изучается на основе ранее освоенных дисциплин учебного плана:

1. «Алгебраические структуры»
2. «Параллельные алгоритмы»

и обеспечивает изучение последующих дисциплин:

1. «Производственная практика (преддипломная практика)»
2. «Распределенные вычислительные системы»

### 3.3 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

В результате освоения образовательной программы обучающийся должен достичь следующие результаты обучения по дисциплине:

<b>Код компетенции/ индикатора компетенции</b>	<b>Наименование компетенции/индикатора компетенции</b>
ПК-0	Способен разрабатывать информационные модели и применять их для решения задач профессиональной деятельности
<i>ПК-0.1</i>	<i>Знает современные виды информационных моделей, применяемых при решении задач профессиональной деятельности</i>
<i>ПК-0.2</i>	<i>Создает и модифицирует информационные модели для решения задач профессиональной деятельности</i>
<i>ПК-0.3</i>	<i>Применяет информационные модели для решения задач профессиональной деятельности</i>

## 4 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### 4.1 Содержание разделов дисциплины

#### 4.1.1 Наименование тем и часы на все виды нагрузки

№ п/п	Наименование темы дисциплины	ЭЛек, ач	ЭПр, ач	ИКР, ач	СР, ач
1	Введение	2	2		
2	Тема 1. Работа с потоками	10	10		10
3	Тема 2. Параллельное программирование.	10	10		10
4	Тема 3. Управление памятью в многопоточных системах	10	10		10
5	Заключение	2	2	1	9
	Итого, ач	34	34	1	39
	Из них ач на контроль	0	0	0	0
	Общая трудоемкость освоения, ач/зе	108/3			

#### 4.1.2 Содержание

№ п/п	Наименование темы дисциплины	Содержание
1	Введение	Место дисциплины в ООП. Современное состояние проблемы.
2	Тема 1. Работа с потоками	Создание/завершение потоков. Прimitives синхронизации. Алгоритмы синхронизации. Атомарные снимки регистров.
3	Тема 2. Параллельное программирование.	Ошибки параллельного программирования. Профилирование многопоточных приложений. Java.util.concurrent и Fork-Join Framework. OpenMP и Intel TBB. Шаблоны параллельного программирования. Кластерные вычисления. Консенсус. Сети Петри.
4	Тема 3. Управление памятью в многопоточных системах	Оптимизации в компиляторах. Транзакционная память. Асинхронный ввод/вывод. Wait-free MRMW снимок регистров. Средства поиска ошибок. Модель памяти. Lock-free схемы управления памятью. Модель акторов.
5	Заключение	Общие выводы по курсу. Рекомендации для подготовки к дифференцированному зачету.



## 4.2 Перечень лабораторных работ

Лабораторные работы не предусмотрены.

## 4.3 Перечень практических занятий

Наименование практических занятий	Количество ауд. часов
1. Примитивы и алгоритмы синхронизации.	2
2. Ошибки параллельного программирования.	4
3. Профилирование многопоточных приложений.	4
4. Шаблоны параллельного программирования.	4
5. Сети Петри.	4
6. Оптимизации в компиляторах.	4
7. Асинхронный ввод/вывод.	4
8. Wait-free MRMW снимок регистров.	4
9. Средства поиска ошибок.	4
Итого	34

## 4.4 Курсовое проектирование

Курсовая работа (проект) не предусмотрены.

## 4.5 Реферат

Реферат не предусмотрен.

## 4.6 Индивидуальное домашнее задание

Индивидуальное домашнее задание не предусмотрено.

## 4.7 Доклад

Доклад не предусмотрен.

## 4.8 Кейс

Кейс не предусмотрен.

#### **4.9 Организация и учебно-методическое обеспечение самостоятельной работы**

Изучение дисциплины сопровождается самостоятельной работой студентов с рекомендованными преподавателем литературными источниками и информационными ресурсами сети Интернет.

Планирование времени для изучения дисциплины осуществляется на весь период обучения, предусматривая при этом регулярное повторение пройденного материала. Обучающимся, в рамках внеаудиторной самостоятельной работы, необходимо регулярно дополнять сведениями из литературных источников материал, законспектированный на лекциях. При этом на основе изучения рекомендованной литературы целесообразно составить конспект основных положений, терминов и определений, необходимых для освоения разделов учебной дисциплины.

Особое место уделяется консультированию, как одной из форм обучения и контроля самостоятельной работы. Консультирование предполагает особым образом организованное взаимодействие между преподавателем и студентами, при этом предполагается, что консультант либо знает готовое решение, которое он может предписать консультируемому, либо он владеет способами деятельности, которые указывают путь решения проблемы.

Самостоятельной записи на курс нет. Студент заходит на курс, используя логин/пароль от единой учетной записи университета (единый логин и пароль). С материалами курса - презентациями и конспектами обучающиеся смогут ознакомиться в любое удобное время. Все темы включают практические занятия, которые предусматривают самостоятельное выполнение заданий, а также задания с автоматической проверкой, результаты которых учитываются при общей аттестации полученных знаний. Рекомендуем изучать материал последовательно, что существенно облегчит работу. У каждого практического зада-

ния есть срок выполнения (окончательный срок), по истечении которого даже правильные ответы система принимать не будет. Весь учебный курс рассчитан на 16 недель. Его итоги будут подведены в течение последней недели семестра.

<b>Текущая СРС</b>	<b>Примерная трудоемкость, ач</b>
Работа с лекционным материалом, с учебной литературой	5
Опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	
Самостоятельное изучение разделов дисциплины	5
Выполнение домашних заданий, домашних контрольных работ	
Подготовка к лабораторным работам, к практическим и семинарским занятиям	10
Подготовка к контрольным работам, коллоквиумам	10
Выполнение расчетно-графических работ	
Выполнение курсового проекта или курсовой работы	
Поиск, изучение и презентация информации по заданной проблеме, анализ научных публикаций по заданной теме	
Работа над междисциплинарным проектом	
Анализ данных по заданной теме, выполнение расчетов, составление схем и моделей, на основе собранных данных	
Подготовка к зачету, дифференцированному зачету, экзамену	9
<b>ИТОГО СРС</b>	<b>39</b>

## 5 Учебно-методическое обеспечение дисциплины

### 5.1 Перечень основной и дополнительной литературы, необходимой для освоения дисциплины

№ п/п	Название, библиографическое описание	К-во экз. в библ.
Основная литература		
1	Таненбаум Э. Современные операционные системы : 4-е изд / Э. Таненбаум, Х. Бос, 2019. -1120 с. -Текст : непосредственный.	неогр.
2	Сиротинина Н. Ю. Параллельные вычислительные системы [Электронный ресурс] : учебное пособие, 2019. -178 с.	неогр.
Дополнительная литература		
1	Бабичев С. Л., Коньков К. А. РАСПРЕДЕЛЕННЫЕ СИСТЕМЫ. : Учебник для вузов / Л., Коньков К. А. Бабичев С., 2025. -507 с. -Текст : непосредственный.	неогр.
2	Малявко А. А. ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ НА ОСНОВЕ ТЕХНОЛОГИЙ OPENMP, CUDA, OPENCL, MPI 3-е изд., испр. и доп. : Учебник для вузов / А. Малявко А., 2025. -135 с. -Текст : непосредственный.	неогр.

### 5.2 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», используемых при освоении дисциплины

№ п/п	Электронный адрес
1	Материалы НГУ <a href="http://www.hpcc.unn.ru/multicore/programm.html">http://www.hpcc.unn.ru/multicore/programm.html</a>
2	Архитектура и реализация lock-free алгоритмов <a href="http://habrahabr.ru/hub/parallel_programming/">http://habrahabr.ru/hub/parallel_programming/</a>

### 5.3 Адрес сайта курса

Адрес сайта курса: <https://vec.etu.ru/moodle/course/view.php?id=25240>

## 6 Критерии оценивания и оценочные материалы

### 6.1 Критерии оценивания

Для дисциплины «Параллельные системы» формой промежуточной аттестации является дифф. зачет. Оценивание качества освоения дисциплины производится с использованием рейтинговой системы.

#### Дифференцированный зачет

Оценка	Количество баллов	Описание
Неудовлетворительно	0 – 7	теоретическое содержание курса не освоено, необходимые практически навыки и умения не сформированы, выполненные учебные задания содержат грубые ошибки, дополнительная самостоятельная работа над курсом не приведет к существенному повышению качества выполнения учебных заданий
Удовлетворительно	8 – 11	теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки и умения сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками
Хорошо	12 – 13	теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки и умения сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено количеством баллов, близким к максимальному
Отлично	14 – 15	теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки и умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки

## Особенности допуска

Допуск к дифф. зачету производится при достижении прогресса не менее 80% в курсе Moodle (сайт [ves.etu.ru](http://ves.etu.ru)), выполнении 9 практических работ в Moodle на оценку "зачтено".

Дифф. зачет проводится в виде тестирования в электронной образовательной среде Moodle. Тест содержит 15 вопросов с развернутыми ответами. Каждый вопрос теста оценивается в 1 балл в случае правильного ответа.

## 6.2 Оценочные материалы для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине

### Вопросы к дифф.зачету

№ п/п	Описание
1	Создание/завершение потоков.
2	Примитивы синхронизации.
3	Алгоритмы синхронизации.
4	Атомарные снимки регистров.
5	Ошибки параллельного программирования.
6	Профилирование многопоточных приложений.
7	Java.util.concurrent и Fork-Join Framework.
8	OpenMP и Intel TBB.
9	Шаблоны параллельного программирования.
10	Кластерные вычисления.
11	Консенсус.
12	Сети Петри.
13	Оптимизации в компиляторах.
14	Транзакционная память.
15	Асинхронный ввод/вывод.
16	Wait-free MRMW снимок регистров.
17	Средства поиска ошибок.
18	Модель памяти.
19	Lock-free схемы управления памятью.
20	Модель акторов.

## Вариант теста

Вопросы для формирования итогового теста, по результатам которого ставится дифф. зачет.

1. Расположите в порядке увеличения степени абстракции технологии высокопроизводительных вычислений:

- 1) OS Threads, SSE/MMX, C++11 Threads/Java Threads, TBB/Fork-Join framework
- 2) SSE/MMX, OS Threads, TBB/Fork-Join framework, C++11 Threads/Java Threads;
- 3) *SSE/MMX, OS Threads, C++11 Threads/Java Threads, TBB/Fork-Join framework*
- 4) OS Threads, TBB/Fork-Join framework, C++11 Threads/Java Threads, SSE/MMX

2. Что показывает закон Амдала:

1) *Прирост производительности в зависимости от доли последовательного кода и числа вычислительных элементов;*

2) Прирост производительности в зависимости от числа вычислительных элементов и используемой технологии распараллеливания;

3) Число вычислительных элементов, необходимое для достижения заданного уровня роста производительности;

4) Долю последовательного кода, производительность которого невозможно.

3. Как называются функции, внутри которых, в зависимости от используемой технологии, происходит проверка на необходимость завершения работы потока и по результатам проверки возможно выкидывание исключений или завершение потока с применением стека зарегистрированных функций:

- 1) Функции завершения потока;
- 2) join-функции;

3) *Cancellation / interruption points;*

4) Signal handlers.

4. Чем отличается spin mutex от обычного mutex:

1) *Остаётся всегда в user space;*

2) Остаётся всегда в kernel space;

3) Ничем не отличается;

4) Позволяет читателям производить операции параллельно.

5. В каком состоянии находится примитив синхронизации (mutex) в приведённом коде на 1, 3 и 4 строках:

```
0: boost::unique_lock<boost::mutex> lock(mutex);
```

```
1:
```

```
2: while (messageQueue.empty())
```

```
3: conditionVariable.wait(lock);
```

```
4:
```

1) Во всех строках захвачен;

2) Во всех строках освобождён;

3) *Захвачен в 1 и 4 строках, в 3 на время вызова wait освобождается;*

4) Захвачен в 1 строке, в 3 и 4 освобождён.

6. Укажите, какой шаблон проектирования позволяет эффективно организовать потоковую обработку данных:

1) *Thread Pool;*

2) *Double Check;*

3) Recursive Data;

4) Pipeline.



7. Выберите технологию, предоставляющую инструменты работы в терминах задач, скрывая вычислительные потоки внутри:

- 1) OpenMP;
- 2) *MPI*;
- 3) *Intel TBB / ForkJoin framework*;
- 4) Java Threads.

8. Выберите технологию, для которой характерна адресация в терминах групп и коммутаторов, с возможностью создания виртуальных топологий:

- 1) *OpenMP*;
- 2) *MPI*;
- 3) *Intel TBB / ForkJoin framework*;
- 4) Java Threads.

9. Выберите модель памяти, которая гарантирует acquire/release семантику доступа к памяти:

- 1) *Sequential Consistency*;
- 2) *Strong-ordered*;
- 3) *Weak-ordered*;
- 4) *Super-weak*.

10. Выберите оптимизацию JIT-компилятора, которая позволяет уменьшить время, затрачиваемое на захват и освобождение примитива синхронизации, не переходя в некоторых случаях в kernel space:

- 1) Объединение захвата примитивов;
- 2) *Оптимистичный захват*;
- 3) *Адаптивные блокировки*;

#### 4) Замена виртуального вызова.

Весь комплект контрольно-измерительных материалов для проверки сформированности компетенции (индикатора компетенции) размещен в закрытой части по адресу, указанному в п. 5.3

### 6.3 График текущего контроля успеваемости

Неделя	Темы занятий	Вид контроля
1	Тема 1. Работа с потоками	
2		Практическая работа
3	Тема 1. Работа с потоками	
4		Практическая работа
5	Тема 1. Работа с потоками	
6		Практическая работа
7	Тема 2. Параллельное программирование.	
8		Практическая работа
9	Тема 2. Параллельное программирование.	
10		Практическая работа
11	Тема 2. Параллельное программирование.	
12		Практическая работа
13	Тема 3. Управление памятью в многопоточных системах	
14		Практическая работа
15	Тема 3. Управление памятью в многопоточных системах	Практическая работа
16	Тема 3. Управление памятью в многопоточных системах	Практическая работа

### 6.4 Методика текущего контроля

#### на лекционных занятиях

Текущий контроль включает в себя:

– контроль прогресса в ЭОС Moodle (не менее 80%).

#### на практических занятиях

– контроль прогресса в ЭОС Moodle (не менее 80%).

В процессе обучения по дисциплине студент обязан выполнить в ЭОС Moodle 9 практических работ.

Студент должен показать: понимание постановки задачи, подхода к ее решению, верный выбор тех или иных методик решения задачи. Преподаватель определяет глубину понимания теоретического материала, который лежит в основе решения задачи практической работы, а также самостоятельность ее выполнения.

Критерии оценивания практических работ:

«не зачтено» ставится, если основное содержание материала работы не раскрыто, материал не усвоен, допущены грубые ошибки;

«зачтено» ставится, если продемонстрировано усвоение основного содержания материала, работа выполнена полностью и самостоятельно.

### **самостоятельной работы студентов**

Контроль самостоятельной работы студентов осуществляется на лекционных и практических занятиях по методикам, описанным выше.

## 7 Описание информационных технологий и материально-технической базы

Тип занятий	Тип помещения	Требования к помещению	Требования к программному обеспечению
Лекция	Лекционная аудитория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя, персональный компьютер IBM-совместимый Pentium или выше, проектор, экран, меловая или маркерная доска	1) Linux Альт Образование 10 и выше; 2) P7-Офис 7 и выше либо LibreOffice 7 и выше
Практические занятия	Аудитория	Количество посадочных мест, оборудованных компьютерами IBM-совместимыми Pentium или выше, – в соответствии с контингентом, рабочее место преподавателя, проектор, экран, меловая или маркерная доска, персональный компьютер IBM совместимый Pentium или выше	1) Linux Альт Образование 10 и выше; 2) P7-Офис 7 и выше либо LibreOffice 7 и выше
Самостоятельная работа	Помещение для самостоятельной работы	Оснащено компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.	1) Linux Альт Образование 10 и выше; 2) P7-Офис 7 и выше либо LibreOffice 7 и выше

## **8 Адаптация рабочей программы для лиц с ОВЗ**

Адаптированная программа разрабатывается при наличии заявления со стороны обучающегося (родителей, законных представителей) и медицинских показаний (рекомендациями психолого-медико-педагогической комиссии). Для инвалидов адаптированная образовательная программа разрабатывается в соответствии с индивидуальной программой реабилитации.

## ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

<b>№ п/п</b>	<b>Дата</b>	<b>Изменение</b>	<b>Дата и номер протокола заседания УМК</b>	<b>Автор</b>	<b>Начальник ОМОЛА</b>