Приложение № 4
к образовательной программе высшего образования – программе магистратуры
по направлению подготовки 09.04.01 Информатика и вычислительная техника
(направленность (профиль) «Разработка и управление
критической информационной инфраструктурой»)

**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ «НАУЧНО-ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ «СИРИУС»
(АНОО ВО «УНИВЕРСИТЕТ «СИРИУС»)**

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 7835bcdac1c89a3da835a18f6045494f3f404337
Владелец: Некрасова Людмила Андреевна
Действителен: с 08.07.2026 до 08.07.2027

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

«Параллельное программирование»

1. Трудоемкость дисциплины (модуля): 4 з.е.

2. Место дисциплины (модуля) в учебном плане:

Часть, формируемая участниками образовательных отношений, «Профессиональной подготовки» - «Дисциплины (модули) специализации» (Блок дисциплин (модулей) по выбору «Разработка программного обеспечения для критической информационной инфраструктуры») в рамках Блока 1. Дисциплины (модули), реализуется в 3 семестре.

3. Цель дисциплины (модуля): формирование у студентов систематизированных знаний и навыков по организации параллельных вычислений с использованием современных аппаратных платформ, инструментальных средств и технологий разработки ПО, с учетом требований безопасности информации.

4. Задачи дисциплины (модуля):

– сформировать представление о теоретических основах параллельных вычислений (классификация параллельных систем, закон Амдала, ускорение и эффективность), механизмах многопоточности, примитивах синхронизации, lock-free и wait-free алгоритмах, а также о методах профилирования и шаблонах параллельного программирования;

– обеспечить владение практическими навыками разработки многопоточных приложений с использованием POSIX Threads, атомарных операций, условных переменных, алгоритмов синхронизации (тонкая, оптимистичная, неблокирующая), а также инструментами анализа производительности (Valgrind, Intel Parallel Studio) и технологиями кластерных вычислений (MPI);

– научить применять принципы безопасного и эффективного параллельного программирования при создании высокопроизводительных и отказоустойчивых программных продуктов для объектов критической информационной инфраструктуры.

5. Перечень разделов (тем) дисциплины (модуля) и их краткое содержание:

Наименование раздела (темы) дисциплины (модуля)	**Краткое содержание**
Раздел 1. Введение в высокопроизводительные вычисления	1.1. Тенденции развития вычислительных систем, обуславливающие необходимость применения распределённых (параллельных) методов вычислений. Примеры вычислительно ёмких задач из разных областей науки. 1.2. Классификация параллельных систем (SIMD, MISD…, SMP, MPP). 1.3. Современные высокопроизводительные системы: начиная от расширений SSE, через многоядерность к узлам кластеров. 1.4. Понятия ускорения, эффективности (закон Амдала). 1.5. Многопоточность или IPC.
Раздел 2. Создание/завершение потоков	2.1. Механизм запуска потока.

Наименование раздела (темы) дисциплины (модуля)	Краткое содержание
	2.2. Корректное завершение потоков: cancellation points, interrupted exception, примеры кода в glibc. 2.3. Сравнение различных потоков (POSIX, boost, java). 2.4. Проброс исключений между потоками.
Раздел 3. Примитивы синхронизации	3.1. Необходимость синхронизации: простые гонки данных. 3.2. Реализация примитивов синхронизации: алгоритм булочника. 3.3. Виды мьютексов: рекурсивные/нерекурсивные, read/write, spin, futex. 3.4. Корректные захват/освобождение примитивов. 3.5. CAS-операции и атомики. 3.6. Условные переменные: использование wait/notify, Spurious wakeups.
Раздел 4. Алгоритмы синхронизации	4.1. Грубая. 4.2. Тонкая. 4.3. Оптимистичная. 4.4. Ленивая. 4.5. Неблокирующая (параллель с ORM).
Раздел 5. Атомарные снимки регистров	5.1. Классификация алгоритмов: lock-free, wait-free. 5.2. SWMR-регистры. 5.3. Lock-free snapshot. 5.4. Wait-free snapshot.
Раздел 6. Ошибки параллельного программирования	6.1. Основные ошибки многопоточного программирования. Гонки данных (Data Race). Взаимная блокировка (Deadlock). 6.2. Специфические ошибки. Реакция потока на сигнал. Блокировки при fork многопоточных программ. Проблема ABA. Инверсия приоритетов.
Раздел 7. Профилирование многопоточных приложений	7.1. Средства анализа производительности. Утилита time . Intel Parallel Studio. Valgrind (модули callgrind, cachegrind). 7.2. Пример поиска узких мест.
Раздел 8. Шаблоны программирования	8.1. Структурные шаблоны: декомпозиция по задачам. Геометрическая декомпозиция. Recursive Data. Pipeline.

Наименование раздела (темы) дисциплины (модуля)	Краткое содержание
	8.2. Некоторые программные структуры: Parallel loops, Boss/Worker. 8.3. Double check. Local Serializer.
Раздел 9. Кластерные вычисления	9.1. Виды кластерных систем: балансировки нагрузки. 9.2. История и назначение стандарта MPI. 9.3. Обмен сообщениями. 9.4. Взаимодействие процессов: группы и коммуникаторы, операции коллективного взаимодействия процессов, редукция, виртуальные топологии.
Раздел 10. Транзакционная память	10.1. Идея transactional memory. Software transactional memory. Hardware transactional memory. 10.2. Преимущества и круг задач. 10.3. Реализация HTM на линейках кэша. 10.4. Lock teleportation.
Раздел 11. Асинхронный ввод/вывод	11.1. Блокирующий/неблокирующий. 11.2. Синхронный (реактор)/асинхронный (проактор). 11.3. Преимущества асинхронной работы и реализация со стороны операционной системы. 11.4. Библиотеки асинхронного ввода/вывода.

6. Образовательные результаты освоения дисциплины (модуля):

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
ПК-2. Способен разрабатывать и внедрять программные продукты для объектов критической информационной инфраструктуры	ИПК-2.1. Знает основы синтаксиса высокоуровневых и низкоуровневых языков программирования, а также знаком с паттернами разработки программного обеспечения как для микроэлектроники, так и для микросервисных приложений
	ИПК-2.2. Умеет разрабатывать программные продукты с использованием различных языков программирования и внедрять их в объекты критической инфраструктуры на аппаратном и программном уровнях

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
	ИПК-2.3. Владеет методами создания, внедрения, тестирования и обеспечения безопасности программного обеспечения на всех этапах жизненного цикла продукта, в том числе с применением технологий искусственного интеллекта, а также способен разработать программное обеспечение как для микроэлектроники, так и для микросервисных и монолитных систем
ПК-6. Способен проектировать и реализовывать объекты критической информационной инфраструктуры для различных операционных систем и облачных платформ	ИПК-6.1. Знает принципы проектирования информационных систем и объектов критической инфраструктуры, особенности различных операционных систем и облачных платформ
	ИПК-6.2. Умеет реализовывать объекты критической инфраструктуры для различных операционных систем и облачных платформ, учитывая их специфику
	ИПК-6.3. Владеет методами проектирования и реализации объектов критической инфраструктуры, включая установку, настройку и обеспечение их работоспособности с учетом специфики работы в различных операционных системах и облачных платформах