

## Аннотация рабочей программы дисциплины

### **Б1.О.32 «Параллельное и низкоуровневое программирование»**

Курс 2 Семестр 4 Количество з.е. 2

**Объем трудоемкости:** 2 зачетных единиц (72 ч., из них – 32 час. аудиторной нагрузки: лекционных 16 ч., лабораторных работ - 16 ч., 37,8 часов самостоятельной работы, 2 часов КСР, 0,2 часа ИКР.), форма контроля – зачет.

**Цель дисциплины:** формирование у студентов знаний и практических навыков разработки многопоточных и высокопроизводительных приложений на языке C++ с использованием технологий POSIX Threads, OpenMP, CUDA, ROCm и OpenCL. Освоение архитектурных принципов параллельных вычислений на CPU и GPU, а также низкоуровневой оптимизации под современные процессоры (SSE, AVX).

#### **Задачи дисциплины:**

1. Изучить основные принципы параллельного программирования и синхронизации потоков.
2. Освоить API POSIX Threads и средства параллелизма C++.
3. Научиться использовать OpenMP для распараллеливания вычислений на уровне потоков и циклов.
4. Изучить принципы векторизации вычислений с применением SSE и AVX-инструкций.
5. Освоить основы GPU-программирования на CUDA, ROCm и OpenCL.
6. Развить навыки профилирования и оптимизации кода под архитектуру CPU и GPU.

#### **Место дисциплины в структуре ООП ВО:**

Дисциплина «Параллельное и низкоуровневое программирование» относится к базовой части блока Б1 **Дисциплины (модули)**.

Для изучения дисциплины студент должен владеть знаниями, умениями и навыками по дисциплинам: Программирование, Алгоритмы и структуры данных, Операционные системы, с которыми **дисциплина связана логически и содержательно-методически**.

Дисциплина в значительной степени взаимодействует для формирования компетенций с дисциплинами:

- Технологии обработки больших данных;
- Современные методы компьютерного зрения;
- Подготовка данных машинного обучения;
- ИИ в робототехнике.

**Результаты обучения (знания, умения, опыт, компетенции):**

**Содержание и структура дисциплины:**

- ОПК-3** Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности
- ОПК-3.2** Способен участвовать в разработке и реализации программных продуктов на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности  
Знает подходящие методы проектирования параллельных приложений.  
Умеет реализовывать лабораторные работы с pthreads, std::thread, OpenMP, CUDA.  
Умеет аргументированно объясняет решения по структуре программы и способам распараллеливания.
- ОПК-7** Способен разрабатывать алгоритмы и программы, пригодные для практического применения
- ОПК-7.1** Разрабатывает эффективные алгоритмы, формализует задачи и оценивает сложность алгоритмов  
Умеет разрабатывать алгоритмы и программы с применением современных технологий параллельного программирования и оптимизации для CPU/GPU,
- ОПК-7.2** Реализует программный код на выбранном языке программирования, соответствующий стандартам качества и готовый к интеграции в реальные системы  
Умеет применять отечественные и международные инструменты (компиляторы, библиотеки, профайлеры) и  
Владеет технологиями и обосновывает выбор технологий для лабораторных работ и проектов.
- PL-1 Б** *Способен применять язык программирования Python для решения задач в области ИИ*
- PL-1.4** *Проектирует системы распределённых вычислений на Python для эффективной обработки большого количества задач*  
Оптимизирует код с использованием библиотек для научных вычислений. Способен применять основные функции фреймворка PySpark, может самостоятельно построить процесс обработки больших данных с использованием Airflow
- PL-2 П** *(П) Способен применять JVM-совместимые языки программирования для решения задач в области ИИ*
- PL-2.1** *Разрабатывает и отлаживает прикладные решения разного уровня сложности и для широкого круга конечных пользователей с использованием JVM-совместимых языков программирования, тестирует, испытывает и оценивает качество таких решений*  
Понимает модель памяти Java и способен поддерживать приложения с высоким параллелизмом и конкуренцией. Понимает алгоритмы сборки мусора и способен оптимизировать сборку мусора.
- PL-3 Б** *(Б) Способен применять языки программирования C/C++ для решения задач в области ИИ*
- PL-3.1** *Разрабатывает и отлаживает эффективные многопоточные решения на C++, тестирует, испытывает и оценивает качество таких решений*  
Знает основы синтаксиса языка.

Знает общие принципы параллельных вычислений и понимает проблемы, возникающие при распараллеливании алгоритмов.

Проводит распараллеливание простого алгоритма с применением OpenMP, стандартных библиотек C/C++ или др.

Решает проблемы одновременного доступа к данным из нескольких потоков, грамотно применяет атомарные операции и механизм блокировок.

Оценивает производительность, умеет профилировать код и устраняет найденные узкие места.

Понимает возможности и ограничения встроенных систем.

Находит и использует библиотеки, соответствующие решаемой задаче.

Знает основы синтаксиса языка C/C++, основы построения систем ИИ, общие принципы параллельных вычислений.

Умеет использовать средства отладки и профилирования кода, находить участки кода, ограничивающие производительность системы.

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.

Разделы дисциплины, изучаемые в 4 семестре (очная форма)

№	Наименование разделов (тем)	Всего	Количество часов			
			Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	
1	2	3	4	5	6	7
1.	Введение в параллельное программирование. Потоки и процессы в ОС.	8	2		2	4
2.	POSIX Threads (pthreads): создание, управление и синхронизация потоков.	8	2		2	4
3.	Параллелизм в C++: std::thread, mutex, future, async.	8	2		2	4
4.	OpenMP: директивы распараллеливания, секции, редукции, профилирование	8	2		2	4
5.	Векторизация и оптимизация вычислений: SIMD, SSE, AVX.	8	2		2	4
6.	GPU-программирование: основы архитектуры CUDA, модель потоков и памяти.	8	2		2	4
7.	OpenCL и ROCm: кроссплатформенные вычисления на GPU.	8	2		2	4
8.	Оптимизация и профилирование параллельных приложений на CPU и GPU.	13,8	2		2	9,8
<b>ИТОГО по разделам дисциплины</b>		<b>69,8</b>	<b>16</b>		<b>16</b>	<b>37,8</b>
Контроль самостоятельной работы (КСР)		2				
Промежуточная аттестация (ИКР)		0,2				
Подготовка к текущему контролю		0				
<b>Общая трудоемкость по дисциплине</b>		<b>72</b>				

Примечание: Л – лекции, КСР – контрольные и самостоятельные работы, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

**Курсовые проекты или работы.**

Не предусмотрены учебным планом

**Вид аттестации:** ЛР, контрольная работа, зачет.

Автор Приходько Т.А. – кандидат технических наук, доцент кафедры вычислительных технологий;

Нигодин Е.А. – старший преподаватель кафедры вычислительных технологий.