

Аннотация рабочей программы дисциплины

Б1.О.32 «Параллельное и низкоуровневое программирование»

Курс 3 Семестр 5 Количество з.е. 2

Объем трудоемкости: 2 зачетных единиц (72 ч., из них – 50 час. аудиторной нагрузки: лекционных 16 ч., лабораторных работ - 34 ч., 19,8 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-4	Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности
ОПК-4.1	Аргументировано применяет современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения

- ОПК-4.2 Знает принципы работы современных технологий параллельного программирования (pthreads, std::thread, OpenMP, CUDA)
Ориентируется в современных положениях и концепциях прикладного и системного программного обеспечения, архитектуры компьютеров и сетей (в том числе и глобальных), технологии создания и сопровождения программных продуктов и программных комплексов
Умеет использовать их для реализации параллельных приложений в ходе лабораторных работ, аргументированно объясняя выбор методов распараллеливания и структуры программы.
- ОПК-5** **Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения**
- ОПК-5.1 Аргументированно применяет методы проектирования, разработки и реализации программных продуктов и программных комплексов в различных областях человеческой деятельности
Умеет разрабатывать алгоритмы и программы с применением современных технологий параллельного программирования и оптимизации для CPU/GPU.
- ОПК-5.2 Использует инструментальные, программные и аппаратные средства измерений для оценки качества программного обеспечения
Владеет способами использования отечественные и международные инструменты (компиляторы, библиотеки, профайлеры) и обосновывает выбор технологий для лабораторных работ и проектов.
- PL-3 П* *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.
- PL-3.2* *Разрабатывает и отлаживает системы ИИ на C++ под конкретные аппаратные платформы с ограничениями по вычислительной мощности, в том числе для встроенных систем*
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.
- PL-3.3* *Разрабатывает и отлаживает решения на C++, использующие GPU и FPGA для массовой параллелизации вычислений в рамках общей системы ИИ, с применением как готовых решений, так и разработкой своих*
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 5 семестре (очная форма)

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа СРС
			Л	ПЗ	ЛР	
1	2	3	4	5	6	7
1.	Введение в параллельное программирование. Поток и процессы в ОС.	8	2		4	2
2.	POSIX Threads (pthreads): создание, управление и синхронизация потоков.	8	2		4	2
3.	Параллелизм в C++: std::thread, mutex, future, async.	8	2		4	2
4.	OpenMP: директивы распараллеливания, секции, редукции, профилирование	8	2		4	2
5.	Векторизация и оптимизация вычислений: SIMD, SSE, AVX.	8	2		4	2
6.	GPU-программирование: основы архитектуры CUDA, модель потоков и памяти.	8	2		4	2
7.	OpenCL и ROCm: кроссплатформенные вычисления на GPU.	8	2		4	2
8.	Оптимизация и профилирование параллельных приложений на CPU и GPU.	13,8	2		6	5,8
ИТОГО по разделам дисциплины		69,8	16		34	19,8
Контроль самостоятельной работы (КСР)		2				
Промежуточная аттестация (ИКР)		0,2				
Подготовка к текущему контролю		0				
Общая трудоемкость по дисциплине		72				

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.