

Аннотация рабочей программы дисциплины

Б1.О.32 «Параллельное и низкоуровневое программирование»

Направление

подготовки/специальность **02.03.03 Математическое обеспечение и администрирование информационных систем**

(код и наименование направления подготовки/специальности)

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)**.

Дисциплина в качестве пререквизитов требует уверенного владения языком C/C++ (указатели, управление памятью, рекурсия, функции), знаний базовых алгоритмов и структур данных (оценка сложности, массивы, списки, деревья), а также понимания принципов работы операционных систем (процессы, потоки, синхронизация, планировщик). Постреквизитами выступают дисциплины, опирающиеся на эффективные параллельные и низкоуровневые вычисления, среди которых «Технологии обработки больших данных» (распределённая и многопоточная обработка), «Современные методы компьютерного зрения» (GPU-свёртки, параллельная фильтрация), «Подготовка данных машинного обучения» (векторизация и ускорение на CPU/GPU), а также «ИИ в робототехнике» (вычисления реального времени, low-level оптимизация).

Связи разделов дисциплины с пререквизитами и постреквизитами:

Связь	Дисциплина (актуальное название из учебного плана)	Раздел(ы) зависящие от дисциплины	текущей от данной	РПД, данной	Характер зависимости
-------	--	---	-------------------------	----------------	-------------------------

Пререквизит	Основы программирования	Разделы 2–5 (работа с потоками, указатели, память в C/C++), разделы 6–7 (написание GPU-ядер)	Критический (без C/C++ невозможно)
Пререквизит	Операционные системы	Раздел 1 (потоки vs процессы), раздел 2 (синхронизация POSIX), раздел 3 (mutex, deadlock), раздел 8 (профилирование)	Сильный
Постреквизит	Технологии обработки больших данных	Раздел 1 (параллельная обработка), разделы 2–4 (многопоточное ускорение), раздел 8 (масштабирование)	Сильный
Постреквизит	Технологии компьютерного зрения	Раздел 5 (SIMD для фильтров), раздел 6 (CUDA для свёрток), раздел 7 (OpenCL/ROCm), раздел 8 (оптимизация под GPU)	Сильный
Постреквизит	Глубокое обучение	Раздел 5 (векторизация матричных операций), раздел 6 (GPU-ускорение нейросетей), раздел 8 (профилирование пайплайнов)	Сильный
Постреквизит	Разработка ИИ агентов	Раздел 1 (многопоточное взаимодействие агентов), раздел 2–3 (синхронизация), раздел 8 (оптимизация real-time)	Сильный
Постреквизит	Высоконагруженные приложения	Раздел 2–4 (потокбезопасность), раздел 5 (SIMD), раздел 8 (профилирование и балансировка нагрузки)	Сильный

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

Содержание и структура дисциплины:

- ОПК-3** Способен понимать и применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения
- ОПК-3.1** Аргументировано применяет современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения
 Умеет выбирать подходящие методы проектирования параллельных приложений, реализует лабораторные работы с pthreads, std::thread, OpenMP, CUDA
 Владеет способами решения и аргументированно объясняет по структуре программы и способам распараллеливания.
- 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** 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-4** Способен применять языки программирования C/C++ для решения задач в области ИИ
- PL-4.1** 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.
- PL-4.2** PL-4.2. Разрабатывает и отлаживает системы ИИ на C++ под конкретные аппаратные платформы с ограничениями по вычислительной мощности, в том числе для встроенных систем
- Знает основы синтаксиса языка, основы построения систем ИИ. Пишет небольшие программы для обработки данных. Понимает возможности и ограничения встроенных систем.
- Понимает методы оптимизации моделей (квантование, сжатие весов модели и пр.) и вычислений ИИ. Находит и использует библиотеки, соответствующие решаемой задаче

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.

Разделы дисциплины, изучаемые в 4 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. 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

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	СРС
1	2	3	4	5	6	7
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
ИТОГО по разделам дисциплины		69,8	16		16	37,8
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.