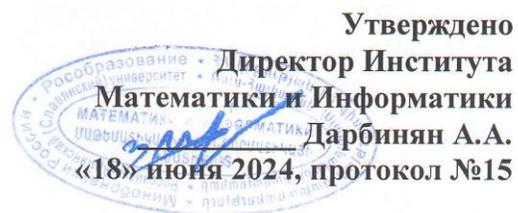


**ГОУ ВПО Российско-Армянский (Славянский)  
университет**



**УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ДИСЦИПЛИНЫ**

**Наименование дисциплины: Параллельное программирование**

**Авторы:** *канд.физ.-мат.наук Саргсян Севак Сеникович*

**Направление подготовки: 01.04.02 Прикладная математика и информатика**

# 1. АННОТАЦИЯ

- 1.1.** Краткое описание содержания данной дисциплины;
- 1.2.** Семинар рассматривает основные проблемы и задачи разработки системного программного обеспечения, а также вопросы стандартизации процессов разработки и обеспечения качества программных систем. Основная форма работы семинара, это доклады его участников, на которых происходит обсуждение рассматриваемых научных и практических вопросов разработки системного программного обеспечения.
- 1.3.** Трудоемкость в академических кредитах и часах, формы итогового контроля (экзамен/зачет);  
2-ой семестр - 6 ЗЕТ- экзамен,
- 1.4.** Взаимосвязь дисциплины с другими дисциплинами учебного плана специальности (направления)  
Для прохождения этого курса изучение других дисциплин не требуются.
- 1.5.** Результаты освоения программы дисциплины:

<b>Код компетенции</b>	<b>Наименование компетенции</b>	<b>Код индикатора достижения компетенций</b>	<b>Наименование индикатора достижений компетенций</b>
<b>УК-2</b>	Способен управлять проектом на всех этапах его жизненного цикла	<b>УК-2.1</b>	Определяет этапы жизненного цикла проекта и выстраивает последовательность их реализации
		<b>УК-2.2</b>	Формулирует проблему, на решение которой направлен проект, грамотно определяет цель проекта.
		<b>УК-2.3</b>	Проектирует решение конкретных задач проекта, выбирая оптимальный способ их решения.

<b>УК-6</b>	Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки	<b>УК-6.1</b>	Применяет рефлексивные методы в процессе оценки разнообразных ресурсов, используемых для решения задач самоорганизации и саморазвития
		<b>УК-6.2</b>	Определяет цели и приоритеты собственной деятельности и способы их достижения
		<b>УК-6.3</b>	Планирует результаты собственной деятельности с учетом необходимых ресурсов
<b>ОПК-2</b>	Способен совершенствовать и реализовывать новые математические методы решения прикладных задач	<b>ОПК-2.1</b>	Обладает знаниями о существующих математических методах, применяемых для решения прикладных задач.
		<b>ОПК-2.2</b>	Демонстрирует умение использования математического языка и математической символики, построения цепочки рассуждений, формулировки математических утверждений для решения прикладных задач.
		<b>ОПК-2.3</b>	Имеет практический опыт совершенствования и реализации различных

			математических методов решения прикладных задач
<b>ПК-2</b>	способностью разрабатывать и анализировать концептуальные и теоретические модели решаемых научных проблем и задач	<b>ПК-2.1</b>	Применяет современные методы и инструменты для представления результатов научно-исследовательской деятельности
		<b>ПК-2.2</b>	Умеет определять реалистические цели научный проектов в области программирования
		<b>ПК-2.3</b>	Определяет основные направления научный исследований в области программирования и применяемых в них подходов

## 2. УЧЕБНАЯ ПРОГРАММА

### 2.1. Цели и задачи дисциплины

Научный семинар является основой образовательной программы, поскольку помогает выстроить студенту индивидуальную траекторию обучения. В рамках курса происходит подбор темы индивидуального обучения, подбор проектных групп и консультантов.

### 2.2. Трудоемкость дисциплины и виды учебной работы (в академических часах и зачетных единицах)

Виды учебной работы	Всего, в акад. часах	Распределение по семестрам					
		I сем	II сем	III сем	IV сем.	V сем	VI сем.
1	2	3	4	5	6	7	8
<b>1.Общая трудоемкость изучения дисциплины по семестрам, в т. ч.:</b>	<b>324</b>						
1.1.Аудиторные занятия, в т. ч.:	<b>154</b>						
1.1.1.Лекции	<b>58</b>	<b>34</b>					
1.1.2.Практические занятия, в т. ч.	<b>96</b>	<b>72</b>					

1.2. Самостоятельная работа, в т. ч.:	107	74					
1.3. Другие методы и формы занятий	63	36					
Итоговый контроль (Экзамен, Зачет, диф. зачет - указать)		Экзамен	Диф.зачет				

### 2.3. Содержание дисциплины

#### 2.3.1. Тематический план и трудоемкость аудиторных занятий (модули, разделы дисциплины и виды занятий) по рабочему учебному плану

Разделы и темы дисциплины	Всего (ак. часов)	Лекции (ак. часов)	Практ. занятия (ак. часов)	Семинары (ак. часов)	Лабор. (ак. часов)	Другие виды занятий (ак. часов)
1	2=3+4+5+6+7	3	4	5	6	7
<b>Модуль 1.</b>						
Введение в CUDA: История и основы	2	2				
Архитектура GPU и модель программирования CUDA	2	2				
Написание первых программ на CUDA	2	2				
Оптимизация производительности CUDA-	2	2				
Инструменты для отладки и профилирования CUDA-кода	2	2				
Расширенные техники оптимизации: память, потоковые мультипроцессоры	2	2				
Примеры и приложения ускоренных вычислений в науке и промышленности	2	2				
Введение в сложные структуры данных: дерево, граф, хэш-таблица	2	2				
Деревья и их разновидности: AVL, красно-черные деревья, B-деревья	2	2				
Графы и алгоритмы работы с ними: поиск в глубину, поиск	2	2				

Хэш-таблицы и методы разрешения коллизий	2	2				
Skip List и ее применение	2	2				
Динамические структуры данных: куча, двойная очередь, кольцевой буфер	2	2				
Примеры использования сложных структур данных в	2	2				
Основы безопасности программного обеспечения:	2	2				
Методы статического анализа кода для выявления	2	2				
Практика безопасного кодирования и стандарт OWASP	2	2				
Инструменты динамического анализа и тестирования на проникновение	2	2				
Методы шифрования и обеспечения конфиденциальности данных	2	2				
Безопасность веб-приложений и защита от распространенных атак (SQL Injection, XSS)	2	2				
Реагирование на инциденты и управление уязвимостями	2	2				
Основы параллельного программирования: термины и концепции	2	2				
Модели параллелизма: SIMD, MIMD, SPMD	2	2				
Инструменты и библиотеки для параллельного	2	2				

Многопоточность в Java и Python: практика и примеры	2	2				
Параллельное программирование на GPU:	2	2				
Синхронизация и коммуникация в параллельных системах	2	2				
Оптимизация производительности параллельных программ	2	2				
Введение в философию разработки программного обеспечения	2	2				
Принципы и методологии разработки ПО: Agile, Scrum, Kanban	2	2				
Архитектура и проектирование ПО: паттерны проектирования и лучшие практики	2	2				
Этические и социальные аспекты разработки ПО	2	2				
Методы статического и динамического анализа программного обеспечения	2	2				
Автоматизация тестирования: инструменты и стратегии	2	2				
Методы реверс-инжиниринга и анализ исполняемых файлов	2	2				
Профилирование и оптимизация	2	2				
Основы машинного обучения: модели, алгоритмы и задачи	2	2				
Классификация, регрессия и кластеризация	2	2				

Введение в глубокое обучение: нейронные сети и их применение	2	2				
Использование библиотек машинного обучения: TensorFlow, PyTorch, Scikit-Learn	2	2				
Основы теории компиляторов: лексический анализ, синтаксический анализ	2	2				
Построение синтаксических деревьев и промежуточного	2	2				
Оптимизация кода на уровне компиляции	2	2				
Генерация машинного кода и архитектурные особенности	2	2				
Современные тенденции в разработке ПО	2	2				
Практические аспекты DevOps и CI/CD	2	2				
Разработка микросервисов и использование Docker, Kubernetes	2	2				
Введение в облачные вычисления и использование облачных платформ	2	2				
Управление проектами в области разработки ПО	2	2				
Введение в блокчейн и его применение в разработке ПО	2	2				
Тестирование и отладка параллельных и распределенных систем	2	2				
Введение в квантовые вычисления и их перспективы	2	2				



результующей оценке промежуточных контролей								
Вес итоговой оценки 2-го промежуточного контроля в результирующей оценке промежуточных контролей							1	
Вес результирующей оценки промежуточных контролей в результирующей оценке итогового контроля								0.5
<b>Вес итогового контроля (Экзамен/зачет) в результирующей оценке итогового контроля</b>								0.5
	$\Sigma = 1$							

### 3. Теоретический блок

#### 3.1. Материалы по теоретической части курса

3.1.1. Брайан Керниган, Деннис Ритчи, Язык программирования Си, 2-е издание

Бьерн Страуструп, Язык программирования C++, 2-е издание

Герберт Шилдт, C++. Базовый курс

3.1.2. Учебное пособие:

«Ալգորիթմներ և ալգորիթմական լեզուներ» Գործնական պարապմունքների մեթոդական ձեռնարկ

[https://rau.am/uploads/post/editor\\_image/C\\_C++%20practice%20\(1\)\\_1717589821.pdf](https://rau.am/uploads/post/editor_image/C_C++%20practice%20(1)_1717589821.pdf)

### 4. Фонды оценочных средств.

#### 4.1. Материалы по практической части курса

4.1.1. Учебно-методические пособия;

Մարգարիտի Ա., Ալիսյանի Շ.

#### 4.2. Вопросы и задания для самостоятельной работы студентов

В системе ejudge

#### 4.3. Перечень экзаменационных вопросов

1. Представление целых и вещественных чисел в двоичной форме. Прямой, обратный и дополнительный код. IEEE-754 формат. Порядок представления

байтов (big and little endian). Размеры типов в языке C. Знаковые и беззнаковые числа. Приведение типов.

Литература:

- Книга Computer Systems Programmers perspective, вторая глава.
- Книга Зубков С. В, первая глава
- Книга В.И. Юров: глава 4, глава 17

2. Процессор x86, регистры. Формат данных. Пересылка данных (mov). Регистр eflags. Арифметические и логические операции. Сдвиги. Переполнение. Флаги OF, CF, ZF. Отображение из C в ассемблер и наоборот.

Литература:

- Книга Computer Systems Programmers perspective: 3.1 - 3.6
- Книга В.И. Юров: глава 7, 8, 9

3. Флаги OF, CF, ZF, SF. Команды передачи управления. Инstrukция jmp и условный переход. Представление операторов условного перехода языка Си в ассемблере.

Оператор switch.

Литература:

- Книга Computer Systems Programmers perspective: 3.1 - 3.6
- Intel reference manual
- Книга В.И. Юров: глава 10

4. Организация циклов. Инstrukция loop. Представление операторов цикла языка Си в ассемблере: do-while, while, for. Инstrukция условной пересылки (Conditional Move Instructions)

Литература:

- Книга Computer Systems Programmers perspective: 3.1 - 3.6
- Книга В.И. Юров: глава 10

5. Представление программы в памяти. Стек и локальные переменные.

Расположение глобальных и статических переменных. Организация вызова функций. Передача аргументов, адрес возврата. Фрейм стека. Различные соглашения о вызовах.

Литература:

- Книга Computer Systems Programmers perspective: 3.7
- Книга Зубков С. В, 5.2, 5.3

- Книга В.И. Юров: глава 15

6. Массивы и указатели. Представление одномерных и двумерных массивов.  
Структуры данных в ассемблере.

Литература:

- Книга Computer Systems Programmers perspective: 3.1 - 3.6

- Книга В.И. Юров: глава 13

7. Процессор x87. Организация процессора и его регистры. Основные команды процессора x87.

Литература:

- Книга В.И. Юров: Глава 17

8. RISC-V

Литература:

- <https://riscv.org/wp-content/uploads/2017/05/riscv-spec-v2.2.pdf>
- <https://github.com/riscv/riscv-bitmanip/releases/download/1.0.0/bitmanip-1.0.0.pdf>