

1. Цели и задачи освоения дисциплины

Цель дисциплины - приобретение студентами теоретических знаний и устойчивых навыков практической работы со средствами параллельного и распределенного программирования.

Основная задача дисциплины: подготовка студентов к работе по созданию и поддержке современных распределенных информационных систем.

2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина относится к разделу «Дисциплины по выбору вариативной части» учебного плана 01.03.02 Прикладная математика и информатика, изучается в 7 семестре, форма промежуточной аттестации – экзамен.

Для успешного овладения курса студенты должны иметь удовлетворительные знания, читаемые в учебных курсах «Языки и методы программирования», «Обобщенное программирование», «Структуры и алгоритмы обработки данных», «Операционные системы».

Формы работы студентов в ходе изучения дисциплины предусмотрены лекционные, практические (семинарские) занятия, выполнение домашних работ.

Самостоятельная работа студентов, предусмотренная учебным планом, выполняется в ходе семестра в форме завершения проектов, теоретически проанализированных на практическом (семинарском) занятии и выполнение этапов которых начато на семинарских занятиях.

Отдельные темы теоретического курса прорабатываются студентами самостоятельно в соответствии с планом самостоятельной работы и конкретными заданиями преподавателя с учетом индивидуальных особенностей студентов.

Виды текущего контроля – проверка домашних заданий, индивидуальная защита выполненных проектов.

3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями выпускников)

После изучения данной дисциплины студенты приобретают знания, умения и владения (навыки), соответствующие результатам основной профессиональной образовательной программы.

Формируемые компетенции	Планируемые результаты
-------------------------	------------------------

<p>(код компетенции, уровень (этап) освоения) (последний – при наличии в карте компетенции)</p>	<p>обучения по дисциплине (модулю), характеризующие этапы формирования компетенций</p>
<p>ОПК-3 – способностью к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям</p>	<p>35 (ОПК-3) Знать: Знание основных классов и особенностей архитектуры современных вычислительных систем, необходимое для разработки эффективных программно-аппаратных комплексов с использованием современных инструментальных средств и технологий параллельного программирования *)</p> <p>У4(ОПК-3) Уметь: применять технологии параллельного программирования для повышения производительности вычислений при решении практических задач</p>
<p>ПК-1 – способностью собирать, обрабатывать и интерпретировать данные современных научных исследований, необходимые для формирования выводов по соответствующим научным исследованиям</p>	<p>32(ПК-1) Знать: методы формирования показателей и критериев эффективности оценки информационного обеспечения и прикладных процессов в соответствии с профилем подготовки</p>
<p>ПК-7 – способностью к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения</p>	<p>34(ПК-7) Знать: способы оптимизации программного кода **)</p> <p>35(ПК-7) Знать: основные способы создания параллельных приложений, методы их программирования и тестирования</p> <p>38(ПК-7) Знать: основные теоретические модели параллельных программ, надежность и масштабируемость параллельных программ</p> <p>У7(ПК-7) Уметь: проектировать и реализовывать многопользовательские программные серверы на базе архитектуры «клиент – сервер»</p>

*) Результат обучения сформулирован на основании требований профессионального стандарта: 06.001 «Программист», для выполнения обобщенной трудовой функции D: «Разработка требований и проектирование программного обеспечения»

**) Результат обучения сформулирован на основании требований профессионального стандарта: 06.015 «Специалист по информационным системам» для выполнения обобщенной трудовой функции С: «Выполнение работ и управление работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы»

4. Объем дисциплины в зачетных единицах с указанием количества академических или астрономических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Объем дисциплины составляет 6 зачетных единицы, всего 216 часов, из которых:

- **68 час** составляет контактная работа обучающегося с преподавателем:
 - 34 часов – лекционные занятия,
 - 34 часов – практические занятия,
- **112 часов** – самостоятельные занятия,
- **36 часов** – мероприятия промежуточного контроля (экзамен в 7 семестре).

5. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических или астрономических часов и виды учебных занятий

Наименование и краткое содержание разделов и тем дисциплины Форма промежуточной аттестации по дисциплине	Всего (часы)	Контактная работа (работа во взаимодействии с преподавателем), часы из них					Самостоятельная работа обучающегося, часы, из них			
		Лекционные занятия	Семинарские занятия	Практические занятия	Лабораторные занятия	Учебные занятия, направленные на проведение текущего контроля успеваемости (коллоквиумы, практические контрольные занятия и др.)*	Всего	Выполнение домашних заданий	Подготовка рефератов и т. п.	Всего
1. Архитектура высокопроизводительных ЭВМ. Векторные и суперскалярные процессоры. SMP и MPP компьютеры, NUMA архитектура. Кластеры ЭВМ, локальные сети ЭВМ. Особенности, отличия, области применения.	34	8		6			14	20		20
2. Архитектура приложений "клиент-сервер". Основные понятия, способы реализации, примеры применений.	12	2		4		ПР-1	6	6		6
3. Распараллеливание вычислений внутри одного процесса - нити (threads). Преимущества и недостатки по сравнению с несколькими процессами. Способы	36	4		3		ПР-1	7	29		29

<p align="center">Наименование и краткое содержание разделов и тем дисциплины Форма промежуточной аттестации по дисциплине</p>	<p align="center">Всего (часы)</p>	<p align="center">Контактная работа (работа во взаимодействии с преподавателем), часы из них</p>					<p align="center">Самостоятельн ая работа обучающегося, часы, из них</p>		
		<p align="center">Лекционные занятия</p>	<p align="center">Семинарские занятия</p>	<p align="center">Практические занятия</p>	<p align="center">Лабораторные занятия</p>	<p align="center">Учебные занятия, направленные на проведение текущего контроля успеваемости (коллоквиумы, практические контрольные занятия и др.)*</p>	<p align="center">Всего</p>	<p align="center">Выполнение домашних заданий</p>	<p align="center">Подготовка рефератов и т.п.</p>
<p>обеспечения надежности при использовании нитей (threads). Методы синхронизации нитей приложения – критическая секция. Нитебезопасный код. Реализация нитей в технологии .NET. Основные свойства и методы, способ использования</p>									
<p>4. Основные механизмы синхронизации и межпроцессной связи в современных операционных системах. Механизмы синхронизации - критическая секция, мьютексы, семафоры. Механизмы межпроцессных соединений (IPC) операционных систем. Классификация, отличия и области применения. Реализация в Windows NT и UNIX. Блок общей памяти (Shared Memory). Область применимости в различных вычислительных системах. Файлы, отображаемые в память. Именованные каналы (Named Pipes). Реализация архитектуры "клиент – сервер" с помощью именованных каналов. Программные сокетты (Sockets). Пространство</p>	14	3	2		ПР-1	5	9		9

Наименование и краткое содержание разделов и тем дисциплины Форма промежуточной аттестации по дисциплине	Всего (часы)	Контактная работа (работа во взаимодействии с преподавателем), часы из них					Самостоятельная работа обучающегося, часы, из них			
		Лекционные занятия	Семинарские занятия	Практические занятия	Лабораторные занятия	Учебные занятия, направленные на проведение текущего контроля успеваемости (коллоквиумы, практические контрольные занятия и др.)*	Всего	Выполнение домашних заданий	Подготовка рефератов и т.п.	Всего
имен, порты, режим связи. Реализация архитектуры "клиент – сервер" с помощью сокетов. Использование сокетов для гетерогенных информационных систем.										
5. Параллельные вычисления и параллельные алгоритмы. Классификация параллельных алгоритмов, их основные показатели. Формула Амдаля. Граф алгоритма и параллельные вычисления. Графовые и операторные модели параллельных программ.	22	3		8		ПР-1	11	11	11	
6. Основные программные средства для параллельных вычислений - PVM, MPI, OpenMP, Linda. Характеристики, область использования. Интерфейс передачи сообщений (MPI). Общая архитектура, основные вызовы для прикладных программ. Сообщения типа «точка-точка». Коллективные сообщения, пользовательские типы данных, группы вычислителей, пользовательская топология	24	8		4		ПР-1	12	12	12	

<p align="center">Наименование и краткое содержание разделов и тем дисциплины Форма промежуточной аттестации по дисциплине</p>	<p align="center">Всего (часы)</p>	<p align="center">Контактная работа (работа во взаимодействии с преподавателем), часы из них</p>					<p align="center">Самостоятельн ая работа обучающегося, часы, из них</p>		
		<p align="center">Лекционные занятия</p>	<p align="center">Семинарские занятия</p>	<p align="center">Практические занятия</p>	<p align="center">Лабораторные занятия</p>	<p align="center">Учебные занятия, направленные на проведение текущего контроля успеваемости (коллоквиумы, практические контрольные занятия и др.)*</p>	<p align="center">Всего</p>	<p align="center">Выполнение домашних заданий</p>	<p align="center">Подготовка рефератов и т.п.</p>
<p>вычислителей. Открытая многопроцессорная модель вычислений (OpenMP). Общая архитектура, организация прикладных программ. Библиотека разработки параллельных программ Linda.</p>									
<p>7. Параллельные вычисления в среде .Net. Библиотека Parallel Extensions. Особенности параллельных вычислений в среде .Net. Основные классы библиотеки, их применение и особенности. Базовые паттерны параллельного программирования: “fork & join”, “aggregate”, “map & reduce”.</p>	17	2		3		ПП-1	5	12	12
<p>8. Промежуточный контроль (экзамен)</p>	36								
<p>Итого</p>	216	34		34			68	112	112

