

РЕКОМЕНДУЕМАЯ ФОРМА для разработчиков
основных профессиональных образовательных программ
при реализации ОС МГУ на основе ФГОС 3+

Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования

Московский государственный университет имени М.В. Ломоносова

Механико-математический факультет

Кафедра газовой и волновой динамики



УТВЕРЖДАЮ
Заведующий кафедрой
/Нигматулин Р.И./
«_10_» _июня_2019г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Наименование дисциплины (модуля):

Технологии параллельного программирования

наименование дисциплины (модуля)

Уровень высшего образования:

специалитет

Направление подготовки (специальность):

01.05.01 Фундаментальные математика и механика

(код и название направления/специальности)

Направленность (профиль) ОПОП: В-ПД

Фундаментальная механика

(если дисциплина (модуль) относится к вариативной части программы)

Форма обучения:

очная

очная, очно-заочная

Рабочая программа рассмотрена и одобрена
на заседании кафедры газовой и волновой динамики
(протокол №_15_, «_10_» ___июня_20_19 года)

Москва 2019

На обратной стороне титула:

Рабочая программа дисциплины (модуля) разработана в соответствии с самостоятельно установленным МГУ образовательным стандартом (ОС МГУ) для реализуемых основных профессиональных образовательных программ высшего образования по направлению подготовки / специальности «Фундаментальные математика и механика» (программы бакалавриата, магистратуры, реализуемых последовательно по схеме интегрированной подготовки; программы специалитета; программы магистратуры) в редакции приказа МГУ от 30 декабря 2016 г.

Год (годы) приема на обучение _____ 2015 _____

1. Место дисциплины (модуля) в структуре ОПОП ВО: вариативная часть ООП. Является специальной дисциплиной (спецкурсом) для студентов 3-6 годов обучения, специализирующихся в данной научной области или смежной научной области, спецкурсом по выбору кафедры.

Освоение дисциплины необходимо для сдачи экзаменов по основной и смежной специальностям, сдача выпускных экзаменов, написание курсовых и дипломных работ, статей и научных отчетов.

2. Входные требования для освоения дисциплины (модуля), предварительные условия (если есть): базовые знания работы с вычислительной техникой.

3. Результаты обучения по дисциплине (модулю), соотнесенные с требуемыми компетенциями выпускников.

Компетенции выпускников (коды)	Планируемые результаты обучения по дисциплине (модулю), соотнесенные с компетенциями
СПК-1	<i>Знать архитектуры современных компьютеров, основные технологии параллельного программирования для многопроцессорных вычислительных систем. Уметь интерпретировать параллельные алгоритмы для многопроцессорных вычислительных систем. Владеть основными технологиями параллельного программирования для многопроцессорных вычислительных систем.</i>
СПК-2	<i>Знать основы построения параллельных разностных схем для решения задач специальных областей механики. Уметь конструировать параллельные разностные схемы для решения задач специальных областей механики. Владеть основными принципами построения параллельных разностных схем для решения задач специальных областей механики.</i>
СПК-3	<i>Знать способы построения параллельных алгоритмов для различных современных архитектур компьютеров. Уметь строить параллельные алгоритмы для различных современных архитектур компьютеров. Владеть способами построения параллельных алгоритмов для различных современных архитектур компьютеров.</i>

4. Формат обучения: очный, лекционные и практические занятия.

5. Объем дисциплины (модуля) составляет 3 з.е., в том числе 36 академических часов, отведенных на контактную работу обучающихся с преподавателем, 72 академических часа на самостоятельную работу обучающихся.

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

Наименование и краткое содержание разделов и тем дисциплины (модуля), Форма промежуточной аттестации по дисциплине (модулю)	Всего (часы)	В том числе			
		Контактная работа (работа во взаимодействии с преподавателем) Виды контактной работы, часы			Самостоятельная работа обучающегося, часы <i>(виды самостоятельной работы – эссе, реферат, контрольная работа и пр. – указываются при необходимости)</i>
		Занятия лекционного типа*	Занятия семинарского типа*	Всего	
Тема 1: Введение. Архитектура фон Неймана. Скалярная и суперскалярная архитектуры. CISC, RISC и VLIW архитектуры. SMP и ccNUMA архитектуры.	6	2		2	4
Тема 2: Многопроцессорные архитектуры. Кластеры, суперЭВМ. Примеры. Иерархия памяти. Кэш память. Когерентность кэша.	6	2		2	4
Тема 3: Классификация Флинна. Ускорение вычислений. Закон Амдала.	6	2		2	4
Тема 4: Параллельные ЭВМ с общей и локальной памятью. Топология вычислительных сетей. Примеры топологий. GRID.	6	2		2	4
Тема 5: Параллельные алгоритмы. Зернистость алгоритма. Граф сдваивания. Компиляторы и эффективность программ.	6	2		2	4
Тема 6: Технологии параллельного программирования. Последовательная и	6	2		2	4

параллельные модели. Параллелизм данных.					
Тема 7: Проектирование коммуникаций. Парадигмы параллельного программирования. «Обедающие философы», «Стена Фокса».	4	2		2	2
Тема 8: Планирование сообщений. Нити и процессы. Синхронизация нитей. Операции обмена сообщениями. Критические секции.	6	2		2	4
Тема 9: Система ПараЛаб. Организация вычислений в этой системе.	4	2		2	2
Промежуточная аттестация: контрольная работа	2				2
Тема 10: Разработка параллельных алгоритмов. Основные языки параллельного программирования C++ и Fortran.	6	2		2	4
Тема 11: Структура программ на C++/Fortran. Основные типы данных. Массивы.	6		2	2	4
Тема 12: Программирование на C++/Fortran. Организация циклов.	6		2	2	4
Тема 13: Условный оператор if. Оператор switch.	6		2	2	4
Тема 14: Построение последовательных программ на C++/Fortran'e	6		2	2	4
Тема 15: Технология параллельного программирования OpenMP. Инициализация OpenMP на C++/Fortran. Задание числа нитей.	6		2	2	4
Тема 16: Построение параллельных циклов.	6		2	2	4
Тема 17: Создание параллельных секций.	6		2	2	4
Тема 18: Организация параллельных вычислений на C++/Fortran с использованием OpenMP.	6		2	2	4
Промежуточная аттестация: контрольная работа	2				2
Итого	108				72

7. Фонд оценочных средств (ФОС) для оценивания результатов обучения по дисциплине (модулю)

Вопросы к экзамену:

1. Архитектура фон Неймана. Скалярная и суперскалярная архитектуры. CISC, RISC и VLIW архитектуры. SMP и ccNUMA архитектуры.
2. Многопроцессорные архитектуры. Кластеры, суперЭВМ. Примеры. Иерархия памяти. Кэш память. Когерентность кэша. Иерархия памяти.
3. Классификация Флинна. Ускорение вычислений. Закон Амдала.
4. Параллельные ЭВМ с общей и локальной памятью. Топология вычислительных сетей. Примеры топологий. GRID.
5. Параллельные алгоритмы. Зернистость алгоритма. Граф сдваивания. Компиляторы и эффективность программ.
6. Технологии параллельного программирования. Последовательная и параллельные модели. Параллелизм данных.
7. Проектирование коммуникаций. Парадигмы параллельного программирования. «Обедающие философы», «Стена Фокса».
8. Планирование сообщений. Нити и процессы. Синхронизация нитей. Операции обмена сообщениями. Критические секции.
9. Технология параллельного программирования OpenMP. Библиотека, директивы и функции OpenMP.
10. Как откомпилировать любую последовательную программу с включением опций поддержки технологии OpenMP и запустить ее с использованием нескольких нитей. Сколько нитей будет реально исполнять операторы данной программы?
11. Организация параллельных вычислений с использованием OpenMP.
12. Определите, сколько процессоров доступно в вашей системе для выполнения параллельной части программы, и займите каждый из доступных процессоров выполнением одной нити в рамках общей параллельной области.
13. Инициализация OpenMP. Задание числа нитей.
14. Может ли программа на OpenMP состоять только из параллельных областей? Только из последовательных областей?
15. Построение параллельных циклов.
16. Определите, какое максимальное количество нитей позволяет породить для выполнения параллельных областей программы ваша система.
17. Создание параллельных секций.
18. Могут ли функции `omp_get_thread_num()` и `omp_get_num_threads()` вернуть одинаковые значения на нескольких нитях одной параллельной области?
19. Синхронизация параллельных вычислений.
20. Чем отличается нить-мастер от всех остальных нитей?
21. В каких случаях может быть необходимо использование опции `if` директивы `parallel`?
22. При помощи трёх уровней вложенных параллельных областей породите 8 нитей (на каждом уровне параллельную область должны исполнять 2 нити). Посмотрите, как будет исполняться программа, если запретить вложенные параллельные области.

23. Чем отличаются директивы single и master?
24. Может ли нить-мастер выполнить область, ассоциированную с директивой single?
25. Может ли нить с номером 1 выполнить область, ассоциированную с директивой master?
26. Можно ли распределить между нитями итерации цикла без использования директивы for (do ... [end do])?
27. Можно ли одной директивой распределить между нитями итерации сразу нескольких циклов?
28. Возможно ли, что при статическом распределении итераций цикла нитям достанется разное количество итераций?
29. Могут ли при повторном запуске программы итерации распределяемого цикла достаться другим нитям? Если да, то при каких способах распределения итераций?
30. Для чего может быть полезно указывать параметр chunk при способе распределения итераций guided?
31. Можно ли реализовать параллельные секции без использования директив sections (sections ... end sections) и section?
32. Как при выходе из параллельных секций разослать значение некоторой локальной переменной всем нитям, выполняющим данную параллельную область?

Билеты формируются по каждой из тем, то есть 1 вопрос соответствует одному билету.

Пример:

Билет 1	Архитектура фон Неймана. Скалярная и суперскалярная архитектуры. CISC, RISC и VLIW архитектуры. SMP и ccNUMA архитектуры.
Билет 2	Многопроцессорные архитектуры. Кластеры, суперЭВМ. Примеры. Иерархия памяти. Кэш память. Когерентность кэша. Иерархия памяти.

Текущий контроль – задачи для самостоятельного решения с последующим обсуждением.

ШКАЛА И КРИТЕРИИ ОЦЕНИВАНИЯ результатов обучения (РО) по дисциплине (модулю)				
Оценка	2	3	4	5
РО и соответствующие виды оценочных средств				
Знания	Отсутствие знаний	Фрагментарные знания	Общие, но не	Сформированные

<i>(виды оценочных средств: устные и письменные опросы и контрольные работы, тесты, и т.п.)</i>			структурированные знания	систематические знания
Умения <i>(виды оценочных средств: практические контрольные задания, написание и защита рефератов на заданную тему и т.п.)</i>	Отсутствие умений	В целом успешное, но не систематическое умение	В целом успешное, но содержащее отдельные пробелы умение (допускает неточности не принципиального характера)	Успешное и систематическое умение
Навыки (владения, опыт деятельности) <i>(виды оценочных средств: выполнение и защита курсовой работы, отчет по практике, отчет по НИР и т.п.)</i>	Отсутствие навыков (владений, опыта)	Наличие отдельных навыков (наличие фрагментарного опыта)	В целом, сформированные навыки (владения), но используемые не в активной форме	Сформированные навыки (владения), применяемые при решении задач

8. Ресурсное обеспечение:

Перечень основной и дополнительной учебной литературы:

1. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб.: БХВ-Петербург. (2002).
2. Немнюгин С., Стесик О. Параллельное программирование для многопроцессорных вычислительных систем – СПб.: БХВ-Петербург. (2002).
3. Гергель В.П. Теория и практика параллельных вычислений. Издательство INTUIT, Нижний Новгород, (2007).
4. Рыбакин Б.П. Численные методы для многопроцессорных ЭВМ. Издательство МолдГУ, Кишинев, (2008).
5. Антонов А. С. Параллельное программирование с использованием технологии OpenMP
6. Артемов И.Л. FORTRAN: основы программирования. Диалог МИФИ, (2007)

7. Таненбаум Э. Архитектура компьютера. – СПб.: Питер. (2002).
8. Миллер Р., Боксер Л. Последовательные и параллельные алгоритмы. БИНОМ, (2006).
9. Богачев К.Ю. Основы параллельного программирования. - М.: БИНОМ. Лаборатория знаний, (2003).
10. Grama, A., Gupta, A., Kumar V. Introduction to Parallel Computing. – Harlow, England: Addison-Wesley. (2003, 2nd edn.).
11. Pacheco, P. Parallel Programming with MPI. - Morgan Kaufmann. (1996).

- Перечень ресурсов информационно-телекоммуникационной сети «Интернет»:
- www.mathnet.ru

9. Язык преподавания.
русский

10. Преподаватель (преподаватели).
д.ф.-м.н., проф. Рыбакин Б.П.

11. Автор (авторы) программы.
д.ф.-м.н., проф. Рыбакин Б.П.