

*Федеральное государственное автономное образовательное учреждение высшего образования  
«Российский университет дружбы народов»*

*Факультет физико-математических и естественных наук*

Рекомендовано МСЧН  
02.00.00 «Компьютерные и  
информационные науки»

## **РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**Наименование дисциплины**

**Параллельное программирование**

**Направление**

**02.03.02 — Фундаментальная информатика и информационные технологии**

*(указываются код и наименования направления(ий))*

*подготовки (специальности (ей) и/или профилей (специализаций))*

**Квалификация выпускника бакалавр**

*(указывается квалификация (степень) выпускника в соответствии с ОС ВО РУДН)*

### 1. Цели и задачи дисциплины:

Целью дисциплины является введение учащихся в предметную область современных параллельных вычислений.

В процессе преподавания дисциплины решаются следующие задачи:

- освоение архитектурных принципов реализации параллельной обработки в вычислительных машинах;
- изучение методов и языковых механизмов конструирования параллельных программ;
- овладение параллельными вычислительными методами.

### 2. Место дисциплины в структуре ООП:

Цикл, к которому относится дисциплина: Блок 1 «Дисциплины (модули)», часть, формируемая участниками образовательных отношений

В таблице № 1 приведены предшествующие и последующие дисциплины, направленные на формирование компетенций дисциплины в соответствии с матрицей компетенций ОП ВО.

Таблица № 1

#### Предшествующие и последующие дисциплины, направленные на формирование компетенций

№ п/п	Шифр и наименование компетенции	Предшествующие дисциплины	Последующие дисциплины (группы дисциплин)
Универсальные компетенции			
1			
Общепрофессиональные компетенции			
1	ОПК-2, ОПК-3, ОПК-6	Основы программирования Технология программирования Алгоритмы и анализ сложности	-
Профессиональные компетенции			
	ПК-1	Основы программирования Технология программирования Алгоритмы и анализ сложности	-

### 3. Требования к результатам освоения дисциплины:

Процесс изучения дисциплины направлен на формирование следующих компетенций

ОПК-2; ОПК-3; ОПК-6; ПК-1

(в соответствии с ОС ВО РУДН)

ОПК-2. Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности.

- ОПК-2.1 Знает основные положения и концепции в области программирования, архитектуру языков программирования, знает основную терминологию, знаком с содержанием Единого Реестра Российских программ
- ОПК-2.2 Умеет анализировать типовые языки программирования, составлять программы
- ОПК-2.3 Имеет практический опыт решения задач анализа, интеграции различных типов программного обеспечения

ОПК-3. Способен 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.

- ОПК-3.1 Знает методы теории алгоритмов, методы системного и прикладного программирования, основные положения и концепции в области математических, информационных и имитационных моделей
- ОПК-3.2 Умеет соотносить знания в области программирования, интерпретацию прочитанного, определять и создавать информационные ресурсы глобальных сетей, образовательного контента, средств тестирования систем
- ОПК-3.3 Имеет практический опыт применения разработки программного обеспечения

ОПК-6. Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности.

- ОПК-6.1 Знает базовые принципы цифровых технологий и методов, необходимых в профессиональной деятельности в области фундаментальной информатики и информационных технологий для: изучения и моделирования объектов профессиональной деятельности, анализа данных, представления информации и пр.
- ОПК-6.2 Умеет применять необходимые в профессиональной деятельности цифровые технологии и методы в области фундаментальной информатики и информационных технологий для: изучения и моделирования объектов профессиональной деятельности, анализа данных, представления информации и пр.
- ОПК-6.3 Владеет необходимыми в профессиональной деятельности технологиями и методами в области фундаментальной информатики и информационных технологий для: изучения и моделирования объектов профессиональной деятельности, анализа данных, представления информации и пр.

ПК-1 Способен разрабатывать и отлаживать программный код

- ПК-1.1 Знает основы программирования; современные объектно-ориентированные языки программирования; современные структурные языки программирования; языки современных бизнес-приложений
- ПК-1.2 Умеет кодировать на языках программирования; тестировать результаты кодирования
- ПК-1.3 Владеет навыками разработки кода информационной системы; навыками верификации кода информационной системы

В результате изучения дисциплины студент должен:

***Знать:***

- общие реализации параллельной обработки в вычислительных машинах;
- методы и языковые механизмы конструирования параллельных программ;
- параллельные вычислительные методы;
- корректные постановки классических задач;

***Уметь:***

- быстро находить, анализировать и грамотно контекстно обрабатывать научно-техническую информацию, приводя ее к проблемно-задачной форме;
- использовать ресурсы Интернета;
- определять общие формы, закономерности, инструментальные средства отдельной предметной области - операционных систем;
- понять поставленную задачу;
- формулировать результат;
- грамотно пользоваться языком предметной области;
- ориентироваться в постановках задач;

- увидеть прикладной аспект в решении научной задачи, грамотно представить и интерпретировать результат;
- проанализировать результат и скорректировать математическую модель, лежащую в основе задачи.

**Владеть:**

- способность применять в научно-исследовательской и профессиональной деятельности базовые знания в области фундаментальной и прикладной математики и естественных наук;
- значительными навыками самостоятельной научно-исследовательской работы
- фундаментальной подготовкой в области и компьютерных наук;
- значительными навыками самостоятельной работы с компьютером, программирования, использования методов обработки информации;
- базовыми знаниями в области информатики и современных информационных технологий, навыками использования программных средств и навыками работы в компьютерных сетях;
- навыками самостоятельного построения алгоритма и его анализа;
- навыками контекстной обработки информации;
- методом алгоритмического моделирования при анализе постановок математических задач;
- методами математического и алгоритмического моделирования при анализе и решении прикладных и инженерно-технических проблем;
- методами алгоритмического моделирования при анализе управленческих задач в научно-технической сфере;
- параллельными вычислительными методами.

**4. Объем дисциплины и виды учебной работы**

Общая трудоемкость дисциплины составляет **4** зачетные единицы.

Вид учебной работы	Всего часов	Семестры
		Сем. 7, мод. 13 (D)
<b>Аудиторные занятия (всего)</b>	54	54
Лекции	18	18
<i>Практические занятия (ПЗ)</i>		
<i>Лабораторные работы (ЛР)</i>	36	36
<b>Самостоятельная работа (всего)</b>	90	90
Общая трудоемкость	час	<b>144</b>
	зач. ед.	<b>4</b>

**5. Содержание дисциплины**

**5.1. Содержание разделов дисциплины**

№ п/п	Наименование раздела дисциплины	Содержание раздела
1	Язык программирования Fortran и параллельные алгоритмы	1. Язык Фортран Основные сведения о языке Фортран. История развития. Его преимущества в области научных вычислений по сравнению с другими языками высокого уровня. Структура программы. Типы данных. Встроенные операции и функции. Операторы управления и

		ветвления. Массивы и работа с ними: описание массивов, задание массивов, динамические массивы, основные функции работы с массивами как матрицами. Ввод и вывод
		2. Параллельные алгоритмы Параллельные алгоритмы: суммирование массивов, кумулятивная сумма и произведение элементов массивов, параллельное перемножение матриц Параллельные алгоритмы вычисления однократных и двукратных определенных интегралов. Итерационные методы решения СЛАУ
2	Технологии параллельного программирования	1. Технология OpenMP Основные сведения. OpenMP и Fortran. Нити и процессы. Параллельные и последовательные области. Параллельные циклы и параллельные области. Автоматическое распараллеливания циклов 2. Параллелизация метода Монте-Карло с помощью OpenMP Основные сведения о метода Монте-Карло, генерация случайных чисел в Fortran, параллелизация генерации случайных чисел, способ вычисления площади и определенного интеграла с помощью метода Монте-Карло.

## 5.2. Разделы дисциплин и виды занятий

№ п/п	Наименование раздела дисциплины	Лекц.	Практ. зан.	Лаб. зан.	Контроль	СРС	Все-го час.
1.	Язык программирования Fortran и параллельные алгоритмы	9		18	13	32	72
2.	Технологии параллельного программирования	9		18	14	31	72
	<i>Всего часов</i>	<b>18</b>		<b>36</b>	<b>27</b>	<b>63</b>	<b>144</b>

## 6. Лабораторный практикум Не предусмотрен

№ п/п	№ раздела дисциплины	Наименование лабораторных работ	Трудо-емкость (час.)
1	1	Написание программ на языке Fortran	9
2	1	Реализация параллельных алгоритмов	9
3	2	Написание параллельных программ с помощью технологии OpenMP	9
4	2	Написание параллельных программ использующих метод Монте-Карло.	9
	<b>Итого:</b>		<b>36</b>

## 7. Практические занятия (семинары)

Не предусмотрены

## 8. Материально-техническое обеспечение дисциплины:

Мультимедийная аудитория для проведения лекционных занятий. Компьютерные



(дисплейные) классы с доступом к сети Интернет и электронно-образовательной среде Университета для выполнения обучающимися практических работ по дисциплине, для проведения обучающимися самостоятельной работы и компьютерного тестирования обучающихся (при необходимости).

## **9. Информационное обеспечение дисциплины**

а) программное обеспечение:

- ОС Linux, офисный пакет LibreOffice (лицензия MPL-2.0), ПО для просмотра pdf (например, evince (лицензия GPL-2+ CC-BY-SA-3.0)), GNU Midnight Commander (Лицензия GNU GPL 3), редакторы emacs (лицензия GPL) или vi (лицензия BSD), компилятор gcc (лицензия GPL), библиотеки MPI (Лицензия GPL-2), OpenMP (Лицензия ( UoI-NCSA MIT ) MIT LLVM-Grant)

б) базы данных, информационно-справочные и поисковые системы:

- Сайт библиотеки РУДН <http://lib.rudn.ru/>
- ТУИС <http://esystem.pfur.ru/>

## **10. Учебно-методическое обеспечение дисциплины:**

**а) основная литература**

1. Параллельное программирование: Лабораторные работы [Текст/электронный ресурс] : Учебное пособие / М.Н. Геворкян, А.В. Королькова, Д.С. Кулябов. - Электронные текстовые данные. - М. : Изд-во РУДН, 2014. - 87 с. : ил. - ISBN 978-5-209-06152-6 : 109.31. Режим доступа: <http://lib.rudn.ru/ProtectedView/Book/ViewBook/4366>
2. Гергель, В.П. Теория и практика параллельных вычислений : учебное пособие / В.П. Гергель. - Москва : Интернет-Университет Информационных Технологий, 2007. - 424 с. : ил.,табл. - (Основы информационных технологий). - ISBN 978-5-9556-0096-3 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=233067>
3. Антонов А.С. "Параллельное программирование с использованием технологии OpenMP: Учебное пособие". — М.: Изд-во МГУ, 2009. - 77 с. ISBN 978-5-211-05702-9 Режим доступа: <https://parallel.ru/sites/default/files/info/parallel/openmp/OpenMP.pdf>
4. Левин, М.П. Параллельное программирование с использованием OpenMP : учебное пособие / М.П. Левин. - Москва : Интернет-Университет Информационных Технологий, 2008. - 120 с. - (Основы информационных технологий). - ISBN 978-5-94774-857-4 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=233111>

б) дополнительная литература

5. Основы параллельного программирования [Текст] / К.Ю. Богачев. - М. : БИНОМ. Лаборатория знаний, 2015. - 342 с. : ил. - ISBN 978-5-94774-037-0 : 270.00. (ЕТ 4)
6. Параллельное программирование для многопроцессорных вычислительных систем [Текст] / С.А. Немнюгин, О.Л. Стесик. - СПб. : БХВ-Петербург, 2002. - 400 с. : ил. - ISBN 5-94157-188-7 : 138.93. (ЕТ 10)

## **11. Методические рекомендации по организации изучения дисциплины:**

Учебным планом на изучение дисциплины отводится один семестр. Промежуточный контроль знаний предусматривает: проведение проверочной работы (тестирование) в середине семестра, подготовку и сдачу практических работ в течение семестра. В качестве итогового контроля знаний предусмотрен контроль знаний в форме тестирования.

Первый модуль составляют:

- Основные сведения о языке Фортран. История развития. Его преимущества в области научных вычислений по сравнению с другими языками высокого уровня. Структура программы. Типы данных. Встроенные операции и функции. Операторы управления и

ветвления. Массивы и работа с ними: описание массивов, задание массивов, динамические массивы, основные функции работы с массивами как матрицами. Ввод и вывод

- Основные сведения. Матричные разложения и их использование для численных расчетов. Система наименований подпрограмм LAPACK. Матричные разложения. Процедуры SGESV, SGETRF, SGEEV, SGESVD.

Второй модуль составляют:

- Основные сведения. OpenMP и Fortran. Нити и процессы. Параллельные и последовательные области. Параллельные циклы и параллельные области. Автоматическое распараллеливания циклов
- Основные сведения. Способы распараллеливания численных методов. Основные процедуры MPI. Типы данных MPI. Способы передачи сообщений. Прием и передача сообщений процессами.

### **11.1 Методические указания по самостоятельному освоению теоретического материала по дисциплине**

Рекомендуется по указанным темам изучить литературу, указанную в п. 10 программы дисциплины и учебно-методические материалы в ТУИС (<http://esystem.pfur.ru>).

### **11.2 Методические указания по выполнению практических работ**

- Срок сдачи: первые две работы сдаются до конца октября, вторые две до конца семестра (до 25-29 декабря).
- В случае сдачи не в срок за работу ставится максимум 8 баллов.
- Для получения максимального балла необходимо защитить работы (устно рассказать в чем заключались задания лабораторной, показать написанные программы, запустить их на компьютере в ДК или на личном ноутбуке, показать построенные графики замеров производительности и ответить на вопросы преподавателя).
- В случае сдачи без защиты необходимо оформить отчет и загрузить на портал ТУИС. Максимальный балл в этом случае составляет 10 баллов.
- Исходные коды программ без отчета и без защиты, загруженные на ТУИС, не проверяются и не оцениваются.

### **11.3. Методические указания по подготовке к контрольным мероприятиям**

Контрольные мероприятия по дисциплине проводятся в форме тестирования в ТУИС (<http://esystem.pfur.ru>). Итоговый контроль в форме теста проводится по темам всех разделов дисциплины. Вопросы для подготовки к итоговому тестированию размещены в соответствующем разделе ТУИС (<http://esystem.pfur.ru>).

## **12. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю)**

ФОС по дисциплине представлен в приложении к данной программе. Программа составлена в соответствии с требованиями ОС ВО РУДН.

### **Разработчики:**

доцент кафедры прикладной информатики  
и теории вероятностей, к.ф.-м.н.,



М. Н. Геворкян

### **Руководитель программы**

#### **Заведующий кафедрой**

прикладной информатики  
и теории вероятностей, проф.



К.Е. Самуйлов

*Федеральное государственное автономное образовательное учреждение высшего образования  
«Российский университет дружбы народов»*

*Факультет физико-математических и естественных наук*

*Кафедра прикладной информатики и теории вероятностей*

# **ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

## **ПО УЧЕБНОЙ ДИСЦИПЛИНЕ**

Параллельное программирование

(наименование дисциплины)

02.03.02 — Фундаментальная информатика и информационные технологии

(код и наименование направления подготовки)

бакалавр

Квалификация (степень) выпускника



## Паспорт фонда оценочных средств по дисциплине Параллельное программирование

Направление 02.03.02 — Фундаментальная информатика и информационные технологии

Код контролируемой компетенции или ее части	Контролируемый раздел дисциплины	Контролируемая тема дисциплины	Формы контроля уровня освоения ООП			Баллы темы	Баллы раздела
			Выполнение ЛР	Промежуточный тест	Итог. контр. (тест)		
ОПК-2; ОПК-3; ОПК-6; ПК-1	Язык программирования Fortran и параллельные алгоритмы	Fortran	15	20		25	<b>50</b>
		Алгоритмы	15			25	
	Технологии параллельного программирования	OpenMP	15	20		25	<b>50</b>
		Метод Монте-Карло	15			25	
Итого			60	20	20	<b>100</b>	<b>100</b>

Процесс изучения дисциплины направлен на формирование следующих компетенций

ОПК-2; ОПК-3; ОПК-6; ПК-1

*(в соответствии с ОС ВО РУДН)*

ОПК-2. Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности.

- ОПК-2.1 Знает основные положения и концепции в области программирования, архитектуру языков программирования, знает основную терминологию, знаком с содержанием Единого Реестра Российских программ
- ОПК-2.2 Умеет анализировать типовые языки программирования, составлять программы
- ОПК-2.3 Имеет практический опыт решения задач анализа, интеграции различных типов программного обеспечения

ОПК-3. Способен 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.

- ОПК-3.1 Знает методы теории алгоритмов, методы системного и прикладного программирования, основные положения и концепции в области математических, информационных и имитационных моделей
- ОПК-3.2 Умеет соотносить знания в области программирования, интерпретацию прочитанного, определять и создавать информационные ресурсы глобальных сетей, образовательного контента, средств тестирования систем

- ОПК-3.3 Имеет практический опыт применения разработки программного обеспечения

ОПК-6. Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности.

- ОПК-6.1 Знает базовые принципы цифровых технологий и методов, необходимых в профессиональной деятельности в области фундаментальной информатики и информационных технологий для: изучения и моделирования объектов профессиональной деятельности, анализа данных, представления информации и пр.
- ОПК-6.2 Умеет применять необходимые в профессиональной деятельности цифровые технологии и методы в области фундаментальной информатики и информационных технологий для: изучения и моделирования объектов профессиональной деятельности, анализа данных, представления информации и пр.
- ОПК-6.3 Владеет необходимыми в профессиональной деятельности технологиями и методами в области фундаментальной информатики и информационных технологий для: изучения и моделирования объектов профессиональной деятельности, анализа данных, представления информации и пр.

ПК-1 Способен разрабатывать и отлаживать программный код

- ПК-1.1 Знает основы программирования; современные объектно-ориентированные языки программирования; современные структурные языки программирования; языки современных бизнес-приложений
- ПК-1.2 Умеет кодировать на языках программирования; тестировать результаты кодирования
- ПК-1.3 Владеет навыками разработки кода информационной системы; навыками верификации кода информационной системы

## Балльно-рейтинговая система оценки уровня знаний

**Сводная оценочная таблица дисциплины**

Раздел	Тема	Формы контроля уровня освоения ООП			Баллы темы	Баллы раздела
		Выполнение ЛР	Промежуточный тест	Итог. контрол. (тест)		
Язык программирования Fortran и параллельные алгоритмы	Fortran	15	20		25	50
	Алгоритмы	15			25	
Технологии параллельного программирования	OpenMP	15		20	25	50
	Метод Монте-Карло	15			25	
<b>ИТОГО:</b>		<b>60</b>	<b>20</b>	<b>20</b>	<b>100</b>	<b>100</b>

**Таблица соответствия баллов и оценок**

Баллы БРС	Традиционные оценки РФ	Оценки ECTS
95 - 100	5	A
86 - 94		B
69 - 85	4	C
61 - 68	3	D
51 - 60		E
31 - 50	2	FX
0 - 30		F
	Зачет	Passed

### Правила применения БРС

- Раздел (тема) учебной дисциплины считаются освоенными, если студент набрал более 50 % от возможного числа баллов по этому разделу (теме).
- Студент не может быть аттестован по дисциплине, если он не освоил все темы и разделы дисциплины, указанные в сводной оценочной таблице дисциплины.
- По решению преподавателя и с согласия студентов, не освоивших отдельные разделы (темы) изучаемой дисциплины, в течение учебного семестра могут быть повторно проведены мероприятия текущего контроля успеваемости или выданы дополнительные учебные задания по этим темам или разделам. При этом студентам за данную работу засчитывается минимально возможный положительный балл (51 % от максимального балла).
- При выполнении студентом дополнительных учебных заданий или повторного прохождения мероприятий текущего контроля полученные им баллы засчитываются за конкретные темы. Итоговая сумма баллов не может превышать максимального количества баллов, установленного по данным темам.

- График проведения мероприятий текущего контроля успеваемости формируется в соответствии с календарным планом курса. Студенты обязаны сдавать все задания в сроки, установленные преподавателем.
- Время, которое отводится студенту на выполнение мероприятий текущего контроля успеваемости, устанавливается преподавателем. По завершение отведенного времени студент должен сдать работу преподавателю, вне зависимости от того, завершена она или нет.
- Использование источников (в том числе конспектов лекций и лабораторных работ) во время выполнения контрольных мероприятий возможно только с разрешения преподавателя.
- Отсрочка в прохождении мероприятий текущего контроля успеваемости считается уважительной только в случае болезни студента, что подтверждается наличием у него медицинской справки, заверенной круглой печатью в поликлинике № 25, предоставляемой преподавателю не позднее двух недель после выздоровления. В этом случае выполнение контрольных мероприятий осуществляется после выздоровления студента в срок, назначенный преподавателем. В противном случае, отсутствие студента на контрольном мероприятии признается не уважительным.
- Студент допускается к итоговому контролю знаний с любым количеством баллов, набранных в семестре, но при условии, что у студента имеется теоретическая возможность получить за весь курс не менее 31 балла.
- Итоговая контроль знаний оценивается из 15 баллов независимо от числа баллов за семестр.
- Если в итоге за семестр студент получил менее 31 балла, то ему выставляется оценка F и студент должен повторить эту дисциплину в установленном порядке. Если же в итоге студент получил 31-50 баллов, т.е. FX, то студенту разрешается добор необходимого (до 51) количества баллов путем повторного одноразового выполнения предусмотренных контрольных мероприятий, при этом по усмотрению преподавателя аннулируются соответствующие предыдущие результаты. Ликвидация задолженностей проводится в период с 07.02 по 28.02 (с 07.09 по 28.09) по согласованию с деканатом.



## Примерный перечень оценочных средств

п/ п	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
<i><b>Аудиторная работа</b></i>			
2	Тест	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.	База тестовых заданий
5	Лабораторная работа	Система практических заданий, направленных на формирование практических навыков у обучающихся. Предполагает как аудиторное, так и самостоятельное выполнение	Фонд практических заданий
11	Зачет	Финальная оценка ставится исходя из суммы набранных за семестр баллов. На зачетной/экзаменационной неделе студент имеет возможность сдать несданные задания, получив при этом половину баллов.	База тестовых заданий
<i><b>Самостоятельная работа</b></i>			
1	Домашняя работа	В качестве домашней работы предлагается выполнять задания из практических работ и составлять отчет по результатам	Фонд практических заданий

Учебным планом на изучение дисциплины отводится один семестр. В дисциплине предусмотрены лекции, лабораторные работы, контрольные мероприятия по проверке отчетов по практическим работам. В конце семестра проводится итоговый контроль знаний.

Оценивание результатов освоения дисциплины производится в соответствии с больно-рейтинговой системой. По дисциплине предусмотрен зачет.-

Итоговый контроль знаний по дисциплине проводится в форме тестирования, но при необходимости экзамен может проводиться в форме письменного ответа на вопросы из билетов.

## Критерии оценки по дисциплине

*95-100 баллов:*

- полное и своевременное выполнение на высоком уровне лабораторных работ с оформлением отчетов, успешное прохождение контрольных мероприятий, предусмотренных программой курса;
- систематизированное, глубокое и полное освоение навыков и компетенций по всем разделам программы дисциплины;
- использование научной терминологии, стилистически грамотное, логически правильное изложение ответов на вопросы, умение делать обоснованные выводы;
- безупречное владение программным обеспечением, умение эффективно использовать его в постановке и решении научных и профессиональных задач;
- выраженная способность самостоятельно и творчески решать поставленные задачи;
- полная самостоятельность и творческий подход при изложении материала по программе дисциплины;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной программой дисциплины и преподавателем.

*86- 94 балла:*

- полное и своевременное выполнение на хорошем уровне лабораторных работ с оформлением отчетов, успешное прохождение контрольных мероприятий, предусмотренных программой курса;
- систематизированное, глубокое и полное освоение навыков и компетенций по всем разделам программы дисциплины;
- использование научной терминологии, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;
- хорошее владение программным обеспечением, умение эффективно использовать его в постановке и решении научных и профессиональных задач;
- способность самостоятельно решать поставленные задачи в нестандартных производственных ситуациях;
- усвоение основной и дополнительной литературы, рекомендованной программой дисциплины и преподавателем.

*69-85 баллов:*

- своевременное выполнение на хорошем уровне лабораторных работ с оформлением отчетов, прохождение контрольных мероприятий, предусмотренных программой курса;
- хороший уровень культуры исполнения лабораторных работ;
- систематизированное и полное освоение навыков и компетенций по всем разделам программы дисциплины;
- владение программным обеспечением, умение использовать его в постановке и решении научных и профессиональных задач;
- способность самостоятельно решать проблемы в рамках программы дисциплины;
- усвоение основной литературы;

*51-68 баллов:*

- выполнение на удовлетворительном уровне лабораторных работ с оформлением отчетов, прохождение контрольных мероприятий, предусмотренных программой курса;
- систематизированное и полное освоение навыков и компетенций по всем разделам программы дисциплины;
- удовлетворительное владение программным обеспечением, умение использовать

- его в постановке и решении научных и профессиональных задач;
- способность решать проблемы в рамках программы дисциплины;
- удовлетворительное усвоение основной литературы;

*31 - 50 баллов – НЕ ЗАЧТЕНО:*

- не выполнение, несвоевременное выполнение или выполнение на неудовлетворительном уровне лабораторных работ, не прохождение контрольных мероприятий, предусмотренных программой курса;
- недостаточно полный объем навыков и компетенции в рамках программы дисциплины;
- неумение использовать в практической деятельности научной терминологии, изложение ответа на вопросы с существенными стилистическими и логическими ошибками;
- слабое владение программным обеспечением по разделам программы дисциплины, некомпетентность в решении стандартных (типовых) задач;
- способность решать проблемы в рамках программы дисциплины;
- удовлетворительное усвоение основной литературы;

*0-30 баллов, НЕ ЗАЧТЕНО:*

- отсутствие умений, навыков, знаний и компетенции в рамках программы дисциплины;
- невыполнение лабораторных заданий, не прохождение контрольных мероприятий, предусмотренных программой курса; отказ от ответов по программе дисциплины;
- игнорирование занятий по дисциплине по неуважительной причине.

# Комплект экзаменационных билетов

Дисциплина: параллельное программирование

## ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №1

1. Как с помощью встроенных функций вычислить дробную часть вещественного числа?
2. Что такое масштабируемость алгоритма?
3. Как рассчитывается ускорение и эффективность алгоритма?
4. Какова структура программы на фортране?

Составитель /М.Н. Геворкян/

Заведующий кафедрой /К.Е. Самуйлов/

Дисциплина: параллельное программирование

## ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №2

1. В чем заключается основное отличие процесса от потока/нити ?
2. Какие разновидности процедуры `mpi_send()` существуют? В чем их различие друг от друга?
3. Как на практике реализуется парадигма MIMD (Multiple Instructions Multiple Data)?
4. Кратко охарактеризуйте библиотеку LAPACK.

Составитель /М.Н. Геворкян/

Заведующий кафедрой /К.Е. Самуйлов/

Дисциплина: параллельное программирование

## ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №3

1. Как называется встроенная функция, вычисляющая определитель матрицы (двухмерного массива)?
2. Какие встроенные типы данных поддерживает Фортран? Как описать вещественную переменную двойной точности?
3. Какие способы классификации параллельных систем вам известны?
4. Дайте описание процедуры `sgesvd`.

Составитель /М.Н. Геворкян/

Заведующий кафедрой /К.Е. Самуйлов/



Дисциплина: параллельное программирование

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №4

1. Последовательная часть программы (до директивы `mri_init(ierr)`) будет выполняться всеми процессами одновременно или же только одним процессом?
2. На основании чего в MPI можно однозначно идентифицировать тот или иной процесс?
3. Укажите основные различия между синхронным (блокирующим) и асинхронным способами передачи сообщений.
4. Перечислите наиболее используемые директивы OpenMP.

Составитель /М.Н. Геворкян/

Заведующий кафедрой /К.Е. Самуйлов/

Дисциплина: параллельное программирование

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №5

1. Какие преимущества дает технология OpenMP?
2. Оправданно ли использование технологии OpenMP (и вообще многопоточного программирования) при написании пользовательских программ для персонального компьютера?
3. Поясните суть закона Амдала. Какие ограничения он налагает?
4. Кратко опишите суть технологии MPI.

Составитель /М.Н. Геворкян/

Заведующий кафедрой /К.Е. Самуйлов/

Дисциплина: параллельное программирование

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №6

1. Что такое степень параллелизма алгоритма?
2. Какой оператор прерывает виток цикла, и переходит к следующей итерации?
3. Перечислите наиболее используемые функции из библиотеки OpenMP.
4. В чем преимущество модулей?

Составитель /М.Н. Геворкян/

Заведующий кафедрой /К.Е. Самуйлов/

Дисциплина: параллельное программирование

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №7

1. Поясните смысл идеологии Single Program Multiple Data на конкретном примере (можно воспользоваться одной из вышеизложенных программ).
2. Необходимо описать 5 массивов размерности (1:100,-100:1). Как сделать это наиболее кратко?
3. Какие языки программирования поддерживает LAPACK? На каком языке написан сам LAPACK?
4. Какие способы задания значений элементов массива вы знаете?

Составитель /М.Н. Геворкян/

Заведующий кафедрой /К.Е. Самуйлов/

Дисциплина: параллельное программирование

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №8

1. Как будет именоваться процедура по вычислению собственных векторов для комплексной трехдиагональной матрицы?
2. Насколько сложна задача параллелизации алгоритма?
3. Назовите главные отличия OpenMP и MPI. В каких случаях какую из этих технологий предпочтительнее использовать?
4. Опишите основные операторы для ввода-вывода данных. Как записать данные в файл?

Составитель /М.Н. Геворкян/

Заведующий кафедрой /К.Е. Самуйлов/

Дисциплина: параллельное программирование

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №9

1. Что такое динамические массивы и как они задаются?
2. Как реализуется применение OpenMP при написании программ на языке Fortran? На языках C/C++ ?
3. Что такое экстенд, размер и ранг массива?
4. Процедуры LAPACK являются подпрограммами или функциями? Приведите несколько примеров.

Составитель /М.Н. Геворкян/

Заведующий кафедрой /К.Е. Самуйлов/

Дисциплина: параллельное программирование

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №10

1. Какие процедуры в LAPACK можно использовать для вычисления определителя? Почему нет отдельной процедуры для этого?
2. Как правильно проверить равны ли две вещественные переменные?
3. Для чего применяются дополнительные атрибуты при описании переменной?
4. Какова схема названий процедур библиотеки LAPACK?

Составитель /М.Н. Геворкян/

Заведующий кафедрой /К.Е. Самуйлов/

Дисциплина: параллельное программирование

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №11

1. Какие переменные окружения влияют на число создаваемых потоков?
2. Опишите процедуру LAPACK для решения переопределенной или неопределенной системы алгебраических линейных уравнений.
3. Дайте описание процедуры sgeqrf.
4. Что такое сечение и вырезка массива?

Составитель /М.Н. Геворкян/

Заведующий кафедрой /К.Е. Самуйлов/

Дисциплина: параллельное программирование

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №12

1. Почему комплексная переменная двойной точности занимает в памяти 16 байт?
2. В каком случае можно говорить о сверхлинейном ускорении?
3. Какие предпосылки способствовали развитию параллельных вычислений?
4. Чем подпрограмма отличается от функции?

Составитель /М.Н. Геворкян/

Заведующий кафедрой /К.Е. Самуйлов/

# Комплект заданий для промежуточного и итогового контроля знаний

Промежуточный и итоговый контроль знаний по дисциплине проводится в форме компьютерного тестирования.

## Примерный перечень вопросов теста:

- Какие встроенные типы данных поддерживает Фортран? Как описать вещественную переменную двойной точности?
- Почему комплексная переменная двойной точности занимает в памяти 16 байт?
- Для чего применяются дополнительные атрибуты при описании переменной?
- Как спомощью встроенных функций вычислить дробную часть вещественного числа?
- Как правильно проверить равны ли две вещественные переменные?
- Какой оператор прерывает виток цикла, и переходит к следующей итерации?
- Чем подпрограмма отличается от функции?
- Какова структура программы на фортране?
- В чем преимущество модулей?
- Необходимо описать 5 массивов размерности (1:100,-100:1). Как сделать это наиболее кратко?
- Что такое экстенд, размер и ранг массива?
- Что такое сечение и вырезка массива?
- Какие способы задания значений элементов массива вы знаете?
- Как называется встроенная функция, вычисляющая определитель матрицы (двухмерного массива)?
- Что такое динамические массивы и как они задаются?
- Опишите основные операторы для ввода-вывода данных. Как записать данные в файл?
- Какие предпосылки способствовали развитию параллельных вычислений?
- Насколько сложна задача параллелизации алгоритма?
- Какие способы классификации параллельных систем вам известны?
- Что такое степень параллелизма алгоритма?
- Что такое масштабируемость алгоритма?
- Как рассчитывается ускорение и эффективность алгоритма?
- В каком случае можно говорить о сверхлинейном ускорении?
- Поясните суть закона Амдаля. Какие ограничения он налагает?
- Назовите главные отличия OpenMP и MPI. В каких случаях какую из этих технологий предпочтительнее использовать?
- Как на практике реализуется парадигма MIMD (Multiple Instructions Multiple Data)?
- Какие типы данных определены в MPI? Как они соотносятся с типами данных языка Фортран?
- Кратко опишите суть технологии MPI.
- Какие разновидности процедуры `mpi_send()` существуют? В чем их различие друг от друга?
- Укажите основные различия между синхронным (блокирующим) и асинхронным способами передачи сообщений.
- На основании чего в MPI можно однозначно идентифицировать тот или иной процесс?



# Фонд лабораторных работ

Предлагаются к выполнению 4 лабораторных работы. Отчеты по работам выполняются студентом самостоятельно, на занятии студент может получить консультацию и методические указания от преподавателя. Задания лабораторных работ приведены в соответствующих разделах ТУИС (<http://esystem.pfur.ru>) и в приложении к данному документу.

## Методические указания

### Порядок выполнения лабораторной работы и Критерии оценки

- Срок сдачи: первые две лабораторные сдаются до конца октября, вторые две до конца семестра (до 25-29 декабря).
- Оценивается полнота выполнения работы, оформление результатов, полнота ответов на контрольные вопросы, если это предусмотрено заданием.
- В случае сдачи не в срок за лабораторную ставится максимум 8 баллов.
- Для получения максимального балла необходимо защитить лабораторные (устно рассказать в чем заключались задания лабораторной, показать написанные программы, запустить их на компьютере в ДК или на личном ноутбуке, показать построенные графики замеров производительности и ответить на вопросы преподавателя).
- В случае сдачи без защиты необходимо оформить отчет и загрузить на портал ТУИС. Максимальный балл в этом случае составляет 10 баллов.
- Исходные коды программ без отчета и без защиты, загруженные на ТУИС, не проверяются и не оцениваются.

### Лабораторная работа № 1

В первой работе предлагается реализовать несколько алгоритмов и вспомогательных задач, которые часто встречаются в многопоточных параллельных программах. Для выполнения данной лабораторной требуется знакомство с языком Fortran.

#### Задание 1.

При параллельной работе с массивами, часто необходимо поделить большой массив между процессами/потоками. Так как количество доступных процессоров может изменяться в зависимости от конфигурации компьютера на котором запускают программу, то необходимо уметь разделять одномерные и двумерные массивы на равные части автоматически. Так как Fortran поддерживает вырезки из массивов, то достаточно определить граничные индексы каждой вырезки.

- Написать функцию `calc_vector_slice`, которая делит одномерный массив из произвольного числа элементов с произвольными начальным и конечным индексом на  $m$  частей. Если количество элементов не кратное, то последняя часть должна быть большей и со держать остаток элементов. Функция должна быть чистой и иметь интерфейс

приведенный в листинге ниже.

- Написать функцию `calc_matrix_slice`, которая делит двумерный массив из произвольного числа элементов с произвольными начальным и конечным индексом на сетку из блоков. Функция должна быть чистой и иметь интерфейс. Для реализации данной функции удобно использовать функцию из предыдущего пункта.

#### Задание 2.

Реализовать алгоритм быстрой сортировки. Быстрая сортировка основана на пара-

дигме «разделяй и властвуй». Процесс сортировки массива  $A[p \dots q]$  с начальным индексом  $p$  и конечным индексом  $q$  является рекурсивным и состоит из трех стадий.

### **Задание №3**

Реализовать два алгоритма умножения матриц: итерационный и блочный. При реализации блочного алгоритма используйте функцию, определяющую границы блоков, которую вы реализовали в задании №1. Везде, где это возможно, используйте инструкцию `do concurrent` для того, чтобы компилятор мог эффективно оптимизировать циклы. Сравните производительность программы при компиляции с флагом `-O2` и без него. Функции должны поддерживать работу с числами одинарной (`real32`) и двойной (`real64`) точности.

## **Лабораторная работа № 2**

Во второй лабораторной работе необходимо реализовать несколько простых алгоритмов, которые можно распараллелить базовыми средствами OpenMP.

### **Задание №1**

Реализовать подпрограмму `test_omp`, которая принимает в качестве аргументов целочисленный массив `array` и количество потоков, которые необходимо породить с помощью OpenMP. Каждый поток записывает в свою часть массива свой номер. Для деления массива между потоками используете функцию `calc_vector_slice` из первого задания практической №1.

### **Задание №2**

Используя средства OpenMP для редукции действий с массивами, реализуйте параллельные функции по нахождению суммы, минимума и максимума некоторого массива.

### **Задание №3**

Используйте формулу трапеции для реализации функции `trapezoidal`, которая аппроксимирует значение интеграла от скалярной функции  $f(x)$ .

$$\int_a^b f(x) dx \approx \frac{h}{2}(f(a) + f(b)) + \sum_{i=1}^{n-1} \frac{f(x_i)}{2} \cdot h, \quad x_i = a + ih.$$

### **Задание №4**

Используя директивы OpenMP измените функцию по блочному умножению матриц из задания №3 первой практической работы так, чтобы он работал параллельно.

## **Лабораторная работа № 3**

Третья практическая работа состоит всего из двух заданий, однако учащиеся должны самостоятельно придумать тесты для заданий, которые позволят убедиться в корректности работы программ, а также самостоятельно построить графики времени, ускорения и эффективности. Для построения графиков можно пользоваться программами из предыдущих практических работ.

### **Задание №1**

Распараллелить алгоритм Штрассена умножения матриц.

### **Задание №2**

Параллельный алгоритм решета Эратосфена. Для тестирования используйте сайт [the Prime](#)

[pages](#), откуда можно [скачать текстовые файлы](#) со списком простых чисел вплоть до 50 миллионного простого числа. Алгоритм решета Эратосфена см по [ссылке](#) (Википедия)

## Лабораторная работа № 4

В четвертой практической работе все задания связаны с методом Монте-Карло, который эффективно поддается распараллеливанию. Требуется знакомство с Fortran и OpenMP.

### Задание №1

Напишите программу, которая использует встроенный генератор псевдослучайных равномерно распределенных чисел. Программа должна выдавать последовательность равномерно распределенных псевдослучайных чисел из промежутка  $(-1; 1)$  или  $[0; 1)$ . Ваша программа должна распечатывать столбик псевдослучайных чисел в консоль. Для проверки правильности работы вашей программы, следует построить гистограмму. Чем больше чисел вы возьмете для построения этой гистограммы, тем больше гистограмма будет похожа на теоретическую функцию плотности вероятности непрерывного равномерного распределения. Для наглядности рекомендуется построить две гистограммы: одну для 105, а другую для 106 чисел. Для построения гистограмм можете использовать Excel, Gnuplot, Matplotlib или любое другое средство построения графиков.

### Задание №2

Создайте параллельную версию генератора из предыдущего задания. Убедитесь, что каждый поток генерирует свою последовательность случайных чисел. Возможно придется инициализировать генератор для каждого потока отдельно.

### Задание №3

Реализуйте метод Монте Карло в параллельном режиме, используя его для вычисления площади круга. Экономьте память и постарайтесь не сохранять в массив все сгенерированные точки — для решения задачи в этом нет необходимости

### Задание №4

Усовершенствуйте предыдущую программу так, чтобы она вычисляла двукратные интегралы от произвольной подынтегральной функции. Для проверки вашей программы используйте пример из методического пособия или любой интеграл, вычисляемый аналитически. Программа должна работать параллельно и давать выигрыш от увеличения числа нитей.