

ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»

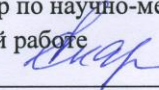
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Кафедра теории упругости и вычислительной математики  
им. академика А.С. Космодамианского



УТВЕРЖДАЮ:

проректор по научно-методической  
и учебной работе

 Е.И. Скафа

«22» апреля 2020 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**  
**«ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ»**

Направление подготовки:	09.03.04 Программная инженерия
Профиль подготовки:	Программная инженерия
Образовательная программа:	бакалавриат
Квалификация:	Академический бакалавр
Форма обучения:	<u>очная, очно-заочная, заочная, в том</u> <u>числе с ускоренным сроком обучения</u> нужное подчеркнуть

Донецк 2020

**УТВЕРЖДАЮ:**

Декан факультета математики  
и информационных технологий

И. А. Моисеенко

«16» апреля 2020

МП

Программа учебной дисциплины «Параллельное программирование» составлена на основании Государственного образовательного стандарта высшего профессионального образования (ГОС ВПО) Донецкой Народной Республики (ДНР) по направлению подготовки 09.03.04 Программная инженерия, утвержденного приказом Министерства образования и науки ДНР от 21 января 2016 г. № 33;

Порядка организации учебного процесса в образовательных организациях высшего профессионального образования Донецкой Народной Республики, утвержденного приказом Министерства образования и науки ДНР № 1171 от «10» ноября 2017 г.;

учебного плана и основной образовательной программы высшего профессионального образования направления подготовки 09.03.04 Программная инженерия, разработанных в ГОУ ВПО «Донецкий национальный университет».

Разработчик:

Доцент кафедры теории упругости  
и вычислительной математики  
им. акад. А.С. Космодамианского

Р.Н. Нескородев

Программа учебной дисциплины утверждена на заседании кафедры теории упругости и вычислительной математики им. акад. А.С. Космодамианского

Протокол № 11 от «09» апреля 2020 г.  
Заведующий кафедрой

В.И. Сторожев

Программа учебной дисциплины одобрена учебно-методической комиссией факультета математики и информационных технологий  
Протокол № 8 от «15» апреля 2020 г.

Председатель учебно-методической  
комиссии факультета

Л.И. Селякова

## 1. ОБЛАСТЬ ПРИМЕНЕНИЯ И МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПРОЦЕССЕ

Учебная дисциплина «Параллельное программирование» относится к вариативной части профессионального блока и входит в состав дисциплин по выбору студента. Для изучения данной учебной дисциплины необходимы знания, умения и навыки, формируемые предшествующими дисциплинами – «Информатика», «Программирование», «Архитектура компьютеров», «Операционные системы», «Тестирование и отладка программного обеспечения», «Алгоритмы и структуры данных».

Этот курс, опираясь на математическую (математический анализ, алгебра, дискретная математика, вычислительная математика и др.) и компьютерную (архитектура компьютеров, программирование, операционные системы, теория алгоритмов) подготовку студентов, закладывает фундамент научно-методической подготовки будущих исследователей в области программной инженерии.

## 2. СТРУКТУРА ДИСЦИПЛИНЫ

<i>Характеристика учебной дисциплины</i>				
Направление подготовки	09.03.04 Программная инженерия			
Профиль	Программная инженерия			
Образовательная программа	Бакалавриат			
Квалификация	Академический бакалавр			
Количество содержательных модулей	1			
Дисциплина базовой / вариативной части образовательной программы	Дисциплина вариативной части по выбору студентов			
Формы контроля (МК, экзамен, зачет)	модульный контроль, экзамен			
Показатели	очная форма обучения		заочная форма обучения	
	нормат. срок	ускор. срок	нормат. срок	ускор. срок
Количество зачётных единиц (кредитов)	4	4	4	–
Год подготовки	4	3	4	–
Семестр	8	6	8	–
Количество часов	144	144	144	–
- лекционных	40	40	10	–
- практических, семинарских	–	–	–	–
- лабораторных	40	40	10	–
- самостоятельной работы	64	64	124	–
в т.ч. индивидуальное задание	24	24	24	–
Недельное количество часов,	14	14	–	–
в т.ч. аудиторных	8	8	–	–

## 3. ОПИСАНИЕ ДИСЦИПЛИНЫ

### Цели и задачи

*Целью изучения дисциплины “Параллельное программирование”* является формирование у студентов практических навыков организации параллельных алгоритмов для многопроцессорных вычислительных систем, в приобретении студентами теоретических и практических знаний в области параллельного программирования.



**Задачи:**

- изучение общих положений и принципов программирования параллельных систем;
- изучение технологий и средств параллельного программирования средствами операционной системы и прикладного окружения;
- изучение методов программирования параллельных приближенных вычислений средствами технологии OpenMP.

**Требования к результатам освоения дисциплины.** Процесс изучения дисциплины «Параллельное программирование» направлен на формирование элементов следующих компетенций в соответствии с ГОС ВПО ДНР по направлению подготовки 09.03.04 Программная инженерия и основной образовательной программы высшего профессионального образования направления подготовки 09.03.04 Программная инженерия (Профиль: Программная инженерия):

**а) общекультурных (ОК):**

- ОК-7 – способность к самоорганизации и самообразованию;

**б) общепрофессиональных (ОПК):**

- ОПК-1 – владение основными концепциями, принципами, теориями и фактами, связанными с информатикой;
- ОПК-2 – владением архитектурой электронных вычислительных машин и систем;
- ОПК-3 – способность применять знания и умения из информатики и программирования к проектированию, конструированию и тестированию программных продуктов;

**в) профессиональных (ПК):****производственно-технологическая деятельность:**

- ПК-1 – готовность применять основные методы и инструменты разработки программного обеспечения;
- ПК-3 – владением навыками использования различных технологий разработки программного обеспечения (ПК-3)

**научно-исследовательская деятельность:**

- ПК-12 – способностью к формализации в своей предметной области с учётом ограничений используемых методов исследования;
- ПК – 13 – готовностью к использованию методов и инструментальных средств исследования объектов профессиональной деятельности

**проектная деятельность:**

- ПК-19 – владением навыками моделирования, анализа и использования формальных методов проектирования и конструирования программного обеспечения;
- ПК-20 – способностью оценивать временную и ёмкостную сложность программного обеспечения

**В результате изучения учебной дисциплины студент должен:****Знать:**

- принципы построения параллельных вычислительных систем;
- основы моделирования и анализа параллельных вычислений;
- основные операции передачи данных между процессами и производные типы данных;
- гибридные вычислительные системы с использованием различных технологий;
- основных команд библиотеки OpenMP для практической реализации на ПК параллельных алгоритмов.

**Уметь:**

- разрабатывать параллельные алгоритмы и программы для решения разного класса задач на компьютерах с общей памятью;

- использовать библиотеки для параллельных программ;
- оценивать трудоемкость параллельных алгоритмов;
- ставить и решать задачи, возникающие в процессе конструирования параллельных программ.

**Владеть:**

- теоретическими знаниями в области организации взаимодействующих процессов;
- навыками параллельного программирования на основе языка высокого уровня;
- средствами параллельного программирования OpenMP;

#### 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ И ФОРМЫ ОРГАНИЗАЦИИ УЧЕБНОГО ПРОЦЕССА

Порядковый номер и тема	Краткое содержание темы
	<b>Содержательный модуль 1</b>
<b>Тема 1.</b> <b>Принципы построения параллельных вычислительных систем.</b>	Параллельные компьютеры и суперЭВМ. Классификация параллельных вычислительных систем. Типы параллелизма. Параллелизм на уровне команд. Конвейер. Суперскалярность. SIMD команды. Параллелизм на уровне задач.
<b>Тема 2.</b> <b>Анализ вычислительной сложности параллельных алгоритмов</b>	Аналитические методы определения вычислительной сложности. Оценка вычислительной сложности для последовательных алгоритмов. Показатели для оценки параллельных алгоритмов: ускорение, эффективность и стоимость параллельных алгоритмов. Экспериментальные методы определения вычислительной сложности.
<b>Тема 3.</b> <b>Разработка параллельных программ</b>	Декомпозиция по функциям и по данным. Планирование параллельных программ. Зависимости данных. Способы синхронизации. Определение стоимости программы с учетом параллельного выполнения. Масштабируемость.
<b>Тема 4.</b> <b>Параллельное программирование при помощи технологии OpenMP</b>	Параллельные и последовательные области. Директива parallel. Распараллеливание оператора цикла. Использование параллельных секций. Синхронизация потоков. Параллельные методы матричного умножения. Вычисление определенного интеграла в многопроцессорной системе. Параллельные алгоритмы сортировки данных. Задача поиска кратчайших путей в графе.

## Тематический план

	Содержательный модуль 1																					
Названия содержательных модулей и тем	Количество часов																					
	Очная форма обучения												Заочная форма обучения									
	Нормативный срок обучения						Ускоренный срок обучения						Нормативный срок обучения						Ускоренный срок обучения			
	всего	в т.ч.					всего	в т.ч.					всего	в т.ч.					вс	в т.ч.		
		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	самостоятельная индивидуальная
Тема 1. Принципы построения параллельных вычислительных систем	32	8	–	8	10	6	32	8	–	8	10	6	35	2	–	2	25	6	–	–	–	–
Тема 2. Анализ вычислительной сложности параллельных алгоритмов	36	12	–	8	10	6	36	12	–	8	10	6	37	3	–	3	25	6	–	–	–	–
Тема 3. Разработка параллельных программ	40	12	–	12	10	6	40	12	–	12	10	6	35	2	–	2	25	6	–	–	–	–
Тема 4. Параллельное программирование при помощи технологии OpenMP	36	8	–	12	10	6	36	8	–	12	10	6	37	3	–	3	25	6				
Итого по содержательному модулю 1	144	40	–	40	40	24	144	40	–	40	40	24	144	10	–	10	100	24	–	–	–	–

## 5. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ПРОВЕДЕНИЯ ЛЕКЦИОННЫХ, ПРАКТИЧЕСКИХ И ЛАБОРАТОРНЫХ ЗАНЯТИЙ

### Темы лекционных занятий

<i>№ n/n</i>	<i>Название темы</i>	<i>Количество часов</i>
1	Введение. Параллельные компьютеры и суперЭВМ. Классификация параллельных вычислительных систем. Типы параллелизма.	4
2	Параллелизм на уровне команд. Конвейер. Суперскалярность. SIMD команды. Параллелизм на уровне задач.	4
3	Аналитические методы определения вычислительной сложности. Оценка вычислительной сложности для последовательных алгоритмов.	4
4	Показатели для оценки параллельных алгоритмов: ускорение, эффективность и стоимость параллельных алгоритмов. Законы Амдала и Густафсона.	4
5	Экспериментальные методы определения вычислительной сложности. Средства профилирования.	4
6	Декомпозиция по функциям и по данным. Планирование параллельных программ.	4
7	Зависимости данных. Способы синхронизации и взаимного исключения потоков.	4
8	Определение стоимости программы с учетом параллельного выполнения.	4
9	Организация параллельных вычислений средствами технологии OpenMP.	4
10	Синхронизация потоков. Примеры параллельных программ на OpenMP.	4
	<b>ВСЕГО</b>	<b>40</b>

### Темы лабораторных занятий

<i>№ n/n</i>	<i>Название темы</i>	<i>Количество часов</i>
1	Параллелизм на уровне команд. Конвейер. Суперскалярность. Параллелизм на уровне данных. SIMD команды.	4
2	Параллелизм на уровне задач. Оценка вычислительной сложности для последовательных алгоритмов.	4
3	Ускорение, эффективность и стоимость параллельных алгоритмов. Экспериментальные методы определения вычислительной сложности.	4
4	Декомпозиция по функциям. Декомпозиция по данным.	4
5	Зависимости данных. Типы зависимостей. Способы синхронизации.	4
6	Определение стоимости программы с учетом параллельного выполнения.	4
7	Масштабируемость.	2
8	Параллельные и последовательные области. Директива parallel.	2
9	Распараллеливание оператора цикла. Синхронизация и редукция.	4
10	Параллельные алгоритмы матричного умножения.	2
11	Вычисление определенного интеграла в многопроцессорной системе.	2
12	Параллельные алгоритмы сортировки данных.	2
13	Задача поиска кратчайших путей в графе.	2
	<b>ВСЕГО</b>	<b>40</b>

## 6. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ

## САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

### Организация самостоятельной работы студентов

<b>№ п/п</b>	<b>Название темы</b>	<b>Количество часов</b>
1	Ускорение, эффективность и стоимость параллельных алгоритмов.	5
2	Экспериментальные методы определения вычислительной сложности.	5
3	<i>Индивидуальное задание № 1</i>	6
4	Умножение матрицы на вектор.	4
5	Умножение матриц. Эффективность параллельного подхода..	5
6	<i>Индивидуальное задание № 2</i>	6
7	Интегрирование методом прямоугольников.	5
8	Вычисление определенного интеграла в многопроцессорной системе.	5
9	<i>Индивидуальное задание № 3</i>	6
10	Параллельные алгоритмы сортировки данных.	3
11	<i>Индивидуальное задание № 4</i>	5
12	Задача поиска кратчайших путей в графе.	4
13	<i>Индивидуальное задание № 5</i>	5
	<b>ВСЕГО</b>	<b>64</b>

## 7. ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

### **ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ ПАРАЛЛЕЛЬНОГО ПРОГРАММИРОВАНИЯ OPENMP ПРИ РЕАЛИЗАЦИИ ПАРАЛЛЕЛЬНОГО УМНОЖЕНИЯ МАТРИЦ**

Цель: овладение технологией OpenMP для реализации параллельных алгоритмов умножения матрицы на вектор, перемножения матриц, вычисления определенного интеграла, чет-нечетной сортировки, алгоритма Флойда поиска кратчайших путей в графе.

#### **Задания:**

1) Вычислить элементы двумерного квадратного массива  $(a_{ij}) \ (i, j = \overline{0, N-1})$ , используя для вычисления функцию  $a_{ij} = f(i, j)$  при помощи последовательного и параллельного алгоритмов. Для различных значений размерности массива  $N$  и количества процессорных ядер определить экспериментально показатели эффективности для этих алгоритмов и сравнить их с ожидаемыми. Сделать соответствующие выводы.

2) а) Выполните реализацию параллельного алгоритма умножения матрицы на вектор, основанного на ленточном разбиении матрицы на горизонтальные полосы. Проведите вычислительные эксперименты с учетом параметров используемой вычислительной системы. Сравните результаты реальных экспериментов с теоретическими оценками.

б) Выполните параллельную реализацию матричного умножения  $A \cdot A$ , проведите соответствующие эксперименты и сделайте выводы.

3) Провести численные эксперименты для последовательной и двух параллельных версий двумерного интегрирования функции  $f(x, y)$  по области  $D = [0, a] \times [0, a]$  с помощью формулы параллелепипедов в многоядерной вычислительной системе. Построить сравнительные диаграммы. Сделать соответствующие выводы.



4) Выполните реализацию последовательного и параллельного алгоритмов обобщенной пузырьковой сортировки (чет-нечетная перестановка). Массив сортируемых данных генерировать случайным образом. Проведите вычислительные эксперименты. Сравните полученные экспериментальные результаты с теоретическими оценками. Сделайте выводы.

5) Выполните реализацию параллельного алгоритма Флойда. Проведите вычислительные эксперименты. Постройте теоретические оценки с учетом параметров используемой вычислительной системы. Сравните полученные оценки с экспериментальными данными.

## 8. КОНТРОЛЬНЫЕ ВОПРОСЫ К ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

1. Предположим, что один землекоп может за час вырыть яму размером  $1 \times 1 \times 1$  м<sup>3</sup> и способен работать в таком режиме достаточно долго. За какое время бригада из 5, 10, 20 землекопов выроет яму размером  $2 \times 2$  м<sup>2</sup> и глубиной 1 м.

2. Постройте график времени выполнения работы в зависимости от числа землекопов в бригаде.

3. Повторите задания п. 1, 2 для ямы  $10 \times 10$  м<sup>2</sup>, глубиной 1 м и бригады из 10, 100 землекопов.

4. Конвейерное устройство состоит из пяти устройств. Времена срабатываний ступеней равны 1, 1, 2, 1 и 3 такта соответственно. С какой максимальной частотой на выходе данного устройства будут появляться результаты, если на его вход аргументы поступают без перебоев?

5. За какое минимальное число тактов может быть выполнено 70 операций, если в распоряжении есть устройство, описанное в задаче 5.

6. В компьютере есть 7 параллельно работающих устройств, каждое из которых может выполнить операцию за 7 единиц времени. За какое минимальное время этот компьютер обработает 7 независимых операций?

7. Конвейерное устройство состоит из  $k$  ступеней, срабатывающих за  $n_1, n_2, \dots, n_k$  тактов соответственно. За какое минимальное число тактов может быть выполнено  $m$  операций на таком устройстве?

8. Назовите уровни параллелизма. К какому уровню относится конвейерная обработка команд?

9. Что такое временная сложность алгоритма?

10. Какие показатели используются для оценки параллельных алгоритмов?

11. Оцените 2 алгоритма, стоимости которых в последовательном и параллельном режиме одинаковы. Имеет ли смысл использовать параллельный алгоритм в этом случае?

12. Определите значения ускорения, эффективности и стоимости, если известны следующие временные характеристики: для 6 процессоров время выполнения последовательной части составляет 25 % и время выполнения параллельной части составляет 75 %.

13. Определите временную сложность для алгоритма вычисления суммы элементов одномерного массива в последовательном режиме. Проверьте экспериментально полученные соотношения.

14. Определите временную сложность для алгоритма сортировки методом простой вставки одномерного массива в последовательном режиме. Проверьте экспериментально полученные соотношения.

15. Определите временную сложность для алгоритма вычисления скалярного произведения двух одномерных векторов в последовательном режиме. Проверьте экспериментально полученные соотношения.

16. Определите временную сложность для алгоритма вычисления произведения матрицы на вектор в последовательном режиме. Проверьте экспериментально полученные соотношения.

17. Определите временную сложность для алгоритма вычисления произведения прямоугольных матриц в последовательном режиме. Проверьте экспериментально полученные соотношения.

18. Определите значения ускорения, эффективности и стоимости для алгоритма вычисления суммы элементов одномерного массива, если: а) число процессоров не ограничено; б) число процессоров задано ( $p > 2$ ).

19. Что определяют условия Бернштейна?

20. Используя *GCD* тест, проверить наличие или отсутствие зависимости для цикла: `for(i=0; i<n; ++i) x[i]=x[2*i];`

21. Может ли ускорение быть больше, чем число ядер процессора, в каком случае это возможно?

22. Для каких операционных систем и языков программирования есть поддержка *OpenMP*?

23. Как программно проверить, что режим *OpenMP* включен?

24. В каких единицах возвращает время функция *omp\_get\_wtime()*? Как определить точность измерения времени? Напишите соответствующий код.

25. Пусть число потоков задано параметром *num\_threads* (2), параметром среды *OMP\_NUM\_THREAD* (3) и функцией *omp\_set\_num\_threads* (4). Сколько потоков будет создано для параллельной секции? Проверьте ответ соответствующим кодом программы.

26. Какие параметры надо передать потоковой функции для распараллеливания цикла с известным числом повторений?

27. Определите потоковую функцию для параллельного выполнения цикла сложения двух векторов с фиксированным размером. Напишите код для создания потоков и определите время выполнения суммирования без потоков и с использованием потоков. Определите значение ускорения в зависимости от размера векторов.

28. Составьте функции для вычисления скалярного произведения двух векторов в последовательном и параллельном режимах. Для накопления суммы в параллельном режиме используйте *reduction*. Сравните производительности функций.

29. Составьте функции для умножения матрицы на вектор в последовательном и параллельном режимах. Для накопления сумм в параллельном режиме используйте *reduction*. Сравните производительности функций.

30. Пусть параллельный участок программы содержит вывод элементов массива, сформированных в процессе вычислений. Как обеспечить вывод в естественном порядке, а не в порядке выполнения потоков? Напишите функции с выводом в произвольном и фиксированном порядке. Измерьте время выполнения функций для обоих вариантов. Сделайте выводы.

31. Пусть необходимо накапливать сумму элементов. Для этого можно использовать *reduction* или директиву *#pragma omp atomic*. Какой из этих способов более эффективный? Почему?

32. Как исключить возможность взаимной блокировки при использовании нескольких вложенных критических секций?

## 9. ОБРАЗЕЦ МОДУЛЬНОГО КОНТРОЛЯ

ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»

## Факультет математики и информационных технологий

Направление подготовки: 09.03.04 Программная инженерия  
 Профиль: Программная инженерия  
 Программа подготовки: **бакалавриат**  
 Семестр: **8**  
 Учебная дисциплина: Параллельное программирование

### МОДУЛЬНАЯ КОНТРОЛЬНАЯ РАБОТА ВАРИАНТ №1

1. Назовите уровни параллелизма. К какому уровню относится конвейерная обработка команд?
2. Какие показатели используются для оценки параллельных алгоритмов?
3. Используя *GCD* тест, проверить наличие или отсутствие зависимости для цикла:  
`for(i=0; i<n; ++i) x[i]=x[2*i];`
4. Составьте функции для вычисления скалярного произведения двух векторов в последовательном и параллельном режимах. Для накопления суммы в параллельном режиме используйте *reduction*. Сравните производительности функций.

Утверждено на заседании кафедры теории упругости и вычислительной математики, протокол № \_\_\_\_ от “\_\_” \_\_\_\_\_ 20\_\_ г.

Зав. кафедрой  
Преподаватель

\_\_\_\_\_  
\_\_\_\_\_

#### Критерии оценивания модульного контроля

<i>Номер задания</i>	<i>Количество баллов</i>
Задание 1	10
Задание 2	10
Задание 3	10
Задание 4	10
<b><i>Всего</i></b>	<b><i>40</i></b>

### 10. ОБРАЗЕЦ ЭКЗАМЕНАЦИОННОГО БИЛЕТА

#### *Теоретические вопросы к экзамену*

1. Параллельные компьютеры и суперЭВМ
2. Классификация параллельных вычислительных систем
3. Типы параллелизма. Параллелизм на уровне команд. Конвейер
4. Типы параллелизма. Параллелизм на уровне данных. SIMD команды
5. Аналитические методы определения вычислительной сложности
6. Показатели для оценки параллельных алгоритмов
7. Экспериментальные методы определения вычислительной сложности
8. Планирование параллельных программ (зависимости данных, типы зависимостей).
9. Планирование параллельных программ (способы синхронизации)
10. Планирование параллельных программ (определение стоимости программы с учетом параллельного выполнения)
11. Параллельные и последовательные области в технологии OpenMP
12. Директива `parallel`

13. Распараллеливание оператора цикла в технологии OpenMP
14. Использование параллельных секций. Директива sections
15. Способы синхронизации потоков в OpenMP

**Образец экзаменационного билета**

**ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»**

Факультет математики и информационных технологий

Направление подготовки: **09.03.04 Программная инженерия**  
 Профиль : **Программная инженерия**  
 Программа подготовки: **бакалавриат**  
 Семестр **8**  
 Учебная дисциплина **Параллельное программирование**

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1**

1. Параллельные компьютеры и суперЭВМ.
2. Способы синхронизации потоков в OpenMP.
3. Пусть в решаемой задаче используется восемь процессоров и при этом достигается ускорение в два раза. Если использовать закон Амдаля, какова доля последовательной части
  - а) 3/7
  - б) 1/3
  - в) 2/5

Утверждено на заседании кафедры теории упругости и вычислительной математики  
 Протокол № \_\_\_\_ от

Экзаменатор  
 Зав. кафедрой

Нескородев Р.Н.  
 Сторожев В.И.

**Критерии оценивания экзамена**

<b>Номер задания</b>	<b>Количество баллов</b>
Задание 1	20
Задание 2	20
Задание 3	10
<b>Всего</b>	<b>50 баллов</b>

**11. ОБРАЗЕЦ ТЕСТОВОГО ЗАДАНИЯ**

Пусть в решаемой задаче последовательная часть составляет четыре единицы времени, а часть, допускающая линейное распараллеливание, шесть единицы времени. Если использовать закона Густавсона-Барсиса, сколько потребуется процессоров для достижения ускорения в два раза

- а) 6;                      б) 8;                      в) 2.

## 12. КРИТЕРИИ ОЦЕНИВАНИЯ

По курсу предполагается проведение промежуточной аттестации в виде модульного контроля, выполнение индивидуальной работы и экзамена. Экзамен сдают студенты с целью повышения рейтинга.

*Распределение баллов, которые могут получить студенты  
в процессе изучения дисциплины*

Организационно учебная работа студента	СРС		Всего
	Индивидуальная работа	Модульный контроль	
Max 10 баллов	max 50 баллов	max 40 баллов	100 баллов

*Шкала соответствия баллов национальной шкале*

Оценка по шкале ECTS	Оценка по 100-балльной шкале	Оценка по государственной шкале (экзамен, дифференцированный зачёт)	Оценка по государствен ной шкале (зачёт)
A	90-100	5 (отлично)	зачтено
B	80-89	4 (хорошо)	зачтено
C	75-79	4 (хорошо)	зачтено
D	70-74	3 (удовлетворительно)	зачтено
E	60-69	3 (удовлетворительно)	зачтено
FX	35-59	2 (неудовлетворительно) с возможностью повторной сдачи	не зачтено
F	0-34	2 (неудовлетворительно) с возможностью повторной сдачи при условии обязательного набора дополнительных баллов	не зачтено

## 13. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОГО ПРОЦЕССА

Лекционные и лабораторные занятия проводятся в компьютерном классе, оборудованном компьютерами с лицензионным программным обеспечением, доступом к сети Интернет, столами, доской.

## 14. РЕКОМЕНДОВАННАЯ ЛИТЕРАТУРА

№ п/п	Наименование	Кол-во экземпляров в библиотеке ДонНУ	Наличие электронной версии в ЭБС
<i>Основная литература</i>			
1.	Нескородев, Р.Н. Параллельное программирование [Электронный ресурс]: учебное пособие / Р.Н. Нескородев; ГОУ ВПО “Донецкий национальный университет”. – Донецк: ДонНУ, 2019. – электронные данные (1 файл).	10	+

2.	Нескородев, Р.Н. Организация параллельных вычислений при помощи технологии OpenMP [Электронный ресурс]: учебно-методическое пособие для студентов направления подготовки 01.04.02 Прикладная математика и информатика / Р. Н. Нескородев; ГОУ ВПО “Донецкий национальный университет”. – Донецк: ДонНУ, 2019. – электронные данные (1 файл).	0	+
<i><b>Дополнительная литература</b></i>			
3.	Воеводин В.В. Параллельные вычисления: Учеб. пособие для вузов по направлению "Прикладная математика и информатика" / В.В. Воеводин, Вл.В. Воеводин. – СПб.: БХВ-Петербург, 2002. – 608 с.	1	-
4.	Качко Е.Г. Параллельное программирование / Е.Г. Качко. – Харьков: Форт, 2011. – 528 с.	6	-
5.	Барский А.Б. Параллельные информационные технологии / А.Б. Барский. – М.: Интернет-Ун-т информ. технологий: БИНОМ. Лаб. знаний, 2007. – 502 с.	2	-
6.	Гергель В.П. Теория и практика параллельных вычислений / В.П. Гергель. – М.: Бином. Лаборатория знаний, 2007. – 424 с.	2	-
7.	Калоеров С.А. Программирование на языке C++: учеб. пособие / С.А. Калоеров. – Донецк: Юго-Восток, 2009. – 298 с.	85	-
8.	Бахвалов, Н. С. Численные методы: Учеб. пособие для студентов физ.-мат. спец. вузов / Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков. – 2-е изд. – М.: Лаб. Баз. Знаний; СПб.: Невский диалект, 2008. – 640 с.	7	-

## 15. ИНФОРМАЦИОННЫЕ РЕСУРСЫ

<http://parallel.ru> - Лаборатории Параллельных информационных технологий Научно-исследовательского вычислительного центра Московского государственного университета имени М.В.Ломоносова;

## 16. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

1. Windows 7 PRO (корпоративная лицензия ДОННУ № 46484614);
2. Microsoft Office (корпоративная лицензия ДОННУ лицензия № 46472919);
3. Microsoft Visual Studio (лицензия программы DreamSpark для высших учебных заведений);

Рабочая программа рассмотрена и переутверждена на заседании кафедры прикладной механики и компьютерных технологий с изменениями (без изменений) на 20\_\_ год.

Протокол № \_\_ от “\_\_” \_\_\_\_\_ 20\_\_ г. Заведующий. кафедрой \_\_\_\_\_



Рабочая программа рассмотрена и переутверждена на заседании кафедры прикладной механики и компьютерных технологий с изменениями (без изменений) на 20\_\_\_\_ год.

Протокол № \_\_\_\_ от “\_\_” \_\_\_\_\_ 20\_\_ г. Заведующий. кафедрой \_\_\_\_\_

Рабочая программа рассмотрена и переутверждена на заседании кафедры прикладной механики и компьютерных технологий с изменениями (без изменений) на 20\_\_\_\_ год.

Протокол № \_\_\_\_ от “\_\_” \_\_\_\_\_ 20\_\_ г. Заведующий. кафедрой \_\_\_\_\_

Рабочая программа рассмотрена и переутверждена на заседании кафедры прикладной механики и компьютерных технологий с изменениями (без изменений) на 20\_\_\_\_ год.

Протокол № \_\_\_\_ от “\_\_” \_\_\_\_\_ 20\_\_ г. Заведующий. кафедрой \_\_\_\_\_

Рабочая программа рассмотрена и переутверждена на заседании кафедры прикладной механики и компьютерных технологий с изменениями (без изменений) на 20\_\_\_\_ год.

Протокол № \_\_\_\_ от “\_\_” \_\_\_\_\_ 20\_\_ г. Заведующий. кафедрой \_\_\_\_\_