

ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Кафедра прикладной математики и теории систем управления



УТВЕРЖДАЮ:

профессор по научно-методической
и учебной работе

Е.И. Скафа

«22» апреля 2020 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
«ПАРАЛЛЕЛЬНОЕ И РАСПРЕДЕЛЕННОЕ
ПРОГРАММИРОВАНИЕ»

Направление подготовки:	02.04.02 Фундаментальная информатика и информационные технологии
Магистерская программа:	Фундаментальная информатика и информационные технологии
Образовательная программа:	академическая магистратура
Квалификация:	магистр
Форма обучения:	<u>очная</u> , очно-заочная, заочная нужное подчеркнуть

Донецк 2020

УТВЕРЖДАЮ:

Декан факультета математики
и информационных технологий

И. А. Моисеенко

«16» апреля 2020 г.

МП



Программа учебной дисциплины «Параллельное и распределенное программирование» составлена с учетом Федерального государственного образовательного стандарта высшего образования направления подготовки 02.04.02 Фундаментальная информатика и информационные технологии, утвержденного приказом Министерства образования и науки Российской Федерации от «23» августа 2017 г. № 811; основной образовательной программы и учебного плана направления подготовки 02.04.02 Фундаментальная информатика и информационные технологии, разработанных в ГОУ ВПО «Донецкий национальный университет».

Разработчик:

Доцент кафедры прикладной математики
и теории систем управления

 Д.В. Шевцов

Программа учебной дисциплины утверждена на заседании кафедры прикладной математики и теории систем управления

Протокол № 12 от « 9 » апреля 2020 г.
Заведующий кафедрой

 Д.В. Шевцов

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 и распределенное программирование» относится к вариативной части подготовки студентов по направлению подготовки 02.04.02 «Фундаментальная информатика и информационные технологии» образовательной программы «Академическая магистратура». Дисциплина реализуется на факультете математики и информационных технологий ДонНУ кафедрой прикладной математики и теории систем управления.

Содержание дисциплины является логическим продолжением содержания дисциплин:

- Архитектура современных ЭВМ;
- Объектно-ориентированное программирование;
- Программирование;
- Технологии баз данных;
- Распределенные технологии.

и формирует основу для освоения дисциплин:

- Проектирование и администрирование сетевых коммуникаций;
- Математические основы защиты информации и информационной безопасности;
- магистерской диссертации.

2. СТРУКТУРА ДИСЦИПЛИНЫ

<i>Характеристика учебной дисциплины</i>		
Направление подготовки	02.04.02 Фундаментальная информатика и информационные технологии	
Магистерская программа	Фундаментальная информатика и информационные технологии	
Образовательная программа	академическая магистратура	
Квалификация	магистр	
Количество содержательных модулей	1	
Дисциплина базовой / вариативной части образовательной программы	вариативная часть	
Формы контроля (МК, экзамен, зачет)	модульный контроль, зачет	
Показатели	очная форма обучения	заочная форма обучения
Количество зачетных единиц (кредитов)	4	
Год подготовки	1	
Семестр	2	
Количество часов	144	
- лекционных	18	
- практических, семинарских		
- лабораторных	36	
- самостоятельной работы	90	
в т.ч. индивидуальное задание		
Недельное количество часов,	8	
в т.ч. аудиторных	3	

3. ОПИСАНИЕ ДИСЦИПЛИНЫ

Цели и задачи

Цель – изучение основ параллельного программирования, развитие мышления, связанного с параллельным программированием. Систематизация знаний о методах и

алгоритмах параллельного программирования, моделях параллельных вычислений.

Задачи – изучить и закрепить методы и способы распараллеливания и построения параллельных программ; изучить основы программирования потоков; исследовать и реализовать набор параллельных алгоритмов для стандартных типовых задач.

Требования к результатам освоения дисциплины. Процесс изучения дисциплины «Параллельное и распределенное программирование» направлен на формирование элементов следующих компетенций в соответствии с ФГОС ВО РФ направления подготовки 02.04.02 – «Фундаментальная информатика и информационные технологии» и основной образовательной программы высшего профессионального образования направления подготовки 02.04.02 – «Фундаментальная информатика и информационные технологии» (магистерская программа: Фундаментальная информатика и информационные технологии):

а) универсальных (УК): способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий (УК-1); способен управлять проектом на всех этапах его жизненного цикла (УК-2); способен определять и реализовывать приоритеты собственной деятельности и способы её совершенствования на основе самооценки (УК-6);

б) общепрофессиональных (ОПК): способен находить, формулировать и решать актуальные проблемы прикладной математики, фундаментальной информатики и информационных технологий (ОПК-1); способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение (в том числе отечественного производства) для решения задач профессиональной деятельности (ОПК-2); способен проводить анализ математических моделей, создавать инновационные методы решения прикладных задач профессиональной деятельности в области информатики и математического моделирования (ОПК-3); способен устанавливать и сопровождать программное обеспечение информационных систем, осуществлять эффективное управление разработкой программных средств и проектов (ОПК-5);

в) профессиональных (ПК): способен формализовать и алгоритмизировать поставленные задачи (ПК-3); способен написать программный код с использованием языков программирования, определять и манипулировать данными (ПК-4); способен определять входные-выходные данные каждого компонента и программного средства в целом (ПК-5); способен испытывать создаваемое программное средство и его компоненты (ПК-6); способен разрабатывать тестовые документы, включая план тестирования (ПК-7); способен осуществлять сбор данных для выявления требований к типовой ИС в соответствии с трудовым заданием (ПК-10); способен оформлять технические документы в соответствии с заданным стандартом (ПК-13); способен формализовать и документировать требования к функциям системы (ПК-15); способен формализовать и документировать требования к системе и подсистеме (ПК-16).

В результате изучения учебной дисциплины студент должен:

Знать:

- Среду визуального программирования Microsoft Studio;
- методы параллельного программирования с разделяемыми переменными;
- синхронизацию процессов через доступ к общим ресурсам;
- понятие о критических интервалах, семафорах;
- программирование параллельных алгоритмов с помощью критических интервалов и семафоров;
- эволюцию развития приемов и методов параллельного программирования.

Уметь:

- Использовать возможности программирования и тестирования в системах визуального программирования Visual Studio;

– использовать полученные знания в своей практической деятельности для проектирования, реализации и сопровождения параллельных программ.

Владеть:

– технологиями распараллеливания последовательных алгоритмов.

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ И ФОРМЫ ОРГАНИЗАЦИИ УЧЕБНОГО ПРОЦЕССА

Порядковый номер и тема	Краткое содержание темы
	Содержательный модуль 1
Тема 1. Введение	Тенденции развития вычислительных систем, обуславливающие необходимость применения распределённых (параллельных) методов вычислений. Примеры вычислительно ёмких задач из разных областей науки. Классификация архитектур компьютерных систем. Классификация Флинна
Тема 2. Параллелизм	Способы распараллеливания программ. Проблема автоматизации распараллеливания: текущее состояние средств, способных выявлять некоторые виды параллелизма. Понятия ускорения, эффективности (закон Амдала), масштабируемости распараллеленного алгоритма. Процессы и потоки (контексты, состояния) (Причины некоторых преимуществ потоков над процессами). Модель процесса, Иерархия процессов. Реализация процессов
Тема 3. Программирование потоков	Posix-потоки (Posix-threads). Создание потоков. Корректное завершение потоков. Передача данных в поток. Объединение потоков. Отмена потока. Обработка исключений. Примитивы синхронизации: инварианты, критичные секции и предикаты, Состояние гонки, мьютекс (futex), семафор. Событие
Тема 4. Методы Монте Карло, параллельные реализации	Случайные числа. Псевдослучайные числа (ПСЧ). Методы генерации ПСЧ. Метод фон Неймана. Пример Кнута. Линейно-конгруэнтный метод. Сдвиговый регистр. Современные методы. Проблема инициализации. Аппаратные и программные реализации. Вычисления многомерного интеграла методом Монте-Карло.

Тематический план

Содержательный модуль 1												
Названия содержательных модулей и тем	Количество часов											
	Очная форма обучения						Заочная форма обучения					
	всего	в т.ч.					всего	в т.ч.				
		лекции	практические	лабораторные	самостоятельна я работа	индивидуальна я работа		лекции	практические	лабораторные	самостоятельна я работа	индивидуальна я работа
Тема 1. Введение	19	3		6	10							
Тема 2. Параллелизм	33	3		10	20							
Тема 3. Программирование потоков	46	6		10	30							
Тема 4. Методы Монте Карло, параллельные реализации	46	6		10	30							
<i>Итого по содержательному модулю 1</i>	180	18		36	90							
<i>Всего по дисциплине</i>	180	18		36	90							

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

Темы лекционных занятий

№ п/п	Название темы	Количество часов
1	Тенденции развития вычислительных систем. Примеры вычислительно ёмких задач из разных областей науки. К	2
2	Классификация архитектур компьютерных систем. Классификация Флинна	2
3	Способы распараллеливания программ.	2
4	Проблема автоматизации распараллеливания.	2
5	Понятия ускорения и эффективности. Закон Амдала.	2
6	Процессы и потоки (контексты, состояния)	2
7	Реализация процессов	2
8	Posix-потоки (Posix-threads).	2
9	Примитивы синхронизации: инварианты, критичные секции и предикаты, Состояние гонки, мьютекс (futex), семафор.	2
	ВСЕГО	18

Темы лабораторных занятий

№ n/n	Название темы	Количество часов
1	Способы распараллеливания программ	3
2	Понятия ускорения, эффективности (закон Амдала), масштабируемости распараллеленного алгоритма.	3
3	Процессы и потоки (контексты, состояния). Модель процесса.	2
4	Иерархия процессов. Реализация процессов.	2
5	Posix-потоки (Posix-threads). Создание потоков. Корректное завершение потоков.	2
6	Передача данных в поток. Объединение потоков. Отмена потока.	2
7	Обработка исключений. Примитивы синхронизации. Инварианты, критичные секции и предикаты. Состояние гонки.	2
8	Мьютекс.	2
9	Семафор	2
10	Основные директивы препроцессора. Создание потоков, параллельные секции (parallel).	2
11	Работа с данными (shared, private...). Синхронизация (critical, atomic).	2
12	Взаимодействие потоков (reduction, coryin...). Контроль числа потоков. Применимость для вложенных вызовов	2
13	Отправка сообщения	4
14	Передачи данных между двумя процессами	4
15	Типы данных	2
16	Понятие коммуникатора	2
	ВСЕГО	36

6. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Организация самостоятельной работы студентов

№ n/n	Название темы	Количество часов
1	Способы распараллеливания программ	5
2	Понятия ускорения, эффективности (закон Амдала), масштабируемости распараллеленного алгоритма.	5
3	Процессы и потоки (контексты, состояния). Модель процесса.	5
4	Иерархия процессов. Реализация процессов.	5
5	Posix-потоки (Posix-threads). Создание потоков. Корректное завершение потоков.	5
6	Передача данных в поток. Объединение потоков. Отмена потока.	5
7	Обработка исключений. Примитивы синхронизации. Инварианты, критичные секции и предикаты. Состояние гонки.	5
8	Мьютекс.	7
9	Семафор	7
10	Основные директивы препроцессора. Создание потоков, параллельные	7

	секции (parallel).	
11	Работа с данными (shared, private...). Синхронизация (critical, atomic).	7
12	Взаимодействие потоков (reduction, coryin...). Контроль числа потоков. Применимость для вложенных вызовов	7
13	Отправка сообщения	5
14	Передачи данных между двумя процессами	5
15	Типы данных	5
16	Понятие коммуникатора	5
	ВСЕГО	90

7. ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

В течение года студенты выполняют индивидуальное задание, по результатам выполнения которого осуществляется устная защита с изложением полученных результатов и объяснениями проделанных действий. Индивидуальные задания в количестве 10 вариантов находятся в электронном виде на кафедре.

Примеры индивидуальных заданий.

Индивидуальное задание №1

Создание набора потоков

Цель: научиться формировать набор потоков, использующих параметры. Приостановка и возобновление потоков.

Задание:

Для приведенных ниже вариантов заданий составить многопоточную Windows-программу, в которой производится создание и выполнение 2 потоков: 1-й поток выполняет операции в главном окне приложения, 2-й поток - в другом окне (временное или дочернее), а также – запуск дочернего процесса. Для управления потоками и запуска дочерних процессов используется немодальное диалоговое окно.

Требования к программе. В программе должно быть следующее:

- при запуске приложения должно появиться немодальное диалоговое окно, состоящее из трех частей (групп) с заголовками: Поток 1; Поток 2; Дочерний процесс и кнопкой "Выход" внизу, по нажатию которой приложение завершает работу;

- первая и вторая части диалогового окна должны содержать следующее: а) Состояние потока - (Работает, Приостановлен, Завершен), начальное состояние - Приостановлен; б) Текущий приоритет потока – число; в) Изменение приоритета потока – используется интерфейсный элемент согласно варианту задания; г) 3 кнопки для запуска потока, его приостановки и завершения;

- в третьей части диалогового окна должно быть следующее: а) Запускаемый процесс – (путь, имя), вначале пусто; б) Кнопка "Выбрать", по нажатию которой вызывается стандартный диалог Open и производится выбор на диске запускаемого программного файла, после чего его полный путь и имя отображаются в окне; в) Кнопка "Запустить", с помощью которой производится запуск программы, определенной заданием, если имя программы не выбрано, то запуск не производится, выдается сообщение с помощью функции MessageBox;

- 1-й и 2-й потоки должны быть созданы в приостановленном состоянии в оконных функциях (по сообщению WM_CREATE), для потоков должны быть описаны соответствующие потоковые функции, выполняющие в цикле операции в окнах согласно варианту задания. Запуск потоков в работу, их приостановка и завершение производятся по нажатию соответствующих кнопок на диалоговом окне;

- в рабочей области главного окна, при запуске 1-го потока должен выводиться график функции согласно варианту задания. Выводимый график должен размещаться на всю рабочую область окна, для этого значения функции должны быть масштабированы с учетом количества выводимых пикселей по осям X и Y. При приостановке потока вывод графика останавливается, при завершении – окно очищается;

- временное или дочернее окно должно появляться на экране при запуске 2-го потока, в нем должна выводиться движущаяся графическая фигура или текстовая строка, для этого окна должна быть описана своя оконная функция, в которой обрабатываются сообщения для этого окна;

- немодальное диалоговое окно создается с помощью редактора ресурсов, для него в программе описывается соответствующая диалоговая процедура, содержащая код для обработки сообщений от элементов управления;

- при запуске приложения необходимо проверять наличие уже запущенного экземпляра этого приложения, выдавать об этом сообщение с помощью функции MessageBox и завершать работу.

Используемые Win API функции: для работы с потоками необходимо использовать функции CreateThread, GetThreadPriority, SetThreadPriority, ResumeThread, SuspendThread, Sleep; для запуска дочернего процесса необходимо использовать функцию CreateProcess.

№ вар	Поток или процесс	Вид выполняемого действия потоком/ имя запускаемого процесса	Элемент управления для задания приоритета потока
1	Поток 1 (гл. окно)	Вывод графика функции $10 \cdot e^x \cos(2 \cdot x)$ в диапазоне $[-5; 5]$ с интервалом 0,2 сек	ListBox
	Поток 2 (врем. окно)	Квадрат перемещается по углам окна по часовой стрелке с интервалом 0,5 сек	ComboBox
	Процесс	calc.exe	

Индивидуальное задание №2

Взаимодействие потоков

Цель: научиться формировать набор потоков, использующих общие данные. Сравнение эффективности в зависимости от количества потоков и процессоров.

Задание:

Для приведенных ниже вариантов заданий составить многопоточную Windows-программу, в которой производится создание и выполнение 3 потоков: 1-й поток выполняет операции в главном окне приложения, 2-й и 3-й потоки - в другом окне (временном или дочернем). Для синхронизации работы 1-го потока с аналогичным потоком в дочернем процессе (запущенный 2-й экземпляр этого приложения) используется именованный объект синхронизации. Для синхронизации работы 2-го и 3-го потока используется неименованный объект синхронизации.

Требования к программе. В программе должно быть следующее:

- в главном окне должно быть создано меню, состоящее из пунктов: Поток 1 (Запустить, Приостановить); Поток 2 (Запустить, Приостановить); Поток 3 (Запустить, Приостановить); Дочерний процесс (Запустить, Завершить);

- 1-й, 2-й и 3-й потоки должны быть созданы в приостановленном состоянии в оконных функциях (по сообщению WM_CREATE), для потоков должны быть описаны соответствующие потоковые функции, выполняющие в цикле операции в окнах согласно варианту задания. Запуск потоков в работу, их приостановка производятся при выборе соответствующих пунктов меню;

- в потоковых функциях 1-го, 2-го и 3-го потоков должны быть использованы указанные в варианте задания средства синхронизации. При захвате свободного объекта

синхронизации потоковая функция выполняет операции вывода в окно заданное число раз (количество повторов) и освобождает этот объект синхронизации;

- 1-й поток выполняет операции в рабочей области главного окна, для него должен быть создан именованный объект синхронизации, доступный в других процессах. Работа этого потока синхронизируется с аналогичным потоком дочернего процесса, который представляет собой еще один запущенный экземпляр данного приложения;

- 2-й и 3-й потоки синхронизируются одним неименованным объектом синхронизации и по очереди выполняют свои операции во временном или дочернем окне;

- в качестве дочернего процесса должен быть запущен еще один экземпляр данного приложения. Т.к. 1-й поток этого экземпляра использует именованный объект синхронизации, то его работа будет синхронизироваться с аналогичным потоком предыдущего экземпляра;

- при запуске приложения необходимо проверять наличие уже запущенного экземпляра данного приложения, выдавать об этом сообщение с помощью функции MessageBox с кнопками "Да", "Нет", по выбору пользователя продолжать или завершать работу

Используемые Win API функции: для работы с потоками необходимо использовать функции: CreateThread, ResumeThread, SuspendThread, Sleep; для запуска дочернего процесса необходимо использовать функцию CreateProcess, а для его завершения - TerminateProcess; для синхронизации потоков необходимо использовать: Create-функции для создания объектов синхронизации, Wait-функцию для ожидания освобождения объекта, соответствующие функции освобождения объектов синхронизации.

№ вар	Поток	Выводимый объект	Выполняемые действия	Кол-во повторов	Объект синхронизации
1	Поток 1 (гл. окно)	Битовый образ	Перемещается по периметру окна против часовой стрелки с шагом 30 пикселей с интервалом 0,5 сек	5	Мьютекс
	Поток 2 (доч. окно)	Окружность	Перемещается слева-направо и наоборот с шагом 20 пикселей с интервалом 0,4 сек	7	Ожидаемый таймер
	Поток 3 (доч. окно)	Прямоугольник	Перемещается сверху-вниз и наоборот с шагом 10 пикселей с интервалом 0,2 сек	9	

Критерии оценивания индивидуального задания

Оценка индивидуального задания осуществляется на основе вычисления среднего арифметического оценок, полученных за программную часть и защиту.

Программная часть

- Неудовлетворительно. Программа заимствована более чем на 75%; программа не работоспособна в принципе.
- Удовлетворительно. Программа заимствована более чем на 25%; уровень сложности не соответствует отведенному под выполнение проекта времени.
- Хорошо. Авторская разработка, основанная на материалах занятий.
- Отлично. Полностью самостоятельная разработка (использование отдельных модулей и библиотек допустимо с указанием источника кода) соответствующего уровня сложности; использование дополнительных материалов по алгоритмам и технологиям программирования; соблюдение принципов объектно-ориентированного программирования, грамотное оформление текста программы (в том числе — использование комментариев); грамотное использование структур данных.

Защита

- Неудовлетворительно. Отказ от защиты в установленные сроки; неспособность

объяснить существенные аспекты работы программы.

- Удовлетворительно. Затруднения с ответом на вопросы; отсутствие логики выступления; неграмотная речь.
- Хорошо. Неполное соответствие требованиям на "Отлично"

Отлично. Грамотная речь с правильным использованием терминологии; заранее продуманная логика выступления; полнота освещения проекта; свободный ответ на вопросы. Приветствуется наличие презентации

8. КОНТРОЛЬНЫЕ ВОПРОСЫ К ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

1. Тенденции развития вычислительных систем, обуславливающие необходимость применения распределённых (параллельных) методов вычислений

2. Проблема автоматизации распараллеливания: текущее состояние средств, способных выявлять некоторые виды параллелизма

3. Создание потоков.

4. Корректное завершение потоков .

5. Передача данных в поток.

6. Объединение потоков.

7. Отмена потока.

8. Обработка исключений

9. Примитивы синхронизации

10. Инварианты, критичные секции и предикаты

11. Состояние гонки

12. Мьютекс (futex)

13. Семафор

14. Событие

15. Какие бывают параллельные архитектуры? Каковы их основные особенности?

16. Какие средства программирования могут быть использованы для данной архитектуры? Какие средства программирования эффективны для данной архитектуры?

17. Какова должна быть структура параллельной программы при использовании данной программной библиотеки?

18. Каковы основные функции OpenMP/MPI/PVM?

19. Почему важна балансировка нагрузки? Как ее добиться на разных архитектурах?

20. Для чего нужна синхронизация потоков/процессов?

21. Какова область применения метода Монте-Карло?

22. Напишите параллельную программу, реализующую скалярное произведение двух векторов.

23. Напишите параллельную программу, реализующую поиск максимального значения вектора.

24. Напишите параллельную программу, реализующую транспонирование матрицы $n \times n$.

25. Реализуйте параллельный алгоритм метода Гаусса решения систем линейных алгебраических уравнений.

26. Создать 2 вектора с 1000 элементами из случайных чисел. Скалярно умножить эти вектора. Сравнить время счета параллельного и последовательного алгоритмов.

9. ОБРАЗЕЦ МОДУЛЬНОГО КОНТРОЛЯ

ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»

Факультет математики и информационных технологий

Направление подготовки: 02.04.02 – «Фундаментальная информатика и ИТ»
 Магистерская программа: Фундаментальная информатика и информационные технологии
 Программа подготовки: академическая магистратура
 Семестр 3
 Учебная дисциплина Параллельное и распределенное программирование

МОДУЛЬНАЯ КОНТРОЛЬНАЯ РАБОТА ВАРИАНТ №1

1. Для чего нужна синхронизация потоков/процессов.
2. Вычислить двукратный интеграл $\int_c^d dy \int_a^b x \sin(x+y) dx$, пределы интегрирования по x от a до b , по y от c до d , число интервалов разбиения по x и по y задано числом N , $N > 10000$; использовать метод параллелепипедов, вычисляя объем i,j -го параллелепипеда.

Утверждено на заседании кафедры ПМ и ТСУ 31.01.2020 г., протокол № 7.

Заведующий кафедрой
Преподаватель

Д.В. Шевцов
А.-В. В. Мельник

Критерии оценивания модульного контроля

Номер задания	Количество баллов
1	10
2	30
Всего	40

10. ОБРАЗЕЦ ЭКЗАМЕНАЦИОННОГО БИЛЕТА

Не предусмотрено учебным планом.

11. ОБРАЗЕЦ ТЕСТОВОГО ЗАДАНИЯ

Не предусмотрено.

12. КРИТЕРИИ ОЦЕНИВАНИЯ

В течение семестра обучающийся может заработать баллы за следующие виды деятельности: индивидуальное задание (максимум 60 баллов), модульные контрольные работы по теории и практике (максимум 40 баллов), активность на занятиях (бонусные баллы).

№ п/п	Виды контрольных мероприятий	Количество баллов
	Текущий контроль	
1	Отчет о выполнении задания №1	25

2	Отчет о выполнении задания №2	35
Всего по текущему контролю		60
Итоговый контроль		
1.	Модульный контроль	40
Всего по итоговому контролю		40
Всего за семестр:		100

Шкала соответствия баллов национальной шкале

Оценка по шкале ECTS	Оценка по 100-балльной шкале	Оценка по государственной шкале (экзамен, дифференцированный зачет)	Оценка по государственной шкале (зачет)
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 занятия проводятся в аудитории, оснащенной мультимедийной техникой и доской.

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.	Нескороев, Р.Н. Параллельное программирование [Электронный ресурс]: учебное пособие / Р.Н. Нескороев; ГОУ ВПО «Донецкий национальный университет». – Донецк: ДонНУ, 2019. – электронные данные (1 файл).	10	+
2.	Нескороев, Р.Н. Организация параллельных вычислений при помощи технологии OpenMP [Электронный ресурс]: учебно-методическое пособие / Р. Н. Нескороев; ГОУ ВПО «Донецкий национальный университет». – Донецк: ДонНУ, 2019. – электронные данные (1 файл).	0	+
3.	Лабораторные работы по курсу «Распределённые информационные системы» [Электронный ресурс]:	0	+

	учебно-методическое пособие. / сост. С.А.Прийменко – Донецк: ДонНУ, 2016 (1 файл).		
Дополнительная литература			
4.	Бройдо, В. Л. Архитектура ЭВМ и систем: учебник для студентов вузов, обучающихся по направлению подготовки дипломированных специалистов "Информационные системы" / В. Л. Бройдо, О. П. Ильина. - 2-е изд. - Москва [и др.] : Питер, 2009. - 720	50	+
5.	Качко, Е. Г. Параллельное программирование: учеб. пособие / Е. Г. Качко; Харьковский нац. ун-т радиоэлектроники; ЗАТ "Институт информ. технологий". - Харьков: Форт, 2011. - 527 с.	7	-

15. ИНФОРМАЦИОННЫЕ РЕСУРСЫ

(с указанием названия и полного электронного адреса)

1. Электронный каталог библиотеки Донецкого национального университета: <http://library.donnu-support.ru/catalog/scripts/wek2.exe/mb> (дата обращения: 04.01.2020).
2. Электронно-библиотечная система «Znanium.com»: <http://znanium.com/> (дата обращения: 04.01.2020).
3. Электронно-библиотечная система «КнигаФонд»: <http://www.knigafund.ru/> (дата обращения: 04.01.2020).
4. Электронно-библиотечная система «Университетская библиотека онлайн»: www.bibloclub.ru (дата обращения: 04.01.2020).
5. Научная электронная библиотека (НЭБ): <http://elibrary.ru/defaultx.asp> (дата обращения: 04.01.2020).
6. БД российских научных журналов на Elibrary.ru (РУНЭБ): http://elibrary.ru/projects/subscription/rus_titles_open.asp (дата обращения: 04.01.2020).
7. БД российских журналов East View : <http://dlib.eastview.com> (дата обращения: 04.01.2020).
8. Базы данных компании EBSCO Publishing: <http://search.ebscohost.com/> (дата обращения: 04.01.2020).

16. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Visual Studio (не позднее 2015 года).

Рабочая программа рассмотрена и переутверждена на заседании кафедры ПМ и ТСУ с изменениями (без изменений) на 20____ год.

Протокол № ____ от «____» _____ 20____ г.

Заведующий кафедрой _____

Рабочая программа рассмотрена и переутверждена на заседании кафедры ПМ и ТСУ с изменениями (без изменений) на 20____ год.

Протокол № ____ от «____» _____ 20____ г.

Заведующий кафедрой
