



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И. Носова»



УТВЕРЖДАЮ
Директор ИЭиАС
С.И. Лукьянов

26.02.2020 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

***ТЕОРИЯ И ПРАКТИКА РАСПРЕДЕЛЕННЫХ И ПАРАЛЛЕЛЬНЫХ
ВЫЧИСЛЕНИЙ***

Направление подготовки (специальность)
09.03.01 Информатика и вычислительная техника

Направленность (профиль/специализация) программы
Проектирование и разработка приложений для мобильных устройств

Уровень высшего образования - бакалавриат


Форма обучения
очная

Институт/ факультет	Институт энергетики и автоматизированных систем
Кафедра	Вычислительной техники и программирования
Курс	2
Семестр	4

Магнитогорск
2020 год

Рабочая программа составлена на основе ФГОС ВО - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника (приказ Минобрнауки России от 19.09.2017 г. № 929)

Рабочая программа рассмотрена и одобрена на заседании кафедры
Вычислительной техники и программирования
19.02.2020 г. протокол № 5

Зав. кафедрой _____  О.С. Логунова

Рабочая программа одобрена методической комиссией ИЭ и АС
26.02.2020 г. протокол № 5

Председатель _____  С.И. Лукьянов

Рабочая программа составлена:

доцент кафедры ВТиП, канд. техн. наук _____  А.Н. Калитаев

Рецензент:

начальник отдела технологических платформ
ООО «Компас Плюс», канд. техн. наук _____

 Д.С. Сафонов

Лист актуализации рабочей программы

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2021 - 2022 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2022 - 2023 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2023 - 2024 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2024 - 2025 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ О.С. Логунова

1 Цели освоения дисциплины (модуля)

Целями освоения дисциплины (модуля) «Теория и практика распределенных и параллельных вычислений» являются ознакомление студентов с методами и средствами организации распределенных и параллельных вычислений, с основными принципами параллельного программирования, видами современных высокопроизводительных средств вычислительной техники.

Для достижения поставленной цели в курсе «Теория и практика распределенных и параллельных вычислений» решаются задачи:

- изучение методов и средств высокопроизводительного программирования;
- изучение основных принципов параллельного программирования с использованием технологии OpenMP, языка программирования Java;
- изучение видов современных высокопроизводительных средств вычислительной техники.

2 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина Теория и практика распределенных и параллельных вычислений входит в часть учебного плана формируемую участниками образовательных отношений образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин/ практик:

Прикладная математика

Программирование

Численные методы решения инженерных задач

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/практик:

Выполнение и защита выпускной квалификационной работы

Программные решения для бизнеса

Разработка кроссплатформенных приложений для мобильных устройств

3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения

В результате освоения дисциплины (модуля) «Теория и практика распределенных и параллельных вычислений» обучающийся должен обладать следующими компетенциями:

Код индикатора	Индикатор достижения компетенции
ПК-2	Способность к анализу проблемной ситуации, разработке требований к системе, постановке целей создания системы, разработке концепции системы и технического задания на создание системы, представления концепции, технического задания на систему и изменений в них заинтересованным лицам
ПК-2.1	Оценивает выбор средств и методов для проведения системного анализа при проектировании программного обеспечения для мобильных устройств
ПК-5	Способность к формализации и алгоритмизации поставленных задач, к написанию программного кода с использованием языков программирования, определения и манипулирования данными и оформлению программного кода в соответствии установленными требованиями для мобильных устройств
ПК-5.1	Оценивает качество математической модели при формализации задачи предметной области
ПК-5.2	Оценивает качество разработанных алгоритмов для последующего кодирования
ПК-5.3	Оценивает выбор программных средств для программирования и манипулирования данными в соответствии установленными

требованиями

4. Структура, объём и содержание дисциплины (модуля)

Общая трудоемкость дисциплины составляет 3 зачетных единиц 108 академических часов, в том числе:

- контактная работа – 51,95 академических часов;
- аудиторная – 51 академических часов;
- внеаудиторная – 0,95 академических часов
- самостоятельная работа – 56,05 академических часов;

Форма аттестации - зачет

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в академических часах)			Самостоятельная работа студента	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код компетенции
		Лек.	лаб. зан.	практ. зан.				
1. Введение в параллельные вычисления.								
1.1 Основы и парадигмы параллельного программирования. Методы и средства параллельного программирования.	4	2			6	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками.	Устный опрос (собеседование).	ПК-2.1, ПК-5.1, ПК-5.2, ПК-5.3
1.2 Многопроцессорные архитектуры. Параллельные машины с общей и раздельной памятью (классификация по Флину). Поддержка многозадачности современными		2			6	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками.	Устный опрос (собеседование).	ПК-2.1, ПК-5.1, ПК-5.2, ПК-5.3
Итого по разделу		4			12			
2. Программирование в системах с общей памятью. Процессы и потоки. Объекты синхронизации.								
2.1 Реализация потоков и объектов синхронизации в Windows (WinAPI и объекты классов Thread).	4	2	6/2И		8	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию.	1. Устный опрос (собеседование). 2. Лабораторные работы.	ПК-2.1, ПК-5.1, ПК-5.2, ПК-5.3

2.2 Реализация потоков и объектов синхронизации в Unix (POSIX Threads).		2	6/2И		8	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию.	1. Устный опрос (собеседование). 2. Лабораторные работы.	ПК-2.1, ПК-5.1, ПК-5.2, ПК-5.3
Итого по разделу		4	12/4И		16			
3. Распределенные вычисления.								
3.1 Основы распределенных систем. Передача сообщений. Механизмы передачи: очереди, почтовые ящики, порты. Удаленный вызов процедур. Модели взаимодействия процессов. Интерфейс обмена данными MPI. Основные программные примитивы интерфейса MPI.	4	2	4/2И		6	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию.	1. Устный опрос (собеседование). 2. Лабораторные работы.	ПК-2.1, ПК-5.1, ПК-5.2, ПК-5.3
Итого по разделу		2	4/2И		6			
4. Языки, библиотеки и инструментальные средства реализации параллельных вычислений.								
4.1 Параллелизм в языке Java.	4	4	10/4И		12	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию.	1. Устный опрос (собеседование). 2. Лабораторные работы.	ПК-2.1, ПК-5.1, ПК-5.2, ПК-5.3

4.2	Технология OpenMP.	3	8/4И	10,05	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию.	1. Устный опрос (собеседование). 2. Лабораторные работы.	ПК-2.1, ПК-5.1, ПК-5.2, ПК-5.3
Итого по разделу		7	18/8И	22,05			
Итого за семестр		17	34/14И	56,05		зачёт	
Итого по дисциплине		17	34/14И	56,05		зачет	

5 Образовательные технологии

1. Традиционные образовательные технологии, ориентированные на организацию образовательного процесса и предполагающую прямую трансляцию знаний от преподавателя к студенту.

Формы учебных занятий с использованием традиционных технологий:

Информационная лекция – последовательное изложение материала в дисциплинарной логике, осуществляемое преимущественно вербальными средствами (монолог преподавателя).

Лабораторная работа – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.

2. Технологии проблемного обучения – организация образовательного процесса, которая предполагает постановку проблемных вопросов, создание учебных проблемных ситуаций для стимулирования активной познавательной деятельности аспирантов.

3. Интерактивные технологии – организация образовательного процесса, которая предполагает активное и нелинейное взаимодействие всех участников, достижение на этой основе лично-значимого для них образовательного результата.

Формы учебных занятий с использованием специализированных интерактивных технологий:

Лекция «обратной связи» – лекция–провокация (изложение материала с заранее запланированными ошибками), лекция-беседа, лекция-дискуссия, лекция-пресс-конференция.

4. Информационно-коммуникационные образовательные технологии – организация образовательного процесса, основанная на применении программных сред и технических средств работы с знаниями в различных предметных областях.

6 Учебно-методическое обеспечение самостоятельной работы обучающихся

Представлено в приложении 1.

7 Оценочные средства для проведения промежуточной аттестации

Представлены в приложении 2.

8 Учебно-методическое и информационное обеспечение дисциплины

а) Основная литература:

1. Кареева, Е. Д. Основы многопоточного и параллельного программирования: Учебное пособие / Кареева Е.Д. - Краснояр.:СФУ, 2016. - 356 с.: ISBN 978-5-7638-3385-0. - Текст : электронный. - URL: <https://znanium.com/catalog/product/966962> (дата обращения: 29.10.2020). – Режим доступа: по подписке.

2. Немнюгин, С. А. Параллельное программирование для многопроцессорных вычислительных систем: Пособие / Немнюгин С.А., Стесик О.Л. - СПб:БХВ-Петербург, 2014. - 397 с. ISBN 978-5-9775-1877-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/940180> (дата обращения: 29.10.2020). – Режим доступа: по подписке.

б) Дополнительная литература:

1. Богачев, К. Ю. Основы параллельного программирования / Богачев К.Ю., - 3-е изд., (эл.) - Москва :БИНОМ. ЛЗ, 2015. - 345 с.: ISBN 978-5-9963-2995-3. - Текст : электронный. - URL: <https://znanium.com/catalog/product/476284> (дата обращения: 29.10.2020).

2. Практикум по дисциплине "Теория вычислительных процессов" : практикум / А. Н. Калитаев, Ю. В. Кочержинская, В. Д. Тутарова, Д. Н. Мазнин ; МГТУ. - Магнитогорск : МГТУ, 2016. - 1 электрон. опт. диск (CD-ROM). - Загл. с титул. экрана. - URL: <https://magtu.informsystema.ru/uploader/fileUpload?name=2765.pdf&show=dcatalogues/1/132866/2765.pdf&view=true> (дата обращения: 23.10.2020). - Макрообъект. - Текст : электронный. - Сведения доступны также на CD-ROM.

в) Методические указания:

г) Программное обеспечение и Интернет-ресурсы:

Программное обеспечение

Наименование ПО	№ договора	Срок действия лицензии
Eclipse	свободно распространяемое ПО	бессрочно
JetBrains IDEA Community Edition	свободно распространяемое ПО	бессрочно
NetBeans	свободно распространяемое ПО	бессрочно
MS Visual Studio 2017 Community Edition	свободно распространяемое ПО	бессрочно
MS Windows 10 Professional (для классов)	Д-1227-18 от 08.10.2018	11.10.2021

Профессиональные базы данных и информационные справочные системы

Название курса	Ссылка
Национальная информационно-аналитическая система – Российский индекс научного цитирования (РИНЦ)	URL: https://elibrary.ru/project_risc.asp
Информационная система - Единое окно доступа к информационным ресурсам	URL: http://window.edu.ru/
Федеральное государственное бюджетное учреждение «Федеральный институт промышленной собственности»	URL: http://www1.fips.ru/

9 Материально-техническое обеспечение дисциплины (модуля)

Материально-техническое обеспечение дисциплины включает:

1. Лекционная аудитория ауд. 282. Мультимедийные средства хранения, передачи и представления информации.
2. Компьютерные классы Центра информационных технологий ФГБОУ ВО «МГТУ». Персональные компьютеры, объединенные в локальные сети с выходом в Internet, оснащенные современными программно-методическими комплексами для решения задач в области информатики и вычислительной техники.
3. Аудитории для самостоятельной работы: компьютерные классы; читальные залы библиотеки. Все классы УИТ и АСУ с персональными компьютерами, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета.
4. Аудиторий для групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Ауд. 282 и классы УИТ и АСУ.
5. Помещения для самостоятельной работы обучающихся, оснащенных компьютерной техникой с возможностью подключения к сети «Интернет» и наличием доступа в электронную информационно-образовательную среду организации. Классы УИТ и АСУ.
6. Помещения для хранения и профилактического обслуживания учебного оборудования. Центр информационных технологий – ауд. 372.

Учебно-методическое обеспечение самостоятельной работы обучающихся

По дисциплине «Теория и практика распределенных и параллельных вычислений» предусмотрена аудиторная и внеаудиторная самостоятельная работа обучающихся.

Аудиторная самостоятельная работа студентов предполагает выполнение работ на лабораторных занятиях. Внеаудиторная самостоятельная работа обучающихся осуществляется в виде изучения литературы по соответствующему разделу с проработкой материала при подготовке к выполнению лабораторных работ по данной дисциплине.

Примерные задания к лабораторным занятиям:

Работа №1. Параллельное программирование с использованием WinAPI.

Написать многопоточную программу для вычисления приближенного значения интеграла методами прямоугольников, трапеций и парабол (метод Симпсона). Программа с помощью цикла запускает n потоков, которые начинают вычислять значение определенного интеграла по заданной формуле, главный поток ожидает завершения вычислений всеми потоками и затем выводит значение вычислений на экран. Потоки, работая параллельно, используют мьютекс для синхронизации вывода в переменную. Параллельные вычисления в потоках производятся с небольшими случайными задержками.

Функция `CreateThread` создает в памяти пространство для запуска процессом еще одной поточной функции.

Функция `CreateMutex` регистрирует в системе объект Мьютекс, нужный для синхронизации параллельных процессов.

Функция `CreateSemaphore` регистрирует в системе объект Семафор, который, также как и мьютекс, помогает синхронизировать параллельные процессы.

Проверить результат вычисления с помощью подпрограммы, состоящей из одного цикла.

Пример приближенного решения интеграла:

$$\int_0^x \sin(x) dx \approx \sum_{k=0}^{n-1} \frac{\pi}{n} * \sin\left(\frac{\pi k}{n}\right).$$

Работа №2. Параллельное программирование с использованием OpenMP.

Задание 1. Введение в OpenMP. Создайте консольное приложение в среде Visual Studio 2010 с поддержкой OpenMP.

Указания к выполнению задания:

Для включения поддержки OpenMP установите дополнительные параметры компиляции проекта:

- В главном меню выберите *Project-> Имя_проекта Properties*
- В открывшемся окне выберите *Configuration Properties / C/C++ / Language*. Установите для опции *OpenMP Support* значение *Yes(/openmp)*. Нажмите кнопку *OK*. См. рис.1.

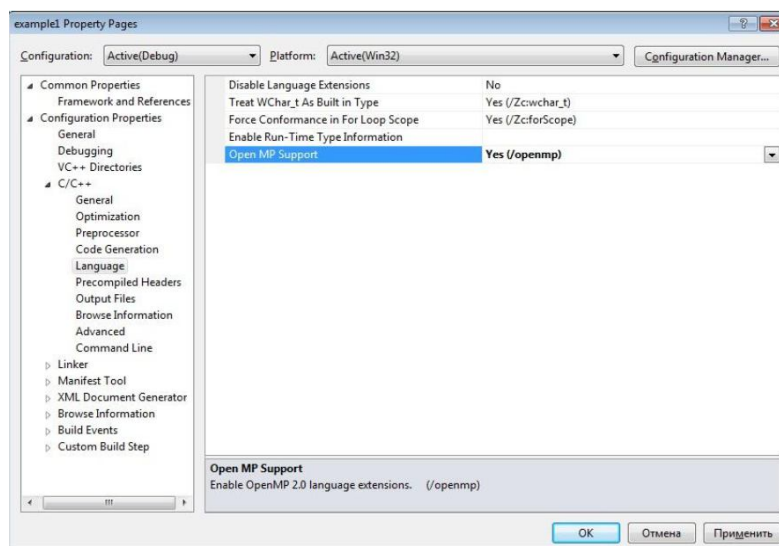


Рис. 1. Окно настройки параметров компиляции проекта

Задание 2. Напишите программу, в которой создается 4 нити и каждая нить выводит на экран строку "Hello World!".

Указания к выполнению задания:

1. Создайте консольное приложение в среде Visual Studio с поддержкой OpenMP.

2. Напишите на языке C/C++ программу, печатающую на экран строку «Hello World!».

3. Подключите заголовочный файл `omp.h` с функциями и переменными OpenMP.

Строка подключения заголовочного файла:

```
#include <omp.h>
```

4. В функции `main` создайте параллельную область с помощью OpenMP-директивы `parallel`. Обратите внимание, что открывающаяся фигурная скобка и название директивы должны находиться в разных строках! Поместите команду вывода строки «Hello World!» внутрь параллельной области.

```
#pragma omp parallel
```

```
{
printf("Hello World!\n");
}
```

5. Задайте количество нитей в параллельной области одним из следующих способов:

Способ 1. Вызовите функцию `omp_set_num_threads()` перед началом параллельной области. В качестве параметра укажите одно целое число – количество нитей в параллельной области:

```
omp_set_num_threads(4);
#pragma omp parallel
{
printf("Hello World!\n");
}
```

Способ 2. Добавьте к директиве `parallel` параметр `num_threads()`. В качестве параметра укажите одно целое число – количество нитей в параллельной области:

```
#pragma omp parallel num_threads(4)
{
printf("Hello World!\n");
}
```

6. Скомпилируйте и запустите ваше приложение. Убедитесь, что строка «Hello World!» выводится на экран столько раз, сколько нитей вы задали в параллельной области.

Задание 3. Использование библиотеки OpenMP для вычисления определенного интеграла функции методами прямоугольников, трапеций, парабол. Выполнить сравнительный анализ результатов работы с результатами, полученными в задании №3.

Сравнительный анализ результатов

Метод вычисления значения интеграла функции	Значение и время вычисления интеграла функции $\sin(x)$ ($0 \leq x \leq \pi$)			
	средствами WinAPI		средствами OpenMP	
	значение	время	значение	время
Метод прямоугольников				
Метод трапеций				
Метод парабол				

Работа №3. Распараллеливание вычислений с помощью OpenMP.

Задание. Реализовать и распараллелить с помощью OpenMP заданный алгоритм (согласно варианту).

Варианты заданий

№ варианта	Размерность задачи	Тип элемента данных	Алгоритм
1.	120	Длинное целое	Сортировка массива с помощью Шейкер-сортировки
2.	20×60	Знаковый короткий целый	Сортировка столбцов матрицы по возрастанию
3.	70	Беззнаковый целый	Сортировка массива методом пузырька
4.	80	Длинное целое	Сортировка массива методом QuickSort (быстрая сортировка)
5.	50	Целое	Сортировка массива методом Шелла.
6.	90	Беззнаковый целый	Сортировка массива с помощью сортировки вставками
7.	70×90	Длинное целое	Сортировка строк матрицы по возрастанию
8.	80×30	Знаковый короткий целый	Сортировка строк матрицы по убыванию
9.	90×60	Беззнаковый целый	Сортировка столбцов матрицы по убыванию
10.	100×50	Число с плавающей запятой	Перемножение матриц
11.	100	Длинное целое	Сортировка массива с помощью пирамидальной сортировки

Работа №4. Многопоточное программирование на языке Java. Класс Thread и интерфейс Runnable

Написать многопоточную программу для решения следующей задачи: *составить алгоритм и программу для нахождения количества четных и нечетных N -значных чисел, состоящих из цифр, которые попарно являются соседними в натуральном ряду. N задается пользователем (например, $N=8$, $N=10$). И провести сравнительный анализ быстродействия работы алгоритмы при разном количестве потоков k , согласно нижеприведенной таблице.*

Сравнительный анализ результатов

Компьютер	Количество найденных чисел (четных / нечетных)	Количество потоков k	Время выполнения*, сек
<i>N</i> -значность числа $N=8$			
Класс Thread		1	
		2	
		4	
		8	
		64	
Интерфейс Runnable		1	
		2	
		4	
		8	
		64	
<i>N</i> -значность числа $N=10$			
Класс Thread		1	
		2	
		4	
		8	
		64	
Интерфейс Runnable		1	
		2	
		4	
		8	
		64	

Работа №5. Многопоточное программирование в решении задач численного интегрирования

Написать многопоточную программу для вычисления приближенного значения интеграла методами прямоугольников, трапеций и парабол (метод Симпсона). Программа запускает k потоков, которые начинают вычислять значение определенного интеграла по заданной формуле, главный поток ожидает завершения вычислений всеми потоками и затем выводит значение вычислений на экран. Потоки, работая параллельно, используют объекты синхронизации для изменения значения общей переменной.

Проверить результат вычисления с помощью подпрограммы, состоящей из одного цикла.

Сравнительный анализ результатов

Метод вычисления значения интеграла функции	Значение и время вычисления интеграла функции $f(x)$ ($a \leq x \leq b$)				
	значение	Время расчета при k -потоках, секунд			
		$k = 1$	$k = 2$	$k = 4$	$k = 8$
Метод прямоугольников					
Метод трапеций					
Метод парабол (Симпсона)					

Варианты заданий:

Вариант	Интеграл
1.	$f(x) = \int_1^{3,5} \frac{\ln(x)}{x\sqrt{1 + \ln(x)}} dx$

2.	$f(x) = \int_{\pi/6}^{\pi/2} (tg^2(x) + ctg^2(x))dx$
3.	$f(x) = \int_2^3 \frac{1}{x \cdot \ln(x)} dx$
4.	$f(x) = \int_1^4 \frac{\ln^2(x)}{x} dx$
5.	$f(x) = \int_1^{\ln(2)} \sqrt{e^x - 1} dx$
6.	$f(x) = \int_0^1 x \cdot e^x \cdot \sin(x) dx$
7.	$f(x) = \int_1^4 \frac{dx}{\sqrt{9 + x^2}}$
8.	$f(x) = \int_0^{\sqrt{2}} x \cdot arctg(x) dx$
9.	$f(x) = \int_1^9 (x \cdot \ln(x))^2 dx$
10.	$f(x) = \int_0^2 arcsin \sqrt{\frac{x}{1+x}} dx$
11.	$f(x) = \int_0^1 \frac{ x^2 - 1 }{(x^2 + 1)\sqrt{x^4 + 1}} dx$
12.	$f(x) = \int_0^{1.5} \frac{e^x \cdot (1 + \sin(x))}{1 + \cos(x)} dx$
13.	$f(x) = \int_0^{3/4} \frac{dx}{(1+x)\sqrt{x^2 + 1}}$

Работа №6. Многопоточное программирование на языке Java. Синхронизация доступа к общему ресурсу нескольких потоков

Нередко возникает ситуация, когда несколько потоков имеют доступ к некоторому объекту, проще говоря, пытаются использовать общий ресурс и начинают мешать друг другу. Более того, они могут повредить этот общий ресурс. Например, когда два потока записывают информацию в файл/объект/поток.

Выполнить вариант синхронизации доступа к файлу для одновременной записи информации N различными потоками.

1) **Задание 1:** В результате в файл должно быть выведено (например, для N = 2 – количество потоков, M = 5 – количество сообщений):

```

First0->0
Second0->0
First1->1
Second1->1
First2->2
Second2->2
First3->3
Second3->3
First4->4
Second4->4

```

2) **Задание 2:** В результате в файл должно быть выведено (например, для N = 3 – количество потоков, M = 4 – количество сообщений):

A
AA
AAA
AAAA
AAAAB
AAAABB
AAAABBB
AAAABBBB
AAAABBBBC
AAAABBBBCC
AAAABBBBCCC
AAAABBBBCCCC

Оценочные средства для проведения промежуточной аттестации

а) Оценочные средства для проведения промежуточной аттестации:

Код индикатора	Индикатор достижения компетенции	Оценочные средства
ПК-2: Способность к анализу проблемной ситуации, разработке требований к системе, постановке целей создания системы, разработке концепции системы и технического задания на создание системы, представления концепции, технического задания на систему и изменений в них заинтересованным лицам		
ПК-2.1	Оценивает выбор средств и методов для проведения системного анализа при проектировании программного обеспечения для мобильных устройств	<p><i>Перечень теоретических вопросов:</i></p> <ol style="list-style-type: none"> 1. Классификация компьютеров по областям применения. Персональные компьютеры и рабочие станции. X-терминалы. Серверы. Мейнфреймы. 2. Современные процессоры. Многопроцессорные системы. 3. Системы высокой готовности и отказоустойчивые системы. 4. Многопроцессорные системы с общей памятью. 5. Многопроцессорные системы с локальной памятью и многомашинные системы. 6. Классификация систем параллельной обработки данных. Кластерные решения Sun Microsystems. Высокопроизводительные вычисления на кластерах. 7. Понятие – «высокопроизводительные средства вычислительной техники». 8. Общие требования, предъявляемые к современным компьютерам. 9. Совместимость и мобильность программного обеспечения. 10. Оценка производительности вычислительных систем. <p><i>Практические задания:</i></p> <ol style="list-style-type: none"> 1. Написать многопоточную программу для вычисления приближенного значения интеграла методом прямоугольников. 2. Написать многопоточную программу для вычисления приближенного значения интеграла методом трапеций. 3. Написать многопоточную программу для вычисления приближенного значения интеграла методом парабол (метод Симпсона). <p><i>Задания на решение задач из профессиональной области, комплексные задания:</i></p> <p>Выполнить вариант синхронизации доступа к файлу для одновременной записи</p>

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		<p>информации N различными потоками.</p> <p>1. В результате в файл должно быть выведено (например, для N = 2 – количество потоков, M = 5 – количество сообщений):</p> <p>First0->0 Second0->0 First1->1 Second1->1 First2->2 Second2->2 First3->3 Second3->3 First4->4 Second4->4</p> <p>2. В результате в файл должно быть выведено (например, для N = 3 – количество потоков, M = 4 – количество сообщений):</p> <p>A AA AAA AAAA AAAAB AAAABB AAAABBB AAAABBBB AAAABBBBC AAAABBBBCC AAAABBBBCCC AAAABBBBCCCC</p>

ПК-5: Способность к формализации и алгоритмизации поставленных задач, к написанию программного кода с использованием языков программирования, определения и манипулирования данными и оформлению программного кода в соответствии установленными требованиями для мобильных устройств

ПК-5.1	Оценивает качество математической модели при формализации задачи предметной области	<p><i>Теоретические вопросы</i></p> <p>1. Технологии, ориентированные на кластеры/суперкомпьютеры. Технология MPI.</p>
ПК-5.2	Оценивает качество разработанных алгоритмов для последующего кодирования	<p>2. Технологии, ориентированные на многоядерные центральные процессоры. Технологии OpenMP, Intel (Intel Cilk Plus, Intel TBB, Intel ArBB).</p> <p>3. Технологии, ориентированные на использование графических процессоров. Технологии NVIDIA CUDA, OpenCL.</p>
ПК-5.3	Оценивает выбор программных средств для программирования и манипулирования данными в соответствии установленными	<p>4. Параллельное программирование с использованием технологии OpenMP.</p> <p>5. Структура программы, использующей средства Win API. Создание, ожидание и завершение работы потоков. Критические секции. Семафоры.</p> <p>6. Алфавитный указатель по директивам,</p>

Код индикатора	Индикатор достижения компетенции	Оценочные средства
	требованиями	<p>функциям, опциям и переменным окружения OpenMP.</p> <ol style="list-style-type: none"> 7. OpenMP. Параллельные и последовательные области. 8. OpenMP. Разработка параллельных алгоритмов и программ для решения задач вычислительной математики. 9. Синхронизация в OpenMP. 10. Способ создания и запуска потока на основе расширения класса Thread. Пример. 11. Способ создания и запуска потока на основе реализации интерфейса Runnable. Пример. 12. Пакет java.util.concurrent. Перечисление TimeUnit. 13. Объекты синхронизации. Блокирующие очереди. Пример. 14. Объекты синхронизации. Семафоры. Пример. 15. Объекты синхронизации. Барьеры. Пример. 16. Объекты синхронизации. «Щеколда». Пример. 17. Объекты синхронизации. Обмен блокировками. Пример. 18. Объекты синхронизации. Альтернатива synchronized. Интерфейсы Lock. Пример. 19. Объекты синхронизации. Класс ExecutorService и интерфейс Callable. Пример. 20. Объекты синхронизации. Класс Phaser. Пример. 21. Способы создания и запуска потока. 22. Управление потоками. Жизненный цикл потока. 23. Управление приоритетами и группами потоков. 24. Потоки-демоны. 25. Методы и инструкции (блок кода) synchronized. 26. Контроль за доступом к объекту-ресурсу (монитор). Методы wait(), notify() и notifyAll(). 27. Пакет java.util.concurrent. Способы управления потоками. <p><i>Практические задания</i></p> <p>Применять технологии параллельных вычислений для высокопроизводительной реализации решения задач обработки данных:</p> <p><i>Задания:</i></p> <ol style="list-style-type: none"> 1. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		<p>реализацию алгоритма, реализующего перемножение двух квадратных матриц $A[M][M]$.</p> <p>2. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждого столбца матрицы $A[M][N]$ по возрастанию.</p> <p>3. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждого столбца матрицы $A[M][N]$ по убыванию абсолютных величин.</p> <p>4. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждой строки матрицы $A[M][N]$ по возрастанию абсолютных величин.</p> <p>5. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждой строки матрицы $A[M][N]$ по убыванию суммы значений цифр элементов матрицы.</p> <p>6. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждого столбца матрицы $A[M][N]$ по возрастанию суммы значений цифр элементов матрицы.</p> <p>7. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки всех нечетных элементов строк матрицы $A[M][N]$ по возрастанию.</p> <p>8. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки всех четных элементов столбцов матрицы $A[M][N]$ по убыванию.</p> <p>9. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждого столбца матрицы $A[M][N]$ по возрастанию абсолютных величин.</p> <p>10. На основе многопоточного</p>

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		<p>распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждой строки матрицы $A[M][N]$ по убыванию абсолютных величин.</p> <p>11. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждой четной строки матрицы $A[M][N]$ по возрастанию, каждого четного столбца по возрастанию.</p> <p>12. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждой четной строки матрицы $A[M][N]$ по возрастанию, каждого нечетного столбца по возрастанию абсолютных величин.</p> <p>13. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждой нечетной строки матрицы $A[M][N]$ по возрастанию абсолютных величин, каждого четного столбца по возрастанию.</p> <p>14. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждой строки матрицы матрицы $A[M][N]$ по возрастанию.</p> <p>15. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждого четного столбца матрицы $A[M][N]$ по убыванию, каждой строки по убыванию.</p> <p>16. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждой строки матрицы $A[M][N]$ по возрастанию четных чисел.</p> <p>17. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждой строки матрицы $A[M][N]$ по возрастанию, каждого столбца по убыванию.</p> <p>18. На основе многопоточного</p>

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		<p>распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждой строки матрицы $A[M][N]$ по возрастанию отрицательных величин.</p> <p>19. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждого четного столбца матрицы $A[M][N]$ по убыванию суммы значений цифр элементов матрицы.</p> <p>20. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждой нечетной строки матрицы $A[M][N]$ по возрастанию суммы значений цифр элементов матрицы.</p>

б) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:

Промежуточная аттестация по дисциплине «Теория и практика распределенных и параллельных вычислений» включает теоретические вопросы, позволяющие оценить уровень усвоения обучающимися знаний, и практические задания, выявляющие степень сформированности умений и владений, проводится в форме зачета.

Зачет по дисциплине проводится по результатам отчетности на практических занятиях с опросом в устной форме по этапам выполнения и активного выступления в беседе-обсуждении на лекционных занятиях.

Показатели и критерии для зачета:

– на оценку «**зачтено**» – обучающийся демонстрирует уровень сформированности компетенций, знание учебного материала, свободно выполняет практические задания, свободно оперирует знаниями, умениями, применяет их в различных ситуациях.

– на оценку «**не зачтено**» - обучающийся не может показать знания на уровне воспроизведения и объяснения информации, не может показать интеллектуальные навыки решения простых задач.