

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»

УТВЕРЖДАЮ
Проректор по учебной работе

_____ Василенко В.Н.

«25» мая 2023

РАБОЧАЯ ПРОГРАММА
ДИСЦИПЛИНЫ

Параллельное программирование
(наименование в соответствии с РУП)

Специальность

10.05.03 Информационная безопасность автоматизированных систем
(шифр и наименование направления подготовки/специальности)

Специализация

Безопасность открытых информационных систем
(наименование профиля/специализации)

Квалификация выпускника

специалист по защите информации

(в соответствии с Приказом Министерства образования и науки РФ от 12 сентября 2013 г. N 1061 "Об утверждении перечней специальностей и направлений подготовки высшего образования" (с изменениями и дополнениями))

1. Цели и задачи дисциплины

Целью освоения дисциплины «Параллельное программирование» является формирование компетенций обучающегося в области профессиональной деятельности и сфере профессиональной деятельности:

- 06 Связь, информационные и коммуникационные технологии (в сфере обеспечения безопасности информации в автоматизированных системах).

Дисциплина направлена на решение задач профессиональной деятельности научно-исследовательского, проектного, контрольно-аналитического, эксплуатационного типов.

Программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по специальности 10.05.03 Информационная безопасность автоматизированных систем.

2. Перечень планируемых результатов обучения, соотнесенных с планируемыми результатами освоения образовательной программы

№ п/п	Код компетенции	Формулировка компетенции	Код и наименование индикатора достижения компетенции
1	ОПК-7	Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ;	ИД1опк-7 – обладает способностью применять языки, системы и инструментальные средства программирования в профессиональной деятельности

Код и наименование индикатора достижения компетенции	Результаты обучения (показатели оценивания)
ИД1опк-7 – обладает способностью применять языки, системы и инструментальные средства программирования в профессиональной деятельности	Знает: основные подходы к разработке параллельных программ; инструментальные средства параллельного программирования; методы создания параллельных программ для задач многопоточного программирования в профессиональной деятельности.
	Умеет: создавать параллельные программы для вычислительных систем с распределенной, общей оперативной памятью; проводить распараллеливание вычислительных алгоритмов; строить модель выполнения параллельных программ; оценивать эффективности параллельных вычислений; анализировать сложность вычислений и возможность распараллеливания разрабатываемых алгоритмов; применять общие схемы разработки параллельных программ и систем.
	Владеет: навыками параллельного программирования для решения профессиональных задач, с помощью современных инструментальных средств; навыками создания параллельных программ для вычислительных систем с распределенной, общей оперативной памятью.

3. Место дисциплины в структуре ООП ВО

Дисциплина относится к обязательной части Блока 1 ООП. Дисциплина является обязательной к изучению.

Изучение дисциплины основано на знаниях, умениях и навыках, полученных при изучении обучающимися дисциплин: Информатика, Языки программирования, Алгоритмы и структуры данных, Проектирование и архитектура программного обеспечения, Математический анализ, Линейная алгебра и аналитическая геометрия, Иностранный язык, Программирование на Java, Технологии и методы программирования.

Дисциплина является предшествующей для изучения: Разработка и эксплуатация автоматизированных систем в защищенном исполнении, Средства

проектирования для аппаратных средств защиты информации, Защита информации от утечки по техническим каналам, практик.

4. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины составляет 7 зачетных единиц.

Виды учебной работы	Всего ак. ч	Распределение трудоемкости по семестрам, ак. ч	
		8 семестр	9 семестр
Общая трудоемкость дисциплины (модуля)	252	108	144
Контактная работа в т. ч. аудиторные занятия:	112,1	57,1	55
Лекции	66	36	30
<i>в том числе в форме практической подготовки</i>	-	-	-
Практические занятия	66	36	30
<i>в том числе в форме практической подготовки</i>	-	-	-
Консультации текущие	3,3	1,8	1,5
Консультации перед экзаменом	2	–	2
Вид аттестации (зачет/экзамен)	0,3	0,1	0,2
Самостоятельная работа:	80,6	34,1	46,5
Проработка материалов по лекциям, учебникам, учебным пособиям	33	18	15
Подготовка к практическим занятиям	24	9	15
Курсовой проект/работа	-	-	-
Домашнее задание, реферат,	15	5	10
Другие виды самостоятельной работы	8,6		6,5
Контроль (подготовка к экзамену)	33,8	–	33,8

5 Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

5.1 Содержание разделов дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела	Трудоемкость раздела, ак.ч
8 семестр			
1	Введение. Создание потоков. Синхронизация. Атомарные снимки регистров	Введение в высокопроизводительные вычисления Тенденции развития вычислительных систем, обуславливающие необходимость применения распределенных (параллельных) методов вычислений. Примеры вычислительно емких задач из разных областей науки. Классификация параллельных систем (SIMD, MISD..., SMP, MPP) Современные высокопроизводительные системы: начиная от расширений SSE, через многоядерность к узлам кластеров Понятия ускорения, эффективности (закон Амдала) Многопоточность или IPC	51,1
2	Характерные ошибки, профилирование, пулы потоков и шаблоны	Создание/завершение потоков Механизм запуска потока. Корректное завершение потоков: cancellation points, interrupted exception, примеры кода в glibc. Сравнение	55

		различных потоков (POSIX, boost, java). Проброс исключений между потоками	
	<i>Консультации текущие</i>		1,9
	<i>Консультации перед экзаменом</i>		-
	<i>Зачет</i>		-
9 семестр			
3	Кластеры, сети Петри. Модели памяти, асинхронный ввод/вывод	<p>Кластерные вычисления Виды кластерных систем: Балансировки нагрузки, Высокой надёжности, Вычислительные. История и назначение стандарта MPI. Обмен сообщениями: С блокировкой, Без блокировки, Отложенные запросы на взаимодействие. Взаимодействие процессов: Группы и коммуникаторы, Операции коллективного взаимодействия процессов, Редукция, Виртуальные топологии</p> <p>Консенсус. Сети Петри Консенсус: Консенсусное число RMW- регистров, Универсальность CAS-операций. Верификация программ (сети Петри) Транзакционная память Идея transactional memory: Software transactional memory, Hardware transactional memory. Преимущества и круг задач. Реализация HTM на линейках кэша. Lock teleportation Асинхронный ввод/вывод Блокирующий/неблокирующий. Синхронный (реактор)/асинхронный (проактор). Преимущества асинхронной работы и реализация со стороны операционной системы. Библиотеки асинхронного ввода/вывода Модель памяти Устройство кэшей процессора. Пример на протоколе MESI. Барьеры памяти (store/load). Модели памяти: Sequential consistency.... Acquire/release семантика</p>	52
4	Линеаризуемость, lock-free структуры данных	<p>Линеаризуемость Понятие линеаризуемости. Lock-free стек Trieber. Пример на очередях. Lock-free очередь Michael & Scott. Точки линеаризации Flat-Combining Схема Flat-Combining. Возможные оптимизации за счёт интерференции операций. Сравнение производительности с lock-free очередью Michael & Scott Древовидные Lock-free структуры данных User-space RCU. MultiArray. Схемы управления памятью: Tagged pointers, Hazard pointer OpenCL Архитектура вычислений на GPU. Принципы работы со стандартом OpenCL</p>	54,5
	<i>Консультации текущие</i>		33,8
	<i>Консультации перед экзаменом</i>		3,7
	<i>Экзамен</i>		-

5.2 Разделы дисциплины и виды занятий

№ п/п	Наименование раздела дисциплины	Лекции, ак. ч	Практические/лабораторные занятия, ак. ч	СРО, ак. ч
8 семестр				
1	Введение. Создание потоков. Синхронизация. Атомарные снимки регистров	18	18	15,1
2	Характерные ошибки, профилирование, пулы потоков и шаблоны	18	18	19
	<i>Консультации текущие</i>		1,9	
	<i>Консультации перед экзаменом</i>		-	
	<i>Зачет</i>		-	
9 семестр				
3	Кластеры, сети Петри. Модели памяти, асинхронный ввод/вывод	15	15	22
4	Линеаризуемость, lock-free структуры данных	15	15	24,5
	<i>Консультации текущие</i>		33,8	
	<i>Консультации перед экзаменом</i>		3,7	
	<i>Экзамен</i>		-	

5.2.1 Лекции

№ п/п	Наименование раздела дисциплины	Тематика лекционных занятий	Трудоемкость, ак. ч
1	Введение. Создание потоков. Синхронизация. Атомарные снимки регистров	Примитивы синхронизации Необходимость синхронизации: простые гонки данных. Алгоритмы синхронизации Грубая. Тонкая. Оптимистичная. Ленивая. Неблокирующая (параллель с ORM) Атомарные снимки регистров Классификация алгоритмов: lock-free, wait-free. SWMR-регистры. Lock-free snapshot. Wait-free snapshot	18
2	Характерные ошибки, профилирование, пулы потоков и шаблоны	Ошибки параллельного программирования Основные ошибки многопоточного программирования: Гонки данных (Data Race), Взаимная блокировка (Deadlock). Специфические ошибки: Реакция потока на сигнал, Блокировки при fork многопоточных программ, Проблема ABA, Инверсия приоритетов Профилирование многопоточных приложений Средства анализа производительности: Утилита time, Intel Parallel Studio, Valgrind (модули callgrind, cachegrind). Структурные шаблоны: Декомпозиция по задачам, Геометрическая декомпозиция, Recursive Data, Pipeline. Некоторые программные структуры: Parallel loops, Boss/Worker. Разное: Double check, Local Serializer	18
3	Кластеры, сети Петри. Модели	Кластерные вычисления Виды кластерных систем: Балансировки	15

	памяти, асинхронный ввод/вывод	нагрузки, Высокой надёжности, Вычислительные. История и назначение стандарта MPI. Обмен сообщениями: С блокировкой, Без блокировки, Отложенные запросы на взаимодействие. Взаимодействие процессов: Группы и коммутаторы, Транзакционная память Идея transactional memory: Software transactional memory, Hardware transactional memory. Преимущества и круг задач. Преимущества асинхронной работы и реализация со стороны операционной системы. Библиотеки асинхронного ввода/вывода Модель памяти Устройство кэшей процессора. Пример на протоколе MESI. Барьеры памяти (store/load). Модели памяти: Sequential consistency.... Acquire/release семантика	
4	Линеаризуемость, lock-free структуры данных	Линеаризуемость Понятие линеаризуемости. Lock-free стек Trieber. Пример на очередях. Lock-free очередь Michael & Scott. Точки линеаризации Flat-Combining Схема Flat-Combining. Возможные оптимизации за счёт интерференции операций. Сравнение производительности с lock-free очередью Michael & Scott Древовидные Lock-free структуры данных User-space RCU. MultiArray. Схемы управления памятью: Tagged pointers, Hazard pointer OpenCL Архитектура вычислений на GPU. Принципы работы со стандартом OpenCL	15

5.2.2 Практические занятия (семинары)

№ п/п	Наименование раздела дисциплины	Тематика практических занятий (семинаров)	Трудоемкость, ак. ч
1	Введение. Создание потоков. Синхронизация. Атомарные снимки регистров	Реализация примитивов синхронизации: алгоритм булочника. Виды мьютексов: рекурсивные/нерекурсивные, read/write, spin, futex. Корректные захват/освобождение примитивов. CAS-операции и атомики. Условные переменные: использование wait/notify, Spurious wakeups	18
2	Характерные ошибки, профилирование, пулы потоков и шаблоны	Пример поиска узких мест Java.util.concurrent и пулы потоков Пулы потоков, корректное завершение пула. Контроль задач через Future. Потокбезопасные контейнеры OpenMP и Intel TBB Обзор OpenMP: параллельные секции, области видимости переменных, ограничения. Обзор Intel TBB: алгоритмы, аллокаторы, деревья задач,	18

		особенности планирования (work stealing...), flow graphs (параллель с BPEL) Шаблоны параллельного программирования	
3	Кластеры, сети Петри. Модели памяти, асинхронный ввод/вывод	Операции коллективного взаимодействия процессов, Редукция, Виртуальные топологии Консенсус. Сети Петри Консенсус: Консенсусное число RMW-регистров, Универсальность CAS-операций. Верификация программ (сети Петри)	15
4	Линеаризуемость, lock-free структуры данных	Реализация НТМ на линейках кэша. Lock teleportation Асинхронный ввод/вывод Блокирующий/неблокирующий. Синхронный (реактор)/асинхронный (проактор).	15

5.2.3 Лабораторный практикум - не предусмотрен

5.2.4 Самостоятельная работа обучающихся

№ п/п	Наименование раздела дисциплины	Вид СРО	Трудоемкость, ак. ч
1	Введение. Создание потоков. Синхронизация. Атомарные снимки регистров	Проработка материалов по лекциям, учебникам, учебным пособиям	9
		Подготовка к практическим/лабораторным занятиям	2
		Виды СРО (Создание программ без графической оболочки)	2
		Виды СРО (Оформление текста отчета по практической работе)	2,1
2	Характерные ошибки, профилирование, пулы потоков и шаблоны	Проработка материалов по лекциям, учебникам, учебным пособиям	9
		Подготовка к практическим занятиям	2
		Виды СРО (Создание программ без графической оболочки)	5
		Виды СРО (Оформление текста отчета по практической работе)	2
3	Кластеры, сети Петри. Модели памяти, асинхронный ввод/вывод	Проработка материалов по лекциям, учебникам, учебным пособиям	7
		Подготовка к практическим занятиям	7
		Виды СРО (Создание программ без графической оболочки)	5
		Виды СРО (Оформление текста отчета по практической работе)	2
4	Линеаризуемость, lock-free структуры данных	Виды СРО (Анализ и расчет по известным математическим моделям)	1
		Проработка материалов по лекциям, учебникам, учебным пособиям	8
		Подготовка к практическим занятиям	8
		Виды СРО (Создание программ без графической оболочки)	4
		Виды СРО (Оформление текста отчета по практической работе)	2
		Виды СРО (Анализ и расчет по известным математическим моделям)	2,5

6 Учебно-методическое и информационное обеспечение дисциплины (модуля)

Для освоения дисциплины обучающийся может использовать:

6.1 Основная литература

1. Федорова, Г. Н. Разработка модулей программного обеспечения для компьютерных систем [Текст] : учебник (гриф МО/ФИРО) / Г. Н. Федорова. - 2-е изд.,

стер. - М. : Академия, 2018. - 384 с. - (Профессиональное образование). - 15 экз. - Библиогр.: с. 378-379. - ISBN 978-5-4468-6292-3 : 899-03.

2. Семакин, И. Г. Основы алгоритмизации и программирования [Текст] : учебник (гриф МО/ФИРО) / И. Г. Семакин. - 2-е изд., стер. - М. : Академия, 2018. - 304 с. - (Профессиональное образование). - 5 экз. - Библиогр.: с. 298-299. - ISBN 978-5-4468-6228-3 : 629-20.

3. Павловская, Т. С/С++. Процедурное и объектно-ориентированное программирование [Текст] : учебник для студ. вузов (гриф МО) / Т. Павловская. - СПб. : Питер, 2015. - 496 с. - (Учебник для вузов. Стандарт третьего поколения). - 10 экз. - ISBN 978-5-469-00109-0 : 841-50.

4. Васильев, А. Н. Java. Объектно-ориентированное программирование [Текст] : для магистров и бакалавров : базовый курс по объектно-ориентированному программированию / А. Н. Васильев. - СПб. : Питер, 2011. - 400 с. - (Учебное пособие. Стандарт третьего поколения). - ISBN 978-5-49807-948-6 : 215-00.

6.2 Дополнительная литература

1. Стуколов, С. В. Параллельное программирование. Практикум : учебное пособие / С. В. Стуколов. — Кемерово : КемГУ, 2020. — 273 с. — ISBN 978-5-8353-2723-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/173547>

2. Богачёв, К. Ю. Основы параллельного программирования : учебное пособие / К. Ю. Богачёв. — 4-е изд. — Москва : Лаборатория знаний, 2020. — 345 с. — ISBN 978-5-00101-758-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/135516>

3. Гуськова, О. И. Объектно ориентированное программирование в Java : учебное пособие : [16+] / О. И. Гуськова. — Москва : Московский педагогический государственный университет (МПГУ), 2018. — 240 с. : ил. — Режим доступа: по подписке. — URL: <https://biblioclub.ru/index.php?page=book&id=500355>. — Библиогр. в кн. — ISBN 978-5-4263-0648-6. — Текст : электронный.

4. Зайцева, О. С. Технологии разработки web-ресурсов : учебное пособие : [16+] / О. С. Зайцева ; Тюменский индустриальный университет. — Тюмень : Тюменский индустриальный университет, 2020. — 75 с. : ил., табл. — Режим доступа: по подписке. — URL: <https://biblioclub.ru/index.php?page=book&id=611103>. — ISBN 978-5-9961-2274-5. — Текст : электронный.

5. Наир, В. Предметно-ориентированное проектирование в Enterprise Java : руководство / В. Наир ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2020. — 306 с. — ISBN 978-5-97060-872-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/179503>

6. Коузен, К. Современный Java: рецепты программирования / К. Коузен. — Москва : ДМК Пресс, 2018. — 275 с. — ISBN 978-5-97060-134-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/116121>.

7. Кареева, Е. Д. Основы многопоточного и параллельного программирования : учебное пособие / Е. Д. Кареева ; Сибирский федеральный университет, Институт вычислительного моделирования Сибирского отделения Российской академии наук, Сибирский научно-образовательный центр суперкомпьютерных технологий. — Красноярск : Сибирский федеральный университет (СФУ), 2016. — 355 с. : ил. — Режим доступа: по подписке. — URL: <https://biblioclub.ru/index.php?page=book&id=497217>

6.3 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся

1. Данылиев, М. М. Методические указания для обучающихся по освоению дисциплин (модулей) в ФГБОУ ВО ВГУИТ [Электронный ресурс] : методические указания для обучающихся на всех уровнях высшего образования / М. М. Данылиев, Р. Н. Плотникова ; ВГУИТ, Учебно-методическое управление. - Воронеж : ВГУИТ, 2016. - 32 с. - Электрон. ресурс. - <http://biblos.vsu.ru/ProtectedView/Book/ViewBook/2488>

2. Защита Web-приложений [Текст] : учебное пособие / А. В. Скрыпников [и др.] ; ВГУИТ, Кафедра информационной безопасности. - Воронеж : ВГУИТ, 2020. - 75 с. - 25 экз. + Электрон. ресурс. - <http://biblos.vsu.ru/ProtectedView/Book/ViewBook/1766>. - Библиогр.: с. 73-74. - ISBN 978-5-00032-469-1 : 799-00. <http://biblos.vsu.ru/ProtectedView/Book/ViewBook/1766>

3. Черняева, С. Н. Алгоритмизация и программирование [Электронный ресурс] : методические указания к самостоятельной работе обучающихся для студентов, обучающихся по направлению 09.03.03 – “Прикладная информатика”, дневной формы обучения / С. Н. Черняева, Л. А. Коробова, И. С. Толстова ; ВГУИТ, Кафедра высшей математики и информационных технологий. - Воронеж, 2020. - 20 с. - Электрон. ресурс. - <http://biblos.vsu.ru/ProtectedView/Book/ViewBook/1910>

6.4 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)

Наименование ресурса сети «Интернет»	Электронный адрес ресурса
«Российское образование» - федеральный портал	https://www.edu.ru/
Научная электронная библиотека	https://elibrary.ru/defaultx.asp?
Национальная исследовательская компьютерная сеть России	https://niks.su/
Информационная система «Единое окно доступа к образовательным ресурсам»	http://window.edu.ru/
Электронная библиотека ВГУИТ	http://biblos.vsu.ru/megapro/web
Сайт Министерства науки и высшего образования РФ	https://minobrnauki.gov.ru/
Портал открытого on-line образования	https://npoed.ru/
Электронная информационно-образовательная среда ФГБОУ ВО «ВГУИТ»	https://education.vsu.ru/

6.5 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем

При изучении дисциплины используется программное обеспечение и информационные справочные системы: информационная среда для дистанционного обучения «Moodle», автоматизированная информационная база «Интернет-тренажеры», «Интернет-экзамен», Microsoft Windows Server 2003, Microsoft Office 2007, Russian Academic OPEN No Level #44822753 Linux, VM Oracle -бесплатное.

При освоении дисциплины используется лицензионное и открытое программное обеспечение – н-р, ОС Windows

7 Материально-техническое обеспечение дисциплины (модуля)

Аудитория для проведения лекционных занятий:

Ауд. 420: Комплекты мебели для учебного процесса. ПЭВМ-11 (компьютер Core i5-4460), проектор Acer projector X1383WH, экран, стенды – 5 шт., блок управления комплекса радиоконтроля и поиска радиопередающих устройств «ОМЕГА» (переносной), МУ защиты ресурсов сети от внутренних и внешних атак CISCO ASA5505-KB, переносной комплекс для автоматизации измерений при проведении исследований и контроля технических средств ЭВТ «НАВИГАТОР-ПЗГ»; средство активной защиты информации от утечки за счет побочных электромагнитных излучений и наводок «СОНАТА-РЗ.1»; система защиты речевой информации «Соната-АВ-4Б» (Центральный блок питания и управления + Размыкатели в составе СВАЗ Соната АВ); профессиональный обнаружитель скрытых видеочамер СОКОЛ-М (переносной);

портативный обнаружитель закладок Protect1203 (переносной); устройство активной защиты информации «ВЕТО-М»; электронный замок Samsung SHS-2920; средство активной защиты информации изделие «Салют 2000С» с регулятором выходного уровня шума

Ауд. 420: Microsoft Windows 7 (64 разрядная); Microsoft Office (standart) 2007; Microsoft Access 2007; Microsoft Project 2007; Microsoft Share Point 2007; Microsoft Visio 2007; Microsoft SQL server 2008; 1 С Предприятие Лицензия; 7-Zip File Manager (архиватор); Adobe Acrobat Reader; Adobe Flash Player; FAR file manager; Google Chrome; Java TM 7 (64-bit); K-Lite Codec Pack; Mozilla Firefox; Oracle VM VirtualBox; Sublime Text; Symantec Endpoint Protection 12 (Заменен на AVP Kaspersky); VMWare Player; Антивирус “Зоркий глаз”; Lazarus; SmathStudio; NanoCAD; Gimp (графический редактор, аналог Photoshop); Avidemux (видео редактор); Virtual Dub (видео редактор); Free Pascal; Страж NT вер.3.0 Сертификат ФСТЭК № 2145 30.07.2013 г.; Ревизор 1XP Сертификат ФСТЭК № 989 08.02.2015 г.; Ревизор 2XP Сертификат ФСТЭК № 990 08.02.2015 г.; Фикс 2.0.2 Сертификат ФСТЭК №1548 15.01.2015 г.; Ревизор сети вер.3.0 Сертификат ФСТЭК №3413 02.06.2015 г.; СЗИ Панцирь К Сертификат ФСТЭК №1973 09.12.2015 г.; СЗИ Dallas Lock 8.0 К Сертификат ФСТЭК №2720 25.09.2015; СЗИ Dallas Lock 8.0 С Сертификат ФСТЭК №2945 16.08.2013

Аудитории для проведения занятий лекционного типа, практических занятий

Ауд. 332а: Комплекты мебели для учебного процесса. ПЭВМ – 12 (компьютер Core i5-4570), стенды – 5 шт.

Ауд.332а: ОС Alt Linux (Альт Образование 8.2) Geany. Lazarus. Qt Creator. Quanta Plus. Веб-редактор Bluefish. Среда разработки Code::Blocks. Офисный пакет Libre Office 5.4: Base, Calc, Draw, Impress, Math, Writer. Персональная бухгалтерия HomeBank. Словарь Star Dict. iTest. VM Maxima. Кумир. Avidemux. Audacios. Brasero. Cheese. SMPlayer. Медиаплеер Parole. Редактор тегов Easy TAG. Smath Studio. Pinta. Веб-браузер Mozilla Firefox. Графический редактор. FP – free Pascal.

8 Оценочные материалы для промежуточной аттестации обучающихся по дисциплине (модулю)

Оценочные материалы (ОМ) для дисциплины (модуля) включают в себя:

- перечень компетенций с указанием индикаторов достижения компетенций, этапов их формирования в процессе освоения образовательной программы;
- описание шкал оценивания;
- типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков;
- методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности.

ОМ представляются отдельным комплектом и **входят в состав рабочей программы дисциплины.**

Оценочные материалы формируются в соответствии с П ВГУИТ «Положение об оценочных материалах».

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

по дисциплине

Параллельное программирование

1 Перечень компетенций с указанием этапов их формирования

п/п	Код компетенции	Формулировка компетенции	Код и наименование индикатора достижения компетенции
1	ОПК-7	Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ;	ИД1опк-7 – обладает способностью применять языки, системы и инструментальные средства программирования в профессиональной деятельности

Код и наименование индикатора достижения компетенции	Результаты обучения (показатели оценивания)
ИД1опк-7 – обладает способностью применять языки, системы и инструментальные средства программирования в профессиональной деятельности	Знает: основные подходы к разработке параллельных программ; инструментальные средства параллельного программирования; методы создания параллельных программ для задач многопоточного программирования в профессиональной деятельности.
	Умеет: создавать параллельные программы для вычислительных систем с распределенной, общей оперативной памятью; проводить распараллеливание вычислительных алгоритмов; строить модель выполнения параллельных программ; оценивать эффективности параллельных вычислений; анализировать сложность вычислений и возможность распараллеливания разрабатываемых алгоритмов; применять общие схемы разработки параллельных программ и систем.
	Владеет: навыками параллельного программирования для решения профессиональных задач, с помощью современных инструментальных средств; навыками создания параллельных программ для вычислительных систем с распределенной, общей оперативной памятью.

2 Паспорт оценочных материалов по дисциплине

№ п/п	Разделы дисциплины	Индекс контролируемой компетенции (или ее части)	Оценочные материалы		Технология/процедура оценивания (способ контроля)
			наименование	№№ заданий	
1	Введение. Создание потоков. Синхронизация. Атомарные снимки регистров	ИД1опк-7	Тест	1-10	Бланочное или компьютерное тестирование
			Практическая работа	1-2	Защита практической работы
			Кейс-задача	1-2	Проверка преподавателем
2	Характерные ошибки, профилирование, пулы потоков и шаблоны	ИД1опк-7	Тест	11-20	Бланочное или компьютерное тестирование
			Практическая работа	3-4	Защита практической работы
			Кейс-задача	3-4	Проверка преподавателем
3	Кластеры, сети Петри. Модели памяти, асинхронный ввод/вывод	ИД1опк-7	Тест	21-30	Бланочное или компьютерное тестирование
			Практическая работа	5-6	Защита практической работы
			Кейс-задача	5-6	Проверка преподавателем
4	Линеаризуемость, lock-free структуры данных	ИД1опк-7	Практическая работа	7-10	Защита практической работы
			Кейс-задача	7-10	Проверка преподавателем

3 Оценочные материалы для промежуточной аттестации.

Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Аттестация обучающегося по дисциплине проводится в форме тестирования (или письменного ответа или выполнения расчетно-графической (практической) работы или решения контрольных задач и т.п.) и предусматривает возможность последующего собеседования (зачета, экзамена).

Каждый вариант теста включает 30 контрольных заданий, из них:

- 10 контрольных заданий на проверку знаний;
- 10 контрольных заданий на проверку умений;
- 10 контрольных заданий на проверку навыков;

Или

Каждый билет включает 3 контрольных вопросов (задач), из них:

- 1 контрольный вопрос на проверку знаний;
- 1 контрольный вопрос на проверку умений;
- 1 контрольная задача на проверку навыков.

3.1 Тесты (тестовые задания)

3.1.1 Шифр и наименование компетенции ОПК-7 Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ.

№ задания	Тестовое задание с вариантами ответов и правильными ответами
1.	Чем отличаются потоки данных (Д) от процессов (П) параллельной задачи, порождаемых в многопроцессорных системах? (Символ означает взаимодействие) 1)Ничем; 2)Д ₁ ↔ Д ₂ ; 3)П ₁ ↔ П ₂ ; 4)Д _к ↔ П _к ;
2.	Что определяет момент запуска на исполнение последовательного фрагмента в многопроцессорной параллельной задаче? 1)Готовность данных; 2)Тактовый генератор; 3)Вызов команд; 4)Семафор
3.	Что представляет собою процессорный объект в параллельном языке C\C++? 1)Структура; 2)Блок; 3)Функция; 4)Код программы;
4.	Любому процессу параллельной программы априори предоставлена возможность обмениваться данными через глобальный системный канал связи 1)MPI_Intracomm(...); 2)MPI_Intercomm(...); 3)MPI_COMM_WORLD; 4)MPI_GROUP_WORLD;
5.	Какую операцию с группами выполнит подпрограмма ^ MPI_Group_intersection(group1, group2, *newgr) ? 1)разность; 2)конкатенацию; 3)сравнение; 4)объединение;
6.	Имена групп процессов при их объявлении должны иметь тип 1)MPI_Group; 2)MPI_Comm; 3)MPI_Status; 4)MPI_Init;
7.	Максимальное значение эффективности, получаемое при использовании параллельного алгоритма для р процессора: а)0; б) 1; в)р; г)8
8.	После завершения вызова MPI_Wait:

	<ul style="list-style-type: none"> а) неблокирующий обмен не выполнен б) возобновляется выполнение всех процессов, относящихся к данной параллельной программе в) неблокирующий обмен выполнен
9.	<p>BIOS хранится ...</p> <ul style="list-style-type: none"> а) в ПЗУ б) в ОЗУ в) на HDD г) на дискете
10.	<p>Параллельное программирование с использованием MPI имеет дело с параллелизмом на уровне:</p> <ul style="list-style-type: none"> (1) задач (2) заданий (3) машинных команд
11.	<p>Особенностью параллельной модели программирования является:</p> <ul style="list-style-type: none"> (1) нелокальный и динамический характер ошибок (2) вероятность утраты детерминизма выполнения программы (3) вероятность возникновения блокировок
12.	<p>Потоки выполнения, относящиеся к одному параллельному приложению, характеризуются:</p> <ul style="list-style-type: none"> (1) общим адресным пространством (2) общими дескрипторами файлов (3) разными и непересекающимися адресными пространствами
13.	<p>MPICH это:</p> <ul style="list-style-type: none"> (1) программный пакет, позволяющий разрабатывать параллельные программы на основе модели обмена сообщениями (2) язык программирования, позволяющий разрабатывать многопоточные программы для вычислительных систем с общей памятью (3) коммерческая библиотека, реализующая операции обмена сообщениями
14.	<p>Одним из распространённых средств разработки многопоточных программ является:</p> <ul style="list-style-type: none"> (1) OpenMP (2) PVM (3) MPI
15.	<p>При неблокирующем двухточечном обмене:</p> <ul style="list-style-type: none"> (1) буфер передачи/приема можно использовать сразу после завершения вызова подпрограммы инициализации обмена (2) передача сообщения происходит одновременно с выполнением процесса (3) не блокируется выполнение всех процессов параллельной программы
16.	<p>Неблокирующий вариант операций передачи сообщений существует для:</p> <ul style="list-style-type: none"> (1) стандартного обмена (2) обмена с буферизацией (3) обмена «по готовности»
17.	<p>Неблокирующий прием сообщений реализован в MPI:</p> <ul style="list-style-type: none"> (1) одной подпрограммой (2) двумя подпрограммами (3) тремя подпрограммами
18.	<p>Неблокирующая передача с буферизацией выполняется подпрограммой:</p> <ul style="list-style-type: none"> (1) MPI_Ibsend (2) MPI_Bsend (3) MPI_Immediate_bsend
19.	<p>Второй этап выполнения неблокирующего обмена это:</p> <ul style="list-style-type: none"> (1) создание буфера обмена (2) проверка выполнения обмена (3) проверка доступности буфера обмена
20.	<p>Подпрограмма MPI_Wait предназначена для:</p> <ul style="list-style-type: none"> (1) блокирующей проверки выполнения обмена (2) неблокирующей проверки выполнения обмена (3) приостановки выполнения программы на заданный период времени
21.	<p>Размер полученного сообщения можно определить с помощью подпрограммы:</p> <ul style="list-style-type: none"> (1) MPI_Get_count (2) MPI_Count (3) MPI_Comm_size
22.	<p>Подпрограмма MPI_Testany выполняет проверку:</p> <ul style="list-style-type: none"> (1) завершения любого из нескольких обменов (2) завершения всех обменов (3) завершения первого обмена
23.	<p>Выберите правильную последовательность действий при создании пользовательского типа:</p> <ul style="list-style-type: none"> (1) регистрация, описание, аннулирование (2) описание, регистрация, аннулирование (3) аннулирование, описание, регистрация
24.	<p>Производный тип данных MPI используется для:</p> <ul style="list-style-type: none"> (1) описания в MPI-программах тех типов, для которых нет аналога в соответствующем языке

	программирования (2) преодоления ограничений в организации пересылаемых данных (3) создания сложных структур данных
25.	Создание структурного типа в MPI выполняется подпрограммой: (1) MPI_Type_struct (2) MPI_Type_commit (3) MPI_Create_struct
26.	Карта типа: (1) позволяет сделать выборку данных базового типа (2) используется для создания производного типа MPI (3) представляет собой набор пар (базовый_тип, смещение)
27.	Аннулирование производного типа выполняется подпрограммой: (1) MPI_Comm_free (2) MPI_Type_null (3) MPI_Type_free
28.	Следующие подпрограммы позволяют сконструировать производный тип из однотипных базовых наборов: (1) MPI_Type_indexed (2) MPI_Type_contiguous (3) MPI_Type_struct
29.	В MPI имеются следующие виртуальные топологии: (1) топология гиперкуба (2) декартова топология (3) топология графа
30.	Подпрограмма MPI_Cart_sub: (1) используется для создания составных виртуальных топологий (2) используется для регистрации новой топологии (3) используется для расщепления коммутатора, наделенного декартовой топологией, на подгруппы, соответствующие декартовым подрешеткам меньшей размерности

3.2 Практические работы

3.2.1 Шифр и наименование компетенции ОПК-7 Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ.

Номер задания	Темы групповых и/или индивидуальных практических работ
1.	Примитивы синхронизации. Атомарные снимки регистров
2.	Алгоритмы синхронизации. Flat-Combining.
3.	Пулы потоков. Модель памяти.
4.	Линеаризуемость. Понятие линеаризуемости. Lock-free стек Trieber. Пример на очередях.
5.	Lock-free очередь. Michael & Scott. Точки линеаризации
6.	Flat-Combining. Схема Flat-Combining. Возможные оптимизации за счёт интерференции операций.
7.	Древовидные Lock-free структуры данных
8.	OpenCL Архитектура вычислений на GPU.

3.3 Кейс-задания

3.3.1 Шифр и наименование компетенции ОПК-7 Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ.

Номер задания	Текст задания
1.	Задача consumer-потока отреагировать на каждое изменение переменной data и набирать сумму полученных значений. После того как достигнуто последнее обновление, функция потока должна вернуть результирующую сумму. Также этот поток должен защититься от попыток потока-interruptor его остановить.

2.	Реализовать классический паттерн producer-consumer с небольшими дополнительными условиями. Программа должна состоять из четырех потоков: один - главный, producer, consumer и interruptor. На стандартный ввод программе подается строка - список чисел произвольной длины.
3.	Функция run_threads должна запускать все три потока, дожидаться их выполнения, и возвращать результат потока-consumer.
4.	Реализовать интерфейс множества в виде lock-free структуры данных. Задание должно состоять из одного файла LockFreeSetImpl.java содержащего класс, реализующий интерфейс LockFreeSet, приложенный к задаче. Спецификация реализации описана в документации к интерфейсу.
5.	Реализовать с использованием графовой модели вычислений следующую задачу, разбив её на чати и обеспечив максимальную потенциальную утилизацию ресурсов (изображение == матрица MxN) - на вход подаются генерированные случайным образом изображения.
6.	Необходимо рассчитать при помощи OpenCL двухмерную свёртку 2 заданных матриц по заданной формуле
7.	Реализовать с использованием графовой модели вычислений следующую задачу, разбив её на чати и обеспечив максимальную потенциальную утилизацию ресурсов (изображение == матрица MxN): по результатам предыдущего шага на изображении выделяются все найденные элементы (например, квадрат с максимальной яркостью вокруг точки)
8.	Задача consumer-потока отреагировать на каждое изменение переменной data и набирать сумму полученных значений. После того как достигнуто последнее обновление, функция потока должна вернуть результирующую сумму. Также этот поток должен защититься от попыток потока-interruptor его остановить.
9.	Реализовать классический паттерн producer-consumer с небольшими дополнительными условиями. Программа должна состоять из четырех потоков: один - главный, producer, consumer и interruptor. На стандартный ввод программе подается строка - список чисел произвольной длины.
10.	Функция run_threads должна запускать все три потока, дожидаться их выполнения, и возвращать результат потока-consumer.

3.4 Экзамен (зачет)

Вопросы (задачи, задания) для зачета

3.4.1 Шифр и наименование компетенции ОПК-7 Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ.

Номер вопроса (задачи, задания)	Текст вопроса (задачи, задания)
1.	Задача producer-потока - получить на вход список чисел, и по очереди использовать каждое значение из этого списка для обновления переменной разделяемой между потоками. После этого поток должен дождаться реакции consumer-потока, и продолжить обновление значений только после того как поток-consumer принял это изменение.
2.	Задача consumer-потока отреагировать на каждое изменение переменной data и набирать сумму полученных значений. После того как достигнуто последнее обновление, функция потока должна вернуть результирующую сумму. Также этот поток должен защититься от попыток потока-interruptor его остановить.
3.	Задача потока-interruptor проста: пока происходит процесс обновления значений, он должен постоянно пытаться остановить поток consumer. Как только поток producer произвел последнее обновление, этот поток завершается.
4.	Функция run_threads должна запускать все три потока, дожидаться их выполнения, и возвращать результат потока-consumer.
5.	Реализовать интерфейс множества в виде lock-free структуры данных. Задание должно состоять из одного файла LockFreeSetImpl.java содержащего класс, реализующий интерфейс LockFreeSet, приложенный к задаче. Спецификация реализации описана в документации к интерфейсу.

6.	<p>Реализовать с использованием графовой модели вычислений следующую задачу, разбив её на части и обеспечив максимальную потенциальную утилизацию ресурсов (изображение == матрица MxN):</p> <ol style="list-style-type: none"> На вход подаются генерированные случайным образом изображения Различные алгоритмы ищут на изображении: <ul style="list-style-type: none"> Набор максимальных элементов (максимальная яркость) Набор минимальных элементов (минимальная яркость) Набор элементов, равных заданному в командной строке значению (0-255)
7.	Необходимо рассчитать при помощи OpenCL двухмерную свёртку 2 заданных матриц по заданной формуле.

Вопросы (задачи, задания) для экзамена

3.4.2 Шифр и наименование компетенции ОПК-7 Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ.

Номер вопроса (задачи, задания)	Текст вопроса (задачи, задания)
8.	Примитивы синхронизации
9.	Алгоритмы синхронизации
10.	Атомарные снимки регистров
11.	Ошибки параллельного программирования
12.	Пулы потоков
13.	Шаблоны параллельного программирования
14.	Консенсус
15.	Транзакционная память
16.	Асинхронный ввод/вывод
17.	Модель памяти
18.	Линеаризуемость

4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Процедуры оценивания в ходе изучения дисциплины знаний, умений и навыков, характеризующих этапы формирования компетенций, регламентируются положениями:

- П ВГУИТ 2.4.03 Положение о курсовых экзаменах и зачетах;
- П ВГУИТ 4.1.02 Положение о рейтинговой оценке текущей успеваемости.

5. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания для каждого результата обучения по дисциплине/практике

Результаты обучения по этапам формирования компетенций	Предмет оценки (продукт или процесс)	Показатель оценивания	Критерии оценивания сформированности компетенций	Шкала оценивания	
				Академическая оценка или баллы	Уровень освоения компетенции
<u>Шифр и наименование компетенции</u> <u>ОПК-7 Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ.</u>					
ЗНАТЬ: основные подходы к разработке параллельных программ; инструментальные средства параллельного программирования; методы создания параллельных программ для задач многопоточного программирования в профессиональной деятельности.	Собеседование (зачет, экзамен)	Уровень владения материалом	Обучающийся грамотно решил кейс-задание, ответил на все вопросы, допустил не более 5 ошибок в ответах на вопросы	Зачтено	Освоена (повышенный, базовый)
			Обучающийся не предложил вариантов решения кейс-задания, в ответе допустил более пяти ошибок	Не зачтено /Неудовлетворительно	Не освоена (недостаточный)
	Тест (зачет, экзамен)	Результат тестирования	0-59,99% правильных ответов - неудовлетворительно;	Не зачтено /Неудовлетворительно	Не освоена (недостаточный)
			60-74,99% - удовлетворительно;	Зачтено, удовлетворительно	Освоена (повышенный, базовый)
			75- 84,99% -хорошо;	Зачтено, хорошо	Освоена (повышенный, базовый)
		85-100% - отлично.	Зачтено, отлично	Освоена (повышенный, базовый)	
УМЕТЬ: создавать параллельные программы для вычислительных систем с распределенной, общей оперативной памятью; проводить распараллеливание	Собеседование (защита практической работы)	Уровень владения материалом техники.	Обучающийся активно участвовал в выполнении работы, допустил не более 5 ошибок в ответах на вопросы при защите работы	Зачтено	Освоена (повышенный, базовый)
			Обучающийся не выполнил и не защитил работу	Не зачтено / Неудовлетворительно	Не освоена (недостаточный)

<p>вычислительных алгоритмов; строить модель выполнения параллельных программ; оценивать эффективности параллельных вычислений; анализировать сложность вычислений и возможность распараллеливания разрабатываемых алгоритмов; применять общие схемы разработки параллельных программ и систем.</p>			Обучающийся не разобрался в сложившейся ситуации, не выявил причины случившегося и не предложил вариантов решения	Не зачтено / Неудовлетворительно	Не освоена (недостаточный)
	Задача	Содержание решения	Обучающийся грамотно и без ошибок решил задачу	Зачтено/Отлично	Освоена (повышенный)
			Обучающийся правильно решил задачу, но в вычислениях допустил ошибки	Зачтено/Хорошо	Освоена (Базовый)
			Обучающийся не предложил вариантов решения задачи	Не зачтено / Неудовлетворительно	Не освоена (недостаточный)
<p>ВЛАДЕТЬ: навыками параллельного программирования для решения профессиональных задач, с помощью современных инструментальных средств; навыками создания параллельных программ для вычислительных систем с распределенной, общей оперативной памятью.</p>	Тест	Результат тестирования	0-59,99% правильных ответов - неудовлетворительно;	Не зачтено / Неудовлетворительно	Не освоена (недостаточный)
			60-74,99% - удовлетворительно;	Зачтено, удовлетворительно	Освоена (повышенный, базовый)
			75- 84,99% -хорошо;	Зачтено, хорошо	Освоена (повышенный, базовый)
			85-100% - отлично.	Зачтено, отлично	Освоена (повышенный, базовый)
	Собеседование (Зачет, экзамен)	Владение навыками параллельного	Обучающийся грамотно решил кейс-задание, ответил на все вопросы, допустил не более 5 ошибок в ответах на вопросы	Зачтено	Освоена (повышенный, базовый)

		программирования для решения профессиональных задач, с помощью современных инструментальных средств; навыками создания параллельных программ для вычислительных систем с распределенной, общей ОП.	Обучающийся не предложил вариантов решения кейс-задания, в ответе допустил более пяти ошибок	Не зачтено /Неудовлетворительно	Не освоена (недостаточный)
--	--	--	--	---------------------------------	----------------------------