

**Федеральное государственное образовательное бюджетное учреждение
высшего образования
«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ
РОССИЙСКОЙ ФЕДЕРАЦИИ»
(Финансовый университет)**

Тульский филиал Финуниверситета

Кафедра «Математика и информатика»

СОГЛАСОВАНО

ЗАО «ЛИМ»

Генеральный директор

В.В. Куликов

«19» декабря 2024 г.



УТВЕРЖДАЮ

Директор филиала

Г.В. Кузнецов

«24» декабря 2024 г.



Н.О. Козлова, Манохин Е.В.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ПО ДИСЦИПЛИНЕ
ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ**

для студентов, обучающихся по направлению подготовки
09.03.02 Информационные системы и технологии,
образовательная программа «Информационные технологии в экономике»,
профиль «Информационные технологии в управлении цифровой
экономикой»

Рекомендовано Ученым советом Тульского филиала Финуниверситета
(протокол от 24 декабря 2024 г. № 22)

Одобрено кафедрой «Математика и информатика»
(протокол от 02 декабря 2024 г. № 5)

Тула 2024

«Параллельное программирование»

КОД И НАИМЕНОВАНИЕ ОПОП:

09.03.02 Информационные системы и технологии

Образовательная программа - Информационные технологии в экономике

Профиль «Информационные технологии в управлении цифровой экономикой»

Общее количество заданий по дисциплине

Код компетенции	Наименование компетенции	Количество заданий
ПКП-5	Способен разрабатывать программные продукты в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению программных систем	20
Всего:		20

Количество заданий по дисциплине в одном варианте

Код компетенции	Наименование компетенции	Количество заданий
ПКП-5	Способен разрабатывать программные продукты в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению программных систем	20
Всего:		20

Типы заданий

Наименование типа заданий	Номер типа заданий
Задание комбинированного типа с выбором одного верного ответа из предложенных	1
Задание открытого типа	2
Задание открытого типа с развернутым ответом	3

Распределение заданий по компетенциям

Код компетенции	Наименование компетенции	Наименование индикаторов сформированности компетенции	Семестр	Номер задания
ПКП-5	Способен разрабатывать программные продукты в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению программных систем	1. Разрабатывает архитектуру и дизайн программной системы на основе анализа требований к программной системе в целом.	6	1.1, 1.2, 2.1, 2.2, 3.1, 3.2, 4.1, 4.2, 5.1, 5.2, 6.1, 6.2, 7.1, 7.2, 8.1, 8.2, 9.1, 9.2, 10.1, 10.2
		2. Проектирует пользовательское взаимодействие с программной системой, включая пользовательский интерфейс, на основе аналитики пользовательского опыта.	6	11.1, 11.2, 12.1, 12.2, 13.1, 13.2, 14.1, 14.2, 15.1, 15.2, 16.1, 16.2, 17.1, 17.2, 18.1, 18.2, 19.1, 19.2, 20.1, 20.2

Типы, уровень сложности и время выполнения заданий

Код компетенции	Индикатор сформированности компетенции	Номер задания	Тип задания	Уровень сложности задания	Время выполнения задания (мин.)
ПКП-5	Разрабатывает архитектуру и дизайн программной системы на основе анализа требований к программной системе в целом.	1.1, 1.2.	1	базовый	2
	Разрабатывает архитектуру и дизайн программной системы на основе анализа требований к программной системе в целом.	2.1, 2.2	1	базовый	2
	Разрабатывает архитектуру и дизайн программной системы на основе анализа требований к программной системе в целом.	3.1, 3.2	1	базовый	2
	Разрабатывает архитектуру и дизайн программной системы на основе анализа требований к программной системе в целом.	4.1, 4.2	1	базовый	2
	Разрабатывает архитектуру и дизайн программной системы на основе анализа требований к программной системе в целом.	5.1, 5.2	1	базовый	2
	Разрабатывает архитектуру и дизайн программной системы на основе анализа требований к программной системе в целом.	6.1, 6.2.	2	повышенный	4
	Разрабатывает архитектуру и дизайн программной системы на основе анализа требований к программной системе в целом.	7.1, 7.2	2	повышенный	4
	Разрабатывает архитектуру и дизайн программной системы на основе анализа требований к программной системе в целом.	8.1, 8.2	2	повышенный	4
	Разрабатывает архитектуру и дизайн программной системы на основе анализа требований к программной системе в целом.	9.1, 9.2	3	высокий	8
	Разрабатывает архитектуру и дизайн программной системы на основе анализа требований к программной системе в целом.	10.1, 10.2	3	высокий	8
	Проектирует пользовательское взаимодействие с программной системой, включая пользовательский интерфейс, на основе аналитики пользовательского опыта.	11.1, 11.2.	1	базовый	2
	Проектирует пользовательское взаимодействие с программной системой, включая пользовательский интерфейс, на основе аналитики пользовательского опыта.	12.1, 12.2	1	базовый	2
	Проектирует пользовательское взаимодействие с программной	13.1, 13.2	1	базовый	2

	системой, включая пользовательский интерфейс, на основе аналитики пользовательского опыта.				
	Проектирует пользовательское взаимодействие с программной системой, включая пользовательский интерфейс, на основе аналитики пользовательского опыта.	14.1, 14.2	1	базовый	2
	Проектирует пользовательское взаимодействие с программной системой, включая пользовательский интерфейс, на основе аналитики пользовательского опыта.	15.1, 15.2	1	базовый	2
	Проектирует пользовательское взаимодействие с программной системой, включая пользовательский интерфейс, на основе аналитики пользовательского опыта.	16.1, 16.2.	2	повышенный	4
	Проектирует пользовательское взаимодействие с программной системой, включая пользовательский интерфейс, на основе аналитики пользовательского опыта.	17.1, 17.2	2	повышенный	4
	Проектирует пользовательское взаимодействие с программной системой, включая пользовательский интерфейс, на основе аналитики пользовательского опыта.	18.1, 18.2	3	высокий	8
	Проектирует пользовательское взаимодействие с программной системой, включая пользовательский интерфейс, на основе аналитики пользовательского опыта.	19.1, 19.2	3	высокий	8
	Проектирует пользовательское взаимодействие с программной системой, включая пользовательский интерфейс, на основе аналитики пользовательского опыта.	20.1, 20.2	3	высокий	8

Задания, позволяющие осуществлять оценку компетенций, установленных для дисциплины

Вариант 1

Задание 1.1.

Какая цель параллельного программирования?

- А) Уменьшить объем памяти
- Б) Увеличить скорость выполнения программ
- В) Уменьшить объем кода

Ответ:

Задание 2.1.

Что такое мьютекс?

- А) Механизм для управления доступом потоков к разделяемой памяти
- Б) Устройство для ускорения процессора
- В) Библиотека для параллельных вычислений

Ответ:

Задание 3.1.

Какая библиотека наиболее популярна для параллельных вычислений на GPU?

- А) OpenMP
- Б) MPI
- В) CUDA

Ответ:

Задание 4.1.

Какое из следующих утверждений является верным?

- А) Потокам всегда выделяется отдельный адрес памяти
- Б) Потоки в процессе могут разделять одну и ту же область памяти
- В) Потоки могут работать только на одном процессоре

Ответ:

Задание 5.1.

Какую проблему может вызвать использование кэш-памяти в параллельном программировании?

- А) Задержка в выполнении операций
- Б) Потеря данных
- В) Проблема когерентности кэша

Ответ:

Задание 6.1.

Укажите не менее трех основных моделей параллельного программирования.

Ответ:

Задание 7.1.

Перечислите методы синхронизации потоков?

Ответ:

Задание 8.1.

Какой вид программирования позволяет управлять задачами, которые могут выполняться без блокировки основного потока, не обязательно параллельно.

Ответ:

Задание 9.1.

Поясните различия между параллельным и конкурентным программированием?

Ответ:

Задание 10.1.

Объясните, что такое гонка данных (data race) и почему это важно.

Ответ:

Задание 11.1.

Что такое параллельное программирование?

- А) Метод выполнения одного задания многими потоками одновременно
- Б) Метод использования одного процессора для выполнения нескольких задач последовательно
- В) Метод уменьшения времени выполнения программы путем увеличения числа процессоров
- Г) Метод оптимизации памяти программы

Ответ:

Задание 12.1.

Какой из следующих методов синхронизации потоков используется для предотвращения одновременного доступа к общим ресурсам?

- А) Кэш
- Б) Лок
- В) Декларация
- Г) Компрессия

Ответ:

Задание 13.1.

Что такое гонка данных (data race) в контексте параллельного программирования?

- А) Когда два потока одновременно читают и записывают в один и тот же ресурс
- Б) Когда два потока выполняют разные операции на разных ресурсах
- В) Когда потоки синхронизированы и выполняются последовательно
- Г) Когда один поток выполняет задачи в то время, как другой ожидал

Ответ:

Задание 14.1.

Какой из следующих инструментов используется для контроля доступа к ресурсам в многопоточном приложении?

- А) Стек
- Б) Кэш-память
- В) Мьютекс +
- Г) Очередь

Ответ:

Задание 15.1.

Какой из следующих подходов используется для достижения параллелизма в многопоточном программировании?

- А) Разделение задач на подзадачи и выполнение их одновременно +

- Б) Уменьшение количества операций ввода-вывода
- В) Уменьшение размера программы
- Г) Использование статической памяти

Ответ:

Задание 16.1.

За счет использования каких систем, происходит ускорение выполнения программ и повышение производительности при параллельном программировании?

Ответ:

Задание 17.1.

Как называются операции, которые выполняются без прерываний?

Ответ:

Задание 18.1.

Каковы основные проблемы, возникающие при параллельном программировании? Укажите не менее трех проблем.

Ответ:

Задание 19.1.

Что такое барьер синхронизации и как он используется в параллельном программировании?

Ответ:

Задание 20.1.

Как параллельное программирование может повлиять на потребление ресурсов, таких как память и процессорное время?

Ответ:

Вариант 2

Задание 1.2.

Какой механизм используется для управления доступом потоков к разделяемой памяти?

- А) Кэширование
- Б) Синхронизация
- В) Многозадачность

Ответ:

Задание 2.2.

Как называется ошибка, возникающая из-за одновременного доступа нескольких потоков к общим данным?

- А) Ошибка компиляции
- Б) Гонка данных
- В) Поточковая блокировка

Ответ:

Задание 3.2.

Что такое OpenMP?

- А) Интерфейс для написания многопоточных программ на GPU
- Б) Среда для отладки программ
- В) API для параллельного программирования на CPU

Ответ:

Задание 4.2.

Как называется модель параллельного программирования, в которой данные передаются между процессами через сообщения?

- А) Модель передачи сообщений
- Б) Модель потоков
- В) Модель разделяемой памяти

Ответ:

Задание 5.2.

Какой из ниже перечисленных терминов относится к методу синхронизации потоков?

- А) Массив
- Б) Семафор
- В) Декодер

Ответ:

Задание 6.2.

Назовите не менее трех примеров популярных библиотек для параллельного программирования.

Ответ:

Задание 7.2.

Метод разработки программного обеспечения, при котором задачи выполняются одновременно с использованием нескольких процессоров или ядер – это ...

Ответ:

Задание 8.2.

Коллекция потоков, которые повторно используются для выполнения задач – это ...

Ответ:

Задание 9.2.

Что такое поток выполнения (thread) и чем он отличается от процесса?

Ответ:

Задание 10.2.

Какой подход к параллельному программированию предпочтительнее для задач с высоким уровнем взаимозависимости между потоками?

Ответ:

Задание 11.2.

Какой из следующих терминов относится к управлению выполнением потоков?

- А) Синхронизация
- Б) Локализация
- В) Аллокация
- Г) Интерпретация

Ответ:

Задание 12.2.

Какой механизм обеспечивает корректное завершение выполнения потока в многопоточном приложении?

- А) Ключевые слова
- Б) Прерывания
- В) Ожидания
- Г) Запросы

Ответ:

Задание 13.2.

Что такое "мёртвый блок" (deadlock) в многопоточном программировании?

- А) Состояние, при котором потоки могут продолжить выполнение
- Б) Состояние, при котором потоки блокируют друг друга и не могут продолжить выполнение
- В) Состояние, при котором потоки успешно синхронизируются
- Г) Состояние, при котором поток завершает выполнение

Ответ:

Задание 14.2.

Что такое "параллелизм" в контексте параллельного программирования?

- А) Процесс выполнения нескольких задач одновременно для улучшения производительности +
- Б) Процесс выполнения одной задачи последовательно
- В) Процесс оптимизации использования памяти
- Г) Процесс разделения программы на отдельные модули

Ответ:

Задание 15.2.

Что такое "параллельное программирование" в контексте архитектуры многопроцессорных систем?

- А) Процесс использования нескольких процессоров для выполнения одной задачи одновременно
- Б) Процесс увеличения скорости одного процессора
- В) Процесс оптимизации загрузки одного процессора
- Г) Процесс последовательного выполнения инструкций на одном процессоре

Ответ:

Задание 16.2.

Какая библиотека наиболее популярна для параллельных вычислений на GPU?

Ответ:

Задание 17.2.

Какую проблему может вызвать использование кэш-памяти в параллельном программировании?

Ответ:

Задание 18.2.

Что такое мьютекс и как он используется для синхронизации потоков?

Ответ:

Задание 19.2.

Как можно избежать проблемы мертвых блокировок в параллельном программировании?

Ответ:

Задание 20.2.

Как определить, когда целесообразно использовать параллельное программирование в проекте?

Ответ:

Ключи к оцениванию заданий

Вариант 1

№ задания	Верный ответ	Критерии оценивания
1.1	Б	1 балл – полное совпадение с верным ответом 0 баллов – неверный ответ или его отсутствие
2.1	А	1 балл – полное совпадение с верным ответом 0 баллов – неверный ответ или его отсутствие
3.1	В	1 балл – полное совпадение с верным ответом 0 баллов – неверный ответ или его отсутствие
4.1	Б	1 балл – полное совпадение с верным ответом 0 баллов – неверный ответ или его отсутствие
5.1	В	1 балл – полное совпадение с верным ответом 0 баллов – неверный ответ или его отсутствие
6.1	Потоковая модель, модель разделяемой памяти, модель передачи сообщений и модель на основе задач (task-based).	2 балла – полное совпадение с верным ответом 1 балл – правильный ответ, без обоснования 0 баллов – неверный ответ или его отсутствие
7.1	Мьютексы, семафоры, барьеры, критические секции и атомарные операции.	2 балла – полное совпадение с верным ответом 1 балл – правильный ответ, без обоснования 0 баллов – неверный ответ или его отсутствие
8.1	Асинхронное программирование	2 балла – полное совпадение с верным ответом 1 балл – правильный ответ, без обоснования 0 баллов – неверный ответ или его отсутствие
9.1	Параллельное программирование предполагает одновременное выполнение задач на нескольких процессорах, тогда как конкурентное программирование - это управление выполнением задач так, чтобы они казались выполняемыми одновременно, даже если они работают на одном процессоре.	3 балла – полное совпадение с верным ответом 1 балл – правильный ответ, без обоснования 0 баллов – неверный ответ или его отсутствие
10.1	Гонка данных - это ситуация, когда несколько потоков обращаются к одним и тем же данным одновременно, и хотя бы один поток изменяет эти данные, что может привести к некорректным результатам. Очень важно избегать гонок данных для корректной работы программ.	3 балла – полное совпадение с верным ответом 1 балл – правильный ответ, без обоснования 0 баллов – неверный ответ или его отсутствие
11.1	А	1 балл – полное совпадение с верным ответом 0 баллов – неверный ответ или его отсутствие
12.1	Б	1 балл – полное совпадение с верным ответом 0 баллов – неверный ответ или его отсутствие
13.1	А	1 балл – полное совпадение с верным ответом 0 баллов – неверный ответ или его отсутствие
14.1	В	1 балл – полное совпадение с верным ответом 0 баллов – неверный ответ или его отсутствие
15.1	А	1 балл – полное совпадение с верным ответом 0 баллов – неверный ответ или его отсутствие
16.1	За счет использования многопроцессорных и многоядерных систем	2 балла – полное совпадение с верным ответом 1 балл – правильный ответ, без обоснования 0 баллов – неверный ответ или его отсутствие
17.1	Атомарные операции	2 балла – полное совпадение с верным ответом 1 балл – правильный ответ, без обоснования 0 баллов – неверный ответ или его отсутствие
18.1	Основные проблемы включают гонки данных, мертвые блокировки (deadlocks), неправильное распределение задач и накладные расходы на синхронизацию. Эти	3 балла – полное совпадение с верным ответом 1 балл – правильный ответ, без обоснования 0 баллов – неверный ответ или его отсутствие

	проблемы могут приводить к ошибкам и снижению производительности программ.	
19.1	Барьер синхронизации - это примитив, который заставляет несколько потоков ожидать, пока все они не достигнут определенной точки выполнения. Это позволяет обеспечить, что все потоки синхронизированы на этом этапе, что полезно для координации и предотвращения несогласованности данных.	3 балла – полное совпадение с верным ответом 1 балл – правильный ответ, без обоснования 0 баллов – неверный ответ или его отсутствие
20.1	Параллельное программирование может увеличить потребление ресурсов, так как каждый поток или процесс требует выделенной памяти и процессорного времени. Эффективное использование ресурсов требует грамотного распределения задач и управления потоками для минимизации накладных расходов и избегания избыточного потребления.	3 балла – полное совпадение с верным ответом 1 балл – правильный ответ, без обоснования 0 баллов – неверный ответ или его отсутствие

Вариант 2

№ задания	Верный ответ	Критерии оценивания
1.2	Б	1 балл – полное совпадение с верным ответом 0 баллов – неверный ответ или его отсутствие
2.2	Б	1 балл – полное совпадение с верным ответом 0 баллов – неверный ответ или его отсутствие
3.2	В	1 балл – полное совпадение с верным ответом 0 баллов – неверный ответ или его отсутствие
4.2	А	1 балл – полное совпадение с верным ответом 0 баллов – неверный ответ или его отсутствие
5.2	Б	1 балл – полное совпадение с верным ответом 0 баллов – неверный ответ или его отсутствие
6.2	OpenMP, MPI, POSIX Threads (pthread), CUDA для GPU	2 балла – полное совпадение с верным ответом 1 балл – правильный ответ, без обоснования 0 баллов – неверный ответ или его отсутствие
7.2	Параллельное программирование	2 балла – полное совпадение с верным ответом 1 балл – правильный ответ, без обоснования 0 баллов – неверный ответ или его отсутствие
8.2	Пул потоков	2 балла – полное совпадение с верным ответом 1 балл – правильный ответ, без обоснования 0 баллов – неверный ответ или его отсутствие
9.2	Поток - это наименьшая единица выполнения в процессе. Процессы могут содержать несколько потоков, разделяющих память процесса. В отличие от процессов, потоки легче создаются и переключаются, но требуют синхронизации при доступе к разделяемым данным.	3 балла – полное совпадение с верным ответом 1 балл – правильный ответ, без обоснования 0 баллов – неверный ответ или его отсутствие
10.2	Для задач с высоким уровнем взаимозависимости между потоками лучше использовать подходы, основанные на разделяемой памяти и механизмах синхронизации, таких как мьютексы и условные переменные. Эти подходы помогают координировать выполнение потоков и управлять доступом к разделяемым ресурсам.	3 балла – полное совпадение с верным ответом 1 балл – правильный ответ, без обоснования 0 баллов – неверный ответ или его отсутствие
11.2	А	1 балл – полное совпадение с верным ответом

		0 баллов – неверный ответ или его отсутствие
12.2	Б	1 балл – полное совпадение с верным ответом 0 баллов – неверный ответ или его отсутствие
13.2	Б	1 балл – полное совпадение с верным ответом 0 баллов – неверный ответ или его отсутствие
14.2	А	1 балл – полное совпадение с верным ответом 0 баллов – неверный ответ или его отсутствие
15.2	А	1 балл – полное совпадение с верным ответом 0 баллов – неверный ответ или его отсутствие
16.2	CUDA	2 балла – полное совпадение с верным ответом 1 балл – правильный ответ, без обоснования 0 баллов – неверный ответ или его отсутствие
17.2	Проблема когерентности кэша.	2 балла – полное совпадение с верным ответом 1 балл – правильный ответ, без обоснования 0 баллов – неверный ответ или его отсутствие
18.2	Мьютекс (взаимное исключение) — это механизм синхронизации, который обеспечивает эксклюзивный доступ к ресурсам. Он используется для предотвращения одновременного доступа нескольких потоков к критическим секциям кода или общим данным, чтобы избежать гонок данных и других проблем.	3 балла – полное совпадение с верным ответом 1 балл – правильный ответ, без обоснования 0 баллов – неверный ответ или его отсутствие
19.2	Можно избежать мертвых блокировок, применяя методы управления взаимоблокировками, такие как иерархия блокировок, таймауты и алгоритмы предотвращения взаимоблокировок. Также важно тщательно проектировать систему синхронизации и избегать сложных зависимостей между ресурсами.	3 балла – полное совпадение с верным ответом 1 балл – правильный ответ, без обоснования 0 баллов – неверный ответ или его отсутствие
20.2	Параллельное программирование целесообразно использовать, когда задачи можно эффективно разделить на независимые подзадачи, которые могут выполняться одновременно. Это особенно полезно при обработке больших объемов данных, выполнении вычислительных задач и работе с ресурсоемкими приложениями.	3 балла – полное совпадение с верным ответом 1 балл – правильный ответ, без обоснования 0 баллов – неверный ответ или его отсутствие