

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Гнатюк Максим Александрович  
Должность: Первый проректор  
Дата подписания: 11.07.2022 09:51:21  
Уникальный программный ключ:  
8873f497f100e798ae8c92c0d38e105c818d5410

Приложение  
к рабочей программе дисциплины

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ  
ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

**Параллельные вычисления**

---

*(наименование дисциплины(модуля))*

Направление подготовки / специальность

**09.03.01 Информатика и вычислительная техника**

---

*(код и наименование)*

Направленность (профиль)/специализация

Проектирование АСОИУ на транспорте

*(наименование)*

## Содержание

1. Пояснительная записка.
2. Типовые контрольные задания или иные материалы для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций.
3. Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации.

## 1. Пояснительная записка

Цель промежуточной аттестации – оценивание промежуточных и окончательных результатов обучения по дисциплине, обеспечивающих достижение планируемых результатов освоения образовательной программы.

Формы промежуточной аттестации: **зачет с оценкой в 4 семестре**

Перечень компетенций, формируемых в процессе освоения дисциплины

Код и наименование компетенции	Код достижения индикатора компетенции
ПК-1 Способен разрабатывать требования и проектировать программное обеспечение	ПК-1.3 Разрабатывать программный код на языках программирования высокого уровня
	ПК-1.4 Осуществлять отладку программ, написанных на языке высокого уровня

Результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

Код и наименование индикатора достижения компетенции	Результаты обучения по дисциплине	Оценочные материалы (семестр)
ПК-1.3 Разрабатывать программный код на языках программирования высокого уровня	Обучающийся знает: разновидности архитектур целевых аппаратных и программных платформ параллельных ЭВМ, для которых разрабатывается программный код на языках высокого уровня	Вопросы № (1–52)
	Обучающийся умеет: разрабатывать программный код на языках параллельного программирования высокого уровня	Задания №(1- 5)
	Обучающийся владеет: навыками использования инструментальных сред для разработки параллельных программ на языках высокого уровня	Задания №(11- 15)
ПК-1.4 Осуществлять отладку программ, написанных на языке высокого уровня	Обучающийся знает: инструментальные среды для отладки параллельных программ, написанных на языках высокого уровня	Вопросы № (53–103)
	Обучающийся умеет: отлаживать параллельные программы, написанные на языках программирования высокого уровня	Задания № (6-10)
	Обучающийся владеет: навыками использования инструментальных сред для отладки параллельных программ, написанных на языках высокого уровня	Задания №(16- 20)

### 4 семестр

Промежуточная аттестация (зачет с оценкой) проводится в одной из следующих форм:

- 1) Собеседование
- 2) выполнение заданий в ЭИОС СамГУПС.

## 2. Типовые<sup>1</sup> контрольные задания или иные материалы для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций

<sup>1</sup> Приводятся типовые вопросы и задания. Оценочные средства, предназначенные для проведения аттестационного мероприятия, хранятся на кафедре в достаточном для проведения оценочных процедур количестве вариантов. Оценочные средства подлежат актуализации с учетом развития науки, образования, культуры, экономики, техники, технологий и

## 2.1 Типовые вопросы (тестовые задания) для оценки знаниевого образовательного результата

Проверяемый образовательный результат:

Код и наименование индикатора достижения компетенции	Образовательный результат
ПК-1.3 Разрабатывать программный код на языках программирования высокого уровня	Обучающийся знает: разновидности архитектур целевых аппаратных и программных платформ ЭВМ, для которой разрабатывается программный код на языках низкого уровня
<p>Примеры вопросов</p> <p>1. Когда была создана первая супер ЭВМ?</p> <p>1.1 в середине 70-х 1.2 в середине 60-х 1.3 в начале 80-х 1.4 в начале 80-х 1.5 в конце 70-х</p> <p>2. Кем была разработана первая супер-ЭВМ?</p> <p>2.1 Джоном фон Нейманом 2.2 Сеймуром Крэем 2.3 Томасом Стерлингом 2.4 Доном Беккером 2.5 Биллом Гейтсом</p> <p>3. Укажите неправильное утверждение. SISD - это обычные последовательные компьютеры SIMD - большинство современных ЭВМ относятся к этой категории MISD - вычислительных машин такого класса мало MIMD - это реализация нескольких потоков команд и потоков данных</p> <p>4. Для конвейерной обработки присуще: загрузка операндов в векторные регистры операций с матрицами выделение отдельных этапов выполнения общей операции сложение 2-х операндов одновременным сложением всех их двоичных разрядов</p> <p>5. Приоритет - это... описание алгоритма на некотором формализованном языке число, приписанное ОС каждому процессу или задаче отдельный этап выполнения общей операции оповещение со стороны ОС о той или иной форме взаимодействия</p> <p>6. Стек - это... "память", в адресном пространстве которой работает процесс тот или иной способ передачи инструкции из одного процесса в другой область памяти для локальных переменных, аргументов и возвращаемых функциями значений организация доступа 2х (или более) процессов к одному и тому же блоку памяти</p> <p>7. Кластер (в контексте параллельного программирования)- это... область оперативной памяти управляющее устройство, выполненное на одном или более кристаллах 2 или более узлов, соединенных при помощи локальной сети раздел жесткого диска суперкомпьютер для выполнения особых задач</p> <p>8. Выберите шаг(и), не присущий(е) для цикла выполнения команды: запись результата в память выборка команды кэширование следующей команды выполнение команды декодирование команды, вычисление адреса операнда и его выборка</p>	

социальной сферы. Ответственность за нераспространение содержания оценочных средств среди обучающихся университета несут заведующий кафедрой и преподаватель – разработчик оценочных средств.

обращение к памяти

9. Конвейерная технология предполагает ...  
последовательную обработку команд  
обработку команд, удовлетворяющих определенным критериям  
обработку несколько команд одновременно  
общий доступ команд к памяти

10 Система, главной особенностью является наличие общей физической памяти, разделяемой всеми процессорами называется ...

NUMA  
SMP  
MPP  
PVP

11. Главная особенность архитектуры NUMA?  
неоднородный доступ к памяти  
сверхвысокая производительность  
наличие векторно-конвейерных процессоров  
наличие общей физической памяти, разделяемой всеми процессорами

12. Вычислительные машины с какой архитектурой наиболее дешевы?  
симметричная многопроцессорная обработка  
параллельная архитектура с векторными процессорами  
кластерные системы  
массивно-параллельная архитектура

13. Пиковая производительность системы измеряется в:  
Мегагерц  
MIPS  
MFlops  
MByte

14. Пиковая производительность системы определяется:  
временем выполнения реальных задач  
произведением производительности 1-го процессора на число процессоров в системе  
временем выполнения тестовых задач  
количеством переданной информации

15. Производительность многопроцессорной вычислительной системы характеризуется:  
количеством операций, производимых за единицу времени  
количеством байт информации, переданных в единицу времени  
числом импульсов, генерируемых в единицу времени  
объемом располагаемой для вычислений памяти

16. Какое понятие характеризует возрастание сложности соединений при добавлении в конфигурацию новых узлов.

масштабируемость  
ускорение  
эффективность  
пиковая производительность

17. Коммуникационным ... сети именуется максимальный путь между любыми двумя узлами. Впишите недостающее слово  
(диаметром)

18. Найдите неверное утверждение.  
По способу взаимодействия процессоров с оперативной памятью архитектуры бывают:  
с распределенно-разделяемой памятью  
с разделяемой памятью  
с распределенной памятью  
с когерентной кэш-памятью

19. Укажите наиболее быструю организацию сети для кластера.  
Gigabit Ethernet  
Myrinet  
Infinyband  
Ethernet

20. Параллельная программа – это...  
программа, работающая одновременно на нескольких компьютерах  
программа, обрабатывающая большой объем данных  
программа, осуществляющая обмен сообщениями в сети  
программа, содержащая несколько процессов, работающих совместно

21. Асинхронная модель параллельных вычислений имеет следующие особенности:  
все процессы выполняют одни и те же действия с собственными данными  
различные процессы решают разные задачи  
все процессы используют общую память  
все процессы выполняются в своих критических секциях

22. Синхронная модель параллельных вычислений имеет следующие особенности:  
все процессы выполняют одни и те же действия с собственными данными  
различные процессы решают разные задачи  
все процессы используют общую память  
все процессы выполняются в своих критических секциях

23. Две операции называются независимыми если  
множество чтения одной не пересекается с множеством чтения другой  
множество чтения одной не пересекается с множеством записи другой  
множество чтения одной пересекается с множеством записи другой  
множество чтения одной пересекается с множеством чтения другой

24. Какие операции могут выполняться параллельно?  
независимые  
зависимые  
элементарные  
неделимые

25. Какой процесс называется потребителем?  
Процесс, передающий данные  
Процесс, получающий данные  
Процесс, вводящий данные  
Процесс, выводящий данные

26. Какой процесс называется производителем?  
Процесс, передающий данные  
Процесс, получающий данные  
Процесс, вводящий данные  
Процесс, выводящий данные

27. Какие технологии повышения производительности применяются в современных процессорах?  
Суперскалярность (30%)  
Многопоточность  
Конвейеризация (30%)  
Векторная обработка данных (40%)

28. Что такое конвейеризация?  
Исполнение нескольких команд одновременно  
Параллельное выполнение различных частей команд  
Сохранение данных в сверхбыстрой памяти  
Обработка данных большого размера

29. Что такое суперскалярность?  
Исполнение нескольких команд одновременно  
Параллельное выполнение различных частей команд  
Сохранение данных в сверхбыстрой памяти  
Обработка данных большого размера

31. Какие виды оптимизации применяются при конвейеризации?  
Предсказание переходов (30%)  
Замена команд  
Перестановка команд (30%)  
Переименование регистров (40%)

32. Какие системы относятся к технологии SIMD?

Векторные процессоры (50%)  
Матричные процессоры (50%)  
Кластеры  
SMP

33. Какие системы относятся к технологии MIMD?

Векторные процессоры  
Матричные процессоры  
Кластеры (50%)  
Симметричные многопроцессорные (50%)

34. К какому классу относятся многоядерные системы?

Матричные процессоры  
Распределенные системы  
Кластеры  
Системы с общей памятью

35. К какому классу относятся кластерные системы?

Матричные процессоры  
Распределенные системы  
Симметричные мультипроцессоры  
Системы с общей памятью

36. Процесс - это...

сетевой интерфейс контроллера блочных передач  
это число, приписанное операционной системой каждой задаче  
это динамическая сущность программы, ее код в процессе своего выполнения  
система, выполняющая повторяющуюся операцию

37. Ресурс - это...

объект, необходимый для работы процессу или задаче  
сообщение, доставляемое процессу посредством ОС  
процесс превращения скомпилированного кода в программу  
число, приписанное ОС каждому процессу и задаче

38. Выберите верное утверждение.

Активные ресурсы...  
используют взаимные исключения  
могут быть использованы одновременно несколькими процессами  
способны изменять информацию в памяти  
используются только одним процессором, пока тот не завершит работу с ресурсом

39. Функцией мьютекса является:

регистрация обработчика сообщения в операционной системе  
распределение квантов времени в системе между выполняющимися процессами  
способ синхронизации параллельных процессов через разделяемый критический ресурс  
способ обмена данными процессорами через разделяемую память или коммутируемый канал

40. Процесс имеет:

собственное состояние  
собственный процессор  
собственную систему  
собственный семафор

41. Барьер - это...

подпрограмма, определяющая факт прихода сообщения  
место в программе, где процесс ожидает подхода к нему остальных процессов  
блокировка процесса до тех пор, пока все операции обмена не будут завершены  
ожидание завершения асинхронных процедур, ассоциированных с идентификатором

42. Семафор - это ...

аппаратный коммутатор  
устройство синхронизации для параллельных ЭВМ  
программный механизм синхронизации в виде переменной в общей памяти

43. Какие сущности имеют общую память?

Два процесса  
Два потока

Поток и процесс  
Вычислительные узлы кластера

44. Что имеет собственную память для данных?

Процесс  
Поток  
И процесс, и поток

45. Как организуется взаимодействие процессов?

Через общую память  
Через обмен сообщениями (50%)  
Через файловую систему (50%)  
Через регистры процессора

46. Какие общие ресурсы есть у потоков?

Память для данных (50%)  
Стек  
Отображение виртуальной памяти на реальную (50%)  
Все перечисленное  
Ничего из перечисленного

47. Какие общие ресурсы есть у процессов?

Память  
Стек  
Отображение виртуальной памяти на реальную  
Все перечисленное  
Ничего из перечисленного

48. Чем характеризуется состояние параллельной программы?

адресами выполняемых команд  
последовательностью состояний  $s_0 \rightarrow s_1 \rightarrow \dots \rightarrow s_n$ .  
значениями переменных в некоторый момент времени  
объемом занимаемой оперативной памяти

49. Чем характеризуется история параллельной программы?

значением переменных в некоторый момент времени  
последовательностью состояний  $s_0 \rightarrow s_1 \rightarrow \dots \rightarrow s_n$ .  
адресами выполняемых команд  
объемом занимаемой оперативной памяти

50. Цель синхронизации процессов

исключить нежелательные истории  
обеспечить одновременное выполнение  
обеспечить исключительный доступ к данным  
исключить заикливание программы

51. Какими свойствами должна обладать параллельная программа?

Живучесть (50%)  
Эффективность  
Верифицируемость  
Безопасность (50%)

52. Какие существуют виды синхронизации?

Исключительная ситуация  
Взаимное исключение (50%)  
Условная синхронизация (50%)  
Абсолютная синхронизация

ПК-1.4 Осуществлять отладку программ, написанных на языке высокого уровня

Обучающийся знает: инструментальные среды для отладки программ, написанных на языках низкого уровня

Примеры вопросов

53. Взаимное исключение состоит в...  
обеспечение совместного доступа к общей памяти  
ожидании в одном процессе окончания выполнения другого



задержке процесса, пока не выполнится некоторое условие выделения в процессах критических секций, которые не прерываются другими процессами, использующими те же переменные

54. Условная синхронизация заключается в ...  
обеспечение совместного доступа к общей памяти  
ожидании в одном процессе окончания выполнения другого  
задержке процесса, пока не выполнится некоторое условие выделения в процессах критических секций, которые не прерываются другими процессами, использующими те же переменные

55. Какие из приведенных условий относятся к свойствам безопасности?  
Взаимное исключение. В любой момент только один процесс может выполнять свою критическую секцию (30%)

Отсутствие взаимной блокировки. Если несколько процессов пытаются войти в свои критические секции, хотя бы один сделает это (30%)

Если процесс пытается войти в критическую секцию, а другие выполняют некритические секции, то ему разрешается вход (40%)

Процесс, который пытается войти в критическую секцию когда-нибудь это сделает.

56. Какие из приведенных условий относятся к свойствам живучести?  
Взаимное исключение. В любой момент только один процесс может выполнять свою критическую секцию.  
Отсутствие взаимной блокировки. Если несколько процессов пытаются войти в свои критические секции, хотя бы один сделает это.

Если процесс пытается войти в критическую секцию, а другие выполняют некритические секции, то ему разрешается вход

Процесс, который пытается войти в критическую секцию когда-нибудь это сделает.

57. Перечислите алгоритмы критической секции со справедливой стратегией  
алгоритм разрыва узла (30%)  
алгоритм билета (30%)  
алгоритм семафора  
алгоритм поликлиники (40%)

58. Что представляет из себя справедливая стратегия?  
дать возможность каждому процессу попасть в критическую секцию  
дать возможность некоторым процессам попасть в критическую секцию  
дать возможность процессам попасть в критическую секцию в порядке очереди  
дать возможность каждому процессу выйти из критической секции

59. Недостатком алгоритма разрыва узла (Питерсона) является  
сложно обобщается на случай более двух процессов  
сложность, отсутствие грани между переменными синхронизации и другими переменными,  
неэффективность (ожидающие процессы постоянно проверяют переменные, что занимает время процессора)

60. Алгоритм билета основан на том, что  
обеспечивает поочередный вход двух процессов в критическую секцию  
каждый процесс, который пытается войти в CS получает номер, который больше номера любого из ранее вошедших  
каждый процесс запоминает номер выполняющегося процесса

61. В чем отличие мьютекса от критической секции?  
Критическая секция может находиться только в двух состояниях, а мьютекс – в нескольких  
Критическая секция должна быть описана в программе, а мьютекс – нет.  
Критическая секция действует в пределах одного процесса, а мьютекс может использоваться для взаимодействия разных процессов  
Мьютекс действует в пределах одного процесса, а критическая секция может использоваться для взаимодействия разных процессов

62. Выберите правильное утверждение  
Критическая секция позволяет реализовать взаимное исключение  
Критическая секция позволяет реализовать условную синхронизацию  
Критическая секция позволяет реализовать и взаимное исключение, и условную синхронизацию  
Критическая секция не позволяет реализовать ни взаимное исключение, ни условную синхронизацию

63. Что такое семафор?  
Процедура  
Объект

Специальная системная переменная  
Класс

64. Какие операции можно выполнить с семафором?

Открыть (50%)

Увеличить

Уменьшить

Закрывать (50%)

65. Какая операция с семафором может привести к приостановке процесса?

Открыть

Увеличить

Уменьшить

Закрывать

66. Что такое барьерная синхронизация?

взаимное исключение нескольких процессов

синхронизация по времени окончания операций в разных процессах

обеспечение общего доступа к данным

исключение взаимоблокировок

67. Основное требование, предъявляемое к барьерной синхронизации?

ни один из процессов не должен перейти барьер, пока к нему не подошли все процессы

ни один процесс не должен войти в секцию, если в нее вошел другой процесс

ни один процесс не может получить доступ к общим данным

ни один процесс не должен блокировать другие процессы

68. MPI - это ...

модуль параллельной обработки в системе

специальная ОС для параллельного программирования

интерфейс, содержащий набор функций, типов и констант для параллельного программирования

организация, координирующая разработку параллельных интерфейсов

70. В Вашей программе доля последовательных операций равна 0,4. Какое ускорение расчета программы Вы получите на ЭВМ с 4 процессорами? Ответ округлить до сотых.

71. Закон Амдаля рассчитывает:

время, затрачиваемое на вычисления

количество вложенных операций

глубину конвейера

ускорение при расчетах на нескольких процессорах

72. Как называется наиболее популярная модель параллельных вычислений

Граф «операции-операнды»

Граф «поток-данные»

Граф «чтение – запись»

Граф «переменные – алгоритмы»

73. Какие операции в модели параллельных вычислений могут выполняться параллельно?

Смежные в графе «операции-операнды»

Связанные путем в графе «операции-операнды»

Не связанные путем в графе «операции-операнды»

Не смежные в графе «операции-операнды»

74. Пусть  $p$  – количество процессоров.

Расписание  $H_p$  для каждой вершины (операции)  $i$  указывает номер процессора  $P_i$  и время начала операции  $t_i$ .

Расписание реализуемо, если

Для любых  $i, j$  :  $t_i = t_j \Rightarrow P_i \neq P_j$  т.е. один и тот же процессор не должен назначаться разным операциям в один и тот же момент. (50%)

Для любой дуги  $(i, j)$   $t_j \geq t_i + 1$  т.е. к началу операции все данные должны быть вычислены. (50%)

Для любых  $i, j$  :  $t_i < t_j \Rightarrow P_i \neq P_j$  т.е. один и тот же процессор не должен назначаться разным операциям в один и тот же момент.

Для любой дуги  $(i, j)$   $t_j < t_i + 1$  т.е. к началу операции все данные должны быть вычислены.

75. Что такое ускорение параллельной программы?

Отношение времени работы параллельной программы к времени работы последовательной программы

Отношение времени работы последовательной программы ко времени работы параллельной программы

Отношение времени работы самого медленного процесса к времени работы самого быстрого  
Отношение времени работы самого быстрого процесса к времени работы самого медленного

76. Что такое эффективность параллельной программы?

Отношение времени работы параллельной программы к времени работы последовательной программы  
Отношение времени работы последовательной программы ко времени работы параллельной программы  
Отношение количества процессоров к ускорению программы  
Отношение ускорения параллельной программы к количеству процессоров

77. Что такое стоимость вычислений по параллельной программе?

Наибольшее время выполнения параллельных процессов  
Наименьшее время выполнения параллельных процессов  
Полное время выполнения всех параллельных процессов  
Эффективность, умноженная на количество процессоров

78. Что такое сверхлинейное ускорение?

Когда ускорение равно количеству процессоров  
Когда ускорение меньше количества процессоров  
Когда эффективность больше единицы  
Когда эффективность меньше единицы

79. Когда возможно сверхлинейное ускорение?

При эффективной реализации параллельного алгоритма  
При грамотном распределении данных  
При нелинейной сложности алгоритма  
При увеличении количества процессоров

80. Пусть  $f$  – доля последовательных вычислений в алгоритме. Сформулируйте закон Амдала

$$S_p > 1/(f(1-f)/p)$$

$$S_p \leq 1/(f(1-f)/p)$$

$$S_p \leq 1/(f - (1-f)/p)$$

$$S_p > 1/(f(1-f)/p)$$

81. Масштабируемость алгоритма определяет

степень сохранения эффективности при уменьшении количества процессоров.  
степень сохранения эффективности при росте количества процессоров.  
степень увеличения ускорения при росте количества процессоров.  
степень увеличения стоимости при росте количества процессоров

82. Для сохранения эффективности обычно требуется

Увеличивать объем обрабатываемой информации.  
Уменьшать объем обрабатываемой информации.  
Увеличивать количество процессоров  
Уменьшать количество процессоров

83. Виды декомпозиции при разработке параллельных программ

Итеративный параллелизм  
Рекурсивный параллелизм  
Функциональный параллелизм (50%)  
Параллелизм по данным (50%)

84. Какие способы распределения данных используются при разработке матричных параллельных алгоритмов

Ленточное разбиение (30%)

Блочное разбиение (30%)

Диагональное разбиение

Циклическое разбиение (40%)

85. Какую роль играют семафоры в задаче о производителе и потребителе

флага доступа к данным  
нумеруют процессы в очереди обработки данных  
обеспечивают взаимное исключение (60%)  
счетчика ресурсов (40%)

86. В решении какой задачи используется метод передачи эстафеты?

Об обедающих философах  
О производителе и потребителе  
О писателях и читателях  
О критической секции

87. Как моделируются вилки в задаче об обедающих философях?

Каждая вилка – это процесс

Каждая вилка – это поток

Каждая вилка – это семафор

Каждая вилка – это массив мьютексов

88. Какая функция в Windows соответствует операции P для семафора

ReleaseSemaphore

WaitForSingleObject

CreateSemaphore

OpenSemaphore

90. Какая функция в Windows соответствует операции V для семафора

ReleaseSemaphore

WaitForSingleObject

CreateSemaphore

OpenSemaphore

91. Какая функция в Unix клонирует текущий процесс?

fork

execl, execv

waitpid

kill

signal

92. Интерфейс OpenMP задуман как стандарт параллельного программирования для

многопроцессорных систем с общей памятью

многопроцессорных систем с разделенной памятью

кластерных систем

матричных систем

93. Основания для достижения эффекта при использовании OPEN MP

разделяемые для параллельных процессов данные располагаются в общей памяти (50%)

для организации взаимодействия не требуется операций передачи сообщений (50%)

разделяемые для параллельных процессов данные располагаются в распределенной памяти

используется удаленный вызов процедур

94. Положительные стороны использования технологии OPEN MP

Можно распараллеливать последовательные программы поэтапно, не меняя их структуру (30%)

Нет необходимости поддерживать последовательный и параллельный вариант программы (40%)

Эффективно реализован обмен сообщениями

Поддержка в наиболее распространенных языках (C/C , Fortran) и платформах (Windows, Unix) (30%)

95. Принципы организации параллелизма в Open MP

Использование потоков (50%)

Условная синхронизация

Использование семафоров

Пульсирующий параллелизм (50%)

96. При появлении директивы #parallel происходит

синхронизация, все потоки, кроме главного, уничтожаются

продолжается последовательное выполнение кода (до очередного появления директивы #parallel)

создание “команды” (team) потоков для параллельного выполнения вычислений

97. После выхода из области действия директивы #parallel происходит

синхронизация, все потоки, кроме master, уничтожаются

продолжается последовательное выполнение кода (до очередного появления директивы #parallel)

создание “команды” (team) потоков для параллельного выполнения вычислений

98. Перечислите типы директив Open MP

Определение параллельной области (30%)

Описание общих данных

Разделение работы (30%)

Синхронизация (40%)

99. Какой параметр не может использоваться в директиве parallel?

operator (list)

private (list)  
 shared (list)  
 reduction (operator: list)

100. Параметр shared определяет список переменных, которые будут общими для всех потоков параллельной области: правильность использования таких переменных должна обеспечиваться программистом  
 будут локальными для каждого потока; переменные создаются в момент формирования потоков параллельной области; начальное значение переменных является неопределенным  
 перед использованием инициализируются значениями исходных переменных  
 запоминаются в исходных переменных после завершения параллельной области (используются значения потока, выполнившего последнюю итерацию цикла или последнюю секцию)

101. Взаимное исключение в Open MP может осуществляться  
 С помощью директивы critical (50%)  
 С помощью директивы atomic  
 С помощью функций библиотеки (50%)  
 С помощью директивы barrier

102. Переменные окружения в Open MP используются для  
 Управления барьерами в программе  
 Установки количества потоков (50%)  
 Управления распределением итераций в цикле (50%)  
 Получения номера потока

103. Что такое транспьютеры?  
 Кластерные системы  
 Многоядерные системы  
 Многопроцессорные системы, образующие двумерную решетку  
 Многопроцессорные системы, образующие гиперкуб

## 2.2 Типовые задания для оценки навыкового образовательного результата

Проверяемый образовательный результат:

Код и наименование индикатора достижения компетенции	Образовательный результат
ПК-1.3 Разрабатывать программный код на языках программирования высокого уровня	Обучающийся умеет: разрабатывать программный код на языках программирования низкого уровня
Примеры заданий 1. Изучить систему ПАРАЛАБ и ее функции 2. Смоделировать параллельные алгоритмы умножение матрицы на вектор: ленточное, разделение данных 3. Смоделировать параллельные алгоритмы умножение матрицы на вектор: столбцовое разделение данных 4. Смоделировать параллельные алгоритмы умножение матрицы на вектор: блочное разделение данных 5. Смоделировать параллельные алгоритмы умножение матриц: алгоритм Фокса	
ПК-1.4 Осуществлять отладку программ, написанных на языке высокого уровня	Обучающийся умеет: отлаживать программы, написанные на языках программирования низкого уровня
Примеры заданий 6. Смоделировать параллельные алгоритмы умножения матриц: алгоритм Кэннона 7. Разработать показатель эффективности параллельных вычислений 8. Смоделировать параллельные алгоритмы решения систем линейных уравнений: алгоритм Гаусса 9. Смоделировать параллельные алгоритмы решения систем линейных уравнений: алгоритм сопряженных градиентов 10. Смоделировать параллельные алгоритмы сортировки: пузырьковая сортировка	
ПК-1.3 Разрабатывать программный код на языках программирования высокого уровня	Обучающийся владеет: навыками использования инструментальных сред для разработки программ на языках низкого уровня

Примеры заданий	
11. Смоделировать параллельные алгоритмы сортировки: алгоритм Шелла 12. Смоделировать параллельные алгоритмы сортировки: быстрая сортировка сортировка 13. Смоделировать параллельные алгоритмы графов: алгоритм Флойда 14. Смоделировать параллельные алгоритмы обработки графов: алгоритм Прима 15. Смоделировать параллельные алгоритмы обработки графов: алгоритм Дейкстры	
ПК-1.4 Осуществлять отладку программ, написанных на языке высокого уровня	Обучающийся владеет: навыками использования инструментальных сред для отладки программ, написанных на языках низкого уровня
Примеры заданий	
Для заданной группы вычислительных процессов организовать доступ к критической секции с использованием (по указанию преподавателя): блокирующей переменной, 16.семафора, 17.мьютекса, 18.монитора, 19.барьера, 20.обмена сообщениями.	

### 2.3. Перечень вопросов для подготовки обучающихся к промежуточной аттестации

#### 4 семестр

##### Вопросы к зачету с оценкой:

- 1.Классификация вычислительных систем (ВС). Классификации Флина, Хокни, Фенга, Хендлера, Шнайдера. Взаимосвязь классификаций ВС.
- 2.Параллелизм как основа высокопроизводительных вычислений.
- 3.Показатели, характеристики и критерии эффективности ВС. Способы построения критериев эффективности ВС.
- 4.Технико-экономическая эффективность функционирования ВС.
- 5.Основные направления развития архитектуры процессоров ВС.
- 6.Конвейеризация вычислений. Показатели эффективности конвейеров.
- 7.Методы решения проблемы условного перехода. Суперконвейерные процессоры.
- 8.Основные направления развития архитектуры процессоров ВС. Процессоры с полным набором команд (CISC).
- 9.Процессоры с сокращенным набором команд (RISC). Особенности архитектуры RISC процессоров. Типы серийно производимых RISC процессоров.
- 10.Основные направления развития архитектуры процессоров ВС. Суперскалярные процессоры. Особенности реализации суперскалярных процессоров. Аппаратная поддержка суперскалярных операций.
- 11.Процессоры со сверхдлинным командным словом (VLIW архитектурой).
- 12.Эмпирические законы Мура, Х. Гроша, кривая обучаемости. Уровни параллелизма и метрики параллельных вычислений.
- 13.Предельные оценки ускорения вычислений. Первый, второй и третий законы Дж. Амдала.
- 14.Закон Густавсона – Барсиса.
- 15.Топологии ВС. Метрика сетевых топологий. Функции маршрутизации данных.
- 16.Статические топологии: линейная, кольцевая, звездообразная, древовидная и др.
- 17.Динамические топологии ВС. ВС с программируемой структурой.
- 18.Векторные и матричные ВС. Понятие вектора и размещения данных в памяти.
- 19.Структура векторного процессора. Обработка длинных векторов и матриц. Массив процессоров.
- 20.Ассоциативные ВС. ВС с систолической архитектурой. Классификация систолических структур. Топология систолических структур. Процессорные элементы систолических структур.
- 21.Симметричные (SMP) и асимметричные (ASMP) ВС. Архитектура SMP и ASMP систем.
- 22.ВС с массовым параллелизмом (MPP). Кластерные ВС. Архитектура кластерных ВС.
- 23.ВС с управлением вычислений от потока данных. Вычислительная модель потоковой обработки. Статические и динамические потоковые ВС.
- 24.Проблемно-ориентированные и специализированные ВС. Показатели специализации и их количественная оценка. Определение критерия степени специализации МС и выбор его рационального значения.
- 25.Программируемые контроллеры, программируемые логические интегральные схемы, сигнальные процессоры. Особенности их архитектуры и организации вычислений.
- 26.Перспективные методы обработки данных. Проблема отображения структуры алгоритма решаемого класса задач на структуры ВС.
- 27.ВС с обработкой по принципу волнового фронта.
- 28.Нейрокомпьютеры и искусственные нейронные сети.
- 29.Организация памяти в ВС. Модели архитектур совместно используемой памяти.
- 30.Мультипроцессорный и мультипрограммный способы организации вычислительных процессов. Мультипроцессорные (многопроцессорные) вычислительные системы. Многопроцессорный режим работы, его достоинства и недостатки.

- 31.Определение арбитража. Виды централизованного и распределенного арбитража.
- 32.Мультипрограммные системы. Способы реализации мультипрограммного режима. Мультипрограммирование в системах пакетной обработки, в системах разделения времени, системах реального времени.
- 33.Управление задачами в ОС. Планирование и диспетчеризация процессов потоков.
- 34.Стратегии планирования и дисциплины диспетчеризации. Граф состояния процессов и потоков.
- 35.Принципы планирования процессов и потоков. Классификация алгоритмов планирования.
- 36.Вытесняющие и невытесняющие алгоритмы планирования ОС. Приоритетные и бесприоритетные алгоритмы планирования.
- 37.Алгоритмы планирования, основанные на квантовании. Обоснование выбора величины квантов времени. Задание квантов времени в мультипрограммных ОС и управление их величиной.
- 38.Алгоритмы планирования, основанные на приоритетах. Понятие приоритета и очереди процессов. Абсолютные и относительные приоритеты.
- 39.Смешанные алгоритмы планирования. Алгоритмы планирования в ОС реального времени. Планирование на основе предельных начальных или конечных сроков решения задач.
- 40.Частотно-монотонное планирование в ОС. Законы Лью – Лейланда.
- 41.Алгоритмы планирования в ОС Windows 2000 и Windows XP. Учет квантов и управление их величиной. Динамическое повышение приоритета.
- 42.Синхронизация процессов и потоков в ОС. Эффект гонок. Необходимость синхронизации. Критические секции и критические данные.

### **3. Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации**

#### **Критерии формирования оценок по ответам на вопросы по выполнению тестовых заданий**

- оценка **«отлично»** выставляется обучающемуся, если количество правильных ответов на вопросы составляет 100 – 90 % от общего объёма заданных вопросов;
- оценка **«хорошо»** выставляется обучающемуся, если количество правильных ответов на вопросы – 89 – 76 % от общего объёма заданных вопросов;
- оценка **«удовлетворительно»** выставляется обучающемуся, если количество правильных ответов на тестовые вопросы – 75–60 % от общего объёма заданных вопросов;
- оценка **«неудовлетворительно»** выставляется обучающемуся, если количество правильных ответов – менее 60 % от общего объёма заданных вопросов.

#### **Критерии формирования оценок по результатам выполнения заданий**

**Зачтено»** – ставится за работу, выполненную полностью без ошибок и недочетов в соответствии с заданием, выданным для выполнения лабораторной работы. Обучающийся полностью владеет информацией по теме работы, решил все поставленные в задании задачи.

**«Не зачтено»** - ставится за работу, если обучающийся правильно выполнил менее 2/3 всей работы, использовал при выполнении работы неправильные алгоритмы, допустил грубые ошибки при программировании, сформулировал неверные выводы по результатам работы

*Виды ошибок:*

- *грубые ошибки: незнание основных понятий, правил, норм; незнание приемов решения задач; ошибки, показывающие неправильное понимание условия предложенного задания.*
- *негрубые ошибки: неточности формулировок, определений; нерациональный выбор хода решения.*
- *недочеты: нерациональные приемы выполнения задания; отдельные погрешности в формулировке выводов; небрежное выполнение задания.*

#### **Критерии формирования оценок по зачету с оценкой**

**«Отлично»** – студент приобрел необходимые умения и навыки, продемонстрировал навык практического применения полученных знаний, не допустил логических и фактических ошибок

**«Хорошо»** – студент приобрел необходимые умения и навыки, продемонстрировал навык практического применения полученных знаний; допустил незначительные ошибки и неточности.

**«Удовлетворительно»** – студент допустил существенные ошибки.

**«Неудовлетворительно»** – студент демонстрирует фрагментарные знания изучаемого курса; отсутствуют необходимые умения и навыки, допущены грубые ошибки.