

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[[1]](#footnote-2) контрольные задания или иные материалы для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций**

**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 |
| Примеры вопросов1.Когда была создана первая супер ЭВМ?1.1 в середине 70-х1.2 в середине 60-х1.3 в начале 80-х1.4 в начале 80-х1.5 в конце 70-х2.Кем была разработана первая супер-ЭВМ?2.1 Джоном фон Нейманом2.2 Сеймуром Крэем2.3 Томасом Стерлингом2.4 Доном Беккером2.5 Биллом Гейтсом3. Укажите неправильное утверждение.SISD - это обычные последовательные компьютерыSIMD - большинство современных ЭВМ относятся к этой категорииMISD - вычислительных машин такого класса малоMIMD -это реализация нескольких потоков команд и потоков данных4. Для конвейерной обработки присуще:загрузка операндов в векторные регистрыопераций с матрицамивыделение отдельных этапов выполнения общей операциисложение 2-х операндов одновременным сложением всех их двоичных разрядов5. Приоритет - это...описание алгоритма на некотором формализованном языкечисло, приписанное ОС каждому процессу или задачеотдельный этап выполнения общей операцииоповещение со стороны ОС о той или иной форме взаимодействия6. Стек - это..."память", в адресном пространстве которой работает процесстот или иной способ передачи инструкции из одного процесса в другойобласть памяти для локальных переменных, аргументов и возвращаемых функциями значенийорганизация доступа 2х (или более) процессов к одному и тому же блоку памяти7. Кластер (в контексте параллельного программирования)- это...область оперативной памятиуправляющее устройство, выполненное на одном или более кристаллах2 или более узлов, соединенных при помощи локальной сети раздел жесткого дискасуперкомпьютер для выполнения особых задач8. Выберите шаг(и), не присущий(е) для цикла выполнения команды:запись результата в памятьвыборка командыкэширование следующей командывыполнение командыдекодирование команды, вычисление адреса операнда и его выборкаобращение к памяти9. Конвейерная технология предполагает …последовательную обработку командобработку команд, удовлетворяющих определенным критериямобработку несколько команд одновременнообщий доступ команд к памяти 10 Система, главной особенностью является наличие общей физической памяти, разделяемой всеми процессорами называется ...NUMАSMPMPPPVP11. Главная особенность архитектуры NUMA?неоднородный доступ к памятисверхвысокая производительностьналичие векторно-конвейерных процессоровналичие общей физической памяти, разделяемой всеми процессорами12. Вычислительные машины с какой архитектурой наиболее дешевы?симметричная многопроцессорная обработка параллельная архитектура с векторными процессорамикластерные системымассивно-параллельная архитектура13. Пиковая производительность системы измеряется в:МегагерцMIPSМFlopsMByte14. Пиковая производительность системы определяется:временем выполнения реальных задачпроизведением производительности 1-го процессора на число процессоров в системевременем выполнения тестовых задачколичеством переданной информации15. Производительность многопроцессорной вычислительной системы характеризуется:количеством операций, производимых за единицу времениколичеством байт информации, переданных в единицу временичислом импульсов, генерируемых в единицу времениобъемом располагаемой для вычислений памяти16. Какое понятие характеризует возрастание сложности соединений при добавлении в конфигурацию новых узлов. масштабируемостьускорениеэффективность пиковая производительность17. 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 памятьюс разделяемой памятьюс распределенной памятьюс когерентной кэш-памятью19. Укажите наиболее быструю организацию сети для кластера.Gigabit EthernetMyrinetInfinybandEthernet20. Параллельная программа – это…программа, работающая одновременно на нескольких компьютерахпрограмма, обрабатывающая большой объем данныхпрограмма, осуществляющая обмен сообщениями в сетипрограмма, содержащая несколько процессов, работающих совместно21. Асинхронная модель параллельных вычислений имеет следующие особенности:все процессы выполняют одни и те же действия с собственными даннымиразличные процессы решают разные задачивсе процессы используют общую памятьвсе процессы выполняются в своих критических секциях22. Синхронная модель параллельных вычислений имеет следующие особенности:все процессы выполняют одни и те же действия с собственными даннымиразличные процессы решают разные задачивсе процессы используют общую памятьвсе процессы выполняются в своих критических секциях23. Две операции называются независимыми еслимножество чтения одной не пересекается с множеством чтения другоймножество чтения одной не пересекается с множеством записи другоймножество чтения одной пересекается с множеством записи другоймножество чтения одной пересекается с множеством чтения другой24. Какие операции могут выполняться параллельно?независимыезависимые элементарныенеделимые 25. Какой процесс называется потребителем?Процесс, передающий данные Процесс, получающий данныеПроцесс, вводящий данныеПроцесс, выводящий данные26. Какой процесс называется производителем?Процесс, передающий данные Процесс, получающий данныеПроцесс, вводящий данныеПроцесс, выводящий данные27. Какие технологии повышения производительности применяются в современных процессорах?Суперскалярность (30%)МногопоточностьКонвейеризация (30%)Векторная обработка данных (40%)28. Что такое конвейеризация? Исполнение нескольких команд одновременноПараллельное выполнение различных частей командСохранение данных в сверхбыстрой памятиОбработка данных большого размера29. Что такое суперскалярность? Исполнение нескольких команд одновременноПараллельное выполнение различных частей командСохранение данных в сверхбыстрой памятиОбработка данных большого размера31. Какие виды оптимизации применяются при конвейеризации?Предсказание переходов (30%)Замена командПерестановка команд (30%)Переименование регистров (40%)32. Какие системы относятся к технологии SIMD?Векторные процессоры (50%)Матричные процессоры (50%)КластерыSMP33. Какие системы относятся к технологии MIMD?Векторные процессорыМатричные процессорыКластеры (50%)Симметричные многопроцессорные (50%)34. К какому классу относятся многоядерные системы?Матричные процессорыРаспределенные системыКластерыСистемы с общей памятью35. К какому классу относятся кластерные системы?Матричные процессорыРаспределенные системыСимметричные мультипроцессорыСистемы с общей памятью36. Процесс - это...сетевой интерфейс контроллера блочных передачэто число, приписанное операционной системой каждой задачеэто динамическая сущность программы, ее код в процессе своего выполнениясистема, выполняющая повторяющуюся операцию37.Ресурс - это...объект, необходимый для работы процессу или задачесообщение, доставляемое процессу посредством ОСпроцесс превращения скомпилированного кода в программучисло, приписанное ОС каждому процессу и задаче38. Выберите верное утверждение. Активные ресурсы...используют взаимные исключениямогут быть использованы одновременно несколькими процессамиспособны изменять информацию в памятииспользуются только одним процессором, пока тот не завершит работу с ресурсом39. Функцией мьютекса является:регистрация обработчика сообщения в операционной системераспределение квантов времени в системе между выполняющимися процессамиспособ синхронизации параллельных процессов через разделяемый критический ресурсспособ обмена данными процессорами через разделяемую память или коммутируемый канал 40. Процесс имеет:собственное состояниесобственный процессорсобственную системусобственный семафор41. Барьер - это...подпрограмма, определяющая факт прихода сообщенияместо в программе, где процесс ожидает подхода к нему остальных процессовблокировка процесса до тех пор, пока все операции обмена не будут завершеныожидание завершения асинхронных процедур, ассоциированных с идентификатором42. Семафор - это ...аппаратный коммутаторустройство синхронизации для параллельных ЭВМпрограммный механизм синхронизации в виде переменной в общей памяти43. Какие сущности имеют общую память? Два процессаДва потока Поток и процессВычислительные узлы кластера44. Что имеет собственную память для данных?ПроцессПотокИ процесс, и поток45. Как организуется взаимодействие процессов?Через общую памятьЧерез обмен сообщениями (50%)Через файловую систему (50%)Через регистры процессора46. Какие общие ресурсы есть у потоков?Память для данных (50%)СтекОтображение виртуальной памяти на реальную (50%)Все перечисленноеНичего из перечисленного47. Какие общие ресурсы есть у процессов?ПамятьСтекОтображение виртуальной памяти на реальнуюВсе перечисленноеНичего из перечисленного48. Чем характеризуется состояние параллельной программы?адресами выполняемых командпоследовательностью состояний s0->s1->…->sn. значениями переменных в некоторый момент времениобъемом занимаемой оперативной памяти49. Чем характеризуется история параллельной программы?значением переменных в некоторый момент временипоследовательностью состояний s0->s1->…->sn. адресами выполняемых командобъемом занимаемой оперативной памяти50. Цель синхронизации процессов исключить нежелательные историиобеспечить одновременное выполнениеобеспечить исключительный доступ к даннымисключить зацикливание программы51. Какими свойствами должна обладать параллельная программа?Живучесть (50%)ЭффективностьВерифицируемостьБезопасность (50%)52. Какие существуют виды синхронизации?Исключительная ситуацияВзаимное 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 вопросов53. Взаимное исключение состоит в…обеспечение совместного доступа к общей памятиожидании в одном процессе окончания выполнения другогозадержке процесса, пока не выполнится некоторое условиевыделении в процессах критических секций, которые не прерываются другими процессами, использующими те же переменные54 Условная синхронизация заключается в …обеспечение совместного доступа к общей памятиожидании в одном процессе окончания выполнения другогозадержке процесса, пока не выполнится некоторое условиевыделении в процессах критических секций, которые не прерываются другими процессами, использующими те же переменные55. 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 входПроцесс, который пытается войти в критическую секцию когда-нибудь это сделает.57. Перечислите алгоритмы критической секции со справедливой стратегиейалгоритм разрыва узла (30%)алгоритм билета (30%)алгоритм семафораалгоритм поликлиники (40%)58. Что представляет из себя справедливая стратегия?дать возможность каждому процессу попасть в критическую секциюдать возможность некоторым процессам попасть в критическую секциюдать возможность процессам попасть в критическую секцию в порядке очередидать возможность каждому процессу выйти из критической секции59. Недостатком алгоритма разрыва узла (Питерсона) являетсясложно обобщается на случай более двух процессовсложность, 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 секциюкаждый процесс, который пытается войти в CS получает номер, который больше номера любого из ранее вошедшихкаждый процесс запоминает номер выполняющегося процесса61. 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 утверждениеКритическая секция позволяет реализовать взаимное исключениеКритическая секция позволяет реализовать условную синхронизациюКритическая секция позволяет реализовать и взаимное исключение, и условную синхронизациюКритическая секция не позволяет реализовать ни взаимное исключение, ни условную синхронизацию63. Что такое семафор?ПроцедураОбъектСпециальная системная переменнаяКласс64. Какие операции можно выполнить с семафором?Открыть (50%)УвеличитьУменьшитьЗакрыть (50%)65. Какая операция с семафором может привести к приостановке процесса? ОткрытьУвеличитьУменьшитьЗакрыть 66. Что такое барьерная синхронизация?взаимное исключение нескольких процессовсинхронизация по времени окончания операций в разных процессахобеспечение общего доступа к даннымисключение взаимоблокировок67. Основное требование, предъявляемое к барьерной синхронизации?ни один из процессов не должен перейти барьер, пока к нему не подошли все процессыни один процесс не должен войти в секцию, если в нее вошел другой процессни один процесс не может получить доступ к общим даннымни один процесс не должен блокировать другие процессы68. MPI - это ...модуль параллельной обработки в системеспециальная ОС для параллельного программированияинтерфейс, содежащий набор функций, типов и констант для параллельного программированияорганизация, координирующая разработку параллельных интерфейсов70. 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 вычисленияколичество вложенных операцийглубину конвейераускорение при расчетах на нескольких процессорах72. Как называется наиболее популярная модель параллельных вычисленийГраф «операции-операнды»Граф «потоки-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 Hp для каждой вершины (операции) i указывает номер процессора Pi и время начала операции ti. Расписание реализуемо, если Для любых i,j : ti = tj => Pi ≠ Pj т.е. один и тот же процессор не должен назначаться разным операциям в один и тот же момент. (50%)Для любой дуги (i,j) tj ≥ ti 1 т.е. к началу операции все данные должны быть вычислены. (50%) Для любых i,j : ti <> tj => Pi ≠ Pj т.е. один и тот же процессор не должен назначаться разным операциям в один и тот же момент.Для любой дуги (i,j) tj < ti 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 программыОтношение времени работы последовательной программы ко времени работы параллельной программыОтношение времени работы самого медленного процесса к времени работы самого быстрогоОтношение 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 программыОтношение времени работы последовательной программы ко времени работы параллельной программыОтношение количества процессоров к ускорению программыОтношение ускорения параллельной программы к количеству процессоров77. Что такое стоимость вычислений по параллельной программе? Наибольшее время выполнения параллельных процессовНаименьшее время выполнения параллельных процессовПолное время выполнения всех параллельных процессовЭффективность, умноженная на количество процессоров78. Что такое сверхлинейное ускорение? Когда ускорение равно количеству процессоровКогда ускорение меньше количества процессоровКогда эффективность больше единицыКогда эффективность меньше единицы79. Когда возможно сверхлинейное ускорение?При эффективной реализации параллельного алгоритмаПри грамотном распределении данныхПри нелинейной сложности алгоритма При увеличении количества процессоров80. Пусть f – доля последовательных вычислений в алгоритме. Сформулируйте закон АмдаляSp > 1/(f (1-f)/p) Sp ≤ 1/(f (1-f)/p) Sp ≤ 1/(f - (1 f)/p) Sp > 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 процессоров82. Для сохранения эффективности обычно требуется Увеличивать объем обрабатываемой информации. Уменьшать объем обрабатываемой информации.Увеличивать количество процессоровУменьшать количество процессоров83. Виды декомпозиции при разработке параллельных программИтеративный параллелизмРекурсивный параллелизмФункциональный параллелизм (50%)Параллелизм по данным (50%)84. Какие способы распределения данных используются при разработке матричных параллельных алгоритмов Ленточное разбиение (30%)Блочное разбиение (30%)Диагональное разбиениеЦиклическое разбиение (40%)85. Какую роль играют семафоры в задаче о производителе и потребителефлага доступа к даннымнумеруют процессы в очереди обработки данныхобеспечивают взаимное исключение (60%)счетчика ресурсов (40%)86. В решении какой задачи используется метод передачи эстафеты?Об обедающих философахО производителе и потребителеО писателях и читателяхО критической секции87. Как моделируются вилки в задаче об обедающих философах?Каждая вилка – это процессКаждая вилка – это потокКаждая вилка – это семафорКаждая вилка – это массив мьютексов88. Какая функция в Windows соответствует операции P для семафораReleaseSemaphoreWaitForSingleObjectCreateSemaphore OpenSemaphore90. Какая функция в Windows соответствует операции V для семафораReleaseSemaphoreWaitForSingleObjectCreateSemaphore OpenSemaphore91. Какая функция в Unix клонирует текущий процесс?forkexecl, execv waitpidkillsignal92. Интерфейс OpenMP задуман как стандарт параллельного программирования длямногопроцессорных систем с общей памятью многопроцессорных систем с разделенной памятью кластерных системматричных систем93. Основания для достижения эффекта при использовании OPEN MPразделяемые 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 процедур94. Положительные стороны использования технологии OPEN MPМожно 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 MPИспользование потоков (50%)Условная синхронизацияИспользование семафоровПульсирующий параллелизм (50%)96. При появлении директивы #parallel происходит синхронизация, все потоки, кроме главного, уничтожаютсяпродолжается последовательное выполнение кода (до очередного появления директивы #parallel)создание “команды” (team) потоков для параллельного выполнения вычислений97. После выхода из области действия директивы #parallel происходит синхронизация, все потоки, кроме master, уничтожаютсяпродолжается последовательное выполнение кода (до очередного появления директивы #parallel)создание “команды” (team) потоков для параллельного выполнения вычислений98. Перечислите типы директив Open MPОпределение 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, которыебудут общими для всех потоков параллельной области: правильность использования таких переменных должна обеспечиваться программистомбудут локальными для каждого потока; переменные создаются в момент формирования потоков параллельной области; начальное значение переменных является неопределеннымперед использованием инициализируются значениями исходных переменныхзапоминаются в исходных переменных после завершения параллельной области (используются значения потока, выполнившего последнюю итерацию цикла или последнюю секцию)101. Взаимное исключение в Open MP может осуществлятьсяС помощью директивы critical (50%)С помощью директивы atomic С помощью функций библиотеки (50%)С помощью директивы barrier102. Переменные окружения в Open MP используются дляУправления барьерами в программеУстановки количества потоков (50%)Управления распределением итераций в цикле (50%)Получения номера потока103. Что такое транспьютеры? Кластерные системыМногоядерные системыМногопроцессорные 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 заданий1.Изучить систему ПАРАЛАБ и ее функции2.Смоделировать параллельные алгоритмы умножение матрицы на вектор: ленточное, разделение данных3. Смоделировать параллельные алгоритмы умножение матрицы на вектор: столбцовое разделение данных4. Смоделировать параллельные алгоритмы умножение матрицы на вектор: блочное разделение данных5. 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 заданий6. Смоделировать параллельные алгоритмы умножения матриц: алгоритм Кэннона7.Разработать показатель эффективности параллельных вычислений8. 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 заданийДля 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**

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 (МРР). 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 вы-бора 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 кван-тов 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.

1. 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 социальной сферы. Ответственность за нераспространение содержания оценочных средств среди обучающихся университета несут заведующий кафедрой и преподаватель – разработчик оценочных средств. [↑](#footnote-ref-2)