

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Порохня Андрей Алексеевич
Должность: И.о. директора института перспективной инженерии
Дата подписания: 02.06.2026 19:53:32
Уникальный программный ключ:
990bea3aeec848c23bd8699e48288f8d1960c608

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»



А.А. Порохня
Ф.И.О.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ
Параллельные вычисления / Parallel computing
название дисциплины

Направление подготовки	09.04.03 Прикладная информатика
Направленность (профиль)	Управление знаниями
Год начала обучения	2026
Форма обучения	очная
Реализуется в семестре	1

Введение

1. Назначение: ФОС предназначен для объективной оценки уровня сформированности компетенций.

2. ФОС является приложением к программе дисциплины «Параллельные вычисления / Parallel computing»

3. Разработчик Амироков С.Р., доцент.

4. Проведена экспертиза ФОС.

Члены экспертной группы:

Председатель Азаров И.В., и.о. директора департамента

Члены комиссии:

Орлова А.Ю., доцент

Орлинская О.Г., доцент

Представитель организации-работодателя Черевко С.А., генеральный директор ООО "КРАФТКОД"

(Ф.И.О., должность)

Экспертное заключение: ФОС по дисциплине позволяет оценить уровень сформированности компетенций. Приступить к апробации.

5. Срок действия ФОС определяется сроком реализации образовательной программы.

1. Описание критериев оценивания компетенции на различных этапах их формирования, описание шкал оценивания

Уровни сформированности и компетенции(ий), индикатора (ов)	Дескрипторы			
	Минимальный уровень не достигнут (Неудовлетворительно) 2 балла	Минимальный уровень (удовлетворительно) 3 балла	Средний уровень (хорошо) 4 балла	Высокий уровень (отлично) 5 баллов
УК-6				
ИД-1 УК-6. достигает результат путем саморазвития.	С трудом достигает результат путем саморазвития	Достигает результат путем саморазвития не в полном объеме	В целом достигает результат путем саморазвития	достигает результат путем саморазвития
ИД-2 УК-6. использует методики целеполагания.	С трудом использует методики целеполагания	Слабо использует методики целеполагания	В целом использует методики целеполагания	использует методики целеполагания
ОПК-2				
ИД-1 ОПК-2. разрабатывать оригинальные алгоритмы для решения профессиональных задач;	С трудом разрабатывает оригинальные алгоритмы для решения профессиональных задач	Слабо разрабатывает оригинальные алгоритмы для решения профессиональных задач	В основном разрабатывает оригинальные алгоритмы для решения профессиональных задач	разрабатывает оригинальные алгоритмы для решения профессиональных задач
ИД-2 ОПК-2. разрабатывать программные средства для решения профессиональных задач	С трудом разрабатывает программные средства для решения профессиональных задач	Слабо разрабатывает программные средства для решения профессиональных задач	В основном разрабатывает программные средства для решения профессиональных задач	разрабатывает программные средства для решения профессиональных задач
ПК-3				
ИД-1 ПК-3. развивать фундаментальные и междисциплинарные знания	Не может развивать фундаментальные и междисциплинарные знания	Допускает серьезные ошибки в развитии фундаментальных и междисциплинарных знаний	В основном точно развивает фундаментальные и междисциплинарные знания	развивает фундаментальные и междисциплинарные знания
ИД-2 ПК-3. Развивать методы научных исследований	С трудом развивает методы научных исследований	Допускает серьезные ошибки в развитии методов научных исследований	В основном точно развивает методы научных исследований	развивает методы научных исследований
ПК-10				
ИД-1 ПК-10. развертывать, информационные системы, их компоненты и информационные сервисы	С трудом развертывает, информационные системы, их компоненты и информационные сервисы	Слабо развертывает, информационные системы, их компоненты и информационные сервисы	В целом развертывает, информационные системы, их компоненты и информационные сервисы	развертывает, информационные системы, их компоненты и информационные сервисы
ИД-2 ПК-10 инсталлировать информационные системы, их компоненты и	С трудом инсталлирует информационные системы, их компоненты и	Слабо инсталлирует информационные системы, их компоненты и	В целом инсталлирует информационные системы, их компоненты и	инсталлирует информационные системы, их компоненты и

информационные сервисы	информационные сервисы	информационные сервисы	информационные сервисы	информационные сервисы
---------------------------	---------------------------	---------------------------	---------------------------	---------------------------

Оценивание уровня сформированности компетенции по дисциплине осуществляется на основе «Положения о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся по образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры - в федеральном государственном автономном образовательном учреждении высшего образования «Северо-Кавказский федеральный университет» в актуальной редакции.

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
		Форма обучения очная, семестр 1	
1.	од параллельной программой в рамках MPI понимается множество одновременно выполняемых процессов. Процессы могут выполняться на разных процессорах, но на одном процессоре могут располагаться и несколько процессов (в этом случае их исполнение осуществляется в режиме разделения времени). В предельном случае для выполнения параллельной программы может использоваться один процессор – как правило, такой способ применяется для начальной проверки правильности параллельной программы.	Понятие параллельной программы	
2.	Процессы параллельной программы объединяются в группы. Другим важным понятием MPI, описывающим набор процессов, является понятие коммуникатора. Под коммуникатором в MPI понимается специально создаваемый служебный объект, который объединяет в своем составе группу процессов и ряд дополнительных параметров (контекст), используемых при выполнении операций передачи данных. Парные операции передачи данных выполняются только для процессов, принадлежащих одному и тому же коммуникатору. Коллективные операции применяются одновременно для всех процессов	Понятие коммуникаторов	

	одного коммуникатора. Как результат, указание используемого коммуникатора является обязательным для операций передачи данных в MPI.		
3.	Первой вызываемой функцией MPI должна быть функция: int MPI_Init(int *argc, char ***argv), где argc — указатель на количество параметров командной строки, argv — параметры командной строки, применяемая для инициализации среды выполнения MPI -программы. Параметрами функции являются количество аргументов в командной строке и адрес указателя на массив символов текста самой командной строки.	Инициализация MPI-программ	
4.	Одним из наиболее распространенных способов классификации ЭВМ является систематика Флинна (Flynn), в рамках которой основное внимание при анализе архитектуры вычислительных систем уделяется способам взаимодействия последовательностей (поток) выполняемых команд и обрабатываемых данных.	Что положено в основу классификации Флинна?	
5.	1,3	Чем характеризуется многопроцессность (concurrency) в контексте параллельных вычислений? (1) обеспечение минимального времени выполнения одной программы (2) первичность пропускной способности (3) не требуется обеспечение максимальной изоляции процессов друг от друга (4) обеспечение как можно более равномерного распределения ресурсов между процессами	
6.	3,4	В каких ситуациях может быть реализован истинный параллелизм вычислений?	

		(1) вычисления производятся на ЭВМ с одноядерным процессором в многозадачной ОС (2) вычисления производятся на ЭВМ с одноядерным процессором в однозадачной ОС (3) вычисления производятся на многопроцессорном устройстве (4) для вычислений применяется процессор, поддерживающий физическую векторизацию	
7.	3	Какой из режимов вычислений поддерживает классический последовательный компьютер фон Неймана? (1) обработка нескольких инструкций и одиночного элемента данных в каждый момент времени (2) обработка одиночной инструкции и нескольких потоков данных в каждый момент времени (3) обработка одиночной инструкции и одиночного элемента данных в каждый момент времени (4) обработка нескольких инструкций и нескольких потоков данных в каждый момент времени	
8.	1	Симметричный мультипроцессор характеризуется (1) однородным доступом к памяти всех процессорных устройств (2) доступом каждого процессорного устройства к отдельной части физически распределенной памяти (3) неоднородным доступом к памяти всех процессорных устройств (4) однородным доступом части процессорных устройств к части общей памяти	
9.	1,3	Какие факторы препятствуют получению результата с ожидаемой точностью при распараллеливании арифметических расчетов? (1) машинная операция сложения чисел не обладает свойством коммутативности (порядок сложения двух чисел важен)	

		(2) распараллеленные алгоритмы реализуются на одноплатном процессоре (3) машинная операция умножения чисел не обладает свойством ассоциативности (важен порядок перемножения трёх чисел) (4) машинная операция умножения чисел обладает свойством коммутативности (порядок перемножения двух чисел не важен)	
10.	1,2,3	Какие из предложенных стратегий распараллеливания алгоритма нахождения среднего арифметического последовательности из 1000 чисел корректны? (1) последовательность разбивается на 4 части, элементы в каждой части суммируются и делятся на 1000 на отдельном процессорном устройстве, полученные значения складываются на одном процессорном устройстве и делятся на 4 (2) последовательность разбивается на 4 равные части, элементы в каждой части суммируются, полученные значения складываются и делятся на 1000 на одном процессорном устройстве (3) последовательность разбивается на 4 равные части, элементы в каждой части суммируются и делятся на 1000 на отдельном процессорном устройстве, полученные значения складываются на одном процессорном устройстве (4) последовательность разбивается на 4 равные части, находится среднее арифметическое каждой части на отдельном процессорном устройстве, полученные значения складываются на одном процессорном устройстве и делятся на 1000	
11.	1,4	Выберите условия реализуемости расписания параллельного алгоритма. (1) на каждой вычислительной единице все операции выполняются одна за другой	

		(2) количество операций, выполняющихся на каждом вычислительном устройстве, постоянно (3) вычислительные устройства, выполняющие разные операции, не могут обмениваться информацией между собой (4) каждая операция выполняется не более чем на одном вычислительном устройстве	
12.	4	Что называется длиной критического пути в графе, представляющем некоторый параллельный алгоритм? (1) длина максимального пути в графе (2) максимальная длина пути в графе, состоящего из однотипных операций (3) средняя высота графа алгоритма (4) диаметр графа алгоритма, определяющий минимально теоретически-возможное время выполнение алгоритма	
13.	1,3,4	Чем характеризуется ускорение параллельного алгоритма? (1) минимальное время выполнения последовательного алгоритма (2) отношение количества процессоров к количеству потоков исполнения (3) минимальное время выполнения параллельного алгоритма (4) размер входных данных	
14.	1	Что такое эффективность параллельного алгоритма? (1) отношение ускорения алгоритма к количеству процессоров (2) произведение минимального времени выполнения параллельного алгоритма и количества процессоров (3) отношение размера входных данных к размеру выходных данных (4) суммарное время, затрачиваемое всеми процессорами на выполнение алгоритма	
15.	2,3	Выберите верные утверждения.	

		(1) ускорение программы с помощью параллельных вычислений зависит только от количества вычислительных узлов (2) суммарное время выполнения параллельной задачи не меньше времени выполнения самого длинного последовательного фрагмента (3) в реальных задачах добавление новых процессоров может увеличивать время расчета (4) в реальных задачах ускорения программы с помощью параллельных вычислений нельзя добиться добавлением вычислительных узлов	
16.	1,3,4	Какие части программы являются последовательными? (1) чтение входных данных с жесткого диска (2) запись выходных данных на несколько жестких дисков (3) синхронизация в параллельной программе (4) критическая секция в параллельной программе	
17.	1,1	Найдите согласно закону Густавсона ускорение масштабирования некоторой параллельной программы, если известно, что время последовательной части программы равно 1000 мс, время части программы, которая может быть распараллелена, равно 100 мс, количество процессоров равно 10. Ответ округлите до десятых.	
18.	1,1	Найдите согласно закону Густавсона ускорение масштабирования некоторой параллельной программы, если известно, что время последовательной части программы равно 800 мс, время части программы, которая может быть распараллелена, равно 100 мс, количество процессоров равно 10. Ответ округлите до десятых.	
19.	3	Что представляет собой task parallelism?	

		(1) комбинация исходных вычислений в более крупную комплексную задачу (2) декомпозиция рекурсивной процедурой исходной задачи на несколько более мелких (3) декомпозиция линейной процедурой исходной задачи на несколько более мелких (4) слияние однотипных задач в более крупную с помощью рекурсивной процедуры	
20.	4	Какой тип вычислительных задач называется embarrassingly parallel? (1) задачи, к которым не применима парадигма "разделяй и властвуй" (2) задачи с большим количеством внутренних связей (3) любые задачи, вычисление которых может быть реализовано с помощью параллельного алгоритма (4) задачи с большим количеством вычислительных подзадач, не имеющих зависимостей между собой	
21.	1,3	В каких случаях для вычислений применяется конвейерная обработка? (1) данные поступают непрерывным односторонним регулярным потоком (2) данные поступают в виде нестабильного двустороннего потока (3) над каждым элементом из набора данных необходимо произвести обработку в несколько стадий (4) данные принимаются однажды и требуют индивидуального подхода к обработке и анализу	
22.	1,4	В каких случаях для вычислений применяется координация на основе событий? (1) данные поступают в виде регулярного одностороннего потока	

		(2) применяемая структура данных характеризуется непредсказуемостью взаимодействия между своими составными частями (3) над каждым элементом из набора данных необходимо произвести обработку в несколько стадий (4) применяется двусторонний поток данных	
23.	1,2,4	Выберите методы решения с локальными взаимодействиями между подзадачами. (1) метод Гаусса-Зейделя (2) метод Red-Black (3) редукция методом "разделяй и властвуй" (4) метод хаотической релаксации	
24.	2,4	В каких случаях для вычислений применяется динамическое планирование с балансировкой нагрузки? (1) число подзадач намного больше числа процессоров (2) количество процессоров в вычислительной системе меняется во времени (3) подзадачи сильно различаются по размерам (4) требуется равномерная загрузка процессоров, но вычислительная система содержит разнородные процессоры	
25.	1,3	В каких случаях для вычислений применяется статическое планирование с балансировкой нагрузки? (1) число подзадач намного больше числа процессоров (2) количество процессоров в вычислительной системе меняется во времени (3) между подзадачами возникают неструктурированные взаимодействия (4) требуется равномерная загрузка процессоров, но вычислительная система содержит разнородные процессоры	

26.	процессы обработки данных, в которых одновременно могут выполняться нескольких машинных операций	Под параллельными вычислениями понимаются ...	
27.	- независимость функционирования отдельных устройств ЭВМ – данное требование относится в равной степени ко всем основным компонентам вычислительной системы – к устройствам ввода-вывода, к обрабатывающим процессорам и к устройствам памяти; избыточность элементов вычислительной системы – организация избыточности может осуществляться в следующих основных формах: использование специализированных устройств таких, например, как отдельных процессоров для целочисленной и вещественной арифметики, устройств многоуровневой памяти (регистры, кэш); дублирование устройств ЭВМ путем использования, например, нескольких однотипных обрабатывающих процессоров или нескольких устройств оперативной памяти.	Достижение параллелизма возможно только при выполнении следующих требований к архитектурным принципам построения вычислительной системы	
28.	это компьютеры, обладающие множеством процессоров, работающих на общей памяти. В этот класс входит большинство продаваемых сегодня на рынке многоядерных компьютеров	Мультипроцессорные вычислительные комплексы - это	
29.	представляют множество компьютеров, соединенных высокоскоростными линиями связи. Каждый компьютер обладает собственной памятью и обменивается сообщениями с другими компьютерами системы для передачи данных. В этот класс входят кластеры. Под кластером понимается вычислительный комплекс, рассматриваемый как единое целое, с некоторым выделенным компьютером, играющим роль	Мультикомпьютерные вычислительные комплексы - ...	

	сервера. Поскольку компьютеры, входящие в состав кластера, могут быть обычными компьютерами, то кластеры относительно недороги. Большинство входящих в Топ 500 суперкомпьютеров являются кластерами		
30.	состоят из множества узлов, каждый из которых может быть мультикомпьютером, мультипроцессором, графическим или векторным процессором. Такие комплексы, как правило, являются суперкомпьютерами	Гибридные вычислительные комплексы - ...	
31.	создание ресурса параллелизма (получение параллельного алгоритма) в процессах решения задач и управление реализацией этого параллелизма с целью достижения наибольшей эффективности использования многопроцессорной вычислительной техники	Задача параллельных вычислений – ...	
32.	группа компьютеров, объединенных в локальную вычислительную сеть и способных работать в качестве единого вычислительного ресурса	Кластер – ...	
33.	технология обработки данных, в которой большая задача распределяется для выполнения между множеством компьютеров, объединенных вычислительной сетью или интернетом	Распределённые вычисления –	
34.	Если при решении некоторой задачи процессоры выполняют одинаковую последовательность вычислений, но используют разные данные, то говорят о параллелизме по данным. Например, при поиске по базе данных каждый процессор может работать со своей частью базы данных. Если процессоры выполняют разные задания одной задачи, выполняют разные функции, то говорят о функциональном параллелизме.	Параллелизм по данным. Функциональный параллелизм.	

35.	Пусть операция разбита на микрооперации. Расположим микрооперации в порядке выполнения и для каждого выполнения выделим отдельную часть устройства. В первый момент времени входные данные поступают для обработки в первую часть. После выполнения первой микрооперации первая часть передает результаты своей работы второй части, а сама берет новые данные. Когда входные аргументы пройдут все этапы обработки, на выходе устройства появится результат выполнения операции. Таким образом, реализуется функциональный параллелизм. Каждая часть устройства называется ступенью конвейера, а общее число ступеней – длиной конвейера.	Конвейерная обработка данных операции (конвейерный параллелизм на уровне операции)	
36.	Классификация основана на понятии потока, под которым понимается последовательность команд или данных, обрабатываемых процессором. На основе числа потоков команд и потоков данных выделяют четыре класса архитектур: - SISD (Single Instruction stream/Single Data stream) – один поток команд и один поток данных; - SIMD (Single Instruction stream/Multiple Data stream) - один поток команд и множество потоков данных; - MISD (Multiple Instruction stream/Single Data stream) - множество потоков команд и один поток данных; - MIMD (Multiple Instruction stream/Multiple Data stream) - множество потоков команд и множество потоков данных.	Классификация М. Флинна	
37.	состоят из совокупности процессоров, имеющих разделяемую общую память с единым адресным пространством и функционирующих под управлением одной операционной системы. Недостаток - число процессоров, имеющих доступ к общей памяти, нельзя сделать большим. Существует предел наращивания числа процессоров, превышение	Симметричные мультипроцессоры (SMP)	

	которого ведет к быстрому росту потерь на межпроцессорный обмен данными.		
38.	Основная идея классификации состоит в том, что множественный поток команд может быть обработан либо по конвейерной схеме в режиме разделения времени, либо каждый поток обрабатывается своим устройством. В соответствии с этим различают следующие MIMD - компьютеры: - конвейерные; - переключаемые (с общей памятью и распределенной памятью); - сети, реализованные в виде: регулярной решетки, гиперкуба, иерархической структуры и изменяемой конфигурации.	Систематизация MIMD-компьютеров Р. Хокни	
39.	Классификация Т. Фенга позволяет также строить оценки достижимой степени параллелизма. Она основана на двух характеристиках: - число n бит в машинном слове, обрабатываемых параллельно; - число слов m, обрабатываемых одновременно вычислительной системой.	Классификация Т. Фенга	
40.	В основе этой классификации явное описание параллельной и конвейерной обработки. При этом различают три уровня обработки данных: - уровень выполнения программы; - уровень выполнения команд; - уровень битовой обработки.	Классификация В. Хендлера	
41.	1.	Если полагать, что единственным общим ресурсом является оперативная память, то какие проблемы характерны для параллельных вычислений, выполняемых на системах с общей памятью: 1. гонка данных (DataRace)	

		2. распределить – собрать (Map – Reduce)	
42.	1.	Если полагать, что единственным общим ресурсом является оперативная память, то какие проблемы характерны для параллельных вычислений, выполняемых на системах с распределенной памятью (кластеров): 1. синхронизация 2. гонка данных (DataRace) 3. клинч 4. блокировка	
43.	1.	Если полагать, что единственным общим ресурсом является оперативная память, то что понимается под гонкой данных? 1. конкурирование потоков при записи данных в одно и то же слово памяти компьютера 2. чистка памяти от данных 3. конкурирование компьютеров за быструю запись данных в память компьютера 4. конкурирование потоков при записи данных в разные слова памяти компьютера	
44.	1.	Что такое блокировка в параллельных вычислениях? 1. блокирование критической секции программы с целью исключения возможности ее одновременного выполнения в параллельных процессах	

		2. блокирование области памяти с целью исключения возможности ее использования в параллельных процессах 3. блокирование программы с целью исключения возможности ее использования в параллельных процессах 4. блокирование программы за исключением критических секций для получения возможности одновременного выполнения критических секций в нескольких параллельных процессах	
45.	1.	Векторные процессоры согласно классификации Флинна относятся к классу: 1. SIMD 2. SISD 3. MISD 4. MIMD	
46.	1.	. Из закона Амдаля следует: 1. для некоторой задачи за счет увеличения числа используемых процессоров нельзя добиться 10-кратного ускорения 2. для любой задачи можно добиться сколь угодно высокого ускорения за счет увеличения числа используемых процессоров 3. для любой задачи за счет увеличения числа используемых процессоров можно добиться более чем 10-кратного ускорения	

47.	1.	Разбиение исходной задачи на подзадачи меньшей размерности: 1. зачастую приводит к уменьшению общего времени работы как в последовательных, так и в параллельных алгоритмах 2. возможно только в последовательных алгоритмах 3. возможно только в параллельных алгоритмах 4. всегда приводит к увеличению общего времени работы 5. всегда приводит к уменьшению общего времени работы	
48.	1.	Укажите истинные высказывания: 1. для мультимикрокомпьютерных комплексов, где каждый компьютер комплекса имеет собственную память, проблемы гонки данных и клинча не столь актуальны 2. гонка данных всегда приводит к возникновению исключительных ситуаций 3. клинч всегда приводит к возникновению исключительных ситуаций	
49.	1.	Под параллельными вычислениями в данном курсе понимается параллельное выполнение: 1. одной и той же программы 2. разных программ на одном компьютере 3. разных программ на разных компьютерах	

		4. одной программы в разных сеансах выполнения	
50.	1.	Что такое мультипрограммный режим работы? 1. параллельное выполнение разных программ 2. параллельное выполнение одной программы 3. выполнение приложений мультимедиа 4. параллельная разработка нескольких программ	
51.	1.	Приоритет, счетчик команд, стек могут принадлежать: 1. Потoku 2. Процессу 3. Набору инструкций 4. Программисту	
52.	1. Декомпозиция задачи на подзадачи, которые реализуются независимо. 2. Определение для сформированного набора подзадач информационных взаимодействий. 3. Масштабирование подзадач, определение количества процессоров. 4. Определение архитектуры системы, закрепление подзадач за процессорами, составление расписания.	Общая схема этапов разработки параллельных алгоритмов	
53.	1. Обнаружение в программе вычислений, реализацию которых можно осуществлять параллельно. 2. Последующее распределение данных по модулям локальной памяти процессоров. 3. Последующее согласование параллелизма вычислений и распределения данных.	Многопоточность предполагает:	

54.	Данный вид параллелизма реализуется компилятором и заключается в замене множества выполнений однотипных операций одной операцией над множеством данных. Обработка осуществляется на векторных процессорах. В программах, обрабатываемых компилятором, задаются директивы компиляции, что предполагает использования языков параллельных вычислений.	Параллелизм на уровне данных	
55.	Данный вид параллелизма предполагает замену последовательных алгоритмов некоторых вычислений на параллельные. Это касается алгоритмов поиска, сортировки и т.п. Организация процесса распараллеливания осуществляется за счет использования различных средств параллельного программирования, таких как, специальные библиотеки, переменные окружения, директивы компилятора и т.п. Примером может служить технология OpenMP	Параллелизм на уровне алгоритмов	
56.	- минимизировать конфликты в памяти; - оптимизировать распределение вычислительной нагрузки между процессорами; - оптимизировать распределение данных; - оптимизировать структуру программы; - заменить последовательные части алгоритма на параллельные; - решать проблемы, характерные для последовательных компьютеров и традиционного последовательного программирования.	Для повышения производительности программы можно:	
57.	флопс определяется количеством арифметических операций в секунду, выполненных над вещественными числами с плавающей точкой	Единица производительности параллельного компьютера -	
58.	создание ресурса параллелизма (получение параллельного алгоритма) в процессах решения	Задача параллельных вычислений	

	задач и управление реализацией этого параллелизма с целью достижения наибольшей эффективности использования многопроцессорной вычислительной техники		
59.	способ решения трудоёмких вычислительных задач с использованием нескольких компьютеров, объединённых в параллельную вычислительную систему. Слабосвязанные, гетерогенные вычислительные системы выделяют в отдельный класс распределённых систем – Grid	Распределённые вычисления	
60.	алгоритм, операции которого могут выполняться одновременно (не обязательно независимо); подразумевается, что в явном или неявном виде указаны одновременно выполняемые операции или множества операций. Строгое понятие параллельного алгоритма не введено	Параллельный алгоритм	
61.	параллельный алгоритм, записанный в некоторой системе программирования, ориентированной на вычислительные системы параллельной архитектуры	Параллельная программа	
62.	Если при решении некоторой задачи процессоры выполняют одинаковую последовательность вычислений, но используют разные данные, то говорят о параллелизме по данным. Например, при поиске по базе данных каждый процессор может работать со своей частью базы данных. Если процессоры выполняют разные задания одной задачи, выполняют разные функции, то говорят о функциональном параллелизме.	Параллелизм по данным. Функциональный параллелизм	
63.	Пусть операция разбита на микрооперации. Расположим микрооперации в порядке выполнения и для каждого выполнения выделим отдельную часть устройства. В первый момент времени	Конвейерная обработка данных операции (конвейерный параллелизм на уровне операции)	

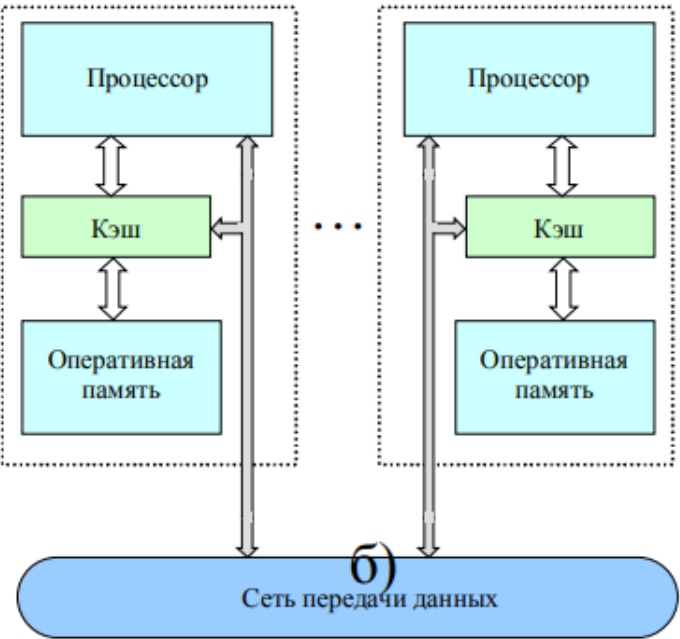
	входные данные поступают для обработки в первую часть. После выполнения первой микрооперации первая часть передает результаты своей работы второй части, а сама берет новые данные. Когда входные аргументы пройдут все этапы обработки, на выходе устройства появится результат выполнения операции. Таким образом, реализуется функциональный параллелизм. Каждая часть устройства называется ступенью конвейера, а общее число ступеней – длиной конвейера		
64.	количество операций, реально выполняемых в среднем за единицу времени.	Реальная производительность вычислительной системы	
65.	максимальное количество операций, которое может быть выполнено системой за единицу времени.	Пиковая производительность вычислительной системы	
66.	отношение времени реальной работы процессора на данном отрезке к длине всего отрезка. Загруженность вычислительной системы, состоящей из одинаковых процессоров – среднее арифметическое загруженностей всех процессоров. Из определения следует, что загруженность p удовлетворяет условию $0 \leq p \leq 1$	загруженность процессора на данном отрезке времени	
67.	отношение времени выполнения алгоритма на одном процессоре (на одном ядре процессора) ко времени параллельного выполнения. Ускорение зависит от выбора параллельного алгоритма и от того, насколько этот алгоритм адекватен архитектуре вычислительной системы.	Ускорение реализации алгоритма на вычислительной системе, состоящей из одинаковых процессоров –	
68.	системы с общей разделяемой памятью (но каждый процессор имеет и свою кэш-память).	Мультипроцессоры	
69.	системы с распределенной памятью. Когда память физически распределена между процессорами.	Мультимикрокомпьютеры	

70.	В многопроцессорной системе необходимо координировать обмен данными между процессорами, а также, при использовании общей памяти, между процессорами и общей памятью. Синхронизация – это согласование по времени выполнения параллельных заданий. Она включает в себя ожидание того, что выполнение задачи достигнет особой точки, называемой точкой синхронизации. После того, как все задания достигнут точки синхронизации, выполнение заданий может быть продолжено до следующей точки синхронизации. При использовании модели передачи сообщений (обычно MPI – Message Passing Interface) работу процессоров синхронизируют функции обмена данными.	Синхронизация (работы процессоров).	
71.	Под зерном вычислений (или тайлом) понимается множество операций алгоритма, выполняемых атомарно: вычисления, принадлежащие одному зерну, не могут прерываться синхронизацией или обменом данными, требуемыми для выполнения этих операций. Зернистость – это мера отношения количества вычислений, сделанных в параллельной задаче, к количеству пересылок данных. Мелкозернистый параллелизм – очень мало вычислений на каждую пересылку данных. Крупнозернистый параллелизм – интенсивные вычисления на каждую пересылку данных (данные пересылаются большими порциями). Чем мельче зернистость, тем больше точек синхронизации.	Зернистость	
72.	создаваемый операционной системой объект внутри процесса (процесс – выполняемое приложение с собственным виртуальным адресным пространством), который выполняет инструкции	Поток, нить (thread), легковесный процесс	

	программы. Процесс может иметь один или несколько потоков, выполняемых в контексте данного процесса. Потоки (нити) позволяют осуществлять параллельное выполнение процессов и одновременное выполнение одним процессом различных частей программы на различных процессорах		
73.	Алгоритм обладает естественным параллелизмом, если его можно разбить на независимо выполняемые части	Естественный параллелизм	
74.	Алгоритм обладает внутренним параллелизмом, если его можно разбить на параллельно (но не обязательно независимо) выполняемые части	Внутренний параллелизм	
75.	Выявление (указание) операций или множеств операций последовательного алгоритма, которые могут выполняться одновременно	Распараллеливание	
76.	Распараллеливание, осуществляемое до начала выполнения алгоритма (программы)	Статическое распараллеливание	
	Распараллеливание, осуществляемое во время выполнения алгоритма (программы)	Динамическое распараллеливание	
77.	Масштабируемость данного параллельного алгоритма при реализации на данной параллельной системе означает, что производительность системы пропорциональна числу содержащихся в ней процессоров (ядер). Различают понятия слабой и сильной масштабируемости. Слабая (отнюдь не в смысле «плохая») масштабируемость (weak scaling) означает линейный рост размера задачи с ростом числа процессоров при фиксированном времени выполнения алгоритма. Сильная масштабируемость (strong scaling) означает линейный рост ускорения с ростом числа процессоров при фиксированном размере задачи.	Масштабируемость	

78.	это стратегия программирования, откладывающая вычисление значения выражения или переменной до тех пор, пока это значение не понадобится. Это антипод прямого, или немедленного вычисления, при котором значения вычисляются сразу, когда вызывается расчет. Отложенные вычисления позволяют повысить уровень оптимизации вычислений и сократить нагрузку на память, избегая ненужных повторений расчетов.	Отложенное вычисление значения (lazy evaluation)	
79.	ОрепМР - это стандарт параллельного программирования в среде с общей памятью. ОрепМР предоставляет программистам набор прагм, процедур и переменных среды, позволяющих легко параллелизовать код, написанный на Фортране, С или С++. Прагма - это директива компилятора, указывающая, как обрабатывать код, следующий за прагмой. Когда прагмы ОрепМР используются в программе, они дают указание компилятору, поддерживающему ОрепМР, создать исполнимый модуль, который будет выполняться параллельно с использованием нескольких потоков. Прагмы ОрепМР позволяют использовать единообразный и переносимый интерфейс для параллелизации программ на различных архитектурах и системах	Стандарт параллельного программирования ОрепМР	
80.	Message Passing Interface (MPI) - программный интерфейс для передачи информации, который позволяет процессам, выполняющим одну задачу, обмениваться сообщениями. Стандарт MPI ориентирован на системы с распределенной памятью, когда затраты на передачу данных велики, в то время как ОрепМР ориентирован на системы с общей памятью (для	Интерфейс передачи сообщений MPI	

	многоядерных процессоров с общей кэш-памятью). Обе технологии могут использоваться совместно для оптимизации использования многоядерных систем в одном кластере.		
81.	Архитектура многопроцессорных систем с общей (разделяемой) памятью – системы с однородным доступом к памяти	<pre> graph TD subgraph Unit1 [] P1[Процессор] <--> C1[Кэш] end subgraph Unit2 [] P2[Процессор] <--> C2[Кэш] end Unit1 -.- Unit2 C1 <--> MP[Оперативная память] C2 <--> MP </pre>	

82.	Архитектура многопроцессорных систем с общей (разделяемой) памятью – системы с неоднородным доступом к памяти		
83.	1) осуществляется управление работой множества процессов; 2) организуется обмен данными между процессами; 3) утрачивается детерминизм поведения из-за асинхронности доступа к данным; 4) преобладают нелокальные и динамические ошибки; 5) появляется возможность тупиковых ситуаций; 6) возникают проблемы масштабируемости программы и балансировки загрузки вычислительных узлов.	Какими особенностями обладают параллельные программы	
84.	- Переносимость – MPI позволяет в значительной степени снизить остроту проблемы переносимости параллельных программ между разными	Преимущества MPI	

	компьютерными системами – параллельная программа, разработанная на алгоритмическом языке С или Fortran с использованием библиотеки MPI, как правило, будет работать на любых вычислительных платформах, для которых имеется реализация стандарта. - Повышение эффективности – MPI содействует повышению эффективности параллельных вычислений, поскольку в настоящее время практически для каждого типа вычислительных систем существуют реализации стандарта, в максимальной степени учитывающие возможности используемого компьютерного оборудования.		
85.	- Простота использования – разработчик не создает новую параллельную программу, а добавляет в текст последовательной программы необходимые директивы и, возможно, вызовы библиотечных функций, указывающие компилятору способы распределения вычислений и данных между потоками, а также доступа к ним. Главной идеей OpenMP является распараллеливание циклов, обычно несущих основную вычислительную нагрузку. - Гибкость – OpenMP предоставляет разработчику достаточно большие возможности контроля над поведением параллельной программы. - Повторное использование – OpenMP-программа во многих случаях может быть использована как обычная последовательная, если необходимо обеспечить ее выполнение на однопроцессорной платформе. При этом, чтобы избавиться от реализации OpenMP в исполняемом модуле, достаточно пересобрать его последовательным	Преимущества OpenMP	

	компилятором. Директивы OpenMP будут проигнорированы, а вызовы функций библиотеки могут быть заменены на заглушки, текст которых приведен в спецификациях стандарта		
86.	4	Средствами C# можно создавать: 1. Процессы 2. Потоки 3. Ни то, ни другое 4. Процессы и потоки	
87.	1.	Среда CLR не остановит приложение (что означает выгрузку текущего домена приложения) до тех пор, пока: 1. не будут завершены все потоки переднего плана 2. не будут завершены вообще все потоки 3. не будет завершен основной поток приложения 4. не будет завершен GUI поток приложения	
88.	1.	За планирование потоков к запуску отвечает 1. Операционная система 2. Программист 3. Среда CLR 4. Систему управления потоками приложения	
89.	1.	Операционная система планирует выполнение потоков на основе 1. Приоритетов 2. Задач 3. Инструкций 4. Загрузки ЦП	
90.	1.	Выберите верное описание класса Task (TPL). 1. Данный класс описывает отдельную задачу, которая запускается асинхронно в одном из потоков из пула потоков	

		2. Представляет собой независимую последовательность инструкций, существующую в рамках процесса 3. Представляет собой последовательность инструкций которой выделяется определённое адресное пространство в памяти и которая изолирована от других процессов	
91.	1.	Метод Parallel.For 1. Позволяет выполнять итерации цикла параллельно 2. Позволяет выполнять итерации цикла последовательно 3. Используется для соответствия стандарту MPI 4. Не существует в C#	
92.	1.	Параллелизм - это 1. Способ обработки множественных запросов одновременно 2. Разновидность многопоточности 3. Разновидность многозадачности	
93.	1.	Среда CLR не остановит приложение (что означает выгрузку текущего домена приложения) до тех пор, пока: 1. не будут завершены все потоки переднего плана 2. не будут завершены вообще все потоки 3. не будет завершен основной поток приложения 4. не будет завершен GUI поток приложения	
94.	1.	Единственное отличие приоритетных потоков от фоновых заключается в том, что: 1. Фоновый поток автоматически завершается, если в его процессе остановлены все приоритетные потоки 2. Отличий нет 3. Фоновые потоки имеют меньший приоритет 4. Приоритетные потоки имеют больший приоритет	

95.	1.	По умолчанию создаваемый поток (.NET Framework) автоматически становится: 1. Приоритетным 2. Фоновым 3. Рабочим 4. Основным	
96.	1,2	Какие разновидности потоков определены в среде .NET Framework? 1. Приоритетный 2. Фоновый 3. Рабочий 3. Основной 4. Главный	
97.	1,2,3,4,5	В каких состояниях может находиться поток? 1. Выполняется 2. Готов к выполнению 3. Приостановлен 4. Заблокирован 5. Завершен 6. Ожидание события 7. Ожидание ввода	
98.	1.	Для чего организуется многозадачность на основе потоков? 1. Параллельного выполнения отдельных частей одной программы 2. Параллельного выполнения программ 3. Параллельного выполнения инструкций 4. Нет верного ответа	
99.	1.	Для чего организуется многозадачность на основе процессов? 1. Параллельного выполнения программ 2. Параллельного выполнения потоков 3. Параллельного выполнения инструкций	

		4. Все ответы верные	
100.	1,2	Какие разновидности многозадачности различают? 1. Многозадачность на основе процессов 2. Многозадачность на основе потоков 3. Многозадачность на основе инструкций 4. Многозадачность на основе команд	