

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Ефанов Алексей Валерьевич

Должность: Директор Невиномысского технологического института (филиал) СКФУ

Дата подписания: 12.10.2021 09:19:00

Уникальный программный идентификатор: 49214306dd433e7a1b0f8632f645f9d53c99e3d0

Министерство науки и высшего образования российской федерации

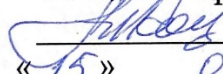
Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

И. о. зав. кафедрой ИСЭиА

 Колдаев А.И.

«15» 03 2021 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

для проведения текущего контроля успеваемости и промежуточной аттестации по

дисциплине:

«Технологии программирования»

(ЭЛЕКТРОННЫЙ ДОКУМЕНТ)

Направление подготовки 09.03.02 Информационные системы и технологии

Профиль Информационные системы и технологии в бизнесе

Квалификация выпускника бакалавр

Форма обучения заочная

Год начала обучения 2021

Изучается на 3 курсе

Предисловие

1. Назначение: для проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине «Технологии программирования» для студентов направления подготовки 09.03.02 Информационные системы и технологии.
2. Фонд оценочных средств текущего контроля успеваемости и промежуточной аттестации на основе рабочей программы дисциплины «Технологии программирования» в соответствии с образовательной программой по направлению подготовки 09.03.02 Информационные системы и технологии, утвержденной на заседании Учёного совета НТИ (филиал) СКФУ.
3. Разработчик: Кочеров Ю. Н. канд., техн., наук, доцент базовой кафедры регионального индустриального парка
4. ФОС рассмотрен и утвержден на заседании базовой кафедры регионального индустриального парка.
5. ФОС согласован с выпускающей кафедрой информационных систем, электропривода и автоматики.
6. Проведена экспертиза ФОС. Члены экспертной группы, проводившие внутреннюю экспертизу:

Председатель: Кузьменко В.В., и.о. директора НТИ (филиал) СКФУ, профессор кафедры гуманитарных и математических дисциплин

Члены экспертной группы:

Должикова М.В. – заместитель директора по учебно-воспитательной работе НТИ (филиал) СКФУ;

Колдаев А.И. – доцент кафедры информационных систем, электропривода и автоматики.

Эксперт, проводивший внешнюю экспертизу:

Остапенко Н.А., – кандидат технических наук, ведущий инженер-конструктор КБ модернизации ООО КИЭП «Энергомера» филиал АО «Электротехнические заводы «Энергомера»

7. Экспертное заключение: фонд оценочных средств отвечают основным требованиям федерального государственного образовательного стандарта и способствует формированию требуемых компетенций.

Срок действия ФОС: на срок реализации образовательной программы.

Паспорт фонда оценочных средств
для проведения текущего контроля успеваемости и промежуточной аттестации
дисциплине:
«Технологии программирования»

Направление подготовки	09.03.02 Информационные системы и технологии
Направленность (профиль)	Информационные системы и технологии в бизнесе
Квалификация выпускника	Бакалавр
Форма обучения	заочная
Год начала обучения	2021 г.
Изучается	На 3 курсе

Код оцениваемой компетенции	Этап формирования компетенции (№ темы)	Средства и технологии оценки	Вид контроля, аттестация	Тип контроля	Наименование оценочного средства
ОПК-5 ОПК-6	Тема 1-11	Собеседование	Устный	Текущий	Вопросы для собеседования
ОПК-5 ОПК-6	Тема 1-11	Собеседование	Устный	Промежуточный	Вопросы к экзамену

Составитель Кочеров Ю.Н.

Министерство науки и высшего образования российской федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

**Вопросы для собеседования
по дисциплине технологии программирования**

3 курс летняя сессия

Пороговый уровень

Тема 1. Обзор и установка Visual Studio 2019 Community на Windows 10

1. Интегрированная среда разработки (IDE)

Тема 2. Структуры данных, разработка алгоритмов и программ работы в различных структурами данных

1. Что понимается под структурой данных? Что определяет мощность структуры? Какие модели структур данных используются при разработке программ?

2. Что считается базовым типом структуры? Какие основные операции выполняются над структурами данных?

3. Какая информация содержится в дескрипторе структуры и дискриминанте ее элемента?

Тема 3. Информационные технологии, техника, прикладные программные средства для проектирования линейных структур данных

1. Какие структуры данных считаются линейными? Какие основные операции выполняются над ними? Какие существуют линейные структуры с особыми правилами доступа?

2. По каким принципам строятся линейные статические структуры? Что входит в состав их элементов? В чем различаются регулярные и комбинированные структуры?

3. По каким принципам строятся линейные динамические структуры? Что входит в состав их элементов?

4. Каково назначение массивов? По каким принципам они строятся? В чем их достоинства и недостатки?

5. Как массивы размещаются в памяти? Что понимается под векторизацией массивов? Как производится доступ к их отдельному элементу?

6. Как организуются и по каким правилам обрабатываются последовательные стеки и очереди?

7. В чем особенности строковых структур?

8. Каково назначение комбинированных структур? По каким принципам они строятся? В чем их достоинства и недостатки?

Тема 4. Решение стандартных задач профессиональной деятельности с применением иерархических структур данных

1. Какие структуры данных считаются иерархическими? Какие основные операции выполняются над ними?

2. По какому принципу организуются структуры данных типа «дерево»? Какие основные понятия теории деревьев?

3. Что понимается под поиском по дереву? Какие существуют виды поиска по дереву с произвольным ветвлением?

4. Как рекурсивно определяется бинарное дерево? Какие би-нарные деревья считаются подобными, какие — эквивалентными? Какие бинарные деревья считаются полными, какие — завершёнными, какие — вырожденными?
5. Какие существуют порядки обхода бинарного дерева?
6. Как строится бинарное дерево с использованием технологии связного распределения памяти?
7. Как строится бинарное дерево с обратными связями с использованием технологии связного распределения памяти?
8. Как в виде бинарного дерева представляются деревья с произвольной структурой?

Тема 5. Анализ исходной информации для проектирования систем автоматизации с применением многосвязных структур данных

1. Какие структуры данных считаются многосвязными? Как определяется граф? В чем отличие ориентированного графа от не-ориентированного? Как можно выполнить их взаимное преобразование?
2. Что считается путем в графе? Как определяется длина пути? Какой путь является простым?
3. Что считается циклом в графе? Какой цикл является простым?
4. Что называется подграфом, суграфом, надграфом, сверхграфом, дополнением?
5. Какие графы считаются изоморфными?
6. Какие графы считаются плоскими?
7. Какие графы считаются тривиальными?
8. Какие графы считаются полными?
9. Какие графы считаются двудольными?
10. Какие графы обозначаются S_k ?
11. Как представить граф с помощью матриц инцидентности и смежности?
12. Как представить граф с помощью списков инцидентности и смежности?
13. Как осуществляется поиск по графу в глубину?

Тема 6. Реализация структур данных средствами языков программирования

1. Как объявляется массив в языке Pascal? Каковы ограничения на размер массива? Как организуются многомерные массивы? Как задать значения элементов массива через блок констант?
2. Как осуществляется доступ к элементу массива? Какие действия можно производить над отдельным элементом массива, а какие — над массивом в целом? Контролируется ли принадлежность индексов допустимым диапазонам?
3. Как осуществляется передача массивов в процедуры и функции языка Pascal? Можно ли возвращать массивы как результаты работы функций?
4. Как организованы строки в языке Pascal? Как они объявляются? В каких операциях строки могут принимать участие? Какие функции могут использоваться при работе со строками?
5. Как объявляется запись в языке Pascal? Как задать значения полям записи через блок констант?
6. Как осуществляется доступ к полю записи? Что дает присоединение общего имени к полям записи?
7. Какие действия можно производить над отдельным полем записи, а какие — над записью в целом?
8. Как осуществляется передача записей в процедуры и функции языка Pascal? Можно ли возвращать записи как результаты работы функций?
9. Для чего используются записи с вариантами? Как они объявляются? Как осуществляется контроль работы с вариантами?
10. Как объявляется множество в языке Pascal? Каковы ограничения на размер множества? Как задать значения элементов множества через блок констант?

11. Какие операции можно производить над множествами? Как осуществляется доступ к отдельному элементу множества?
12. Как осуществляется передача множеств в процедуры и функции языка Pascal? Можно ли возвращать множества как результаты работы функций?
13. Как организуются и обрабатываются линейные связные списки в языке Pascal?
14. Как организуются и обрабатываются деревья в языке Pascal?
15. Как организуются и обрабатываются графы в языке Pascal?

Повышенный уровень

Тема 1. Обзор и установка Visual Studio 2019 Community на Windows 10

1. Средства повышения производительности

Тема 2. Структуры данных, разработка алгоритмов и программ работы в различными структурами данных

1. Какие структуры данных считаются статическими? Как и ко-гда выполняется их размещение в машинной памяти? Каковы досто-инства и недостатки статических структур?
2. Какие структуры данных считаются динамическими? Как и когда выполняется их размещение в машинной памяти? Каковы до-стоинства и недостатки динамических структур?
3. Каковы основные свойства линейных, иерархических и мно-госвязных структур?

Тема 3. Информационные технологии, техника, прикладные программные средства для проектирования линейных структур данных

1. Как комбинированные структуры размещаются в памяти? Как производится доступ к их отдельному элементу?
2. Каково назначение множеств? По каким принципам они строятся? В чем их достоинства и недостатки? Какие существуют способы задания множеств? Каковы основные закономерности тео-рии множеств?
3. По каким принципам на базе одномерного массива строится множество? Каков порядок выполнения операций включения и ис-ключения его элементов?
4. Какие действия выполняются над множествами как структу-рами данных? Каков порядок выполнения пересечения, объединения и вычитания множеств?
5. Каково назначение списков? По каким принципам они стро-ятся? В чем их достоинства и недостатки?
6. Как списки размещаются в памяти? Как производится до-ступ к их отдельному элементу?
7. Как организуются и по каким правилам обрабатываются од-носвязные списки?
8. Как организуются и по каким правилам обрабатываются связные стеки и очереди?
9. Как организуются и по каким правилам обрабатываются од-носвязные циклические списки?
10. Как организуются и по каким правилам обрабатываются двухсвязные списки?
11. В чем особенности двухсвязного списка с ограничителем? В чем его преимущества?

Тема 4. Решение стандартных задач профессиональной деятельности с применением иерархических структур данных

1. Для чего бинарное дерево дополняется ограничителем?
2. Какое бинарное дерево называется бинарным деревом пойс-ка? Что понимается под его упорядоченностью?
3. По каким правилам выполняются основные операции над бинарными деревьями пойска?
4. Какие бинарные деревья пойска называются «прошитыми»? В чем назначение «прошивки» деревьев?

5. Для чего необходимо балансировать бинарное дерево поиска? Какое бинарное дерево поиска называется оптимальным, какое — идеально сбалансированным, какое — сбалансированным по AVL?
6. Когда нарушается сбалансированность по AVL при вставке вершин в дерево?
7. Когда нарушается сбалансированность по AVL при удалении вершин из дерева?
8. Как восстанавливается сбалансированность AVL-дерева? Какие виды вращений существуют и когда они применяются?
9. В чем заключаются достоинства и недостатки AVL-деревьев?

Тема 5. Анализ исходной информации для проектирования систем автоматизации с применением многосвязных структур данных

1. Как осуществляется поиск по графу в ширину?
2. Что понимается под связностью ориентированных и неориентированных графов? В чем отличие компонент сильной, односторонней и слабой связности?
3. Как определить число компонент связности графа?
4. Что называется стягивающими деревьями графа? Как они строятся?
5. Что называется экстремальными стягивающими деревьями графа? Как они строятся?
6. Что считается фундаментальными циклами графа? Как находится их множество? Что такое цикломатическое число графа?
7. Какой цикл в графе считается эйлеровым? Каковы условия его существования? Как он строится? Что считается эйлеровым путем в графе?
8. Какой цикл в графе считается гамильтоновым? Как он строится? Что считается гамильтоновым путем в графе?
9. По какому принципу отыскиваются кратчайшие пути между фиксированными вершинами графа?
10. Как работает алгоритм Форда-Беллмана поиска кратчайшего пути между фиксированными вершинами графа?
11. Как работает алгоритм Дейкстры поиска кратчайшего пути между фиксированными вершинами графа?
12. Как работает алгоритм Флойда-Уоршалла поиска кратчайших путей между всеми парами вершин графа?
13. Что считается транзитивным замыканием графа? Как оно строится?

Тема 6. Реализация структур данных средствами языков программирования

1. Как объявляется массив в языке C++? Каковы ограничения на размер массива? Как организуются многомерные массивы? Как задать значения элементов массива через список инициализации?
2. Как осуществляется доступ к элементу массива? Какие действия можно производить над отдельным элементом массива? Контролируется ли принадлежность индексов допустимым диапазонам?
3. Как осуществляется передача массивов в функции языка C++? Можно ли возвращать массивы как результаты работы функций?
4. Как работать с массивами с помощью указателей?
5. Как организованы строки в языке C++? Как они объявляются? В каких операциях строки могут принимать участие? Какие функции могут использоваться при работе со строками?
6. Как объявляется структура в языке C++? Как задать значения полям структуры через список инициализации?
7. Как осуществляется доступ к полю структуры? Как получить доступ к полю структуры с помощью указателя?
8. Какие действия можно производить над отдельным полем структуры, а какие — над структурой в целом?

9. Как осуществляется передача структур в функции языка C++? Можно ли возвращать структуры как результаты работы функций?
10. Для чего используются «битовые» поля структур? Как с ними работать?
11. Для чего используются объединения? Как они объявляются?
12. Как решаются задачи теории множеств в языке C++?
13. Как организуются и обрабатываются линейные связные списки в языке C++?
14. Как организуются и обрабатываются деревья в языке C++?
15. Как организуются и обрабатываются графы в языке C++?

3 курс зимняя сессия

Пороговый уровень

Тема 7. Принципы построения параллельных вычислительных систем

1. В чем заключаются основные способы достижения параллелизма?
2. В чем могут состоять различия параллельных вычислительных систем?
3. Что положено в основу классификации Флинна?
4. В чем состоит принцип разделения многопроцессорных систем на мультипроцессоры и мультимикросистемы?
5. Какие классы систем известны для мультипроцессоров?
6. В чем состоят положительные и отрицательные стороны симметричных мультипроцессоров?

Тема 8. Моделирование и анализ параллельных вычислений

1. Как определяется модель "операции — операнды"?
2. Как определяется расписание для распределения вычислений между процессорами?
3. Как определяется время выполнения параллельного алгоритма?
4. Какое расписание является оптимальным?
5. Как определить минимально возможное время решения задачи?
6. Что понимается под паракомпьютером и для чего может оказаться полезным данное понятие?
7. Какие оценки следует использовать в качестве характеристики времени последовательного решения задачи?
8. Как определить минимально возможное время параллельного решения задачи по графу "операнды - операции"?
9. Какие зависимости могут быть получены для времени параллельного решения задачи при увеличении или уменьшении числа используемых процессоров?
10. При каком числе процессоров могут быть получены времена выполнения параллельного алгоритма, сопоставимые по порядку с оценками минимально возможного времени решения задачи?
11. Как определяются понятия ускорения и эффективности?
12. Возможно ли достижение сверхлинейного ускорения?

Тема 9. Оценка коммуникационной трудоемкости параллельных алгоритмов

1. Какие основные характеристики используются для оценки топологии сети передачи данных? Приведите значения характеристик для конкретных типов коммуникационных структур (полный граф, линейка, решетка и др.).
2. Какие основные методы применяются при маршрутизации передаваемых данных по сети?
3. В чем состоят основные методы передачи данных? Приведите для этих методов аналитические оценки времени выполнения.
4. Какие операции передачи данных могут быть выделены в качестве основных?
5. В чем состоят алгоритмы выполнения передачи данных от одного процессора всем процессорам сети для топологий кольца, решетки и гиперкуба? Приведите оценки временной трудоемкости для этих алгоритмов.

Тема 10. Принципы разработки параллельных методов

1. В чем состоят исходные предположения для возможности применения рассмотренной в лекции методики разработки параллельных алгоритмов?
2. Каковы основные этапы проектирования и разработки методов параллельных вычислений?
3. Как определяется модель "подзадачи - сообщения"?
4. Как определяется модель "процессы - каналы"?
5. Какие основные требования должны быть обеспечены при разработке параллельных алгоритмов?
6. В чем состоят основные действия на этапе выделения подзадач?

Тема 11. Параллельное программирование на основе MPI

1. Какой минимальный набор средств является достаточным для организации параллельных вычислений в системах с распределенной памятью?
2. В чем состоит важность стандартизации средств передачи сообщений?
3. Что следует понимать под параллельной программой?
4. В чем различие понятий процесса и процессора?
5. Какой минимальный набор функций MPI позволяет начать разработку параллельных программ?
6. Как описываются передаваемые сообщения?
7. Как можно организовать прием сообщений от конкретных процессов?
8. Как определить время выполнения MPI -программы?
9. В чем различие парных и коллективных операций передачи данных?
10. Какая функция MPI обеспечивает передачу данных от одного процесса всем процессам?
11. Что понимается под операцией редукции?
12. В каких ситуациях следует применять барьерную синхронизацию?
13. Какие режимы передачи данных поддерживаются в MPI?

Повышенный уровень

Тема 7. Принципы построения параллельных вычислительных систем

1. Какие классы систем известны для мультимикрокомпьютеров?
2. Чем состоят положительные и отрицательные стороны кластерных систем?
3. Какие топологии сетей передачи данных наиболее широко используются при построении многопроцессорных систем?
4. В чем состоят особенности сетей передачи данных для кластеров?
5. Каковы основные характеристики сетей передачи данных?
6. Какие системные платформы могут быть использованы для построения кластеров?

Тема 8. Моделирование и анализ параллельных вычислений

1. В чем состоит противоречивость показателей ускорения и эффективности?
2. Как определяется понятие стоимости вычислений?
3. В чем состоит понятие стоимостно-оптимального алгоритма?
4. В чем заключается проблема распараллеливания последовательного алгоритма суммирования числовых значений?
5. В чем состоит каскадная схема суммирования? С какой целью рассматривается модифицированный вариант данной схемы?
6. В чем состоит различие показателей ускорения и эффективности для рассматриваемых вариантов каскадной схемы суммирования?
7. В чем состоит параллельный алгоритм вычисления всех частных сумм последовательности числовых значений?
8. Как формулируется закон Амдаля? Какой аспект параллельных вычислений позволяет учесть данный закон?
9. Какие предположения используются для обоснования закона Густавсона - Барсиса?

10. Как определяется функция изоэффективности?
11. Какой алгоритм является масштабируемым? Приведите примеры методов с разным уровнем масштабируемости.

Тема 9. Оценка коммуникационной трудоемкости параллельных алгоритмов

1. В чем состоят алгоритмы выполнения передачи данных от всех процессоров всем процессорам сети для топологий кольца, решетки и гиперкуба? Приведите оценки временной трудоемкости для этих алгоритмов.
2. В чем состоят возможные алгоритмы выполнения операции редукции? Какой из алгоритмов является наилучшим по времени выполнения?
3. В чем состоит алгоритм выполнения операции циклического сдвига?
4. В чем состоит полезность использования логических топологий? Приведите примеры алгоритмов логического представления структуры коммуникационной сети.
5. В чем состоит различие моделей для оценки времени выполнения операций передачи данных в кластерных вычислительных системах? Какая модель является более точной? Какая модель может быть использована для предварительного анализа временной трудоемкости коммуникационных операций?

Тема 10. Принципы разработки параллельных методов

1. Каковы основные действия на этапе определения информационных зависимостей?
2. В чем состоят основные действия на этапе масштабирования имеющегося набора подзадач?
3. В чем состоят основные действия на этапе распределения подзадач по процессорам вычислительной системы?
4. Как происходит динамическое управление распределением вычислительной нагрузки при помощи схемы "менеджер -исполнитель"?
5. Какой метод параллельных вычислений был разработан для решения гравитационной задачи N тел?
6. Какой способ выполнения операции обобщенного сбора данных является более эффективным?

Тема 11. Параллельное программирование на основе MPI

1. Как организуется неблокирующий обмен данными в MPI?
2. В чем состоит понятие тупика? Когда функция одновременного выполнения передачи и приема гарантирует отсутствие тупиковых ситуаций?
3. Какие коллективные операции передачи данных предусмотрены в MPI?
4. Что понимается под производным типом данных в MPI?
5. Какие способы конструирования типов имеются в MPI?
6. В каких ситуациях может быть полезна упаковка и распаковка данных?
7. Что понимается в MPI под коммуникатором?
8. Для чего может потребоваться создание новых коммуникаторов?
9. Что понимается в MPI под виртуальной топологией?
10. Какие виды топологий предусмотрены в MPI?
11. Для чего может оказаться полезным использование виртуальных топологий?
12. В чем состоят особенности разработки параллельных программ с использованием MPI на алгоритмическом языке Fortran?
13. Какие основные дополнительные возможности предусмотрены в стандарте MPI?

Компетентностно-ориентированные задания

1. Приведите дополнительные примеры параллельных вычислительных систем.
2. Рассмотрите дополнительные способы классификации компьютерных систем.

3. Рассмотрите способы обеспечения когерентности кэшей в системах с общей разделяемой памятью.
4. Подготовьте обзор программных библиотек, обеспечивающих выполнение операций передачи данных для систем с распределенной памятью.
5. Рассмотрите топологию сети передачи данных в виде двоичного дерева.
6. Выделите эффективно реализуемые классы задач для каждого типа топологий сети передачи данных.
7. 5. Разработайте модель и выполните полный анализ эффективности параллельных вычислений (ускорение, эффективность, максимально достижимое ускорение, ускорение масштабирования, функция изоэффективности) для задачи умножения матрицы на вектор.
8. Разработайте алгоритмы выполнения основных операций передачи данных для топологии сети в виде 3-мерной решетки.
9. Разработайте алгоритмы выполнения основных операций передачи данных для топологии сети в виде двоичного дерева.
10. Разработайте алгоритмы логического представления двоичного дерева для различных физических топологий сети.
11. Разработайте схему параллельных вычислений для задачи умножения матрицы на вектор, используя рассмотренную в разделе методику проектирования и разработки параллельных методов.
12. Разработайте программу для нахождения минимального (максимального) значения среди элементов вектора.
13. Разработайте программу для вычисления скалярного произведения двух векторов.
14. Разработайте программу, в которой два процесса многократно обмениваются сообщениями длиной p байт. Выполните эксперименты и оцените зависимость времени выполнения операции передачи данных от длины сообщения. Сравните с теоретическими оценками, построенными по модели Хокни.
15. Подготовьте варианты ранее разработанных программ с разными режимами выполнения операций передачи данных. Сравните время выполнения операций передачи данных при разных режимах работы.
16. Подготовьте варианты ранее разработанных программ с использованием неблокирующего способа выполнения операций передачи данных. Оцените количество вычислительных операций, необходимое для того, чтобы полностью совместить передачу данных и вычисления. Разработайте программу, в которой бы полностью отсутствовали задержки вычислений из-за ожидания передаваемых данных.
17. Разработайте программу-пример для каждой имеющейся в MPI коллективной операции.
18. Разработайте реализации коллективных операций при помощи парных обменов между процессами. Выполните вычислительные эксперименты и сравните время выполнения разработанных программ и функций MPI для коллективных операций.
19. Разработайте программу, выполните эксперименты и сравните результаты для разных алгоритмов реализации операции сбора, обработки и рассылки данных всех процессам (функция `MPI_Allreduce`).
20. Разработайте программу-пример для каждого имеющегося в MPI способа конструирования производных типов данных.

21. Разработайте программу-пример с использованием функций упаковки и распаковки данных. Выполните эксперименты и сравните с результатами при использовании производных типов данных.
22. Разработайте производные типы данных для строк, столбцов, диагоналей матриц.
23. Разработайте программу-пример для каждой из рассмотренных функций для управления процессами и коммутаторами.
24. Разработайте программу для представления множества процессов в виде прямоугольной решетки. Создайте коммутаторы для каждой строки и столбца процессов. Выполните коллективную операцию для всех процессов и для одного из созданных коммутаторов. Сравните время выполнения операции.
25. Изучите самостоятельно и разработайте программы-примеры для передачи данных между процессами разных коммутаторов.
26. Разработайте программу-пример для декартовой топологии.
27. Разработайте программу-пример для топологии графа.
28. Разработайте подпрограммы для создания некоторого набора дополнительных виртуальных топологий (звезда, дерево и др.).

Составитель Кочеров Ю.Н.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Вопросы к экзамену

Технологии программирования

Пороговый уровень

Принципы построения параллельных вычислительных систем

Пути достижения параллелизма

Примеры параллельных вычислительных систем

Классификация вычислительных систем

Моделирование и анализ параллельных вычислений

Модель вычислений в виде графа "операции -операнды"

Описание схемы параллельного выполнения алгоритма

Определение времени выполнения параллельного алгоритма

Оценка коммуникационной трудоемкости параллельных алгоритмов

Общая характеристика механизмов передачи данных

Анализ трудоемкости основных операций передачи данных

Принципы разработки параллельных методов

Моделирование параллельных программ

Этапы разработки параллельных алгоритмов

Параллельное программирование на основе MPI

MPI: основные понятия и определения

Введение в разработку параллельных программ с использованием MPI

Операции передачи данных между двумя процессами

Повышенный уровень

Принципы построения параллельных вычислительных систем

Характеристика типовых схем коммуникации в многопроцессорных вычислительных системах

Характеристика системных платформ для построения кластеров

Моделирование и анализ параллельных вычислений

Показатели эффективности параллельного алгоритма

Вычисление частных сумм по следовательности числовых значений

Оценка максимально достижимого параллелизма

Анализ масштабируемости параллельных вычислений

Оценка коммуникационной трудоемкости параллельных алгоритмов

Методы логического представления топологии коммуникационной среды

Оценка трудоемкости операций передачи данных для кластерных систем

Принципы разработки параллельных методов

Параллельное решение гравитационной задачи N тел

Параллельное программирование на основе MPI

Коллективные операции передачи данных

Производные типы данных в MPI

Управление группами процессов и коммутаторами

Виртуальные топологии

Составитель Кочеров Ю.Н.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

**Паспорт фонда тестовых заданий
по дисциплине Технологии программирования**

№ п/п	Тест	Ключ
1.	Плавающие регистры содержат ... -символы -вещественные числа -целые числа	вещественные числа
2.	Языки, формирующие в процессе трансляции окончательный выполнимый код называются ... языками -компилируемыми -транслируемыми -интерпретируемыми	компилируемыми
3.	Регистрами процессора называют... -оперативную память компьютера -инструкции процессора -внутреннюю память процессора	внутреннюю память процессора
4.	Языки программирования можно разделить на языки ... уровня -переходного -среднего -высокого и низкого	высокого и низкого
5.	Синтаксис языка определяет... -последовательность операторов -правила использования языка -правила построения элементов языка	правила построения элементов языка
6.	Выражение - это... -набор лексем и символов, задающий правило вычисления некоторого значения -набор символов, определяющий смысл и правила использования единица языка -минимальная единица языка, имеющая самостоятельный смысл	набор лексем и символов, задающий правило вычисления некоторого значения
7.	Чтобы компиляция одной и той же программы различными компиляторами всегда давала одинаковый результат, разрабатываются ... -алфавит языка, его символы, лексемы -средства разработки программ -стандарты языков программирования	стандарты языков программирования
8.	Операторы бывают ... -зависимыми и независимыми	-исполняемыми и неисполняемыми

	-действующими и недействующими -исполняемыми и неисполняемыми	
9.	Команды процессора состоят из... -регистров -кода операции и операндов -программ	кода операции и операндов
10.	Группа операторов, которые решают логически самостоятельную часть задачи, называется ... -кодировщиком -подпрограммой -циклом	подпрограммой
11.	- это программа, переводящая текст инструкций для компьютера с какого-либо языка программирования на машинный язык	Компилятор
12.	Элементарной единицей памяти современных компьютеров является ... -один бит -один байт -четыре байта	один байт
13.	Неисполняемые операторы - это операторы, ... -служащие для описания данных -устанавливающие правила построения элементов языка -задающие действия над данными	служащие для описания данных
14.	Оператор - это ... -законченное описание некоторого действия -язык программирования -основной неделимый символ	законченное описание некоторого действия
15.	Исполняемые операторы - это операторы,... -служащие для описания данных -задающие действия над данными -устанавливающие правила построения элементов языка	задающие действия над данными
16.	В процессе трансляции, на этапе лексического анализа выполняется ... -выделение основных составляющих программы - символов -операторы, устанавливающие правила построения элементов языка -выделение основных составляющих программы – лексем	выделение основных составляющих программы – лексем
17.	Аппаратный стек реализуется... -во внутренней памяти процессора -на базе постоянного запоминающего устройства	на базе оперативной памяти
18.	Языки, представляющие собой последовательность выполняемых операторов, называют ... -процедурными -аппликативными	процедурными
19.	Машинно-ориентированный язык программирования -... -Base -Cobol	Assembler

	-Fortran -Assembler	
20.	Общие регистры содержат... -вещественные числа -целые числа или адреса -команды условных переходов	целые числа или адреса
21.	Величина, принимающая в процессе выполнения приложения различные значения, называется ... -переменной -транслятором -константой	переменной
22.	Система программирования - это ... -программа, которая сразу выполняет переводимое команды -средство автоматизации программирования, включающее язык программирования, транслятор этого языка, документацию а также средства подготовки и выполнения программ -программа, преобразующая программу, написанную на одном (входном) языке, в программу, представленную на другом (выходном) языке	средство автоматизации программирования, включающее язык программирования, транслятор этого языка, документацию а также средства подготовки и выполнения программ
23.	Языки, основанные на построении объектов, объединяющих данные и операции над ними, называют ... -Объектно-ориентированными -процедурными	Объектно-ориентированными
24.	Единица текста программы, которая при компиляции воспринимается, как единое целое и по смыслу не может быть разделена на более мелкие элементы, называется... -мантиссой -лексемой -транслятором -константой	лексемой
25.	Графическое изображение логической структуры алгоритма, в котором каждый этап процесса переработки данных представляется в виде геометрических фигур(блоков), имеющих определенную конфигурацию в зависимости от характера выполняемых операций, – это ... форма записи алгоритмов -графическая (блок-схемная) -словесно-пошаговая -программная	графическая (блок-схемная)
26.	Оператор присваивания в языке C/C++ обозначается ... (вписать символ)	=
27.	Упорядоченная совокупность однотипных переменных, обладающих одинаковыми	массивом

	<p>свойствами, называется ...</p> <ul style="list-style-type: none"> -массивом -списком -последовательностью -множеством 	
28.	<p>К нейтральным или выполнимым формулам логики относятся выражения, принимающие значение ...</p> <ul style="list-style-type: none"> -«истинно» независимо от логических значений входящих в них переменных -«ложно» независимо от логических значений входящих в них переменных -как «истинно», так и «ложно» 	<p>как «истинно», так и «ложно»</p>
29.	<p>Структура – это ...</p> <ul style="list-style-type: none"> -составной объект, в который входят компоненты любых типов, имеющих нулевое смещение от начала, за исключением функций, сгруппированных под одним именем для удобства обработки, и в каждый момент времени может использоваться только один из его компонентов -имеющая единое имя совокупность однотипных элементов, расположенных в определенном порядке и различающихся индексами, и над этой совокупностью возможны любые операции, допускаемые типом элементов массива -составной объект, в который входят компоненты любых типов, за исключением функций, сгруппированных под одним именем для удобства обработки, и в каждый момент времени может использоваться только один из его компонентов 	<p>составной объект, в который входят компоненты любых типов, за исключением функций, сгруппированных под одним именем для удобства обработки, и в каждый момент времени может использоваться только один из его компонентов</p>
30.	<p>Если «А» имеет значение «истина», а «В» – «ложь»: $C = A \parallel B$, то величине «С» будет присвоено значение ...</p> <ul style="list-style-type: none"> -«ложь» -«истина» или «ложь» -«истина» 	<p>«истина»</p>

Составитель Кочеров Ю.Н.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

**КРИТЕРИИ И ШКАЛЫ ОЦЕНИВАНИЯ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО
КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ**

«5» (отлично): студент демонстрирует системные теоретические знания, владеет терминологией, делает аргументированные выводы и обобщения, приводит примеры, показывает свободное владение монологической речью и способность быстро реагировать на уточняющие вопросы.

«4» (хорошо): студент демонстрирует прочные теоретические знания, владеет терминологией, делает аргументированные выводы и обобщения, приводит примеры, показывает свободное владение монологической речью, но при этом делает несущественные ошибки, которые быстро исправляет самостоятельно или при незначительной коррекции преподавателем.

«3» (удовлетворительно): студент демонстрирует неглубокие теоретические знания, проявляет слабо сформированные навыки анализа явлений и процессов, недостаточное умение делать аргументированные выводы и приводить примеры, показывает недостаточно свободное владение монологической речью, терминологией, логичностью и последовательностью изложения, делает ошибки, которые может исправить только при коррекции преподавателем.

«2» (неудовлетворительно): студент демонстрирует незнание теоретических основ предмета, не умеет делать аргументированные выводы и приводить примеры, показывает слабое владение монологической речью, не владеет терминологией, проявляет отсутствие логичности и последовательностью изложения, делает ошибки, которые не может исправить даже при коррекции преподавателем, отказывается отвечать на занятии