

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Чернецкая Ирина Евгеньевна
Должность: Заведующий кафедрой
Дата подписания: 27.02.2023 23:30:03
Уникальный программный ключ:
bdf214c64d8a381b0782ea566b0dce05e3f5ea2d

МИНОБРНАУКИ РОССИИ

Юго-Западный государственный университет

УТВЕРЖДАЮ

Заведующий кафедрой

вычислительной техники

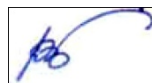
 И.Е.Чернецкая

« 01 » сентября 2022 г.

ОЦЕНОЧНЫЕ СРЕДСТВА
для текущего контроля успеваемости
и промежуточной аттестации обучающихся
по дисциплине
ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ
(наименование дисциплины)

09.03.01 Информатика и вычислительная техника, направленность
(профиль) «Вычислительные машины, комплексы, системы и сети»
(код и наименование ОПОП ВО)

Разработчик: доцент кафедры ВТ



Ватулин Э.И.

1 ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ

1.1 ВОПРОСЫ ДЛЯ СОБЕСЕДОВАНИЯ

1. Раздел (тема) дисциплины «Введение»

1. Краткий исторический экскурс в историю развития параллельных систем и вычислений.
2. Основные современные проблемы и перспективы развития параллельного программирования. Закон Мура.

2. Раздел (тема) дисциплины «Понятие производительности вычислительной системы и способы ее измерения»

3. Пиковая и реальная производительность.
4. Единицы измерения производительности.
5. Бенчмарки.

3. Раздел (тема) дисциплины «Закон Амдала»

6. Теоретическая оценка выигрыша от распараллеливания.

4. Раздел (тема) дисциплины «Средства аппаратной поддержки параллельного исполнения»

7. Основные способы организации (микро)архитектуры современных вычислительных средств
8. Поддержка программирования аппаратных средств со стороны разработчика
9. Виды параллелизма.

5. Раздел (тема) дисциплины «Профилирование и оптимизация программных средств»

10. Понятие и функции профайлера.
11. Идентификация узких мест.
12. Понятие алгоритмической, высокоуровневой и микроархитектурной оптимизации.
13. Примеры оптимизации и рекомендации
14. Анализ качества кода современных компиляторов.

Шкала оценивания: 48-балльная.

Критерии оценивания

48 баллов выставляется обучающемуся, если он принимает активное участие в беседе по большинству обсуждаемых вопросов (в том числе самых сложных); демонстрирует сформированную способность к диалогическому мышлению, проявляет уважение и интерес к иным мнениям; владеет глубокими (в том числе дополнительными) знаниями по существу обсуждаемых вопросов, ораторскими способностями и правилами ведения полемики; строит логичные, аргументированные, точные и лаконичные высказывания, сопровождаемые яркими примерами; легко и заинтересованно откликается на неожиданные ракурсы беседы; не нуждается в уточняющих и (или) дополнительных вопросах преподавателя.

36 баллов выставляется обучающемуся, если он принимает участие в обсуждении

не менее 50% дискуссионных вопросов; проявляет уважение и интерес к иным мнениям, доказательно и корректно защищает свое мнение; владеет хорошими знаниями вопросов, в обсуждении которых принимает участие; умеет не столько вести полемику, сколько участвовать в ней; строит логичные, аргументированные высказывания, сопровождаемые подходящими примерами; не всегда откликается на неожиданные ракурсы беседы; не нуждается в уточняющих и (или) дополнительных вопросах преподавателя.

24 балла выставляется обучающемуся, если он принимает участие в беседе по одному-двум наиболее простым обсуждаемым вопросам; корректно выслушивает иные мнения; неуверенно ориентируется в содержании обсуждаемых вопросов, порой допуская ошибки; в полемике предпочитает занимать позицию заинтересованного слушателя; строит краткие, но в целом логичные высказывания, сопровождаемые наиболее очевидными примерами; теряется при возникновении неожиданных ракурсов беседы и в этом случае нуждается в уточняющих и (или) дополнительных вопросах преподавателя.

0 баллов (выставляется обучающемуся, если он не владеет содержанием обсуждаемых вопросов или допускает грубые ошибки; пассивен в обмене мнениями или вообще не участвует в дискуссии; затрудняется в построении монологического высказывания и (или) допускает ошибочные высказывания; постоянно нуждается в уточняющих и (или) дополнительных вопросах преподавателя.

2 ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ

2.2 БАНК ВОПРОСОВ И ЗАДАНИЙ В ТЕСТОВОЙ ФОРМЕ

1 Вопросы в закрытой форме

1.1. Допускается ли задание начальных значений глобальным переменным в разделе описаний программы?

Варианты ответа:

- a: да
- b: нет
- c: зависит от установки опций компилятора

1.2. Для преждевременного прерывания цикла используется

Варианты ответа:

- a: break
- b: continue
- c: jump

1.3. Для преждевременного перехода к следующей итерации цикла используется

Варианты ответа:

- a: continue
- b: break
- c: jump
- d: вложение цикла в еще один цикл

1.4. В каких случаях оправдано использование оператора goto?

Варианты ответа:

- a: для выхода из многократно вложенных циклов

- b: для организации циклов
- c: для организации условных ветвлений
- d: при организации перегрузки функций

1.5. Массивы какой минимальной размерности поддерживаются в Delphi?

Варианты ответа:

- a: 1
- b: 0
- c: 2
- d: не ограничено

1.6. Массивы какой максимальной размерности поддерживаются в Delphi?

Варианты ответа:

- a: 1
- b: 0
- c: 2
- d: не ограничено

1.7. Допускается ли задание начальных значений глобальным переменным в разделе описаний программы?

Варианты ответа:

- a: не ограничено
- b: 1
- c: 0
- d: 2

1.8. Размер динамического массива

Варианты ответа:

- a: определяется в процессе выполнения программы
- b: известен на этапе компиляции
- c: динамически меняется компилятором прозрачно для пользователя

1.9. Размер статического массива

Варианты ответа:

- a: известен на этапе компиляции
- b: определяется в процессе выполнения программы
- c: динамически меняется компилятором прозрачно для пользователя

1.10. Может ли динамический массив быть многомерным?

Варианты ответа:

- a: да
- b: нет
- c: может, только если память выделяется с помощью GetMem() или SetLength()

2 Вопросы в открытой форме

2.1. Процессор исполняет программы в виде ...

- двоичного кода
- Delphi
- C++
- ассемблера

2.2. Компилятор предназначен для ...

- преобразования программы из языка высокого уровня в объектный код
- получения готового исполнимого файла из объектного кода
- преобразования exe-файла в com-файл
- проверки синтаксиса и семантики программы

2.3. Линкер (компоновщик) предназначен для ...

- получения готового исполнимого файла из объектного кода
- преобразования программы из языка высокого уровня в объектный код
- преобразования exe-файла в com-файл
- проверки синтаксиса и семантики программы

2.4. К исполняемым файлам НЕ относятся файлы ...

- pas
- exe
- dll
- com

2.5. К файлам с исходными кодами программы на Delphi не относятся файлы ...

- scr
- pas
- dfm
- dpr

2.6. Оптимизация программы – это ...

- структурные преобразования, уменьшающие время выполнения программы или ее фрагмента
- опция компилятора
- средство затруднения отладки
- структурные преобразования, направленные на изменение алгоритма программы с целью его упрощения

2.7. Интегрированная среда разработки (IDE) Delphi не включает ...

- профайлер
- компилятор
- отладчик
- редактор

2.8. VCL – это ...

- библиотека функций и компонентов
- конструкция языка
- предопределенная переменная
- зарезервированное слово

2.9. VCL не содержит компонента для ...

- вывода математических формул
- вывода текстовых надписей

- отображения таблиц
- отображения графиков и диаграмм

2.10. Окно инспектора объектов используется для ...

- назначения свойств и событий компонентам
- создания свойств и событий компонентов
- выбора сообщений Windows, обрабатываемых программой
- отладки компонентов

3 Установление правильной последовательности

3.1. В каком порядке производится разработка программного обеспечения?

- анализ требований
- составление технического задания
- проектирование структур данных
- разработка алгоритмов
- написание кода
- отладка
- тестирование
- сопровождение

3.2. В каком порядке описываются элементы заголовка процедуры?

- ключевое слово `procedure`
- имя процедуры
- параметры
- директивы
- локальные объявления
- тело процедуры

3.3. В каком порядке описываются элементы заголовка функции?

- ключевое слово `function`
- имя процедуры
- параметры
- директивы
- локальные объявления
- тело процедуры

3.4. В каком порядке указываются ключевые слова, обозначающие элементы модуля (unit)?

- `unit`
- `interface`
- `implementation`
- `initialization`
- `finalization`

3.5. Расположите типы памяти в порядке увеличения времени доступа к ним.

- регистры процессора
- кэш-память первого уровня (L1)
- кэш-память второго уровня (L2)
- кэш-память третьего уровня (L3), при наличии
- оперативная память
- жесткий диск

3.6. Расположите в порядке запуска элементы компилятора.

- препроцессор (при наличии)
- компилятор
- линкер (компоновщик)

3.7. Расположите обработчики событий формы оконного приложения в порядке их срабатывания.

- OnCreate()
- OnShow()
- OnClick()
- OnClose()

3.8. Расположите ключевые слова в том порядке, в котором они используются при записи консольной программы без подпрограмм.

- program
- uses
- var
- begin
- end

3.9. Расположите в нужном порядке лексемы языка при объявлении переменной.

- var
- имя переменной
- :
- тип переменной
- ;

3.10. Расположите типы данных в порядке увеличения их размера в байтах.

- Byte
- Word
- LongWord
- Int64

4 Установление соответствия

4.1. Укажите синтаксически неправильное значение константы

- +8.903E-17.1
- -123.17
- \$F0
- 1547

4.2. При делении целых чисел при помощи операции div результат получается

- целым
- вещественным
- символьным

4.3. При делении целых чисел при помощи операции / результат получается

- вещественным
- целым
- символьным

4.4. Операция mod обозначает

- остаток от деления

- модуль числа
- изменение (модифицирование) знака

4.5. Если $A = 1001\ 1100$ и $B = 1100\ 1111$, то $A \text{ and } B =$

- 1000 1100
- 1001 1100
- 1101 1111
- нет правильного ответа

4.6. Если $A = 1001\ 1100$ и $B = 1100\ 1111$, то $A \text{ or } B =$

- 1101 1111
- 1000 1100
- 1111 1101
- нет правильного ответа

4.7. Если $A = 1001\ 1100$ и $B = 1100\ 1111$, то $A \text{ xor } B =$

- 0101 0011
- 1101 1111
- 0110 1011
- нет правильного ответа

4.8. Если $A = 1001\ 1100$, то $\text{not } A =$

- 0110 0011
- 0011 1000
- 0110 1011
- нет правильного ответа

4.9. Если $A = 1001\ 1100$, то $A \text{ shl } 1 =$

- 0011 1000
- 0001 1100
- 0000 1110
- нет правильного ответа

4.10. Если $A = 1001\ 1100$, то $A \text{ shr } 2 =$

- 0010 0111
- 0100 1110
- 0011 1000
- нет правильного ответа

Шкала оценивания результатов тестирования: в соответствии с действующей в университете балльно-рейтинговой системой оценивание результатов промежуточной аттестации обучающихся осуществляется в рамках 100-балльной шкалы, при этом максимальный балл по промежуточной аттестации обучающихся по очной форме обучения составляет 36 баллов, по заочной форме обучения – 60 баллов (установлено положением П 02.016).

Максимальный балл за тестирование представляет собой разность двух чисел: максимального балла по промежуточной аттестации для данной формы обучения (36 или 60) и максимального балла за решение компетентностно-ориентированной задачи (6). Балл, полученный обучающимся за тестирование, суммируется с баллом, выставленным ему за решение компетентностно-ориентированной задачи. Общий балл по промежуточной аттестации суммируется с баллами, полученными обучающимся по результатам текущего контроля успеваемости в течение семестра; сумма баллов

переводится в оценку по 5-балльной шкале следующим образом.

Соответствие 100-балльной и 5-балльной шкал

<i>Сумма баллов по 100-балльной шкале</i>	<i>Оценка по 5-балльной шкале</i>
100-85	отлично
84-70	хорошо
69-50	удовлетворительно
49 и менее	неудовлетворительно

ВОПРОСЫ К ЗАЧЕТУ

1. Краткая история развития параллельных вычислительных средств.
2. Основные тенденции развития параллельных вычислительных средств.
3. Производительность вычислительных средств и способы ее оценки.
4. Профайлеры и профилирование программ. Способы измерения времени выполнения фрагментов кода. Поиск узких мест.
5. Закон Амдала.
6. Закон Густафсона-Барсиса.
7. Способы повышения производительности аппаратных средств вычислительной техники и их программная поддержка.
8. Виды параллельных вычислительных структур. Достоинства и недостатки. Особенности программирования.
9. Оптимизация программных средств: уровни оптимизации и основные правила оптимизации.
10. Оптимизирующие компиляторы: обзор возможностей, поддерживаемые оптимизации.
11. Грид-системы: принципы организации и программирования. Добровольный метакомпьютинг.
18. Многопоточное программирование. Понятие потоков и процессов. Приоритеты, привязка к ядрам процессора. Примитивы синхронизации.

2.2 КОМПЕТЕНТНОСТНО-ОРИЕНТИРОВАННЫЕ ЗАДАЧИ

Постановка задачи.

1. Вывести на экран все простые числа в диапазоне от 1 до N . Число называется простым, если оно делится без остатка только само на себя и на 1.
2. Найти натуральное число в диапазоне от 1 до N с максимальной суммой делителей.
3. Представить заданное число в виде произведения простых чисел (например, $5040 = 2^4 \cdot 3^2 \cdot 5 \cdot 7$).
4. Найти наименьшее натуральное число N из диапазона $[A, B]$, представимое в виде суммы кубов двух натуральных чисел $N = x^3 + y^3$.
5. Натуральное число из N цифр называется числом Армстронга, если оно может быть представлено в виде суммы его цифр, возведенных в N -ую степень (например, $153 = 1^3 + 5^3 + 3^3$). Найти числа Армстронга в заданном диапазоне.
6. Указать все способы представления заданного числа N в виде суммы квадратов 3 натуральных чисел $N = x^2 + y^2 + z^2$.
7. Согласно теореме Лагранжа о сумме четырех квадратов любое число можно представить в виде суммы не более чем 4 квадратов натуральных чисел. Проверить справедливость теоремы для чисел из заданного диапазона.
8. (Плоские числа) Найти и вывести на экран все числа, меньшие N , единственным образом представимые в виде произведения простых чисел A и B (например, $985 = 5 \cdot 197$).
9. (Телесные числа) Найти и вывести на экран все числа, меньшие N , единственным образом представимые в виде произведения простых чисел A , B и C (например, $385 = 5 \cdot 7 \cdot 11$).
10. В приведенной последовательности сумм

$$S_1 = 1,$$

$$S_2 = 3 + 5 = 8,$$

$$S_3 = 7 + 9 + 11 = 27,$$

...

вычислить значение i -ой суммы. Вывести на экран все промежуточные значения.

11. Найти наилучшее приближение к заданному иррациональному числу (π , e , $\sqrt{2}$ и т.д.) в виде рациональной дроби $\frac{A}{B}$ в заданном диапазоне ($1 \leq A \leq N$, $1 \leq B \leq N$).

12. Проверить справедливость соотношения $\sum_{i=0}^n C_n^i = 2^n$, где $C_n^m = \frac{n!}{m!(n-m)!}$ – число сочетаний из n по m .

13. Проверить справедливость соотношения $\sum_{i=0}^n (-1)^i C_n^i = 0$, где $C_n^m = \frac{n!}{m!(n-m)!}$ – число сочетаний из n по m .

14. (Тождество Абеля) Убедиться в справедливости соотношения $2(n-1)n^{n-2} = \sum_{\substack{k+m=n, \\ k>1, m>1}} C_n^k k^{k-1} m^{m-1}$, где $C_n^m = \frac{n!}{m!(n-m)!}$ – число сочетаний из n по m .

15. Проверить справедливость соотношения $n! = 1 + \sum_{i=1}^{n-1} i \cdot i!$.

16. Проверить справедливость соотношения $\sum_{i=1}^n i^3 = \left(\sum_{i=1}^n i \right)^2 = \frac{n^2(n+1)^2}{4}$.

17. Проверить справедливость соотношения $\sum_{k=0}^{n-1} 2^k = 2^n - 1$.
18. Проверить справедливость соотношения $(x-a)^n \equiv x^n - a \pmod{n}$, где x, a, n – целые положительные числа, n – простое число, $a \leq x$, с использованием датчика случайных чисел.
19. (малая теорема Ферма) Если a не делится на простое число p , то $a^{p-1} - 1$ делится на p . Проверить справедливость теоремы с использованием датчика случайных чисел.
20. Проверить или опровергнуть гипотезу Ферма. Простые числа вида $4k+1$ (k – натуральное число) можно единственным способом представить в виде суммы квадратов двух целых чисел. Например, при $k=1$ $p=4 \cdot 1 + 1 = 5 = 1^2 + 2^2$, а при $k=3$ $p=4 \cdot 3 + 1 = 13 = 2^2 + 3^2$.
21. Найти все простые сомножители заданного числа N .
22. Для всех нечетных чисел n значение $n^2 - 1$ делится на 8. Проверить справедливость утверждения.
23. Если $p > 3$ – простое число, то значение $p^2 - 1$ делится на 24. Убедиться в справедливости утверждения.
24. Найти все возможные значения цифр $0 \leq D, E, M, N, O, R, S, Y \leq 9$ такие, что выполняется равенство

$$\begin{array}{r} SEND \\ + MORE \\ \hline MONEY \end{array}$$

Пример решения: $D = 0, E = 0, M = 1, N = 0, O = 0, R = 0, S = 9, Y = 0$.

25. Убедиться в справедливости равенства $\sum_{i=1}^n H_i = (n+1)[H_{n+1} - 1]$, где $H_n = \sum_{k=1}^n \frac{1}{k}$ – ряд гармонических чисел.

26. Убедиться в справедливости равенства $\sum_{i=1}^n iH_i = C_{n+1}^2 \left(H_{n+1} - \frac{1}{2} \right)$, где $C_n^m = \frac{n!}{m!(n-m)!}$

– число сочетаний из n по m , $H_n = \sum_{k=1}^n \frac{1}{k}$ – сумма ряда гармонических чисел.

27. Убедиться в справедливости равенства $\sum_{i=1}^n C_i^k H_i = C_{n+1}^{k+1} \left(H_{n+1} - \frac{1}{k+1} \right)$ для заданных

значений n и k , где $C_n^m = \frac{n!}{m!(n-m)!}$ – число сочетаний из n по m , $H_n = \sum_{k=1}^n \frac{1}{k}$ – сумма ряда

гармонических чисел.

28. Проверить справедливость соотношения $\sum_{k=0}^n x^k = \frac{x^{n+1} - 1}{x - 1}$.

29. Проверить справедливость соотношения $\sum_{k=1}^n \frac{1}{k(k+1)} = \frac{n}{n+1}$.

30. Проверить справедливость соотношения $\sum_{k=1}^n k \cdot 2^k = (n-1) \cdot 2^{n+1} + 2$.

Произвести разработку последовательной (однопоточной) программной реализации, выявить узкие места, разработать параллельную программную реализацию для указанной преподавателем задачи, оценить полученный выигрыш.

Шкала оценивания решения компетентностно-ориентированной задачи: в соответствии с действующей в университете балльно-рейтинговой системой оценивание результатов промежуточной аттестации обучающихся осуществляется в рамках 100-балльной шкалы, при этом максимальный балл по промежуточной аттестации обучающихся по очной форме обучения составляет 36 баллов, по очно-заочной и заочной формам обучения – 60 (установлено положением П 02.016).

Максимальное количество баллов за решение компетентностно-ориентированной задачи – 6 баллов.

Балл, полученный обучающимся за решение компетентностно-ориентированной задачи, суммируется с баллом, выставленным ему по результатам тестирования.

Общий балл промежуточной аттестации суммируется с баллами, полученными обучающимся по результатам текущего контроля успеваемости в течение семестра; сумма баллов переводится в оценку по 5-балльной шкале.

Критерии оценивания решения компетентностно-ориентированной задачи:

6-5 баллов выставляется обучающемуся, если решение задачи демонстрирует глубокое понимание обучающимся предложенной проблемы и разностороннее ее рассмотрение; свободно конструируемая работа представляет собой логичное, ясное и при этом краткое, точное описание хода решения задачи и формулировку доказанного, правильного вывода (ответа); задача решена в установленное преподавателем время или с опережением времени.

4-3 балла выставляется обучающемуся, если решение задачи демонстрирует понимание обучающимся предложенной проблемы; задача решена типовым способом в установленное преподавателем время; имеют место несущественные недочеты в описании хода решения и (или) вывода (ответа).

2-1 балла выставляется обучающемуся, если решение задачи демонстрирует поверхностное понимание обучающимся предложенной проблемы; осуществлена попытка шаблонного решения задачи, но при ее решении допущены ошибки и (или) превышено установленное преподавателем время.

0 баллов выставляется обучающемуся, если решение задачи демонстрирует непонимание обучающимся предложенной проблемы и (или) задача не решена.