

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Порохня Андрей Алексеевич
Должность: И.о. директора института перспективной инженерии
Дата подписания: 17.06.2026 14:34:15
Уникальный программный ключ:
990bea3aeec848c23bd8699e48288f8d1960c608

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

УТВЕРЖДАЮ

И.о. директора института, кандидат
технических наук, доцент

А.А. Порохня

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ
Многопоточное программирование
название дисциплины

Направление подготовки	09.03.03 Прикладная информатика
Направленность (профиль)	Прикладная информатика в экономике
Год начала обучения	2026
Форма обучения	очная
Реализуется в семестре	6

Введение

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

2. ФОС является приложением к программе дисциплины «Многопоточное программирование»

3. Разработчик, Орлова А.Ю. доцент департамента цифровых, робототехнических систем и электроники.

4. Проведена экспертиза ФОС.

Члены экспертной группы:

Председатель Азаров И.В., и.о. директора департамента цифровых, робототехнических систем и электроники

Члены комиссии:

Амироков С.Р., доцент департамента цифровых, робототехнических систем и электроники

Орлинская О.Г., доцент департамента цифровых, робототехнических систем и электроники

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 компетенци(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
ПК-2				
ИД-1 ПК-2 разрабатывать прикладное программное обеспечение.	С трудом разрабатывает прикладное программное обеспечение.	Не в полном объеме разрабатывает прикладное программное обеспечение	В целом разрабатывает прикладное программное обеспечение	разрабатывает прикладное программное обеспечение
ИД-2 ПК-2. адаптировать прикладное программное обеспечение.	С трудом адаптирует прикладное программное обеспечение.	Не в полном объеме адаптирует прикладное программное обеспечение.	В целом адаптирует прикладное программное обеспечение.	адаптирует прикладное программное обеспечение.

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 6	
1.	3,7,8	Граф: (1) состоит из множества вершин (2) состоит из множества дуг (3) состоит из множества вершин и дуг (4) всегда ацикличен (5) всегда содержит циклы (6) всегда содержит непустое множество дуг (7) всегда содержит непустое множество вершин (8) бывает цикличным	ПК-2
2.	2	Верно ли, что информационная история любого фрагмента всегда содержит больше дуг, чем операционная история того же фрагмента? (1) да (2) нет	ПК-2
3.	1,2,4	Для некоторого фрагмента программы построили четыре модели. Может ли оказаться несвязным граф: (1) информационный граф (2) граф управления (3) операционная история (4) информационная история	ПК-2
4.	1	Верно ли утверждение: если в тесновложенном гнезде циклов все итерации каждого цикла гнезда независимы, то перестановка циклов в таком гнезде будет эквивалентным преобразованием? (1) да (2) нет	ПК-2
5.	3	Подход к оптимизации вычислений, основанный на использовании команд SIMD (SSE, SSE2...), называется: (1) интерполяция (2) модуляция	ПК-2

		(3) векторизация (4) комбинация	
6.	1,2	Статическая конвейеризация выполняется (1) оптимизирующим компилятором (2) программистом (3) ядром CPU	ПК-2
7.	1	Динамическое планирование это: (1) аппаратный механизм переупорядочивания исполнения инструкций (out-of-order engine) (2) генерация машинного кода с учетом результатов профилирования (3) метод решения задач дискретной оптимизации	ПК-2
8.	1	Степень параллелизма программы (1) $D(t)$ – число процессоров, участвующих в исполнении программы в момент времени (2) отношение времени выполнения последовательной версии к времени выполнения параллельной версии программы (3) отношение времени выполнения программы на одноядерном процессоре к времени выполнения на многоядерном процессоре	ПК-2
9.	2	Ускорение за счёт параллельного выполнения (1) $D(t)$ – число процессоров, участвующих в исполнении программы в момент времени (2) отношение времени выполнения последовательной версии к времени выполнения параллельной версии программы (3) отношение времени выполнения программы на одноядерном процессоре к времени выполнения на многоядерном процессоре	ПК-2
10.	1,3,4	Отметьте ресурсы, которые потоки одного процесса используют совместно (1) глобальные и статические переменные (2) локальные переменные (3) динамически распределяемая память (4) открытые файлы и сетевые соединения (5) аппаратный контекст	ПК-2
11.	2,5	Отметьте частные ресурсы каждого потока (1) глобальные и статические переменные	ПК-2

		(2) локальные переменные (3) динамически распределяемая память (4) открытые файлы и сетевые соединения (5) аппаратный контекст	
12.	1	Как называется состояние потока, обладающего всеми необходимыми ресурсами, в том числе возможностью использования ЦП (1) выполнение (2) готов к выполнению (3) ожидание	ПК-2
13.	2	Как называется состояние потока, обладающего всеми необходимыми для выполнения ресурсами за исключением ресурса "время ЦП" (1) выполнение (2) готов к выполнению (3) ожидание	ПК-2
14.	1	Что отвечает за управление процессами? (1) ядро операционной системы (2) библиотека пользовательского уровня (3) возможны оба варианта	ПК-2
15.	3	Что отвечает за управление потоками? (1) ядро операционной системы (2) библиотека пользовательского уровня (3) возможны оба варианта	ПК-2
16.	1,4,5	Что включает в себя контекст процесса? (1) дескриптор процесса (2) дескриптор потока (3) аппаратный контекст (4) контекст ввода-вывода (5) описание структуры и содержимое виртуального адресного пространства	ПК-2
17.	2,3	Что включает в себя контекст потока? (1) дескриптор процесса (2) дескриптор потока (3) аппаратный контекст (4) контекст ввода-вывода	ПК-2

		(5) описание структуры и содержимое виртуального адресного пространства	
18.	4,5	Отметьте действия, не относящиеся к операции создания процесса (1) обнуление статистики в дескрипторе создаваемого объекта (2) создание виртуального адресного пространства (3) создание региона кода в виртуальном адресном пространстве (4) создание региона стека в виртуальном адресном пространстве (5) создание и инициализация аппаратного контекста (6) выделение ресурсов по умолчанию	ПК-2
19.	2,3,6	Отметьте действия, не относящиеся к операции создания потока (1) обнуление статистики в дескрипторе создаваемого объекта (2) создание виртуального адресного пространства (3) создание региона кода в виртуальном адресном пространстве (4) создание региона стека в виртуальном адресном пространстве (5) создание и инициализация аппаратного контекста (6) выделение ресурсов по умолчанию	ПК-2
20.	3	Какой тип планирования преимущественно используется в интерактивных системах? (1) долгосрочное (2) среднесрочное (3) краткосрочное	ПК-2
21.	6	Укажите оценку сложности современных алгоритмов планирования (N – число субъектов планирования). (1) $O(N^2)$ (2) $O(N \log N)$ (3) $O(N)$ (4) $O(\log N)$ (5) $O(\log \log N)$ (6) $O(1)$	ПК-2
22.	3	Укажите наиболее важный критерий оценки алгоритмов планирования в системах пакетной обработки. (1) справедливость (2) масштабируемость (3) пропускная способность (4) эффективность	ПК-2

		(5) время отклика (6) время ожидания	
23.	3,4	Выберите утверждения, верные для невытесняющих алгоритмов планирования. (1) поток может быть вытеснен только при выполнении им системного вызова (2) поток может быть вытеснен в любой не зависящий от него момент (3) поток может использовать ЦП в течение неограниченного времени (4) возможно голодание потоков (5) реализация алгоритма предъявляет требования к архитектуре	ПК-2
24.	2,4,5	Выберите утверждения, верные для вытесняющих алгоритмов планирования. (1) поток может быть вытеснен только при выполнении им системного вызова (2) поток может быть вытеснен в любой не зависящий от него момент (3) поток может использовать ЦП в течение неограниченного времени (4) возможно голодание потоков (5) реализация алгоритма предъявляет требования к архитектуре	ПК-2
25.	2,5	На значения каких критериев влияет размер кванта при использовании вытесняющих алгоритмов планирования? (1) баланс (2) накладные расходы (3) масштабируемость (4) обратное время (5) время отклика	ПК-2
26.	2,3,5	При использовании каких алгоритмов планирования возможно голодание потоков? (1) First Come – First Served (2) Shortest Job First (3) Shortest Remaining Time (4) Round Robin (5) приоритетное планирование	ПК-2
27.	2,3	В каких случаях возможно возникновение гонок (races) при использовании несколькими потоками одного объекта данных? (1) два или более потоков читают данные	ПК-2

		(2) один или более потоков читают данные, один поток изменяет данные (3) два или более потоков изменяют данные (4) ни один из перечисленных случаев	
28.	4	В каких случаях обязательно возникают гонки (races) при использовании несколькими потоками одного объекта данных? (1) два или более потоков читают данные (2) один или более потоков читают данные, один поток изменяет данные (3) два или более потоков изменяют данные (4) ни один из перечисленных случаев	ПК-2
29.	1	Укажите положения, верные для решения задачи взаимного исключения, основанного на прерываниях. (1) можно синхронизировать выполнение потоков разных процессов (2) требуется наличие разделяемых переменных для синхронизирующихся потоков (3) можно использовать в прикладных программах современных операционных систем общего назначения (Windows, UNIX,...) (4) можно использовать на многопроцессорных системах	ПК-2
30.	1,2,3,4	Укажите положения, верные для алгоритма Петерсона решения задачи взаимного исключения. (1) можно синхронизировать выполнение потоков разных процессов (2) требуется наличие разделяемых переменных для синхронизирующихся потоков (3) можно использовать в прикладных программах современных операционных систем общего назначения (Windows, UNIX,...) (4) можно использовать на многопроцессорных системах	ПК-2
31.	1,2,3,4	Укажите положения, верные для решений задачи взаимного исключения, основанного на использовании специальных команд ЦП (Test&Set, Swap,...). (1) можно синхронизировать выполнение потоков разных процессов (2) требуется наличие разделяемых переменных для синхронизирующихся потоков (3) можно использовать в прикладных программах современных операционных систем общего назначения (Windows, UNIX,...) (4) можно использовать на многопроцессорных системах	ПК-2

32.	3	Рассмотрим решение задачи взаимного исключения для двух потоков (i - номер потока). Какое из условий постановки задачи взаимного исключения может быть нарушено в предлагаемом решении? <pre> bool flag0 = true, flag1 = false; (true){ while(!flagi) ; CSi(); flagi = false; flag1-i = true; NCSi(); } </pre> (1) одновременно внутри критической секции должно находиться не более одного потока (2) критические секции не должны иметь приоритета в отношении друг друга (3) остановка какого-либо потока вне его критической секции не должна влиять на дальнейшую работу потоков по использованию критического ресурса (4) решение о вхождении потоков в их критические секции не откладывается на неопределенный срок, а является конечным во времени (5) предлагаемое решение является корректным и не нарушает ни одного из перечисленных условий	ПК-2
33.	5	Рассмотрим решение задачи взаимного исключения для двух потоков (i - номер потока). Какое из условий постановки задачи взаимного исключения может быть нарушено в предлагаемом решении? <pre> bool flag0 = false, flag1 = false; flag2 = 0; (true){ flagi = true; flag2 = 1 - i; while(flag1-i && flag2 == 1 - i) ; CSi(); flagi = false; NCSi(); } </pre>	ПК-2

		} (1) одновременно внутри критической секции должно находиться не более одного потока (2) критические секции не должны иметь приоритета в отношении друг друга (3) остановка какого-либо потока вне его критической секции не должна влиять на дальнейшую работу потоков по использованию критического ресурса (4) решение о вхождении потоков в их критические секции не откладывается на неопределенный срок, а является конечным во времени (5) предлагаемое решение является корректным и не нарушает ни одного из перечисленных условий	
34.	2,3	Технология программирования OpenMP расширяет язык программирования за счет: (1) новых ключевых слов (2) новых библиотечных функций и переменных окружения (3) новых директив и специальных комментариев	ПК-2
35.	2,3,4	Отметьте верные утверждения об OpenMP: (1) OpenMP ориентирован в первую очередь на написание программ для векторно-конвейерных компьютеров (2) большинство конструкций OpenMP реализуется с помощью директив и/или специальных комментариев (3) все переменные программы делятся на два класса: локальные (private) и общие (shared) (4) число параллельных потоков OpenMP приложения может определяться переменной окружения (5) весь параллелизм приложения реализуется только с помощью параллельных циклов	ПК-2
36.	3	Для сборки OpenMP-программы необходимо. (1) подключить заголовочный файл <code>omp.h</code> (2) подключить библиотеку <code>omp.lib</code> (3) использовать ключ компилятора <code>/Qopenmp (/openmp)</code>	ПК-2
37.	1	Для распределения итераций цикла между потоками необходимо использовать следующую директиву OpenMP: (1) <code>#pragma omp parallel for</code>	ПК-2

		(2) <code>#pragma omp for</code> , если данная директива находится внутри параллельной области (3) <code>#pragma parallel for</code> (4) <code>#pragma omp sections</code>	
38.	2	Можно ли в OpenMP программе управлять количеством потоков выполняющих работу, не меняя кода программы? (1) нет, программа сама определяет количество потоков (2) да, если в программе явно не определяется количество потоков (3) да, используя переменные окружения	ПК-2
39.	1,3	Ваша OpenMP-программа содержит две параллельные области. Необходимо, чтобы число потоков в первой из них равнялось трем, а во второй определялось числом процессоров, доступных операционной системе. Отметьте верные варианты: (1) <code>#pragma omp parallel num_threads(3) ... #pragma omp parallel</code> (2) <code>omp_set_num_threads(3); #pragma omp parallel ... #pragma omp parallel</code> (3) <code>omp_set_num_threads(3); #pragma omp parallel ... set_num_threads(omp_get_num_procs()); #pragma omp parallel</code>	ПК-2
40.	1	В критические секции: (1) потоки входят последовательно в произвольном порядке (2) потоки входят последовательно в соответствии с порядковым номером потока (3) все потоки входят одновременно	ПК-2