

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Рукович Александр Владимирович

Должность: Директор

Дата подписания: 30.05.2025 14:33:21

Уникальный программный ключ:

f45eb7c44954caac05ea7d4f32eb8d7d6b3cb96ae6d9b4bda094afdda1fb705f

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования

«СЕВЕРО-ВОСТОЧНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ ИМЕНИ М.К. АММОСОВА»

Технический институт (филиал) ФГАОУ ВО «СВФУ» в г. Нерюнгри

Кафедра Математики и информатики

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Б1.В.ДВ.04.02 Системное программирование

для программы бакалавриата

по направлению подготовки 01.03.02 «Прикладная математика и информатика»

Направленность (профиль) программы: Системное программирование и компьютерные технологии

Форма обучения: очная

Нерюнгри 2023

УТВЕРЖДЕНО на заседании
выпускающей кафедры

МиИ

« 05 » 05 20 23 г., протокол № 10

Заведующий кафедрой

« 05 » 05 20 23 г.

/ Самохина В.М.

УТВЕРЖДЕНО на заседании
обеспечивающей кафедры

МиИ

« 05 » 05 20 23 г., протокол № 10

Заведующий кафедрой

« 05 » 05 20 23 г.

/ Самохина В.М.

СОГЛАСОВАНО:

Эксперты¹:

Юданова В.В., ст. преподаватель кафедры МиИ

Ф.И.О., должность, организация


подпись

Самохина В.М., к.п.н, доцент кафедры МиИ

Ф.И.О., должность, организация


подпись

СОСТАВИТЕЛЬ (И):

Похорукова М.Ю., к.т.н., доцент кафедры МиИ

Ф.И.О., должность, организация


подпись

¹ Эксперт первый: со стороны выпускающей кафедры (или работодатель). Эксперт второй: со стороны обеспечивающей кафедры.

Паспорт фонда оценочных средств
Б1.В.ДВ.04.02 Системное программирование

№	Контролируемые разделы (темы)	Код контролируемой компетенции (или ее части)	Требования к уровню усвоения компетенции	Наименование оценочного средства
1.	Введение в системное программирование	ОПК-3: способностью к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей,	Знать: принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения; типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения; методы и средства проектирования программного обеспечения; методы и средства проектирования баз данных; методы и средства проектирования программных интерфейсов; методы и приемы формализации и алгоритмизации поставленных задач; алгоритмы решения типовых задач, области и способы их применения; языки программирования, стандартные библиотеки языков программирования; методологии разработки программного обеспечения; особенности выбранной среды программирования и системы управления базами данных; методы и приемы отладки программного кода; типы и форматы сообщений об ошибках. Уметь: использовать существующие типовые решения и шаблоны проектирования программного обеспечения; применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов; осуществлять коммуникации с заинтересованными сторонами; применять стандартные алгоритмы решения задач в соответствующих областях; применять выбранные языки программирования и среды программирования, системы управления базами данных при разработке программного обеспечения; выявлять ошибки в программном коде, использовать современные компиляторы и отладчики программного кода. Владеть: навыками разработки, изменения и согласования архитектуры программного обеспечения с системным аналитиком и архитектором программного обеспечения; навыками проектирования структур данных, баз данных, программных интерфейсов; навыками оценки и согласования сроков выполнения поставленных задач; навыками формализованного описания решений; навыками разработки алгоритмов решения поставленных задач в соответствии с требованиями технического задания; навыками оценки и согласования сроков выполнения поставленных задач; навыками создания программного кода в соответствии с техническим заданием и с использованием специализированных программных средств; навыками анализа и проверки программного кода, его отладки на уровне программных модулей и межмодульных взаимодействий.	лабораторные занятия, самостоятельные работы, аттестационная работа
2.	Особенности выполнения программ. Процессы и потоки	созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям.		
3.	Файловые системы Windows	ПК-4: способностью работать в составе научно-исследовательского и производственного коллектива и решать задачи профессиональной деятельности. ПК-7: способностью к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения		

Лабораторные работы

В период освоения дисциплины студенты посещают лекционные занятия, самостоятельно изучают дополнительный теоретический материал к лабораторным занятиям. Критериями оценки работы на лабораторных занятиях является: полнота и правильность выполненного задания; степень осознанности, понимания изученного; оформление задания.

Содержание отчета.

1. Титульный лист: название дисциплины; номер и наименование работы; фамилия, имя, отчество студента; дата выполнения.

2. Задание.

3. Листинг программы с выполнением задания.

4. Результаты работы программы.

5. Вывод по работе, сформулированные из цели.

Темы лабораторных работ

Тема 1. Введение в системное программирование.

Задания к лабораторной работе

1. Разработать две программы. Первая вычисляет сумму и произведение чисел от L до U , где L – это нижняя граница диапазона, U – верхняя граница диапазона, границы вводятся пользователем, и выводит полученные значения на экран. Вторая программа запускает первую в качестве вновь созданного процесса.

2. Разработать две программы. Первая вычисляет число Фибоначчи по номеру, введенному пользователем, и формуле $F_i = F_{i-1} + F_{i-2}$, $F_0 = F_1 = 1$ и

3. выводит его на экран. Вторая программа запускает первую в качестве вновь созданного процесса.

4. Разработать две программы. Первая принимает от пользователя строку, хранящую знаковое целое число, и выводит на экран строковый эквивалент этого числа прописью (например, ввод «-1211» должен приводить к выводу «минус тысяча двести одиннадцать»). Вторая программа запускает первую в качестве вновь созданного процесса.

5. Разработать две программы. Первая принимает от пользователя строку, хранящую число со знаком и плавающей точкой, и выводит на экран строковый эквивалент этого числа прописью (например, ввод «-12.11» должен приводить к выводу «минус двенадцать целых одиннадцать сотых»). Вторая программа запускает первую в качестве вновь созданного процесса.

6. Разработать две программы. Первая принимает от пользователя две строки. Далее, если обе строки хранят целые числа со знаком, то на экран выводится сумма чисел, в противном случае – конкатенация двух введенных строк. Вторая программа запускает первую в качестве вновь созданного

7. процесса.

8. Разработать две программы. Первая принимает от пользователя две прямоугольных матрицы, а затем выводит на экран их сумму и произведение. Вторая программа запускает первую в качестве вновь созданного процесса.

9. Разработать две программы. Первая принимает от пользователя одномерный целочисленный массив, упорядочивает его по возрастанию любым из так называемых «улучшенных алгоритмов» сортировки массивов и выводит на экран. Вторая программа запускает первую в качестве вновь созданного процесса.

10. Разработать две программы. Первая принимает от пользователя одномерный массив чисел с плавающей точкой, упорядочивает его по убыванию любым из так называемых «улучшенных алгоритмов» сортировки массивов и выводит на экран. Вторая программа запускает первую в качестве вновь созданного процесса.

11. Разработать две программы. Первая принимает от пользователя одномерный массив строк, упорядочивает его любым из так называемых «улучшенных алгоритмов» сортировки массивов и выводит на экран. Вторая программа запускает первую в качестве

вновь созданного процесса.

12. Разработать две программы. Первая принимает от пользователя квадратную матрицу, вычисляет сумму элементов, лежащих на главной и побочной диагоналях, и выводит на экран. Вторая программа запускает первую в качестве вновь созданного процесса.

13. Разработать программу, которая вычисляет сумму и произведение чисел от L до U , где L – это нижняя граница диапазона, U – верхняя граница диапазона. Вычисление суммы и произведения оформить как две функции потока. Значения границ диапазонов вводятся пользователем, затем запускаются два требуемых потока, а потом на экран выводятся полученные значения.

14. Разработать программу, которая вычисляет число Фибоначчи по номеру, введенному пользователю, и формуле $F_i = F_{i-1} + F_{i-2}$, $F_0 = F_1 = 1$. Вычисление числа Фибоначчи оформить как функцию потока. По завершению функции потока программа выводит число на экран.

15. Разработать программу для перевода целого числа со знаком в его строковый эквивалент прописью. Перевод числа оформить как функцию потока. Ввод числа происходит до запуска потока, а вывод строки – по его завершению. Например, ввод «-1211» должен приводить к выводу «минус тысяча двести одиннадцать».

16. Разработать программу для перевода знакового числа с плавающей точкой в его строковый эквивалент прописью. Перевод числа оформить как функцию потока. Ввод числа происходит до запуска потока, а вывод строки – по его завершению. Например, ввод «-12.11» должен приводить к выводу «минус двенадцать целых одиннадцать сотых».

17. Разработать программу, осуществляющую ввод двух строк, введенных пользователем. Далее, если обе строки хранят целые числа со знаком, то на экран выводится сумма чисел, в противном случае – конкатенация двух введенных строк. Проверку на соответствие строки целому числу, вычисление суммы чисел и конкатенации строк оформить как три разных функции потока. Ввод строк осуществляется до запуска всех потоков, а вывод результатов – после их завершения.

18. Разработать программу, вычисляющую сумму и произведение двух матриц. Выполнение этих операций оформить как две функции потока.

19. Сначала программа осуществляет ввод элементов матриц, далее запускает оба потока, а затем выводит результаты на экран.

20. Разработать программу для упорядочивания одномерного целочисленного массива. Сортировка массива по возрастанию должна осуществляться любым из так называемых «улучшенных алгоритмов» сортировки и оформляется как функция потока. Сначала выполняется ввод элементов матрицы, затем запускается поток и далее – вывод упорядоченного массива.

21. Разработать программу для упорядочивания одномерного массива чисел с плавающей точкой. Сортировка массива по возрастанию должна осуществляться любым из так называемых «улучшенных алгоритмов» сортировки и оформляется как функция потока. Сначала выполняется ввод элементов матрицы, затем запускается поток и далее – вывод упорядоченного массива.

22. Разработать программу для упорядочивания одномерного массива строк. Сортировка массива по возрастанию должна осуществляться любым из так называемых «улучшенных алгоритмов» сортировки и оформляется как функция потока. Сначала выполняется ввод элементов матрицы, затем запускается поток и далее – вывод упорядоченного массива.

23. Разработать программу для вычисления суммы элементов, лежащих на главной и побочной диагоналях квадратной матрицы. Выполнение этой операции оформляется как функция потока. Ввод элементов матрицы осуществляется до запуска потока, а вывод полученного значения – по его завершению.

Тема 2. Особенности выполнения программ. Процессы и потоки

Задание №1

В пансионе отдыхают и предаются размышлениям 5 философов, пронумерованные от 1 до 5. В столовой расположен круглый стол, вокруг которого расставлены 5 стульев, также пронумерованные от 1 до 5. На столе находится одна большая тарелка со спагетти, которая пополняется бесконечно, также там расставлены 5 тарелок, в которые накладывается спагетти, и 5 вилок, назначение которых очевидно.

Для того чтобы пообедать, философ входит в столовую и садится на стул со своим номером. При этом есть философ сможет только в том случае, если свободны две вилки – справа и слева от его тарелки. При выполнении этого условия философ поднимает одновременно обе вилки и может поглощать пищу в течение какого-то заданного времени. В противном случае, философу приходится ждать освобождения обеих вилок.

Пообедав, философ кладет обе вилки на стол одновременно и уходит.

Описанный процесс происходит бесконечно.

Воспользоваться объектами синхронизации типа «ожидаемый таймер».

Задание №2

В пансионе отдыхают и предаются размышлениям 5 философов, пронумерованные от 1 до 5. В столовой расположен круглый стол, вокруг которого расставлены 5 стульев, также пронумерованные от 1 до 5. На столе находится одна большая тарелка со спагетти, которая пополняется бесконечно, также там расставлены 5 тарелок, в которые накладывается спагетти, и 5 вилок, назначение которых очевидно.

Для того чтобы пообедать, философ входит в столовую и садится на стул со своим номером. При этом есть философ сможет только в том случае, если свободны две вилки – справа и слева от его тарелки. При выполнении этого условия философ поднимает одновременно обе вилки и может поглощать пищу в течение какого-то заданного времени. В противном случае, философу приходится ждать освобождения обеих вилок.

Пообедав, философ кладет обе вилки на стол одновременно и уходит.

Описанный процесс происходит бесконечно.

Воспользоваться объектами синхронизации типа «событие».

Задание №3

В пансионе отдыхают и предаются размышлениям 5 философов, пронумерованные от 1 до 5. В столовой расположен круглый стол, вокруг которого расставлены 5 стульев, также пронумерованные от 1 до 5. На столе находится одна большая тарелка со спагетти, которая пополняется бесконечно, также там расставлены 5 тарелок, в которые накладывается спагетти, и 5 вилок, назначение которых очевидно.

Для того чтобы пообедать, философ входит в столовую и садится на стул со своим номером. При этом есть философ сможет только в том случае, если свободны две вилки – справа и слева от его тарелки. При выполнении этого условия философ поднимает одновременно обе вилки и может поглощать пищу в течение какого-то заданного времени. В противном случае, философу приходится ждать освобождения обеих вилок.

Пообедав, философ кладет обе вилки на стол одновременно и уходит.

Описанный процесс происходит бесконечно.

Воспользоваться объектами синхронизации типа «семафор».

Задание №4

В пансионе отдыхают и предаются размышлениям 5 философов, пронумерованные от 1 до 5. В столовой расположен круглый стол, вокруг которого расставлены 5 стульев, также пронумерованные от 1 до 5. На столе находится одна большая тарелка со спагетти, которая пополняется бесконечно, также там расставлены 5 тарелок, в которые накладывается спагетти, и 5 вилок, назначение которых очевидно.

Для того чтобы пообедать, философ входит в столовую и садится на стул со своим номером. При этом есть философ сможет только в том случае, если свободны две вилки – справа и слева от его тарелки. При выполнении этого условия философ поднимает

одновременно обе вилки и может поглощать пищу в течение какого-то заданного времени. В противном случае, философу приходится ждать освобождения обеих вилок.

Пообедав, философ кладет обе вилки на стол одновременно и уходит.

Описанный процесс происходит бесконечно.

Воспользоваться объектами синхронизации типа «мьютекс».

Задание №5

В пансионе отдыхают и предаются размышлениям 5 философов, пронумерованные от 1 до 5. В столовой расположен круглый стол, вокруг которого расставлены 5 стульев, также пронумерованные от 1 до 5. На столе находится одна большая тарелка со спагетти, которая пополняется бесконечно, также там расставлены 5 тарелок, в которые накладывается спагетти, и 5 вилок, назначение которых очевидно.

Для того чтобы пообедать, философ входит в столовую и садится на стул со своим номером. При этом есть философ сможет только в том случае, если свободны две вилки – справа и слева от его тарелки. При выполнении этого условия философ поднимает одновременно обе вилки и может поглощать пищу в течение какого-то заданного времени. В противном случае, философу приходится ждать освобождения обеих вилок.

Пообедав, философ кладет обе вилки на стол одновременно и уходит.

Описанный процесс происходит бесконечно.

Воспользоваться объектами синхронизации типа «критическая секция».

Задание №6

В парикмахерской расположено единственное кресло, на котором спит парикмахер, и несколько стульев для клиентов.

Когда клиент приходит в парикмахерскую, он будит парикмахера, садится в кресло. Стрижка производится в течение заданного времени. Если же кресло занято другим клиентом, то вновь прибывший клиент занимает любой свободный стул и ожидает своей очереди (клиенты обслуживаются в порядке очередности, например, времени прибытия). Если все стулья заняты, то клиент поворачивается и уходит.

Когда обслужены все клиенты, парикмахер садится в кресло и снова засыпает.

Описанный процесс происходит бесконечно.

Воспользоваться объектами синхронизации типа «ожидаемый таймер».

Задание №7

В парикмахерской расположено единственное кресло, на котором спит парикмахер, и несколько стульев для клиентов.

Когда клиент приходит в парикмахерскую, он будит парикмахера, садится в кресло. Стрижка производится в течение заданного времени. Если же кресло занято другим клиентом, то вновь прибывший клиент занимает любой свободный стул и ожидает своей очереди (клиенты обслуживаются в порядке очередности, например, времени прибытия). Если все стулья заняты, то клиент поворачивается и уходит.

Когда обслужены все клиенты, парикмахер садится в кресло и снова засыпает. Описанный процесс происходит бесконечно.

Воспользоваться объектами синхронизации типа «событие».

Задание №8

В парикмахерской расположено единственное кресло, на котором спит парикмахер, и несколько стульев для клиентов.

Когда клиент приходит в парикмахерскую, он будит парикмахера, садится в кресло. Стрижка производится в течение заданного времени. Если же кресло занято другим клиентом, то вновь прибывший клиент занимает любой свободный стул и ожидает своей очереди (клиенты обслуживаются в порядке очередности, например, времени прибытия). Если все стулья заняты, то клиент поворачивается и уходит.

Когда обслужены все клиенты, парикмахер садится в кресло и снова засыпает.

Описанный процесс происходит бесконечно.

Воспользоваться объектами синхронизации типа «семафор».

Задание №9

В парикмахерской расположено единственное кресло, на котором спит парикмахер, и несколько стульев для клиентов.

Когда клиент приходит в парикмахерскую, он будит парикмахера, садится в кресло. Стрижка производится в течение заданного времени. Если же кресло занято другим клиентом, то вновь прибывший клиент занимает любой свободный стул и ожидает своей очереди (клиенты обслуживаются в порядке очередности, например, времени прибытия). Если все стулья заняты, то клиент поворачивается и уходит.

Когда обслужены все клиенты, парикмахер садится в кресло и снова засыпает.

Описанный процесс происходит бесконечно.

Воспользоваться объектами синхронизации типа «мьютекс».

Задание №10

В парикмахерской расположено единственное кресло, на котором спит парикмахер, и несколько стульев для клиентов.

Когда клиент приходит в парикмахерскую, он будит парикмахера, садится в кресло. Стрижка производится в течение заданного времени. Если же кресло занято другим клиентом, то вновь прибывший клиент занимает любой свободный стул и ожидает своей очереди (клиенты обслуживаются в порядке очередности, например, времени прибытия). Если все стулья заняты, то клиент поворачивается и уходит.

Когда обслужены все клиенты, парикмахер садится в кресло и снова засыпает.

Описанный процесс происходит бесконечно.

Воспользоваться объектами синхронизации типа «критическая секция».

Тема 3. Файловые системы Windows.

1. Разработать программу, которая позволяет удалять из каталога, указанного пользователем только те файлы, которые не соответствуют указанной пользователем маске.

2. Разработать программу, которая позволяет переименовывать файлы следующим образом. Все буквы русского алфавита из имени файла заменяются на буквы английского алфавита и наоборот. Имя файла указывается пользователем. Заменяются только короткие имена.

3. Разработать программу, которая позволяет инвертировать содержимое и имя файла, указанного пользователем. Под инвертированием понимается запись строки в обратном порядке. Инвертируются только короткие имена файлов.

4. Разработать программу, которая позволяет двум каталогам, указанным пользователем, обмениваться содержимым, то есть подкаталогами и файлами.

5. Разработать программу, которая позволяет переименовывать файлы следующим образом. Все строчные символы в имени файла заменяются на прописные и наоборот. Имя файла указывается пользователем. Заменяются только короткие имена.

6. Разработать программу, которая позволяет перемещать файлы из текущего каталога во вновь создаваемые программой подкаталоги. Короткие имена этих подкаталогов должны быть эквивалентны расширениям перемещаемых файлов.

7. Разработать программу, которая позволяет удалять каталоги, содержащие файлы. Имя каталога указывается пользователем.

8. Разработать программу, которая удаляет из каталога, указанного пользователем, все пустые файлы.

9. Разработать программу, которая перемещает все файлы в указанный пользователем каталог все файлы из его подкаталогов.

10. Некоторые файловые системы могут поддерживать версии файлов. Однако, в ОС семейства Windows этого не предусмотрено. Необходимо разработать программу, которая позволяла бы добавлять номер версии к короткому имени файла через символ подчеркивания. Например, при появлении очередной версии для файла f_1.txt должен быть

создан файл f_2.txt. Содержимое файлов может просто копироваться. Учесть ограничения на длину имени файла и диапазон значений для чисел.

Критерии оценки:

0 баллов - ставится, если студент не выполнил лабораторную работу.

1 балл - ставится, если студент обнаруживает знание и понимание основных положений лабораторной работы, но при выполнении заданий допущены ошибки или задание выполнено на 50%; оформление работы выполнено недостаточно последовательно (отсутствуют цель/листинг/результаты/выводы).

2 балла - ставится, если студентом при выполнении заданий допущены неточности или задание выполнено на 70%; оформление работы выполнено с ошибками (отсутствуют цель/выводы).

3 балла - ставится, если студент полностью выполнил задание, правильно ответил на теоретические вопросы преподавателя, оформление работы выполнено последовательно и полно (присутствуют цели работы, задания, листинг программ, результаты и выводы).

Самостоятельная работа студента

Включает проработку конспектов лекций, обязательной и дополнительной учебной литературы в соответствии с планом занятия; выполнение заданий. Основной формой проверки СРС является устный фронтальный опрос на занятии или письменные ответы на вопросы для проверки знаний по теме.

Темы заданий для самостоятельной работы студентов

Тема 1. Системное программное обеспечение: основные понятия, классификация и структура СПО.

Тема 2. Классификация системных программ.

Тема 3. Объекты ядра: создание, уничтожение, таблица описателей, учет пользователей объектов ядра, наследование.

Тема 4. Процесс выполнения программ: создание, завершение процессов и потоков.

Тема 5. Синхронизация потоков: механизмы синхронизации (семафоры, мониторы, сообщения, барьеры).

Тема 6. Файлы: структура и типы файлов; доступ к файлу; атрибуты файла; операции с файлами.

Тема 7. Реализация файловой системы.

Тема 8. Взаимоблокировки, их обнаружение.

Перечень теоретических вопросов:

1. Что такое программа, программное обеспечение?
2. Какие уровни включает в себя структура ПО?
3. Для чего предназначено системное программное обеспечение?
4. В чем состоит назначение прикладного программного обеспечения?
5. Для чего предназначены инструментальные системы (системы программирования)?
6. Какова структура системного программного обеспечения?
7. Каково назначение базового программного обеспечения?
8. Перечислите основные функции операционной системы.
9. Расскажите о видах интерфейса пользователя, применяемых в разных операционных системах.
10. Что означает «интерфейс командной строки»?
11. Какие виды интерфейса обеспечивает операционная система?
12. Приведите классификацию операционных систем.
13. Что такое файл?
14. Как образуется имя файла и полное имя файла?
15. Приведите примеры общепринятых расширений для популярных типов файлов.
16. Для чего предназначена папка?
17. Дайте представление об иерархической структуре подчинённости папок.
18. Что такое путь или адрес?
19. Приведите символы замещения, используемые в шаблонах при групповых операциях с файлами.
20. Перечислите основные служебные программы.
21. Какие программные средства относятся к утилитам?

Критерии оценки:

0 баллов – самостоятельная работа не выполнена.

1 балл – демонстрирует, лишь поверхностный уровень выполнения работы, в содержании выполнения задания допущены принципиальные ошибки.

2 балла – ставится тогда, когда студент выполнил самостоятельную работу, но дает не точные ответы на заданные вопросы.

3 балла – ставится тогда, когда студент выполнил самостоятельную работу, показан высокий уровень освоения студентом учебного материала, содержание выполнения задания не содержит ошибок.

Аттестационная работа

Аттестационная работа предполагает выполнение письменной работы с обязательными практическими примерами по одной из тем: изучение теоретического материала по заданной теме, самостоятельное рассмотрение примеров, анализ различных источников, формулирование выводов. Работа выполняется печатным способом на листах формата А4, объемом 10-15 страниц. Работа должна быть защищена, то есть студент должен отвечать на вопросы преподавателя по тематике работы. Студент должен подготовить презентацию к работе, в которой кратко дается описание работы с уместным использованием мультимедийных технологий (изображения, видео- или аудио-информация).

Структура работы

Титульный лист

Оглавление

Введение

Содержание работы, состоящее из 2-3 глав, в достаточной мере описывающее предметную область с примерами.

Заключение

Список используемых источников

Приложения (при необходимости)

Тематика аттестационных работ

1. Объекты ядра, процесс выполнения программ.
2. Синхронизация потоков, решение проблем синхронизации.
3. Принципы аппаратуры ввода-вывода, программное обеспечение.
4. Интерфейс операционной системы: основные принципы и стандарты.
5. Межпроцессные взаимодействия: механизмы, каналы, очереди сообщений, разделяемые сегменты памяти, сокеты, вызов удаленных процедур.
6. Драйвера: задачи, классификация и особенности их функционирования, основные свойства и характеристики.
7. Службы, особенности их создания и работы.
8. Подсистема безопасности: цели, защита объектов, аудит, права и привилегии.
9. Процессы и потоки.

Критерии оценки:

0 баллов – контрольная работа не выполнена.

1-10 баллов – демонстрирует, лишь поверхностный уровень выполнения работы, в содержании выполнения задания допущены принципиальные ошибки, на заданные вопросы отвечает нечетко и неполно.

11-34 баллов – ставится тогда, когда студент выполнил контрольную работу, твердо знает материал, но дает не точные ответы на заданные вопросы, в содержании выполнения задания допущены не принципиальные ошибки.

35-40 баллов – ставится тогда, когда студент выполнил контрольную работу, показан высокий уровень освоения студентом учебного материала, содержание выполнения задания не содержит ошибок или допущены неточности, которые были устранены после замечаний, в работе присутствуют четкие и обоснованные выводы.