

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Якушин Владимир Андреевич
Должность: ректор, д.ю.н., профессор
Дата подписания: 06.10.2023
Уникальный программный ключ:
a5427c2559e1ff4b007ed9b1994671e27053e0dc

Министерство науки и высшего образования РФ
Образовательная автономная некоммерческая организация
высшего образования
«Волжский университет имени В.Н. Татищева» (институт)

УТВЕРЖДАЮ

Ректор Якушин В.А.

от 02.05.2023г. № 77/1

Рабочая программа

Системное программное обеспечение

Направление подготовки 09.03.01 Информатика и вычислительная техника

Квалификация (степень) выпускника – бакалавр

Форма обучения – очная, заочная, очно-заочная

Тольятти, 2023 г.

Рабочая программа **Системное программное обеспечение** составлена с требованиями ФГОС, ВО, ОПОП по направлению подготовки 09.03.01 Информатика и вычислительная техника (уровень высшего образования: бакалавриат) и учебного плана.

Программа обсуждена и рекомендована к использованию и (или) изданию решением кафедры на заседании кафедры «Информатика и системы управления»

протокол № 09 от 19.04.2023г.

Зав. кафедрой ИиСУ

к.п.н., доцент Е.Н. Горбачевская

Одобрено Учебно-методическим советом вуза

протокол № 4/23 от 27.04.2023г

Председатель УМС

к.п.н. И.И. Муртаева

1. ПЕРЕЧЕНЬ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

В результате освоения дисциплины у обучающихся должны быть сформированы следующие общепрофессиональные компетенции и профессиональные компетенции:

Наименование компетенции	Код компетенции
Управление программно- аппаратными средствами информационных служб инфокоммуникационной системы организации	ПК-1
Администрирование сетевой подсистемы инфокоммуникационной системы организации	ПК-2

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Данная учебная дисциплина относится к к части, формируемой участниками образовательных отношений 09.03.01 «Информатика и вычислительная техника».

В таблице 1 представлен перечень компетенций с указанием перечня дисциплин, формирующих эти компетенции согласно учебному плану ОПОП

Таблица 1

Код компетенции	Наименование компетенции, формируемой в рамках освоения дисциплины	Предшествующие дисциплины, формирующие указанную компетенцию	Последующие дисциплины, формирующие указанную компетенцию
ПК-1	Управление программно-аппаратными средствами информационных служб инфокоммуникационной системы организации	Моделирование Операционные системы Linux и системы реального времени	Базовые технологии и процессы Интеллектуальные системы и технологии Надежность систем Электронный бизнес Микропроцессорные системы Проектирование вычислительных систем и комплексов Научно исследовательская работа Анализ информационных проектов Корпоративные информационные системы/Конструирование модулей и систем Производственная практика. Технологическая (проектно-технологическая) практика Преддипломная практика Защита выпускной квалификационной работы, включая подготовку к процедуре защиты и процедуру защиты
ПК-2	Администрирование сетевой подсистемы	Операционные системы Linux и	Сети и телекоммуникации Защита информации

	инфокоммуникационной системы организации	системы реального времени	Микропроцессорные системы Проектирование вычислительных систем и комплексов Корпоративные сети/Промышленные сети Производственная практика. Технологическая (проектно- технологическая) практика Преддипломная практика Защита выпускной квалификационной работы, включая подготовку к процедуре защиты и процедуру защиты
--	---	------------------------------	---

* в качестве этапа формирования компетенций используются номера семестров согласно учебного плана ОПОП

Перечень планируемых результатов обучения, соотнесенных с планируемыми результатами освоения основной профессиональной образовательной программы направления подготовки, представлен в таблице:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
ПК-1. Управление программно- аппаратными средствами информационных служб инфокоммуникационной системы организации	ПК-1.1. Планирует архитектуру и функционирование аппаратных, программных и программно-аппаратных средств информационных служб инфокоммуникационной системы организации ПК-1.2. Использует правила и методы технического обслуживания и восстановления аппаратных, программных и программно-аппаратных средств информационных служб инфокоммуникационной системы организации ПК-1.3. Участвует в проектировании программно- аппаратных средств информационных служб инфокоммуникационной системы организации ПК-1.4. Участвует в конфигурировании, управлении, восстановлении работоспособности программно- аппаратных средств информационных служб инфокоммуникационной системы организации ПК-1.5. Использует нормативно-техническую документацию в области инфокоммуникационных технологий
ПК-2. Администрирование сетевой подсистемы инфокоммуникационной системы организации	ПК-2.1. Анализирует принципы функционирования аппаратных, программных и программно-аппаратных средств сетевой подсистемы инфокоммуникационной системы организации ПК-2.2. Проектирует архитектуру аппаратных, программных и программно-аппаратных средств сетевой подсистемы инфокоммуникационной системы организации ПК-2.5. Использует различные методы управления аппаратными, программными и программно-аппаратными средствами сетевой подсистемы инфокоммуникационной системы организации ПК-2.6. Использует современные стандарты, нормативно-техническую документацию в области инфокоммуникационных технологий при администрировании устройств и программного обеспечения ПК-2.7. Участвует в проектировании, конфигурировании и планировании с требуемой производительностью и необходимой безопасностью сетевых подсистем инфокоммуникационной системы организации ПК-2.8. Участвует в настройке, администрировании, восстановлении при сбоях аппаратных, программных и программно-аппаратных средств сетевой подсистемы инфокоммуникационной системы организации ПК-2.9. Выполняет формирование документации по работе с сетевой подсистемой инфокоммуникационной системы организации

3. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

ОЧНАЯ ФОРМА ОБУЧЕНИЯ

Вид учебной работы	Всего	Семестр
		5
Общая трудоёмкость дисциплины	144 час 4 з.е.	144 час 4 з.е.
Контактная работа с преподавателем (всего)	64	64
В том числе:		
Лекции	32	32
Практические / семинарские занятия	-	-
Лабораторные занятия	32	32
Консультации	-	-
Самостоятельная работа (всего)	44	44
В том числе (если есть):		
Курсовой проект / работа	20	20
Расчетно-графическая работа	-	-
Контрольная работа	-	-
Реферат / эссе / доклад	-	-
Иное	24	24
Вид промежуточной аттестации (зачет, экзамен)	Экзамен (36)	Экзамен (36)

ЗАОЧНАЯ ФОРМА ОБУЧЕНИЯ

Вид учебной работы	Всего	Семестр
		7
Общая трудоёмкость дисциплины	144 час 4 з.е.	144 час 4 з.е.
Контактная работа с преподавателем (всего)	12	12
В том числе:		
Лекции	6	6
Практические / семинарские занятия	-	-
Лабораторные занятия	6	6
Консультации	-	-
Самостоятельная работа (всего)	96	96
В том числе (если есть):		
Курсовой проект / работа	20	20
Расчетно-графическая работа	-	-
Контрольная работа	-	-
Реферат / эссе / доклад	-	-
Иное	76	76
Вид промежуточной аттестации (зачет, экзамен)	Экзамен(36)	Экзамен(36)

ОЧНО-ЗАОЧНАЯ ФОРМА ОБУЧЕНИЯ

Вид учебной работы	Всего	Семестр
		7
Общая трудоёмкость дисциплины	144 час 4 з.е.	144 час 4 з.е.
Контактная работа с преподавателем (всего)	24	24
В том числе:		
Лекции	12	12
Практические / семинарские занятия	-	-
Лабораторные занятия	12	12
Консультации	-	-
Самостоятельная работа (всего)	84	84
В том числе (если есть):		
Курсовой проект / работа	20	20
Расчетно-графическая работа	-	-
Контрольная работа	-	-
Реферат / эссе / доклад	-	-
Иное	64	64
Вид промежуточной аттестации (зачет, экзамен)	Экзамен(36)	Экзамен(36)

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1. КАЛЕНДАРНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

ОЧНАЯ ФОРМА ОБУЧЕНИЯ

№ п/п	Тема	Количество часов на			
		лекции	практические /семинарские занятия	лабораторные занятия	самостоятел ьную работу
1	Тема 1. Классификация системного программного обеспечения ПЭВМ Классификация программного обеспечения. Классификация системного программного обеспечения. Назначение, структура, функции операционной системы.	2			2
2	Тема 2. Пользовательский интерфейс операционной среды Режим командной строки. Графический пользовательский			8	2

	интерфейс. Пример современной операционной системы				
3	Тема 3. Основные принципы проектирования системного программного обеспечения Модульность, функциональная избыточность, функциональная избирательность, переносимость, совместимость, мобильность, открытость, защищенность.	2			2
4	Тема 4. Управление процессами Понятие процесса. Планирование и диспетчеризация процессов. Синхронизация процессов. Системные функции управления процессами. Программирование в операционной среде (управление многозадачностью)	2		4	2
5	Тема 5. Управление памятью Иерархия запоминающих устройств. Управление оперативной памятью. Виртуальная память. Свопинг. Виды переменных. Статическое и динамическое распределение памяти. Программирование в операционной среде (управление оперативной памятью)	2		4	2
6	Тема 6. Управление файловой системой Файловая система на диске. Структура логического диска. Методы сохранения	2		4	2

	файлов на диске. Системные функции управления файлами. Программирование в операционной среде (управление файловой системой)				
7	Тема 7. Управление вводом/выводом Понятие ресурса. Виды ресурсов. Управление ресурсами. Устройства ПЭВМ. Виды устройств. Драйверы устройств. Системные функции управления устройствами ввода/вывода (УВВ). Программирование в операционной среде (управление УВВ)	2		4	2
8	Тема 8. Методы программирования Алгоритмическое, модульное, структурное программирование, объектно-ориентированный подход в проектировании СПО (ООП СПО), клиентсерверные технологии, проектирование микро, макроядра ОС. Языки программирования, системы программирования. Макроязыки. Ассемблеры. Программирование в операционной среде (ООП СПО)	2		4	2
9	Тема 9. Трансляция программ. Подключение библиотек. Статическое и динамическое связывание. Компоновка и загрузка программ. Загрузчики. Функции загрузчика. Настраивающий и динамический загрузчики.	2			2
10	Тема 10. Трансляторы и интерпретаторы.	8			3

	Формальные языки и грамматики, типы грамматик; вывод цепочек; конечный и магазинный автоматы, распознаватели и преобразователи, построение автомата по заданной грамматике.				
11	Тема 11. Структура компиляторов и интерпретаторов Лексический, синтаксический и семантический анализаторы. Оптимизация кода. Генерация кода. Распределение памяти.	8			3
12	Курсовая работа				20
Итого		32		32	44

ЗАОЧНАЯ ФОРМА ОБУЧЕНИЯ

№ п/п	Тема	Количество часов на			
		лекции	практические /семинарские занятия	лабораторные занятия	самостоятельную работу
1	Тема 1. Классификация системного программного обеспечения ПЭВМ Классификация программного обеспечения. Классификация системного программного обеспечения. Назначение, структура, функции операционной системы.	0,5			10
2	Тема 2. Пользовательский интерфейс операционной среды Режим командной строки. Графический пользовательский интерфейс. Пример современной операционной системы	0,5		1	10
3	Тема 3. Основные принципы проектирования системного программного обеспечения Модульность, функциональная избыточность, функциональная избирательность, переносимость, совместимость, мобильность, открытость, защищенность.	0,5			10
4	Тема 4. Управление процессами Понятие процесса. Планирование и диспетчеризация	0,5		1	10

	процессов. Синхронизация процессов. Системные функции управления процессами. Программирование в операционной среде (управление многозадачностью)				
5	Тема 5. Управление памятью Иерархия запоминающих устройств. Управление оперативной памятью. Виртуальная память. Свопинг. Виды переменных. Статическое и динамическое распределение памяти. Программирование в операционной среде (управление оперативной памятью)	0,4		1	10
6	Тема 6. Управление файловой системой Файловая система на диске. Структура логического диска. Методы сохранения файлов на диске. Системные функции управления файлами. Программирование в операционной среде (управление файловой системой)	0,5		1	10
7	Тема 7. Управление вводом/выводом Понятие ресурса. Виды ресурсов. Управление ресурсами. Устройства ПЭВМ. Виды устройств. Драйверы устройств. Системные функции управления устройствами ввода/вывода (УВВ). Программирование в операционной среде (управление УВВ)	0,5		1	10
8	Тема 8. Методы программирования Алгоритмическое, модульное, структурное программирование, объектно-ориентированный подход в проектировании СПО (ООП СПО), клиентсерверные технологии, проектирование микро, макроядра ОС. Языки программирования, системы программирования. Макроязыки. Ассемблеры. Программирование	0,5		1	10

	в операционной среде (ООП СПО)				
9	Тема 9. Трансляция программ. Подключение библиотек. Статическое и динамическое связывание. Компоновка и загрузка программ. Загрузчики. Функции загрузчика. Настраивающий и динамический загрузчики.	0,5			10
10	Тема 10. Трансляторы Трансляторы и интерпретаторы. Формальные языки и грамматики, типы грамматик; вывод цепочек; конечный и магазинный автоматы, распознаватели и преобразователи, построение автомата по заданной грамматике.	0,5			10
11	Тема 11. Структура компиляторов и интерпретаторов Лексический, синтаксический и семантический анализаторы. Оптимизация кода. Генерация кода. Распределение памяти.	1			9
	Курсовая работа				20
	Итого	6		6	96

ОЧНО-ЗАОЧНАЯ ФОРМА ОБУЧЕНИЯ

№ п/п	Тема	Количество часов на			
		лекции	практические /семинарские занятия	лабораторные занятия	самостоятельную работу
1	Тема 1. Классификация системного программного обеспечения ПЭВМ Классификация программного обеспечения. Классификация системного программного обеспечения. Назначение, структура, функции операционной системы.	1			7
2	Тема 2. Пользовательский интерфейс операционной среды Режим командной строки. Графический пользовательский интерфейс. Пример современной операционной системы	1		2	7

3	Тема 3. Основные принципы проектирования системного программного обеспечения Модульность, функциональная избыточность, функциональная избирательность, переносимость, совместимость, мобильность, открытость, защищенность.	1			7
4	Тема 4. Управление процессами Понятие процесса. Планирование и диспетчеризация процессов. Синхронизация процессов. Системные функции управления процессами. Программирование в операционной среде (управление многозадачностью)	1		2	7
5	Тема 5. Управление памятью Иерархия запоминающих устройств. Управление оперативной памятью. Виртуальная память. Свопинг. Виды переменных. Статическое и динамическое распределение памяти. Программирование в операционной среде (управление оперативной памятью)	1		2	8
6	Тема 6. Управление файловой системой Файловая система на диске. Структура логического диска. Методы сохранения файлов на диске. Системные функции управления файлами. Программирование в операционной среде (управление файловой системой)	1		2	8
7	Тема 7. Управление вводом/выводом Понятие ресурса. Виды ресурсов. Управление ресурсами. Устройства ПЭВМ. Виды устройств. Драйверы устройств. Системные функции управления устройствами ввода/вывода (УВВ). Программирование в операционной среде (управление УВВ)	1		2	8

8	Тема 8. Методы программирования Алгоритмическое, модульное, структурное программирование, объектно-ориентированный подход в проектировании СПО (ООП СПО), клиентсерверные технологии, проектирование микро, макроядра ОС. Языки программирования, системы программирования. Макроязыки. Ассемблеры. Программирование в операционной среде (ООП СПО)	1		2	8
9	Тема 9. Трансляция программ. Подключение библиотек. Статическое и динамическое связывание. Компоновка и загрузка программ. Загрузчики. Функции загрузчика. Настраивающий и динамический загрузчики.	1			8
10	Тема 10. Трансляторы Трансляторы и интерпретаторы. Формальные языки и грамматики, типы грамматик; вывод цепочек; конечный и магазинный автоматы, распознаватели и преобразователи, построение автомата по заданной грамматике.	1			8
11	Тема 11. Структура компиляторов и интерпретаторов Лексический, синтаксический и семантический анализаторы. Оптимизация кода. Генерация кода. Распределение памяти.	2			8
	Курсовая работа				20
	<i>Итого</i>	12		12	84

4.2. КРАТКОЕ СОДЕРЖАНИЕ ЛЕКЦИОННОГО КУРСА

Раздел 1 Введение

Тема 1. Классификация системного программного обеспечения ПЭВМ.

Классификация обеспечения ПЭВМ, классификация программного обеспечения ПЭВМ, классификация системного программного обеспечения. Анализ основной категории СПО – операционных систем: назначение, структура, функции операционной системы, обзор современных ОС.

Контрольные вопросы

1. Классифицируйте обеспечение ПЭВМ.
2. Классифицируйте программное обеспечение ПЭВМ.
3. Классифицируйте системное программное обеспечение.
4. Представьте перечень современных ОС, их особенности проектирования, функционирования, применения.
5. Классифицируйте ОС.
6. Классифицируйте языки программирования.
7. Классифицируйте системы программирования.
8. Охарактеризуйте методы алгоритмического, структурного, функционального программирования.
9. Охарактеризуйте методы нисходящего проектирования, визуального проектирования, среды быстрого проектирования.
10. Охарактеризуйте методы объектно-ориентированного программирования.

Тема 2. Интерфейсы.

Пользовательский интерфейс операционной среды. Программный интерфейс. Режим командной строки. Графический пользовательский интерфейс. Пример современной операционной системы (работа в режиме командной строки, с использованием графического интерфейса – на примере ОС семейства Linux).

Контрольные вопросы

1. Виды пользовательского интерфейса с краткой характеристикой.
2. Основные команды, используемые в написании bat-файлов.
3. Приведите примеры bat-файлов.
4. Приведите примеры работы с переменными при использовании shell.
5. Основные команды, используемые в написании shell-скриптов.
6. Приведите примеры shell-скриптов.
7. Что такое программный интерфейс?
8. Какие используются в современных ОС механизмы для обеспечения совместимости, взаимодействия системных программных модулей?
9. Какие элементы используются графическим интерфейсом?
10. В чем преимущества и недостатки графического интерфейса?

Тема 3. Основные принципы проектирования операционных систем, системного ПО.

Принцип модульности, функциональной избирательности, функциональной избыточности, переносимости, совместимости, безопасности и др.

Контрольные вопросы

1. В чем заключается принцип модульности? В чем его основное преимущество?
2. Какие виды программных модулей используются в системном программировании?
3. В чем преимущества реентерабельных модулей? В чем сложность их проектирования?
4. В чем заключается принцип функциональной избирательности? В чем его основное преимущество? Приведите пример.
5. В чем заключается принцип функциональной избыточности? В чем его основное преимущество? Приведите пример.
6. В чем заключается принцип переносимости, совместимости? Какими методами обеспечивается переносимость и совместимость современных ПП?
7. Какими методами обеспечивается безопасность информации современных систем?
8. Перечислите основные уровни, определяющие безопасность вычислительных машин, сетей, систем, комплексов.
9. В чем суть избирательного контроля доступа к информационным ресурсам?
10. Какие инструментальные средства используются для разработки системных ПМ?

Раздел 2 Основные подходы проектирования операционных системы

Тема 4. Управление процессами.

Понятие процесса. Способы диспетчеризации процессов. Синхронизация процессов. Порождение процессов, изменение приоритетности процессов, уничтожение процессов. Использование семафоров, сообщений для решения задач взаимного исключения и синхронизации. Тупики. Способы борьбы с тупиками. Системные функции управления процессами. Программирование в операционной среде (управление многозадачностью).

Контрольные вопросы

1. Дайте определение процесса, потока. Что означает, что запущен процесс?
2. Какие задачи выполняет диспетчер и планировщик процесса?
3. Какие используются алгоритмы выполнения задач в многозадачных средах?
4. Какие данные сохраняются в таблице дескрипторов процесса?
5. Что такое синхронизация процессов? С какой целью необходимо синхронизировать процессы в многозадачных средах?
6. Системные функции управления процессами (на примере ОС Linux). Приведите примеры.
7. Приведите примеры команд (на примере ОС Linux) для управления процессами (просмотра запущенных процессов, используемых ими ресурсов и пр.).
8. Приведите пример программной реализации при решении задачи взаимного исключения.
9. Приведите пример программной реализации при решении задачи «читатель-писатель».
10. Какие методы используются для предотвращения и распознавания тупиковых ситуаций?

Тема 5. Управление памятью

Методы управления оперативной памятью. Методы управления оперативной памятью без использования дискового пространства, с использованием дискового пространства. Виртуальная память. Свопинг. Способы организации памяти. Методы совместного использования памяти. Статическое и динамическое распределение памяти. Программирование в операционной среде (управление оперативной памятью).

Контрольные вопросы

1. Что такое виртуальное адресное пространство процесса?
2. Что такое виртуальный ресурс? В чем преимущества организации виртуальных ресурсов?
3. Какие используются методы организации виртуальной памяти?
4. Какие информационные структуры генерируются в ОП при реализации методов виртуальной памяти?
5. Как осуществляется загрузка кода программы в ОП?
6. Какие функции выполняет загрузчик программы?
7. Как распределяется адресное пространство памяти при выполнении программы?
8. Системные вызовы управления памятью. Приведите пример программной реализации при статическом и динамическом распределении памяти?
9. Приведите примеры команд, использующихся при работе с памятью, например, определение свободного, занятого дискового пространства.
10. Приведите примеры программ, использующих доступ к участкам ОП, например, работа с указателями (на примере C, C++).

Тема 6. Управление файловой системой

Файловая система на диске. Структура логического диска. Структура логического диска в MsDos, Linux. Системные функции управления файлами. Программирование в операционной среде (управление файловой системой)

Контрольные вопросы

1. Структура ЖД.
2. Какие операции выполняются на этапе низкоуровневого форматирования?
3. Какие операции выполняются на этапе высокоуровневого форматирования?
4. Структура логического диска в MsDos.
5. Структура логического диска Linux.
6. Монтируемые ФС. Их преимущества. Примеры монтирования.
7. Методы проверки файловых систем.
8. Основные команды для работы с ФС (на примере ОС Linux).
9. Системные функции управления файлами.
10. Приведите примеры программ, позволяющих работать с файлами: Создавать; Открывать; Читать; Писать; Осуществлять поиск и пр.

Тема 7. Управление вводом/выводом

Понятие ресурса. Виды ресурсов. Управление ресурсами. Устройства ПЭВМ. Виды устройств. Блокориентированные, байториентированные устройства. Драйверы устройств. Системные функции управления устройствами. Системные функции управления устройствами ввода/вывода (УВВ). Программирование в операционной среде (управление УВВ).

Контрольные вопросы

1. Дайте определение ресурса в контексте ОС и СПО.
2. Перечислите основные системные ресурсы.
3. Какие методы используются современными ОС для эффективного управления ресурсами.
4. Перечислите основные устройства ПЭВМ.
5. Перечислите основные периферийные устройства, использующиеся современными ВМ.
6. Охарактеризуйте блокориентированные, байториентированные устройства.
7. Что такое драйверы устройств? Как они загружаются, запускаются?
8. Какие используются инструментальные средства разработки драйверов устройств?
9. Системные функции управления устройствами ввода/вывода (УВВ). Приведите примеры программ.
10. Основные команды для работы с блокориентированными, байториентированными устройствами. Приведите примеры команд и программ, на примере ОС Linux.

Тема 8. Методы программирования

Алгоритмическое, модульное, структурное программирование, объектноориентированный подход в проектировании СПО (ООП СПО), клиент-серверные технологии, проектирование микро, макроядра ОС. Языки программирования, системы программирования. Макроязыки. Ассемблеры. Программирование в операционной среде (ООП СПО). Сравнение процедурного и объектноориентированного программирования. Абстракция. Инкапсуляция. Скрытие информации. Классы. Конструкторы. Деструкторы. Способы объявления и использования. Наследование. Полиморфизм. Виртуальные функции. Дружественные функции.

Контрольные вопросы

1. В чем особенности алгоритмического, модульного, структурного программирования.
2. Объектно-ориентированный подход в проектировании СПО (ООП СПО). Достоинства, недостатки.
3. Охарактеризуйте метод нисходящего проектирования. Приведите пример модели.
4. Охарактеризуйте клиент-серверные технологии при проектировании СПО.
5. Основные подходы при проектировании микро, макроядра ОС.
6. Приведите примеры языков и систем программирования.
7. Проведите сравнение процедурного и объектно-ориентированного программирования.

8. Основные понятия ООП: Абстракция. Скрытие информации. Классы. Конструкторы. Деструкторы. Способы объявления и использования. Примеры.
9. Охарактеризуйте основные свойства ООП: Инкапсуляция. Наследование. Полиморфизм. Примеры.
10. Примеры команд в написании shell-скриптов.

Раздел 3 Основы проектирования трансляторов

Тема 9. Трансляция программ

Подключение библиотек. Статическое и динамическое связывание. Компоновка и загрузка программ. Загрузчики. Функции загрузчика. Настраивающий и динамический загрузчики.

Контрольные вопросы

1. Трансляция: компиляция и интерпретация.
2. Алгоритм формирования исполняемого кода программы.
3. Библиотеки. Преимущества использования. Способы создания.
4. Методы подключения библиотек.
5. Статическое и динамическое распределение ОП.
6. Статическое и динамическое связывание ПМ.
7. Компоновка программ
8. Загрузка программ.
9. Функции загрузчика.
10. Настраивающий и динамический загрузчики.

Тема 10. Трансляторы

Трансляторы. Интерпретаторы и компиляторы. Формальные языки и грамматики, типы грамматик; вывод цепочек; конечный и магазинный автоматы, распознаватели и преобразователи, построение автомата по заданной грамматике.

Контрольные вопросы

1. Трансляция программ.
2. Формальные языки и грамматики.
3. Типы грамматик.
4. Основные понятия, используемые при построении компиляторов: алфавит, язык, лексема, цепочка, предложение, синтаксис, семантика, семиотика, грамматика и пр.
5. Методы вывода цепочек.
6. Конечный и магазинный автоматы.
7. Распознаватели и преобразователи.
8. Форма Бэкуса-Наура при проектировании грамматики языка.
9. Приведите пример построения грамматики для языка (например, для десятичных чисел, двоичного кода).
10. Алгоритм построения автомата по заданной грамматике.

Тема 11. Структура компиляторов и интерпретаторов

Лексический анализатор. Основные функции. Лексемы. Распознавание лексем. Построение таблиц идентификаторов. Построение дескрипторного текста. Синтаксический анализатор. Основные функции синтаксического анализатора. Построение абстрактного синтаксического дерева вывода. Семантический анализатор. Соглашения языков программирования. Оптимизация кода. Генерация кода. Распределение памяти.

Контрольные вопросы

1. Лексический анализатор. Основные функции.
2. Лексемы. Распознавание лексем.
3. Построение таблиц идентификаторов. Построение дескрипторного текста.
4. Синтаксический анализатор. Основные функции синтаксического анализатора.
5. Построение абстрактного синтаксического дерева вывода.
6. Семантический анализатор. Соглашения языков программирования.

7. Оптимизация кода.
8. Генерация кода.
9. Распределение памяти.
10. Пример компиляции программы.

4.3. ТЕМАТИКА ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Раздел, тема дисциплины	Номер и тема лабораторной работы
Тема 2. Пользовательский интерфейс операционной среды Режим командной строки. Графический пользовательский интерфейс. Пример современной операционной системы.	Лабораторная работа № 1. Пользовательский интерфейс операционной среды: разработка конфигурационных файлов (MsDos, Windows, Linux).
Тема 4. Управление процессами Понятие процесса. Планирование и диспетчеризация процессов. Синхронизация процессов. Системные функции управления процессами. Программирование в операционной среде (управление многозадачностью)	Лабораторная работа № 2. Управление процессами (разработка конфигурационных файлов (в ОС семейства Linux) с использованием команд ps, top, nice, kill и др.; си- программ (на примере СИ для Linux): fork, execl, wait, nice, exit и др.).
Тема 5. Управление памятью Иерархия запоминающих устройств. Управление оперативной памятью. Виртуальная память. Свопинг. Виды переменных. Статическое и динамическое распределение памяти. Программирование в операционной среде (управление оперативной памятью)	Лабораторная работа № 3. Управление памятью (разработка конфигурационных файлов (в ОС семейства Linux) с использованием конструкций языка shell, разработка си- программ с использованием локальных, глобальных переменных, классов extern, static, auto, register; указателей, указателей на массивы, указателей на структуры; функций динамического распределения памяти: malloc, calloc, free, new, delete).
Тема 6. Управление файловой системой Файловая система на диске. Структура логического диска. Методы сохранения файлов на диске. Системные функции управления файлами. Программирование в операционной среде (управление файловой системой).	Лабораторная работа № 4. Управление файлами (разработка конфигурационных файлов (в ОС семейства Linux) с использованием конструкций языка shell, разработка си-программ с использованием функций работы с файлами: creat, open, read, write, access, lseek и др.)
Тема 7. Управление вводом/выводом Понятие ресурса. Виды ресурсов. Управление ресурсами. Устройства ПЭВМ. Виды устройств. Драйверы устройств. Системные функции управления устройствами ввода/вывода (УВВ). Программирование в операционной среде (управление УВВ)	Лабораторная работа № 5. Управление вводом/выводом.
Тема 8. Методы программирования Алгоритмическое, модульное, структурное программирование, объектно-ориентированный подход в проектировании СПО (ООП СПО), клиент-серверные технологии, проектирование микро, макроядра ОС. Языки программирования,	Лабораторная работа № 6. Принципы проектирования операционных систем. Объектно-ориентированный подход в проектировании системных программ (СИ++).

системы программирования. Макроязыки. Ассемблеры. Программирование в операционной среде (ООП СПО)	
Тема 11. Структура компиляторов и интерпретаторов Лексический, синтаксический и семантический анализаторы. Оптимизация кода. Генерация кода. Распределение памяти.	Лабораторная работа № 7. Компиляция, компоновка, загрузка программ. Оптимизация кода.

5. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

5.1 Основная литература

Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем : учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2022. — 432 с. — (Высшее образование). — ISBN 978-5-534-07604-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/491029>

Гостев, И. М. Операционные системы : учебник и практикум для вузов / И. М. Гостев. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2022. — 164 с. — (Высшее образование). — ISBN 978-5-534-04520-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/490157>

5.2 Дополнительная литература

Бабищев, С. Л. Распределенные системы : учебное пособие для вузов / С. Л. Бабищев, К. А. Коньков. — Москва : Издательство Юрайт, 2022. — 507 с. — (Высшее образование). — ISBN 978-5-534-11380-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/476142>

5.3. Ресурсы информационно-коммуникационной сети «Интернет»

Адрес Интернет ресурса	Название Интернет ресурса	Режим доступа
http://intuit.ru/	Интернет-университет информационных технологий	Свободный
http://vkit.ru/	Сайт журнала «Вестник компьютерных и информационных технологий»	Свободный
http://ru.wikipedia.org/	Свободная общедоступная мультязычная универсальная интернет-энциклопедия	Свободный

6. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ

Дисциплина «Системное программное обеспечение» изучается в течение одного семестра. При планировании и организации времени, необходимого на изучение обучающимся дисциплины, необходимо придерживаться следующих рекомендаций.

В период между сессиями студенты должны вести конспект лекций, изучать теоретический материал в соответствии с программой курса, выполнять предложенные преподавателем задания для самостоятельной работы, готовиться к сдаче зачета и экзамена,

прорабатывая необходимый материал согласно перечню вопросов для подготовки к зачету и экзамену и списку рекомендованной литературы.

Выполнение лабораторных работ относится к числу обязательных видов работ. Перед выполнением работы необходимо внимательно ознакомиться с теоретическим материалом, представленным в методических указаниях к соответствующей лабораторной работе. При необходимости можно воспользоваться рекомендуемой литературой. В ходе выполнения работы необходимо руководствоваться порядком выполнения лабораторной работы и указаниями преподавателя, при этом должны соблюдаться правила техники безопасности. Результатом выполнения работы является отчет, который должен быть аккуратно оформлен и выполнен в соответствии с требованиями, приведенными в методических указаниях.

В указанное преподавателем время обучающиеся защищают отчеты. Защита проводится в виде собеседования по контрольным вопросам, приведенным в методических указаниях. Кроме того, преподаватель может задавать дополнительные вопросы, касающиеся результатов эксперимента, выводов по результатам опытов и т.п. К промежуточной аттестации допускаются обучающиеся, выполнившие все лабораторные работы и защитившие отчеты по ним. При наличии задолженности по лабораторным работам, по согласованию с преподавателем, возможна замена работы по выполнению отчета на реферат по теме соответствующего лабораторного занятия с последующей его защитой.

В семестре изучения дисциплины учебным планом предусмотрен курсовой проект. При получении задания, необходимо внимательно с ним ознакомиться и, в случае возникновения вопросов, задать их преподавателю. Регулярное посещение консультаций, внимательное изучение методических указаний к выполнению курсового проекта, а так же строгое соблюдение графика выполнения проекта позволит избежать ненужных проблем. Оценка за курсовой проект выставляется по результатам его защиты.

В течение семестра и во время сессии основным видом подготовки являются самостоятельные занятия. Они включают в себя изучение вопросов, вынесенных на самостоятельное изучение, оформление отчетов по лабораторным работам, курсовое проектирование, а так же подготовку к промежуточной аттестации.

Систематическая работа в соответствии с программой дисциплины – условие успешного освоения материала.

Методические рекомендации по обучению лиц с ограниченными возможностями здоровья и инвалидов

Освоение дисциплины обучающимися с ограниченными возможностями здоровья может быть организовано как совместно с другими обучающимися, так и в отдельных группах. Предполагаются специальные условия для получения образования обучающимися с ограниченными возможностями здоровья.

Профессорско-педагогический состав знакомится с психолого-физиологическими особенностями обучающихся инвалидов и лиц с ограниченными возможностями здоровья, индивидуальными программами реабилитации инвалидов (при наличии). При необходимости осуществляется дополнительная поддержка преподавания тьюторами, психологами, социальными работниками, прошедшими подготовку ассистентами.

В соответствии с методическими рекомендациями Минобрнауки РФ (утв. 8 апреля 2014 г. N АК-44/05вн) в курсе предполагается использовать социально-активные и рефлексивные методы обучения, технологии социокультурной реабилитации с целью оказания помощи в установлении полноценных межличностных отношений с другими студентами, создании комфортного психологического климата в студенческой группе. Подбор и разработка учебных материалов производятся с учетом предоставления материала в различных формах: аудиальной, визуальной, с использованием специальных технических средств и информационных систем.

Медиа материалы также следует использовать и адаптировать с учетом индивидуальных особенностей обучения лиц с ОВЗ.

Освоение дисциплины лицами с ОВЗ осуществляется с использованием средств обучения общего и специального назначения (персонального и коллективного использования). Материально-техническое обеспечение предусматривает приспособление аудиторий к нуждам лиц с ОВЗ.

Форма проведения аттестации для студентов-инвалидов устанавливается с учетом индивидуальных психофизических особенностей. Для студентов с ОВЗ предусматривается доступная форма предоставления заданий оценочных средств, а именно:

- в печатной или электронной форме (для лиц с нарушениями опорно-двигательного аппарата);
- в печатной форме или электронной форме с увеличенным шрифтом и контрастностью (для лиц с нарушениями слуха, речи, зрения);
- методом чтения ассистентом задания вслух (для лиц с нарушениями зрения).

Студентам с инвалидностью увеличивается время на подготовку ответов на контрольные вопросы. Для таких студентов предусматривается доступная форма предоставления ответов на задания, а именно:

- письменно на бумаге или набором ответов на компьютере (для лиц с нарушениями слуха, речи);
- выбором ответа из возможных вариантов с использованием услуг ассистента (для лиц с нарушениями опорно-двигательного аппарата);
- устно (для лиц с нарушениями зрения, опорно-двигательного аппарата).

При необходимости для обучающихся с инвалидностью процедура оценивания результатов обучения может проводиться в несколько этапов.

7. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ

При проведении занятий по дисциплине используются следующие программные продукты:

Windows (для академических организациях, лицензия Microsoft Imagine (ранее MSDNAA, DreamSpark);

Linux (свободное ПО)

OpenOffice (свободное ПО)

8. НЕОБХОДИМАЯ МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА

Аудитория для проведения занятий лекционного типа, занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации.

Перечень основного оборудования:

ауд.Б-509

офисная мебель на 18 мест, 8 ПК с доступом в Интернет и ЭИОС, демонстрационное оборудование: проектор – 1 шт.; экран, рабочее место преподавателя, доска ученическая.

Разработчик:

Кафедра ИиСУ

(место работы)

**Доцент кафедры
ИиСУ**

(занимаемая должность)

Е.Н. Горбачевская

(инициалы, фамилия)

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ОБРАЗОВАТЕЛЬНАЯ АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОЛЖСКИЙ УНИВЕРСИТЕТ имени В.Н. ТАТИЩЕВА» (институт)**

Фонд оценочных средств

«Системное программное обеспечение»

для направления подготовки

09.03.01 «Информатика и вычислительная техника»

Квалификация (степень) выпускника – бакалавриат

1 Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

Оценочные средства разработаны для оценки профессиональных компетенций: ПК-1, ПК-2.

1. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Планируемые результаты освоения основной профессиональной образовательной программы (ОПОП) – компетенции обучающихся определяются требованиями стандарта по направлению подготовки (специальности) и формируются в соответствии с матрицей компетенций ОПОП (Таблица 2)

Планируемые результаты обучения по дисциплине – знания, умения, навыки и (или) опыт деятельности, характеризующие этапы формирования компетенций и обеспечивающие достижение планируемых результатов освоения образовательной программы, формируются в соответствии с картами компетенций ОПОП.

Таблица 1

Планируемые результаты обучения по дисциплине

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
ПК-1. Управление программно- аппаратными средствами информационных служб инфокоммуникационной системы организации	ПК-1.1. Планирует архитектуру и функционирование аппаратных, программных и программно-аппаратных средств информационных служб инфокоммуникационной системы организации ПК-1.2. Использует правила и методы технического обслуживания и восстановления аппаратных, программных и программно-аппаратных средств информационных служб инфокоммуникационной системы организации ПК-1.3. Участвует в проектировании программно- аппаратных средств информационных служб инфокоммуникационной системы организации ПК-1.4. Участвует в конфигурировании, управлении, восстановлении работоспособности программно- аппаратных средств информационных служб инфокоммуникационной системы организации ПК-1.5. Использует нормативно-техническую документацию в области инфокоммуникационных технологий
ПК-2. Администрирование сетевой подсистемы инфокоммуникационной системы организации	ПК-2.1. Анализирует принципы функционирования аппаратных, программных и программно-аппаратных средств сетевой подсистемы инфокоммуникационной системы организации ПК-2.2. Проектирует архитектуру аппаратных, программных и программно-аппаратных средств сетевой подсистемы инфокоммуникационной системы организации ПК-2.5. Использует различные методы управления аппаратными, программными и программно-аппаратными средствами сетевой подсистемы инфокоммуникационной системы организации ПК-2.6. Использует современные стандарты, нормативно-техническую документацию в области инфокоммуникационных технологий при администрировании устройств и программного обеспечения ПК-2.7. Участвует в проектировании, конфигурировании и планировании с требуемой производительностью и необходимой безопасностью сетевых подсистем инфокоммуникационной системы организации ПК-2.8. Участвует в настройке, администрировании, восстановлении при сбоях аппаратных, программных и программно-аппаратных средств сетевой подсистемы инфокоммуникационной системы организации

	ПК-2.9. Выполняет формирование документации по работе с сетевой подсистемой инфокоммуникационной системы организации
--	--

2 Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Результаты обучения по дисциплине «Системное программное обеспечение» направления подготовки 09.03.01 «Информатика и вычислительная техника» определяются показателями и критериями оценивания сформированности компетенций на этапах их формирования представлены в табл. 2.

Таблица 2

Матрица соответствия оценочных средств запланированным результатам обучения

Компетенции	Оценочные средства		
	Текущий контроль		Промежуточный контроль
	Оценочное средство 1 (лабораторное задания)	Оценочное средство 2	Экзамен
ПК-1	ПК-1.1. ПК -1.2. ПК -1.3. ПК -1.4. ПК -1.5.		ПК-1.1. ПК -1.2. ПК -1.3. ПК -1.4. ПК -1.5.
ПК-2	ПК-2.1. ПК -2.2. ПК -2.5. ПК -2.6. ПК -2.7. ПК -2.8. ПК -2.9.		ПК-2.1. ПК -2.2. ПК -2.5. ПК -2.6. ПК -2.7. ПК -2.8. ПК -2.9.

Показатели и критерии оценивания сформированности компетенций (промежуточного контроля)

На этапе промежуточной аттестации используется система оценки успеваемости обучающихся, которая позволяет преподавателю оценить уровень освоения материала обучающимися. Критерии оценивания сформированности планируемых результатов обучения (дескрипторов) представлены в карте компетенции ОПОП.

Форма оценки знаний: оценка - 5 «отлично»; 4 «хорошо»; 3 «удовлетворительно»; 2 «неудовлетворительно». Лабораторные работы, практические занятия, практика оцениваются: «зачет», «незачет». Возможно использование балльно-рейтинговой оценки.

Шкала оценивания:

«Зачет» – выставляется, если сформированность заявленных дескрипторов компетенций на _51_% и более оценивается не ниже «удовлетворительно» при условии отсутствия критерия «неудовлетворительно». Выставляется, когда обучающийся показывает хорошие знания изученного учебного материала; самостоятельно, логично и последовательно излагает и интерпретирует материалы учебного курса; полностью раскрывает смысл предлагаемого вопроса; владеет основными терминами и понятиями изученного курса; показывает умение переложить теоретические знания на предполагаемый практический опыт.

«Отлично» – выставляется, если сформированность заявленных дескрипторов компетенций

85% более (в соответствии с картами компетенций ОПОП) оценивается критериями «хорошо» и «отлично», при условии отсутствия оценки «неудовлетворительно»: студент показал прочные знания основных положений фактического материала, умение самостоятельно решать конкретные практические задачи повышенной сложности, свободно использовать справочную литературу, делать обоснованные выводы из результатов анализа конкретных ситуаций;

«Хорошо» – выставляется, если сформированность заявленных дескрипторов компетенций на 61% и более (в соответствии с картами компетенций ОПОП) оценивается критериями «хорошо» и «отлично», при условии отсутствия оценки «неудовлетворительно», допускается оценка «удовлетворительно»: обучающийся показал прочные знания основных положений фактического материала, умение самостоятельно решать конкретные практические задачи, предусмотренные рабочей программой, ориентироваться в рекомендованной справочной литературе, умеет правильно оценить полученные результаты анализа конкретных ситуаций;

«Удовлетворительно» – выставляется, если сформированность заявленных дескрипторов компетенций 51% и более (в соответствии с картами компетенций ОПОП) оценивается критериями «удовлетворительно», «хорошо» и «отлично»: обучающийся показал знание основных положений фактического материала, умение получить с помощью преподавателя правильное решение конкретной практической задачи из числа предусмотренных рабочей программой, знакомство с рекомендованной справочной литературой;

«Неудовлетворительно» «Незачет» – выставляется, если сформированность заявленных дескрипторов компетенций менее чем 51% (в соответствии с картами компетенций ОПОП): при ответе обучающегося выявились существенные пробелы в знаниях основных положений фактического материала, неумение с помощью преподавателя получить правильное решение конкретной практической задачи из числа предусмотренных рабочей программой учебной дисциплины.

Ответы и решения обучающихся оцениваются по следующим общим критериям: распознавание проблем; определение значимой информации; анализ проблем; аргументированность; использование стратегий; творческий подход; выводы; общая грамотность.

Соответствие критериев оценивания сформированности планируемых результатов обучения (дескрипторов) системам оценок представлено в табл.

Таблица 4

Интегральная оценка

Критерии	Традиционная оценка	Балльно-рейтинговая оценка
5	5	86 - 100
4	4	61-85
3	3	51-60
2 и 1	2, Незачет	0-50
5, 4, 3	Зачет	51-100

Обучающиеся обязаны сдавать все задания в сроки, установленные преподавателем. Оценка «Удовлетворительно» по дисциплине, может выставляться и при неполной сформированности компетенций в ходе освоения отдельной учебной дисциплины, если их формирование предполагается продолжить на более поздних этапах обучения, в ходе изучения других учебных дисциплин.

Показатели и критерии оценки достижений студентом запланированных результатов освоения дисциплины в ходе текущего контроля и промежуточной аттестации

Оценка,	Критерии
---------	----------

уровень	
«отлично», повышенный уровень	Студент показал прочные знания основных положений фактического материала, умение самостоятельно решать конкретные практические задачи повышенной сложности, свободно использовать справочную литературу, делать обоснованные выводы из результатов анализа конкретных ситуаций
«хорошо», пороговый уровень	Студент показал прочные знания основных положений фактического материала, умение самостоятельно решать конкретные практические задачи, предусмотренные рабочей программой, ориентироваться в рекомендованной справочной литературе, умеет правильно оценить полученные результаты анализа конкретных ситуаций
«удовлетворительно», пороговый уровень	Студент показал знание основных положений фактического материала, умение получить с помощью преподавателя правильное решение конкретной практической задачи из числа предусмотренных рабочей программой, знакомство с рекомендованной справочной литературой
«неудовлетворительно», уровень не сформирован	При ответе студента выявились существенные пробелы в знаниях основных положений фактического материала, неумение с помощью преподавателя получить правильное решение конкретной практической задачи из числа предусмотренных рабочей программой учебной дисциплины

3 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

3.1 Перечень вопросов для промежуточной аттестации (экзамен)

1. Классификация системного программного обеспечения ВМ.
2. Основные принципы проектирования СПО: модульность, избирательность, функциональная избыточность, мобильность, переносимость, безопасность вычислений.
3. Принципы проектирования ядер современных ОС.
4. Управление задачами. Контекст и дескриптор процесса.
5. Управление задачами. Синхронизация процессов.
6. Управление задачами. Семафоры. Использование семафоров для решения задач взаимного исключения и синхронизации.
7. Управление задачами. Тупики. Способы борьбы с тупиками.
8. Управление задачами. Системные вызовы работы с процессами (на примере ОС семейства Linux).
9. Управление памятью. Способы организации памяти. Виртуальная память.
10. Управление файлами. Системные вызовы управления файлами.
11. Программирование в операционной среде (разработка программных модулей (ОС семейства Linux /Linux, язык Shell).
12. Классификация языков программирования. Трансляторы.
13. Формальные языки и грамматики. Типы языков и грамматик.
14. Структура компиляторов и интерпретаторов.
15. Алфавит, лексема, цепочки вывода, синтаксис, семантика языка.
16. Лексический анализатор.
17. Синтаксический анализатор.
18. Семантический анализатор.
19. Оптимизация кода при компиляции.
20. Генерация кода при компиляции.
21. Распределение оперативной памяти при компиляции.
22. Виды переменных. Статическое и динамическое распределение памяти.

23. Методы подключения библиотек. Статическое и динамическое связывание.
 24. Компоновка и загрузка программ.
 25. Функции загрузчиков. Настраиваемый и перемещаемый загрузчики.

3.3 Оценочное средство 1 (лабораторное задания)

Раздел, тема дисциплины	Номер и тема лабораторной работы
Тема 2. Пользовательский интерфейс операционной среды Режим командной строки. Графический пользовательский интерфейс. Пример современной операционной системы.	Лабораторная работа № 1. Пользовательский интерфейс операционной среды: разработка конфигурационных файлов (MsDos, Windows, Linux).
Тема 4. Управление процессами Понятие процесса. Планирование и диспетчеризация процессов. Синхронизация процессов. Системные функции управления процессами. Программирование в операционной среде (управление многозадачностью)	Лабораторная работа № 2. Управление процессами (разработка конфигурационных файлов (в ОС семейства Linux) с использованием команд ps, top, nice, kill и др.; си- программ (на примере СИ для Linux): fork, execl, wait, nice, exit и др.).
Тема 5. Управление памятью Иерархия запоминающих устройств. Управление оперативной памятью. Виртуальная память. Свопинг. Виды переменных. Статическое и динамическое распределение памяти. Программирование в операционной среде (управление оперативной памятью)	Лабораторная работа № 3. Управление памятью (разработка конфигурационных файлов (в ОС семейства Linux) с использованием конструкций языка shell, разработка си- программ с использованием локальных, глобальных переменных, классов extern, static, auto, register; указателей, указателей на массивы, указателей на структуры; функций динамического распределения памяти: malloc, calloc, free, new, delete).
Тема 6. Управление файловой системой Файловая система на диске. Структура логического диска. Методы сохранения файлов на диске. Системные функции управления файлами. Программирование в операционной среде (управление файловой системой).	Лабораторная работа № 4. Управление файлами (разработка конфигурационных файлов (в ОС семейства Linux) с использованием конструкций языка shell, разработка си-программ с использованием функций работы с файлами: creat, open, read, write, access, lseek и др.)
Тема 7. Управление вводом/выводом Понятие ресурса. Виды ресурсов. Управление ресурсами. Устройства ПЭВМ. Виды устройств. Драйверы устройств. Системные функции управления устройствами ввода/вывода (УВВ). Программирование в операционной среде (управление УВВ)	Лабораторная работа № 5. Управление вводом/выводом.
Тема 8. Методы программирования Алгоритмическое, модульное, структурное программирование, объектно-ориентированный подход в проектировании СПО (ООП СПО), клиент-серверные технологии, проектирование микро, макроядра ОС. Языки программирования, системы программирования. Макроязыки.	Лабораторная работа № 6. Принципы проектирования операционных систем. Объектно-ориентированный подход в проектировании системных программ (СИ++).

Ассемблеры. Программирование в операционной среде (ООП СПО)	
Тема 11. Структура компиляторов и интерпретаторов Лексический, синтаксический и семантический анализаторы. Оптимизация кода. Генерация кода. Распределение памяти.	Лабораторная работа № 7. Компиляция, компоновка, загрузка программ. Оптимизация кода.

**Критерии конкретного оценочного средства (согласно ПОЛОЖЕНИЮ
о промежуточной аттестации обучающихся ВУиТ
по программам высшего образования – программам бакалавриата и программам
специалитета)**

По итогам тестирования оценка знаний обучающегося производится в соответствии со следующими критериями:

- правильных ответов 0-39% – «неудовлетворительно»/«не зачтено»;
- правильных ответов 40-59% – «удовлетворительно»/«зачтено»;
- правильных ответов 60-79% – «хорошо»/«зачтено»;
- правильных ответов 80-100% – «отлично»/«зачтено».

**Темы курсовых работ
Тематика курсовых работ**

1. Планирование процессов в многозадачных средах.
2. Синхронизация параллельно выполняющихся процессов в многозадачных средах.
3. Страничное распределение оперативной памяти.
4. Сегментное распределение оперативной памяти.
5. Странично-сегментное распределение оперативной памяти.
6. Методы сохранения файлов на диске.
7. Лексический анализатор компилятора.
8. Синтаксический анализатор компилятора.
9. Оптимизация кода программы в процессе компиляции.
10. Генерация кода программы в процессе компиляции.
11. Компоновка модулей проекта.
12. Настройка принтера в Linux-системе.
13. Настройка сети в Linux-системе.
14. Мониторинг процессов в Linux-системе.
15. Монтирование файловых систем.
16. Архивация в Linux-системе.
17. Разработка пользовательской библиотеки ввода/вывода.
18. Разработка приложения с использованием классов.
19. Разработка приложения с использованием средств динамического распределения оперативной памяти.
20. Разработка конфигурационных файлов MsDos.
21. Разработка конфигурационных файлов Linux.
22. Геометрия диска. Низкоуровневое форматирование.
23. Структура логического диска MsDos.
24. Структура логического диска Linux.
25. Установка программного обеспечения в Linux.

Тема курсовой работы, закрепленная за конкретным студентом, утверждается приказом ректора в начале семестра.

3.3 Оценочное средство 1 (лабораторное задания)

Раздел, тема дисциплины	Номер и тема лабораторной работы
Тема 2. Пользовательский интерфейс операционной среды Режим командной строки. Графический пользовательский интерфейс. Пример современной операционной системы.	Лабораторная работа № 1. Пользовательский интерфейс операционной среды: разработка конфигурационных файлов (MsDos, Windows, Unix).
Тема 4. Управление процессами Понятие процесса. Планирование и диспетчеризация процессов. Синхронизация процессов. Системные функции управления процессами. Программирование в операционной среде (управление многозадачностью)	Лабораторная работа № 2. Управление процессами (разработка конфигурационных файлов (в ОС семейства Unix) с использованием команд ps, top, nice, kill и др.; си- программ (на примере СИ для Unix): fork, execl, wait, nice, exit и др.).
Тема 5. Управление памятью Иерархия запоминающих устройств. Управление оперативной памятью. Виртуальная память. Свопинг. Виды переменных. Статическое и динамическое распределение памяти. Программирование в операционной среде (управление оперативной памятью)	Лабораторная работа № 3. Управление памятью (разработка конфигурационных файлов (в ОС семейства Unix) с использованием конструкций языка shell, разработка си- программ с использованием локальных, глобальных переменных, классов extern, static, auto, register; указателей, указателей на массивы, указателей на структуры; функций динамического распределения памяти: malloc, calloc, free, new, delete).
Тема 6. Управление файловой системой Файловая система на диске. Структура логического диска. Методы сохранения файлов на диске. Системные функции управления файлами. Программирование в операционной среде (управление файловой системой).	Лабораторная работа № 4. Управление файлами (разработка конфигурационных файлов (в ОС семейства Unix) с использованием конструкций языка shell, разработка си-программ с использованием функций работы с файлами: creat, open, read, write, access, lseek и др.)
Тема 7. Управление вводом/выводом Понятие ресурса. Виды ресурсов. Управление ресурсами. Устройства ПЭВМ. Виды устройств. Драйверы устройств. Системные функции управления устройствами ввода/вывода (УВВ). Программирование в операционной среде (управление УВВ)	Лабораторная работа № 5. Управление вводом/выводом.
Тема 8. Методы программирования Алгоритмическое, модульное, структурное программирование, объектно-ориентированный подход в проектировании СПО (ООП СПО), клиент-серверные технологии, проектирование микро, макроядра ОС. Языки программирования, системы программирования. Макроязыки. Ассемблеры. Программирование в операционной среде (ООП СПО)	Лабораторная работа № 6. Принципы проектирования операционных систем. Объектно-ориентированный подход в проектировании системных программ (СИ++).
Тема 11. Структура компиляторов и	Лабораторная работа № 7. Компиляция,

интерпретаторов Лексический, синтаксический и семантический анализаторы. Оптимизация кода. Генерация кода. Распределение памяти.	компоновка, загрузка программ. Оптимизация кода.
--	---

**Критерии конкретного оценочного средства (согласно ПОЛОЖЕНИЮ
о промежуточной аттестации обучающихся ВУиТ
по программам высшего образования – программам бакалавриата и программам
специалитета)**

По итогам тестирования оценка знаний обучающегося производится в соответствии со следующими критериями:

- правильных ответов 0-39% – «неудовлетворительно»/«не зачтено»;
- правильных ответов 40-59% – «удовлетворительно»/«зачтено»;
- правильных ответов 60-79% – «хорошо»/«зачтено»;
- правильных ответов 80-100% – «отлично»/«зачтено».

3. Промежуточный контроль

Перечень компетенций и индикаторов достижения компетенций, для оценки
сформированности которых используется данный ФОС

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
ПК-2. Администрирование сетевой подсистемы инфокоммуникационной системы организации	ПК-2.1. Анализирует принципы функционирования аппаратных, программных и программно-аппаратных средств сетевой подсистемы инфокоммуникационной системы организации ПК-2.2. Проектирует архитектуру аппаратных, программных и программно-аппаратных средств сетевой подсистемы инфокоммуникационной системы организации ПК-2.5. Использует различные методы управления аппаратными, программными и программно-аппаратными средствами сетевой подсистемы инфокоммуникационной системы организации ПК-2.6. Использует современные стандарты, нормативно-техническую документацию в области инфокоммуникационных технологий при администрировании устройств и программного обеспечения ПК-2.7. Участвует в проектировании, конфигурировании и планировании с требуемой производительностью и необходимой безопасностью сетевых подсистем инфокоммуникационной системы организации ПК-2.8. Участвует в настройке, администрировании, восстановлении при сбоях аппаратных, программных и программно-аппаратных средств сетевой подсистемы инфокоммуникационной системы организации ПК-2.9. Выполняет формирование документации по работе с сетевой подсистемой инфокоммуникационной системы организации

Номер задания	Содержание вопроса	Правильный ответ на задание
1.	Выберите правильный вариант ответа Какую архитектуру использует сетевая операционная система? А) файл-серверную В) клиент-серверную	В

Номер задания	Содержание вопроса	Правильный ответ на задание
	С) иерархическую D) стратегическую	
2.	Выберите правильный вариант ответа _____ – способность программы системного программного обеспечения корректно взаимодействовать с программами выполненными на других языках, с другими модулями и т.д. А) Принцип функциональной избирательности В) Принцип функциональной избыточности С) Принцип виртуализации <u>D) Принцип совместимости</u>	D
3.	Выберите правильный вариант ответа. _____ - определение смыслового значения текста программы и проверка особых свойств (например, типов переменных и областей их видимости). работы анализа компилятора. А) Лексический анализ В) Синтаксический анализ (разбор) <u>С) Семантический анализ</u> D) Технический анализ	C
4.	Выберите правильный вариант ответа. Процесс создания эффективного целевого кода системного программного обеспечения. А) компиляция <u>В) оптимизация</u> С) агрегация D) синтезация	B
5.	Выберите правильный вариант ответа. В современных средах программирования для поставки и интеграции библиотек используются _____, предназначенные для управления библиотеками подпрограмм и установления зависимостей между ними. Таким образом, осуществляется внедрение необходимой версии библиотеки и ее непрерывная актуализация, при обновлении библиотеки поставщиком. <u>А) диспетчеры пакетов</u> В) диспетчер программ С) диспетчер систем D) диспетчеры каналов	A
6.	Область памяти, размер которой известен на этапе компиляции А) область памяти кэш В) динамическая область памяти	C

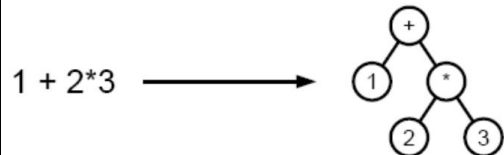
Номер задания	Содержание вопроса	Правильный ответ на задание
	<u>С) статическая область памяти</u> D) стратегическая память	
7.	Задачей _____ является согласование адресов во всех фрагментах кода: присвоение каждой программной секции адреса и также добавление к программе коды библиотечных функций. A) тестировщика <u>В) компоновщика</u> C) компилятора D) анализатора	В
8.	Перечислите критерии эффективности оптимизации системного программного обеспечения (СПО).	Критерии эффективности оптимизации системного программного обеспечения (СПО): 1) время выполнения программы; 2) объем используемой памяти в процессе выполнения программы; 3) объем (размер) программы; 4) равномерность загрузки оборудования.
9.	Перечислите действия машинно-зависимой оптимизации СПО.	Машинно-зависимая оптимизация СПО заключается, как правило, в выполнении следующих действий: 1) замена медленных команд на быстрые; 2) замена общих команд на команды, специфичные для текущего процессора; 3) распределение регистров; 4) учет технических особенностей компьютера при распределении нагрузки.
10.	Перечислите варианты организации межпрограммных связей.	Организация межпрограммных связей СПО возможна способами: 1) на уровне библиотек (для программ, функционирующих на одном устройстве); 2) на уровне сервисов, осуществляющих обмен данными по различным протоколам (для клиент-серверных приложений, для интеграции серверов); 3) на уровне данных (совместный доступ к данным); 4) через различные технологии интеграции (ESB, CORBA, COM, DCOM, ActiveX и др.).
11.	Практически все компиляторы выполняют оптимизацию на этапе генерации кода. Приведите примеры оптимизирующих преобразований компилятора на этапе	Различают 2 основных вида оптимизирующих преобразований компилятора на этапе генерации кода: 1) преобразования, не зависящие от

Номер задания	Содержание вопроса	Правильный ответ на задание
	генерации кода.	объектного языка, не зависящие от архитектуры, на которой будет выполняться программа; 2) преобразования, зависящие от свойств объектного языка, от архитектуры вычислительной машины.
12.	Приведите примеры динамической области памяти (ДП).	Динамические области памяти можно разделить на: 1) ДП, выделяемая пользователем; 2) ДП, выделяемая компилятором.
13.	Опишите кратко преобразования процесса компоновки программ	Рассмотрим процесс компоновки: 1) программа строится из конструкций-операторов; 2) операторы включают выражения; 3) выражения состоят из операндов и операций; 4) операндам соответствуют отдельные участки оперативной памяти для сохранения результатов вычисления; 5) в ходе трансляции устанавливается соответствие между операндами и адресами оперативной памяти.
14.	Перечислите этапы работы лексического анализатора	Порядок конструирования анализатора: 1) выделить во входном языке $L(G)$ множество классов лексем; 2) построить для каждого класса лексем грамматику; 3) построить модель распознавателя; 4) выбрать формат и код лексем дескриптора вида – тип, адрес; построить дескрипторный текст входной программы.
15.	Перечислить варианты взаимосвязи лексического и синтаксического анализатора в компиляторах.	Возможны 2 различных метода организации взаимосвязи лексического анализатора и синтаксического анализатора: 1) последовательный; 2) параллельный.
16.	Основная задача синтаксического анализатора?	Основная задача синтаксического анализатора - разбор структуры программы.
17.	На рисунке представлено абстрактное синтаксическое дерево. Какие компоненты являются терминалами.	Листья абстрактного синтаксического дерева называются также терминалами.

Номер задания	Содержание вопроса	Правильный ответ на задание
18.	На рисунке представлено абстрактное синтаксическое дерево. Какие компоненты являются нетерминалами.	Корень и внутренние узлы абстрактного синтаксического дерева называются также нетерминалами.
19.	Приведите примеры функций программ, относящихся к Сервисному программному обеспечению.	Сервисное программное обеспечение включает программы и программные комплексы, которые расширяют возможности базового программного обеспечения и организуют более удобную среду работы пользователя (программы диагностики работоспособности компьютера, антивирусные программы, программы обслуживания дисков, программы обслуживания сети и т.д.).
20.	Достоинства пользовательского интерфейса операционной системы в виде командной строки.	Пользовательский интерфейс в виде командной строки требует меньше системных ресурсов для работы, исключает вмешательство неопытных пользователей, часто предоставляет больше возможностей чем графический интерфейс.

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
ПК-1. Управление программно- аппаратными средствами информационных служб инфокоммуникационной системы организации	ПК-1.1. Планирует архитектуру и функционирование аппаратных, программных и программно-аппаратных средств информационных служб инфокоммуникационной системы организации ПК-1.2. Использует правила и методы технического обслуживания и восстановления аппаратных, программных и программно-аппаратных средств информационных служб инфокоммуникационной системы организации ПК-1.3. Участвует в проектировании программно- аппаратных средств информационных служб инфокоммуникационной системы организации

	ПК-1.4. Участвует в конфигурировании, управлении, восстановления работоспособности программно- аппаратных средств информационных служб инфокоммуникационной системы организации ПК-1.5. Использует нормативно-техническую документацию в области инфокоммуникационных технологий
--	--

Номер задания	Содержание вопроса	Правильный ответ на задание
1.	По описанию особенностей выберите тип взаимосвязи лексического и синтаксического анализатора. Лексический анализатор просматривает весь текст исходной программы от начала до конца, преобразовывает его в структурированный раздел данных (таблицу лексем). В таблице лексем, лексемы заменяются на специальные коды – дескрипторы лексем. <u>A) последовательный вариант</u> B) параллельный вариант C) линейный вариант D) вариант выбора	A
2.	По описанию особенностей выберите тип взаимосвязи лексического и синтаксического анализатора. Лексический анализ осуществляется поэтапно. Синтаксический анализатор, выполнив очередной разбор, обращается за следующей лексемой. A) последовательный вариант <u>B) параллельный вариант</u> C) линейный вариант D) вариант выбора	B
3.	Выберите правильный вариант ответа. На рисунке схема _____.  <u>A) лексического анализатора</u> B) синтаксического анализатора C) семантического анализатора D) стратегического анализатора	A
4.	Выберите правильный вариант ответа. На рисунке представлен результат работы _____ анализатора _____ 	B

Номер задания	Содержание вопроса	Правильный ответ на задание
	А) лексического <u>В) синтаксического</u> С) семантического D) стратегического	
5.	Выберите правильный вариант ответа. Он использует синтаксическое дерево и таблицу символов, чтобы проверить, соответствует ли данная программа семантически определению языка. Он собирает информацию о типе и сохраняет ее либо в синтаксическом дереве, либо в таблице символов. Эта информация о типе впоследствии используется компилятором во время генерации промежуточного кода. А) лексический анализатор В) синтаксический анализатор <u>С) семантический анализатор</u> D) стратегического анализатора	С
6.	Выберите правильный вариант ответа. В системе Linux для работы компилятора Gcc прописан код \$ gcc -g myfile.c -o exefile <u>А) Скомпилирован myfile.c с отладочной информацией и ссылкой на выходной файл exefile</u> В) Скомпилирован myfile.c с включенными предупреждающими сообщениями и сделана ссылка на выходной файл exefile С) Скомпилирован myfile.c с оптимизацией и ссылкой на выходной файл exefile D) Скомпилирован myfile.c с оптимизацией	А
7.	Выберите правильный вариант ответа. В системе Linux для работы компилятора Gcc прописан код \$./exefile <u>А) Запустить выходной файл exefile</u> В) Связать выходной файл exefile С) Собрать выходной файл exefile D) Просмотреть выходной файл exefile	А
8.	Выберите правильный вариант ответа. Задача. Необходимо организовать строку для каждого параметра-файла (или для стандартного ввода) которая содержит, в зависимости от ключа, число строк в файле (ключ -l), число слов (ключ -w) или число символов (ключ -c). По умолчанию (без ключей) выдаются все три числа. Какой командой Shell можно воспользоваться в	D

Номер задания	Содержание вопроса	Правильный ответ на задание
	системе Linux? A) tail [ключи] [файл] B) grep [ключи] образец [список_файлов] C) head [ключи] [файл] <u>D) wc [ключи] [файлы]</u> E) less [файл]	
9.	Дайте пояснение понятию системное программирование.	Системное программирование — создание системного программного обеспечения, выполняет данный тип программирования системный программист (разработчик системного программного обеспечения).
10.	Как называется тип программного обеспечения которое обеспечивает аппаратную совместимость конкретной аппаратной платформы с программным обеспечением прикладных программ?	Между аппаратной частью конкретной аппаратной платформы и программным обеспечением прикладных программ обеспечивает системное программное обеспечение.
11.	В соответствии с типами грамматик, языки делятся на типы. Дайте описание контекстно-зависимых языков.	Контекстно-зависимые языки. Языки и грамматики применяются в анализе и переводе текстов на естественных языках. На основе таких грамматик может выполняться автоматизированный перевод с одного естественного языка на другой.
12.	В соответствии с типами грамматик, языки делятся на типы. Дайте описание регулярных языков.	Регулярные языки являются самыми распространенными и широко используемыми в области проектирования вычислительных систем. Для работы с ними используют регулярные множества, регулярные выражения и конечные автоматы.
13.	Дайте краткое описание понятию супервизор.	Супервизор – ядро операционной системы как части системного программного обеспечения.
14.	Перечислите этапы работы синтеза компилятора	Этапы работы синтеза компилятора: 1 Генерация машинно-независимого кода 2 Оптимизация машинно-независимого кода 3 Распределение памяти 4 Генерация машинного кода 5 Оптимизация машинного кода
15.	Предложен поток токенов лексического анализатора, представьте исходный код строки программы ИМЯ "net_worth_future" ПРИСВАИВАНИЕ ОТКРЫВАЮЩАЯ СКОБКА	net_worth_future = (assets — liabilities);

Номер задания	Содержание вопроса	Правильный ответ на задание
	ИМЯ "assets" МИНУС ИМЯ "liabilities" ЗАКРЫВАЮЩАЯ СКОБКА ТОЧКА С ЗАПЯТОЙ	
16.	Перечислите функции семантического анализа	Функции семантического анализа 1. Проверка типов 2. Проверка меток 3. Проверка управления потоком
17.	Приведите примеры языков программирования для написания системных программ в Linux	Писать программы для Linux можно почти на любом языке начиная от Java и Python и заканчивая C# и даже Pascal.
18.	Действия данной команды, написанной для Linux \$ sudo apt install gcc	Установка компилятора Gcc.
19.	Задача. Представлен код. Определите действия компилятора в системе Linux \$ gcc file1.c file2.c -o execfile	Компилятор Gcc скомпилирует file1.c и file2.c и создаст ссылку на выходной файл execfile.
20.	Представлен код Shell-скрипта в системе Linux. Перечислите используемые переменные. #!/bin/bash newdir=\$1 indent=\$2 [-n \$newdir] && cd \$newdir for i in * ; do echo \$indent\$i; if [-d \$i] && [-x \$newdir] ; then ls -l \$i "\$indent.." fi done	Две переменные newdir и indent.
21.	Представлен код Shell-скрипта в системе Linux. Какой командой требуется информация выбирается из системного файла /etc/passwd. #!/bin/bash info= while getopts nug option ; do case \$option in n) info=\$info `grep '^\$2 /etc/passwd   cut -f 5 -d :`" u) info=\$info `grep '^\$2 /etc/passwd   cut -f 3 -d :`" g) info=\$info `grep '^\$2 /etc/passwd   cut -f 4 -d :`" *) echo Bad option: \$option esac done echo \$info	С помощью команды grep.
22.	Представлен код Shell-скрипта в системе	Символ ^ перед именем заставляет

Номер задания	Содержание вопроса	Правильный ответ на задание
	Linux. Что обозначает символ ^ ? #!/bin/bash info= while getopts nug option ; do case \$option in n) info=\$info" `grep '^\$2 /etc/passwd   cut -f 5 -d :`" u) info=\$info" `grep '^\$2 /etc/passwd   cut -f 3 -d :`" g) info=\$info" `grep '^\$2 /etc/passwd   cut -f 4 -d :`" *) echo Bad option: \$option esac done echo \$info	искать имя только в начале строки.
23.	Представлен код Shell-скрипта в системе Linux. Какими ключевыми словами начинается и заканчивается цикл? #!/bin/bash info= while getopts nug option ; do case \$option in n) info=\$info" `grep '^\$2 /etc/passwd   cut -f 5 -d :`" u) info=\$info" `grep '^\$2 /etc/passwd   cut -f 3 -d :`" g) info=\$info" `grep '^\$2 /etc/passwd   cut -f 4 -d :`" *) echo Bad option: \$option esac done echo \$info	Тело цикла всегда начинается с do и закрывается done , а между этими двумя ключевыми словами перечисляются команды, которые необходимо выполнить.
24.	Представлен код в системе Linux. Кратко опишите настройки интерфейса eth0? auto lo iface lo inet loopback auto eth0 iface eth0 inet static address 100.40.15.78 netmask 255.255.255.0 gateway 10.10.55.10	Интерфейсу eth0 задан основной адрес 100.40.15.78, маска 255.255.255.0, адрес dns-сервера 10.10.55.10.
25.	Представлен код в системе Linux. Кратко опишите настройки первых двух строчек кода? auto lo iface lo inet loopback auto eth0 iface eth0 inet dhcp	Первые две строчки показывают настройку локального сетевого интерфейса, используемого для тестирования передачи
26.	Представлен код Shell в системе Linux. Кратко опишите работу программного	Проверка доступности нескольких интернет ресурсов по ICMP (ping). В

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre> кода? #!/bin/sh TEST1=0 TEST1=`ping -c 4 -q ya.ru   grep -oP 'd+(?=% packet loss)'\` echo \$TEST1 if [\$TEST1 -ne 0] then TEST2=0 TEST2=`ping -c 4 -q 8.8.8.8   grep -oP 'd+(?=% packet loss)'\` if [\$TEST2 -ne 0] then echo "Alarm!!!" else echo "false alarm" fi else echo "Ok" fi </pre>	данном примере проверяется доступность сайта ya.ru и если он не доступен, то пингуется 8.8.8.8 (google public dns).
27.	Задача. По представленной ниже части команды и реакции системы определите доступность сайта google.com <pre># ping -c 1 -q google.com >&/dev/null; echo \$?</pre> <pre>0</pre>	В выводе «0» означает отсутствие подключения к Интернету.
28.	Задача. По представленной ниже части команды и реакции системы определите доступность подключения к Интернет. <pre># nslookup google.com Server: 192.168.0.10 Address: 192.168.0.10#53 Non-authoritative answer: Name: google.com Address: 172.217.166.110 Name: google.com Address: 2404:6800:4007:811::200e</pre>	Проведен успешный поиск DNS.
29.	Задача. По представленной ниже части команды и реакции системы определите доступность подключения к Интернет. <pre># curl -Is http://www.google.com head -n 1</pre> <pre>HTTP/1.1 200 OK</pre>	Если получено что-либо, кроме 200 ОК это означает, что серверу не удалось подключиться к предоставленной странице. Возможно указана недействительная веб-страницу. , Уверенности подключения к Интернету нет.
30.	Задача. По представленной ниже части команды и реакции системы определите доступность подключения к Интернет. <pre># nmap -p 443 google.com Starting Nmap 7.70 (https://nmap.org) at 2019-11-19 07:40 IST Nmap scan report for google.com (172.217.166.110) Host is up (0.054s latency). Other addresses for google.com (not scanned): 2404:6800:4007:811::200e DNS record for 172.217.166.110: maa05s09-in-f14.1e100.net PORT STATE SERVICE 443/tcp open https Nmap done: 1 IP address (1 host up) scanned in 0.77 seconds</pre>	Сканирование google.com через порт 443. nmap удалось установить соединение с google.com.

