

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Агафонов Александр Викторович

Должность: директор филиала

Дата подписания: 18.06.2026 09:51:35

Университет: Московский политех

2539477a8ecf706dc9cff164bc411eb6d3c4ab06

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**  
**ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**  
**ЧЕБОКСАРСКИЙ ИНСТИТУТ (ФИЛИАЛ) МОСКОВСКОГО ПОЛИТЕХНИЧЕСКОГО УНИВЕРСИТЕТА**

## **Кафедра Информационных технологий и систем управления**



# **РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

## **«Системное программное обеспечение»**

(наименование дисциплины)

Направление подготовки	<b>09.03.01 «Информатика и вычислительная техника»</b> (код и наименование направления подготовки)
Направленность (профиль) подготовки	<b>«Программное обеспечение вычислительной техники и автоматизированных систем»</b> (наименование профиля подготовки)
Квалификация выпускника	<b>бакалавр</b>
Форма обучения	<b>очная, заочная</b>
Год начала обучения	<b>2026</b>

Чебоксары, 2026

Рабочая программа дисциплины разработана в соответствии с:

- Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.01 – Информатика и вычислительная техника, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 929 от 19 сентября 2017 г. зарегистрированный в Минюсте 10 октября 2017 года, рег. номер 48489 (далее – ФГОС ВО);
- учебным планом (очной, заочной форм обучения) по направлению подготовки 09.03.01 «Информатика и вычислительная техника».

Рабочая программа дисциплины включает в себя оценочные материалы для проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине (п.6 Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины).

Автор Пикина Наталия Евгеньевна, кандидат педагогических наук, доцент кафедры информационных технологий и систем управления

(указать ФИО, ученую степень, ученое звание или должность)

Программа одобрена на заседании кафедры Информационных технологий и систем управления (протокол № 9 от 22.05.2026 г.).

**1. Перечень планируемых результатов обучения, соотнесенных с планируемыми результатами освоения образовательной программы (Цели освоения дисциплины)**

1.1. *Целями* освоения дисциплины «Системное программное обеспечение» является:

изучение особенностей функционирования системного уровня компьютерной системы, принципов и методов его работы и организации.

Для достижения целей дисциплины необходимо решить следующую *основную задачу* – сформировать у обучающихся теоретические знания и практические навыки, необходимые для:

- ~ освоение студентами системного программного обеспечения;
- ~ приобретение навыков самостоятельного изучения отдельных тем дисциплины и решения типовых задач;
- ~ приобретение навыков проектирования программных и аппаратных средств;
- ~ применение современных инструментальных средств при разработке программного обеспечения;
- ~ изучение организации и принципов построения современных операционных систем и системных программ.

1.2. Области профессиональной деятельности и(или) сферы профессиональной деятельности, в которых выпускники, освоившие программу, могут осуществлять профессиональную деятельность:

*Об Связь, информационные и коммуникационные технологии (в сфере проектирования, разработки, внедрения и эксплуатации средств вычислительной техники и информационных систем, управления их жизненным циклом).*

1.3. К основным задачам изучения дисциплины относится подготовка обучающихся к выполнению трудовых функций в соответствии с профессиональными стандартами:

Код и наименование профессионального стандарта	Обобщенные трудовые функции			Трудовые функции		
	код	наименование	уровень квалификации и	наименование	код	уровень (подуровень) квалификации
06.001 Программист Профессиональный стандарт «Программист», утвержденный приказом Министерством труда и социальной защиты Российской Федерации от 18.11.2013 № 679н (зарегистрирован в	D	Разработка требований и проектирование программного обеспечения	6	Анализ требований к программному обеспечению	D/01.6	6
			6	Разработка технических спецификаций на программные компоненты и их взаимодействие	D/02.6	

Код и наименование профессионального стандарта	Обобщенные трудовые функции			Трудовые функции		
	код	наименование	уровень квалификации	наименование	код	уровень (подуровень) квалификации
Министерстве труда и социальной защиты Российской Федерации от 18 декабря 2013 г. №30635)			6	Проектирование программного обеспечения	D/03.6	
06.028 Системный программист Профессиональный стандарт «Системный программист», утв. Приказом Министерством труда и социальной защиты РФ от 29 сентября 2020 года N 678н	А	Разработка компонентов системных программных продуктов	6	Разработка драйверов устройств	A/01.6	6
			6	Разработка компиляторов, загрузчиков, сборщиков	A/02.6	6
				Разработка системных утилит	A/03.6	6
				Создание инструментальных средств программирования	A/04.6	6
06.015 Специалист по информационным системам Профессиональный стандарт «Специалист по информационным системам», утв. приказом Министерством труда и социальной защиты РФ 18 ноября 2014 г. №896н	С	Выполнение работ и управление работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы	6	Разработка модели бизнес-процессов заказчика	C/08.6	6
				Выявление требований к ИС	C/11.6	6
				Анализ требований	C/12.6	6
				Разработка архитектуры ИС	C/14.6	6
				Проектирование и дизайн ИС	C/16.6	6
				Разработка баз данных ИС	C/17.6	6
				Организационное и технологическое обеспечение кодирования на языках программирования	C/18.6	6
				Организационное и технологическое обеспечение модульного тестирования ИС (верификации)	C/19.6	6

Код и наименование профессионального стандарта	Обобщенные трудовые функции			Трудовые функции		
	код	наименование	уровень квалификации	наименование	код	уровень (подуровень) квалификации
				Создание пользовательской документации к ИС	C/22.6	6
				Организация репозитория хранения данных о создании (модификации) и вводе ИС в эксплуатацию	C/40.6	6

#### 1.4. Компетенции обучающегося, формируемые в результате освоения дисциплины

Наименование категории (группы) компетенций	Код и наименование компетенций	Код и наименование индикатора достижения компетенции	Перечень планируемых результатов обучения
Разработка, отладка, проверка работоспособности, модификация программного обеспечения	ПК-2 Способен разрабатывать требования и проектировать программное обеспечение	ПК-2.1 Выполняет анализ требований к программному обеспечению	<p><i>на уровне знаний:</i> знать принципы и алгоритмы работы системного программного обеспечения;</p> <p><i>на уровне умений:</i> уметь применять системное программное обеспечение для формирования компьютерных систем;</p> <p><i>на уровне навыков:</i> владеть системными утилитами и инструментами для настройки и обслуживания компьютерной системы</p>
		ПК - 2 . 2 Разрабатывает технические спецификации на программные компоненты и их взаимодействие	<p><i>на уровне знаний:</i> знать принципы работы с аппаратным обеспечением на системном уровне; штатные режимы работы программно-аппаратных комплексов;</p> <p><i>на уровне умений:</i> уметь применять возможности системного программного обеспечения для настройки и наладки аппаратного обеспечения;</p> <p><i>на уровне навыков:</i> техническими спецификациями на программные компоненты и</p>

			их взаимодействие
		П К - 2 . 3 Проектирует программное обеспечение	<i>на уровне знаний:</i> знать назначение системного программного обеспечения в составе компьютерных систем; <i>на уровне умений:</i> уметь определять нарушения и аномалии в работе программно-аппаратных комплексов. <i>на уровне навыков:</i> инструментами мониторинга работы программно-аппаратных комплексов.

## 2. Место дисциплины в структуре ОПОП

Дисциплина Б1.Д(М).В.5 «Системное программное обеспечение» реализуется в рамках части, формируемой участниками образовательных отношений Блока 1 «Элективные дисциплины (модули)» программы бакалавриата.

Дисциплина «Системное программное обеспечение» преподаётся обучающимся по очной форме обучения – в 6-м семестре, по заочной форме – в 10-м семестре.

Дисциплина «Системное программное обеспечение» является промежуточным этапом формирования компетенций ПК-2 в процессе освоения ОПОП.

Дисциплина «Системное программное обеспечение» основывается на знаниях, умениях и навыках, приобретенных при изучении дисциплин Учебная практика: ознакомительная практика, Информационные сети и коммуникации, Проектная деятельность, Учебная практика: научно-исследовательская работа (получение первичных навыков научно-исследовательской работы), Учебная практика: технологическая (проектно-технологическая) практика, Базы данных, Интеллектуальные системы, Основы систем искусственного интеллекта, Параллельное программирование, Программирование на языке низкого уровня и является предшествующей для изучения дисциплин Теория вычислительных процессов и языков программирования, Эргономика, Технический дизайн, Производственная практика: технологическая (проектно-технологическая) практика, Производственная практика: научно-исследовательская работа, Цифровая обработка сигналов, Архитектура вычислительных систем, Производственная практика: преддипломная практика, Государственная итоговая аттестация: подготовка к сдаче и сдача государственного экзамена, Государственная итоговая аттестация: выполнение и защита выпускной квалификационной работы.

Формой промежуточной аттестации знаний обучаемых по очной форме обучения является экзамен в 6-м семестре, по заочной форме экзамен в 10-м семестре.

### 3. Объем дисциплины

очная форма обучения:

Вид учебной работы по дисциплине	Всего в з.е. и часах	Семестр 6 в часах
<b>Общая трудоёмкость дисциплины</b>	<b>5 з.е. -180 ак.час</b>	<b>180 ак.час</b>
<b>Контактная работа - Аудиторные занятия</b>	<b>55</b>	<b>55</b>
<i>Лекции</i>	18	18
<i>Лабораторные занятия</i>	36	36
<i>Семинары, практические занятия</i>	-	-
<i>Консультация</i>	1	1
<b>Самостоятельная работа</b>	<b>89</b>	<b>89</b>
<b>Курсовая работа (курсовой проект)</b>	-	-
<b>Вид промежуточной аттестации</b>	Экзамен – 36 часов	Экзамен – 36 часов

заочная форма обучения:

Вид учебной работы по дисциплине	Всего в з.е. и часах	Семестр 10 в часах
<b>Общая трудоёмкость дисциплины</b>	<b>5 з.е. -180 ак.час</b>	<b>180 ак.час</b>
<b>Контактная работа - Аудиторные занятия</b>	<b>21</b>	<b>21</b>
<i>Лекции</i>	10	10
<i>Лабораторные занятия</i>	10	10
<i>Семинары, практические занятия</i>	-	-
<i>Консультация</i>	1	1
<b>Самостоятельная работа</b>	<b>150</b>	<b>150</b>
<b>Курсовая работа (курсовой проект)</b>	-	-
<b>Вид промежуточной аттестации</b>	Экзамен – 9 часов	Экзамен – 9 часов

**4. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий**

#### 4.1. Учебно-тематический план

Очная форма обучения

Тема (раздел)	Распределение часов			Самостоя- тельная работа	Формируемые компетенции (код)
	Лекции	Лабораторные занятия	Практические занятия		
Тема 1. Введение в системное программное обеспечение	2	4	-	10	ПК-2.1., ПК-2.2., ПК-2.3.
Тема 2. Электронная вычислительная машина	2	4	-	10	ПК-2.1., ПК-2.2., ПК-2.3.
Тема 3. Трансляция	2	4	-	10	ПК-2.1., ПК-2.2., ПК-2.3.
Тема 4. Ассемблеры	2	4	-	11	ПК-2.1., ПК-2.2., ПК-2.3.

Тема 5. Загрузчики	2	4	-	12	ПК-2.1., ПК-2.2., ПК-2.3.
Тема 6. RunTime Interface	2	4	-	12	ПК-2.1., ПК-2.2., ПК-2.3.
Тема 7. Средства разработки	2	4	-	12	ПК-2.1., ПК-2.2., ПК-2.3.
Тема 8. Программирование ОС	4	8	-	12	ПК-2.1., ПК-2.2., ПК-2.3.
Консультация	1			-	ПК-2.1., ПК-2.2., ПК-2.3.
Курсовая работа (курсовой проект)	-				ПК-2.1., ПК-2.2., ПК-2.3.
Контроль (экзамен)	36				ПК-2.1., ПК-2.2., ПК-2.3.
<b>ИТОГО</b>	<b>55</b>			<b>89</b>	

### Заочная форма обучения

Тема (раздел)	Распределение часов			Самостоятельная работа	Формируемые компетенции (код)
	Лекции	Лабораторные занятия	Практические занятия		
Тема 1. Введение в системное программное обеспечение	2	-	-	18	ПК-2.1., ПК-2.2., ПК-2.3.
Тема 2. Электронная вычислительная машина	2	2	-	18	ПК-2.1., ПК-2.2., ПК-2.3.
Тема 3. Трансляция	2	-	-	18	ПК-2.1., ПК-2.2., ПК-2.3.
Тема 4. Ассемблеры	-	2	-	18	ПК-2.1., ПК-2.2., ПК-2.3.
Тема 5. Загрузчики	-	2	-	18	ПК-2.1., ПК-2.2., ПК-2.3.
Тема 6. RunTime Interface	-	2	-	20	ПК-2.1., ПК-2.2., ПК-2.3.
Тема 7. Средства разработки	2	-	-	20	ПК-2.1., ПК-2.2., ПК-2.3.
Тема 8. Программирование ОС	2	2	-	20	ПК-2.1., ПК-2.2., ПК-2.3.
Консультация	1			-	ПК-2.1., ПК-2.2., ПК-2.3.

Курсовая работа (курсовой проект)	-		ПК-2.1., ПК-2.2., ПК-2.3.
Контроль (экзамен)	9		ПК-2.1., ПК-2.2., ПК-2.3.
<b>ИТОГО</b>	<b>21</b>	<b>150</b>	

## **4.2. Содержание дисциплины**

### **Тема 1. Введение в системное программное обеспечение**

Определение и структура системного ПО.

Основные функции и классификация системных программ.

Различия между прикладным и системным ПО.

Роль системного ПО в архитектуре вычислительной системы.

Примеры системных программ: компиляторы, драйверы, ОС.

Этапы загрузки и запуска программ в системной среде.

Эволюция системного ПО.

### **Тема 2. Электронная вычислительная машина**

Архитектура ЭВМ: процессор, память, устройства ввода/вывода.

Модель фон Неймана.

Принципы работы центрального процессора.

Организация памяти и адресация.

Машинные команды и форматы представления данных.

Представление чисел и логика вычислений.

### **Тема 3. Трансляция**

Концепция трансляции программ.

Этапы трансляции: лексический, синтаксический, семантический анализ.

Различия между компиляцией и интерпретацией.

Генерация объектного кода.

Оптимизация на уровне трансляции.

Ошибки времени компиляции.

### **Тема 4. Ассемблеры**

Назначение и структура ассемблеров.

Язык ассемблера и машинный код.

Символьные имена, директивы, макросы.

Процесс ассемблирования и структура объектного файла.

Однопроходные и многопроходные ассемблеры.

Роль ассемблера в разработке низкоуровневого ПО.

### **Тема 5. Загрузчики**

Назначение загрузчиков и их роль в ОС.

Этапы загрузки операционной системы.

Типы загрузчиков: начальный (bootloader), второй этап (stage 2).

Адресация и размещение программ в памяти.

Примеры загрузчиков: GRUB, LILO.

Связь загрузчика с BIOS/UEFI.

## **Тема 6. RunTime Interface**

Понятие среды выполнения (RunTime).

Интерфейсы между ОС и исполняемыми программами.

Управление памятью во время выполнения.

Обработка исключений и сигналы.

Динамическая компоновка библиотек.

Мониторинг и отладка во время выполнения.

## **Тема 7. Средства разработки**

Инструменты системного программирования.

Компиляторы, отладчики, профилировщики.

IDE для системных языков: Visual Studio, Eclipse CDT, CLion.

Использование Makefile, CMake и систем сборки.

Контроль версий: Git в системных проектах.

Практика автоматизации сборки и тестирования.

## **Тема 8. Программирование ОС**

Интерфейсы прикладного программирования ОС.

Работа с системными вызовами и API.

Управление процессами, потоками, памятью и файлами.

Межпроцессное взаимодействие.

Разработка драйверов и модулей ядра.

Примеры: POSIX, WinAPI, системное программирование на C.

## **5. Учебно-методическое обеспечение самостоятельной работы студентов**

Самостоятельная работа проводится с целью: систематизации и закрепления полученных теоретических знаний и практических умений обучающихся; углубления и расширения теоретических знаний студентов; формирования умений использовать нормативную, правовую, справочную документацию, учебную и специальную литературу; развития познавательных способностей и активности обучающихся: творческой инициативы, самостоятельности, ответственности, организованности; формирование самостоятельности мышления, способностей к саморазвитию, совершенствованию и самоорганизации; формирования профессиональных компетенций; развитию исследовательских умений студентов.

Формы и виды самостоятельной работы студентов: чтение основной и дополнительной литературы – самостоятельное изучение материала по рекомендуемым литературным источникам; работа с библиотечным каталогом, самостоятельный подбор необходимой литературы; работа со словарем, справочником; поиск необходимой информации в сети Интернет; конспектирование источников; реферирование источников; составление аннотаций к прочитанным литературным источникам; составление рецензий и отзывов на прочитанный материал; составление обзора публикаций по теме; составление и разработка терминологического словаря; составление хронологической таблицы; составление библиографии (библиографической картотеки); подготовка к различным формам текущей и промежуточной

аттестации (к тестированию, контрольной работе, экзамену); выполнение домашних контрольных работ; самостоятельное выполнение практических заданий репродуктивного типа (ответы на вопросы, задачи, тесты; выполнение творческих заданий).

Технология организации самостоятельной работы обучающихся включает использование информационных и материально-технических ресурсов образовательного учреждения: библиотеку с читальным залом, компьютерные классы с возможностью работы в Интернет; аудитории (классы) для консультационной деятельности.

Перед выполнением обучающимися внеаудиторной самостоятельной работы преподаватель проводит консультирование по выполнению задания, который включает цель задания, его содержания, сроки выполнения, ориентировочный объем работы, основные требования к результатам работы, критерии оценки. Во время выполнения обучающимися внеаудиторной самостоятельной работы и при необходимости преподаватель может проводить индивидуальные и групповые консультации.

Контроль самостоятельной работы студентов предусматривает: соотнесение содержания контроля с целями обучения; объективность контроля; валидность контроля (соответствие предъявляемых заданий тому, что предполагается проверить); дифференциацию контрольно-измерительных материалов.

Формы контроля самостоятельной работы: просмотр и проверка выполнения самостоятельной работы преподавателем; организация самопроверки, взаимопроверки выполненного задания в группе; обсуждение результатов выполненной работы на занятии; проведение письменного опроса; проведение устного опроса; организация и проведение индивидуального собеседования; организация и проведение собеседования с группой.

### **Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы**

<b>Наименование тем (разделов) дисциплины</b>	<b>Перечень вопросов, отводимых на самостоятельное освоение</b>	<b>Формы внеаудиторной самостоятельной работы</b>
Тема 1. Введение в системное программное обеспечение	<ol style="list-style-type: none"> <li>1. Определить понятие системного программного обеспечения.</li> <li>2. Перечислить виды системных программ.</li> <li>3. Описать функции системного ПО.</li> <li>4. Указать различия между прикладным и системным ПО.</li> <li>5. Изучить этапы взаимодействия ПО с ОС.</li> <li>6. Определить роль ядра в системном ПО.</li> <li>7. Привести примеры современных системных программ.</li> <li>8. Охарактеризовать архитектуру системного ПО.</li> <li>9. Объяснить роль системного ПО в запуске пользовательских программ.</li> <li>10. Рассмотреть жизненный цикл системного ПО.</li> </ol>	Чтение литературы, подготовка краткого обзора программ, выполнение классификационной таблицы, составление схем, участие в дискуссии, просмотр видеоматериалов.

<p>Тема 2. Электронная вычислительная машина</p>	<ol style="list-style-type: none"> <li>1. Составить структурную схему ЭВМ.</li> <li>2. Описать функции основных компонентов ЭВМ.</li> <li>3. Изучить принципы работы памяти.</li> <li>4. Сравнить архитектуру фон Неймана и Гарвардскую.</li> <li>5. Рассмотреть внутренние и внешние интерфейсы.</li> <li>6. Изучить машинные команды и форматы представления данных.</li> <li>7. Пояснить принципы адресации.</li> <li>8. Охарактеризовать систему прерываний.</li> <li>9. Привести примеры современных ЭВМ.</li> <li>10. Определить роль ЭВМ в системном ПО.</li> </ol>	<p>Создание схем, разбор архитектур, решение задач на адресацию, составление терминологического словаря, моделирование ЭВМ в симуляторах.</p>
<p>Тема 3. Трансляция</p>	<ol style="list-style-type: none"> <li>1. Определить понятие трансляции и ее роль в программировании.</li> <li>2. Описать этапы трансляции программ.</li> <li>3. Различить компиляцию, интерпретацию и ассемблирование.</li> <li>4. Указать преимущества и недостатки различных способов трансляции.</li> <li>5. Изучить структуру компилятора.</li> <li>6. Охарактеризовать лексический, синтаксический и семантический анализ.</li> <li>7. Изучить генерацию объектного кода.</li> <li>8. Пояснить этапы оптимизации программ при трансляции.</li> <li>9. Привести примеры современных трансляторов.</li> <li>10. Изучить ошибки трансляции и методы их устранения.</li> </ol>	<p>Изучение документации к компиляторам (GCC, Clang), работа с примерами трансляции, визуализация этапов трансляции, выполнение практических заданий на компиляцию кода, анализ логов компиляции, подготовка схемы работы компилятора.</p>
<p>Тема 4. Ассемблеры</p>	<ol style="list-style-type: none"> <li>1. Определить понятие ассемблера и его назначение.</li> <li>2. Изучить структуру ассемблерной программы.</li> <li>3. Разобрать форматы команд и способы адресации.</li> <li>4. Указать особенности работы однопроходного и двухпроходного ассемблера.</li> <li>5. Сравнить ассемблер и компилятор.</li> <li>6. Проанализировать примеры простых программ на языке ассемблера.</li> <li>7. Пояснить работу с регистрами процессора.</li> <li>8. Описать процесс формирования объектного кода.</li> <li>9. Изучить директивы ассемблера.</li> <li>10. Рассмотреть популярные ассемблеры (NASM, MASM).</li> </ol>	<p>Решение задач на написание и отладку ассемблерного кода, использование ассемблера в среде разработки, выполнение лабораторных заданий, моделирование выполнения команд, составление таблиц команд.</p>
<p>Тема 5. Загрузчики</p>	<ol style="list-style-type: none"> <li>1. Определить назначение загрузчиков.</li> <li>2. Изучить этапы загрузки операционной системы.</li> <li>3. Рассмотреть различия между загрузчиками первого и второго уровня.</li> <li>4. Изучить пример работы загрузчика GRUB.</li> <li>5. Объяснить процессы размещения программ в памяти.</li> <li>6. Рассмотреть вопросы совместимости загрузчиков.</li> <li>7. Проанализировать схемы взаимодействия BIOS/UEFI с загрузчиком.</li> <li>8. Изучить принципы построения загрузочного сектора.</li> <li>9. Изучить механизм передачи управления ОС.</li> </ol>	<p>Анализ конфигурационных файлов GRUB, практические задания по установке загрузчиков, моделирование процесса загрузки, исследование логов и ошибок загрузки, составление схемы этапов загрузки.</p>

	10. Рассмотреть безопасность при загрузке системы.	
Тема 6. RunTime Interface	<ol style="list-style-type: none"> <li>1. Дать определение среды выполнения (RunTime Environment).</li> <li>2. Изучить роль среды выполнения в системном ПО.</li> <li>3. Рассмотреть взаимодействие RunTime с ОС.</li> <li>4. Изучить обработку исключений в среде выполнения.</li> <li>5. Изучить стек вызовов и управление памятью.</li> <li>6. Рассмотреть работу сборщика мусора.</li> <li>7. Изучить особенности среды выполнения в C/C++ и Java.</li> <li>8. Изучить концепцию времени жизни переменных.</li> <li>9. Рассмотреть механизмы динамической линковки.</li> <li>10. Изучить типичные ошибки во время выполнения и методы их отладки.</li> </ol>	Изучение поведения программ в различных средах исполнения, отладка ошибок времени выполнения, использование валгринда, анализ дампов памяти, выполнение практических заданий по управлению памятью.
Тема 7. Средства разработки	<ol style="list-style-type: none"> <li>1. Перечислить основные средства разработки программного обеспечения.</li> <li>2. Изучить архитектуру IDE (интегрированной среды разработки).</li> <li>3. Изучить функции редактора кода, компилятора, отладчика.</li> <li>4. Рассмотреть системы контроля версий.</li> <li>5. Ознакомиться с CI/CD-подходами.</li> <li>6. Сравнить популярные IDE (Visual Studio, IntelliJ, Eclipse).</li> <li>7. Изучить плагины и расширения среды разработки.</li> <li>8. Разобрать примеры настройки проекта.</li> <li>9. Изучить систему сборки (make, cmake).</li> <li>10. Понять взаимодействие с системой управления проектами (Git, GitHub).</li> </ol>	Практическая работа в IDE, выполнение мини-проекта, настройка CI-сценариев, работа с системами контроля версий, подготовка обзора IDE, разработка и публикация репозитория.
Тема 8. Программирование ОС	<ol style="list-style-type: none"> <li>1. Изучить уровни взаимодействия с ОС при программировании.</li> <li>2. Рассмотреть API операционной системы.</li> <li>3. Изучить работу с файлами, потоками, процессами.</li> <li>4. Изучить работу с сигналами и прерываниями.</li> <li>5. Изучить системные вызовы.</li> <li>6. Изучить многопоточность и синхронизацию.</li> <li>7. Изучить взаимодействие процессов (IPC).</li> <li>8. Изучить механизмы безопасности ОС при выполнении программ.</li> <li>9. Изучить использование POSIX-интерфейсов.</li> <li>10. Изучить пример работы с файловой системой через системные вызовы.</li> </ol>	Разработка программ с использованием системных вызовов, написание многопоточных программ, выполнение заданий на работу с файлами и процессами, отладка программ в Linux, изучение man-страниц, анализ поведения программ на уровне ОС.

### Шкала оценивания

Шкала оценивания	Критерии оценивания
«Отлично»	Обучающийся глубоко и содержательно раскрывает тему самостоятельной работы, не допустив ошибок.

	Ответ носит развернутый и исчерпывающий характер
«Хорошо»	Обучающийся в целом раскрывает тему самостоятельной работы, однако ответ хотя бы на один из них не носит развернутого и исчерпывающего характера
«Удовлетворительно»	Обучающийся в целом раскрывает тему самостоятельной работы и допускает ряд неточностей, фрагментарно раскрывает содержание теоретических вопросов или их раскрывает содержательно, но допуская значительные неточности.
«Неудовлетворительно»	Обучающийся не владеет выбранной темой самостоятельной работы

**6. Оценочные материалы (фонд оценочных средств) для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины**

**6.1. Паспорт фонда оценочных средств**

№	Контролируемые разделы (темы) дисциплины	Код и наименование компетенции	Индикатор достижения компетенции	Наименование оценочного средства
1.	Тема 1. Введение в системное программное обеспечение	ПК-2 Способен разрабатывать требования и проектировать программное обеспечение	ПК-2.1 Выполняет анализ требований к программному обеспечению ПК-2.2 Разрабатывает технические спецификации на программные компоненты и их взаимодействие ПК-2.3 Проектирует программное обеспечение	Опрос, тест, доклад, реферат, экзамен
2.	Тема 2. Электронная вычислительная машина	ПК-2 Способен разрабатывать требования и проектировать программное обеспечение	ПК-2.1 Выполняет анализ требований к программному обеспечению ПК-2.2 Разрабатывает технические спецификации на программные компоненты и их взаимодействие ПК-2.3 Проектирует программное обеспечение	Опрос, тест, доклад, реферат, экзамен
3.	Тема 3. Трансляция	ПК-2 Способен разрабатывать	ПК-2.1 Выполняет анализ требований к программному	Опрос, тест, доклад, реферат,

		требования проектировать программное обеспечение	и обеспечению ПК-2.2 Разрабатывает технические спецификации на программные компоненты и их взаимодействие ПК-2.3 Проектирует программное обеспечение	экзамен
4.	Тема 4. Ассемблеры	ПК-2 Способен разрабатывать требования проектировать программное обеспечение	и ПК-2.1 Выполняет анализ требований к программному обеспечению ПК-2.2 Разрабатывает технические спецификации на программные компоненты и их взаимодействие ПК-2.3 Проектирует программное обеспечение	Опрос, тест, доклад, реферат, экзамен
5.	Тема 5. Загрузчики	ПК-2 Способен разрабатывать требования и проектировать программное обеспечение	ПК-2.1 Выполняет анализ требований к программному обеспечению ПК-2.2 Разрабатывает технические спецификации на программные компоненты и их взаимодействие ПК-2.3 Проектирует программное обеспечение	Опрос, тест, доклад, реферат, экзамен
6.	Тема 6. RunTime Interface	ПК-2 Способен разрабатывать требования и проектировать программное обеспечение	ПК-2.1 Выполняет анализ требований к программному обеспечению ПК-2.2 Разрабатывает технические спецификации на программные компоненты и их взаимодействие ПК-2.3 Проектирует программное обеспечение	Опрос, тест, доклад, реферат, экзамен
7.	Тема 7. Средства разработки	ПК-2 Способен разрабатывать требования и проектировать программное обеспечение	ПК-2.1 Выполняет анализ требований к программному обеспечению ПК-2.2 Разрабатывает технические спецификации на программные	Опрос, тест, доклад, реферат, экзамен

			компоненты и их взаимодействие ПК-2.3 Проектирует программное обеспечение	
8.	Тема 8. Программирование ОС	ПК-2 Способен разрабатывать требования и проектировать программное обеспечение	ПК-2.1 Выполняет анализ требований к программному обеспечению ПК-2.2 Разрабатывает технические спецификации на программные компоненты и их взаимодействие ПК-2.3 Проектирует программное обеспечение	Опрос, тест, доклад, реферат, экзамен

**Этапы формирования компетенций в процессе освоения ОПОП** прямо связаны с местом дисциплин в образовательной программе. Каждый этап формирования компетенции, характеризуется определенными знаниями, умениями и навыками и (или) опытом профессиональной деятельности, которые оцениваются в процессе текущего контроля успеваемости, промежуточной аттестации по дисциплине (практике) и в процессе итоговой аттестации.

Дисциплина «Системное программное обеспечение» является промежуточным этапом комплекса дисциплин, в ходе изучения которых у студентов формируются компетенция ПК-2.

Формирование компетенции ПК-2 начинается с освоения дисциплин Учебная практика: ознакомительная практика, «Информационные сети и коммуникации», «Проектная деятельность», Учебная практика: научно-исследовательская работа (получение первичных навыков научно-исследовательской работы), Учебная практика: технологическая (проектно-технологическая) практика, «Базы данных», «Интеллектуальные системы», «Основы систем искусственного интеллекта», «Параллельное программирование», Программирование на языке низкого уровня».

Завершается работа по формированию у студентов указанных компетенций в ходе изучения дисциплин «Теория вычислительных процессов и языков программирования», «Эргономика», «Технический дизайн», Производственная практика: технологическая (проектно-технологическая) практика, Производственная практика: научно-исследовательская работа, «Цифровая обработка сигналов», «Архитектура вычислительных систем», Производственная практика: преддипломная практика.

Итоговая оценка сформированности компетенций ПК-2 определяется в период Государственная итоговая аттестация: подготовка к сдаче и сдача государственного экзамена, Государственная итоговая аттестация: выполнение и защита выпускной квалификационной работы.

**В процессе изучения дисциплины, компетенции также формируются поэтапно.**

Основными этапами формирования ПК-2 при изучении дисциплины Б1.Д(М).В.5 «Системное программное обеспечение» является последовательное изучение содержательно связанных между собой тем учебных занятий. Изучение каждой темы предполагает овладение студентами необходимыми дескрипторами (составляющими) компетенций. Для оценки уровня сформированности компетенций в процессе изучения дисциплины предусмотрено проведение текущего контроля успеваемости по темам (разделам) дисциплины и промежуточной аттестации по дисциплине – экзамен.

## **6.2. Контрольные задания и материалы, необходимые для оценки знаний, умений и навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы**

### **6.2.1. Контрольные вопросы по темам (разделам) для опроса на занятиях**

Тема (раздел)	Вопросы
Тема 1. Введение в системное программное обеспечение	ПК-2 1. Понятие и роль системного программного обеспечения в вычислительной системе. 2. Отличие системного ПО от прикладного программного обеспечения. 3. Основные компоненты системного программного обеспечения. 4. Эволюция системного ПО и этапы его развития. 5. Взаимодействие пользователя с системным программным обеспечением. 6. Задачи и функции операционной системы как основы системного ПО. 7. Виды системных программ: драйверы, компиляторы, утилиты и др. 8. Примеры системного ПО в современных компьютерных системах.
Тема 2. Электронная вычислительная машина	ПК-2 1. Структура и архитектура ЭВМ. 2. Основные устройства ЭВМ и их функции (процессор, память, устройства ввода/вывода). 3. Архитектура фон Неймана и её особенности. 4. Принцип работы процессора и цикл выполнения команды. 5. Уровни представления информации в ЭВМ. 6. Адресация памяти и регистровая архитектура. 7. Влияние архитектуры ЭВМ на разработку ПО. 8. Классификация ЭВМ и их применимость.
Тема 3. Трансляция	ПК-2 1. Понятие трансляции и её цели. 2. Этапы трансляции программного кода. 3. Различия между компиляцией, интерпретацией и ассемблированием. 4. Структура и функции компилятора.

	<p>5. Лексический, синтаксический и семантический анализ.</p> <p>6. Генерация и оптимизация объектного кода.</p> <p>7. Ошибки трансляции и методы их обнаружения.</p> <p>8. Примеры трансляторов в различных языках программирования.</p>
Тема 4. Ассемблеры	<p>ПК-2</p> <p>1. Понятие и назначение ассемблера.</p> <p>2. Сравнение ассемблера и компилятора.</p> <p>3. Формат и структура ассемблерной программы.</p> <p>4. Прimitives команды и директивы ассемблера.</p> <p>5. Процесс ассемблирования и его фазы.</p> <p>6. Обратимость кода на ассемблере и отладка.</p> <p>7. Роль ассемблера в системном программировании.</p> <p>8. Примеры программ на ассемблере и их анализ.</p>
Тема 5. Загрузчики	<p>ПК-2</p> <p>1. Назначение и функции загрузчиков.</p> <p>2. Виды загрузчиков: одно- и двухступенчатые.</p> <p>3. Этапы загрузки программ в память.</p> <p>4. Отличия статической и динамической загрузки.</p> <p>5. Взаимодействие загрузчика с ОС и транслятором.</p> <p>6. Форматы исполняемых файлов и их обработка загрузчиком.</p> <p>7. Загрузчики в контексте современных операционных систем.</p> <p>8. Примеры реализации загрузчиков.</p>
Тема 6. RunTime Interface	<p>ПК-2</p> <p>1. Понятие RunTime-среды и её задачи.</p> <p>2. Компоненты RunTime-среды: стек вызовов, управление памятью и пр.</p> <p>3. Обработка ошибок и исключений во время выполнения.</p> <p>4. Механизмы управления ресурсами во время исполнения программы.</p> <p>5. Связь RunTime и операционной системы.</p> <p>6. Интерпретаторы и виртуальные машины как часть RunTime.</p> <p>7. Примеры RunTime-сред для различных языков (Java, .NET и др.).</p> <p>8. Проблемы и особенности разработки с учётом RunTime.</p>
Тема 7. Средства разработки	<p>ПК-2</p> <p>1. Классификация средств разработки ПО: IDE, редакторы, отладчики.</p> <p>2. Обзор популярных сред разработки: Visual Studio, Eclipse, IntelliJ IDEA и др.</p> <p>3. Интегрированная среда разработки: преимущества и структура.</p> <p>4. Инструменты для сборки и управления проектами.</p> <p>5. Отладка и профилирование программ.</p> <p>6. Системы контроля версий и их роль в разработке.</p>

	7. Автоматизация сборки и CI/CD-практики. 8. Средства разработки в контексте системного программного обеспечения.
Тема 8. Программирование ОС	ПК-2 1. Уровни доступа к функционалу операционной системы. 2. Системные вызовы и их использование в прикладных программах. 3. Программирование работы с файлами, процессами, памятью. 4. Интерфейсы взаимодействия пользователя и ядра ОС. 5. Особенности программирования под Windows, Linux, Unix. 6. Примеры системных программ и утилит. 7. Безопасность и управление правами в системных приложениях. 8. Примеры использования API ОС в системном программировании.

### Шкала оценивания ответов на вопросы

Шкала оценивания	Критерии оценивания
«Отлично»	Обучающийся глубоко и содержательно раскрывает ответ на каждый теоретический вопрос, не допустив ошибок. Ответ носит развернутый и исчерпывающий характер.
«Хорошо»	Обучающийся в целом раскрывает теоретические вопросы, однако ответ хотя бы на один из них не носит развернутого и исчерпывающего характера.
«Удовлетворительно»	Обучающийся в целом раскрывает теоретические вопросы и допускает ряд неточностей, фрагментарно раскрывает содержание теоретических вопросов или их раскрывает содержательно, но допуская значительные неточности.
«Неудовлетворительно»	Обучающийся не знает ответов на поставленные теоретические вопросы.

### 6.2.2. Темы для докладов

Тема (раздел)	Вопросы
Тема 1. Введение в системное программное обеспечение	ПК-2 1. Роль системного программного обеспечения в функционировании компьютера. 2. Принципы работы операционных систем и их связь с аппаратным обеспечением. 3. Основные компоненты системного программного обеспечения. 4. История развития системного программного обеспечения. 5. Разделение системного и прикладного программного обеспечения. 6. Операционная система как посредник между аппаратным и программным обеспечением. 7. Модели архитектуры операционных систем: монолитная, микроядерная, гибридная.

	<p>8. Влияние системного программного обеспечения на производительность компьютера.</p> <p>9. Современные тенденции в развитии операционных систем.</p> <p>10. Проблемы безопасности и защиты данных в контексте системного программного обеспечения.</p>
<p>Тема 2. Электронная вычислительная машина</p>	<p>ПК-2</p> <p>1. Архитектура вычислительной машины: компоненты и взаимодействие.</p> <p>2. Основные функции процессора: управление, арифметико-логические операции и взаимодействие с памятью.</p> <p>3. Роль системной и оперативной памяти в вычислительных машинах.</p> <p>4. Взаимодействие процессора и периферийных устройств: шины данных и управления.</p> <p>5. Процессоры с множественными ядрами: преимущества и области применения.</p> <p>6. Роль ввода/вывода данных в вычислительных системах.</p> <p>7. Применение вычислительных машин в различных областях: от личных компьютеров до суперкомпьютеров.</p> <p>8. Современные вычислительные архитектуры и их особенности.</p> <p>9. Энергопотребление и тепловыделение в электронных вычислительных машинах.</p> <p>10. Технологии производства процессоров и их влияние на производительность.</p>
<p>Тема 3. Трансляция</p>	<p>ПК-2</p> <p>1. Процесс трансляции программ: от исходного кода до машинного кода.</p> <p>2. Компиляция и интерпретация: основные различия и области применения.</p> <p>3. Этапы компиляции: лексический анализ, синтаксический анализ, генерация кода.</p> <p>4. Проблемы трансляции высокоуровневых языков программирования в машинный код.</p> <p>5. Роль промежуточных представлений в процессе компиляции.</p> <p>6. Оптимизация кода на этапе компиляции.</p> <p>7. Инструменты для трансляции: компиляторы и интерпретаторы.</p> <p>8. Различия между статической и динамической компиляцией.</p> <p>9. Роль трансляторов в разработке многозадачных систем.</p> <p>10. Проблемы совместимости при трансляции и их решение.</p>
<p>Тема 4. Ассемблеры</p>	<p>ПК-2</p> <p>1. Роль ассемблера в процессе разработки программного обеспечения.</p> <p>2. Синтаксис и структура команд ассемблера.</p> <p>3. Ассемблер и машинный код: отличие и связи.</p> <p>4. Преимущества и недостатки использования ассемблера.</p> <p>5. Ассемблеры для различных архитектур процессоров: x86, ARM, MIPS.</p> <p>6. Оптимизация программ на ассемблере.</p> <p>7. Применение ассемблера в встраиваемых системах.</p> <p>8. Процесс написания и отладки программ на ассемблере.</p> <p>9. Сравнение ассемблера с высокоуровневыми языками программирования.</p>

	10. Перспективы использования ассемблера в современных вычислительных системах.
Тема 5. Загрузчики	<p>ПК-2</p> <ol style="list-style-type: none"> <li>1. Задачи и функции загрузчика в операционной системе.</li> <li>2. Как работает процесс загрузки операционной системы.</li> <li>3. Виды загрузчиков: первичный, вторичный, многозадачный.</li> <li>4. Загрузчики для различных архитектур и операционных систем.</li> <li>5. Загрузчик как часть начальной загрузки системы.</li> <li>6. Взаимодействие загрузчика с BIOS/UEFI.</li> <li>7. Загрузчики и их роль в безопасной загрузке системы.</li> <li>8. Процесс инсталляции операционной системы через загрузчик.</li> <li>9. Грузчики на основе GRUB: как работает и где используется.</li> <li>10. Проблемы совместимости и решения при использовании различных загрузчиков.</li> </ol>
Тема 6. RunTime Interface	<p>ПК-2</p> <ol style="list-style-type: none"> <li>1. Определение и роль RunTime Interface в работе программ.</li> <li>2. Взаимодействие приложений с операционной системой через RunTime Interface.</li> <li>3. Этапы работы программы в процессе выполнения через Runtime.</li> <li>4. Преимущества использования RunTime Interface для динамического связывания.</li> <li>5. Различия между статической и динамической линковкой на уровне RunTime.</li> <li>6. Работа с памятью и ресурсоемкими операциями через Runtime.</li> <li>7. Важность управления памятью через Runtime Interface.</li> <li>8. Роль Runtime в интерпретации и выполнении высокоуровневых языков.</li> <li>9. Основы реализации RunTime библиотеки и ее взаимодействие с ОС.</li> <li>10. Современные методы улучшения производительности при использовании Runtime Interface.</li> </ol>
Тема 7. Средства разработки	<p>ПК-2</p> <ol style="list-style-type: none"> <li>1. Обзор современных средств разработки программного обеспечения.</li> <li>2. Интегрированные среды разработки (IDE): преимущества и недостатки.</li> <li>3. Роль компиляторов и отладчиков в процессе разработки.</li> <li>4. Средства для работы с базами данных и их интеграция в средах разработки.</li> <li>5. Управление версиями: системы контроля версий в процессе разработки.</li> <li>6. Использование тестовых фреймворков и инструментов для автоматизации тестирования.</li> <li>7. Контейнеризация и виртуализация как средства разработки программ.</li> <li>8. Средства для разработки многозадачных и распределённых систем.</li> <li>9. Программные средства для анализа и профилирования кода.</li> <li>10. Будущее разработки: искусственный интеллект и машинное обучение в инструментах разработки.</li> </ol>

Тема 8. Программирование ОС	<p>ПК-2</p> <ol style="list-style-type: none"> <li>1. Программирование операционных систем: основы и принципы.</li> <li>2. Архитектура ядра операционной системы и принципы его разработки.</li> <li>3. Взаимодействие ядра ОС с процессами и потоками.</li> <li>4. Системные вызовы и их роль в разработке программ для ОС.</li> <li>5. Управление памятью в операционных системах: виртуальная память, сегментация, страничная организация.</li> <li>6. Программирование устройства ввода-вывода и драйверов устройств.</li> <li>7. Разработка многозадачных и многопроцессных приложений в операционных системах.</li> <li>8. Обработка ошибок и исключений в операционных системах.</li> <li>9. Программирование на уровне пользователя: создание системных утилит и приложений.</li> <li>10. Безопасность операционных систем: принципы программирования для защиты.</li> </ol>
--------------------------------	--

### Шкала оценивания

Шкала оценивания	Критерии оценивания
«Отлично»	Обучающийся глубоко и содержательно раскрывает тему доклада, не допустив ошибок. Ответ носит развернутый и исчерпывающий характер.
«Хорошо»	Обучающийся в целом раскрывает тему доклада, однако ответ хотя бы на один из них не носит развернутого и исчерпывающего характера.
«Удовлетворительно»	Обучающийся в целом раскрывает тему доклада и допускает ряд неточностей, фрагментарно раскрывает содержание теоретических вопросов или их раскрывает содержательно, но допуская значительные неточности.
«Неудовлетворительно»	Обучающийся не владеет выбранной темой

#### 6.2.3. Оценочные средства остаточных знаний (тест)

##### ПК-2.

##### 1. Что называется операционной системой (ОС)?

- 1) Система программирования
- 2) Совокупность программ, используемых для операций с документами
- 3) Набор программ, обеспечивающий управление всеми ресурсами ЭВМ и осуществляющий общую организацию процесса обработки информации и интерфейс ЭВМ с пользователем
- 4) Программы для поиска и уничтожения компьютерных вирусов

##### 2. Какая ОС, из указанных ниже, — многозадачная?

- 1) OS/2
- 2) MS-DOS
- 3) MSX
- 4) PC-DOS

**3. Какая ОС, из указанных ниже, — многопользовательская?**

- 1) MS-DOS
- 2) MSX
- 3) Windows 3.x
- 4) UNIX

**4. Какая ОС, из указанных ниже, относится к системам пакетной обработки?**

- 1) ОС ЕС
- 2) UNIX
- 3) RT/11
- 4) Windows 3.x

**5. Какой критерий характеризует ОС, как ОС с разделением времени?**

1) Главной целью системы является максимальная пропускная способность, то есть решение максимального числа задач в единицу времени.

2) Каждому пользователю системы предоставляется терминал, с которого он может вести диалог со своей программой, а каждой задаче выделяется квант процессорного времени

3) В начале работы формируется пакет заданий, каждое задание содержит требование к системным ресурсам; из этого пакета заданий формируется множество одновременно выполняемых задач

4) ОС применяются для управления различными техническими объектами, такими, например, как станок, спутник, научная экспериментальная установка или технологическими процессами, такими, как гальваническая линия, доменный процесс и т.п.

**6. Как можно загрузить ОС Windows в безопасном режиме?**

- 1) Нажав при загрузке клавишу F8
- 2) Нажав при загрузке клавишу F1
- 3) Нажав при загрузке клавишу F4
- 4) Нажав при загрузке клавишу F12

**7. От чего зависит степень сжатия файла при архивировании?**

- 1) Только от типа файла
- 2) От типа файла и программы архиватора
- 3) Только от программы архиватора
- 4) От производительности компьютера

**8. Чем отличается архивный файл от исходного?**

- 1) Тем, что он занимает меньше места на диске
- 2) Тем, что доступ к нему занимает меньше времени
- 3) Тем, что он в большей степени удобен для редактирования

- 4) Тем, что он легче защищается от вирусов

### **9. Какой критерий характеризует ОС, как ОС реального времени?**

1) ОС применяются для управления различными техническими объектами, такими, например, как станок, спутник, научная экспериментальная установка или технологическими процессами, такими, как гальваническая линия, доменный процесс и т.п.

2) Главной целью системы является максимальная пропускная способность, то есть решение максимального числа задач в единицу времени.

3) В начале работы формируется пакет заданий, каждое задание содержит требование к системным ресурсам; из этого пакета заданий формируется множество одновременно выполняемых задач.

4) Каждому пользователю системы предоставляется терминал, с которого он может вести диалог со своей программой, а каждой задаче выделяется квант процессорного времени.

### **10. Что такое дефрагментация диска?**

1) Дефрагментация - это процесс оптимизации логической структуры раздела диска с целью обеспечения хранения файлов строго в алфавитном порядке.

2) Дефрагментация - это процесс оптимизации логической структуры раздела диска с целью обеспечения хранения файлов строго по возрастанию размера.

3) Дефрагментация - это процесс оптимизации логической структуры раздела диска с целью обеспечения хранения файлов в смежной последовательности кластеров.

4) Дефрагментация - это процесс оптимизации логической структуры раздела диска с целью обеспечения хранения файлов строго по возрастанию даты.

### **11. На что влияет фрагментация файлов на диске?**

1) Скорость чтения файлов с носителя уменьшается.

2) Скорость работы центрального процессора уменьшается.

3) Скорость работы оперативной памяти компьютера уменьшается.

4) Ни на что не влияет.

### **12. Основное предназначение расширения имени файла?**

1) Указать что файл записан

2) Указать операционной системе, какую программу следует вызвать для открытия файла

3) Напомнить пользователю, что файл пора копировать

4) Указать место создания файла

### **13. Для чего предназначена команда COPY в Windows?**

1) Для копирования файлов и каталогов

- 2) Для копирования только
- 3) Для копирования одного или нескольких файлов из одного расположения в другое
- 4) Для копирования файлов и подкаталогов

**14. Что должна знать прикладная программа при обращении к файлу?**

- 1) Программа должна знать имя файла
- 2) Программа должна знать, где на физическом носителе расположен файл (цилиндр, головку, сектор)
- 3) Программа должна знать имя файла и его размер
- 4) Программа должна знать имя файла его размер и атрибуты

**15. Что такое кластер?**

- 1) Совокупность цилиндров
- 2) Совокупность смежных секторов равных степени двойки
- 3) Синоним сектора
- 4) Место для записи блока начальной загрузки

**16. Какое из указанных ниже имен файлов удовлетворяет маске: ?r???\*.\***

- 1) Group7311.txt
- 2) grup.doc
- 3) Vgroup.c
- 4) C\_group

**17. Что такое файловая система ОС?**

- 1) Это место постоянного хранения различных видов информации
- 2) Место на диске, в которых хранятся файлы и информация о их свойствах (имя, размер, дата создания и т.д.)
- 3) Это методы и структуры данных, которые используются ОС для хранения файлов на диске или его разделе
- 4) Область для хранения информации, необходимой для загрузки ОС (размер раздела, количество свободного места, размер кластера и т.д.)

**18. Какие структуры данных могут располагаться в корневом каталоге?**

- 1) Каталоги и файлы
- 2) Только каталоги
- 3) Только файлы
- 4) Некоторая служебная информация

**19. Что делает команда операционной системы Windows CD C:\INFORMATICA\DOC?**

- 1) Устанавливает путь к файлам каталога INFORMATICA

- 2) Устанавливает путь к подкаталогу DOC корневого каталога
- 3) Устанавливает в качестве текущего каталога INFORMATICA
- 4) Устанавливает в качестве текущего каталога DOC

**20. Что делает команда операционной системы Windows DIR C:\INFORMATICA\DOC?**

- 1) Выводит имена файлов и подкаталогов по указанному пути
- 2) Выводит имена файлов и подкаталогов из текущего каталога
- 3) Выводит имена всех файлов из корневого каталога
- 4) Выводит имена всех каталогов диска

**21. К системным программам относится:**

- 1) MS Windows
- 2) MS Excel
- 3) MS Word
- 4) Paint

**22. Для каких целей необходимо системное ПО:**

- 1) для решения задач из проблемных областей
- 2) для управления ресурсами ЭВМ
- 3) для расширения возможностей ОС
- 4) для управления дополнительными устройствами

**23. Порядок размещения, хранения и именования данных на носителе информации:**

- 1) система шифров
- 2) защитная система
- 3) файловая система
- 4) знаковая система

**24. Совокупность всех программ, предназначенных для выполнения на компьютере:**

- 1) программное обеспечение
- 2) система программирования
- 3) операционная система
- 4) файловая система

**25. К системным программам относится:**

- 1) Paint
- 2) MS Word
- 3) Антивирусы
- 4) MS Excel

**26. Последовательность обращения к дискам на этапе загрузки компьютера определяет(ют):**

- 1) BIOS
- 2) операционная система
- 3) прикладные программы
- 4) система программирования

**27. Служебные программы для проверки и настройки компьютера:**

- 1) файлы
- 2) антивирусы
- 3) утилиты
- 4) фото

**28. Какие программы не относятся к группе сервисного ПО:**

- 1) драйверы устройств
- 2) программы для дефрагментации дискового пространства
- 3) программы для организации сетевого взаимодействия
- 4) графические программы

**29. Программа, которая выполняет команды пользователя, введённые в командной строке:**

- 1) командный регистр
- 2) главный процессор
- 3) командный процессор
- 4) главный регистр

**30. Программы, предназначенные для обмена данными с дисковыми, клавиатурой, монитором и принтером использует система:**

- 1) ввода/вывода
- 2) загрузки данных
- 3) копирования
- 4) удаления

**31. Что из перечисленного относится к программам, управляющим операциями с внешними устройствами?**

- 1) Операционная система
- 2) Драйверы устройств
- 3) Прикладные программы
- 4) Утилиты

**32. Какой из компонентов ОС отвечает за управление памятью?**

- 1) Диспетчер процессов
- 2) Диспетчер памяти
- 3) Файловая система
- 4) Утилиты

**33. Что такое многозадачность?**

- 1) Одновременное выполнение нескольких процессов на одном процессоре
- 2) Одновременное выполнение нескольких процессов на нескольких процессорах
- 3) Одновременная работа нескольких пользователей с одной программой
- 4) Одновременное выполнение нескольких приложений на одном экране

**34. Как называется интерфейс между операционной системой и аппаратным обеспечением?**

- 1) Системный вызов
- 2) Ядро
- 3) Оболочка
- 4) Файловая система

**35. Какая из операционных систем была разработана для работы на серверных компьютерах и отличается высокой надежностью и безопасностью?**

- 1) Windows
- 2) Linux
- 3) MS-DOS
- 4) UNIX

**36. Что из перечисленного является задачей системы ввода-вывода (I/O)?**

- 1) Управление процессами
- 2) Обработка запросов от пользователя
- 3) Взаимодействие с внешними устройствами
- 4) Обработка ошибок в операционной системе

**37. Какой тип памяти используется для хранения операционной системы в момент загрузки?**

- 1) Оперативная память
- 2) Постоянная память
- 3) Кэш-память
- 4) Виртуальная память

**38. Что из перечисленного является функцией ядра операционной системы?**

- 1) Обработка системных ошибок
- 2) Предоставление пользователю интерфейса
- 3) Управление взаимодействием с внешними устройствами
- 4) Осуществление настройки параметров пользовательских приложений

**39. Что из ниже перечисленного является примером монолитного ядра?**

- 1) Linux
- 2) Windows NT
- 3) macOS
- 4) MS-DOS

**40. Что такое виртуальная память?**

- 1) Способ организации постоянной памяти
- 2) Механизм, который позволяет программе использовать больше памяти, чем доступно в реальной оперативной памяти
- 3) Способ защиты от сбоев в операционной системе
- 4) Механизм для хранения данных на жестком диске

**41. Какую роль играет драйвер устройства в операционной системе?**

- 1) Управляет запуском программ
- 2) Обеспечивает взаимодействие между операционной системой и аппаратным устройством
- 3) Определяет параметры дисплея
- 4) Защищает данные от вирусов

**42. Какая из перечисленных операционных систем является мультизадачной?**

- 1) MS-DOS
- 2) UNIX
- 3) Windows 3.x
- 4) MSX

**43. Какую роль играет буферизация в операционной системе?**

- 1) Повышает скорость передачи данных между процессами
- 2) Служит для экономии памяти
- 3) Ускоряет процесс загрузки операционной системы
- 4) Повышает эффективность работы с внешними устройствами

**44. Как называется процесс восстановления операционной системы из ошибок после сбоя?**

- 1) Резервное копирование
- 2) Дефрагментация
- 3) Восстановление системы
- 4) Системное обновление

**45. Какой из следующих файловых систем является самой старой?**

- 1) FAT32
- 2) NTFS
- 3) EXT4

#### 4) FAT16

#### Ключ к тесту:

1.3	2.1	3.4	4.1	5.2	6.1	7.2	8.1	9.1
10.3	11.1	12.2	13.3	14.1	15.2	16.1	17.3	18.1
19.4	20.1	21.1	22.2	23.3	24.1	25.3	26.1	27.3
28.4	29.3	30.1	31.2	32.2	33.1	34.2	35.2	36.3
37.1	38.3	39.1	40.2	41.2	42.2	43.4	44.3	45.4

#### Шкала оценивания результатов тестирования

% верных решений (ответов)	Шкала оценивания
85 - 100	отлично
70 - 84	хорошо
50- 69	удовлетворительно
0 - 49	неудовлетворительно

#### 6.2.4. Примеры задач при разборе конкретных ситуаций

##### *Тема 1. Введение в системное программное обеспечение*

##### **ПК-2.**

1. Составить таблицу сравнения трёх операционных систем (Windows, Linux, macOS), указав компоненты системного ПО: ядро, драйверы, загрузчики, библиотеки.

2. Нарисовать схему взаимодействия аппаратного обеспечения и прикладной программы через системное программное обеспечение.

3. Установить и настроить виртуальную машину (например, VirtualBox) с ОС Linux, описать шаги установки.

##### *Тема 2. Электронная вычислительная машина*

##### **ПК-2.**

1. Реализовать на языке C программу, считывающую содержимое регистра времени (timestamp) и замеряющую время выполнения простого цикла.

2. Используя эмулятор (например, CPUlator или EduMIPS64), продемонстрировать выполнение простых инструкций процессора и проанализировать их влияние на регистры.

3. Смоделировать процесс загрузки данных из оперативной памяти в кэш, описать влияние кэширования на скорость выполнения программы.

##### *Тема 3. Трансляция*

##### **ПК-2.**

1. Написать простую программу на языке C и собрать её вручную через gcc по этапам: препроцессинг, компиляция, ассемблирование, компоновка. Привести промежуточные файлы.

2. Изменить код программы на C и проследить, как изменения влияют на ассемблерный код (через gcc -S).

3. Использовать `strace` в Linux, чтобы проследить вызовы при запуске скомпилированного приложения.

#### *Тема 4. Ассемблеры*

##### **ПК-2.**

1. Написать программу на ассемблере NASM, выводящую строку «Hello, World!» в консоль. Скомпилировать и запустить её на Linux.

2. Добавить в программу на ассемблере ввод строки с клавиатуры и обратный вывод.

3. Использовать GDB для пошагового выполнения ассемблерной программы, отследить содержимое регистров.

#### *Тема 5. Загрузчики*

##### **ПК-2.**

1. Создать MBR-загрузчик (boot sector) размером 512 байт, который выводит текст на экран. Запустить с помощью QEMU.

2. Проанализировать структуру ELF-файла с помощью команды `readelf` или `objdump`.

3. Написать небольшой загрузочный код, который загружает исполняемый файл из памяти (используя BIOS-интерфейсы или GRUB).

#### *Тема 6. RunTime Interface*

##### **ПК-2.**

1. Написать на C программу, использующую системный вызов `fork()`, и объяснить, что происходит при клонировании процесса.

2. Добавить вызов `exec()` в дочерний процесс для запуска другой программы.

3. Реализовать обработку ошибок системных вызовов (`open`, `read`, `write`) с выводом соответствующих сообщений ( `perror`).

#### *Тема 7. Средства разработки*

##### **ПК-2.**

1. Создать Makefile для проекта на C, содержащего несколько файлов. Описать зависимость и автоматическую перекомпиляцию.

2. Использовать отладчик `gdb` для пошагового выполнения программы, анализа переменных и поиска ошибки сегментации.

3. Сравнить использование IDE (например, Code::Blocks или VSCode) и сборки в командной строке: собрать один и тот же проект разными способами и зафиксировать отличия.

#### *Тема 8. Программирование ОС*

##### **ПК-2.**

1. Написать программу на C, использующую многопоточность через `pthread_create()`, которая считает сумму элементов массива.

2. Реализовать взаимодействие процессов с помощью именованных каналов (`mkfifo`) или сокетов `Unix domain`.

3. Написать код для чтения и записи в файл, используя системные вызовы `open`, `read`, `write`, `close`, и вывести общее количество прочитанных байт.

### **Шкала оценивания**

Шкала оценивания	Критерии оценивания
«Отлично»	обучающийся ясно изложил условие задачи, решение обосновал
«Хорошо»	обучающийся ясно изложил условие задачи, но в обосновании решения имеются сомнения;
«Удовлетворительно»	обучающийся изложил решение задачи, но обосновал его формулировками обыденного мышления;
«Неудовлетворительно»	обучающийся не уяснил условие задачи, решение не обосновал либо не сдал работу на проверку (в случае проведения решения задач в письменной форме).

### 6.2.3. Темы для рефератов

Тема (раздел)	Вопросы
Тема 1. Введение в системное программное обеспечение	<p>ПК-2</p> <ol style="list-style-type: none"> <li>1. Определение системного программного обеспечения и его роль в вычислительных системах.</li> <li>2. Сравнение системного и прикладного программного обеспечения: основные отличия.</li> <li>3. Функции системного программного обеспечения: ядро, драйверы, утилиты.</li> <li>4. Архитектура системного программного обеспечения: многослойный подход.</li> <li>5. Роль операционных систем в системном программировании.</li> <li>6. Примеры системного программного обеспечения: Windows, Linux, Unix.</li> <li>7. Будущее системного программного обеспечения: новые тренды и технологии.</li> <li>8. Влияние системного программирования на производительность вычислительных систем.</li> <li>9. Основы проектирования системного программного обеспечения: принципы и подходы.</li> <li>10. Применение системного программирования в различных областях: от встраиваемых систем до облачных решений.</li> </ol>
Тема 2. Электронная вычислительная машина	<p>ПК-2</p> <ol style="list-style-type: none"> <li>1. Архитектура электронных вычислительных машин: основные компоненты и их функции.</li> <li>2. Принципы работы процессоров: от выполнения инструкций до обработки данных.</li> <li>3. Влияние архитектуры ЭВМ на производительность системного программного обеспечения.</li> <li>4. Память в электронных вычислительных машинах: виды и иерархия.</li> <li>5. Ввод-вывод в ЭВМ: механизмы и устройства.</li> <li>6. Применение параллельных вычислений в современных ЭВМ.</li> <li>7. Эволюция электронных вычислительных машин: от первых моделей до современных.</li> <li>8. Роль ЭВМ в разработке системного программного обеспечения.</li> <li>9. Обзор современных трендов в архитектуре ЭВМ.</li> <li>10. Примеры использования ЭВМ в научных и коммерческих приложениях.</li> </ol>
Тема 3. Трансляция	<p>ПК-2</p> <ol style="list-style-type: none"> <li>1. Понятие трансляции и её роль в системном программировании.</li> </ol>

	<p>2. Этапы трансляции: лексический анализ, синтаксический анализ, семантический анализ.</p> <p>3. Отличия компиляции от интерпретации: преимущества и недостатки.</p> <p>4. Примеры трансляторов: от языков высокого уровня до ассемблера.</p> <p>5. Влияние трансляции на производительность программного обеспечения.</p> <p>6. Инструменты для трансляции: компиляторы и интерпретаторы.</p> <p>7. Оптимизация трансляции: методы и примеры.</p> <p>8. Трансляция в контексте многопоточных программ.</p> <p>9. Будущее трансляции: новые подходы и технологии.</p> <p>10. Примеры трансляции в различных языках программирования.</p>
Тема 4. Ассемблеры	<p>ПК-2</p> <p>1. Определение ассемблера и его роль в системном программировании.</p> <p>2. Структура ассемблерного языка: синтаксис и семантика.</p> <p>3. Принципы работы ассемблера: от исходного кода до объектного файла.</p> <p>4. Различия между ассемблерами для разных архитектур процессоров.</p> <p>5. Применение ассемблеров в оптимизации производительности программ.</p> <p>6. Примеры использования ассемблера в реальных приложениях.</p> <p>7. Инструменты для работы с ассемблерным кодом: отладчики и редакторы.</p> <p>8. Ассемблер и программирование низкого уровня: преимущества и недостатки.</p> <p>9. Будущее ассемблеров в контексте современных языков программирования.</p> <p>10. Роль ассемблера в системном программировании и разработке драйверов.</p>
Тема 5. Загрузчики	<p>ПК-2</p> <p>1. Определение загрузчика и его роль в работе операционной системы.</p> <p>2. Процесс загрузки программы: этапы и механизмы.</p> <p>3. Различия между статическими и динамическими загрузчиками.</p> <p>4. Применение загрузчиков в современных операционных системах.</p> <p>5. Загрузчики и управление памятью: взаимосвязь и особенности.</p> <p>6. Будущее загрузчиков: новые технологии и подходы.</p> <p>7. Примеры загрузчиков: от классических до современных.</p> <p>8. Роль загрузчиков в обеспечении безопасности программ.</p> <p>9. Загрузчики в контексте встраиваемых систем.</p> <p>10. Оптимизация загрузчиков для повышения производительности.</p>
Тема 6. RunTime Interface	<p>ПК-2</p> <p>1. Понятие Runtime Interface и его значение в системном программировании.</p> <p>2. Роль интерфейсов времени выполнения в управлении памятью.</p> <p>3. Примеры интерфейсов времени выполнения в различных языках программирования.</p> <p>4. Взаимодействие Runtime Interface с операционной системой.</p> <p>5. Применение интерфейсов времени выполнения в управлении потоками.</p>

	<p>6. Будущее Runtime Interface: новые тренды и технологии.</p> <p>7. Оптимизация работы Runtime Interface для повышения производительности.</p> <p>8. Сравнительный анализ различных Runtime Interface.</p> <p>9. Роль Runtime Interface в разработке приложений для мобильных устройств.</p> <p>10. Влияние Runtime Interface на безопасность программного обеспечения.</p>
Тема 7. Средства разработки	<p>ПК-2</p> <p>1. Обзор современных средств разработки системного программного обеспечения.</p> <p>2. Интегрированные среды разработки (IDE): преимущества и недостатки.</p> <p>3. Инструменты для отладки системного программного обеспечения.</p> <p>4. Применение средств тестирования в системном программировании.</p> <p>5. Средства версионного контроля: их роль в разработке.</p> <p>6. Будущее средств разработки: новые технологии и их влияние на программирование.</p> <p>7. Сравнительный анализ популярных средств разработки.</p> <p>8. Инструменты для автоматизации сборки и развертывания программного обеспечения.</p> <p>9. Применение средств разработки в контексте встраиваемых систем.</p> <p>10. Роль открытых средств разработки в системном программировании.</p>
Тема 8. Программирование ОС	<p>ПК-2</p> <p>1. Основы программирования операционных систем: подходы и методы.</p> <p>2. Разработка драйверов устройств: принципы и примеры.</p> <p>3. Взаимодействие приложений с ОС: системные вызовы и API.</p> <p>4. Программирование многопоточных приложений: вызовы и техники.</p> <p>5. Оптимизация программ для работы под управлением ОС.</p> <p>6. Будущее программирования ОС: новые технологии и тенденции.</p> <p>7. Примеры системного программирования в различных операционных системах.</p> <p>8. Программирование для реального времени: особенности и примеры.</p> <p>9. Роль программирования ОС в разработке мобильных приложений.</p> <p>10. Применение системного программирования в облачных вычислениях.</p>

### Шкала оценивания

Шкала оценивания	Критерии оценивания
«Отлично»	Обучающийся глубоко и содержательно раскрывает тему самостоятельной работы, не допустив ошибок. Ответ носит развернутый и исчерпывающий характер.
«Хорошо»	Обучающийся в целом раскрывает тему самостоятельной работы, однако ответ хотя бы на один из них не носит развернутого и исчерпывающего характера.

«Удовлетворительно»	Обучающийся в целом раскрывает тему самостоятельной работы и допускает ряд неточностей, фрагментарно раскрывает содержание теоретических вопросов или их раскрывает содержательно, но допуская значительные неточности.
«Неудовлетворительно»	Обучающийся не владеет выбранной темой самостоятельной работы

### **6.2.6. Индивидуальные задания для курсовой работы (проекта)**

КР и КП по дисциплине «Системное программное обеспечение» рабочей программой и учебным планом не предусмотрены.

## **6.3. ОЦЕНОЧНЫЕ СРЕДСТВА ПРОМЕЖУТОЧНОГО КОНТРОЛЯ**

**Вопросы для подготовки к промежуточной аттестации по итогам освоения дисциплины Системное программное обеспечение:**

### **ПК-2.**

1. СПО: определение, задачи, характеристики, виды.
2. Цели и задачи системного программирования. Системные инструментальные средства.
3. Трансляция. Особенности трансляции языков программирования. Этапы трансляции.
4. Необходимые условия трансляции. Лексика, синтаксис и семантика языков программирования.
5. Вычислительная машина. Виды, структура, принцип работы. Система команд и способы адресации.
6. Наиболее распространенные вычислительные машины. Архитектура компьютера.
7. Абстрактные типы данных: понятие, виды, способы реализации.
8. Формальное описание структур данных и вычислительных конструкций.
9. Базовые вычислительные конструкции: присвоение, выражение, ветвление, повторение, применение.
10. Реализация базовых вычислительных конструкций на низком уровне.
11. Теория формальных грамматик. Определение основных операций с символами, алфавитами, языками. Способы определения языка.
12. Способы задания грамматики. Форма Бекуса-Наура. Автоматы для распознавания и вывода строк.
13. Иерархия грамматик. Проблемы и задачи в теории грамматик.
14. Вывод строки по грамматике. Распознавание свойств грамматик. Классы грамматик.
15. Регулярные грамматики. Регулярные выражения. Регулярные распознаватели.
16. Ассемблер. Синтаксис программы на ассемблере. Структура и принцип работы.
17. Алгоритм работы ассемблера. Структура программы на ассемблере.

18. Загрузчик, компоновщик. Виды и принципы работы. Дополнительные возможности.
19. Программный модуль: виды, структура. Компоновка и загрузка программных модулей.
20. Среда выполнения. Прикладной и двоичный интерфейс программ (API, ABI).
21. Выполнение программ, среда поддержки выполнения. Системные библиотеки.
22. Системный загрузчик. Этапы загрузки. Multiboot-загрузчики.
23. Главная загрузочная запись. UEFI: архитектура, принцип работы.
24. Отладчик. Основные компоненты и принцип их работы. Антиотладочные приемы.
25. Виды отладчиков. Отладочные приемы. Отладчики и Digital Rights Management.
26. Виртуальная машина. Структура. Виды.
27. Принцип работы виртуальной машины. Технология квазивиртуализации.
28. Макропроцессор. Синтаксис макросов. Структура и принцип работы.
29. Препроцессор: назначение, структура и принцип работы.
30. GCC: архитектура, принцип работы, основные команды.
31. Средства автоматизации сборки: назначение, принцип работы, возможности.
32. Назначение toolchain. Состав и особенности. Редактирование и инспектирование объектных файлов.
33. Отладчик gdb: архитектура, принцип работы, основные команды.
34. Библиотеки подпрограмм: назначение, виды, способы использования.
35. FFI: назначение, виды, способы использования. Примеры использования на разных языках.
36. Кросс-компиляция. Способы создания кода для другой платформы. LLVM.
37. Системное программирование Linux: основные принципы, системные вызовы и функции, стандарты возможностей.
38. Системная библиотека Linux glibc: назначение, структура и возможности, способы использования.
39. Файловый ввод-вывод Linux. Основные операции с файлами. Расширенный доступ к файлам.
40. Управление файловой системой и устройствами хранения.
41. Управление процессам Linux: запуск, завершение, синхронизация. Управление планировщиком процессов.
42. Управление памятью Linux: получение информации о выделенной памяти, установка атрибутов, блокирование. Выделение памяти в объектном модуле, в стеке, в куче, отображение в страницы.
43. Управление памятью в Linux: работа с виртуальной памятью, swapon, управление памятью с помощью mmap().

44. Системные вызовы в Linux: основные категории, работа с файловыми системами, управление процессами.
45. Механизмы синхронизации процессов в Linux: семафоры, мьютексы, условные переменные.
46. Рабочие процессы в Linux: процессы, потоки, управление ими.
47. Механизмы межпроцессного взаимодействия (IPC): каналы, очереди сообщений, разделяемая память.
48. Планировщик процессов в Linux: алгоритмы планирования, приоритеты, задержки.
49. Сетевое программирование в Linux: сокеты, взаимодействие процессов через сеть.
50. Программирование драйверов устройств в Linux: основные принципы, структура драйвера, регистрация устройств.
51. Системные вызовы для работы с сетевыми интерфейсами в Linux: сокеты, протоколы, сокс-программы.
52. Структура и работа ядра Linux: основные компоненты ядра, система управления процессами.
53. Обработка прерываний в Linux: как ядро управляет прерываниями, обработка прерываний и прерывания аппаратных устройств.
54. Устройства ввода-вывода в Linux: работа с устройствами, драйверами устройств, системные вызовы для ввода-вывода.
55. Подсистемы ядра Linux: управление процессами, памятью, файловой системой.
56. Модели параллельного программирования в системном программировании.
57. Программирование с использованием потоков в Linux.
58. Безопасность в системном программировании: основы защиты памяти, защиты от переполнения буфера.
59. Методы защиты данных и защиты программных приложений.
60. Основы мониторинга и диагностики системы: инструменты и подходы.
61. Инструменты для анализа производительности программ: профилирование, сбор статистики.
62. Технология создания пользовательских утилит и библиотек в Linux.
63. Работа с протоколами низкого уровня в Linux: TCP/IP, UDP, HTTP.
64. Механизмы хранения данных в Linux: базы данных, файловые системы, управление метаданными.
65. Программирование с использованием библиотеки POSIX.
66. Управление ресурсами в системах с ограниченными вычислительными мощностями.
67. Процессы и потоки в многозадачных системах: параллельность и конкуренция.
68. Механизмы исключений и обработки ошибок в системном программировании.
69. Технология создания виртуальных машин: особенности реализации, поддержка аппаратной виртуализации.

70. Управление виртуальными машинами в Linux: создание, настройка, мониторинг.

71. Модели работы с сетями в многозадачных системах.

72. Программирование параллельных и распределенных систем.

73. Системное программирование в условиях реального времени: основы, особенности, применение.

74. Технология многопоточности в Linux: создание и управление потоками.

75. Оптимизация производительности системных программ: алгоритмы и методы.

#### **6.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

Основной целью проведения промежуточной аттестации является определение степени достижения целей по учебной дисциплине или ее разделам. Осуществляется это проверкой и оценкой уровня теоретической знаний, полученных обучающимися, умения применять их в решении практических задач, степени овладения обучающимися практическими навыками и умениями в объеме требований рабочей программы по дисциплине, а также их умение самостоятельно работать с учебной литературой.

Организация проведения промежуточной аттестации регламентирована «Положением об организации образовательного процесса в федеральном государственном автономном образовательном учреждении «Московский политехнический университет»

##### **6.4.1. Показатели оценивания компетенций на различных этапах их формирования, достижение обучающимися планируемых результатов обучения по дисциплине**

<b>Код и наименование компетенции ПК-2. Способен разрабатывать требования и проектировать программное обеспечение.</b>				
<b>Этап (уровень)</b>	<b>Критерии оценивания</b>			
	<b>неудовлетворительно</b>	<b>удовлетворительно</b>	<b>хорошо</b>	<b>отлично</b>
<b>знать</b>	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие следующих знаний: методики поиска, сбора и обработки информации; актуальные источники информации в сфере профессиональной деятельности; основные	Обучающийся демонстрирует неполное соответствие следующих знаний: методики поиска, сбора и обработки информации; актуальные источники информации в сфере профессиональной деятельности;	Обучающийся демонстрирует частичное соответствие следующих знаний: методики поиска, сбора и обработки информации; актуальные источники информации в сфере	Обучающийся демонстрирует полное соответствие следующих знаний: методики поиска, сбора и обработки информации; актуальные источники информации в сфере профессиональной

	принципы и методы системного анализа .	основные принципы и методы системного анализа.	профессиональной деятельности; основные принципы и методы системного анализа	деятельности; основные принципы и методы системного анализа.
<b>уметь</b>	Обучающийся не умеет или в недостаточной степени умеет выполнять: применять методики поиска, сбора и обработки информации; находить и осуществлять систематизацию, критический анализ и синтез информации, полученной из разных источников; применять системный подход для решения поставленных задач направления подготовки.	Обучающийся демонстрирует неполное соответствие следующих умений: применять методики поиска, сбора и обработки информации; находить и осуществлять систематизацию, критический анализ и синтез информации, полученной из разных источников; применять системный подход для решения поставленных задач направления подготовки.	Обучающийся демонстрирует частичное соответствие следующих умений: применять методики поиска, сбора и обработки информации; находить и осуществлять систематизацию, критический анализ и синтез информации, полученной из разных источников; применять системный подход для решения поставленных задач направления подготовки.	Обучающийся демонстрирует полное соответствие следующих умений: применять методики поиска, сбора и обработки информации; находить и осуществлять систематизацию, критический анализ и синтез информации, полученной из разных источников; применять системный подход для решения поставленных задач направления подготовки.
<b>владеть</b>	Обучающийся не владеет или в недостаточной степени владеет практическими навыками поиска и анализа и синтеза информации; методикой системного подхода для решения поставленных задач направления подготовки.	Обучающийся владеет в неполном объеме и проявляет недостаточность владения практическими навыками поиска и анализа и синтеза информации; методикой системного подхода для решения поставленных задач направления подготовки.	Обучающимся допускаются незначительные ошибки, неточности, затруднения, частично владеет практическими навыками поиска и анализа и синтеза информации; методикой системного подхода для решения поставленных задач направления подготовки.	Обучающийся свободно применяет полученные навыки, в полном объеме владеет практическими навыками поиска и анализа и синтеза информации; методикой системного подхода для решения поставленных задач направления подготовки.

#### 6.4.2. Методика оценивания результатов промежуточной аттестации

Показателями оценивания компетенций на этапе промежуточной аттестации по дисциплине «Системное программное обеспечение» являются результаты обучения по дисциплине.

#### Оценочный лист результатов обучения по дисциплине

Код компетенции	Знания	Умения	Навыки	Уровень сформированности компетенции на данном
-----------------	--------	--------	--------	--

				этапе / оценка
ПК- 2. Способен разрабатывать требования и проектировать программное обеспечение	на уровне знаний: знать принципы и алгоритмы работы системного программного обеспечения; знать принципы работы с аппаратным обеспечением на системном уровне; штатные режимы работы программно- аппаратных комплексов; знать назначение системного программного обеспечения в составе компьютерны х систем; знать способы настройки сетевой операционной системы.	на уровне умений: уметь применять системное программное обеспечение для формирования компьютерных систем; уметь применять возможности системного программного обеспечения для настройки и наладки аппаратного обеспечения; уметь определять нарушения и аномалии в работе программно- аппаратных комплексов.	на уровне навыков: системными утилитами и инструментами для настройки и обслуживания компьютерной системы; инструментами мониторинга работы программно- аппаратных комплексов..	
Оценка по дисциплине (среднее арифметическое)				

Оценка «отлично» выставляется, если среднее арифметическое находится в интервале от 4,5 до 5,0.

Оценка «хорошо» выставляется, если среднее арифметическое находится в интервале от 3,5 до 4,4.

Оценка «удовлетворительно» выставляется, если среднее арифметическое находится в интервале от 2,5 до 3,4.

Оценка «неудовлетворительно» выставляется, если среднее арифметическое находится в интервале от 0 до 2,4.

Промежуточная аттестация обучающихся в форме экзамена проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по дисциплине «Системное программное обеспечение», при этом

учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине проводится преподавателем, ведущим занятия по дисциплине методом экспертной оценки. По итогам промежуточной аттестации по дисциплине выставляется оценка «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

Шкала оценивания	Описание
Отлично	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Хорошо	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует неполное, правильное соответствие знаний, умений, навыков приведенным в таблицах показателей, либо если при этом были допущены 2-3 несущественные ошибки.
Удовлетворительно	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, в котором освещена основная, наиболее важная часть материала, но при этом допущена одна значительная ошибка или неточность.
Неудовлетворительно	Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

## 7. Электронная информационно-образовательная среда

Каждый обучающийся в течение всего периода обучения обеспечивается индивидуальным неограниченным доступом к электронной информационно-образовательной среде Чебоксарского института (филиала) Московского политехнического университета из любой точки, в которой имеется доступ к информационно-телекоммуникационной сети «Интернет» (далее – сеть «Интернет»), как на территории филиала, так и вне ее.

Электронная информационно-образовательная среда – совокупность информационных и телекоммуникационных технологий, соответствующих технологических средств, обеспечивающих освоение обучающимися

образовательных программ в полном объеме независимо от места нахождения обучающихся.

Электронная информационно-образовательная среда обеспечивает:

а) доступ к учебным планам, рабочим программам дисциплин (модулей), практик, электронным учебным изданиям и электронным образовательным ресурсам, указанным в рабочих программах дисциплин (модулей), практик;

б) формирование электронного портфолио обучающегося, в том числе сохранение его работ и оценок за эти работы;

в) фиксацию хода образовательного процесса, результатов промежуточной аттестации и результатов освоения программы бакалавриата;

г) проведение учебных занятий, процедур оценки результатов обучения, реализация которых предусмотрена с применением электронного обучения, дистанционных образовательных технологий;

д) взаимодействие между участниками образовательного процесса, в том числе синхронное и (или) асинхронное взаимодействия посредством сети "Интернет".

Функционирование электронной информационно-образовательной среды обеспечивается соответствующими средствами информационно-коммуникационных технологий и квалификацией работников, ее использующих и поддерживающих.

Функционирование электронной информационно-образовательной среды соответствует законодательству Российской Федерации.

Основными составляющими ЭИОС филиала являются:

а) сайт института в сети Интернет, расположенный по адресу [www.polytech21.ru](http://www.polytech21.ru), <https://chebpolytech.ru/> который обеспечивает:

- доступ обучающихся к учебным планам, рабочим программам дисциплин, практик, к изданиям электронных библиотечных систем, электронным информационным и образовательным ресурсам, указанных в рабочих программах (разделы сайта «Сведения об образовательной организации»);

- информирование обучающихся обо всех изменениях учебного процесса (новостная лента сайта, лента анонсов);

- взаимодействие между участниками образовательного процесса (подразделы сайта «Задать вопрос директору»);

б) официальные электронные адреса подразделений и сотрудников института с Яндекс-доменом [@polytech21.ru](mailto:@polytech21.ru) (список контактных данных подразделений Филиала размещен на официальном сайте Филиала в разделе «Контакты», списки контактных официальных электронных данных преподавателей размещены в подразделах «Кафедры») обеспечивают взаимодействие между участниками образовательного процесса;

в) личный кабинет обучающегося (портфолио) (вход в личный кабинет размещен на официальном сайте Филиала в разделе «Студенту» подразделе «Электронная информационно-образовательная среда») включает в себя портфолио студента, электронные ведомости, рейтинг студентов и обеспечивает:

- фиксацию хода образовательного процесса, результатов промежуточной аттестации и результатов освоения образовательных программ обучающимися,  
- формирование электронного портфолио обучающегося, в том числе с сохранение работ обучающегося, рецензий и оценок на эти работы,

г) электронные библиотеки, включающие электронные каталоги, полнотекстовые документы и обеспечивающие доступ к учебно-методическим материалам, выпускным квалификационным работам и т.д.:

Чебоксарского института (филиала) - «ИРБИС»

д) электронно-библиотечные системы (ЭБС), включающие электронный каталог и полнотекстовые документы:

- ЭБС «ЛАНЬ» -<https://e.lanbook.com/>

- Образовательная платформа Юрайт - <https://urait.ru>

- IPR SMART -<https://www.iprbookshop.ru/>

е) платформа цифрового образования Политеха -<https://lms.mospolytech.ru/>

ж) система «Антиплагиат» -<https://www.antiplagiat.ru/>

з) система электронного документооборота DIRECTUM Standard — обеспечивает документооборот между Филиалом и Университетом;

и) система «1С Управление ВУЗом Электронный деканат» (Московский политехнический университет) обеспечивает фиксацию хода образовательного процесса, результатов промежуточной аттестации и результатов освоения образовательных программ обучающимися;

к) система «POLYTECH systems» обеспечивает информационное, документальное автоматизированное сопровождение образовательного процесса;

л) система «Абитуриент» обеспечивает документальное автоматизированное сопровождение работы приемной комиссии.

## **8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины**

### Основная литература:

1. Волк, В. К. Информатика : учебник для вузов / В. К. Волк. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 226 с. — (Высшее образование). — ISBN 978-5-534-18427-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/567713>.

2. Новожилов, О. П. Информатика в 2 ч. Часть 1 : учебник для вузов / О. П. Новожилов. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 320 с. — (Высшее образование). — ISBN 978-5-534-09964-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/564565>.

3. Новожилов, О. П. Информатика в 2 ч. Часть 2 : учебник для вузов / О. П. Новожилов. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 302 с. — (Высшее образование). — ISBN 978-5-534-09966-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/564566>.

4. Информатика : учебник для вузов — 4-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 752 с. — (Высшее образование). — ISBN 978-5-534-20227-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/568691>.

5. Информатика : учебник для вузов — 4-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 752 с. — (Высшее образование). — ISBN 978-5-534-20227-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/568691>.

6. Торадзе, Д. Л. Информатика : учебник для вузов / Д. Л. Торадзе. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 158 с. — (Высшее образование). — ISBN 978-5-534-18725-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/567749>.

#### Дополнительная литература:

1. Гаврилов, М. В. Информатика и информационные технологии : учебник для вузов / М. В. Гаврилов, В. А. Климов. — 6-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 319 с. — (Высшее образование). — ISBN 978-5-534-20354-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/559723>.

2. Зыков, С. В. Программирование : учебник и практикум для вузов / С. В. Зыков. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 285 с. — (Высшее образование). — ISBN 978-5-534-16031-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/560815>.

3. Мойзес, О. Е. Информатика. Углубленный курс : учебник для вузов / О. Е. Мойзес, Е. А. Кузьменко. — Москва : Издательство Юрайт, 2025. — 150 с. — (Высшее образование). — ISBN 978-5-534-17155-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/561368>.

#### Периодика:

1. Известия Тульского государственного университета. Технические науки: Научный рецензируемый журнал. <https://tidings.tsu.tula.ru/tidings/index.php?id=technical&lang=ru&year=1>.  
- Текст : электронный.

2. Научный периодический журнал «Вестник Южно-Уральского государственного университета. Серия «Вычислительная математика и информатика» : Научный рецензируемый журнал. <https://vestnik.susu.ru/cmi> - Текст : электронный.

3. Научный периодический журнал «Вестник Южно-Уральского государственного университета. Серия «Компьютерные технологии, управление, радиоэлектроника»: Научный рецензируемый журнал. <https://vestnik.susu.ru/ctcr> - Текст : электронный.

## 9. Профессиональные базы данных и информационно-справочные системы

Профессиональная база данных и информационно-справочные системы	Информация о праве собственности (реквизиты договора)
<p>Университетская информационная система РОССИЯ <a href="https://uisrussia.msu.ru/">https://uisrussia.msu.ru/</a></p>	<p>Тематическая электронная библиотека и база для прикладных исследований в области экономики, управления, социологии, лингвистики, философии, филологии, международных отношений, права. свободный доступ</p>
<p>научная электронная библиотека Elibrary <a href="http://elibrary.ru/">http://elibrary.ru/</a></p>	<p>Научная электронная библиотека eLIBRARY.RU - это крупнейший российский информационно-аналитический портал в области науки, технологии, медицины и образования, содержащий рефераты и полные тексты более 26 млн научных статей и публикаций, в том числе электронные версии более 5600 российских научно-технических журналов, из которых более 4800 журналов в открытом доступе свободный доступ</p>
<p>сайт Института научной информации по общественным наукам РАН. <a href="http://www.inion.ru">http://www.inion.ru</a></p>	<p>Библиографические базы данных ИНИОН РАН по социальным и гуманитарным наукам ведутся с начала 1980-х годов. Общий объем массивов составляет более 3 млн. 500 тыс. записей (данные на 1 января 2012 г.). Ежегодный прирост — около 100 тыс. записей. В базы данных включаются аннотированные описания книг и статей из журналов и сборников на 140 языках, поступивших в Фундаментальную библиотеку ИНИОН РАН. Описания статей и книг в базах данных снабжены шифром хранения и ссылками на полные тексты источников из Научной электронной библиотеки.</p>
<p>Федеральный портал «Российское образование» [Электронный ресурс] – <a href="http://www.edu.ru">http://www.edu.ru</a></p>	<p>Федеральный портал «Российское образование» – уникальный интернет-ресурс в сфере образования и науки. Ежедневно публикует самые актуальные новости, анонсы событий, информационные материалы для широкого круга читателей. Ежедневно на портале размещаются эксклюзивные материалы, интервью с ведущими специалистами – педагогами, психологами, учеными, репортажи и аналитические статьи. Читатели получают доступ к нормативно-правовой базе сферы образования, они могут пользоваться самыми различными полезными сервисами – такими, как онлайн-тестирование, опросы по актуальным темам и т.д.</p>
<p>Информационные технологии – периодическое научно-техническое издание в области информационных технологий, автоматизированных систем и использования информатики в различных приложениях</p>	<p>Издательство выпускает теоретические и прикладные научно-технические журналы, обеспечивающие научной, производственной, обзорно-аналитической и образовательной информацией руководящих работников и специалистов промышленных предприятий, научных академических и отраслевых организаций, а также учебных заведений в области приоритетных направлений развития науки и технологий.</p>

<a href="http://novtex.ru">novtex.ru</a>	
Ассоциация инженерного образования России <a href="http://www.ac-raee.ru/">http://www.ac-raee.ru/</a>	Совершенствование образования и инженерной деятельности во всех их проявлениях, относящихся к учебному, научному и технологическому направлениям, включая процессы преподавания, консультирования, исследования, разработки инженерных решений, оказания широкого спектра образовательных услуг, обеспечения связей с общественностью, производством, наукой и интеграции в международное научно-образовательное пространство. свободный доступ

### 10. Программное обеспечение (лицензионное и свободно распространяемое), используемое при осуществлении образовательного процесса

Аудитория	Программное обеспечение	Информация о праве собственности (реквизиты договора, номер лицензии и т.д.)
<b>№ 2196</b> Учебная аудитория для проведения учебных занятий всех видов, предусмотренных программой среднего профессионального образования/бакалавриата/специалитета/ магистратуры, оснащенная оборудованием и техническими средствами обучения, состав которых определяется в рабочих программах дисциплин (модулей) <u>Кабинет систем управления ООО «НПО «Каскад-ГРУП»</u>	Windows 7 OLPNLAcadm	договор №Д03 от 30.05.2012) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)
	Kaspersky Endpoint Security для бизнеса – Расширенный Russian Edition. 150-249 Node 2 year Educational Renewal License	Сублицензионный договор №977_1049.ЕП/25 от 10.12.2025
	Yandex браузер	Свободное распространяемое программное обеспечение (бессрочная лицензия)
	МТС Линк	Договор №2/2026 (091_168.ЕП/26) от 27.03.2026
	Microsoft Office Standard 2007(Microsoft DreamSpark Premium Electronic Software Delivery Academic (Microsoft Open License)	номер лицензии-42661846 от 30.08.2007) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)
<b>№ 2066</b> Учебная аудитория для проведения учебных занятий всех видов, предусмотренных программой среднего профессионального образования/бакалавриата/специалитета/ магистратуры, оснащенная оборудованием и техническими средствами обучения, состав которых определяется в рабочих программах дисциплин (модулей) <u>Лаборатория «Программирования и баз</u>	Kaspersky Endpoint Security для бизнеса – Расширенный Russian Edition. 150-249 Node 2 year Educational Renewal License	Сублицензионный договор №977_1049.ЕП/25 от 10.12.2025
	Windows 7 OLPNLAcadm Windows Server 2012	договор №Д03 от 30.05.2012) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)
	MS Windows 10 Pro	договор № 392_469.223.3К/19 от 17.12.19 (бессрочная лицензия)
	Microsoft Office Standard 2019(Microsoft DreamSpark Premium Electronic Software Delivery Academic(Microsoft Open License)	номер лицензии-42661846 от 30.08.2007) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)
	Eclipse IDE for Java EE Developers, NET Framework, JDK 8, Microsoft SQL Server Express Edition, Microsoft Visio Professional,	свободно распространяемое программное обеспечение (бессрочная лицензия)

<u>данных»</u> <u>Лаборатория</u> <u>информационных технологий</u>	Microsoft Visual Studio, MySQL Installer for Windows, NetBeans, SQL Server Management Studio, Microsoft SQL Server Java Connector, Android Studio, IntelliJ IDEA.	
	КОМПАС-3D v20 и v21	Сублицензионный договор № Нп-22-00044 от 21.03.2022 (бессрочная лицензия)
	MathCADv.15	Сублиц.договор №39331/МОС2286 от 6.05.2013) номер лицензии-42661846 от 30.08.2007) (бессрочная лицензия)
	SimInTech	Отечественное программное обеспечение
	AdobeReader	свободно распространяемое программное обеспечение (бессрочная лицензия)
	AdobeFlashPlayer	свободно распространяемое программное обеспечение (бессрочная лицензия)
	Microsoft Visual Studio 2019	свободно распространяемое программное обеспечение (бессрочная лицензия)
	Python 3.7	свободно распространяемое программное обеспечение (бессрочная лицензия)
	PascalABC	свободно распространяемое программное обеспечение (бессрочная лицензия)
AIMP	отечественное свободно распространяемое программное обеспечение (бессрочная лицензия)	
№ 1126 Помещение для самостоятельной работы обучающихся	Kaspersky Endpoint Security для бизнеса – Расширенный Russian Edition. 150-249 Node 2 year Educational Renewal License	Сублицензионный договор №977_1049.ЕП/25 от 10.12.2025
	Windows 7 OLPNLAcDmc	договор №Д03 от 30.05.2012) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)
	AdobeReader	свободно распространяемое программное обеспечение (бессрочная лицензия)
	Гарант-справочно-правовая система	Договор №С-002-2025 от 09.01.2025
	Yandex браузер	свободно распространяемое программное обеспечение (бессрочная лицензия)
	Microsoft Office Standard 2007(Microsoft DreamSpark Premium Electronic Software Delivery Academic (Microsoft Open License	номер лицензии-42661846 от 30.08.2007) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)
	МТС Линк	Договор №2/2026 (091_168.ЕП/26) от 27.03.2026
AIMP	отечественное свободно распространяемое программное	

## 11. Материально-техническое обеспечение дисциплины

Тип и номер помещения	Перечень основного оборудования и технических средств обучения
Учебная аудитория для проведения учебных занятий всех видов, предусмотренных программой среднего профессионального образования/бакалавриата/ специалитета/ магистратуры, оснащенная оборудованием и техническими средствами обучения, состав которых определяется в рабочих программах дисциплин (модулей) Кабинет систем управления ООО «НПО «Каскад-ГРУП» № 2196 (г. Чебоксары, ул. К.Маркса, 60)	<u>Оборудование:</u> комплект мебели для учебного процесса; доска учебная; стенды <u>Технические средства обучения:</u> мультимедийное оборудование (проектор, экран)
Учебная аудитория для проведения учебных занятий всех видов, предусмотренных программой среднего профессионального образования/бакалавриата/ специалитета/ магистратуры, оснащенная оборудованием и техническими средствами обучения, состав которых определяется в рабочих программах дисциплин (модулей) Лаборатория информационных технологий № 2066 (г. Чебоксары, ул. К.Маркса, 60)	<u>Оборудование:</u> комплект мебели для учебного процесса; доска учебная; автоматизированные рабочие места, автоматизированное рабочее место преподавателя, проектор и экран; маркерная доска; <u>Технические средства обучения:</u> компьютерная техника (процессор Core i3, оперативная память объемом не менее 8 Гб); сервер в лаборатории (8-ядерный процессор с частотой не менее 3 ГГц, оперативная память объемом не менее 16 Гб, жесткие диски общим объемом не менее 1 Тб; мультимедийное оборудование (телевизор)
Помещение для самостоятельной работы обучающихся № 1126 (г. Чебоксары, ул. К.Маркса, 60)	<u>Оборудование:</u> комплект мебели для учебного процесса; <u>Технические средства обучения:</u> компьютерная техника с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Филиала

## 12. Методические указания для обучающегося по освоению дисциплины

### *Методические указания для занятий лекционного типа*

В ходе лекционных занятий обучающемуся необходимо вести конспектирование учебного материала, обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации.

Необходимо задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций. Целесообразно дорабатывать свой конспект лекции, делая в нем соответствующие записи из основной и дополнительной литературы, рекомендованной преподавателем и предусмотренной учебной программой дисциплины.

### *Методические указания для занятий лабораторного типа.*

Выполнению лабораторных работ предшествует проверка знаний студентов – их теоретической готовности к выполнению задания. Проверка

знаний проводится в форме, которую определяет преподаватель дисциплины (тестирование, опрос).

При проведении лабораторных занятий выделяют следующие разделы:

- общие положения (перечень лабораторных или практических занятий);
- ообщие требования к выполнению работ;
- инструкция по каждой работе;
- справочные материалы и т. д.

Лабораторные занятия позволяют развивать у обучающегося творческое теоретическое мышление, умение самостоятельно изучать литературу, анализировать практику; учат четко формулировать мысль, вести дискуссию, то есть имеют исключительно важное значение в развитии самостоятельного мышления.

Готовясь к докладу или выступлению в рамках интерактивной формы, при необходимости, следует обратиться за помощью к преподавателю.

#### ***Методические указания к самостоятельной работе.***

Самостоятельная работа обучающегося является основным средством овладения учебным материалом во время, свободное от обязательных учебных занятий. Самостоятельная работа обучающегося над усвоением учебного материала по учебной дисциплине может выполняться в библиотеке университета, учебных кабинетах, компьютерных классах, а также в домашних условиях. Содержание и количество самостоятельной работы обучающегося определяется учебной программой дисциплины, методическими материалами, практическими заданиями и указаниями преподавателя.

#### ***Самостоятельная работа в аудиторное время может включать:***

- 1) конспектирование (составление тезисов) лекций;
- 2) выполнение контрольных работ;
- 3) решение задач;
- 4) работу со справочной и методической литературой;
- 5) работу с нормативными правовыми актами;
- 6) выступления с докладами, сообщениями на семинарских занятиях;
- 7) защиту выполненных работ;
- 8) участие в оперативном (текущем) опросе по отдельным темам изучаемой дисциплины;
- 9) участие в беседах, деловых (ролевых) играх, дискуссиях, круглых столах, конференциях;
- 10) участие в тестировании и др.

#### ***Самостоятельная работа во внеаудиторное время может состоять***

***из:***

- 1) повторения лекционного материала;
- 2) подготовки к практическим занятиям;
- 3) изучения учебной и научной литературы;
- 4) изучения нормативных правовых актов (в т.ч. в электронных базах данных);
- 5) решения задач, и иных практических заданий;

- 6) подготовки к контрольным работам, тестированию и т.д.;
- 7) подготовки к практическим занятиям устных докладов (сообщений);
- 8) выполнения курсовых работ, предусмотренных учебным планом;
- 9) выполнения выпускных квалификационных работ и др.;
- 10) выделения наиболее сложных и проблемных вопросов по изучаемой теме, получение разъяснений и рекомендаций по данным вопросам с преподавателями на консультациях;
- 11) проведения самоконтроля путем ответов на вопросы текущего контроля знаний, решения представленных в учебно-методических материалах кафедры задач, тестов, докладов;
- 12) текущий контроль осуществляется в форме устных, тестовых опросов, докладов;

В случае пропусков занятий, наличия индивидуального графика обучения и для закрепления практических навыков студентам могут быть выданы типовые индивидуальные задания, которые должны быть сданы в установленный преподавателем срок.

### **13. Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья**

Обучение по данной дисциплине инвалидов и лиц с ограниченными возможностями здоровья (далее ОВЗ) осуществляется преподавателем с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.

Для студентов с нарушениями опорно-двигательной функции и с ОВЗ по слуху предусматривается сопровождение лекций и практических занятий мультимедийными средствами, раздаточным материалом.

Для студентов с ОВЗ по зрению предусматривается применение технических средств усиления остаточного зрения, а также предусмотрена возможность разработки аудиоматериалов.

По данной дисциплине обучение инвалидов и лиц с ограниченными возможностями здоровья может осуществляться как в аудитории, так и с использованием электронной информационно-образовательной среды, образовательного портала и электронной почты.

ЛИСТ ДОПОЛНЕНИЙ И ИЗМЕНЕНИЙ  
рабочей программы дисциплины

Рабочая программа дисциплины рассмотрена, обсуждена и одобрена для исполнения в 202\_\_-202\_\_ учебном году на заседании кафедры, протокол № \_\_\_\_ от « » 202 г.

Внесены дополнения и изменения \_\_\_\_\_

---

---

---

Рабочая программа дисциплины рассмотрена, обсуждена и одобрена для исполнения в 202\_\_-202\_\_ учебном году на заседании кафедры, протокол № \_\_\_\_ от « » 202 г.

Внесены дополнения и изменения \_\_\_\_\_

---

---

---

Рабочая программа дисциплины рассмотрена, обсуждена и одобрена для исполнения в 202\_\_-202\_\_ учебном году на заседании кафедры, протокол № \_\_\_\_ от « » 202 г.

Внесены дополнения и изменения \_\_\_\_\_

---

---

---

Рабочая программа дисциплины рассмотрена, обсуждена и одобрена для исполнения в 202\_\_-202\_\_ учебном году на заседании кафедры, протокол № \_\_\_\_ от « » 202 г.

Внесены дополнения и изменения \_\_\_\_\_

---

---

---