

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
Факультет компьютерных технологий и прикладной математики

УТВЕРЖДАЮ:

Проректор по учебной работе,  
качеству образования – первый  
проректор

\_\_\_\_\_ Хагуров Т.А.

*подпись*

« 29 » августа 2025 г.

## **РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

Б1. О.31 Операционные системы

Направление подготовки 01.03.02 Прикладная математика и информатика

Профиль Современные методы машинного обучения и компьютерного зрения

Форма обучения очная

Квалификация бакалавр

Краснодар 2025

Рабочая программа дисциплины «Операционные системы» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 01.03.02 Фундаментальная информатика и информационные технологии.

Программу составила:

Е.Е. Полупанова, доцент кафедры вычислительных технологий, кандидат технических наук, доцент



Рабочая программа дисциплины «Операционные системы» утверждена на заседании центра искусственного интеллекта протокол № 1 «28» августа 2025 г.

Руководитель центра ИИ Коваленко А.В.

фамилия, инициалы

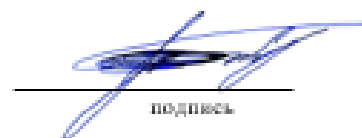


подпись

Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики протокол № 1 «28» августа 2025 г.

Председатель УМК факультета Коваленко А.В.

фамилия, инициалы



подпись

Рецензенты:

Мостовой Евгений Викторович, генеральный директор ООО «Портал-Юг»,  
e-mail: mostovoy@portal-yug.ru

Луценко Евгений Вениаминович, доктор экономических наук, кандидат технических наук, профессор кафедры компьютерных технологий и систем Федерального государственного бюджетное образовательное учреждение высшего образования «Кубанский государственный аграрный университет имени И.Т. Трубилина», e-mail: prof.lutsenko@gmail.com

# **1 Цели и задачи изучения дисциплины (модуля)**

## **1.1 Цель освоения дисциплины**

**Цель** дисциплины «Операционные системы» — ознакомление студентов с организацией современных компьютерных систем, с процессами обработки информации на всех уровнях компьютерных архитектур, включая: цифровой, логический уровень, системы команд, уровень архитектурной поддержки механизмов Unix-подобных операционных систем на примере ОС Astra Linux.

## **1.2 Задачи дисциплины**

– изучение концепций построения операционных систем, их основных характеристик и областей применения, типовых методов организации и свойств основных компонентов ОС;

– знакомство с взаимосвязями архитектурных особенностей аппаратуры ЭВМ и компонентов системного программного обеспечения;

– изучение методов организации файловых систем, подходов к обеспечению безопасности функционирования ОС и взаимодействия процессов.

– изучение механизмов управления оперативной памятью и внешними устройствами ОС.

Отбор материала основывается на необходимости ознакомить студентов со следующей современной научной информацией:

– о концепциях построения операционных систем и системного программного обеспечения;

– о способах синхронизации потоков и процессов;

– об обеспечении безопасности функционирования операционных систем.

– о способах управления оперативной памятью и внешними устройствами операционных систем.

Научной основой для построения программы данной дисциплины является теоретико-прагматический подход в обучении.

## **1.3 Место дисциплины (модуля) в структуре образовательной программы**

Дисциплина «Операционные системы» относится к дисциплинам базовой части учебного плана, код Б1.О.31.

## **1.4 Профессиональные роли в структуре образовательной программы**

Роль 1: Data Engineer (Инженер по данным)

Задачи:

- Проектирование и построение ETL-процессов
- Создание и оптимизация хранилищ данных
- Обеспечение качества и доступности данных
- Настройка инфраструктуры для обработки больших данных
- Интеграция разрозненных источников данных
- Работа с данными в области природопользования, медицины, связи и телекоммуникаций

Роль 2: ML Engineer (Инженер МО)

Задачи:

- Реализация ML-моделей в продуктивных системах

- Оптимизация производительности и масштабирование моделей
- Разработка ML-пайплайнов и автоматизация процессов
- Мониторинг качества моделей в продуктиве
- Интеграция ML-решений с бизнес-приложениями

Роль 3: MLOps (Специалист по эксплуатации ИИ)

Задачи:

- Автоматизация процессов обучения и развертывания моделей
- Мониторинг производительности ML-систем
- Управление версиями моделей и данных
- Обеспечение CI/CD для ML-проектов
- Оптимизация вычислительных ресурсов

**1.5 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы.**

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих компетенций:

Код и наименование индикатора*	Результаты обучения по дисциплине (знает, умеет, владеет (навыки и/или опыт деятельности))
<b>ОПК-4. Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности</b>	
ОПК-4.1 Аргументировано применяет современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения	Знает современные информационные технологии, используемые при создании программных продуктов необходимых для стабильного функционирования ОС
ОПК-4.2 Ориентируется в современных положениях и концепциях прикладного и системного программного обеспечения, архитектуры компьютеров и сетей (в том числе и глобальных), технологии создания и сопровождения программных продуктов и программных комплексов	Умеет ориентироваться в современных положениях и концепциях прикладного и системного программного обеспечения, архитектуре компьютеров
<b>ОПК-5. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения</b>	
ОПК-5.1 Аргументировано применяет методы проектирования, разработки и реализации программных продуктов и программных комплексов в различных областях человеческой деятельности	Знает методы проектирования, разработки и реализации программных продуктов и программных комплексов для решения прикладных задач в области ОС
ОПК-5.2 Использует инструментальные, программные и аппаратные средства измерений для оценки качества программного обеспечения	Умеет использовать инструментальные, программные и аппаратные средства измерений для оценки качества ПО
<b>РЛ-3 Способен применять языки программирования C/C++ для решения задач в области ИИ</b>	
РЛ-3.1 Разрабатывает и отлаживает эффективные многопоточные решения на C++, тестирует, испытывает и оценивает качество таких решений	Знает основы синтаксиса языка. Знает общие принципы параллельных вычислений и понимает проблемы, возникающие при распараллеливании алгоритмов.

Код и наименование индикатора*	Результаты обучения по дисциплине ( <i>знает, умеет, владеет (навыки и/или опыт деятельности)</i> )
	<p>Умеет проводить распараллеливание простого алгоритма с применением OpenMP, стандартных библиотек C/C++ или др.</p> <p>Умеет решать проблемы одновременного доступа к данным из нескольких потоков, грамотно применяет атомарные операции и механизм блокировок.</p> <p>Владеет методами оценки производительности, профилирования кода и устранения найденных узких мест.</p>
<p>PL-3.2 Разрабатывает и отлаживает системы ИИ на C++ под конкретные аппаратные платформы с ограничениями по вычислительной мощности, в том числе для встроенных систем</p>	<p>Знает основы синтаксиса языка.</p> <p>Умеет писать небольшие программы для обработки данных.</p> <p>Понимает возможности и ограничения встроенных систем.</p> <p>Владеет навыками использования библиотек, соответствующих решаемой задаче.</p>
<p><b>SS-2 Способен осуществлять свою трудовую деятельность с учётом необходимости эффективной коммуникации и взаимодействия в рамках коллективной проектной работы в сфере ИИ</b></p>	
<p>SS-2.1 Учитывает профессиональные и ролевые особенности коллег при совместной разработке технических решений и представлении результатов</p>	<p>Участвует в подготовке презентации по своей части (например, рассказывает про архитектуру ИИ-системы или метрики).</p> <p>Принимает предложенное распределение ролей без активного участия в общей координации.</p> <p>Ориентируется в структуре общего результата проекта.</p>
<p><b>DL-3 Способен применять и (или) разрабатывать алгоритмы, методы и технологии компьютерного зрения</b></p>	
<p>DL-3.2 Определяет стек технологий, методов и алгоритмов для построения продуктов с компьютерным зрением (системы видеоаналитики, поисковые системы по изображениям и т.д.)</p>	<p>Понимает принципы представления изображений и кодирования цвета; применяет фильтрацию изображений, включая частотную; Владеет аппаратом математической морфологии; Понимает базовые задачи анализа изображений - классификация, детекция, сегментация. Применяет известные архитектуры нейронных сетей (CNN) в решении простых задач распознавания изображений</p>

Результаты обучения по дисциплине достигаются в рамках осуществления всех видов контактной и самостоятельной работы обучающихся в соответствии с утвержденным учебным планом.

Индикаторы достижения компетенций считаются сформированными при достижении соответствующих им результатов обучения.

## 2. Структура и содержание дисциплины

### 2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 5 зач.ед. (180 часов), их распределение по видам работ представлено в таблице.

Вид учебной работы	Всего часов	Очная форма обучения				
		3 семестр (часы)	—	—	—	
<b>Контактная работа, в том числе:</b>	<b>113,3</b>	<b>113,3</b>				
<b>Аудиторные занятия (всего):</b>	<b>108,3</b>	<b>108,3</b>				
Занятия лекционного типа	50	50	—	—	—	
Лабораторные занятия	52	52	—	—	—	
Занятия семинарского типа (семинары, практические занятия)	—	—	—	—	—	
	—	—	—	—	—	
<b>Иная контактная работа:</b>						
Курсовая работа/проект (КР/КП) (подготовка)	5	5	—	—	—	
Контроль самостоятельной работы (КСР)	6	6	—	—	—	
Промежуточная аттестация (ИКР)	0,3	0,3	—	—	—	
<b>Самостоятельная работа, в том числе:</b>	<b>31</b>	<b>31</b>				
Проработка учебного (теоретического) материала	31	31	—	—	—	
Выполнение индивидуальных заданий (подготовка сообщений, презентаций)			—	—	—	
Реферат	—	—	—	—	—	
Подготовка к текущему контролю			—	—	—	
<b>Контроль:</b>	<b>экзамен</b>	<b>экзамен</b>				
Подготовка к экзамену	35,7	35,7	—	—	—	
<b>Общая трудоёмкость</b>	<b>час.</b>	<b>180</b>	<b>180</b>	—	—	—
	<b>в том числе контактная работа</b>	<b>113,3</b>	<b>113,3</b>	—	—	—
	<b>зач. ед.</b>	<b>5</b>	<b>5</b>	—	—	—

### 2.1 Структура дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.

Разделы (темы) дисциплины, изучаемые в 3 семестре

№	Наименование разделов (тем)	Всего	Количество часов			
			Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	
1	2	3	4	5	6	7
1	Введение	7	2		4	1
2	Основы архитектуры вычислительной системы	14	6		6	2
3	Основы компьютерной архитектуры	16	6		6	4
4	Основы архитектуры операционных систем	16	6		6	4
5	Управление процессами	16	6		6	4
6	Реализация межпроцессного	16	6		6	4

	взаимодействия					
7	Файловые системы	16	6		6	4
8	Управление оперативной памятью	16	6		6	4
9	Управление внешними устройствами	16	6		6	4
<b>ИТОГО по разделам дисциплины</b>		<b>133</b>	<b>50</b>	<b>–</b>	<b>52</b>	<b>31</b>
	Курсовая работа/проект (КР/КП)	<b>5</b>				
	Контроль самостоятельной работы (КСР)	<b>6</b>				
	Промежуточная аттестация (ИКР)	<b>0,3</b>				
	Подготовка к текущему контролю	<b>35,7</b>				
Общая трудоемкость по дисциплине		<b>180</b>				

Примечание: Л – лекции, ПЗ – практические занятия/семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

## 2.3 Содержание разделов (тем) дисциплины

### 2.3.1 Занятия лекционного типа

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
1	Введение	Развитие и становление аппаратных возможностей компьютеров. Понятие операционной системы. История появления и развития операционных систем.	Л
2	Основы архитектуры вычислительной системы	Структура ВС Аппаратный уровень ВС Управление физическими ресурсами ВС Управление логическими/виртуальными ресурсами. Системы программирования. Прикладные системы.	Л
3	Основы компьютерной архитектуры	Структура, основные компоненты. Оперативное запоминающее устройство. Центральный процессор. Внешние устройства. Иерархия памяти. Аппаратная поддержка операционной системы и систем программирования. Многомашинные, многопроцессорные ассоциации. Терминальные комплексы (ТК).	Л
4	Основы архитектуры операционных систем	Структура ОС. Логические функции ОС. Типы операционных систем	Л
5	Управление процессами	Основные концепции. Реализация полновесных процессов в ОС Unix. Реализация легковесных процессов Unix (нити). Планирование. Взаимодействие процессов.	Л
6	Реализация межпроцессного взаимодействия	Базовые средства реализации взаимодействия процессов в ОС Unix. Система межпроцессного взаимодействия IPC (Inter-Process Communication). Сокеты — унифицированный интерфейс программирования распределенных систем. Взаимодействие нитей в ОС Unix.	Л
7	Файловые системы	Структурная организация файлов. Атрибуты файлов. Модели реализации файлов и каталогов. Надежность файловой системы. Проверка	Л

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		целостности файловой системы. Примеры реализаций файловых систем.	
8	Управление оперативной памятью	Одиночное непрерывное распределение. Распределение перемещаемыми разделами. Распределение перемещаемыми разделами. Страничное распределение. Сегментное распределение. Сегментно-страничное распределение.	Л
9	Управление внешними устройствами	Архитектура организации управления внешними устройствами. Программное управление внешними устройствами. Планирование дисковых обменов. RAID-системы. Уровни RAID. Файлы устройств, драйверы. Системные таблицы драйверов устройств. Ситуации, вызывающие обращение к функциям драйвера. Включение, удаление драйверов из системы. Буферизация при блок-ориентированном обмене. Борьба со сбоями.	Л

Примечание: Л – лекции, ПЗ – практические занятия/семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

### 2.3.2 Занятия семинарского типа (практические / семинарские занятия/ лабораторные работы)

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
1.	Введение	Знакомство с операционной системой Linux	ЛР, Т
2.	Основы архитектуры вычислительной системы	Работа с терминалом и оболочкой Bash.	ЛР, Т
3.		Работа со сценариями Bash.	ЛР, Т
4.	Основы компьютерной архитектуры	Загрузка системы и управление службами systemd	ЛР, Т
5.		Управление пользователями и группами.	ЛР, Т
6.		Работа с правами доступа, ACL	ЛР, Т
7.		Работа с профилями пользователей.	ЛР, Т
8.	Основы архитектуры операционных систем	Ядро ОС.	ЛР, Т
9.		Управление ПО.	ЛР, Т
10.		Базовый комплекс средств защиты информации.	ЛР, Т
11.	Управление процессами	Процессы	ЛР, Т
	Реализация межпроцессного взаимодействия		
12.	Файловые системы	Файловая система в Linux.	ЛР, Т
13.		Работа с файлами и каталогами. FHS	ЛР, Т

14.		Работа с блочными устройствами и LVM	ЛР, Т
15.	Управление внешними устройствами	Управление устройствами и модулями.	ЛР, Т, КП

*Примечание: ЛР – отчет/защита лабораторной работы, КП - выполнение курсового проекта, КР - курсовой работы, РГЗ - расчетно-графического задания, Р - написание реферата, Э - эссе, К - коллоквиум, Т – тестирование, РЗ – решение задач.*

### **2.3.3 Примерная тематика курсовых работ (проектов)**

#### **Обработка изображений**

1. Модификация драйвера USB-камеры для применения простых аппаратных корректировок (яркость, контрастность) через интерфейс V4L2 в Astra Linux.

2. Добавление поддержки формата изображения MJPG в драйвер камеры для снижения нагрузки на ЦПУ в Astra Linux.

3. Разработка модуля ядра для зеркального отображения видеопотока камеры средствами драйвера V4L2 в Astra Linux.

4. Создание простого фильтр-драйвера для наложения текстового водяного знака (например, временной метки) на видеопоток в Astra Linux.

5. Модификация драйвера камеры для создания виртуального устройства, выводящего монохромное (черно-белое) изображение в Astra Linux.

#### **Управление устройствами**

6. Разработка драйвера для светодиодной подсветки, управляемой через GPIO, и его интеграция с драйвером камеры в Astra Linux.

7. Добавление в драйвер USB-камеры поддержки управления фокусом через стандартный интерфейс V4L2 в Astra Linux.

8. Создание простого драйвера "кнопки" для инициализации захвата кадра по аппаратному сигналу в Astra Linux.

9. Модификация драйвера камеры для сохранения последнего кадра в буфер ядра по внешнему прерыванию в Astra Linux.

10. Реализация в драйвере камеры возможности циклического захвата кадров с заданной частотой в Astra Linux.

#### **Виртуализация и эмуляция устройств**

11. Разработка драйвера виртуальной камеры, транслирующего статическое изображение (например, шахматную доску) в Astra Linux.

12. Создание драйвера виртуального устройства, эмулирующего работу камеры с заданным разрешением и частотой кадров в Astra Linux.

13. Модификация драйвера loopback-камеры для передачи видеопотока между процессами в Astra Linux.

14. Разработка простого драйвера, предоставляющего интерфейс V4L2 для чтения изображений из директории на диске в Astra Linux.

15. Создание драйвера виртуального устройства для вывода тестового видео PATTERN (например, цветных полос) в Astra Linux.

#### **Безопасность**

16. Реализация в драйвере камеры простого механизма блокировки устройства при превышении порога неудачных попыток открытия в Astra Linux.

17. Модификация драйвера камеры для ведения базового журнала доступа (логирование операций open/close) в Astra Linux.

18. Создание драйвера-фильтра, запрещающего изменение разрешения камеры непривилегированным пользователям в Astra Linux.

19. Добавление в драйвер камеры проверки цифровой подписи загружаемой микропрограммы (на примере заглушки) в Astra Linux.

20. Разработка драйвера виртуальной камеры, транслирующего "заклушку" при невозможности доступа к физическому устройству в Astra Linux.

## 2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1	Проработка и повторение лекционного материала, материала учебной и научной литературы, подготовка к семинарским занятиям	Методические указания по выполнению самостоятельной работы, утвержденные на заседании кафедры вычислительных технологий ФКТиПМ ФГБОУ ВО «КубГУ», протокол №7 от 07.05.2025

### 3. Образовательные технологии

Программа дисциплины предусматривает использование в учебном процессе следующих образовательных технологий: чтение лекций с использованием мультимедийных технологий; метод малых групп, разбор практических задач и кейсов.

При обучении используются следующие образовательные технологии:

- Технология коммуникативного обучения – направлена на формирование коммуникативной компетентности студентов, которая является базовой, необходимой для адаптации к современным условиям межкультурной коммуникации.
- Технология разноуровневого (дифференцированного) обучения – предполагает осуществление познавательной деятельности студентов с учётом их индивидуальных способностей, возможностей и интересов, поощряя их реализовывать свой творческий потенциал. Создание и использование диагностических тестов является неотъемлемой частью данной технологии.
- Технология модульного обучения – предусматривает деление содержания дисциплины на достаточно автономные разделы (модули), интегрированные в общий курс.
- Информационно-коммуникационные технологии (ИКТ) – расширяют рамки образовательного процесса, повышая его практическую направленность, способствуют интенсификации самостоятельной работы учащихся и повышению познавательной активности. В рамках ИКТ выделяются 2 вида технологий:
- Технология использования компьютерных программ – позволяет эффективно дополнить процесс обучения языку на всех уровнях.
- Интернет-технологии – предоставляют широкие возможности для поиска информации, разработки научных проектов, ведения научных исследований.
- Технология индивидуализации обучения – помогает реализовывать личностно-ориентированный подход, учитывая индивидуальные особенности и потребности учащихся.
- Проектная технология – ориентирована на моделирование социального взаимодействия учащихся с целью решения задачи, которая определяется в рамках профессиональной подготовки, выделяя ту или иную предметную область.
- Технология обучения в сотрудничестве – реализует идею взаимного обучения, осуществляя как индивидуальную, так и коллективную ответственность за решение учебных задач.
- Игровая технология – позволяет развивать навыки рассмотрения ряда возможных способов решения проблем, активизируя мышление студентов и раскрывая личностный потенциал каждого учащегося.

- Технология развития критического мышления – способствует формированию разносторонней личности, способной критически относиться к информации, умению отбирать информацию для решения поставленной задачи.

Комплексное использование в учебном процессе всех вышеназванных технологий стимулируют личностную, интеллектуальную активность, развивают познавательные процессы, способствуют формированию компетенций, которыми должен обладать будущий специалист.

Основные виды интерактивных образовательных технологий включают в себя:

- работа в малых группах (команде) - совместная деятельность студентов в группе под руководством лидера, направленная на решение общей задачи путём творческого сложения результатов индивидуальной работы членов команды с делением полномочий и ответственности;
- проектная технология - индивидуальная или коллективная деятельность по отбору, распределению и систематизации материала по определенной теме, в результате которой составляется проект;
- анализ конкретных ситуаций - анализ реальных проблемных ситуаций, имевших место в соответствующей области профессиональной деятельности, и поиск вариантов лучших решений;
- развитие критического мышления – образовательная деятельность, направленная на развитие у студентов разумного, рефлексивного мышления, способного выдвинуть новые идеи и увидеть новые возможности.

Подход разбора конкретных задач и ситуаций широко используется как преподавателем, так и студентами во время лекций, лабораторных занятий и анализа результатов самостоятельной работы. Это обусловлено тем, что при исследовании и решении каждой конкретной задачи имеется, как правило, несколько методов, а это требует разбора и оценки целой совокупности конкретных ситуаций.

Темы, задания и вопросы для самостоятельной работы призваны сформировать навыки поиска информации, умения самостоятельно расширять и углублять знания, полученные в ходе лекционных и практических занятий.

Подход разбора конкретных ситуаций широко используется как преподавателем, так и студентами при проведении анализа результатов самостоятельной работы.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

#### **4. Оценочные средства для текущего контроля успеваемости и промежуточной аттестации**

##### **4.1 Оценочные средства для текущего контроля успеваемости и промежуточной аттестации**

Оценочные средства предназначены для контроля и оценки образовательных

достижений обучающихся, освоивших программу учебной дисциплины «Операционные системы».

Фонд оценочных средств дисциплины состоит из средств текущего контроля выполнения заданий, лабораторных работ, тестирования, средств итоговой аттестации (экзамен в 3 семестре).

Оценка успеваемости осуществляется по результатам:

- выполнения лабораторных работ;
- ответов на теоретические вопросы при сдаче лабораторных работ;
- тестирования;
- ответов на вопросы экзаменационного билета.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

#### **Структура оценочных средств для текущей и промежуточной аттестации**

№ п/п	Код и наименование индикатора	Результаты обучения	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
<b><u>на базовом уровне:</u></b>				
1	ОПК-4.1 Аргументировано применяет современные информационные технологии, в том числе отечественные, при создании программных продуктов и	Знает базовые информационные технологии, используемые при создании программных продуктов	Опрос по теме, лабораторная работа	Тест Вопросы на экзамен 1-15

	программных комплексов различного назначения	необходимых для функционирования ОС		
2	ОПК-4.2 Ориентируется в современных положениях и концепциях прикладного и системного программного обеспечения, архитектуры компьютеров и сетей (в том числе и глобальных), технологии создания и сопровождения программных продуктов и программных комплексов	Умеет ориентироваться в современных положениях прикладного программного обеспечения	Опрос по теме, лабораторная работа	Тест Вопросы на экзамен 16-28
3	ОПК-5.1 Аргументировано применяет методы проектирования, разработки и реализации программных продуктов и программных комплексов в различных областях человеческой деятельности	Знает базовые методы проектирования, разработки и реализации программных продуктов для решения прикладных задач в области ОС	Опрос по теме, лабораторная работа	Тест Вопросы на экзамен 29-37
4	ОПК-5.2 Использует инструментальные, программные и аппаратные средства измерений для оценки качества программного обеспечения	Умеет использовать базовые программные и аппаратные средства измерений для оценки качества ПО	Опрос по теме, лабораторная работа	Тест Вопросы на экзамен 38-48
5	PL-3.1 Разрабатывает и отлаживает эффективные многопоточные решения на C++, тестирует, испытывает и оценивает качество таких решений	Знает основы синтаксиса языка. Знает общие принципы параллельных вычислений и понимает проблемы, возникающие при распараллеливании алгоритмов. Проводит распараллеливание простого алгоритма с применением OpenMP, стандартных библиотек C/C++ или др.	Опрос по теме, лабораторная работа	Тест Вопросы на экзамен 49-53
6	PL-3.2 Разрабатывает и отлаживает системы ИИ на C++ под конкретные аппаратные платформы с ограничениями по вычислительной мощности, в том числе для встроенных систем	Знает основы синтаксиса языка, основы построения систем ИИ. Пишет небольшие программы для обработки данных. Понимает возможности	Опрос по теме, лабораторная работа	Тест Вопросы на экзамен 54-57

		и ограничения встроенных систем.		
7	SS-2.2 Учитывает профессиональные и ролевые особенности коллег при совместной разработке технических решений и представлении результатов	Знает распределение ролей (без активного участия в общей координации). Умеет ориентироваться в структуре общего результата проекта. Владеет навыками подготовки презентации по своей части (например, рассказывает про архитектуру ИИ-системы или метрики).	Опрос по теме, выполнение курсового проекта	Тест Вопросы на экзамен 55-56
8	DL-3.2 Определяет стек технологий, методов и алгоритмов для построения продуктов с компьютерным зрением (системы видеоаналитики, поисковые системы по изображениям и т.д.)	Знает принципы представления изображений и кодирования цвета; базовые задачи анализа изображений (классификация, детекция, сегментация). Умеет применять фильтрацию изображений, включая частотную; применять известные архитектуры нейронных сетей (CNN) в решении простых задач распознавания изображений Владеет аппаратом математической морфологии	Опрос по теме, выполнение курсового проекта	Тест Вопросы на экзамен 55-56
<b><u>на продвинутом уровне:</u></b>				
9	ОПК-4.1 Аргументировано применяет современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения	Знает современные информационные технологии, используемые при создании программных продуктов необходимых для и стабильного функционирования ОС	Опрос по теме, лабораторная работа	Тест Вопросы на экзамен 1-15

10	ОПК-4.2 Ориентируется в современных положениях и концепциях прикладного и системного программного обеспечения, архитектуры компьютеров и сетей (в том числе и глобальных), технологии создания и сопровождения программных продуктов и программных комплексов	Умеет ориентироваться в современных положениях и концепциях прикладного и системного программного обеспечения, архитектуре компьютеров	Опрос по теме, лабораторная работа	Тест Вопросы на экзамен 16-28
11	ОПК-5.1 Аргументировано применяет методы проектирования, разработки и реализации программных продуктов и программных комплексов в различных областях человеческой деятельности	Знает методы проектирования, разработки и реализации программных продуктов и программных комплексов для решения прикладных задач в области ОС	Опрос по теме, лабораторная работа	Тест Вопросы на экзамен 29-37
12	ОПК-5.2 Использует инструментальные, программные и аппаратные средства измерений для оценки качества программного обеспечения	Умеет использовать инструментальные, программные и аппаратные средства измерений для оценки качества ПО	Опрос по теме, лабораторная работа	Тест Вопросы на экзамен 38-48
13	PL-3.1 Разрабатывает и отлаживает эффективные многопоточные решения на C++, тестирует, испытывает и оценивает качество таких решений	Решает проблемы одновременного доступа к данным из нескольких потоков, грамотно применяет атомарные операции и механизм блокировок. Оценивает производительность, умеет профилировать код и устраняет найденные узкие места.	Опрос по теме, лабораторная работа	Тест Вопросы на экзамен 49-53
14	PL-3.2 Разрабатывает и отлаживает системы ИИ на C++ под конкретные аппаратные платформы с ограничениями по вычислительной мощности, в том числе для встроженных систем	Понимает методы оптимизации моделей (квантование, сжатие весов модели и пр.) и вычислений ИИ. Находит и использует библиотеки, соответствующие решаемой задаче	Опрос по теме, лабораторная работа	Тест Вопросы на экзамен 54-57

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

### **Пример заданий на лабораторную работу**

#### **Задание 1.**

1. Перезагрузите тестовую виртуальную машину и в момент её запуска измените команды запуска так, чтобы отключить «тихую» загрузку ОС.
2. Перенастройте загрузчик, чтобы при последующих запусках ОС «тихий» режим загрузки ОС был отключен, а тайм-аут отображения меню загрузки GRUB был 7 секунд. Убедитесь, что настройки применились.
3. Выведите на экран целевое состояние системы по умолчанию. Перезагрузите машину, используя целевое состояние systemd.
4. Выведите на экран сообщения ядра ОС от начала загрузки и ознакомьтесь с ними.

#### **Задание 2.**

*Для успешного прохождения этого задания у виртуальной машины должны быть два CPU.*

1. Если вы не создали модули службы fly-message-monitor.service и resource-demanding1.service (раздел 17.6) в процессе изучения материала модуля, то создайте их, в том числе настройте контрольную группу для службы resource-demanding1.service.
2. Создайте службу **resource-demanding2.service** по аналогии со службой resource-demanding1.service, но так, чтобы скрипт, который она запускает, назывался **resource-demanding2.sh** и запускался на **втором процессоре**.
3. Создайте скрипт **/usr/local/bin/cpu\_usage.sh** со следующим содержимым:

```
#!/bin/bash
CPU_THRESHOLD_WARNING=85 # %
CPU_THRESHOLD_CRITICAL=95 # %
CPU_THRESHOLD_PERIOD=5 # секунд
CPU_THRESHOLD_ITERATION=3 # раз
MSG_FILE="/tmp/fly-msg/msg.txt"

while ;; do
# Get the first line with aggregate of all CPUs
cpu_now=$(head -n1 /proc/stat)
# Get all columns but skip the first (which is the "cpu" string)
cpu_sum="${cpu_now[@]:1}"
# Replace the column separator (space) with +
cpu_sum=$(( ${cpu_sum// /+} ))
# Get the delta between two reads
cpu_delta=$((cpu_sum - cpu_last_sum))
# Get the idle time Delta
cpu_idle=$((cpu_now[4] - cpu_last[4]))
# Calc time spent working
cpu_used=$((cpu_delta - cpu_idle))
# Calc percentage
cpu_usage=$((100 \* cpu_used / cpu_delta))

# Keep this as last for our next read
cpu_last="${cpu_now[@]}"
cpu_last_sum=$cpu_sum

if [[ "$cpu_usage" -ge "$CPU_THRESHOLD_WARNING" ]]; then
((cpu_warning_count++))
```

```

if [[ "$cpu_usage" -ge "$CPU_THRESHOLD_CRITICAL" ]]; then
    ((cpu_critical_count++))
else
    cpu_critical_count=0
fi
else
    cpu_warning_count=0
    cpu_critical_count=0
fi

msg_critical="critical\\ndialog-warning\\n<H1>Критически высокая нагрузка на
CPU!</H1><p>Использование CPU $cpu_usage%\\t$(date)</p>"
msg_warning="critical\\ndialog-warning\\n<H1>Высокая нагрузка на CPU!</H1><p>Использование CPU
$cpu_usage%\\t$(date)</p>"
msg_ok="normal\\ndialog-information\\n<H1>Нагрузка на CPU в пределах
нормы</H1><p>Использование CPU
$cpu_usage%\\t$(date)</p>"

if [[ cpu_critical_count -ge CPU_THRESHOLD_ITERATION ]]; then
    echo -en $msg_critical > $MSG_FILE
    send_ok=1
elif [[ cpu_warning_count -ge CPU_THRESHOLD_ITERATION ]]; then
    echo -en $msg_warning > $MSG_FILE
    send_ok=1
elif [[ send_ok -eq 1 ]]; then
    echo -en $msg_ok > $MSG_FILE
    send_ok=0
fi

# Wait a second before the next read
sleep $CPU_THRESHOLD_PERIOD

echo "CPU usage at $cpu_usage%"
echo "Warning counter $cpu_warning_count"
echo "Critical counter $cpu_critical_count"
done

```

4. Создайте модуль службы и назовите его **cpu-monitor.service**, чтобы он запускал скрипт **/usr/local/bin/cpu\_usage.sh** и стартовал автоматически при загрузке ОС только при цели **graphical.target**.

5. Выведите на экран содержимое конфигурационного файла модуля **cpu-monitor.service**, используя **systemctl**.

6. Отредактируйте файл модуля так, чтобы он запускался только после службы **fly-message-monitor.service**, используя утилиту **systemctl**.

7. Запустите службу **cpu-monitor.service** и убедитесь, что она работает.

8. Переведите систему в целевое состояние **multi-user.target** и убедитесь, что службы **cpu-monitor.service** и **fly-message-monitor.service** не запущены.

9. Переведите систему в целевое состояние **graphical.target** и убедитесь, что службы **cpu-monitor.service** и **fly-message-monitor.service** запустились.

10. Запустите службу **resource-demanding1.service**.

11. Запустите службу **resource-demanding2.service** и проверьте, что при высокой (более 85%) нагрузке на CPU вы получаете сообщения от скрипта мониторинга.

### Задание 3.

1. Измените настройки контрольной группы службы **resource-demanding1.service** так, чтобы суммарная нагрузка на два CPU была в пределах **85-94%**. Убедитесь, что сообщение от скрипта мониторинга изменилось.

2. Измените настройки контрольной группы службы **resource-demanding1.service** так, чтобы **суммарная нагрузка** на два CPU была ниже **85%**. Убедитесь, что сообщение от скрипта мониторинга изменилось и больше не появляется.

3. Выведите на экран дерево контрольных групп и найдите в нем контрольную группу для сервиса **resource-demanding1.service**.

#### **Задание 4.**

1. Выведите на экран все запущенные модули подсистемы **systemd**.

2. Выведите на экран все модули служб подсистемы **systemd**.

3. Выведите на экран в отдельном терминале информацию обо всех поступающих в реальном времени событиях подсистемы **systemd**.

4. Перезапустите службу **resource-demanding1.service**. Выведите на экран её статус.

5. Остановите службы **resource-demanding1.service**, **resource-demanding2.service** и **cpu-monitoring.service**.

6. Выведите на экран информацию из журналов о сервисе **resource-demanding1.service** в сокращённом виде.

### **Тест по разделам курса «Операционные системы»**

1.Какая версия Astra Linux является свободно распространяемой ОС общего назначения?

- а) Special Edition,
- б) Free Edition,
- в) Ни одна из перечисленных,
- г) Common Edition.

2. Что происходит на этапе «Развитие» жизненного цикла ОС?

а) Разработка и выпуск оперативных и срочных обновлений направленных на исправление ошибок и устранение уязвимостей, с возможностью включения доработанной или новой функциональности.

б) Выпуск оперативных и срочных обновлений, направленных только на исправление ошибок и устранение уязвимостей.

в) Разработка и выпуск очередного обновления, поколения ОС

г) Консультации по эксплуатации ОС на первой и второй линии поддержки, поддержка третьей линии исключена.

3. К какому классу относится ядро Linux?

- а) Pico/nano,
- б) Монолитное,
- в) Гибридное,
- г) Микро.

4. В каком компоненте архитектуры ОС Linux приложения пользователя могут получить доступ к ресурсам компьютера через системные вызовы ядра?

Пользовательское пространство

- а) Пространство ядра,
- б) Информационное обеспечение,
- в) Аппаратное обеспечение.

5. В каком режиме процессора работают пользовательские процессы?

- а) Смешанном,
- б) Непривилегированном,
- в) Эмуляции,
- г) Привилегированном.
- д)

6. Какие шаги нужно будет пройти в процессе установки ОС Astra Linux SE?

- а) Настройка учётных записей пользователей и паролей.
- б) Настройка средств криптографической защиты.
- в) Установка драйверов для материнской и видеокарты.
- г) Создание групп пользователей и настройка прав групп пользователей.
- д) Разметка дисков.
- е) Выбор программного обеспечения.

7. Какие настройки можно выполнить в программе настройки GRUB2?

- а) Установить настройки режима «Киоск».
- б) Включить/отключить проверку наличия операционных систем на других разделах дискового пространства.
- в) Настроить автоматическое монтирование логических дисков.
- г) Включить пункты меню для загрузки в режиме восстановления.
- д) Настроить внешний вид меню загрузчика.
- е) Изменить список программ автоматической загрузки.

8. Для запуска программы настройки параметров графического входа в систему, а также внешнего вида экрана авторизации необходимо нажать:

- а) Пуск → Панель управления → Безопасность → Конфигурация аудита.
- б) Пуск → Панель управления → Прочее → Параметры системы.
- в) Пуск → Панель управления → Безопасность → Политика безопасности.
- г) Пуск → Панель управления → Система → Вход в систему.

9. Какой компонент репозитория содержит свободное программное обеспечение для работы которого требуется проприетарное ПО?

- а) non-free,
- б) pool,

- в) main,
- г) contrib.

10. Какой репозиторий Astra Linux содержит основной состав дистрибутива, реализующий все функциональные возможности продукта и функции безопасности?

- а) [https://dl.astralinux.ru/astra/stable/1.7\\_x86-64/repository-base/](https://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-base/)
- б) [https://dl.astralinux.ru/astra/stable/1.7\\_x86-64/repository-contrib/](https://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-contrib/)
- в) [https://dl.astralinux.ru/astra/stable/1.7\\_x86-64/repository-main/](https://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-main/)
- г) [https://dl.astralinux.ru/astra/stable/1.7\\_x86-64/repository-extended/](https://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-extended/)

11. В каком разделе панели управления находится программа установки обновлений?

- а) Система,
- б) Прочее,
- в) Рабочий стол,
- г) Безопасность.

12. Сколько репозиториев содержится в одной строке файла /etc/apt/sources.list?

- а) Один,
- б) Четыре,
- в) Ни одного, т. к. запись одного репозитория состоит из двух строк,
- г) Два.

13. Какая максимальная длина имени файла в Astra Linux?

- а) 512 символов,
- б) 125 символов,
- в) 50 символов,
- г) 255 символов.

14. Какие виды файлов устройств бывают?

- а) Объектные,
- б) Блочные,
- в) Модульные,
- г) Символьные,
- д) Строковые.

15. Какая максимальная длина пути к файлу в Astra Linux?

- а) 4096 символов,
- б) 256 символов,
- в) 1024 символов,
- г) 512 символов.

16. Какую роль играют права доступа, установленные на символическую ссылку?

- а) Являются инструментом дополнительной защиты файлов или папок ограниченного доступа.

- б) Расширяет доступность целевого объекта другим пользователям или группам пользователей.
- в) Они не имеют значения, т. к. права доступа определяются правами файла, на который указывает ссылка.
- г) Определяют доступность целевого объекта пользователю или группе пользователей.

17. В файловых системах какого типа данные находятся на сетевых устройствах (серверах), но пользовательские процессы работают с ними, как с локальными?

- а) Временные,
- б) Дисковые,
- в) Псевдофайловые системы,
- г) Сетевые.

### **Зачетно-экзаменационные материалы для промежуточной аттестации (экзамен)**

1. Развитие и становление аппаратных возможностей компьютеров по этапам.
2. Базовые понятия и определения операционных систем.
3. Основы архитектуры вычислительной системы. Структура ВС.
4. Основы компьютерной архитектуры. Структура компьютера фон Неймана.
5. Базовая архитектура современных компьютеров.
6. Основы архитектуры операционных систем.
7. Структура ОС с монолитным ядром.
8. Структура ОС с микроядерной архитектурой.
9. Логические функции ОС.
10. Типы операционных систем.
11. Сетевые и распределенные операционные системы.
12. Управление процессами. Основные концепции.
13. Модель пакетной однопроцессной системы.
14. Модель пакетной мультипроцессной системы.
15. Модель ОС с разделением времени.
16. Модификация модели ОС с разделением времени.
17. Типы процессов
18. Контекст процесса.
19. Процесс ОС Unix. Пользовательская (программная) составляющая процесса.
20. Процесс ОС Unix. Аппаратная и системная составляющая процесса.
21. Базовые средства управления процессами в ОС Unix. `fork()` – создание нового процесса.
22. Базовые средства управления процессами в ОС Unix. `exec()` – семейство системных вызовов смены тела процесса
23. Базовые средства управления процессами в ОС Unix.. Завершение процесса
24. Базовые средства управления процессами в ОС Unix. `wait()` – ждем «завершения» процесса.
25. Жизненный цикл процесса. Состояния процесса.
26. Формирование процессов 0 и 1.
27. Взаимодействие процессов. Разделяемые ресурсы и синхронизация доступа к ним
28. Способы организации взаимного исключения.
29. Базовые средства реализации взаимодействия процессов в ОС Unix.
30. Реализация межпроцессного взаимодействия в ОС Unix. Сигналы.
31. Реализация межпроцессного взаимодействия в ОС Unix. Неименованные каналы.
32. Реализация межпроцессного взаимодействия в ОС Unix. Именованные каналы.
33. Система межпроцессного взаимодействия IPC (Inter-Process Communication).
34. Очередь сообщений IPC.
35. Разделяемая память IPC.
36. Массив семафоров IPC.

37. Сокеты — унифицированный интерфейс программирования распределенных систем
38. Взаимодействие нитей в ОС Unix.
39. Структурная организация файлов. Атрибуты файлов.
40. Модели реализации файлов и каталогов.
41. Надежность файловой системы.
42. Проверка целостности файловой системы.
43. Одиночное непрерывное распределение.
44. Распределение перемещаемыми разделами.
45. Распределение перемещаемыми разделами.
46. Страничное распределение.
47. Сегментное распределение.
48. Сегментно-страничное распределение.
49. Архитектура организации управления внешними устройствами.
50. Программное управление внешними устройствами.
51. Планирование дисковых обменов.
52. RAID-системы. Уровни RAID.
53. Файлы устройств, драйверы.
54. Системные таблицы драйверов устройств.
55. Ситуации, вызывающие обращение к функциям драйвера.
56. Включение, удаление драйверов из системы.
57. Буферизация при блок-ориентированном обмене. Борьба со сбоями.

*Перечень компетенций (части компетенции), проверяемых оценочным средством: ОПК-4.1; ОПК-4.2; ОПК-5.1; ОПК-5.2; SS-2.1; DL-3.2; PL-3.1; PL-3.2.*

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

##### **Методические рекомендации, определяющие процедуры оценивания выполнения лабораторных работ:**

Задание считается выполненным при выполнении следующих условий:

- предоставлен исходный код;
- продемонстрирована работоспособность программы;
- студент понимает исходный код и отвечает на вопросы по его организации.

##### **Методические рекомендации, определяющие процедуры оценивания тестовых заданий:**

- оценка «отлично» выставляется студенту при правильном выполнении 95-100 % заданий;
- оценка «хорошо» выставляется студенту при правильном выполнении 88-94% заданий;
- оценка «удовлетворительно» студенту при правильном выполнении 80-87% заданий;
- оценка «неудовлетворительно» при правильном выполнении менее 80% заданий.

##### **Методические рекомендации, определяющие процедуры оценивания выполнения курсового проекта:**

Задание считается выполненным при выполнении следующих условий:

- предоставлен исходный код;
- продемонстрирована его работоспособность;
- предоставлен оформленный по установленным требованиям текст пояснительной записки;
- студент понимает исходный код и отвечает на вопросы по теме курсового проекта.

**Методические рекомендации, определяющие процедуры оценивания на экзамене:**

**оценка «неудовлетворительно» выставляется в случае выполнения одного из условий:**

- непонимание сущности излагаемых вопросов, грубые ошибки в ответе, неуверенные и неточные ответы на дополнительные вопросы;
- предоставлен не работоспособный код для экзаменационной задачи;

**оценка «удовлетворительно» в случае выполнения условий:**

- частично ответил на два вопроса билета или достаточно полно ответил хотя бы на один вопрос;
- предоставлен код, логика работы которого верна, но содержит более 5 синтаксических ошибок;

**оценка «хорошо» в случае выполнения условий:**

- достаточно полно ответил на два вопроса билета;
- даны частичные ответы на дополнительные вопросы;
- предоставлен код, логика работы которого верна, но содержит не более 5 синтаксических ошибок;

**оценка «отлично» в случае выполнения условий:**

- глубокие исчерпывающие знания по вопросам билета;
- даны правильные и конкретные ответы на дополнительные вопросы;
- предоставлен работоспособный код

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

### 4.3. Методические указания по организации вычислительной инфраструктуры

#### Условия применения:

- Курс рассчитан на студентов 2 года обучения.
- Наличие доступа к ресурсам (Astra Linux, GitLab, компиляторы C++).

#### Цели, задачи и ожидаемые результаты:

##### Цели:

- Обеспечить студентов инструментами для работы с операционными системами.
- Приучить к использованию Git, CI/CD и автотестов для контроля качества кода.

##### Задачи преподавателя:

1. Создание учетных записей студентов в GitLab.
2. Настройка GitLab Runner для автоматического тестирования.
3. Разработка шаблонного репозитория для лабораторных работ.
4. Написание автотестов для проверки корректности реализации алгоритмов.
5. Визуализация результатов тестирования.

##### Ожидаемые результаты студентов:

- Умение работать с Git и CI/CD.
- Навыки написания чистого, тестируемого кода на C++.
- Понимание основных принципов функционирования ОС.

##### Порядок реализации:

#### 1. Создание учетных записей в GitLab:

Каждый студент получает доступ к приватному репозиторию.

#### 2. Настройка GitLab Runner:

Используется Docker-образ с компилятором C++ (g++/clang) и фреймворками для тестирования.

#### 3. Шаблонный репозиторий:

Включает:

- `.gitlab-ci.yml` для автоматического тестирования.
- `README.md` с инструкциями.
- Примеры кода.

#### 4. Визуализация результатов:

Генерация отчетов.

##### Порядок проверки корректности:

Чек-лист:

- Наличие Git-репозитория у всех студентов.
- Корректная работа CI/CD.
- Тесты покрывают ключевые функции.

### 4.4. Методические указания по организации лабораторных работ

#### Условия применения:

- Курс включает 15 лабораторных работ (см. раздел 2.3.3 РПД).
- Для выполнения требуется:

ОС Astra Linux

Среда разработки (Visual Studio).

GitLab для сдачи заданий.

#### Цели, задачи и ожидаемые результаты:

##### Цели:

– Закрепление теоретических знаний о структуре и функционировании ОС на практике.

- Развитие навыков алгоритмического мышления и реализации алгоритмов для решения прикладных задач.
- Подготовка к решению реальных задач в таких областях, как логистика, машинное обучение и проектирование.

**Задачи преподавателя:**

1. Подготовка плана лабораторных работ.
2. Разработка индивидуальных заданий.
3. Организация Git-инфраструктуры и автотестов.

**Ожидаемые результаты студентов:**

- Умение реализовывать и анализировать алгоритмы.
- Навыки работы с Git и CI/CD.
- Опыт решения прикладных задач из разных предметных областей (поиск экстремума функций, условная оптимизация, комбинаторные задачи).

**Порядок реализации:**

**1. План лабораторных работ:**

Соответствует темам из РПД (знакомство с операционной системой Linux, работа с терминалом и оболочкой Bash, загрузка системы и управление службами systemd и т. д.).

**2. Пример задания (ЛР №4: " Загрузка системы и управление службами systemd "):**

- Перезагрузить тестовую виртуальную машину и в момент её запуска изменить команды запуска так, чтобы отключить «тихую» загрузку ОС.
- Перенастроить загрузчик, чтобы при последующих запусках ОС «тихий» режим загрузки ОС был отключен, а тайм-аут отображения меню загрузки GRUB был 7 секунд. Убедиться, что настройки применились.
- Вывести на экран целевое состояние системы по умолчанию. Перезагрузить машину, используя целевое состояние systemd.
- Вывести на экран сообщения ядра ОС от начала загрузки и ознакомиться с ними.

**3. Критерии оценки:**

- **зачтено:** Полная реализация всех заданий.
- **Незачтено:** Задания выполнены не в полном объеме, не соответствует ТЗ.

#### **4.5. Методические указания по организации проектной деятельности студентов**

**Условия применения:**

- Курс включает проектную работу (5 часов).
- Доступны кейсы от преподавателей.

**Цели, задачи и ожидаемые результаты:**

**Цели:**

- Применение знаний ОС в реальных задачах.
- Развитие навыков командной работы.

**Задачи преподавателя:**

1. Сбор кейсов (например, оптимизация поиска в больших данных).
2. Формирование ТЗ для проектов.
3. Разработка системы оценки.

**Ожидаемые результаты студентов:**

Умение решать прикладные задачи с применением полученных знаний.

**Порядок реализации:**

**1. Пример кейса:**

Мультимодальный агент для анализа строительных площадок.

**2. ТЗ для проекта:**

- Реализовать ИИ-агента, способного анализировать изображения со стройплощадки (видео/фото).
- Оценить эффективность распознавания.

### 3. Критерии оценки:

- Корректность реализации.
- Анализ эффективности.

### Порядок проверки корректности:

Чек-лист:

- Наличие минимум 5 кейсов.
- Четкие критерии оценки проектов.

Рассмотрим примеры кейсов.

### **Кейс 1. Мультимодальный ассистент для банковских отделений**

#### **Описание:**

Физические отделения банка внедряют интерактивных консультантов. Предполагается создание мультимодального ИИ-ассистента, который воспринимает речь и визуально ориентируется в пространстве (распознаёт клиента, документы, банкоматы), а также отвечает голосом.

#### **Цель:**

Разработать базовый прототип, имитирующий функциональность помощника: ответы на типовые запросы, визуальные подсказки, навигация по отделению.

#### **Ожидаемый результат:**

Интерактивная модель, объединяющая голосовой ввод, зрительное восприятие (например, QR-код паспорта), текстовый вывод и жестовую реакцию.

### **Кейс 2. Мультимодальный агент для анализа строительных площадок**

#### **Описание:**

Компания разрабатывает систему для мониторинга строительных объектов. Требуется создать прототип мультимодального ИИ-агента, способного анализировать изображения со стройплощадки (видео/фото), а также принимать голосовые и текстовые запросы (например, «проверь монтаж перекрытия на 5 этаже»).

#### **Цель:**

Объединить возможности компьютерного зрения (распознавание стадии строительства, техники, нарушений) и НЛП (понимание запросов, отчётов).

#### **Ожидаемый результат:**

Интерактивный агент, который на запрос специалиста может показать нужный участок, прокомментировать прогресс, зафиксировать нарушения.

### **Кейс 3. Генерация рекламного контента для жилых комплексов**

#### **Описание:**

Компания регулярно запускает маркетинговые предложения для жилых комплексов. Необходимо исследовать использование диффузионных моделей для генерации изображений (визуализации интерьеров, окрестностей, видов из окон) и LLM — для описаний квартир, преимуществ района, инфраструктуры.

#### **Цель:**

Создать инструменты для быстрой генерации продающих материалов без привлечения дизайнеров и копирайтеров на первых этапах.

#### **Ожидаемый результат:**

Набор сгенерированных карточек объектов с текстом, изображением и логикой «живого» рекламного сообщения.

### **Кейс 4. Модель прогнозирования сроков сдачи объектов на основе текстовых и визуальных данных**

**Описание:**

Девелоперская компания ведёт аналитический архив по срокам строительства. С помощью мультимодальных моделей (текстовые отчёты + фото стройки) можно прогнозировать вероятность отклонения от графика сдачи.

**Цель:**

Разработать модель, которая по текущему статусу объекта (фото, отчёт СМР) оценивает риски задержек.

**Ожидаемый результат:**

Прототип, который показывает вероятность отклонений и даёт текстовые пояснения (основанные на распознанных признаках — «не завершены фасадные работы», «монтаж инженерии не начат»).

**Кейс 5. Мультимодальный агент для мониторинга строительной площадки****Описание:**

Компания-застройщик хочет создать систему для мониторинга хода строительства с помощью ИИ. Предлагается прототип мультимодального агента, способного анализировать визуальные данные (фото и видео со стройплощадки) и понимать голосовые или текстовые запросы инженеров.

**Цель:**

Объединить в одном решении анализ изображений и понимание естественного языка для контроля за строительством. Необходимо настроить модель так, чтобы по запросу она находила нужный визуальный фрагмент (например, снимок определённого этажа), оценила его (выявила, что сделано, а что нет) и сформулировала ответ для пользователя. Цель проекта – доказать, что мультимодальный подход может ускорить проверку прогресса работ и обнаружение проблем на объекте.

**Ожидаемый результат:**

Прототип интерактивного агента, который по запросу специалиста показывает нужный участок стройки, комментирует статус (например: «Перекрытие на 5 этаже установлено на 80%») и фиксирует обнаруженные нарушения.

**5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)****5.1 Учебная литература:**

1. Зверева, О. М. Операционные системы : учебное пособие / О. М. Зверева ; науч. ред. Л. Г. Доросинский ; Уральский федеральный университет им. первого Президента России Б. Н. Ельцина. – Екатеринбург : Издательство Уральского университета, 2020. – 223 с. : ил., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=699030>. – Библиогр. в кн. – ISBN 978-5-7996-3146-8. – Текст : электронный.
2. Исаева, Г. Н. Операционные системы, среды и оболочки : практикум : учебное пособие : [16+] / Г. Н. Исаева, Н. П. Сидорова ; Технологический университет. – Москва : Директ-Медиа, 2022. – 51 с. : ил., схем. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=693549>. – Библиогр.: с. 49. – ISBN 978-5-4499-3324-9. – DOI 10.23681/693549. – Текст : электронный.
3. Операционные системы : методические рекомендации по подготовке к экзамену : учебно-методическое пособие : [12+] / сост. Е. Е. Новикова ; Витебский государственный технический колледж. – Витебск : Витебский государственный технический колледж, 2022. – 52 с. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=702623>. – Библиогр. в кн. – Текст : электронный.
4. Иванов, М. С. Методика настройки операционной системы специального назначения «Astra Linux Special Edition» для обеспечения защиты информации, содержащей сведения, составляющие государственную тайну : [16+] / М. С. Иванов ; Пензенский

- Государственный Университет, Политехнический институт, Факультет информационных технологий и электроники, Кафедра Информационная безопасность систем и технологий. – Пенза : б.и., 2022. – 67 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=692251>. – Текст : электронный.
5. Операционные системы : учебное пособие (лабораторный практикум) : направление подготовки 01.03.02 Прикладная математика и информатика : практикум : [16+] / авт.-сост. А. В. Шапошников, П. А. Ляхов, А. С. Ионисян ; Северо-Кавказский федеральный университет. – Ставрополь : Северо-Кавказский Федеральный университет (СКФУ), 2022. – 143 с. : ил., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=712331>. – Библиогр.: с. 139. – Текст : электронный.
  6. Пирогов, В. Ю. Введение в программирование на языке ассемблера GAS в операционной системе Linux : учебное пособие для студентов : [16+] / В. Ю. Пирогов ; Шадринский государственный педагогический университет. – Шадринск : Шадринский государственный педагогический университет, 2022. – 292 с. : ил., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=702869>. – Библиогр. в кн. – ISBN 978-5-87818-642-1. – Текст : электронный.

## 5.2. Периодические издания:

1. Базы данных компании «Ист Вью» <http://dlib.eastview.com>
2. Электронная библиотека GREBENNIKON.RU <https://grebennikon.ru/>

## 5.3. Интернет-ресурсы, в том числе современные профессиональные базы данных и информационные справочные системы

### Электронно-библиотечные системы (ЭБС):

1. ЭБС «ЮРАЙТ» <https://urait.ru/>
2. ЭБС «УНИВЕРСИТЕТСКАЯ БИБЛИОТЕКА ОНЛАЙН» [www.biblioclub.ru](http://www.biblioclub.ru)
3. ЭБС «BOOK.ru» <https://www.book.ru>
4. ЭБС «ZNANIUM.COM» [www.znanium.com](http://www.znanium.com)
5. ЭБС «ЛАНЬ» <https://e.lanbook.com>

### Профессиональные базы данных:

1. Scopus <http://www.scopus.com/>
2. ScienceDirect <https://www.sciencedirect.com/>
3. Журналы издательства Wiley <https://onlinelibrary.wiley.com/>
4. Научная электронная библиотека (НЭБ) <http://www.elibrary.ru/>
5. Полнотекстовые архивы ведущих западных научных журналов на Российской платформе научных журналов НЭИКОН <http://archive.neicon.ru>
6. Национальная электронная библиотека (доступ к Электронной библиотеке диссертаций Российской государственной библиотеки (РГБ) <https://rusneb.ru/>
7. Президентская библиотека им. Б.Н. Ельцина <https://www.prlib.ru/>
8. База данных CSD Кембриджского центра кристаллографических данных (CCDC) <https://www.ccdc.cam.ac.uk/structures/>
9. Springer Journals: <https://link.springer.com/>
10. Springer Journals Archive: <https://link.springer.com/>
11. Nature Journals: <https://www.nature.com/>
12. Springer Nature Protocols and Methods: <https://experiments.springernature.com/sources/springer-protocols>
13. Springer Materials: <http://materials.springer.com/>
14. Nano Database: <https://nano.nature.com/>
15. Springer eBooks (i.e. 2020 eBook collections): <https://link.springer.com/>

16. "Лекториум ТВ" <http://www.lektorium.tv/>
17. Университетская информационная система РОССИЯ <http://uisrussia.msu.ru>

#### **Информационные справочные системы:**

1. Консультант Плюс - справочная правовая система (доступ по локальной сети с компьютеров библиотеки)

#### **Ресурсы свободного доступа:**

1. КиберЛенинка <http://cyberleninka.ru/>;
2. Американская патентная база данных <http://www.uspto.gov/patft/>
3. Министерство науки и высшего образования Российской Федерации <https://www.minobrnauki.gov.ru/>;
4. Федеральный портал "Российское образование" <http://www.edu.ru/>;
5. Информационная система "Единое окно доступа к образовательным ресурсам" <http://window.edu.ru/>;
6. Единая коллекция цифровых образовательных ресурсов <http://school-collection.edu.ru/> .
7. Проект Государственного института русского языка имени А.С. Пушкина "Образование на русском" <https://pushkininstitute.ru/>;
8. Справочно-информационный портал "Русский язык" <http://gramota.ru/>;
9. Служба тематических толковых словарей <http://www.glossary.ru/>;
10. Словари и энциклопедии <http://dic.academic.ru/>;
11. Образовательный портал "Учеба" <http://www.ucheba.com/>;
12. Законопроект "Об образовании в Российской Федерации". Вопросы и ответы [http://xn--273--84d1f.xn--p1ai/voprosy\\_i\\_otvety](http://xn--273--84d1f.xn--p1ai/voprosy_i_otvety)

#### **Собственные электронные образовательные и информационные ресурсы КубГУ:**

1. Электронный каталог Научной библиотеки КубГУ <http://megapro.kubsu.ru/MegaPro/Web>
2. Электронная библиотека трудов ученых КубГУ <http://megapro.kubsu.ru/MegaPro/UserEntry?Action=ToDb&idb=6>
3. Среда модульного динамического обучения <http://moodle.kubsu.ru>
4. База учебных планов, учебно-методических комплексов, публикаций и конференций <http://infoneeds.kubsu.ru/>
5. Библиотека информационных ресурсов кафедры информационных образовательных технологий <http://mschool.kubsu.ru;>
6. Электронный архив документов КубГУ <http://docspace.kubsu.ru/>
7. Электронные образовательные ресурсы кафедры информационных систем и технологий в образовании КубГУ и научно-методического журнала "ШКОЛЬНЫЕ ГОДЫ" <http://icdau.kubsu.ru/>

#### **5.4 Публикации конференций А\***

1. Ashwin Prasad, Sampath Rajendra , Kaushik Rajan , R. Govindarajan , Uday Bondhugula : SilvanForge: A Schedule-Guided Retargetable Compiler for Decision Tree Inference. ACM SIGOPS Symposium on Operating Systems Principles, November 4–6, 2024 · Hilton Austin, Texas, USA, <https://dl.acm.org/doi/epdf/10.1145/3694715.3695958>
2. Practical Verification of System-Software Components Written in Standard C Can Cebeci, Yonghao Zou, Diyu Zhou, George Candea, Clément Pit-Claudel EPFL, Switzerland. ACM SIGOPS Symposium on Operating Systems Principles, November 4–6, 2024 · Hilton Austin, Texas, USA, <https://dl.acm.org/doi/abs/10.1145/3694715.3695980>

## **6. Методические указания для обучающихся по освоению дисциплины (модуля)**

По курсу предусмотрено проведение лекционных занятий, на которых дается основной систематизированный материал, лабораторных работ, тестов, курсового проектирования и экзамена.

Важнейшим этапом курса является самостоятельная работа по дисциплине с использованием указанных литературных источников и методических указаний автора курса.

Виды и формы СР, сроки выполнения, формы контроля приведены выше в данном документе.

Для лучшего освоения дисциплины при защите ЛР студент должен ответить на несколько вопросов из лекционной части курса.

### **Методические рекомендации, определяющие процедуры оценивания выполнения лабораторных работ:**

Задание считается выполненным при выполнении следующих условий:

- предоставлен исходный код;
- продемонстрирована работоспособность программы;
- студент понимает исходный код и отвечает на вопросы по его организации.

### **Методические рекомендации, определяющие процедуры оценивания тестовых заданий:**

- оценка «отлично» выставляется студенту при правильном выполнении 95-100 % заданий;
- оценка «хорошо» выставляется студенту при правильном выполнении 88-94% заданий;
- оценка «удовлетворительно» студенту при правильном выполнении 80-87% заданий;
- оценка «неудовлетворительно» при правильном выполнении менее 80% заданий.

### **Методические рекомендации, определяющие процедуры оценивания выполнения курсового проекта:**

Задание считается выполненным при выполнении следующих условий:

- предоставлен исходный код;
- продемонстрирована его работоспособность;
- предоставлен оформленный по установленным требованиям текст пояснительной записки;
- студент понимает исходный код и отвечает на вопросы по теме курсового проекта.

### **Методические рекомендации, определяющие процедуры оценивания на экзамене:**

Процедура промежуточной аттестации проходит в соответствии с Положением о текущем контроле и промежуточной аттестации обучающихся ФГБОУ ВО «КубГУ».

Итоговой формой контроля сформированности компетенций у обучающихся по дисциплине является экзамен. Студенты обязаны сдать экзамен в соответствии с расписанием и учебным планом.

ФОС промежуточной аттестации состоит из вопросов к экзамену, задач по дисциплине и результатов текущего контроля.

Экзамен по дисциплине преследует цель оценить работу студента за курс, получение теоретических знаний, их прочность, развитие творческого мышления, приобретение навыков самостоятельной работы, умение применять полученные знания для решения практических задач.

Форма проведения экзамена: устно.

Экзаменатору предоставляется право задавать студентам дополнительные вопросы по всей учебной программе дисциплины.

Результат сдачи экзамена заносится преподавателем в экзаменационную ведомость и зачетную книжку.

Оценивание уровня освоения дисциплины основывается на качестве выполнения студентом заданий текущего контроля и ответов на вопросы экзамена.

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

## 7. Материально-техническое обеспечение по дисциплине (модулю)

Наименование специальных помещений	Оснащенность специальных помещений	Перечень лицензионного программного обеспечения
Учебная аудитория для проведения лекционных занятий, аудитория текущего контроля и промежуточной аттестации (ауд. 129, А305, А307).	Мебель: учебная мебель Технические средства обучения: проектор, экран/телевизор, компьютер/ноутбук с соответствующим программным обеспечением (ПО)	Программы для демонстрации и создания презентаций
Учебная аудитория для проведения занятий семинарского типа, учебная аудитория для проведения лабораторных занятий, аудитория текущего контроля и промежуточной аттестации, учебная аудитория для проведения индивидуальных и групповых консультаций (ауд. 147, 148, 149).	Мебель: учебная мебель Технические средства обучения: экран/телевизор, компьютер/ноутбук	Программы для демонстрации и создания презентаций
Компьютерный класс, учебная аудитория для проведения лабораторных занятий, учебная аудитория для проведения курсового проектирования (выполнения курсовых работ) (ауд. 101-107).	Мебель: учебная мебель Технические средства обучения: экран/телевизор, проектор, компьютер/ноутбук. Оборудование: компьютерная техника с выходом в Интернет, доступом в электронную информационно-образовательную среду,	системы программирования на языках высокого уровня, сетевой доступ к ресурсам, в частности С++ и пр. с возможностью многопользовательской работы, виртуальная машина*, ОС Astra Linux

Для самостоятельной работы обучающихся предусмотрены помещения, укомплектованные специализированной мебелью, оснащенные компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.

Наименование помещений для самостоятельной работы обучающихся	Оснащенность помещений для самостоятельной работы обучающихся	Перечень лицензионного программного обеспечения
<p>Помещение для самостоятельной работы обучающихся (читальный зал Научной библиотеки)</p>	<p>Мебель: учебная мебель Комплект специализированной мебели: компьютерные столы Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации, веб-камеры, коммуникационное оборудование, обеспечивающее доступ к сети интернет (проводное соединение и беспроводное соединение по технологии Wi-Fi)</p>	<p>Доступ к печатным и электронным информационным ресурсам</p>
<p>Помещение для самостоятельной работы обучающихся (ауд. 146 )</p>	<p>Мебель: учебная мебель Комплект специализированной мебели: компьютерные столы Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации, веб-камеры, коммуникационное оборудование, обеспечивающее доступ к сети интернет (проводное соединение и беспроводное соединение по технологии Wi-Fi)</p>	<p>Системы программирования на языках C++ и пр. с возможностью многопользовательской работы, виртуальная машина*, ОС Astra Linux</p>

\*Виртуальные машины в облаке предоставляется индустриальным партнером ПАО «Сбербанк»:

№	Продукт	Параметры продукта	Кол-во	Кол-во конфигураций	Ед. изм.
1	Виртуальная машина	Виртуальная машина 10% vCPU 2 vCPU 4 RAM	1	60	Шт
		ОС Ubuntu 22.04	1		Шт
		Системный диск SSD	1		Шт
			10		Гб
		Аренда публичного IP	1		Шт

Дополнительные облачные ресурсы предоставляются технологическим партнером Yandex Cloud.