

Федеральное государственное образовательное бюджетное учреждение
высшего образования

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ
РОССИЙСКОЙ ФЕДЕРАЦИИ»
(Финансовый университет)**

**Кафедра информационных технологий
Факультет информационных технологий и анализа больших данных**

Документ подписан усиленной неквалифицированной электронной подписью
Организация: Финансовый университет при Правительстве РФ
Утверждено: Проректор по учебной и методической работе Е.А. Каменева
Сертификат: 8xolKjlqX1EhPulbL7f5X/4Tywc8qrD4
Дата: 08.10.2025 г.

Г.Л. Деденко

Сетевые системы и приложения

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки:

09.03.04 - Программная инженерия,

Образовательная программа «Инженер-разработчик программного обеспечения»

Профиль:

«Инженер-разработчик программного обеспечения»

Рекомендовано

*Факультет информационных технологий и анализа больших данных
(протокол № 03 от 16.12.2025 г.)*

Одобрено

*Кафедра информационных технологий
(протокол № 12 от 03.12.2025 г.)*

© Москва 2026

СОДЕРЖАНИЕ

1.	Наименование дисциплины	3
2.	Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине	3
3.	Место дисциплины в структуре образовательной программы	5
4.	Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	5
5.	Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий	6
5.1.	Содержание дисциплины	6
5.2.	Учебно-тематический план	12
5.3.	Содержание семинаров	13
6.	Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	19
6.1.	Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	19
6.2.	Перечень вопросов, заданий, тем для подготовки к текущему контролю	26
7.	Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	28
8.	Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	33
9.	Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	34
10.	Методические указания для обучающихся по освоению дисциплины	36
11.	Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	40
12.	Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	41

1. Наименование дисциплины

«Сетевые системы и приложения».

2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
ОПК-5	Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	Устанавливает простое программное обеспечение для информационных и автоматизированных систем	знать: основные типы дистрибутивов Linux, принципы работы менеджеров пакетов (apt, dnf), структуру репозитория ПО, требования к аппаратному обеспечению для развертывания ОС уметь: производить установку операционной системы Linux на виртуальную или физическую машину, выполнять базовую постинсталляционную настройку, устанавливать и обновлять пакеты через консоль
		Настраивает программное и аппаратное обеспечение для решения конкретных задач предметной области	знать: конфигурационные файлы основных системных служб (SSH, Cron, Systemd), принципы настройки сетевых интерфейсов в Linux, механизм прав доступа и управления пользователями уметь: настраивать сетевые подключения (статический IP, DNS), конфигурировать службу удаленного доступа SSH (запрет root, вход по ключам), управлять автозапуском служб
		Проводит анализ информационной инфраструктуры, выявляет слабые места, вырабатывает рекомендации для ее улучшения	знать: средства мониторинга системных ресурсов (top, htop, vmstat), структуру и расположение системных журналов (logs), базовые принципы анализа производительности и безопасности системы уметь: анализировать логи системных служб для выявления

			ошибок, диагностировать узкие места в производительности (CPU, RAM, Disk I/O), проверять целостность файловой системы
ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	Разрабатывает алгоритмы решения простых информационных задач и выражает их на языке программирования	знать: синтаксис командной оболочки Bash, основы языка Python (модули os, sys, socket), основные алгоритмы работы с файловой системой и текстовыми потоками уметь: писать скрипты на Bash для автоматизации рутинных задач (резервное копирование, обработка текста), разрабатывать простые сетевые приложения на Python (клиент-сервер)
		Анализирует алгоритмы в части производительности, оптимальности, вырабатывает рекомендации для оптимизации алгоритмов программ	знать: отличия синхронного и асинхронного выполнения кода, понятия многопоточности и многопроцессности (threading vs multiprocessing), влияние блокирующих операций ввода-вывода на производительность уметь: выбирать подходящую модель конкурентности (потoki, процессы, async/await) для решения задачи, оптимизировать скрипты обработки данных с использованием конвейеров
		Проводит ручное и автоматизированное тестирование программных продуктов по методам черного и белого ящика, составляет набор тестовых случаев	знать: методы отладки сетевых приложений, принципы модульного тестирования кода на Python, инструменты для проверки доступности сетевых узлов и сервисов уметь: проводить ручное тестирование развернутых веб-сервисов (проверка кодов ответа, доступности контента), писать простейшие unit-тесты для функций на Python

3. Место дисциплины в структуре образовательной программы

Дисциплина «Сетевые системы и приложения» относится к «Общефакультетскому (предпрофильному) циклу».

4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся

Очно-заочная форма обучения

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 4 (в часах)	Семестр 5 (в часах)
Общая трудоёмкость дисциплины	8/288	144	144
Контактная работа- Аудиторные занятия	48	24	24
Лекции	16	8	8
Семинары, практические занятия	32	16	16
Самостоятельная работа	240	120	120
Вид текущего контроля	Контрольная работа, Контрольная работа	Контрольная работа	Контрольная работа
Вид промежуточной аттестации	Зачет, Экзамен	Зачет	Экзамен

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

5.1. Содержание дисциплины

Тема 1. Введение в ОС Linux

Базовые представления об операционных системах и месте Linux в современном ИТ-ландшафте. Функции ОС как посредника между оборудованием и прикладными программами, понятия процесса, файла, устройства. История развития UNIX и Linux, семейство дистрибутивов и их основные отличия, модель распространения свободного программного обеспечения. Сравнение Linux с Windows с точки зрения безопасности, устойчивости, использования ресурсов и возможностей администрирования. Работа в графической среде и в командной строке, знакомство со стандартной иерархией каталогов и типами файлов. Базовые навыки использования виртуальных машин в Oracle VirtualBox для установки и безопасного изучения Linux без риска для основной системы и пользовательских данных.

Тема 2. Основы командной строки

Текстовый терминал Linux как основной инструмент администратора и разработчика. Структура команды, обязательные и необязательные части, правила передачи аргументов и опций. Виртуальные терминалы и эмуляторы, механизм обработки команд оболочкой. Основные приёмы работы с файловой системой из командной строки: навигация по каталогам, создание, копирование, перемещение и удаление файлов и папок. Типы файлов в Linux, жёсткие и символические ссылки, их назначение и типичные сценарии использования. Утилиты архивации и сжатия для резервного копирования и переноса данных, встроенная система справки и внешние источники документации для самостоятельного поиска информации по командам.

Тема 3. Использование удалённого доступа

Организация безопасной удалённой работы с сервером Linux. Протокол SSH, его назначение и базовая схема установления защищённого соединения поверх ненадёжных сетей. Режимы аутентификации по паролю и по ключу, формат открытого и закрытого ключей, типовые команды генерации и установки ключей на удалённый хост. Работа в

удалённой текстовой сессии, перенаправление вывода, базовые приёмы администрирования системы «по сети». Программы управления сеансами и мультиплексоры терминала, такие как `tmux`, позволяющие поддерживать несколько одновременных сессий и восстанавливать работу после разрыва соединения. Способы подключения к серверу из среды Windows (например, через PuTTY) и базовые вопросы безопасности при предоставлении удалённого доступа.

Тема 4. Права и пользователи

Система безопасности современных операционных систем на примере Linux. Понятия пользователя, группы и суперпользователя `root`, их роли и область полномочий. Основные модели аутентификации, структура реестра пользователей, хранимые атрибуты учетных записей и принципы управления паролями. Система прав доступа к файлам и каталогам: чтение, запись, выполнение, специальные биты, а также алгоритм проверки прав при обращении к ресурсу. Команды создания, изменения и удаления учетных записей, включения в группы, смены пароля, а также утилиты просмотра и модификации прав доступа. Типичные ошибки настройки прав, риски, связанные с неосторожным использованием `root`, и базовые подходы к разграничению доступа в других семействах операционных систем.

Тема 5. Процессы и пакеты

Внутренняя организация Linux как многозадачная операционная система. Понятие ядра ОС, разграничение между пространством ядра и пользовательским пространством, роль системных вызовов. Понятие процесса, его жизненный цикл, состояния, идентификаторы и контекст выполнения. Базовые команды наблюдения и управления процессами, остановка, приостановка и изменение приоритетов. Установка программного обеспечения: программные репозитории, понятие пакета и его состав, классические менеджеры пакетов и их команды. Современные системы распространения приложений (`snapper`, `flatpak`) и установка программ из исходных текстов как способ развёртывания специализированного ПО при работе с сетевыми системами и приложениями.

Тема 6. Управление загрузкой Linux

Полный путь, который проходит система от включения питания до появления приглашения командной строки или графического интерфейса. Взаимодействие

микропрограммы компьютера с загрузчиком ОС, структура и роль загрузчика GRUB, понятие загрузочных записей и параметров ядра. Инициализатор системы и системные службы, логика перехода между этапами загрузки и рабочим состоянием. Команды просмотра состояния служб, их запуска, остановки, автоматического старта при загрузке. Создание и настройка собственной простой службы для запуска сетевого или вспомогательного приложения. Проблемы при загрузке Linux и общие подходы к их диагностике и устранению.

Тема 7. Файловые системы

Логическая организация данных на дисках и в хранилищах. Физические и логические диски, понятие раздела и файловой системы, причины разбиения диска на разделы. Типовые файловые системы Linux, их особенности и области применения. Процесс монтирования файловых систем, использование соответствующих команд, настройка автомонтирования и точки монтирования для различных разделов. Стандартная иерархия каталогов Linux и назначение основных системных директорий. Основные операции администрирования файловых систем, контроль свободного места, проверка и исправление ошибок. Понятие сетевых файловых систем как способа организации общего доступа к ресурсам в локальной сети, базовые сценарии их использования в прикладных информационных системах.

Тема 8. Основы скриптов на bash

Командный интерпретатор bash как полноценный язык сценарного программирования, позволяющий автоматизировать повседневные задачи администратора и разработчика. Синтаксис и базовые конструкции языка: переменные, арифметика, условия, циклы, функции, работа с параметрами командной строки и кодами возврата. Создание простых административных скриптов для обработки файлов, журналов и сетевых команд, а также организация вывода диагностической информации. Планировщик задач cron и средства создания регулярных заданий по расписанию. Вопросы читаемости и документирования сценариев, принципы безопасного выполнения потенциально опасных команд. Альтернативные оболочки, такие как zsh, и их отличия от bash в контексте работы с сетевыми системами и приложениями.

Тема 9. Средства обработки текста

Текстовые данные в командной строке как важная часть администрирования и разработки. Основные утилиты для просмотра, фильтрация и преобразование текстовых потоков, использование конвейеров и перенаправлений ввода-вывода. Поточковые текстовые редакторы и фильтры: `sed` и `awk`, применение регулярных выражений для поиска и замены фрагментов текста. Обработка структурированных файлов: журналов, конфигурационных файлов, табличных данных в текстовом формате. Написание простых парсеров и отчётов на `bash`, позволяющих извлекать нужную информацию из логов сетевых сервисов и системных сообщений. Типичные шаблоны автоматизации обработки текста в задачах сопровождения сетевых приложений.

Тема 10. Настройка сети в Linux

Базовые представления о компьютерных сетях, практическая конфигурация сетевых параметров в Linux. Ключевые понятия модели OSI и стека TCP/IP, схемы адресации и маршрутизации в локальных и глобальных сетях. Основные консольные команды для диагностики сетевых подключений, просмотра таблиц маршрутизации и настроек интерфейсов. Сценарии настройки статической и динамической адресации, проверки доступности сетевых ресурсов, анализа типовых неполадок. Понятие виртуальной сети и виды виртуальных сетевых подключений в среде VirtualBox (NAT, мост, host-only и др.), примеры их использования для моделирования клиент-серверных конфигураций. Основы трансляции сетевых адресов и общие принципы безопасной работы сетевых сервисов.

Тема 11. Средства ОС в Python

Использование языка Python для решения системных и административных задач в среде Linux. Стандартные библиотеки `os` и `pathlib` для работы с файловой системой: обход каталогов, создание и удаление файлов, получение атрибутов. Способы программного выполнения консольных команд с помощью модуля `subprocess`, получение кода возврата и перенаправление потоков ввода-вывода. Комбинирование возможности Python и ОС для автоматизации резервного копирования, обработки журналов, запуска вспомогательных утилит. Кроссплатформенность, различия в поведении команд и путей на разных операционных системах. Утилиты, обслуживающие сетевые приложения, дополнение скрипты `bash` Python при работе с сетевыми системами и приложениями.

Тема 12. Системы контроля версий

Принципы работы систем контроля версий как основы современной командной разработки программного обеспечения. Ключевые понятия git: репозиторий, коммит, дерево файлов, индекс, история изменений. Базовые команды и типичный цикл работы: инициализация репозитория, добавление файлов, фиксация и просмотр истории. Операции ветвления и слияния, назначение веток в рабочих процессах, способы разрешения конфликтов. Удалённые репозитории: клонирование, форк, отправка и получение изменений, синхронизация проектов через GitHub. Современные методологии командной разработки, в том числе подход GitFlow, и роль систем контроля версий в тестировании, развёртывании и сопровождении сетевых приложений. Графические клиенты и интеграция git со средами разработки.

Тема 13. Использование сокетов

Разработка сетевых приложений на уровне транспортного и прикладного слоя. Понятие сетевого сокета, его роль как программного интерфейса для обмена данными между машинами, виды сокетов и их характеристики. Схема работы клиент-серверного взаимодействия, принципы установления и завершения соединения. Транспортные протоколы TCP и UDP, их достоинства и ограничения для различных классов приложений. Реализация сокетов в языке Python с использованием модуля socket, простейшие клиент-серверные программы. Блокирующие операции, тайм-ауты и типовые ошибки при работе с сетью, способы их обработки и диагностики.

Тема 14. Асинхронное программирование

Современные подходы к организации параллельных и конкурентных вычислений при разработке сетевых приложений. Понятия процесса и потока, параллелизма и конкурентности, различия задач, ограниченных процессором и вводом-выводом. Средства многопоточности в Python (модуль threading), особенности планирования и типичные проблемы потокобезопасности. Многопроцессное программирование с использованием модуля multiprocessing как способ задействовать несколько ядер процессора. Асинхронное программирование и библиотека asyncio, позволяющие обслуживать множество сетевых запросов в одном потоке за счёт неблокирующего ввода-вывода. Механизмы синхронизации и блокировок, защита общих ресурсов, типовые шаблоны построения производительных асинхронных серверов.

Тема 15. Веб-сервер

Базовые веб-технологии и организация работы веб-серверов. Понятия URL, клиент-серверная модель web, протокол HTTP и структура типичного запроса и ответа. Роль веб-сервера как программного компонента, принимающего запросы от браузеров и передающего им статический или динамически формируемый контент. Установка и начальная настройка популярных веб-серверов Linux, таких как nginx или Apache, создание виртуальных хостов и привязку сайтов к доменным именам. Концептуальная схема работы веб-сервера, взаимодействие с приложениями, журналирование запросов и базовые настройки безопасности. Простейший веб-сервер и веб-страница, демонстрирующие принципы работы HTTP-службы в контексте сетевых систем и приложений.

Тема 16. Развёртывание сетевых приложений

Полный жизненный цикл вывода сетевого приложения в эксплуатацию. Общие этапы процесса развёртывания: подготовка инфраструктуры, конфигурация сервера, копирование артефактов приложения, настройка зависимостей и окружения. Различия тестовой и рабочей среды, принципы безопасного обновления и отката версий. Способы публикации приложений на веб-сервере, организация структуры каталогов, прав доступа и логирования. Основы мониторинга работы приложения и системных ресурсов, использование журналов и простых средств оповещения о сбоях. Средства автоматизации развёртывания и обновления, написание вспомогательных скриптов, которые сокращают количество ручных операций и снижают риск ошибок при сопровождении сетевых систем и приложений.

5.2. Учебно-тематический план

№ п/п	Наименование тем(разделов) дисциплины	Трудоемкость в часах				
		Всего	Контактная работа - Аудиторная работа			Самостоя тельная работа
			Общая, в т.ч.:	Лекции	Семинары, практическ ие занятия	
1	Введение в ОС Linux	18	3	1	2	15
2	Основы командной строки	18	3	1	2	15
3	Использование удалённого доступа	18	3	1	2	15
4	Права и пользователи	18	3	1	2	15
5	Процессы и пакеты	18	3	1	2	15
6	Управление загрузкой Linux	18	3	1	2	15
7	Файловые системы	18	3	1	2	15
8	Основы скриптов на bash	18	3	1	2	15
9	Средства обработки текста	18	3	1	2	15
10	Настройка сети в Linux	18	3	1	2	15
11	Средства ОС в Python	18	3	1	2	15
12	Системы контроля версий	18	3	1	2	15
13	Использование сокетов	18	3	1	2	15
14	Асинхронное программирование	18	3	1	2	15
15	Веб-сервер	18	3	1	2	15
16	Развёртывание сетевых приложений	18	3	1	2	15
	Итого	288	48	16	32	240

5.3. Содержание семинаров

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятий
Введение в ОС Linux	Каковы основные функции операционной системы и чем Linux отличается от других ОС? В чём ключевые особенности семейства UNIX и эволюции Linux-дистрибутивов? Какие преимущества и ограничения даёт использование свободного ПО в образовательной и профессиональной среде? Сравнение Linux и Windows: где и почему предпочтителен каждый из вариантов? Для чего в учебном процессе и в промышленности применяют виртуальные машины (VirtualBox и др.)? Как стандартная иерархия каталогов Linux влияет на организацию данных и администрирование?	Мини-лекция с обсуждением. Демонстрация работы ОС Linux и Windows (проектор). Практическая работа за ПК в VirtualBox (установка/запуск). Фронтальная беседа, ответы на вопросы.
Основы командной строки	Чем работа в терминале отличается от работы в графической среде, какие преимущества это даёт администратору? Какова структура команды в Linux и почему важно понимать аргументы и опции? Чем жёсткие ссылки отличаются от символических и где они применяются? Какие типичные комбинации команд используются для работы с файлами и каталогами? Как организовать простейшее резервное копирование средствами командной строки? Почему важно уметь пользоваться встроенной справкой (man, --help) и внешней документацией?	Практическая работа за ПК (отработка команд). Разбор типовых задач «как сделать...» через командную строку. Мини-тест по базовым командам. Работа в парах (один формулирует задачу, другой показывает решение в терминале).
Использование удалённого доступа	Почему SSH стал стандартом для удалённого администрирования Linux-серверов? В чём принципиальная разница между аутентификацией по паролю и по ключу? Какие риски возникают при неправильной настройке SSH-доступа? Для чего используются такие инструменты, как tmux, при удалённой работе? Как организовать удобную и безопасную работу с сервером из среды Windows (PuTTY и аналоги)? Какие минимальные меры безопасности стоит реализовать сразу после поднятия SSH-сервера?	Практическая работа за ПК (подключение по SSH к учебному серверу / ВМ). Демонстрация генерации и установки ключей. Разбор типовых ошибок подключения и их причин. Мини-коллоквиум по вопросам безопасности удалённого доступа.
Права и пользователи	Какова роль суперпользователя root и чем опасно его некорректное использование? Что такое	Практическая работа за ПК (создание

	пользователь и группа в Linux, как они связаны с безопасностью?Как интерпретировать строку прав доступа gwxg-x--- и аналогичные?Какие модели аутентификации применяются в современных ОС? Какие типичные ошибки в настройке прав приводят к уязвимостям?Как организовать разграничение доступа к ресурсам в учебной и производственной среде?	пользователей, групп, изменение прав).Разбор ситуационных задач «что произойдёт, если права заданы так...».Фронтальный опрос по синтаксису прав и командам chmod, chown, usermod.Мини-кейс: проектирование набора пользователей и групп для гипотетического проекта.
Процессы и пакеты	Что такое процесс, какие стадии его жизненного цикла можно выделить?Чем отличается процесс от потока с точки зрения ОС?Какие инструменты позволяют наблюдать и управлять процессами в Linux?Что такое «пакет» и «репозиторий», зачем нужны пакетные менеджеры?Как возникают зависимости пакетов и как они разрешаются?В каких случаях оправдана установка ПО из исходных текстов?	Практическая работа за ПК (наблюдение и управление процессами, ps, top, kill).Практика установки и удаления пакетов из репозитория.Обсуждение кейсов «сломалась зависимость» / «конфликт версий».Мини-тест по командам работы с процессами и пакетами.
Управление загрузкой Linux	Какие этапы проходит система от включения питания до входа пользователя?Какова роль загрузчика GRUB и что такое «запись загрузки»? Что такое системные службы и как они связаны с процессом инициализации?Для чего администратору нужно уметь управлять автозапуском служб?Какие типичные проблемы возникают при загрузке Linux и как их диагностировать?Зачем может понадобиться написание своей простой службы (unit-файла) для приложения?	Демонстрация последовательности загрузки с комментарием преподавателя.Практическая работа: просмотр и управление службами (systemctl).Разбор примера собственного сервиса (unit-файл, запуск, логирование).Ситуационные задачи: «система не загружается, пользователь видит... что делать?»

Файловые системы	Чем различаются физический диск, раздел и файловая система?Какие файловые системы наиболее типичны для Linux и почему?Зачем требуется монтирование и чем отличается временное монтирование от постоянного?Какова логика стандартной иерархии каталогов Linux (/ , /home, /var, /etc и др.)?Какие риски связаны с переполнением файловой системы и как их контролировать?Что такое сетевые файловые системы и какие задачи они решают в корпоративной сети?	Практическая работа за ПК (просмотр таблицы разделов, монтирование/размонтирование).Задания на анализ структуры каталогов ОС.Разбор примеров /etc/fstab и обсуждение вариантов автоматического монтирования.Мини-коллоквиум по терминам и понятиям файловых систем.
Основы скриптов на bash	В чём отличие одиночной команды от сценария (скрипта)?Как устроен цикл обработки параметров командной строки в bash-скрипте?Какие типичные конструкции используются для проверки условий и организации циклов?Почему важно анализировать коды возврата команд (\$?)?Когда целесообразно применять sleep вместо ручного запуска скрипта?Какие требования к читаемости и документированию скриптов оправданы в учебных и реальных проектах?	Практическая работа по написанию и отладке простых скриптов.Работа в малых группах: разработка сценария для автоматизации конкретной задачи.Демонстрация настройки sleep-заданий.Обсуждение и взаимная рецензия скриптов студентов (код-ревью в аудитории).
Средства обработки текста	Чем потоковая обработка текста отличается от работы в обычном текстовом редакторе?В каких задачах sed и awk оказываются незаменимыми инструментами?Что такое регулярные выражения и почему они так важны для администрирования?Как организовать цепочку команд для анализа журналов (логов) сервисов?Какие типичные шаблоны автоматизации обработки конфигурационных файлов применяются на практике?Какие ошибки чаще всего возникают при неправильном использовании регулярных выражений?	Практическая работа по обработке логов и конфигурационных файлов.Решение задач на составление регулярных выражений.Мини-проект: написание небольшого «репортера логов» на bash + sed/awk.Фронтальный разбор типичных ошибок и «ловушек» регулярных выражений.
Настройка сети в Linux	Чем отличаются модель OSI и стек TCP/IP, как они используются на практике?Как понять, на каком уровне сети возникла проблема связи?Какие команды в Linux наиболее полезны для	Лабораторная работа по настройке сетевых интерфейсов и маршрутизации.Модел

	диагностики сетевых неполадок?Как реализуется статическая и динамическая адресация в локальной сети?Для чего нужны различные режимы сетей в VirtualBox (NAT, мост, host-only)? Какие базовые меры безопасности необходимо соблюдать при настройке сетевых сервисов?	ирование клиент-серверных конфигураций на виртуальных машинах.Решение ситуационных задач: «клиент не видит сервер, что проверяем по шагам?»Мини-тест по командам диагностики сети.
Средства ОС в Python	Какие преимущества даёт использование Python поверх стандартных средств ОС?В чём различия между os. path и pathlib при работе с путями? Какие риски возникают при запуске внешних команд через subprocess?Как корректно обрабатывать ошибки при файловых операциях из Python-программ?Какие задачи администрирования удобно автоматизировать с помощью Python вместо bash?Как обеспечить переносимость таких утилит между Linux и Windows?	Практическая работа по написанию утилит на Python для работы с файлами и каталогами.Задания на использование subprocess для вызова внешних команд.Обсуждение примеров кроссплатформенных решений (с демонстрацией).Мини-коллоквиум по стандартным модулям os, pathlib, subprocess.
Системы контроля версий	Зачем вообще нужны системы контроля версий, какие проблемы они решают?Что такое коммит и почему важно «правильно» формулировать комментарии к нему?Как организовать работу с ветками в небольшом учебном проекте?В чём суть конфликтов при слиянии и как их разрешать? Какую роль играют удалённые репозитории (GitHub и др.) в командной работе?Какие принципы рабочего процесса (workflow) стоит использовать в учебных проектах по сетевым приложениям?	Практическая работа по инициализации репозитория, выполнению коммитов и работе с ветками.Моделирование командной работы: разделение ролей и совместная доработка проекта.Разбор примеров конфликтов слияния и их ручное разрешение.Визуальная демонстрация истории репозитория в графическом клиенте или IDE.
Использование сокетов	Что такое сокет и как он связан с транспортным уровнем сети?В чём различия между TCP и UDP и какие приложения лучше использовать каждый из них?Как выглядит типичная схема работы TCP-	Лабораторная работа по написанию TCP/UDP-клиента и сервера на

	клиента и TCP-сервера?Какие типичные ошибки допускают начинающие при программировании сети (блокировки, забытый close, отсутствие таймаутов)?Как тестировать и отлаживать сетевое приложение в виртуальной среде?Какие простейшие протоколы прикладного уровня можно реализовать в учебных проектах?	Python.Совместный разбор примеров протоколов «запрос–ответ».Тестирование приложений между виртуальными машинами (моделирование сети).Мини-коллоквиум по ключевым понятиям сокетов и протоколов.
Асинхронное программирование	В чём различие между параллелизмом и конкурентностью?Когда достаточно многопоточности, а когда разумнее использовать многопроцессность?Какие преимущества даёт асинхронный подход (asynсio) для сетевых приложений?Что такое гонка данных и взаимоблокировка, как их избегать?Как выбирать модель параллелизма под конкретный тип задач (CPU-bound, IO-bound)?Какие практические ограничения асинхронного подхода важно помнить?	Практическая работа по разработке многопоточного или асинхронного сервера.Анализ и обсуждение примеров кода с ошибками конкурентности.Обсуждение в малых группах вариантов архитектуры для разных типов задач.Мини-тест по ключевым понятиям многопоточности и asynсio.
Веб-сервер	Как устроена клиент-серверная модель web, какие компоненты в неё входят?Что такое HTTP-запрос и HTTP-ответ, какие у них обязательные элементы?Какую роль играет веб-сервер (nginx/Apache) между клиентом и приложением?Для чего нужны виртуальные хосты и как их использовать на одном сервере?Какие элементы настройки веб-сервера важны для безопасности и производительности?Как организовать журналирование и анализ логов веб-сервера для сопровождения приложений?	Практическая работа по установке и базовой настройке веб-сервера на Linux.Создание и тестирование простого сайта/страницы на учебном сервере.Разбор конфигурационных файлов виртуальных хостов.Анализ и обсуждение фрагментов логов веб-сервера (поиск ошибок и аномалий).
Развёртывание сетевых приложений	Какие этапы жизненного цикла сетевого приложения можно выделить от разработки до эксплуатации?Чем тестовая среда отличается от рабочей и зачем нужна каждая из них?Какие	Лабораторная работа по развёртыванию учебного сетевого приложения на

	типичные ошибки допускаются при ручном развёртывании приложений?Как организовать структуру каталогов и прав доступа для серверного приложения?Какие минимальные механизмы мониторинга следует реализовать для отслеживания состояния сервиса?Как использовать скрипты и простые инструменты автоматизации при обновлении приложений?	сервере.Проработка мини-кейса: «обновление версии сервиса» (обсуждение стратегии).Написание вспомогательных скриптов развёртывания (bash/Python).Защита мини-проекта: краткий отчёт по развёртыванию собственного учебного приложения.
--	---	---

6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Введение в ОС Linux	Классификация ОС: настольные, серверные, мобильные, встроенные. Сравнение популярных дистрибутивов Linux (Ubuntu, Debian, CentOS и др.). Лицензии свободного ПО (GPL, MIT и др.) и их влияние на использование ОС. Обзор областей применения Linux в индустрии (серверы, встраиваемые системы, облака). Сравнение модели безопасности Linux и Windows на концептуальном уровне.	Подбор и конспектирование материалов из учебников и онлайн-документации. Самостоятельная установка дистрибутива Linux в VirtualBox. Подготовка краткого обзора выбранного дистрибутива (1–2 стр.). Просмотр и конспектирование обучающих видеоматериалов по истории Linux.
Основы командной строки	Расширенные возможности команд ls, cp, mv, rm, find. Метасимволы и подстановки оболочки (globbing). Перенаправление ввода-вывода и конвейеры (pipe). Использование man, info и онлайн-справочников. Типовые «односложные» команды администратора для ежедневных задач.	Решение набора тренировочных задач по работе с файлами и каталогами. Ведение «дневника команд» (запись новых команд и примеров использования). Самостоятельное выполнение мини-чек-листа по резервному копированию каталога. Подготовка шпаргалки по часто используемым командам (1–2 стр.).
Использование удалённого доступа	Форматы ключей SSH, понятия authorized_keys, known_hosts. Типичные настройки SSH-сервера (sshd_config) для повышения безопасности. Отличия SSH-порта по умолчанию и нестандартного порта. Обзор альтернативных	Настройка SSH-доступа по ключу к собственной ВМ. Выполнение серии команд

	клиентов SSH для разных платформ. Подходы к организации туннелирования трафика через SSH (общий обзор).	администрирования на удалённой машине и фиксация результатов. Поиск и конспектирование рекомендаций по «best practices» настройки SSH. Подготовка краткого отчёта о проделанной работе (скриншоты, команды, выводы).
Права и пользователи	Файлы конфигурации учётных записей (/etc/passwd, /etc/shadow, /etc/group). Маска прав по умолчанию (umask) и её влияние на создаваемые файлы. Специальные биты SUID, SGID и sticky bit: назначение и примеры. Сравнение модели прав доступа в Linux и Windows. Практические подходы к проектированию групп пользователей в проекте.	Самостоятельное создание нескольких пользователей и групп на ВМ. Настройка разных схем прав доступа к общему каталогу. Документирование проделанных операций (команды, объяснение получившихся прав). Анализ примеров уязвимостей, связанных с неверной настройкой прав (по материалам из открытых источников).
Процессы и пакеты	Подробное изучение формата вывода ps, top, htop. Понятие «демон» и фоновый процесс. Отличия разных систем инициализации (общий обзор). Особенности работы популярных менеджеров пакетов (apt, yum/dnf, pacman — обзорно). Обзор форматов пакетов и их структуры (deb, rpm и др.).	Самостоятельный анализ списка процессов и выделение ключевых системных служб. Установка и удаление нескольких пакетов, документирование зависимостей. Поиск и конспектирование официальной документации по выбранному менеджеру пакетов. Подготовка мини-отчёта о сравнении двух

		менеджеров пакетов (по выбору студента).
Управление загрузкой Linux	Структура конфигурации GRUB, примеры параметров ядра. Отличия загрузки в текстовый и графический режим. Системные журналы (logs), связанные с этапом загрузки. Сравнение подходов к управлению службами в разных системах инициализации. Обзор типичных ошибок в конфигурации загрузчика и способов их устранения.	Изучение содержимого конфигурации загрузчика на собственной ВМ (только просмотр!). Самостоятельный анализ логов загрузки (journalctl, лог-файлы). Конспектирование теоретического материала по этапам загрузки ОС. Подготовка краткого описания своей системной конфигурации (сервисы, режим загрузки).
Файловые системы	Обзор файловых систем ext4, XFS, Btrfs (назначение, особенности). Понятие inode и его роль в файловой системе. Журнальные файловые системы и их преимущества. Основные утилиты проверки и исправления файловых систем. Обзор технологий RAID (на концептуальном уровне).	Изучение параметров смонтированных файловых систем на ВМ (df, mount). Конспектирование структуры стандартной иерархии каталогов Linux. Подготовка схемы (диаграммы) типичного разбиения диска под Linux. Поиск информации о базовых возможностях сетевых файловых систем (NFS, SMB — обзорно).
Основы скриптов на bash	Подробный синтаксис условных операторов (if, case). Циклы (for, while, until) и их типичные паттерны использования. Массивы в bash, особенности работы с ними. Организация функций, область видимости переменных. Практика обработки ошибок и диагностики в скриптах.	Разработка набора небольших скриптов для автоматизации бытовых задач (очистка каталога, резервное копирование и т.п.). Ведение репозитория своих

		скриптов (в дальнейшем можно связать с git). Конспектирование примеров из учебников и документации по bash. Самостоятельная настройка одного–двух cron-заданий и описания их назначения.
Средства обработки текста	Расширенный синтаксис sed (многострочные шаблоны, скрипты). Базовый и расширенный синтаксис регулярных выражений. Основы awk как языка обработки текстовых таблиц. Концепция потоковой обработки данных в Unix-подобных системах. Сравнение различных подходов обработки логов (bash+sed/awk, Python и др.).	Написание серии команд sed/awk для анализа учебного набора логов. Подготовка шпаргалки по регулярным выражениям. Самостоятельное решение задач на выборку строк из файлов по сложным условиям. Документирование «рецептов» обработки текстов (чек-лист с примерами).
Настройка сети в Linux	Подробный разбор основных протоколов стека TCP/IP (на уровне концепций). Статическая маршрутизация и понятие «шлюз по умолчанию». DNS: назначение, базовые принципы работы и типы записей. Основы NAT и его роль в локальных сетях. Обзор типичных сетевых утилит (ping, traceroute, netstat/ss и др.).	Самостоятельная настройка сетевых параметров на ВМ (в рамках домашней сети/виртуальной среды). Анализ таблицы маршрутизации и DNS-настроек собственной системы. Конспектирование теоретических материалов по модели OSI и TCP/IP. Подготовка схемы (блок-схемы) типичной локальной сети с несколькими хостами.
Средства ОС в Python	Документация по модулям os и pathlib:	Написание простых

	расширенные возможности. Безопасное построение путей и работа с различными файловыми системами. Варианты использования subprocess (run, Popen и др.). Обработка исключений при системных вызовах из Python. Понятие логирования в Python и его роль в системных утилитах.	Python-скриптов для обхода каталогов и фильтрации файлов.Разработка утилиты, вызывающей внешнюю команду и анализирующей её результат.Ведение мини-проекта по автоматизации одной системной задачи (с сохранением версий).Конспектирование полезных фрагментов кода и шаблонов (snippet-коллекция).
Системы контроля версий	Структура внутренних объектов git (коммит, дерево, blob — обзорно). Различие между локальными и удалёнными ветками. Типовые workflows (GitFlow, GitHub Flow — на базовом уровне). Практики оформления сообщений коммитов и ведения истории. Общие принципы code review и работы с pull request.	Создание личного репозитория и регулярное ведение истории изменений небольшого проекта.Практика работы с ветками: создание, слияние, откат.Работа с удалённым репозиторием (GitHub или аналог): push/pull, clone, fork.Подготовка краткого «руководства для новичка» по git (внутригрупповой документ).
Использование сокетов	Подробный разбор API модуля socket в Python. Типовые шаблоны обмена данными в протоколах «запрос–ответ». Основы обработки ошибок в сетевых приложениях. Примеры простых текстовых протоколов прикладного уровня. Обзор средств для отладки сети (telnet, nc/netcat, tcpdump — на уровне знакомства).	Самостоятельная реализация простого ТСП-клиента и сервера по заданным требованиям.Эксперименты с изменением формата сообщений и наблюдением реакции сервера.Документирование собственного «мини-протокола» (описание команд и формата данных).Анализ поведения

		приложений при обрыве связи и некорректном завершении соединения.
Асинхронное программирование	Подробное изучение интерфейса threading и multiprocessing. Основа работы цикла событий (event loop) в asyncio. Прimitives синхронизации (Lock, Queue, Event и др.). Паттерны обработки множества одновременных запросов. Анализ примеров типичных ошибок конкурентности и способов их устранения.	Реализация небольшого многопоточного или асинхронного примера (например, параллельная загрузка нескольких URL). Эксперименты с разными моделями параллелизма для одной и той же задачи. Конспектирование найденных в документации и статьях «best practices» по asyncio. Подготовка краткого аналитического отчёта: «что лучше для моей задачи — потоки, процессы или asyncio?»
Веб-сервер	Структура HTTP-запроса и ответа: методы, заголовки, коды состояния. Основные директивы конфигурации nginx/Apache (на уровне простых сайтов). Принципы организации виртуальных хостов. Простые механизмы ограничения доступа к ресурсам (basic auth и т. п.). Обзор средств анализа логов веб-сервера.	Самостоятельная настройка простого сайта на локальном веб-сервере. Анализ логов веб-сервера по учебному сценарию (поиск ошибок, статистика запросов). Конспектирование основных директив конфигурации, используемых в практикуме. Подготовка небольшой инструкции по развертыванию простого сайта для однокурсника.
Развёртывание сетевых приложений	Общая структура типичного серверного приложения (front-end, back-end, БД — обзорно).	Самостоятельное развёртывание

	Стратегии обновления приложений (пошаговое, «синий/зелёный» деплой — обзорно). Принципы ведения логов приложения и системных логов. Основы мониторинга (ключевые метрики, пороговые значения). Обзор простых инструментов автоматизации развёртывания (скрипты, Makefile, простые CI-сценарии — на уровне знакомства).	учебного сетевого приложения на своей ВМ. Разработка и документирование простого скрипта обновления приложения. Подготовка файла с чек-листом шагов развёртывания и отката версии. Анализ логов и ресурсов системы после запуска приложения (нагрузка, ошибки, предупреждения).
--	---	--

6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю

4 семестр

Примерные вопросы контрольной работы

1. Структура файловой системы Linux. Назначение каталогов /etc, /var, /home, /bin.
2. Права доступа в Linux: чтение, запись, выполнение. Битовая маска прав (chmod).
3. Понятие процесса. Команды ps, top, kill. Жизненный цикл процесса.
4. Потоки ввода-вывода (stdin, stdout, stderr) и их перенаправление. Конвейеры (pipes).
5. Основные конструкции языка Bash: переменные, циклы, условия.
6. Утилиты обработки текста: grep, sed, awk. Основные принципы работы.
7. Загрузчик GRUB и процесс инициализации системы (systemd).
8. Монтирование файловых систем. Файл /etc/fstab.

Примерные задания контрольной работы

1. Одной командой найти все файлы с расширением .log в папке /var, содержащие слово "error", и сохранить список в файл.
2. Написать скрипт, который проверяет наличие пользователя в системе и, если его нет, создает его.
3. Настроить права доступа к папке так, чтобы владелец мог всё, группа только читать, а остальные не имели доступа.
4. С помощью cron настроить выполнение скрипта очистки временных файлов каждую ночь в 03:00.

5 семестр

Примерные вопросы контрольной работы

1. Модель OSI и стек TCP/IP. Различия протоколов TCP и UDP.
2. Понятие сокета. Клиент-серверная архитектура.

3. Работа с модулями os, sys, subprocess в Python.
4. Принципы многопоточного и асинхронного программирования (threading, asyncio).
5. Протокол HTTP: структура запроса и ответа, методы, коды состояния.
6. Веб-серверы (Nginx/Apache): назначение, роль обратного прокси.
7. Системы контроля версий Git: основные команды (commit, push, pull, branch).
8. Стратегии развертывания приложений (Deployment).

Примерные задания контрольной работы

1. Написать на Python TCP-сервер, который принимает сообщение от клиента, переворачивает его и отправляет обратно.
2. Написать скрипт на Python, который параллельно проверяет доступность списка IP-адресов (ping).
3. Настроить конфигурацию Nginx для раздачи статических файлов из определенной папки на порту 8080.
4. Выполнить слияние двух веток в Git с разрешением возникшего конфликта.

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях Кафедры информационных технологий Факультета информационных технологий и анализа больших данных.

7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Перечень компетенций с указанием индикаторов их достижения в процессе освоения образовательной программы содержится в разделе 2. *Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине.*

Типовые контрольные задания или иные материалы, необходимые для оценки индикаторов достижения компетенций, умений и знаний

ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем

1) Устанавливает простое программное обеспечение для информационных и автоматизированных систем

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: основные типы дистрибутивов Linux, принципы работы менеджеров пакетов (apt, dnf), структуру репозитория ПО, требования к аппаратному обеспечению для развертывания ОС

Уметь: производить установку операционной системы Linux на виртуальную или физическую машину, выполнять базовую постинсталляционную настройку, устанавливать и обновлять пакеты через консоль

Типовые контрольные задания

Развернуть виртуальную машину с ОС Linux (Ubuntu/Debian/CentOS) в VirtualBox.

Установить набор базовых утилит (git, curl, vim) используя пакетный менеджер.

2) Настраивает программное и аппаратное обеспечение для решения конкретных задач предметной области

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: конфигурационные файлы основных системных служб (SSH, Cron, Systemd), принципы настройки сетевых интерфейсов в Linux, механизм прав доступа и управления пользователями

Уметь: настраивать сетевые подключения (статический IP, DNS), конфигурировать службу удаленного доступа SSH (запрет root, вход по ключам), управлять автозапуском служб

Типовые контрольные задания

Настроить доступ по SSH к серверу только по ключу с запретом входа по паролю.

Создать службу systemd для автоматического запуска пользовательского скрипта при старте системы.

3) Проводит анализ информационной инфраструктуры, выявляет слабые места, вырабатывает рекомендации для ее улучшения

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: средства мониторинга системных ресурсов (top, htop, vmstat), структуру и расположение системных журналов (logs), базовые принципы анализа производительности и безопасности системы

Уметь: анализировать логи системных служб для выявления ошибок, диагностировать узкие места в производительности (CPU, RAM, Disk I/O), проверять целостность файловой системы

Типовые контрольные задания

С помощью утилит мониторинга определить процесс, потребляющий наибольшее количество памяти, и корректно завершить его.

Написать отчет по анализу логов веб-сервера за определенный период с выявлением кодов ошибок 4xx/5xx.

ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов

1) Разрабатывает алгоритмы решения простых информационных задач и выражает их на языке программирования

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: синтаксис командной оболочки Bash, основы языка Python (модули os, sys, socket), основные алгоритмы работы с файловой системой и текстовыми потоками

Уметь: писать скрипты на Bash для автоматизации рутинных задач (резервное копирование, обработка текста), разрабатывать простые сетевые приложения на Python (клиент-сервер)

Типовые контрольные задания

Написать Bash-скрипт для автоматического резервного копирования указанной директории в архив с текущей датой в имени.

Реализовать на Python простой TCP-клиент, отправляющий строку серверу и получающий ответ.

2) Анализирует алгоритмы в части производительности, оптимальности, вырабатывает рекомендации для оптимизации алгоритмов программ

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: отличия синхронного и асинхронного выполнения кода, понятия многопоточности и многопроцессности (threading vs multiprocessing), влияние блокирующих операций ввода-вывода на производительность

Уметь: выбирать подходящую модель конкурентности (потoki, процессы, async/await) для решения задачи, оптимизировать скрипты обработки данных с использованием конвейеров

Типовые контрольные задания

Сравнить скорость выполнения задачи скачивания 100 файлов последовательно и с использованием пула потоков.

Оптимизировать Bash-скрипт обработки большого лог-файла, заменив циклы на использование утилит grep/sed/awk.

3) Проводит ручное и автоматизированное тестирование программных продуктов по методам черного и белого ящика, составляет набор тестовых случаев

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: методы отладки сетевых приложений, принципы модульного тестирования кода на Python, инструменты для проверки доступности сетевых узлов и сервисов

Уметь: проводить ручное тестирование развернутых веб-сервисов (проверка кодов ответа, доступности контента), писать простейшие unit-тесты для функций на Python

Типовые контрольные задания

Составить чек-лист для проверки работоспособности развернутого веб-сервера (доступность порта, корректность отдачи статики, права доступа).

Написать тест на Python, проверяющий, что функция парсинга логов корректно обрабатывает некорректные входные данные.

Примеры практико-ориентированных заданий

4 семестр

Задание 1. Администрирование пользователей.

Создать скрипт на Bash, который принимает на вход имя пользователя. Если пользователь существует, скрипт выводит список групп, в которых он состоит. Если нет — создает пользователя, задает ему случайный пароль и сохраняет пароль в файл.

Задание 2. Анализ логов.

Дана директория с лог-файлами. Написать однострочную команду (конвейер), которая подсчитает количество уникальных IP-адресов, встречающихся в файлах, и выведет топ-5 самых активных.

Задание 3. Управление процессами.

Запустить процесс, который создает высокую нагрузку на процессор. Найти его PID с помощью консольных утилит, изменить его приоритет (nice), а затем корректно завершить процесс.

5 семестр

Задание 1. Сетевой сканер на Python

Написать утилиту на Python, которая сканирует заданный диапазон портов на указанном хосте и выводит список открытых портов. Использовать модуль socket.

Задание 2. Асинхронный веб-сервер.

Реализовать простейший HTTP-сервер на базе asyncio, который на любой GET-запрос возвращает JSON с текущим временем сервера.

Задание 3. Деплой веб-приложения.

Дано простое веб-приложение на Python. Необходимо:

1. Подготовить requirements.txt.
2. Написать Systemd-unit для запуска приложения как службы.
3. Настроить Nginx как обратный прокси, перенаправляющий запросы с 80 порта на порт приложения.

Примерные вопросы для подготовки к зачету, экзамену

Примерные вопросы для подготовки к зачету

1. Архитектура ОС Linux: ядро, оболочка, файловая система.
2. Типы файлов в Linux. Жёсткие и символические ссылки.
3. Управление пользователями и группами. Команды useradd, usermod, groupadd.
4. Права доступа (RWX). Команды chmod, chown. Специальные биты (SUID, Sticky).
5. Управление процессами в Linux. Сигналы.
6. Потоки ввода-вывода и конвейеры.
7. Переменные окружения. Настройка профиля пользователя (.bashrc).
8. Основы написания скриптов на Bash: циклы, условия, аргументы.
9. Управление пакетами (apt/dnf) и репозиториями.
10. Планировщик задач Cron.

Примерные вопросы для подготовки к экзамену

1. Стек протоколов TCP/IP. Адресация, порты, маршрутизация.
2. Протокол SSH: архитектура, настройка, ключи шифрования.
3. Работа с файловой системой и процессами из Python (модули os, subprocess).
4. Сетевое программирование: сокеты, блокирующий и неблокирующий ввод-вывод.
5. Многопоточность и многопроцессность в Python (GIL, threading, multiprocessing).
6. Асинхронное программирование (asyncio, event loop).
7. Протокол HTTP: методы, заголовки, cookies, сессии.
8. Веб-серверы Nginx и Apache: архитектура, конфигурация виртуальных хостов.
9. Системы контроля версий Git: устройство репозитория, работа с ветками.
10. Методологии развертывания ПО (Deployment, CI/CD - основы).

**Промежуточная аттестация обучающихся проводится в формате тестирования*

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

Основная литература:

1. Малахов, С. В. Принципы работы операционной системы Linux. Bash-скрипты [Электронный ресурс] : учебное пособие / Малахов С. В., Якупов Д. О. Самара : ПГУТИ, 2024 134 с. Книга из коллекции ПГУТИ - Информатика <https://e.lanbook.com/book/463574> ISBN 978-5-907336-50-6. [БИК ID: RU-LAN-BOOK-463574]
2. Коротеев, М.В. Сетевое программирование на Python в Linux : учебное пособие по дисциплине "Машинное обучение" для студентов, обучающихся по направлению: "Прикладная информатика", всех профилей (программы подготовки бакалавров), институт онлайн-образования / М.В. Коротеев, Д.А. Петросов ; Финуниверситет, Департамент анализа данных и машинного обучения факультета информационных технологий и анализа больших данных Москва : Финуниверситет, 2023 1 файл (2,01 Мб) : ил. Доступ из локальной сети Финуниверситета (чтение) (дата обращения 16.10.2023) + Текст : электронный. [БИК ID: RU\FA\books\137316]
3. Кудрявцев, Н. Г. Основы работы в ОС Linux. Начальное конфигурирование и администрирование [Электронный ресурс] : учебное пособие / Кудрявцев Н. Г., Фролов И. Н. Горно-Алтайск : ГАГУ, 2022 108 с. Книга из коллекции ГАГУ - Информатика <https://e.lanbook.com/book/271097>. [БИК ID: RU-LAN-BOOK-271097]

Дополнительная литература:

1. Основы администрирования и системного программирования в операционной системе Linux / Елисеев А. И., Яковлев А. В., Дерябин А. С. Ч. 2 : Основы администрирования и системного программирования в операционной системе Linux. В 2-х ч. Ч. 2 : учебное пособие . Ч. 2 / Елисеев А. И., Яковлев А. В., Дерябин А. С. Тамбов : ТГТУ, 2021 84 с. Книга из коллекции ТГТУ - Информатика <https://e.lanbook.com/book/320438> ISBN 978-5-8265-2437-4. [БИК ID: RU-LAN-BOOK-320438]
2. Гунько, Андрей Васильевич Системное программирование в среде Linux : Учебное пособие / Новосибирский государственный технический университет Новосибирск : Новосибирский государственный технический университет

(НГТУ), 2020 235 с. ВО - Бакалавриат <https://znanium.com/catalog/document?id=398058> ISBN 978-5-7782-4160-2. [БИК ID: RU\infra-m\znanium\bibl\1870577]

3. Бобровский, В. И. Расширенное администрирование сетевой операционной системы GNU/Linux. Локальное системное администрирование [Электронный ресурс] : учебное пособие / Бобровский В. И., Дагаев А. В., Журавель Е. П. Санкт-Петербург : СПбГУТ им. М.А. Бонч-Бруевича, 2022 138 с. Книга из коллекции СПбГУТ им. М.А. Бонч-Бруевича - Информатика <https://e.lanbook.com/book/279176> ISBN 978-5-89160-252-6. [БИК ID: RU-LAN-BOOK-279176]

4. Поводайко, М. Д. Основы операционной системы LINUX [Электронный ресурс] : учебно-методическое пособие по выполнению практических и лабораторных работ / Поводайко М. Д., Гвоздков И. В., Андрианова Е. Е., Денисова Ю. В. Санкт-Петербург : СПбГУТ им. М.А. Бонч-Бруевича, 2023 107 с. Книга из коллекции СПбГУТ им. М.А. Бонч-Бруевича - Информатика <https://e.lanbook.com/book/426020>. [БИК ID: RU-LAN-BOOK-426020]

5. Тормозов, В. С. Основы работы в операционной системе Linux [Электронный ресурс] : учебное пособие / Тормозов В. С., Золкин А. Л. Самара, 2023 66 с. Книга из коллекции - Медицина <https://e.lanbook.com/book/388838> ISBN 978-5-907359-20-8. [БИК ID: RU-LAN-BOOK-388838]

6. Основы администрирования и системного программирования в операционной системе Linux / Елисеев А. И., Яковлев А. В., Дерябин А. С. Ч. 1 : Основы администрирования и системного программирования в операционной системе Linux: в 2-х ч. Ч. I . Ч. 1 / Елисеев А. И., Яковлев А. В., Дерябин А. С. Тамбов : ТГТУ, 2020 82 с. Утверждено Учёным советом университета в качестве учебного пособия для студентов 3–4 курсов направления подготовки 09.03.02 «Информационные системы и технологии» специальности 10.05.03 «Информационная безопасность автоматизированных систем» очной формы обучения Книга из коллекции ТГТУ - Информатика <https://e.lanbook.com/book/320294> ISBN 978-5-8265-2248-6. [БИК ID: RU-LAN-BOOK-320294]

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/>
(<http://library.fa.ru/files/elibfa.pdf>)
2. Электронно-библиотечная система BOOK.RU <http://www.book.ru>
3. Электронно-библиотечная система "Университетская библиотека ОНЛАЙН" <http://biblioclub.ru/>
4. Электронно-библиотечная система издательства "Лань" <https://e.lanbook.com/>
5. Образовательная платформа "ЮРАЙТ" <https://urait.ru/>
6. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>
7. Информационно-образовательный портал Финуниверситета: <https://org.fa.ru>
8. 6.Российская государственная библиотека: <http://rsl.ru/>;
9. •Электронно-библиотечная система Znanium <http://www.znaniy.ru/>
10. •Электронно-библиотечная система издательства «ЮРАЙТ» <https://urait.ru/>
11. Издательство «Открытые системы» <http://www.osp.ru>

10. Методические указания для обучающихся по освоению дисциплины

Изучение дисциплины «Сетевые системы и приложения» предполагает сочетание теоретической подготовки, практических (лабораторных) занятий и значительного объёма внеаудиторной самостоятельной работы. Для успешного освоения курса необходимо систематически опираться на рабочую программу дисциплины, материалы лекций, практических занятий, а также основную и дополнительную литературу, указанные в соответствующих разделах рабочей программы.

10.1. Работа с теоретическим материалом (лекции, теория к лабораторным)

1. Ориентация на рабочую программу и план занятий.

Перед началом семестра рекомендуется ознакомиться со структурой дисциплины, перечнем тем и учебно-тематическим планом. Это позволит планировать нагрузку и заранее понимать, какие разделы связаны между собой (например, темы по работе с ОС Linux, настройке сети, Python, системам контроля версий, веб-серверу и развёртыванию приложений образуют единое сквозное содержание по сетевым системам и приложениям).

1. Подготовка к лекции

- За 1–2 дня до занятия просмотрите формулировку темы и краткое содержание соответствующего раздела рабочей программы.
- По рекомендованным учебникам и учебным пособиям (раздел 8 РПД) найдите главы, соответствующие предстоящей теме (например, основы операционных систем, компьютерных сетей, Linux, Python, сетевого программирования) и выберите ключевые понятия, которые вам пока непонятны.
- Составьте краткий список вопросов, которые хотелось бы прояснить на лекции (например, «чем файловая система отличается от раздела диска», «в чём разница между TCP и UDP», «как работает система прав в Linux» и т.п.).

2. Ведение конспекта во время лекций

- выделять названия тем и подтем (подчёркиванием, цветом, нумерованными заголовками);
- фиксировать основные определения (процесс, файл, сокет, репозиторий, ветка, сервис, виртуальная машина и т.д.);

- записывать ключевые схемы (структура ОС, путь сетевого пакета, схема работы веб-сервера, этапы развёртывания приложения);
- приводить разобранные на лекции примеры команд и небольших программ (команды терминала, фрагменты bash-скриптов, примеры кода на Python, команды git).
- Лекции следует обязательно конспектировать. В конспекте рекомендуется:
 - Важно записывать не только формальные определения, но и пояснения преподавателя, так как именно они помогают связать теорию с практикой.

3. Обработка конспекта после лекции
После занятия рекомендуется в ближайшие 1–2 суток:

- внимательно перечитать записи, исправить неточности, дополнить пропуски;
- структурировать конспект: выделить разделы, перечни команд, примеры конфигураций, рисунки схем;
- сопоставить конспект с соответствующими разделами учебников и учебных пособий, а также с материалами из рекомендуемых электронных библиотечных систем и Интернет-ресурсов (разделы 8 и 9 РПД). При расхождениях или возникновении вопросов следует обратиться к преподавателю за разъяснениями;
- выписать в отдельный список команды, понятия и определения, которые нужно выучить к текущему контролю (тесты, устный опрос) и к промежуточной аттестации.

10.2. Методические указания по участию в практических и лабораторных занятиях

Практические и лабораторные занятия по дисциплине носят деятельностный характер и проводятся в компьютерном классе с использованием ОС семейства Linux, виртуализации, сетевых сервисов, языка Python, систем контроля версий и т.п.

10.2.1. Общие рекомендации

1. Предварительная подготовка

2. Структура практических занятий:
В соответствии с исходной рабочей программой практические занятия можно условно разделить на учебные и контрольные.

Учебные практические занятия включают:

Контрольные практические занятия включают:

1. Поведение студента на практическом занятии

- последовательность шагов (например, установка Linux в VirtualBox, настройка сети, запуск веб-сервера);
- использованные команды и их ключевые параметры;
- важные фрагменты кода (bash, Python);
- типичные сообщения об ошибках и способы их устранения.

2. Проблемно-деятельностный характер занятий

На практике часто создаются ситуации, близкие к реальным профессиональным задачам:

Студент должен не только воспроизвести по образцу, но и проанализировать проблему, выделить противоречие («почему не удаётся подключиться по SSH», «почему веб-сервис недоступен»), предложить возможные варианты решения, проверить гипотезы на практике, а затем обосновать выбранный подход. Такой формат обучения формирует профессиональные компетенции, связанные с администрированием и разработкой сетевых приложений.

10.3. Организация самостоятельной работы обучающихся

Существенная часть трудоёмкости дисциплины приходится на самостоятельную работу

1. Общие рекомендации

- изучить теоретический материал по основной и дополнительной литературе;
- выполнить набор типовых практических заданий (настройка системы, написание скриптов и программ, работа с git и т.д.);
- подготовиться к тестированию и устному опросу.

2. Ведение рабочей тетради (или электронного журнала)

Рекомендуется:

- краткие конспекты по темам;
- решения задач и лабораторных работ (включая команды, скрипты, фрагменты кода);
- выводы и замечания по результатам выполненных практических заданий.

3. Регулярность выполнения домашних заданий

4. Использование информационных ресурсов

Для самостоятельной работы следует активно использовать:

При обращении к Интернет-ресурсам важно сверять найденную информацию с рекомендованной литературой и материалами лекций.

10.4. Работа с программным обеспечением и технической средой

1. Виртуальные машины и Linux
2. Использование Python, git и веб-серверов
3. Соблюдение требований информационной безопасности

10.5. Рекомендации по подготовке к текущему и промежуточному контролю

1. Текущий контроль (тесты, опросы, проверка лабораторных работ)

- повторите основные понятия, определения, схемы и команды;
- перечитайте конспект и учебник по соответствующей теме;
- убедитесь, что выполнены все лабораторные работы и домашние задания, а их результаты понятны (вы можете объяснить, почему делали именно так, а не иначе).
- пониманию логики работы ОС Linux и сетевого стека;
- правильному использованию команд и ключей (без механического заучивания, а с пониманием эффектов);
- интерпретации результатов (выводы команд, логи, сообщения об ошибках).

2. Промежуточная аттестация (зачёт, экзамен)

10.6. Академическая честность и стиль работы

1. Самостоятельность выполнения работ

2. Работа с источниками

Следование данным методическим указаниям, регулярная работа с теорией и практическими заданиями, использование рекомендованной литературы и электронных ресурсов, а также своевременное обращение за консультацией к преподавателю при возникновении сложностей обеспечат успешное освоение дисциплины «Сетевые системы и приложения» и формирование компетенций, предусмотренных рабочей программой.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем

Комплект лицензионного программного обеспечения:

1. Python 3.8
2. Операционная система
3. Текстовый редактор
4. Пакет офисных программ
5. Виртуальные машины
6. Редактор кода VS CODE
7. Комплект свободно распространяемого программного обеспечения
8. Linux
9. Kaspersky

Современные профессиональные базы данных и информационные справочные системы:

1. Информационно-правовая система «Консультант Плюс»
2. Язык программирования Python 3. <https://pythonworld.ru/>
3. Реестр УММ
4. Яндекс 360
5. ИС «Поисковая платформа»
6. Система комплексного раскрытия информации «СКРИН» - <http://www.skrin.ru/>
7. Электронная энциклопедия: <http://ru.wikipedia.org/wiki/Wiki>

Сертифицированные программные и аппаратные средства защиты информации:

1. не предусмотрены

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

1. Учебная аудитория для проведения учебных занятий, предусмотренных программой, в том числе групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенная оборудованием и техническими средствами обучения: мебель аудиторная (столы, стулья, доска аудиторная), набор демонстрационного оборудования (проектор, экран)