МИНИСТЕРСТВО ОБРАЗОВАНИЯ СВЕРДЛОВСКОЙ ОБЛАСТИ
ГАПОУ СО
«НИЖНЕТАГИЛЬСКИЙ ТОРГОВО-ЭКОНОМИЧЕСКИЙ КОЛЛЕДЖ»

Утверждаю:
Зам. директора по УР
_____ М.Г. Дмитриева
«30» июня 2026 г.

# КОМПЛЕКТ ОЦЕНОЧНЫХ СРЕДСТВ

## «Операционные системы и среды»

## (ОП.01)

по специальности
09.02.11 Разработка и управление программным обеспечением

Нижний Тагил 2026



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

**СОДЕРЖАНИЕ**

5.1 ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ.....4
5.1.1 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....4
5.1.2 ФОРМЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ.....5
5.1.3 ОПИСАНИЕ ПРОЦЕДУРЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ.....5
5.1.4 КРИТЕРИИ ОЦЕНКИ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ.....7
5.2 КОМПЛЕКТ «ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ».....10
5.2.1 Вопросы к экзамену.....10
5.2.2 Типовые практические задания.....12
5.2.3 Задания для экзамена.....13
5.3 КОМПЛЕКТ «ТЕКУЩИЙ КОНТРОЛЬ».....41
5.3.1 Задания для срезов знаний.....41
5.3.2 Тестовые задания.....41
5.3.3 Задания для проверочных, самостоятельных и контрольных работ.....44
5.3.4 Лабораторные и практические работы.....69
5.3.5 Компьютерные программы для контроля знаний студентов.....71



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

## 5.1 ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ

### 5.1.1 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Предметом оценки освоения учебной дисциплины (УД) являются умения и знания. Контроль и оценка этих дидактических единиц осуществляются с использованием следующих форм и методов:

Таблица 1 – Формы и методы контроля и оценки дидактических единиц

| Результаты обучения (освоенные умения, усвоенные знания) | Формы и методы контроля и оценки результатов обучения | Наименование оценочного средства |
|---|---|---|
| **Уметь:** | | |
| – управлять параметрами загрузки операционной системы | Оценка на практическом занятии. Экзамен. | ЛР №1, 5, 9; тест 1.2; билеты |
| – выполнять конфигурирование аппаратных устройств | Оценка выполнения лабораторных работ. Тестирование. | ЛР №1, 11, 15; тест 1.2, 2.5 |
| – управлять учётными записями, настраивать параметры рабочей среды пользователей | Оценка выполнения лабораторных работ. Контрольная работа. Экзамен. | ЛР №4, 7; КР №2; билеты |
| – управлять дисками и файловыми системами, настраивать сетевые параметры, управлять разделением ресурсов в локальной сети | Наблюдение и оценка практических работ. Контрольная работа. | ЛР №11, 13, 14; КР №2, 3 |
| – работать в командной строке Linux (Bash) и Windows (PowerShell), писать сценарии автоматизации на Bash и Python | Оценка выполнения лабораторных работ. Тестирование. Экзамен. | ЛР №2, 4, 12, 20–23; КР №5 |
| – создавать виртуальные машины и контейнеры Docker, описывать многоконтейнерные приложения | Оценка выполнения лабораторных работ. Контрольная работа. | ЛР №15–19; КР №4 |
| – читать техническую документацию и справочные страницы на русском и английском языках | Наблюдение за работой с документацией на занятиях. | ЛР №1–24 |
| – использовать российские ИИ-ассистенты с обязательной проверкой результата | Оценка выполнения лабораторной работы. Тестирование. Экзамен. | ЛР №24; доп. тесты; билеты №31–34 |
| **Знать:** | | |
| – основные понятия, функции, состав и принципы работы ОС | Тестирование. Экзамен. | Тесты 1.1–1.6; билеты |



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

| Результаты обучения (освоенные умения, усвоенные знания) | Формы и методы контроля и оценки результатов обучения | Наименование оценочного средства |
|---|---|---|
| – архитектуры современных ОС | Тестирование. Контрольная работа №1. | Тест 1.5; КР №1 |
| – особенности построения и функционирования семейств Unix и Windows | Тестирование. Оценка лабораторных работ. | ЛР №2–4; тест 1.3 |
| – принципы управления ресурсами в ОС | Тестирование. Экзамен. | Тесты 2.2, 2.5; билеты |
| – основные задачи администрирования и способы их выполнения | Оценка лабораторных работ. Контрольные работы №2, 3. | ЛР №7–14; КР №2, 3 |
| – модель прав доступа, процессы, сигналы, демоны, системный журнал; принцип наименьших привилегий | Тестирование. Контрольная работа №2. | Тесты 2.1–2.4; КР №2 |
| – принципы виртуализации и контейнеризации, сокеты, systemd | Тестирование. Контрольные работы №3, 4. | Тесты 3.1–4.6; КР №3, 4 |
| – возможности, ограничения и риски применения ИИ-ассистентов; защита данных | Тестирование. Экзамен. | Доп. тесты; билеты №31–34 |

Оценка освоения УД предусматривает использование пятибалльной системы оценки.

## 5.1.2 ФОРМЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Таблица 2 – Запланированные формы промежуточной аттестации

| № семестра | Формы промежуточной аттестации | Форма проведения |
|---|---|---|
| 3 | Экзамен | Устный ответ по билету и выполнение практического задания |

## 5.1.3 ОПИСАНИЕ ПРОЦЕДУРЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Экзамен по дисциплине проводится в 3 семестре и устанавливает уровень сформированности умений и усвоения знаний по всему курсу. К экзамену допускаются обучающиеся, выполнившие и защитившие все лабораторные работы и получившие положительные оценки за контрольные работы. Билет содержит два теоретических вопроса (по темам разных разделов курса) и практическое задание (разбор фрагмента вывода команды, конфигурационного файла, сценария или Dockerfile). Всего в банке 34 билета. Использование ИИ-ассистентов и посторонних материалов на экзамене не допускается.

**В результате освоения дисциплины студент должен уметь:**

– управлять параметрами загрузки операционной системы;



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

– выполнять конфигурирование аппаратных устройств;
– управлять учётными записями, настраивать параметры рабочей среды пользователей;
– управлять дисками и файловыми системами, настраивать сетевые параметры, управлять разделением ресурсов в локальной сети;
– работать в командной строке Linux (Bash) и Windows (PowerShell), писать и отлаживать сценарии автоматизации на Bash и Python;
– создавать и настраивать виртуальные машины и контейнеры Docker, описывать многоконтейнерные приложения средствами Docker Compose;
– читать техническую документацию и справочные страницы (man) на русском и английском языках;
– использовать российские ИИ-ассистенты (GigaChat, Алиса AI) для подготовки и анализа конфигураций и сценариев с обязательной проверкой результата и соблюдением требований информационной безопасности.

**В результате освоения дисциплины студент должен знать:**

– основные понятия, функции, состав и принципы работы операционных систем;
– архитектуры современных операционных систем;
– особенности построения и функционирования семейств операционных систем Unix и Windows;
– принципы управления ресурсами в операционной системе;
– основные задачи администрирования и способы их выполнения в изучаемых операционных системах;
– модель прав доступа, назначение процессов, сигналов, демонов, системного журнала и принцип наименьших привилегий;
– принципы виртуализации и контейнеризации, устройство сокетов, назначение systemd;
– возможности, ограничения и риски применения ИИ-ассистентов в администрировании; требования к защите персональных данных и учётных данных при работе с ними.

Количество заданий для студента: 3 (два теоретических вопроса и одно практическое задание)

Время выполнения каждого задания и максимальное время на проведение аттестации:

Вопрос 1 – 10 мин.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

Вопрос 2 – 10 мин.

Практическое задание – 15 мин.

Всего на экзамен (с подготовкой) – 45 мин. на одного обучающегося.

**Условия выполнения заданий**

Помещение: лаборатория «Программного обеспечения и сопровождения компьютерных систем» (кабинет операционных систем и сред)

Требования охраны труда: инструктаж по технике безопасности при работе с вычислительной техникой и электрооборудованием.

Оборудование:

– компьютеры по числу экзаменуемых с виртуальными машинами Linux и Windows;

– проектор;

– билеты на бумажном и электронном носителях.

Перечень справочной и нормативной литературы, разрешённой для использования:

1. Справочные страницы man (в оффлайн-режиме) – только для практического задания;
2. Памятка по командам Linux и PowerShell, выданная преподавателем.

## 5.1.4 КРИТЕРИИ ОЦЕНКИ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Таблица 3 – Критерии оценки

| Оценка | Показатели оценки |
|---|---|
| Отлично (90–100 баллов) | Студент умеет увязывать теорию с практикой (решает задачи, формулирует выводы, поясняет полученные результаты), владеет понятийным аппаратом, полно и глубоко овладел материалом, обосновывает свои суждения и даёт правильные ответы на вопросы преподавателя. |
| Хорошо (75–89 баллов) | Студент умеет увязывать теорию с практикой, владеет понятийным аппаратом, полно овладел материалом, но ответы содержат некоторые неточности и требуют уточнения и комментария со стороны преподавателя. |
| Удовлетворительно (60–74 балла) | Студент знает и понимает материал, но изложение неполное, непоследовательное, допускаются неточности в определении понятий, студент не может обосновать свои ответы на уточняющие вопросы. |
| Неудовлетворительно (менее 60 баллов) | Студент допускает ошибки в определении понятий, искажающие их смысл, беспорядочно и неуверенно излагает материал, делает ошибки в ответах на уточняющие вопросы. |

Экзамен охватывает весь курс «Операционные системы и среды» — материал всех пяти циклов (1. Основы операционных систем, 2. Linux вглубь, 3. Сети и сокеты, 4. Виртуализация и контейнеризация, 5. Автоматизация на Python). Банк состоит из 34 экзаменационных билетов; студент вытягивает один билет случайным образом.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

**Формат билета**

Каждый билет содержит три части:

1. **Вопрос 1** — теоретический вопрос по одной из тем курса.
2. **Вопрос 2** — теоретический вопрос по теме из ДРУГОГО цикла курса, отличного от Вопроса 1. Такое пересечение циклов сделано намеренно: экзамен проверяет способность удерживать в голове курс целиком, а не пересказывать одну изолированную лекцию.
3. **Практическое задание** — разбор реального фрагмента (вывод команды терминала, unit-файл, Dockerfile, `docker-compose.yml`, код на Python или Bash) с 2–3 конкретными подвопросами к нему.

**Распределение баллов**

Максимум за билет — **100 баллов**:

| Часть | Баллы |
|---|---|
| Вопрос 1 | 35 |
| Вопрос 2 | 35 |
| Практическое задание | 30 |
| **Итого** | **100** |

**Критерии оценивания теоретических вопросов (Вопрос 1, Вопрос 2)**

Каждый теоретический вопрос (35 баллов) оценивается по трём аспектам:

– **Полнота ответа** (0–15 баллов): раскрыты ли все составляющие вопроса, а не только его часть.

– **Корректность формулировок** (0–15 баллов): нет фактических ошибок, термины использованы правильно и уместно.

– **Связность и собственные примеры** (0–5 баллов): ответ демонстрирует понимание связей между темами курса (а не заучивание изолированных фактов) и, где уместно, подкреплён собственным примером, а не только пересказом лекции.

**Критерии оценивания практического задания**

Практическое задание (30 баллов) оценивается так:

– Правильный итоговый ответ на каждый подвопрос — 15 баллов суммарно.

– Верное пошаговое обоснование ответа (а не угадывание) — 10 баллов.

– Точное использование терминологии и синтаксиса курса (названия команд, флагов, директив без искажений) — 5 баллов.

**Шкала перевода суммарных баллов в оценку**

| Баллы | Оценка |
|---|---|
| 90–100 | «отлично» |
| 75–89 | «хорошо» |
| 60–74 | «удовлетворительно» |
| менее 60 | «неудовлетворительно» |



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

**Дополнительные правила**

– **Практическая часть обязательна.** Если студент набрал менее 60 баллов суммарно, но полностью и верно решил практическое задание, преподаватель вправе засчитать экзамен на «удовлетворительно» — это показывает, что студент способен применять знания на практике, даже если испытывает затруднения с теоретическими формулировками. Обратная ситуация (высокий балл за теорию при нулевом результате в практическом задании) права на повышение оценки не даёт.

– **Существенная фактическая ошибка** в теоретическом вопросе (например, перепутаны местами SIGTERM и SIGKILL, или RUN и CMD в Dockerfile) снижает оценку за этот вопрос не менее чем на треть, даже если остальная часть ответа полна.

– **Списывание и использование посторонних материалов** во время экзамена — основание для аннулирования результата и повторной сдачи в порядке, установленном деканатом/учебной частью.

– Если ответ на Вопрос 1 или Вопрос 2 не по теме билета вовсе (перепутана тема), но по существу верен и относится к материалу курса, преподаватель вправе засчитать его на общих основаниях, зафиксировав это в ведомости — итоговая цель экзамена — проверить владение материалом курса, а не формальное соответствие вытянутому билету при добросовестной ошибке студента в понимании вопроса.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

## 5.2 КОМПЛЕКТ «ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ»

### 5.2.1 Вопросы к экзамену

1. Понятие операционной системы. ОС как расширенная машина и как менеджер ресурсов. Функции ОС.
2. Эволюция операционных систем. Виды ОС. Отечественные операционные системы.
3. Режимы процессора: пользовательский режим и режим ядра. Роль прерываний.
4. Состав аппаратного обеспечения ПК и его взаимодействие с ОС. UEFI и этапы загрузки системы.
5. Семейства ОС Unix и Windows: сходства, различия, области применения.
6. Процесс и поток. Состояния процесса. Создание и завершение процессов.
7. Системные вызовы: назначение, механизм TRAP. Связка fork() + exec() + wait().
8. Структура операционных систем: монолитное ядро, микроядро, гибридные архитектуры. Модули ядра Linux.
9. Язык Си в системном программировании: указатели, макросы, компиляция.
10. Командные оболочки Bash и PowerShell: сравнение, основные возможности.
11. Администрирование Windows: учётные записи, службы, журнал событий, PowerShell.
12. Учётные записи и группы Linux: UID, GID, /etc/passwd, /etc/group, sudo.
13. Права доступа в Linux: символьная и восьмеричная запись, chmod, chown, umask.
14. Специальные биты SUID, SGID и sticky bit. Смысл бита x для файла и каталога.
15. Жизненный цикл процесса в Linux. Процессы-зомби, сироты, демоны.
16. Сигналы в Linux: SIGTERM, SIGKILL, SIGHUP, SIGINT. Перехват и блокировка сигналов.
17. Система инициализации systemd. Юниты, сервисы, таргеты.
18. Unit-файл systemd: секции [Unit], [Service], [Install]. Зависимости Wants/Requires и порядок After/Before.
19. Демоны: классический паттерн двойного форка и запуск демона под управлением systemd.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

20. Системное логирование: journald, journalctl, rsyslog, уровни важности, ротация журналов.
21. Файловая система Linux: иерархия FHS, назначение основных каталогов.
22. Inode, жёсткие и символические ссылки. Монтирование файловых систем.
23. Виртуальные файловые системы /proc и /sys.
24. Сценарии Bash: переменные, условия, циклы, коды возврата.
25. Планирование задач: cron и таймеры systemd, сравнение подходов.
26. Сокеты как абстракция ОС. TCP- и UDP-сокеты. Клиент-серверная модель.
27. Утилиты netcat и ss. Unix-сокеты и сетевые сокеты.
28. Сетевые демоны и systemd socket activation.
29. Защита сетевых служб на уровне ОС: открытые порты, привязка к интерфейсам, межсетевой экран.
30. Виртуализация: полная виртуализация и паравиртуализация. Гипервизоры типа 1 и типа 2.
31. Аппаратная поддержка виртуализации Intel VT-x и AMD-V. Создание виртуальной машины.
32. Контейнеризация: пространства имён (namespaces) и контрольные группы (cgroups).
33. Docker: образ, слой, контейнер, том. Инструкции Dockerfile (FROM, RUN, CMD, ENTRYPOINT, COPY).
34. Docker Compose: структура docker-compose.yml, сети, тома, зависимости служб.
35. Сравнение виртуальных машин и контейнеров. Паравиртуализация virtio.
36. Модуль subprocess и работа с процессами из Python. Безопасный запуск внешних команд.
37. Модули os и pathlib: работа с файлами и правами доступа из Python.
38. Сокеты на Python: простой TCP-клиент и сервер.
39. Автоматизация Docker и systemd из Python.
40. Логирование в автоматизации: модуль logging, уровни, форматирование, ротация.
41. Хранение сценариев и конфигураций в Git. Назначение GitVerse и GitVerse Actions в администрировании.
42. Российские ИИ-ассистенты (GigaChat, GigaCode, Алиса AI, YandexGPT): назначение и возможности в администрировании.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

43. Принципы постановки задач ИИ-ассистенту при подготовке команд и конфигураций.
44. Риски применения ИИ-ассистентов: «галлюцинации», небезопасные команды, утечка данных. Правила проверки результата.
45. Требования законодательства и этики при работе с ИИ: защита персональных данных, академическая добросовестность.

### 5.2.2 Типовые практические задания

1. Определите версию ядра и дистрибутива Linux, объём оперативной памяти и число ядер процессора (uname, /etc/os-release, lscpu, free).
2. Создайте пользователя `student` и группу `devs`, добавьте пользователя в группу, создайте каталог с правами `2770` для группы `devs` и проверьте работу прав.
3. Запустите процесс `sleep 600` в фоне, найдите его PID, отправьте ему SIGTERM, затем объясните разницу с SIGKILL.
4. Напишите unit-файл systemd для простого сервиса, запускающего скрипт, включите автозапуск и найдите записи сервиса в `journalctl`.
5. Напишите Bash-скрипт, который архивирует каталог с логами старше 7 дней, и поставьте его на ежедневный запуск через cron и через таймер systemd.
6. Найдите, какой процесс слушает порт 8080 (`ss -tlnp`), откройте тестовое TCP-соединение с помощью `nc`.
7. С помощью PowerShell выведите список служб Windows в состоянии Running, создайте локального пользователя и добавьте его в группу.
8. Создайте виртуальную машину с Debian: объём памяти 2 ГБ, 2 процессора, сеть NAT; сделайте снимок состояния.
9. Напишите Dockerfile для простого веб-приложения на PHP, соберите образ и запустите контейнер с пробросом порта.
10. Опишите в docker-compose.yml связку «веб-сервер (PHP-FPM + Nginx) + база данных MariaDB» с общим томом для данных БД.
11. Напишите на Python скрипт, который рекурсивно обходит каталог, находит файлы с правами «для всех» и выводит их список.
12. Сформулируйте запрос к ИИ-ассистенту (GigaChat или Алиса AI) для получения скрипта резервного копирования, найдите в ответе не менее двух недостатков и исправьте их.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

### 5.2.3 Задания для экзамена

#### Экзаменационный билет №1

**Вопрос 1**

Объясните, зачем в архитектуре современных процессоров аппаратно выделены режим ядра и пользовательский режим, и почему переключение между ними возможно только через системный вызов (инструкцию TRAP). Приведите свой пример ситуации, в которой отсутствие такого разделения привело бы к проблеме.

**Вопрос 2**

Сравните полную виртуализацию и паравиртуализацию: как в каждом случае гипервизор обрабатывает привилегированные инструкции гостевой ОС, почему паравиртуализация быстрее и какую цену приходится за это платить. Поясните, как аппаратная поддержка виртуализации (Intel VT-x/AMD-V) изменила это соотношение.

**Практическое задание**

Пользователь запускает на своей рабочей станции гипервизор типа 2 (например, VirtualBox), внутри которого работает гостевая Linux, а поверх неё он планирует ещё поднять несколько Docker-контейнеров.

1. Определите, какой компонент здесь выступает «расширенной машиной» по отношению к оборудованию, а какой — по отношению к процессам внутри гостевой ОС, и объясните разницу между этими двумя уровнями абстракции.
2. Объясните, почему контейнеры внутри гостевой ОС нельзя назвать «ещё одним уровнем виртуализации» в том же смысле, что и сам гипервизор.
3. Предположите, изменится ли и как быстродействие ввода-вывода, если гипервизор в этой схеме — типа 1, а не типа 2.

#### Экзаменационный билет №2

**Вопрос 1**

Опишите своими словами роль MMU и DMA-контроллера в архитектуре персонального компьютера: какую задачу решает каждый из них и почему без них процессору пришлось бы выполнять существенно больше работы. Приведите пример операции, где особенно заметна польза DMA.

**Вопрос 2**

Сравните два сетевых режима виртуальной машины — NAT и Bridged (мост): как в каждом из них гостевая ВМ представлена во внешней сети, и какой режим вы выбрали бы для сервера, к которому должны подключаться другие компьютеры в локальной сети. Обоснуйте выбор.

**Практическое задание**

Администратор создаёт виртуальную машину и указывает для неё диск в формате qcow2 объёмом 40 ГБ, после установки гостевой ОС делает снапшот с именем `clean-install`, а затем устанавливает в ВМ тестовое приложение.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

1. Объясните, почему файл qcow2-диска на хосте сразу после создания ВМ занимает значительно меньше 40 ГБ, и как это связано с понятием разреженного хранения.
2. Опишите, что произойдёт с файловой системой и установленным тестовым приложением внутри ВМ, если администратор выполнит откат к снапшоту `clean-install`.
3. Объясните, почему для этой ВМ имеет смысл выделить фиксированное число vCPU и объём памяти заранее, а не менять их на лету во время работы приложения.

## Экзаменационный билет №3

**Вопрос 1**

Сравните понятия «процесс», «файл» и «дескриптор файла»: как они связаны между собой, и почему операционной системе удобно работать с открытыми ресурсами процесса именно через числовые дескрипторы, а не напрямую через имена файлов. Приведите пример из повседневной работы с ОС, где это разделение проявляется.

**Вопрос 2**

Объясните, почему контейнер принято описывать как «обычный процесс с ограниченным полем зрения и аппетитом», а не как мини-версию виртуальной машины. Опишите, за счёт какого механизма ядра Linux ограничивается «поле зрения» контейнера, а за счёт какого — его «аппетит» к ресурсам.

**Практическое задание**

Инженер выполняет на Linux-хосте команду `unshare --pid --mount --net --fork bash`, после чего запускает в открывшемся терминале `ps aux` и видит только один-два процесса, хотя на хосте их выполняются сотни.

1. Объясните, какое пространство имён (namespace) отвечает за то, что `ps aux` внутри показывает так мало процессов, и почему сами процессы хоста при этом продолжают выполняться как обычно.
2. Укажите, какие ещё namespace потребовалось бы дополнительно указать в команде `unshare`, чтобы у процесса появилось собственное имя хоста и изолированный набор IPC-объектов.
3. Объясните, почему одних namespaces недостаточно, чтобы полноценно ограничить долю CPU и объём памяти, которые может потребить процесс в этом изолированном окружении, и какой механизм ядра нужен для этого дополнительно.

## Экзаменационный билет №4

**Вопрос 1**

Объясните, зачем в UNIX-подобных системах создание нового процесса разделено на два системных вызова — `fork()` и `exec()` — вместо одного вызова «запустить новую программу». Опишите, что происходит с адресным пространством процесса на каждом из этих шагов, и какую роль в этой схеме играет `wait()`.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

**Вопрос 2**

Сравните инструкции `RUN` и `CMD` в Dockerfile: на каком этапе жизненного цикла образа и контейнера выполняется каждая из них, и как эта разница связана с понятием слоёв (layers) образа. Приведите пример ошибки, которую совершают новички, путая эти две инструкции.

**Практическое задание**

Дан фрагмент Dockerfile:

```
FROM python:3.11-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install -r requirements.txt
COPY . .
CMD ["python", "app.py"]
```

1. Объясните, почему `requirements.txt` копируется и устанавливается отдельной инструкцией до копирования всего остального кода (`COPY . .`), а не одной общей командой `COPY . .` в начале файла.
2. Опишите, что произойдёт при повторной сборке образа (`docker build`), если изменился только файл `app.py`, а `requirements.txt` остался прежним — какие слои будут пересобраны, а какие возьмутся из кэша.
3. Объясните, что нужно изменить в этом Dockerfile или в команде запуска контейнера, чтобы приложение `app.py`, слушающее порт 8000 внутри контейнера, было доступно с хоста на порту 8080.

## Экзаменационный билет №5

**Вопрос 1**

Сравните монолитную архитектуру ядра и микроядро: где в каждом случае располагается код драйверов и файловых систем, и как это влияет на устойчивость системы к ошибке в одном из компонентов. Объясните, какую цену приходится платить за дополнительную надёжность микроядра.

**Вопрос 2**

Объясните, зачем нужен Docker Compose, если каждый контейнер и так можно запустить отдельной командой `docker run`. Опишите, как сервисы, описанные в одном файле `docker-compose.yml`, обнаруживают и обращаются друг к другу по сети, и почему для этого не нужно вручную настраивать IP-адреса.

**Практическое задание**

Дан фрагмент `docker-compose.yml`:

```
services:
  web:
    build: .
    ports:
      - "5000:5000"
    environment:
      - DB_HOST=db
    depends_on:
```



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

```
      - db
  db:
    image: postgres:15
    volumes:
      - db_data:/var/lib/postgresql/data

volumes:
  db_data:
```

1. Объясните, почему сервис `web` может обращаться к базе данных по имени `db` (см. переменную `DB_HOST=db`), хотя в конфигурации не указан никакой конкретный IP-адрес.
2. Опишите проблему, которая может возникнуть при запуске `docker compose up`, если контейнер `web` попытается подключиться к базе данных сразу при старте, несмотря на наличие `depends_on: - db`. Объясните, почему `depends_on` не решает эту проблему полностью.
3. Объясните, зачем данные PostgreSQL вынесены в именованный том `db_data`, а не хранятся просто внутри слоёв контейнера `db`, и что произойдёт с этими данными при выполнении `docker compose down` без дополнительных флагов.

## Экзаменационный билет №6

**Вопрос 1**

Объясните, почему системный код вроде ядра Linux традиционно пишут на языке Си, а не на языках с тяжёлой рантайм-средой вроде Java или Python. Опишите, как устроена связка «указатель — адрес переменной» в Си на примере `int *p = &a; *p = 10;`, и почему такой прямой доступ к памяти важен именно для системного программирования.

**Вопрос 2**

Сравните виртуальные машины и контейнеры по трём параметрам: степень изоляции, время запуска и типичный размер образа. Объясните, почему на практике эти два подхода часто комбинируют — например, запускают контейнеры внутри виртуальной машины в облаке, а не выбирают что-то одно.

**Практическое задание**

Инженер настраивает виртуальную машину для гостевой Linux в облачной среде и обнаруживает, что при полностью эмулированном (не virtio) сетевом адаптере и диске производительность сети и дисковых операций внутри ВМ заметно ниже, чем при использовании virtio-net и virtio-blk, хотя процессор виртуализируется с аппаратным ускорением (VT-x) в обоих случаях.

1. Объясните, почему аппаратное ускорение виртуализации процессора само по себе не решает проблему медленной работы эмулированных дисковых и сетевых устройств.
2. Опишите, в чём принципиальное отличие virtio-драйверов от полной эмуляции реального устройства, и почему для использования virtio гостевая ОС должна содержать соответствующие драйверы.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

3. Свяжите этот пример с общей идеей паравиртуализации (как в билете про гипервизоры) и объясните, почему современные гипервизоры используют гибридный подход: полную виртуализацию процессора вместе с паравиртуализацией устройств.

## Экзаменационный билет №7

**Вопрос 1**

Опишите три категории прав доступа к файлу в Linux (владелец/группа/остальные) и объясните разницу в смысле бита `x` для обычного файла и для каталога. Приведите пример каталога, на котором отсутствие права `x` при наличии права `r` делает содержимое видимым в списке, но недоступным для открытия по имени, и объясните, почему так происходит.

**Вопрос 2**

Сравните модуль `os` и класс `pathlib.Path` как средства работы с файловой системой из Python: как в каждом из них выполняется склейка путей и рекурсивный поиск файлов по шаблону. Объясните, почему в современном коде предпочтение обычно отдают `pathlib`.

**Практическое задание**

Администратор написал Python-скрипт, который должен рекурсивно выставить владельцу право на выполнение для всех файлов в каталоге проекта:

```
import os

for root, dirs, files in os.walk("/srv/project"):
    for name in files:
        path = os.path.join(root, name)
        os.chmod(path, 755)
```

После запуска права на файлах оказались выставлены совершенно неожиданным образом (например, `--wx-wx--t` вместо `rwxr-xr-x`).

1. Найдите ошибку в вызове `os.chmod` и объясните, почему число `755` было воспринято не так, как ожидал автор скрипта.
2. Предложите исправленный вариант вызова `os.chmod` и поясните синтаксическое правило, которое при этом соблюдается.
3. Объясните, почему подобная ошибка, допущенная в скрипте автоматизации, потенциально опаснее одной опечатки при ручном вызове `chmod` в терминале.

## Экзаменационный билет №8

**Вопрос 1**

Объясните, что такое процесс-зомби, при каких условиях он появляется и почему его нельзя «убить» сигналом `SIGKILL`. Сравните сигналы `SIGTERM` и `SIGKILL` с точки зрения того, может ли процесс их перехватить и корректно завершить свою работу.

**Вопрос 2**

Объясните, чем вызов `subprocess.run(["ls", "-l", path])` предпочтительнее вызова с `shell=True` и склеенной строкой команды, если часть аргументов приходит



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

от внешнего пользователя. Опишите, какую конкретную угрозу устраняет передача аргументов списком.

**Практическое задание**

Разработчик написал функцию для остановки внешнего процесса-обработчика:

```
import subprocess

proc = subprocess.Popen(["python3", "worker.py"])

def stop_worker():
    if proc.poll() is None:
        proc.terminate()
        proc.wait(timeout=5)
```

1. Объясните, что означает результат `proc.poll() is None` в этом коде и почему нельзя было использовать вместо него `proc.wait()`.
2. Опишите, что произойдёт, если процесс `worker.py` перехватывает сигнал, посылаемый методом `terminate()`, и игнорирует его дольше пяти секунд. Как в этом случае корректно завершить процесс принудительно?
3. Объясните, почему в этой функции неуместно было бы использовать `subprocess.run()` вместо `subprocess.Popen()`.

## Экзаменационный билет №9

**Вопрос 1**

Объясните, чем декларативное описание сервиса в unit-файле systemd отличается от императивного скрипта запуска. Сравните директивы `Wants=` и `Requires=` в секции `[Unit]`, а также поясните, почему `After=` не является заменой ни для одной из них.

**Вопрос 2**

Опишите последовательность действий серверного сокета от создания до приёма подключения (создать → bind → listen → accept) и клиентского сокета (создать → connect). Объясните, почему после вызова `accept()` на сервере фактически появляются два разных сокета, и для чего нужен каждый из них.

**Практическое задание**

Дан минимальный сервер на Python:

```
import socket

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(("0.0.0.0", 9000))
s.listen()

while True:
    conn, addr = s.accept()
    data = conn.recv(1024)
    conn.send(data.upper())
    conn.close()
```



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

При второй попытке запустить этот же скрипт (не остановив первый) появляется `OSError: [Errno 98] Address already in use`.

1. Объясните, почему возникает эта ошибка и почему одновременно два процесса не могут занять один и тот же порт на одном интерфейсе.
2. Метод `.recv()` возвращает данные типа `bytes`, а не `str` — почему это так, и что пришлось бы сделать в коде (`.decode()`/`.encode()`), если бы вместо `data.upper()` требовалось, например, применить строковый метод, недоступный у `bytes`, или сохранить результат в текстовый файл в кодировке UTF-8?
3. Опишите ограничение этого сервера при подключении сразу нескольких клиентов и назовите два подхода, которыми это ограничение обычно устраняют.

## Экзаменационный билет №10

**Вопрос 1**

Объясните, что такое демон и почему ему не нужен управляющий терминал. Опишите, как эволюционировало системное логирование в Linux — от текстовых файлов syslog/rsyslog в `/var/log` до структурированного журнала journald — и какие возможности даёт `journalctl`, которых не было у обычных текстовых логов.

**Вопрос 2**

Опишите паттерн watchdog-скрипта на Python, который периодически проверяет состояние сервиса или контейнера и перезапускает его при сбое. Объясните, почему watchdog без ограничения числа попыток перезапуска может не решать проблему, а маскировать её.

**Практическое задание**

Дан фрагмент watchdog-скрипта, который должен следить за Docker-контейнером:

```
import subprocess
import time

while True:
    result = subprocess.run(
        ["docker", "inspect", "-f", "{{.State.Running}}", "web_app"],
        capture_output=True, text=True
    )
    if result.stdout.strip() != "true":
        subprocess.run(["docker", "restart", "web_app"])
    time.sleep(10)
```

1. Объясните назначение параметров `capture_output=True` и `text=True` в первом вызове `subprocess.run` и что изменилось бы в поведении скрипта при их отсутствии.
2. Опишите, к каким последствиям приведёт ситуация, когда контейнер `web_app` падает сразу после каждого перезапуска из-за ошибки в конфигурации, и



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

почему такой скрипт может часами «незаметно» держать контейнер в цикле падений.

3. Предложите, как с помощью модуля `logging` зафиксировать каждый факт перезапуска и его причину, вместо того чтобы полагаться только на `journalctl`, если сам watchdog-скрипт запущен через systemd-таймер.

## Экзаменационный билет №11

**Вопрос 1**

Объясните, чем жёсткая ссылка отличается от символической с точки зрения inode: почему жёсткая ссылка не может «сломаться» при удалении исходного файла, а символическая — может. Опишите, что такое `/proc` и приведите пример информации, которую можно получить из этой виртуальной файловой системы.

**Вопрос 2**

Объясните, почему при постановке Python-скрипта в `crontab` рекомендуется указывать полный путь и к интерпретатору `python3`, и к самому файлу скрипта, а не полагаться на короткие имена, которые прекрасно работают в интерактивном терминале. Сравните этот способ планирования со связкой systemd-таймер + сервис с точки зрения последующего анализа логов.

**Практическое задание**

Скрипт `backup.py` нормально запускается вручную командой `python3 backup.py`, но при добавлении в `crontab` строкой

```
* * * * * python3 backup.py >> /home/user/backup.log 2>&1
```

в файле `backup.log` через минуту появляется ошибка вроде `FileNotFoundError` или `command not found`, хотя вручную скрипт отрабатывает без проблем.

1. Объясните наиболее вероятную причину такого расхождения между запуском вручную и запуском из cron, связанную с окружением, в котором выполняется задание cron.
2. Предложите исправленную строку crontab, устраняющую эту проблему.
3. Опишите, какой была бы диагностика этой же проблемы, если бы `backup.py` запускался не через cron, а через systemd-таймер, и куда в этом случае стоило бы посмотреть в первую очередь вместо файла `backup.log`.

## Экзаменационный билет №12

**Вопрос 1**

Сравните способ постановки задачи на расписание через `crontab` (пять полей: минута, час, день месяца, месяц, день недели) и через связку systemd-таймер + сервис с директивой `OnCalendar=`. Объясните, почему систематическое исполнение таймера удобнее анализировать через `journalctl`, чем исполнение задания cron.

**Вопрос 2**

Объясните смысл адреса сокета как пары IP-адрес + порт и проведите аналогию с «домом» и «номером окошка приёма». Сравните TCP и UDP по наличию



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

установления соединения и по гарантиям доставки и порядка данных, приведя для каждого протокола бытовую аналогию.

**Практическое задание**

На сервере уже запущен процесс, слушающий TCP-порт 8080. Администратор пытается запустить ещё один сервис, тоже настроенный на порт 8080, двумя разными способами и получает разные результаты:

1. При ручном запуске Python-скрипта с `socket.socket(socket.AF_INET, socket.SOCK_STREAM)` и `.bind(("0.0.0.0", 8080))` он немедленно получает `OSError: [Errno 98] Address already in use`. Объясните причину этой ошибки на уровне модели адресации сокетов.
2. Администратор предлагает временное решение — запустить второй сервис не на TCP, а на Unix-domain сокете, адресованном путём `/run/second_service.sock`. Объясните, почему это решает конфликт портов, и укажите, в каких сценариях (локально или по сети) такой сокет может использоваться, а в каких — нет.
3. Предположите, что изменится в порядке действий сервера (bind → listen → accept), если вместо `SOCK_STREAM` использовать `SOCK_DGRAM`, и какая из стадий жизненного цикла в этом случае вообще не нужна.

## Экзаменационный билет №13

**Вопрос 1**

Сравните использование `netcat` в роли сервера и в роли клиента при отладке сетевого взаимодействия. Объясните, как с помощью `ss -tlnp` определить, какой процесс уже занял нужный порт, и почему при попытке повторно запустить `nc -l -p 12345` на этом порту вы получите ошибку «Address already in use».

**Вопрос 2**

Объясните идею socket activation в systemd: почему выгодно разделить задачи «слушать порт» и «обслуживать подключение» между `.socket`-юнитом и связанным с ним `.service`-юнитом. Опишите, какие два конкретных преимущества это даёт по сравнению с постоянно запущенным сервисом, и почему обычный пользователь не может создать `.socket`-юнит с `ListenStream=80`.

**Практическое задание**

На сервере выполнили команду и получили следующий вывод:

```
$ ss -tlnp
State    Recv-Q   Send-Q     Local Address:Port     Peer Address:Port
Process
LISTEN   0        128        0.0.0.0:22            0.0.0.0:*
users:(("sshd",pid=812,fd=3))
LISTEN   0        4096       127.0.0.1:8080         0.0.0.0:*
users:(("python3",pid=2214,fd=4))
```

Также в системе есть unix-сокет:

```
$ ls -l /run/docker.sock
srw-rw---- 1 root docker 0 сен 15 09:00 /run/docker.sock
```



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

1. Объясните, чем отличаются два TCP-сокета в выводе `ss` по адресу прослушивания (`0.0.0.0` против `127.0.0.1`) с точки зрения того, откуда можно установить к ним соединение.
2. Разработчик хочет проверить руками, что сервис на порту 8080 отвечает, используя `nc`. Приведите команду и объясните, как по появлению строки `ESTABLISHED` в `ss -tn` можно убедиться, что соединение установлено.
3. Объясните, что означает буква `s` в начале строки прав доступа `srw-rw----` у файла `docker.sock`, и почему для такого файла бессмысленно применять к нему `cat` или `nc` так же, как к обычному текстовому файлу.

## Экзаменационный билет №14

**Вопрос 1**

Объясните две дополняющие друг друга роли операционной системы: ОС как «расширенная машина» и ОС как «менеджер ресурсов». Приведите для каждой роли собственный пример и поясните, почему мультиплексирование ресурсов бывает временным (во времени) и пространственным (в пространстве), на примере процессора и оперативной памяти.

**Вопрос 2**

Объясните, чем декларативный подход systemd (unit-файлы, описывающие желаемое состояние) принципиально отличается от императивных init-скриптов, которые последовательно выполняют команды запуска. Опишите назначение директив `WantedBy`, `Wants` и `Requires`, и в чём разница между мягкой и жёсткой зависимостью между юнитами.

**Практическое задание**

Дан unit-файл `/etc/systemd/system/backup.service`:

```
[Unit]
Description=Backup service
After=network.target

[Service]
ExecStart=/usr/local/bin/backup.sh
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

1. Объясните, что произойдёт при выполнении `systemctl enable backup.service`, и почему без этой команды сервис не запустится автоматически при следующей перезагрузке, даже если он один раз был запущен вручную через `systemctl start`.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

2. Опишите разницу между директивами `After=network.target` в секции `[Unit]` и `Restart=on-failure` в секции `[Service]`: на что влияет каждая из них и в какой момент жизненного цикла сервиса.
3. Скрипт `backup.sh` завершается с кодом выхода 1 из-за нехватки места на диске. Объясните, что будет делать systemd благодаря `Restart=on-failure`, и почему это не решит исходную проблему.

## Экзаменационный билет №15

**Вопрос 1**

Объясните назначение MMU (модуля управления памятью) и DMA-контроллера в архитектуре персонального компьютера. Почему без MMU было бы невозможно надёжно изолировать адресные пространства процессов друг от друга, и почему DMA снижает нагрузку на процессор при работе с устройствами ввода-вывода.

**Вопрос 2**

Объясните, что такое демон в Linux и почему для него важно отвязаться от управляющего терминала (через двойной форк с `setsid` или через соответствующий `Type=` в systemd). Опишите путь эволюции системного логирования от классического syslog/rsyslog к journald, и какие практические возможности даёт `journalctl` (например, фильтрация по юниту, времени или уровню важности), которых не было у простого чтения текстовых файлов в `/var/log`.

**Практическое задание**

Администратор диагностирует проблему с сервисом и выполняет команды:

```
$ systemctl status nginx.service
$ journalctl -u nginx.service --since "10 min ago" -p err
$ journalctl -u nginx.service -f
```

1. Объясните, чем отличается информация, которую даёт `systemctl status`, от информации, которую даёт `journalctl -u nginx.service`, и почему администратору может быть недостаточно только первой команды.
2. Опишите, что делает флаг `-p err` во второй команде и почему его полезно использовать вместо просмотра всего журнала сервиса подряд.
3. Объясните, что произойдёт при выполнении третьей команды с флагом `-f`, и на что она похожа по поведению среди классических инструментов работы с текстовыми лог-файлами.

## Экзаменационный билет №16

**Вопрос 1**

Опишите классификацию операционных систем (настольные, серверные, встроенные, реального времени, смарт-карт) и объясните, какое требование к системе является определяющим для класса ОС реального времени в отличие от настольной ОС. Поясните на примерах, зачем в базовых понятиях ОС различают процесс, дескриптор и адресное пространство.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

**Вопрос 2**

Объясните назначение стандарта FHS (Filesystem Hierarchy Standard) и приведите по смыслу отличия каталогов `/etc`, `/var` и `/tmp`. Опишите, чем виртуальные файловые системы `/proc` и `/sys` отличаются от обычных файловых систем на диске, и приведите по одному примеру информации, которую можно получить из каждой.

**Практическое задание**

Администратор выполняет на сервере следующие команды:

```
$ df -h /
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1        50G   47G  0.5G  99% /

$ ls -li report.txt report_hardlink.txt report_link.txt
1048577 -rw-r--r-- 2 user user 2048 сен 15 10:00 report.txt
1048577 -rw-r--r-- 2 user user 2048 сен 15 10:00 report_hardlink.txt
1048600 lrwxrwxrwx 1 user user    11 сен 15 10:01 report_link.txt ->
report.txt
```

1. Объясните, что означают числа в первом столбце вывода `ls -li` для трёх файлов, и почему у `report.txt` и `report_hardlink.txt` совпадает номер inode, а у `report_link.txt` — нет.
2. Опишите, что произойдёт с содержимым, доступным через `report_hardlink.txt` и через `report_link.txt`, если исходный файл `report.txt` будет удалён командой `rm report.txt`.
3. Судя по выводу `df -h`, на разделе `/` осталось всего 0.5G свободного места. Объясните, к какому каталогу из FHS стоит в первую очередь обратиться при поиске файлов-кандидатов на удаление (логи, кэши, временные файлы) и почему.

## Экзаменационный билет №17

**Вопрос 1**

Объясните, почему переход из пользовательского режима в режим ядра при системном вызове возможен только через контролируемую точку входа (инструкцию TRAP), а не произвольным способом. Опишите на примере связки `fork()` + `exec()` + `wait()`, как порождается новый процесс и как родительский процесс узнаёт о завершении потомка.

**Вопрос 2**

Сравните автоматизацию задач через `crontab` и через связку `.timer` + `.service` в systemd. Опишите формат пяти полей расписания cron и назначение директивы `OnCalendar=` в таймере systemd, а также объясните, почему результаты работы задачи, запущенной через systemd-таймер, автоматически попадают в `journalctl`, а результаты cron-задачи — нет (если это не настроено отдельно).

**Практическое задание**

Дан bash-скрипт `cleanup.sh`:

```
#!/bin/bash
```



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

```
LOGDIR="/var/log/myapp"
DAYS=7

for file in "$LOGDIR"/*.log; do
    if [ -f "$file" ]; then
        find "$file" -mtime +$DAYS -delete
    fi
done
```

и строка в `crontab -e`:

```
0 3 * * * /usr/local/bin/cleanup.sh >> /var/log/cleanup.log 2>&1
```

1. Объясните, какие права необходимо выставить на файл `cleanup.sh`, чтобы cron мог его запустить как исполняемый скрипт, и какую роль в этом играет шебанг `#!/bin/bash` в первой строке.
2. Расшифруйте расписание `0 3 * * *`: в какие моменты времени будет запускаться скрипт.
3. Предложите, как переписать этот же запуск в виде пары unit-файлов systemd (`.service` и `.timer`), и объясните, какое практическое преимущество (в части журналирования) появится по сравнению с исходной строкой в `crontab`, где перенаправление `>> ... 2>&1` сделано вручную.

## Экзаменационный билет №18

### Вопрос 1

Сравните монолитную архитектуру ядра и микроядро с точки зрения того, какой код выполняется в режиме ядра, а какой — в пользовательском режиме. Объясните, почему отказ драйвера устройства в системе с микроядром обычно не приводит к падению всей ОС, тогда как в монолитном ядре может привести, и какую цену микроядро платит за эту устойчивость.

### Вопрос 2

Объясните смысл разделения прав доступа на владельца, группу и остальных пользователей в Linux, и почему бит `x` для каталога означает не «выполнение», а возможность доступа к файлам внутри него по имени. Опишите назначение битов SUID и sticky bit на конкретных примерах (`/usr/bin/passwd` и каталог `/tmp`).

### Практическое задание

Администратор выполняет команды на сервере:

```
$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 68208 мар  1  2024 /usr/bin/passwd

$ ls -ld /tmp
drwxrwxrwt 15 root root 4096 сен 15 09:30 /tmp

$ stat -c "%a" report.sh
644
```



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

1. Объясните, за счёт какого бита в правах `-rwsr-xr-x` обычный пользователь может изменить свой пароль в `/etc/shadow`, хотя этот файл доступен на запись только root, и почему замена этого бита на обычный `x` сломала бы `passwd` для непривилегированных пользователей.
2. Опишите, что означает буква `t` в конце строки `drwxrwxrwt` у каталога `/tmp`, и какую конкретную проблему она решает в каталоге, доступном на запись всем пользователям системы.
3. Пользователю нужно, чтобы скрипт `report.sh` (права 644) можно было запускать напрямую как `./report.sh`. Приведите команду `chmod` в восьмеричной форме, которая добавит право на выполнение владельцу и группе, но не остальным, и объясните, почему `chmod 777 report.sh` в общем случае считается плохой практикой.

## Экзаменационный билет №19

**Вопрос 1**

Объясните, почему системный код (ядро Linux, утилиты UNIX, компиляторы) исторически и по сей день пишут преимущественно на языке Си, а не на языках с более развитым рантаймом (Java, Python). Раскройте роль указателей в этом контексте на примере `int *p = &a;` и `*p = 10;`: что физически происходит с памятью в каждой из строк и почему такой прямой доступ к адресам недоступен в языках с управляемой памятью.

**Вопрос 2**

Опишите жизненный цикл процесса в Linux от создания до завершения через связку `fork()` → `exec()` → `wait()`. Объясните, что произойдёт с дочерним процессом, если родитель завершится или «забудет» вызвать `wait()`, и почему в этом случае процесс может перейти в состояние зомби, а не просто исчезнуть.

**Практическое задание**

Дан фрагмент вывода команды `ps`:

```
  PID  PPID STAT COMMAND
 1042     1 S    sshd
 2201  1042 S    bash
 2350  2201 Z    [python3] <defunct>
 2401     1 S    nginx
```

Пользователь также сообщает, что нажал `Ctrl+Z` на процессе, запущенном в терминале с PID 2201, а затем закрыл этот терминал, не выполнив `bg`/`disown`.

1. Объясните, что означает состояние `Z` у процесса с PID 2350 и какое действие со стороны процесса 2201 (родителя) устранило бы эту запись.
2. Опишите, какой сигнал отправляет `Ctrl+Z` и что произойдёт с процессом при закрытии терминала (какой сигнал он получит и что нужно было сделать заранее, чтобы процесс пережил закрытие терминала).



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

3. Приведите конкретную команду (используя `kill`, `pkill` или `killall`), которая корректно (с возможностью процессу освободить ресурсы) завершит процесс `nginx`, и объясните, чем это отличается от `kill -9`.

## Экзаменационный билет №20

### Вопрос 1

Сокет — это разновидность файлового дескриптора, представляющая конечную точку сетевого взаимодействия. Опишите полный жизненный цикл TCP-соединения со стороны сервера (создание сокета → `bind` → `listen` → `accept`) и со стороны клиента (создание сокета → `connect`), используя аналогию «адрес = дом (IP) + квартира (порт)». Объясните разницу между TCP и UDP через аналогию «телефонный звонок» и «открытка».

### Вопрос 2

Объясните различие между образом (image) и контейнером в Docker, используя аналогию «класс и экземпляр» или «шаблон и его копия». Опишите, зачем в Dockerfile нужны слои и кэш сборки, и в чём принципиальная разница между инструкциями `RUN` и `CMD` — на каком этапе жизненного цикла каждая из них выполняется.

### Практическое задание

Дан Dockerfile:

```
FROM python:3.11-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install -r requirements.txt
COPY . .
EXPOSE 8000
CMD ["python", "app.py"]
```

После сборки командой `docker build -t myapp .` пользователь пытается запустить контейнер и обращается к сервису в браузере по адресу `http://localhost:8000`, но получает ошибку соединения. Также ему сообщают, что при повторной сборке после правки одной строчки в `app.py` шаг `RUN pip install -r requirements.txt` каждый раз выполняется заново, хотя `requirements.txt` не менялся.

1. Какую команду `docker run` нужно было использовать, чтобы порт 8000 контейнера стал доступен на хосте, и почему без явного указания портов подключение снаружи невозможно?
2. Объясните, почему изменение `app.py` приводит к повторному выполнению `RUN pip install`, хотя сам файл `requirements.txt` не менялся — свяжите это с механизмом слоёв и кэша Docker, и предложите, как переставить инструкции `COPY`, чтобы избежать лишней переустановки зависимостей.
3. Через какой файл (сокет) на хосте демон Docker обычно принимает команды от клиента `docker`, и почему это важно понимать при настройке доступа к Docker без `sudo`?



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

## Экзаменационный билет №21

### Вопрос 1

Опишите, как утилита `netcat` (`nc`) может выступать одновременно и в роли сервера (`nc -l -p порт`), и в роли клиента (`nc host порт`), и почему это делает её удобным инструментом для быстрой диагностики сетевого взаимодействия. Объясните, чем unix-сокет отличается от TCP-сокета с точки зрения адресации (путь в файловой системе вместо IP:порт) и почему такие сокеты видны в выводе `ls -l` с буквой `s` в начале строки прав.

### Вопрос 2

Объясните назначение файла `docker-compose.yml` и то, какую проблему он решает по сравнению с запуском нескольких контейнеров вручную через `docker run`. Раскройте смысл директивы `depends_on`: почему она гарантирует только порядок запуска контейнеров, но не готовность сервиса внутри контейнера принимать соединения, и к каким последствиям это может привести на практике.

### Практическое задание

Дан фрагмент `docker-compose.yml`:

```
services:
  db:
    image: postgres:15
    environment:
      POSTGRES_PASSWORD: secret
    volumes:
      - db_data:/var/lib/postgresql/data
  web:
    build: .
    ports:
      - "5000:5000"
    depends_on:
      - db
volumes:
  db_data:
```

После `docker compose up -d` сервис `web` падает с ошибкой подключения к базе данных в первые секунды после старта, хотя через пару секунд ошибка исчезает при ручном перезапуске. Также администратор выполняет `ss -tlnp | grep 5432` на хосте и не видит порт 5432 в списке прослушиваемых.

1. Объясните, почему `web` не смог подключиться к `db` сразу после запуска, несмотря на `depends_on`, и какой механизм (например, retry-логика в коде или `healthcheck`) решил бы эту проблему корректно.
2. Как сервис `web` должен указывать адрес базы данных в строке подключения (какое имя хоста использовать вместо IP), чтобы обращаться к контейнеру `db` внутри сети Compose?
3. Объясните, почему `ss -tlnp | grep 5432` на хосте ничего не покажет, даже если PostgreSQL внутри контейнера `db` слушает порт 5432, и что нужно было изменить в конфигурации, чтобы порт стал виден на хосте.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

## Экзаменационный билет №22

**Вопрос 1**

Объясните идею systemd socket activation: почему вместо постоянно запущенного сетевого демона systemd может сам слушать порт (через `.socket`-юнит с `ListenStream=`) и запускать связанный `.service` только при первом входящем подключении. Какие два преимущества это даёт (ленивый запуск и отсутствие потери подключений при параллельном старте нескольких сервисов) и почему порты ниже 1024 требуют привилегий?

**Вопрос 2**

Сравните виртуальные машины и контейнеры по уровню изоляции, скорости запуска, размеру образа и плотности размещения на одном физическом хосте. Объясните, почему даже при полной аппаратной виртуализации CPU (VT-x/AMD-V) для сетевых и дисковых устройств часто применяют паравиртуализационные драйверы virtio (virtio-net, virtio-blk), и в чём выигрыш по сравнению с честной эмуляцией железа.

**Практическое задание**

Администратор настраивает `sshd` через socket activation и приводит юнит-файл:

```
# /etc/systemd/system/sshd.socket
[Socket]
ListenStream=22
Accept=no

[Install]
WantedBy=sockets.target
```

После `systemctl enable --now sshd.socket` он проверяет состояние командой `systemctl status sshd.service` и видит, что сервис находится в состоянии `inactive (dead)`, хотя `ss -tlnp | grep :22` показывает, что порт 22 прослушивается процессом `systemd`.

1. Объясните, почему `sshd.service` показывает `inactive`, хотя порт 22 уже прослушивается — свяжите это с тем, кто именно (systemd или sshd) сейчас держит сокет и когда запустится сам сервис `sshd.service`.
2. Что произойдёт технически в момент, когда извне поступит первое TCP-подключение на порт 22 (какой юнит активируется и как передаётся уже установленное соединение)?
3. Предположим, вместо ВМ для этого же хоста рассматривается контейнер. Объясните, почему запуск изолированного sshd в отдельном контейнере обычно избыточен по сравнению с полноценной ВМ, и какое из свойств контейнеризации (общее ядро ОС) здесь становится ограничением, а не преимуществом.

## Экзаменационный билет №23

**Вопрос 1**

Сравните гипервизоры первого типа (bare-metal: KVM, ESXi, Xen) и второго типа (hosted: VirtualBox) — где именно располагается гипервизор по отношению к



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

аппаратуре и хостовой ОС и как это сказывается на производительности и типичных сценариях применения. Объясните разницу между полной виртуализацией (эмуляция привилегированных инструкций с аппаратным ускорением VT-x/AMD-V) и паравиртуализацией (модифицированная гостевая ОС, вызывающая hypercall вместо перехвата инструкций, как в Xen PV) — почему паравиртуализация быстрее, но требует модификации гостя.

**Вопрос 2**

Опишите, какие модули стандартной библиотеки Python (`os`, `pathlib`) применяются системным администратором для работы с файловой системой, и в чём разница подходов `os.path` (строковые операции) и `pathlib.Path` (объектно-ориентированный, с оператором `/` для соединения путей). Объясните, почему вызов `os.chmod(path, 0o755)` обязательно требует префикса `0o`, и что произойдёт, если написать `os.chmod(path, 755)` без этого префикса.

**Практическое задание**

Дан фрагмент Python-скрипта для массовой простановки прав на исполняемые файлы:

```
import os
from pathlib import Path


target_dir = Path("/opt/scripts")


for script in target_dir.glob("**/*.sh"):
    os.chmod(script, 755)
    print(f"Updated: {script}")
```

Скрипт запускается для приведения прав всех `.sh`-файлов в директорию к `rwxr-xr-x`, но после его выполнения администратор обнаруживает, что права установились неверно, а `ls -l` показывает подозрительные символы вместо ожидаемых `rwxr-xr-x`.

1. Найдите ошибку в вызове `os.chmod` и объясните, к каким конкретно правам доступа приведёт число `755` без префикса `0o` (переведите его в восьмеричное представление и опишите итоговые биты прав).
2. Объясните, что делает `target_dir.glob("**/*.sh")` и почему использование `pathlib` здесь предпочтительнее ручной конкатенации строк для путей.
3. Скрипт применяет права ко всем найденным файлам без разбора. Какой риск это создаёт при массовой автоматизации прав в реальной системе, и какую проверку (например, через `os.stat()` и модуль `stat`) стоило бы добавить перед изменением прав каждого файла?

## Экзаменационный билет №24

**Вопрос 1**

Перечислите основные компоненты, из которых складывается конфигурация виртуальной машины (vCPU, память, диск, сеть), и сравните форматы виртуальных дисков qcow2, VDI и VMDK — почему qcow2 обычно предпочитают за поддержку разреженного хранения и снапшотов. Объясните разницу между сетевыми режимами



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

NAT и Bridged для виртуальной машины: как каждый из них влияет на видимость ВМ извне физической сети.

**Вопрос 2**

Объясните, зачем модуль `subprocess` рекомендуется вызывать со списком аргументов (`subprocess.run(["cmd", "arg1", "arg2"])`), а не с одной строкой и `shell=True`, и какую конкретную угрозу безопасности (command injection) это предотвращает. Опишите разницу между `subprocess.run()` (блокирующий вызов) и `subprocess.Popen()` (неблокирующий), а также назначение методов `.poll()`, `.wait()`, `.terminate()` и `.kill()` у объекта `Popen`.

**Практическое задание**

Администратор пишет Python-скрипт для резервного копирования диска виртуальной машины и запуска фоновой проверки:

```
import subprocess

vm_name = input("Введите имя ВМ для снапшота: ")

result = subprocess.run(
    "vboxmanage snapshot " + vm_name + " take backup1",
    shell=True,
    capture_output=True,
    text=True
)

if result.returncode != 0:
    print("Ошибка:", result.stderr)

proc = subprocess.Popen(["long_running_check.sh"])
proc.wait(timeout=5)
```

Скрипт предполагалось использовать для создания снапшота ВМ перед рискованным обновлением, а также для запуска фоновой проверки, которая не должна блокировать выполнение более пяти секунд.

1. Объясните, какую уязвимость создаёт способ формирования команды через конкатенацию строк с `shell=True`, если пользователь введёт в качестве имени ВМ строку `mytest; rm -rf /`. Перепишите вызов `subprocess.run(...)`, используя список аргументов, чтобы устранить эту уязвимость.
2. Объясните, что произойдёт с вызовом `proc.wait(timeout=5)`, если `long_running_check.sh` не завершится за 5 секунд, и какое исключение при этом возникнет. Как с помощью `.poll()` можно было бы проверять состояние процесса без риска получить такое исключение?
3. Что означает снапшот виртуальной машины (в контексте `vboxmanage snapshot ... take backup1`) и почему создание снапшота qcow2/VDI перед обновлением — хорошая практика с точки зрения возможности отката?



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

## Экзаменационный билет №25

**Вопрос 1**

Объясните, почему контейнер правильнее описывать не как «облегчённую виртуальную машину», а как обычный процесс операционной системы с ограниченной видимостью и ограниченным аппетитом к ресурсам. Опишите, какие конкретно namespaces (PID, mount, network, UTS, IPC, user) используются для изоляции «видимости» процесса, и приведите пример того, что увидит процесс внутри своего PID-namespace по сравнению с хостовой системой.

**Вопрос 2**

Опишите последовательность вызовов при построении простейшего TCP-сервера на модуле `socket` в Python (`socket()`, `bind()`, `listen()`, `accept()`) и клиента (`socket()`, `connect()`). Объясните, почему вызов `accept()` возвращает новый объект сокета, отдельный от исходного слушающего сокета, и что произойдёт с этим слушающим сокетом, пока сервер обменивается данными с уже подключённым клиентом.

**Практическое задание**

Дан фрагмент кода однопоточного эхо-сервера на Python:

```
import socket

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind(("0.0.0.0", 9090))
server.listen()
print("Сервер запущен, ожидание подключений...")

while True:
    conn, addr = server.accept()
    print(f"Подключение от {addr}")
    data = conn.recv(1024)
    conn.send(data.upper())
    conn.close()
```

1. К серверу одновременно пытаются подключиться два клиента. Что произойдёт со вторым клиентом, пока сервер обрабатывает первого, и почему данный код не подходит для одновременного обслуживания нескольких клиентов?
2. Клиент запускается раньше сервера и получает исключение `ConnectionRefusedError`. Объясните причину этой ошибки на уровне сетевого стека и предложите, как изменить логику клиента, чтобы избежать падения программы.
3. При повторном запуске сервера сразу после аварийного завершения предыдущего Python выбрасывает `OSError: Address already in use`. Объясните природу этой ошибки и предложите способ её избежать при разработке.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

## Экзаменационный билет №26

**Вопрос 1**

Объясните разницу между образом (image) и контейнером (container) в Docker, включая роль слоёв (layers) и кэширования при сборке. Опишите, зачем в Dockerfile нужны обе инструкции `RUN` и `CMD`, чем они принципиально различаются по моменту выполнения, и какую типичную ошибку допускают новички, путая их назначение.

**Вопрос 2**

Сравните два подхода к автоматизации управления Docker-контейнерами из Python: вызов CLI-команд через `subprocess` и использование официального Docker SDK (`docker.from_env()`). Объясните, к какому системному интерфейсу в обоих случаях в итоге обращается управляющий код, и почему для управления сервисами systemd аналогичного SDK не существует и приходится использовать `subprocess` с `systemctl`.

**Практическое задание**

Дан Dockerfile:

```
FROM python:3.11-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install -r requirements.txt
COPY . .
EXPOSE 8000
CMD ["python", "app.py"]
```

1. Разработчик изменил только код в `app.py` (без изменения `requirements.txt`) и пересобрал образ командой `docker build`. Объясните, какие слои образа Docker переиспользует из кэша, а какие пересоберёт заново, и почему порядок инструкций `COPY` в этом Dockerfile важен для скорости сборки.
2. После сборки образа командой `docker run -p 8080:8000 myapp` приложение недоступно по адресу `http://localhost:8000`. Объясните смысл параметра `-p 8080:8000` и укажите, по какому адресу и порту приложение будет доступно на хосте.
3. Коллега предлагает заменить `EXPOSE 8000` на `CMD` с флагом, «открывающим порт наружу». Объясните, почему инструкция `EXPOSE` сама по себе не публикует порт на хосте и как соотносятся `EXPOSE` и флаг `-p` при запуске контейнера.

## Экзаменационный билет №27

**Вопрос 1**

Опишите назначение файла `docker-compose.yml` и ключевых секций сервиса: `image`/`build`, `ports`, `environment`, `volumes`, `depends_on`. Объясните, как сервисы, описанные в одном compose-файле, обращаются друг к другу по сети, и почему директива `depends_on` определяет только порядок запуска контейнеров, но не гарантирует, что зависимый сервис уже готов принимать запросы.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

**Вопрос 2**

Сравните два способа регулярного запуска Python-скрипта в Linux: через `crontab` и через systemd-таймер (`.timer` + `.service`). Объясните, почему при запуске скрипта из `crontab` необходимо указывать полный путь к интерпретатору `python3` и к самому скрипту, и какое преимущество даёт systemd-таймер в части автоматического журналирования вывода через `journalctl`.

**Практическое задание**

Дан фрагмент `docker-compose.yml`:

```
services:
  db:
    image: postgres:15
    environment:
      POSTGRES_PASSWORD: secret
    volumes:
      - db_data:/var/lib/postgresql/data


  web:
    build: .
    ports:
      - "5000:5000"
    environment:
      DATABASE_URL: postgres://postgres:secret@db:5432/postgres
    depends_on:
      - db


volumes:
  db_data:
```

и фрагмент Python-скрипта, который сервис `web` должен запускать по расписанию раз в сутки для очистки старых логов:

```
import logging


logging.basicConfig(filename="/var/log/cleanup.log", level=logging.INFO)


try:
    do_cleanup()
except Exception:
    pass
```

1. При первом запуске `docker compose up` сервис `web` иногда падает с ошибкой подключения к базе данных, хотя `depends_on: - db` указан. Объясните, почему это происходит, и предложите, что можно сделать на уровне приложения или compose-файла, чтобы избежать сбоя.
2. Объясните, зачем в `DATABASE_URL` используется имя хоста `db`, а не `localhost` или IP-адрес, и как оно разрешается внутри сети, созданной Docker Compose.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

3. В приведённом Python-скрипте блок `except Exception: pass` считается опасным антипаттерном логирования. Перепишите словесно (или укажите конкретное изменение кода), как нужно обработать исключение, чтобы ошибка не терялась молча, используя возможности модуля `logging`.

## Экзаменационный билет №28

**Вопрос 1**

Сравните виртуальные машины и контейнеры по уровню изоляции, скорости запуска, размеру образа и плотности размещения на одном физическом хосте. Объясните, что такое паравиртуализация устройств на примере virtio-net и virtio-blk, и почему даже при полной аппаратной виртуализации процессора эмуляция «настоящего» сетевого адаптера или диска оказывается медленнее, чем использование паравиртуального драйвера.

**Вопрос 2**

Опишите, какие модули и функции языка Python (`os`, `pathlib`) применяются для базовой автоматизации системного администрирования: получение и смена текущей директории, чтение переменных окружения, создание директорий, обход дерева каталогов. Объясните разницу между `os.mkdir()` и `os.makedirs()`, а также почему при вызове `os.chmod(path, 0o755)` обязательно указывать права в восьмеричной системе (префикс `0o`).

**Практическое задание**

Дан скрипт на Python для массовой смены прав доступа к файлам конфигурации:

```
import os
from pathlib import Path

config_dir = Path("/etc/myapp/configs")

for path in config_dir.glob("*.conf"):
    st = os.stat(path)
    print(path, oct(st.st_mode))
    os.chmod(path, 0o644)
```

1. Объясните, что выведет строка `print(path, oct(st.st_mode))` и почему для получения именно прав доступа (а не полного значения `st_mode`) на практике применяют функцию `stat.S_IMODE()`.
2. Скрипт нужно расширить так, чтобы он рекурсивно обходил не только `config_dir`, но и все вложенные поддиректории. Укажите, какую функцию модуля `os` (или метод `pathlib.Path`) стоит использовать вместо `glob("*.conf")`, и почему.
3. Администратор случайно запустил похожий скрипт с `os.chmod(path, 0o777)` рекурсивно на всей директории `/etc`. Опишите, в чём заключается риск подобной массовой автоматизации прав и какие меры предосторожности (например, режим предварительного просмотра, ограничение области действия) стоило предпринять до запуска.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

## Экзаменационный билет №29

### Вопрос 1

Опишите роль systemd как init-системы и процесса с PID 1. Объясните структуру unit-файла сервиса (секции `[Unit]`, `[Service]`, `[Install]`, ключи `ExecStart`, `Restart`, `WantedBy`) и разницу между декларативным описанием зависимостей `Wants`/`Requires` (мягкая и жёсткая зависимость) и порядковыми директивами `After`/`Before`, которые не подразумевают автоматического запуска зависимого юнита.

### Вопрос 2

Опишите основные компоненты, которые необходимо сконфигурировать при создании виртуальной машины: количество вCPU, объём оперативной памяти, виртуальный диск и сетевой адаптер. Сравните сетевые режимы NAT и Bridged (Bridge) с точки зрения доступности ВМ из внешней сети и получения ею собственного IP-адреса, а также объясните назначение снапшотов виртуального диска и возможность отката к предыдущему состоянию.

### Практическое задание

Дан unit-файл systemd-сервиса `/etc/systemd/system/backup-agent.service`:

```
[Unit]
Description=Backup Agent Service
After=network.target

[Service]
ExecStart=/usr/local/bin/backup-agent --daemon
Restart=on-failure
RestartSec=5

[Install]
WantedBy=multi-user.target
```

1. Администратор выполнил `systemctl enable backup-agent.service`, но после перезагрузки сервис не запустился. Объясните разницу между командами `systemctl enable` и `systemctl start`, и какую из них администратор забыл выполнить, а также как убедиться, что сервис действительно активен, с помощью `systemctl status`.
2. Объясните смысл директивы `After=network.target` в секции `[Unit]` и почему для сервиса, который активно использует сеть, обычно недостаточно только указать `After`, если требуется, чтобы сетевая подсистема была не просто «запущена после», а гарантированно доступна.
3. Требуется создать новую виртуальную машину для тестирования этого сервиса в изолированной сети без доступа извне и без получения адреса от домашнего роутера. Какой сетевой режим (NAT или Bridged) стоит выбрать и почему, а также какой формат виртуального диска предпочтителен, если планируется часто делать снапшоты перед тестами обновлений сервиса?



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

## Экзаменационный билет №30

### Вопрос 1

Опишите стандарт иерархии файловой системы Linux (FHS) и назначение нескольких ключевых каталогов. Объясните, что такое inode и почему метаданные файла (владелец, права, размер, время изменения) хранятся отдельно от его имени, а также поясните принципиальную разницу между жёсткой ссылкой (hard link) и символической ссылкой (symlink), включая то, почему символическая ссылка может «сломаться», а жёсткая — нет.

### Вопрос 2

Опишите назначение функции `subprocess.run()` для запуска внешних команд из Python, объяснив роль параметров `capture_output`, `text` и `check`, а также значение атрибута `returncode`. Сравните `subprocess.run()` с классом `Popen` в ситуации, когда нужно запустить долгоживущий процесс и периодически проверять, работает ли он ещё (`.poll()`), а также объясните разницу между `.terminate()` и `.kill()`.

### Практическое задание

Дан фрагмент кода на Python, который монтирует внешний диск и проверяет точки монтирования:

```
import subprocess


user_input = input("Введите имя устройства: ")
result = subprocess.run("mount /dev/" + user_input + " /mnt/data",
shell=True)
print(result.returncode)
```

и вывод команды `lsblk`:

```
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda      8:0    0   100G  0 disk
├─sda1   8:1    0    1G  0 part /boot
└─sda2   8:2    0   99G  0 part /
sdb      8:16   0    50G  0 disk
└─sdb1   8:17   0   50G  0 part /mnt/data
```

1. В коде команда `mount` формируется конкатенацией строк и запускается с `shell=True`. Объясните, в чём заключается уязвимость такого подхода (command injection), и покажите, как правильно переписать вызов `subprocess.run()`, используя список аргументов вместо строки с `shell=True`.
2. По выводу `lsblk` определите, какое устройство и раздел смонтированы в `/mnt/data`, и объясните, чем точка монтирования отличается от самого блочного устройства /раздела.
3. Требуется получить информацию о файловой системе, смонтированной в `/mnt/data` (например, объём занятого места), не считывая её напрямую с диска через `lsblk`. Укажите подходящую команду и то, как с её результатом можно работать программно через `subprocess.run(..., capture_output=True, text=True)`.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

## Экзаменационный билет №31

**Вопрос 1**

Сравните семейства Unix и Windows с точки зрения администрирования: как устроены учётные записи и права доступа, какие оболочки используются, где хранятся журналы. Приведите собственный пример задачи, которую в одной из систем проще решить, чем в другой.

**Вопрос 2**

Объясните, зачем нужны пространства имён (namespaces) и контрольные группы (cgroups) и как на их основе строятся контейнеры. Почему контейнер не является «лёгкой виртуальной машиной»?

**Практическое задание**

Дан фрагмент вывода PowerShell:

```
PS> Get-Service | Where-Object {$_.Status -eq 'Running'} | Select-Object -
First 3
Status   Name      DisplayName
------   ----      -----------
Running  Dhcp      DHCP-клиент
Running  Dnscache  DNS-клиент
Running  EventLog  Журнал событий Windows
```

1. Объясните, что делает каждая часть конвейера (`|`) и чем он отличается от конвейера в Bash.
2. Запишите команду PowerShell, останавливающую службу `Spooler`, и поясните, какие права для этого нужны.
3. Укажите, какой командой Linux можно получить аналогичный список запущенных служб.

## Экзаменационный билет №32

**Вопрос 1**

Опишите механизм управления учётными записями и службами в Windows (локальные пользователи, группы, службы, журнал событий) и сопоставьте его с аналогами в Linux (`/etc/passwd`, systemd, journald).

**Вопрос 2**

Опишите жизненный цикл Docker-образа и контейнера: от Dockerfile до запущенного контейнера. Чем отличаются `CMD` и `ENTRYPOINT`, где хранятся данные контейнера и как их сохранить?

**Практическое задание**

Дан PowerShell-сценарий:

```
$users = Get-LocalUser | Where-Object { $_.Enabled -eq $true }
foreach ($u in $users) {
    "{0} - последний вход: {1}" -f $u.Name, $u.LastLogon
}
```

1. Объясните, что выводит сценарий.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

2. Как изменить сценарий, чтобы он выводил только пользователей, не входивших более 90 дней?
3. Объясните, какие меры безопасности стоит применить к найденным неиспользуемым учётным записям.

## Экзаменационный билет №33

**Вопрос 1**

Расскажите о российских ИИ-ассистентах (GigaChat, GigaCode, Алиса AI, YandexGPT): какие задачи администратора они могут ускорить и каковы их ограничения. Какие данные нельзя передавать ИИ-ассистенту и почему?

**Вопрос 2**

Сравните cron и таймеры systemd: формат расписания, ведение журналов, поведение при пропущенном запуске. В каких случаях следует предпочесть каждый из подходов?

**Практическое задание**

Студент запросил у ИИ-ассистента скрипт очистки временных файлов и получил:

```
#!/bin/bash
cd /tmp
rm -rf *
chmod -R 777 /var/www
```

1. Найдите в скрипте не менее трёх проблем (корректность, безопасность, переносимость) и объясните, к каким последствиям они приведут.
2. Предложите безопасный вариант скрипта.
3. Опишите, как вы проверите исправленный скрипт перед запуском в рабочей среде.

## Экзаменационный билет №34

**Вопрос 1**

Опишите правила безопасного и добросовестного применения ИИ-ассистентов при выполнении учебных и рабочих задач: проверка результата, запрет на передачу конфиденциальных данных, ссылки на использование ИИ, требования Федерального закона №152-ФЗ.

**Вопрос 2**

Объясните, как работает активация службы по сокету (systemd socket activation) и какие преимущества она даёт по сравнению с обычным запуском демона.

**Практическое задание**

Администратор попросил ИИ-ассистента составить `docker-compose.yml` для PHP-приложения с MariaDB и получил:

```
services:
  app:
    image: php:latest
    ports: ["80:80"]
    environment:
```



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

```
      DB_PASSWORD: root
  db:
    image: mariadb:latest
    environment:
      MARIADB_ROOT_PASSWORD: root
    ports: ["3306:3306"]
```

1. Определите не менее трёх недостатков конфигурации с точки зрения безопасности и воспроизводимости.
2. Предложите исправления (версии образов, секреты, сетевое взаимодействие, тома).
3. Укажите, где в этой конфигурации не хватает тома для сохранения данных БД.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

## 5.3 КОМПЛЕКТ «ТЕКУЩИЙ КОНТРОЛЬ»

### 5.3.1 Задания для срезов знаний

1. Назовите режимы работы процессора и объясните, почему приложение не может напрямую обратиться к оборудованию. Какой механизм позволяет ему попросить ядро о действии?
2. Файл имеет права `-rwxr-x---`. Запишите их в восьмеричной форме и определите, что может делать группа-владелец.
3. Процесс завершился, а родитель не вызвал `wait()`. В каком состоянии находится процесс? Как устранить проблему?
4. Чем зависимость `Requires=` отличается от `Wants=` и чем эти директивы отличаются от `After=`?
5. Объясните разницу между жёсткой и символической ссылкой. Что произойдёт при удалении исходного файла в каждом случае?
6. Сопоставьте: cron-выражение `*/15 9-18 * * 1-5` — определите, когда запускается задача.
7. Какой порт и протокол использует сервис, если `ss -tlnp` показывает `0.0.0.0:8080`? Кто может подключиться к такому сервису и как ограничить доступ только локальным компьютером?
8. В чём разница между образом и контейнером? Какая инструкция Dockerfile задаёт команду, выполняемую при запуске контейнера?
9. Назовите три риска использования ИИ-ассистента при написании сценариев администрирования и один способ снижения каждого.

### 5.3.2 Тестовые задания

Текущий контроль знаний по каждой лекции проводится в системе дистанционного обучения Moodle. Для каждой темы создан банк тестовых вопросов (типы: один правильный ответ, верно/неверно, краткий ответ); на занятии каждому обучающемуся выдаётся тест из случайно выбранных вопросов банка. Попыток – 2, время – 15 минут, оценка – по доле правильных ответов (90–100 % – «5», 75–89 % – «4», 60–74 % – «3»).

| Банк вопросов | Число вопросов |
|---|---|
| 1.1. Что такое операционная система | 28 |
| 1.2. Аппаратное обеспечение персонального компьютера | 33 |
| 1.3. Зоопарк операционных систем и ключевые понятия | 29 |
| 1.4. Системные вызовы | 29 |
| 1.5. Структура операционных систем | 29 |
| 1.6. Мир глазами языка Си | 27 |
| 2.1. Пользователи, группы и права доступа | 41 |
| 2.2. Процессы и сигналы в Linux | 51 |



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

| Банк вопросов | Число вопросов |
|---|---|
| 2.3. systemd- юниты, сервисы и таргеты | 37 |
| 2.4. Демоны и системное логирование | 37 |
| 2.5. Файловая система Linux- FHS, proc и sys | 37 |
| 2.6. Автоматизация в shell- скрипты, cron и таймеры systemd | 35 |
| 3.1. Сокеты как абстракция операционной системы | 41 |
| 3.2. Сокеты в Linux на практике- netcat, ss и unix-сокеты | 38 |
| 3.3. Сетевые демоны и systemd socket activation | 40 |
| 4.1. Виртуализация- полная виртуализация, паравиртуализация и гипервизоры | 43 |
| 4.2. Практика- создание и настройка виртуальной машины | 40 |
| 4.3. Контейнеризация изнутри- namespaces и cgroups | 40 |
| 4.4. Docker- образы, контейнеры, Dockerfile | 46 |
| 4.5. Docker Compose и многоконтейнерные приложения | 43 |
| 4.6. Виртуализация и контейнеризация- сравнение, паравиртуализация virtio на практике | 37 |
| 5.1. Python как инструмент системного администратора- os, pathlib, права доступа | 33 |
| 5.2. Процессы из Python- subprocess | 33 |
| 5.3. Сокеты на Python- клиент-сервер | 32 |
| 5.4. Автоматизация Docker и systemd из Python | 31 |
| 5.5. Планирование задач и логирование в автоматизации | 30 |

Всего в банках – 940 вопросов.

**Примеры тестовых заданий**

1. Чем отличается режим ядра от пользовательского режима работы процессора?
   а) В режиме ядра доступны все инструкции и всё оборудование, в пользовательском режиме — только ограниченное подмножество
   б) Режим ядра работает быстрее за счёт отключения кэша
   в) Пользовательский режим используется только во время загрузки компьютера
   г) Различия чисто программные и не влияют на доступ к оборудованию
   *Правильный ответ: а*
2. Что означает PPID процесса?
   а) PID процесса-родителя, который создал данный процесс вызовом fork()
   б) PID самого процесса, только записанный в другом столбце
   в) Количество потомков, которые процесс успел породить
   г) Приоритет процесса при распределении процессорного времени
   *Правильный ответ: а*
3. Что произойдёт при выполнении команды nc localhost 12345, если на порту 12345 уже запущен слушающий процесс nc -l?
   а) nc подключится к слушающему процессу как клиент, и оба терминала смогут обмениваться текстом
   б) Команда автоматически остановит слушающий процесс на этом порту



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

в) Появится ошибка «Address already in use»
г) Ничего не произойдёт, потому что nc не поддерживает роль клиента
*Правильный ответ: а*

4. Почему считается ошибкой выделять виртуальной машине больше vCPU, чем физических ядер у хоста?
а) Это не ускоряет работу ВМ, а лишь увеличивает конкуренцию за реальные такты процессора
б) Гипервизор в этом случае просто откажется запускать виртуальную машину
в) Это автоматически удвоит объём выделенной оперативной памяти
г) Это приведёт к повреждению файла виртуального диска
*Правильный ответ: а*

5. Что делает subprocess.run(["ls", "-l"])?
а) Запускает команду и дожидается её завершения, аналог wait() из цикла 2
б) Запускает команду и сразу возвращает управление, не дожидаясь завершения
в) Только проверяет, существует ли такая команда, но не запускает её
г) Отправляет команду на удалённый сервер по сети
*Правильный ответ: а*

**Дополнительные тестовые задания по темам 1.2 (Windows) и 5.2 (ИИ-ассистенты)**

1. Какая командная оболочка по умолчанию используется для администрирования Windows через конвейер объектов?
а) Bash; б) PowerShell; в) Zsh; г) Fish. *Правильный ответ: б*

2. Какая команда PowerShell выводит список локальных учётных записей?
а) Get-LocalUser; б) ls /etc/passwd; в) net-users-list; г) Show-Accounts.
*Правильный ответ: а*

3. Что содержит журнал событий Windows?
а) только сообщения об ошибках приложений; б) записи о событиях системы, безопасности и приложений; в) историю браузера; г) список установленных драйверов без времени событий. *Правильный ответ: б*

4. Что обязательно нужно сделать с кодом, полученным от ИИ-ассистента, перед запуском в рабочей среде?
а) ничего, ИИ не ошибается; б) прочитать, понять и проверить в изолированной среде по документации; в) отправить другому ИИ без чтения; г) сразу выдать права root. *Правильный ответ: б*

5. Какие данные нельзя передавать ИИ-ассистенту?
а) общие вопросы о синтаксисе команд; б) пароли, ключи доступа, персональные данные и конфиденциальные сведения; в) тексты ошибок без реквизитов доступа; г) публичные фрагменты документации. *Правильный ответ: б*



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

6. Верно ли, что ответ ИИ-ассистента всегда соответствует официальной документации? *Правильный ответ: нет*
7. Назовите принцип, нарушает команда `chmod -R 777 /var/www`, предложенная ассистентом. *Правильный ответ: принцип наименьших привилегий*

### 5.3.3 Задания для проверочных, самостоятельных и контрольных работ



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

## Контрольная работа №1. Основы операционных систем

Работа завершает цикл 1 «Основы операционных систем» и проверяет усвоение материала лекций 1.1–1.6:

– 1.1 Что такое операционная система

– 1.2 Аппаратное обеспечение персонального компьютера

– 1.3 Зоопарк операционных систем и ключевые понятия

– 1.4 Системные вызовы

– 1.5 Структура операционных систем

– 1.6 Мир глазами языка Си

**Цель работы**

Проверить, что студент способен: (1) объяснить назначение и роль операционной системы двумя разными способами (как расширенной машины и как менеджера ресурсов); (2) связать аппаратные компоненты компьютера с задачами, которые решает ОС; (3) свободно оперировать базовыми понятиями ОС (процесс, файл, дескриптор, адресное пространство); (4) объяснить механизм системного вызова и назвать назначение ключевых вызовов POSIX; (5) сравнить архитектурные подходы к построению ОС; (6) объяснить, почему системный код пишут на языке С, и прочитать простейшую программу с указателем.

**Формат проведения**

Письменная работа, 45–60 минут, без использования конспектов и интернета (разрешён только чистый лист для черновика). Работа состоит из четырёх частей: тестовая часть, часть с открытыми вопросами, практическая часть (чтение кода/команд) и часть на сопоставление понятий.

Ниже приведён один из вариантов работы. Преподаватель может подготовить 2–4 равноценных по сложности варианта, меняя конкретные примеры и порядок вопросов, но сохраняя структуру и распределение баллов.

**Часть 1. Тестовые вопросы с одним правильным ответом (10 вопросов × 2 балла = 20 баллов)**

1. Как называется режим работы процессора, в котором доступны все инструкции и вся аппаратура компьютера?
   а) пользовательский режим
   б) режим ядра
   в) режим ожидания
   г) режим совместимости
2. Какой компонент компьютера отвечает за преобразование виртуальных адресов памяти в физические?
   а) кэш-контроллер
   б) MMU (блок управления памятью)
   в) DMA-контроллер
   г) BIOS
3. Что из перечисленного НЕ является типом операционной системы по классификации из лекции 1.3?



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

а) встроенная ОС
б) ОС реального времени
в) ОС смарт-карт
г) ОС виртуальной реальности

4. Какой системный вызов создаёт точную копию текущего процесса?
а) `exec`
б) `fork`
в) `wait`
г) `spawn`

5. Что возвращает дочерний процесс после успешного вызова `fork()`?
а) PID родителя
б) 0
в) −1
г) PID самого себя

6. Какая архитектура ОС оставляет в режиме ядра только минимально необходимый код (планирование процессов, обмен сообщениями, базовые вызовы), а драйверы и файловые системы выносит в пользовательский режим?
а) монолитная
б) многоуровневая
в) микроядро
г) экзоядро

7. Чем гипервизор типа 1 принципиально отличается от гипервизора типа 2?
а) типом процессора, на котором может работать
б) тем, что работает прямо на аппаратуре, а не поверх обычной ОС
в) поддержкой только одной гостевой ОС одновременно
г) отсутствием необходимости в драйверах

8. Что делает в языке C выражение `*p`, если `p` — указатель?
а) берёт адрес переменной `p`
б) объявляет новый указатель
в) обращается к значению, лежащему по адресу, записанному в `p`
г) освобождает память, на которую указывает `p`

9. Какая программа отвечает за то, чтобы при сборке большого проекта на C пересобирались только изменившиеся файлы?
а) компилятор
б) препроцессор
в) `make`
г) компоновщик (linker)

10. Что из перечисленного ОС использует для мультиплексирования ресурса «в пространстве» (а не «во времени»)?
а) поочерёдное выделение процессорного времени разным процессам



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

б) разделение оперативной памяти на области под разные процессы
в) очередь печати на принтер
г) список ожидающих ввода процессов

**Ключ:** 1-б, 2-б, 3-г, 4-б, 5-б, 6-в, 7-б, 8-в, 9-в, 10-б

**Часть 2. Открытые вопросы (4 вопроса × 10 баллов = 40 баллов)**

1. Объясните разницу между двумя взглядами на операционную систему — «ОС как расширенная машина» и «ОС как менеджер ресурсов». Приведите по одному собственному примеру на каждый взгляд (не из лекции).
2. Опишите своими словами, что происходит с момента, когда программа на языке C вызывает `read(fd, buffer, nbytes)`, и до момента, когда управление возвращается в программу с прочитанными данными. Упомяните роль инструкции TRAP.
3. Сравните монолитную архитектуру и архитектуру микроядра: в чём выигрывает каждая и почему ошибка в драйвере устройства ведёт к разным последствиям в этих двух архитектурах.
4. Почему операционные системы (например, ядро Linux) в подавляющем большинстве написаны на языке C, а не на языках вроде Java или Python? Назовите не менее трёх причин, опираясь на материал лекции 1.6.

**Часть 3. Практическая часть (2 задания × 15 баллов = 30 баллов)**

**Задание 1.** Дан фрагмент программы на C:

```
#include <stdio.h>

int main() {
    int a = 5;
    int *p = &a;
    *p = 10;
    printf("%d\n", a);
    return 0;
}
```

Что будет напечатано на экране и почему? Объясните пошагово, что делает каждая из строк `int *p = &a;` и `*p = 10;`.

**Задание 2.** Пользователь в терминале Linux ввёл команду:

```
cat report.txt | sort > result.txt &
```

Опишите, что произойдёт при выполнении этой команды: какие процессы будут созданы, как между ними будут передаваться данные, что произойдёт с управлением терминалом. Какие понятия из лекций 1.3 и 1.4 здесь задействованы (назовите не менее трёх)?

**Часть 4. Сопоставление понятий (10 баллов)**

Сопоставьте термин слева с его определением справа (запишите пары вида «1-В»):

| № | Термин | Буква | Определение |
|---|---|---|---|



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

| № | Термин | Буква | Определение |
|---|---|---|---|
| 1 | Дескриптор файла | А | Управляемая точка входа из пользовательской программы в ядро ОС |
| 2 | Системный вызов | Б | Числовой идентификатор, который ОС выдаёт процессу после открытия файла |
| 3 | Кэш-память | В | Небольшая быстрая память между процессором и оперативной памятью |
| 4 | Гипервизор | Г | Программа, создающая и управляющая виртуальными машинами |
| 5 | Компоновщик (linker) | Д | Программа, объединяющая объектные файлы в один исполняемый файл |

**Ключ:** 1-Б, 2-А, 3-В, 4-Г, 5-Д

**Критерии оценивания**

Максимум за работу — **100 баллов**, распределённых по частям:

| Часть | Баллы |
|---|---|
| Часть 1. Тестовые вопросы (10 × 2) | 20 |
| Часть 2. Открытые вопросы (4 × 10) | 40 |
| Часть 3. Практическая часть (2 × 15) | 30 |
| Часть 4. Сопоставление понятий | 10 |
| **Итого** | **100** |

**Критерии оценивания открытых вопросов (часть 2)**

Каждый открытый вопрос (10 баллов) оценивается по трём аспектам:

– **Полнота ответа** (0–4 балла): раскрыты ли все составляющие вопроса.

– **Корректность формулировок** (0–4 балла): нет фактических ошибок, термины использованы правильно.

– **Наличие собственного примера или пояснения своими словами** (0–2 балла): ответ не сводится к пересказу определения, показывает понимание, а не заучивание.

**Критерии оценивания практической части (часть 3)**

Каждое задание (15 баллов) оценивается так:

– Правильный итоговый ответ/результат — 6 баллов.

– Верное пошаговое объяснение хода рассуждения — 6 баллов.

– Точное использование терминологии курса — 3 балла.

**Шкала перевода суммарных баллов в оценку**

| Баллы | Оценка |
|---|---|
| 90–100 | «отлично» |
| 75–89 | «хорошо» |
| 60–74 | «удовлетворительно» |
| менее 60 | «неудовлетворительно» |



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

**Дополнительное правило**

Если студент набрал менее 60 баллов, но верно выполнил обе задачи практической части (часть 3) полностью, преподаватель вправе засчитать работу на «удовлетворительно» — это показывает, что студент способен применять знания на практике, даже если испытывает затруднения с теоретическими формулировками. Обратная ситуация (высокий балл за теорию при нулевом результате в практической части) не даёт права на повышение оценки: практическая часть обязательна к выполнению.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

## Контрольная работа №2. Linux вглубь

Работа завершает цикл 2 «Linux вглубь» и проверяет усвоение материала лекций 2.1–2.6:

– 2.1 Пользователи, группы и права доступа

– 2.2 Процессы и сигналы в Linux

– 2.3 systemd: юниты, сервисы и таргеты

– 2.4 Демоны и системное логирование

– 2.5 Файловая система Linux: FHS, /proc и /sys

– 2.6 Автоматизация в shell: скрипты, cron и таймеры systemd

**Цель работы**

Проверить, что студент способен: (1) читать и осознанно назначать права доступа к файлам и каталогам, объяснять смысл специальных битов; (2) объяснять жизненный цикл процесса, различать сигналы по их назначению и перехватываемости; (3) читать и минимально модифицировать unit-файлы systemd, понимать разницу между зависимостями `Wants`/`Requires` и порядком `After`/`Before`; (4) объяснять, что такое демон и как устроено системное логирование; (5) ориентироваться в иерархии файловой системы Linux и понимать разницу между жёсткими и символическими ссылками, а также природу `/proc`; (6) написать и поставить на расписание простой bash-скрипт, сравнить cron и systemd-таймеры.

**Формат проведения**

Письменная работа, 45–60 минут, без использования конспектов и интернета (разрешён только чистый лист для черновика). Работа состоит из четырёх частей: тестовая часть, часть с открытыми вопросами, практическая часть (чтение конфигов/вывода команд) и часть на сопоставление понятий.

Ниже приведён один из вариантов работы. Преподаватель может подготовить 2–4 равноценных по сложности варианта, меняя конкретные примеры и порядок вопросов, но сохраняя структуру и распределение баллов.

**Часть 1. Тестовые вопросы с одним правильным ответом (10 вопросов × 2 балла = 20 баллов)**

1. Какому режиму доступа соответствует восьмеричная запись `750`?
   а) `rwxr-x---`
   б) `rwx-w----`
   в) `rw-r-xr--`
   г) `rwxr--r--`
2. Что даёт x-бит, установленный на КАТАЛОГЕ (а не на файле)?
   а) возможность запускать каталог как программу
   б) возможность обращаться к содержимому каталога по имени (входить в него, открывать файлы внутри)
   в) возможность удалять каталог
   г) автоматическое наследование прав всеми вложенными файлами
3. Какой сигнал процесс не может ни перехватить, ни заблокировать?
   а) `SIGTERM`



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

б) `SIGHUP`

в) `SIGKILL`

г) `SIGINT`

4. Что произойдёт с дочерним процессом, если он завершился, а родитель не вызвал `wait()`?

а) дочерний процесс автоматически перезапустится

б) дочерний процесс станет процессом-зомби (zombie) до тех пор, пока родитель не заберёт код завершения

в) дочерний процесс будет усыновлён `systemd` и продолжит работу

г) система немедленно завершит родительский процесс

5. За что отвечает секция `[Service]` в unit-файле systemd?

а) описывает, к какому таргету принадлежит юнит

б) задаёт, как именно запускать и перезапускать сервис (`ExecStart`, `Type`, `Restart`)

в) описывает зависимости от других юнитов

г) хранит журнал последних запусков сервиса

6. Чем systemd-таймер (`.timer`) выгодно отличается от классического `cron` в плане диагностики?

а) таймер выполняется быстрее физически

б) записи о каждом запуске автоматически попадают в `journalctl`, тогда как cron нужно логировать вручную

в) таймер не требует прав на файл-скрипт

г) таймер может запускаться только от имени root

7. Что хранится в inode файла?

а) само имя файла

б) метаданные файла — владелец, права, размер, время изменения, указатели на блоки данных

в) путь к файлу от корня файловой системы

г) содержимое файла в сжатом виде

8. Пользователь создал жёсткую ссылку на файл, а затем удалил оригинальный файл. Что произойдёт?

а) данные будут потеряны, жёсткая ссылка станет нерабочей

б) данные останутся доступны через жёсткую ссылку, так как оба имени указывают на один и тот же inode

в) жёсткая ссылка автоматически превратится в символическую

г) операция удаления оригинала будет заблокирована системой

9. Какая команда покажет журнал конкретного systemd-сервиса `myapp.service`?

а) `cat /var/log/myapp.log`

б) `journalctl -u myapp.service`

в) `systemctl grep myapp`

г) `ps aux | myapp`



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

10. Что делает `umask 022`, установленный в системе?
    а) удаляет права на выполнение у всех новых файлов
    б) вычитает указанные права из прав по умолчанию при создании новых файлов и каталогов
    в) устанавливает пароль на новые файлы
    г) назначает нового владельца для всех создаваемых файлов

**Ключ:** 1-а, 2-б, 3-в, 4-б, 5-б, 6-б, 7-б, 8-б, 9-б, 10-б

**Часть 2. Открытые вопросы (4 вопроса × 10 баллов = 40 баллов)**

1. Объясните разницу между смыслом x-бита для обычного файла и для каталога. Приведите собственный пример ситуации (не из лекции и не из лабораторной), в которой у пользователя есть право на чтение файла внутри каталога, но нет права на `cd` в этот каталог, и объясните, что при этом произойдёт при попытке прочитать файл по полному пути.
2. Опишите своими словами путь от команды `crontab -e` до фактического выполнения задачи, а затем — путь от создания `.timer`-юнита systemd до выполнения связанного с ним `.service`. Сравните оба подхода: в чём практическое преимущество systemd-таймера, а в чём — простота cron.
3. Сравните два способа «отвязать» долгоживущий сервис от терминала и сделать его демоном: классический паттерн двойного форка с `setsid()` и объявление сервиса как unit-файла systemd с `Type=forking` или `Type=simple`. Кто в каждом случае отвечает за корректное поведение при перезапуске и сборе вывода — сам код программы или systemd?
4. Почему процесс-зомби не потребляет процессорное время и почти не потребляет память, но всё же считается проблемой, если таких процессов накапливается много? Объясните, как эта проблема устраняется на практике.

**Часть 3. Практическая часть (2 задания × 15 баллов = 30 баллов)**

**Задание 1.** Дан вывод команды:

```
-rwxr-x--- 1 anna staff  8192 Sep 10 09:14 deploy.sh
drwxr-xr-x 2 anna staff  4096 Sep 10 09:10 scripts
```

Ответьте: 1) какие права из перечисленных есть у пользователя `anna`, у группы `staff` и у всех остальных на файл `deploy.sh`; 2) сможет ли пользователь, входящий в группу `staff`, но не являющийся `anna`, запустить `./deploy.sh`, находясь в каталоге `scripts`; 3) запишите восьмеричным числом права `deploy.sh` и объясните, как вы их посчитали.

**Задание 2.** Дан фрагмент unit-файла `backup.service`:

```
[Unit]
Description=Резервное копирование данных
After=network.target

[Service]
Type=simple
```



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

```
ExecStart=/usr/local/bin/backup.sh
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

Ответьте: 1) в какой момент загрузки системы этот сервис попытается запуститься и почему именно после `network.target`; 2) что произойдёт, если `backup.sh` завершится с ошибкой, и как это регулируется директивой `Restart=`; 3) какой командой администратор должен включить автозапуск этого сервиса при загрузке системы, и чем эта команда отличается от простого запуска сервиса командой `systemctl start`.

**Часть 4. Сопоставление понятий (10 баллов)**

Сопоставьте термин слева с его определением справа (запишите пары вида «1-В»):

| № | Термин | Буква | Определение |
|---|---|---|---|
| 1 | Демон | А | Виртуальная файловая система, отображающая состояние ядра и процессов в реальном времени |
| 2 | Sticky bit | Б | Фоновый процесс без управляющего терминала, обычно запущенный при старте системы |
| 3 | `/proc` | В | Бит доступа на каталоге, запрещающий удалять чужие файлы внутри него всем, кроме владельца |
| 4 | `journalctl` | Г | Утилита для просмотра структурированного системного журнала, который ведёт `journald` |
| 5 | FHS | Д | Стандарт, определяющий назначение и расположение основных каталогов Linux |

**Ключ:** 1-Б, 2-В, 3-А, 4-Г, 5-Д

**Критерии оценивания**

Максимум за работу — **100 баллов**, распределённых по частям:

| Часть | Баллы |
|---|---|
| Часть 1. Тестовые вопросы (10 × 2) | 20 |
| Часть 2. Открытые вопросы (4 × 10) | 40 |
| Часть 3. Практическая часть (2 × 15) | 30 |
| Часть 4. Сопоставление понятий | 10 |
| **Итого** | **100** |

**Критерии оценивания открытых вопросов (часть 2)**

Каждый открытый вопрос (10 баллов) оценивается по трём аспектам:

– **Полнота ответа** (0–4 балла): раскрыты ли все составляющие вопроса.

– **Корректность формулировок** (0–4 балла): нет фактических ошибок, термины использованы правильно.

– **Наличие собственного примера или пояснения своими словами** (0–2 балла): ответ не сводится к пересказу определения, показывает понимание, а не заучивание.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

**Критерии оценивания практической части (часть 3)**

Каждое задание (15 баллов) оценивается так:

– Правильный итоговый ответ/результат — 6 баллов.

– Верное пошаговое объяснение хода рассуждения — 6 баллов.

– Точное использование терминологии курса — 3 балла.

**Шкала перевода суммарных баллов в оценку**

| Баллы | Оценка |
|---|---|
| 90–100 | «отлично» |
| 75–89 | «хорошо» |
| 60–74 | «удовлетворительно» |
| менее 60 | «неудовлетворительно» |

**Дополнительное правило**

Если студент набрал менее 60 баллов, но верно выполнил обе задачи практической части (часть 3) полностью, преподаватель вправе засчитать работу на «удовлетворительно» — это показывает, что студент способен применять знания на практике, даже если испытывает затруднения с теоретическими формулировками. Обратная ситуация (высокий балл за теорию при нулевом результате в практической части) не даёт права на повышение оценки: практическая часть обязательна к выполнению.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

## Контрольная работа №3. Сети и сокеты

Работа завершает цикл 3 «Сети и сокеты» и проверяет усвоение материала лекций 3.1–3.3:

– 3.1 Сокеты как абстракция операционной системы

– 3.2 Сокеты в Linux на практике: netcat, ss и unix-сокеты

– 3.3 Сетевые демоны и systemd socket activation

Цикл 3 — ознакомительный: подробное изучение протоколов, модели OSI, IP-адресации и маршрутизации остаётся за отдельной дисциплиной «Компьютерные сети». Эта работа проверяет только материал, изложенный в данном курсе — сокет как абстракцию ОС и то, как ОС управляет сетевыми службами.

**Цель работы**

Проверить, что студент способен: (1) объяснить, чем сокет похож на файловый дескриптор и чем от него отличается; (2) различать модель клиент-сервер, TCP и UDP на концептуальном уровне; (3) читать вывод `ss` и понимать состояния `LISTEN`/`ESTABLISHED`; (4) отличать сетевой сокет от unix-domain сокета; (5) объяснить механизм socket activation и назначение `.socket`-юнита.

**Формат проведения**

Письменная работа, 30–45 минут (короче предыдущих, соразмерно объёму цикла), без использования конспектов и интернета. Работа состоит из четырёх частей: тестовая часть, часть с открытыми вопросами, практическая часть (чтение вывода команд/конфигов) и часть на сопоставление понятий.

Ниже приведён один из вариантов работы. Преподаватель может подготовить 2–4 равноценных по сложности варианта.

**Часть 1. Тестовые вопросы с одним правильным ответом (10 вопросов × 2 балла = 20 баллов)**

1. Чем сокет похож на обычный файловый дескриптор?
   а) он тоже хранится на диске
   б) программа работает с ним через те же операции чтения и записи
   в) он всегда доступен только процессу root
   г) он автоматически шифрует передаваемые данные
2. Из чего состоит адрес обычного (сетевого) сокета?
   а) только из IP-адреса
   б) только из номера порта
   в) из пары «IP-адрес + номер порта»
   г) из имени процесса
3. Какой протокол устанавливает соединение перед обменом данными и гарантирует порядок доставки?
   а) UDP
   б) TCP
   в) IP
   г) DNS



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

4. Что означает состояние `LISTEN` в выводе команды `ss`?
   а) соединение уже установлено и идёт обмен данными
   б) сервер ждёт входящих подключений, разговор ещё не начался
   в) сокет закрыт и будет удалён
   г) произошла ошибка сети
5. Чем адресуется unix-domain сокет?
   а) MAC-адресом сетевой карты
   б) путём в файловой системе
   в) парой «IP-адрес + порт»
   г) именем пользователя
6. Какая современная утилита в Linux используется вместо устаревающей `netstat`?
   а) `ping`
   б) `ss`
   в) `top`
   г) `dig`
7. Что означает ошибка «Address already in use»?
   а) на диске закончилось место
   б) указанный порт на этом адресе уже слушает другой процесс
   в) у пользователя нет прав на подключение к интернету
   г) введён неправильный пароль
8. Что делает юнит `.socket` в системе socket activation?
   а) хранит журнал сетевых подключений
   б) сам слушает указанный порт вместо службы, до момента её запуска
   в) полностью заменяет собой службу и никогда не запускает `.service`
   г) шифрует трафик связанной службы
9. Что произойдёт, если обычный пользователь (без прав администратора) попытается запустить процесс, слушающий порт 80?
   а) это стандартная операция, доступная любому пользователю
   б) операция будет отклонена, так как порты ниже 1024 привилегированы
   в) порт 80 автоматически будет заменён на порт выше 1024
   г) система выдаст предупреждение, но всё равно выполнит операцию
10. В чём практическое преимущество socket activation по сравнению с постоянно запущенной службой?
    а) служба начинает работать быстрее при каждом запросе, чем обычный процесс
    б) служба не расходует ресурсы, пока к ней никто не подключается
    в) socket activation полностью исключает любые сетевые ошибки
    г) это единственный способ вообще открыть сетевой порт в Linux

**Ключ:** 1-б, 2-в, 3-б, 4-б, 5-б, 6-б, 7-б, 8-б, 9-б, 10-б



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

**Часть 2. Открытые вопросы (3 вопроса × 12 баллов = 36 баллов)**

1. Объясните, в чём заключается сходство между сокетом и файловым дескриптором и в чём принципиальное различие. Приведите собственный пример (не из лекции), иллюстрирующий это сходство.
2. Сравните TCP и UDP на уровне того, что получает программист при выборе каждого из них (без описания заголовков пакетов). Приведите по одному примеру задачи, для которой лучше подходит каждый из протоколов.
3. Опишите пошагово, что происходит при socket activation — от момента загрузки системы до момента, когда служба реально начинает обрабатывать данные подключившегося клиента. Объясните, почему это позволяет экономить ресурсы по сравнению с постоянно запущенной службой.

**Часть 3. Практическая часть (2 задания × 17 баллов = 34 балла)**

**Задание 1.** Дан фрагмент вывода команды `ss -tlnp`:

```
State   Local Address:Port   Process
LISTEN  0.0.0.0:22           users:(("sshd",pid=612,fd=3))
LISTEN  127.0.0.1:8080       users:(("python3",pid=9345,fd=4))
```

Ответьте: 1) какой процесс и с каким PID слушает порт 22, и о каком классе службы это обычно говорит; 2) чем отличается адрес `0.0.0.0:22` от адреса `127.0.0.1:8080` — что означает разница в первой части адреса; 3) если запустить ещё один процесс с командой, пытающейся слушать порт 8080, что произойдёт и почему.

**Задание 2.** Дан фрагмент unit-файла `mysvc.socket`:

```
[Unit]
Description=Тестовый сокет

[Socket]
ListenStream=9090

[Install]
WantedBy=sockets.target
```

Рядом лежит файл `mysvc.service` с секцией `[Service]`, использующей `ExecStart=/usr/local/bin/handler.sh`. Ответьте: 1) какой файл в этой паре включает и запускает администратор командой `systemctl --user enable --now`, и почему именно его, а не `mysvc.service` напрямую; 2) в каком состоянии будет находиться `mysvc.service` сразу после включения `mysvc.socket`, пока никто не подключился к порту 9090; 3) что изменится в состоянии `mysvc.service`, как только придёт первое подключение на порт 9090.

**Часть 4. Сопоставление понятий (10 баллов)**

Сопоставьте термин слева с его определением справа (запишите пары вида «1-В»):

| № | Термин | Буква | Определение |
|---|---|---|---|
| 1 | Сокет | А | Директива unit-файла, задающая порт или путь unix-сокета для прослушивания |
| 2 | `ESTABLISHED` | Б | Конечная точка канала связи между двумя процессами, |



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

| № | Термин | Буква | Определение |
|---|---|---|---|
| | | | разновидность дескриптора |
| 3 | Unix-domain сокет | В | Состояние сокета, когда соединение установлено и идёт обмен данными |
| 4 | `ListenStream=` | Г | Сокет, адресуемый путём в файловой системе, для связи процессов одной машины |
| 5 | Socket activation | Д | Механизм, при котором systemd сам слушает порт и запускает службу по требованию |

**Ключ:** 1-Б, 2-В, 3-Г, 4-А, 5-Д

**Критерии оценивания**

Максимум за работу — **100 баллов**, распределённых по частям:

| Часть | Баллы |
|---|---|
| Часть 1. Тестовые вопросы (10 × 2) | 20 |
| Часть 2. Открытые вопросы (3 × 12) | 36 |
| Часть 3. Практическая часть (2 × 17) | 34 |
| Часть 4. Сопоставление понятий | 10 |
| **Итого** | **100** |

**Критерии оценивания открытых вопросов (часть 2)**

Каждый открытый вопрос (12 баллов) оценивается по трём аспектам:

– **Полнота ответа** (0–5 баллов): раскрыты ли все составляющие вопроса.

– **Корректность формулировок** (0–5 баллов): нет фактических ошибок, термины использованы правильно.

– **Наличие собственного примера или пояснения своими словами** (0–2 балла): ответ не сводится к пересказу определения.

**Критерии оценивания практической части (часть 3)**

Каждое задание (17 баллов) оценивается так:

– Правильный итоговый ответ/результат — 7 баллов.

– Верное пошаговое объяснение хода рассуждения — 7 баллов.

– Точное использование терминологии курса — 3 балла.

**Шкала перевода суммарных баллов в оценку**

| Баллы | Оценка |
|---|---|
| 90–100 | «отлично» |
| 75–89 | «хорошо» |
| 60–74 | «удовлетворительно» |
| менее 60 | «неудовлетворительно» |

**Дополнительное правило**

Если студент набрал менее 60 баллов, но верно выполнил оба задания практической части (часть 3) полностью, преподаватель вправе засчитать работу на «удовлетворительно». Обратная ситуация (высокий балл за теорию при нулевом



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

результате в практической части) не даёт права на повышение оценки: практическая часть обязательна к выполнению.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

## Контрольная работа №4. Виртуализация и контейнеризация

Работа завершает цикл 4 «Виртуализация и контейнеризация» и проверяет усвоение материала лекций 4.1–4.6:

– 4.1 Виртуализация: полная виртуализация, паравиртуализация и гипервизоры
– 4.2 Практика: создание и настройка виртуальной машины
– 4.3 Контейнеризация изнутри: namespaces и cgroups
– 4.4 Docker: образы, контейнеры, Dockerfile
– 4.5 Docker Compose и многоконтейнерные приложения
– 4.6 Виртуализация и контейнеризация: сравнение, паравиртуализация virtio на практике

**Цель работы**

Проверить, что студент способен: (1) различать гипервизоры типа 1 и типа 2, полную виртуализацию и паравиртуализацию; (2) назвать составные части виртуальной машины и объяснить назначение снапшотов и сетевых режимов; (3) объяснить разницу между виртуальной машиной и контейнером через namespaces и cgroups; (4) читать и писать простой Dockerfile, различать инструкции `RUN` и `CMD`; (5) читать `docker-compose.yml` и объяснить, как сервисы обращаются друг к другу; (6) аргументированно выбирать между виртуализацией и контейнеризацией под конкретную задачу.

**Формат проведения**

Письменная работа, 45–60 минут, без использования конспектов и интернета (разрешён только чистый лист для черновика). Работа состоит из четырёх частей: тестовая часть, часть с открытыми вопросами, практическая часть (чтение конфигов) и часть на сопоставление понятий.

Ниже приведён один из вариантов работы. Преподаватель может подготовить 2–4 равноценных по сложности варианта.

**Часть 1. Тестовые вопросы с одним правильным ответом (10 вопросов × 2 балла = 20 баллов)**

1. Чем гипервизор типа 1 принципиально отличается от гипервизора типа 2?
   а) типом процессора, на котором может работать
   б) тем, что работает прямо на аппаратуре, а не поверх обычной ОС
   в) поддержкой только одной гостевой ОС одновременно
   г) отсутствием необходимости в драйверах
2. Что отличает паравиртуализацию от полной виртуализации?
   а) гостевая ОС модифицирована и напрямую обращается к гипервизору через hypercall
   б) паравиртуализация вообще не требует гипервизора
   в) паравиртуализация работает только с проприетарными ОС
   г) паравиртуализация полностью эмулирует физический процессор
3. Для чего нужна аппаратная поддержка виртуализации в процессоре (VT-x/AMD-V)?



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

а) для увеличения объёма оперативной памяти
б) для ускорения перехвата и обработки привилегированных инструкций гостя
в) для подключения нескольких мониторов
г) для шифрования сетевого трафика

4. Что произойдёт с данными внутри виртуальной машины, если откатить её к ранее сделанному снапшоту?
а) данные останутся в том состоянии, в котором были на момент снапшота
б) все данные будут безвозвратно удалены
в) виртуальный диск будет автоматически расширен
г) виртуальная машина будет пересоздана с нуля

5. В чём принципиальное отличие контейнера от виртуальной машины?
а) контейнер — это процесс на общем с хостом ядре, а не отдельная ОС с собственным ядром
б) контейнер не может обращаться к сети
в) контейнер всегда работает медленнее виртуальной машины
г) контейнер не поддерживает изоляцию процессов вообще

6. Что из перечисленного ограничивают cgroups?
а) какие файлы процесс может видеть
б) сколько ресурсов (CPU, память) процесс может потребить
в) под каким именем пользователя работает процесс
г) к какой сети подключён процесс

7. Чем отличается инструкция `RUN` от инструкции `CMD` в Dockerfile?
а) `RUN` выполняется при сборке образа, `CMD` — при запуске контейнера
б) это два разных названия одной и той же инструкции
в) `CMD` можно использовать только один раз на весь Dockerfile, а `RUN` — сколько угодно
г) `RUN` выполняется на хосте, а `CMD` — внутри виртуальной машины

8. Как сервисы, описанные в одном `docker-compose.yml`, обычно обращаются друг к другу?
а) только по IP-адресу, назначенному вручную
б) по имени сервиса, как по доменному имени во внутренней сети Compose
в) через файл на диске, общий для обоих контейнеров
г) обращение между сервисами одного файла Compose невозможно

9. Что случится с данными внутри контейнера при его удалении, если для этих данных НЕ был настроен том (volume)?
а) данные будут сохранены автоматически на хосте
б) данные будут потеряны вместе с контейнером
в) данные автоматически переместятся в новый контейнер
г) Docker запросит подтверждение перед удалением данных

10. Зачем в полностью виртуализированной машине используют паравиртуализационные драйверы устройств (virtio)?



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

а) потому что без них виртуальная машина вообще не запустится
б) потому что честная эмуляция настоящего физического устройства медленнее оптимизированного протокола обмена с хостом
в) потому что virtio заменяет собой гипервизор
г) потому что это единственный способ подключить виртуальную машину к сети

**Ключ:** 1-б, 2-а, 3-б, 4-а, 5-а, 6-б, 7-а, 8-б, 9-б, 10-б

**Часть 2. Открытые вопросы (4 вопроса × 10 баллов = 40 баллов)**

1. Объясните, почему процессор с аппаратной поддержкой виртуализации (VT-x/AMD-V) снижает потребность в паравиртуализации для самого процессора, но не устраняет потребность в паравиртуализационных драйверах устройств (virtio). Приведите свой пример устройства, для которого это особенно заметно.
2. Опишите своими словами разницу между namespaces и cgroups: что изолирует каждый из этих механизмов и почему их разделяют на два независимых механизма, а не объединяют в один.
3. На примере вымышленного Dockerfile с инструкциями `FROM`, `COPY`, `RUN pip install ...` и `CMD ["python", "app.py"]` объясните, что происходит на этапе `docker build`, а что — только при `docker run`, и почему это разграничение важно с точки зрения кэширования слоёв образа.
4. Опишите сценарий (не из лекции), в котором вы бы выбрали виртуальную машину вместо контейнера, и сценарий, в котором выбрали бы контейнер вместо виртуальной машины. Обоснуйте оба выбора через конкретные свойства (изоляция, скорость запуска, плотность размещения).

**Часть 3. Практическая часть (2 задания × 15 баллов = 30 баллов)**

**Задание 1.** Дан фрагмент `Dockerfile`:

```
FROM python:3-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install -r requirements.txt
COPY . .
CMD ["python", "app.py"]
```

Ответьте: 1) в каком порядке будут выполнены эти инструкции и что из них создаёт отдельный слой образа; 2) почему `COPY requirements.txt .` и `RUN pip install` стоят раньше, чем `COPY . .` со всем остальным кодом — какое практическое преимущество это даёт при повторной сборке образа после изменения кода приложения (но не зависимостей); 3) что произойдёт, если выполнить `docker run` собранного образа без каких-либо дополнительных аргументов.

**Задание 2.** Дан фрагмент `docker-compose.yml`:

```
services:
  web:
    build: .
```



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

```
    ports:
      - "8080:80"
    depends_on:
      - cache
  cache:
    image: redis:alpine
```

Ответьте: 1) какой порт хоста нужно открыть в браузере, чтобы обратиться к сервису `web`, и на какой порт внутри контейнера это обращение попадёт; 2) как сервис `web` должен обращаться к сервису `cache` изнутри своего контейнера (по какому адресу/имени); 3) что означает `depends_on` в данном фрагменте и что оно НЕ гарантирует (подсказка: порядок запуска процесса — это не то же самое, что готовность службы принимать соединения).

**Часть 4. Сопоставление понятий (10 баллов)**

Сопоставьте термин слева с его определением справа (запишите пары вида «1-В»):

| № | Термин | Буква | Определение |
|---|---|---|---|
| 1 | Namespace | А | Формат файла-образа виртуального диска с поддержкой снапшотов |
| 2 | Cgroups | Б | Механизм ядра Linux, ограничивающий видимость процесса (например, список других процессов) |
| 3 | qcow2 | В | Механизм ядра Linux, ограничивающий потребление ресурсов процессом |
| 4 | Docker-образ | Г | Неизменный шаблон файловой системы и метаданных, из которого запускается контейнер |
| 5 | Docker Compose | Д | Инструмент декларативного описания и запуска нескольких связанных контейнеров |

**Ключ:** 1-Б, 2-В, 3-А, 4-Г, 5-Д

**Критерии оценивания**

Максимум за работу — **100 баллов**, распределённых по частям:

| Часть | Баллы |
|---|---|
| Часть 1. Тестовые вопросы (10 × 2) | 20 |
| Часть 2. Открытые вопросы (4 × 10) | 40 |
| Часть 3. Практическая часть (2 × 15) | 30 |
| Часть 4. Сопоставление понятий | 10 |
| **Итого** | **100** |

**Критерии оценивания открытых вопросов (часть 2)**

Каждый открытый вопрос (10 баллов) оценивается по трём аспектам:

– **Полнота ответа** (0–4 балла): раскрыты ли все составляющие вопроса.

– **Корректность формулировок** (0–4 балла): нет фактических ошибок, термины использованы правильно.

– **Наличие собственного примера или пояснения своими словами** (0–2 балла): ответ не сводится к пересказу определения.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

**Критерии оценивания практической части (часть 3)**

Каждое задание (15 баллов) оценивается так:

– Правильный итоговый ответ/результат — 6 баллов.

– Верное пошаговое объяснение хода рассуждения — 6 баллов.

– Точное использование терминологии курса — 3 балла.

**Шкала перевода суммарных баллов в оценку**

| Баллы | Оценка |
|---|---|
| 90–100 | «отлично» |
| 75–89 | «хорошо» |
| 60–74 | «удовлетворительно» |
| менее 60 | «неудовлетворительно» |

**Дополнительное правило**

Если студент набрал менее 60 баллов, но верно выполнил обе задачи практической части (часть 3) полностью, преподаватель вправе засчитать работу на «удовлетворительно». Обратная ситуация (высокий балл за теорию при нулевом результате в практической части) не даёт права на повышение оценки: практическая часть обязательна к выполнению.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

## Контрольная работа №5. Автоматизация на Python

Работа завершает цикл 5 «Автоматизация на Python» — последний тематический цикл курса — и проверяет усвоение материала лекций 5.1–5.5:

– 5.1 Python как инструмент системного администратора: os, pathlib, права доступа

– 5.2 Процессы из Python: subprocess

– 5.3 Сокеты на Python: клиент-сервер

– 5.4 Автоматизация Docker и systemd из Python

– 5.5 Планирование задач и логирование в автоматизации

**Цель работы**

Проверить, что студент способен: (1) использовать `os` и `pathlib` для работы с файлами, каталогами и правами доступа программно; (2) запускать и контролировать внешние процессы через `subprocess`, избегая уязвимостей вроде command injection; (3) написать минимальный клиент-серверный обмен на модуле `socket`; (4) автоматизировать управление Docker и systemd из Python; (5) правильно организовать планирование и журналирование скриптов автоматизации.

**Формат проведения**

Письменная работа, 45–60 минут, без использования конспектов и интернета (разрешён только чистый лист для черновика). Работа состоит из четырёх частей: тестовая часть, часть с открытыми вопросами, практическая часть (чтение и анализ кода) и часть на сопоставление понятий.

Ниже приведён один из вариантов работы. Преподаватель может подготовить 2–4 равноценных по сложности варианта.

**Часть 1. Тестовые вопросы с одним правильным ответом (10 вопросов × 2 балла = 20 баллов)**

1. Что вернёт `os.environ.get("DEBUG", "0")`, если переменной окружения `DEBUG` не существует?
   а) вызовет исключение KeyError
   б) вернёт строку "0"
   в) вернёт None
   г) остановит выполнение скрипта
2. Что делает выражение `Path("/etc") / "passwd"`?
   а) выполняет числовое деление
   б) удаляет файл passwd
   в) собирает путь /etc/passwd, используя перегруженный оператор класса Path
   г) проверяет права доступа к файлу
3. Почему `os.chmod("file.sh", 755)` без префикса `0o` — ошибка по смыслу?
   а) Python выдаст синтаксическую ошибку
   б) 755 будет воспринято как десятичное число, а не восьмеричные права
   в) chmod не принимает числовые аргументы вообще
   г) это вызовет удаление файла



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

4. Что из перечисленного считается безопасным способом передачи пользовательского ввода в subprocess?
   а) `subprocess.run("ls " + user_input, shell=True)`
   б) `subprocess.run(["ls", user_input])`
   в) `os.system("ls " + user_input)`
   г) оба способа одинаково безопасны
5. Что означает `result.returncode == 0` после `subprocess.run()`?
   а) команда завершилась с ошибкой
   б) команда всё ещё выполняется
   в) команда завершилась успешно
   г) команда не вывела ничего на экран
6. В каком порядке серверный код на socket вызывает свои методы?
   а) connect → bind → listen → accept
   б) socket → bind → listen → accept
   в) socket → accept → bind → listen
   г) bind → socket → accept → listen
7. Зачем перед отправкой строки через сокет нужно вызывать `.encode()`?
   а) чтобы зашифровать сообщение
   б) сокет передаёт только байты, а не строки Python
   в) чтобы сжать данные
   г) это необязательный шаг
8. К чему напрямую обращается Docker SDK for Python при работе с локальным демоном?
   а) к TCP-порту 80
   б) к unix-сокету /var/run/docker.sock
   в) к файлу /etc/docker.conf
   г) к серверу Docker Hub через интернет
9. Почему в crontab рекомендуется указывать полный путь к интерпретатору python3?
   а) короткое имя работает медленнее
   б) у задач cron урезанное окружение, и PATH может не содержать нужный каталог
   в) cron не поддерживает короткие имена команд синтаксически
   г) это требование только для системных, а не пользовательских заданий cron
10. Почему `except: pass` считается опасным антипаттерном в автоматизации?
    а) это вызывает синтаксическую ошибку
    б) он полностью проглатывает ошибку, делая проблему необнаружимой
    в) он значительно замедляет работу скрипта
    г) он автоматически перезагружает систему

**Ключ:** 1-б, 2-в, 3-б, 4-б, 5-в, 6-б, 7-б, 8-б, 9-б, 10-б



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

**Часть 2. Открытые вопросы (4 вопроса × 10 баллов = 40 баллов)**

1. Объясните, чем `pathlib.Path` удобнее более старого `os.path`, и приведите собственный пример (не из лекции) задачи, где эта разница особенно заметна.
2. Опишите, как модуль `subprocess` под капотом связан с системными вызовами `fork`/`exec`/`wait` из лекции 2.2. Объясните разницу между `subprocess.run()` и `subprocess.Popen` с точки зрения того, когда скрипт получает управление обратно.
3. Сравните два способа автоматизации Docker из Python — через `subprocess` и через Docker SDK. В каком случае вы бы выбрали каждый из них, и почему для systemd аналогичного выбора между «CLI-обёрткой» и «SDK» не существует?
4. Объясните, почему `print()` — плохой инструмент журналирования для скрипта, запускаемого по расписанию, и опишите, как правильно организованный `logging` вместе с `try/except` решает эту проблему. Приведите пример ситуации, где `except: pass` скрыл бы серьёзную проблему.

**Часть 3. Практическая часть (2 задания × 15 баллов = 30 баллов)**

**Задание 1.** Дан фрагмент кода:

```
import subprocess


result = subprocess.run(
    ["systemctl", "--user", "is-active", "myapp.service"],
    capture_output=True, text=True
)
if result.returncode != 0:
    subprocess.run(["systemctl", "--user", "restart", "myapp.service"],
check=True)
```

Ответьте: 1) что произойдёт, если служба `myapp.service` активна на момент запуска скрипта; 2) что произойдёт, если она неактивна; 3) что случится, если сама команда `restart` завершится с ошибкой — почему именно так, учитывая параметр `check=True`.

**Задание 2.** Дан фрагмент серверного кода на Python:

```
import socket


HOST, PORT = "127.0.0.1", 9090
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((HOST, PORT))
server.listen()
conn, addr = server.accept()
data = conn.recv(1024)
conn.send(data)
conn.close()
```



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

Ответьте: 1) что произойдёт со скриптом на строке с `.accept()`, если клиент ещё не подключился; 2) почему `conn` — это НЕ тот же сокет, что `server`, и что произойдёт с сокетом `server` после обработки этого клиента; 3) какую ошибку получит второй экземпляр этого же скрипта, запущенный, пока первый ещё слушает порт 9090, и на каком именно вызове она возникнет.

**Часть 4. Сопоставление понятий (10 баллов)**

Сопоставьте термин слева с его определением справа (запишите пары вида «1-В»):

| № | Термин | Буква | Определение |
|---|---|---|---|
| 1 | `.glob("**/*.sh")` | А | Рекурсивный поиск файлов по маске во всех вложенных подкаталогах |
| 2 | `command injection` | Б | Уязвимость, возникающая при склейке пользовательского ввода в строку с shell=True |
| 3 | `.accept()` | В | Метод, блокирующий выполнение сервера до прихода нового подключения |
| 4 | `docker.from_env()` | Г | Создание клиента Docker SDK, подключающегося к локальному демону |
| 5 | `logging.exception()` | Д | Запись сообщения в журнал вместе с полным traceback ошибки |

**Ключ:** 1-А, 2-Б, 3-В, 4-Г, 5-Д

**Критерии оценивания**

Максимум за работу — **100 баллов**, распределённых по частям:

| Часть | Баллы |
|---|---|
| Часть 1. Тестовые вопросы (10 × 2) | 20 |
| Часть 2. Открытые вопросы (4 × 10) | 40 |
| Часть 3. Практическая часть (2 × 15) | 30 |
| Часть 4. Сопоставление понятий | 10 |
| **Итого** | **100** |

**Критерии оценивания открытых вопросов (часть 2)**

Каждый открытый вопрос (10 баллов) оценивается по трём аспектам:

– **Полнота ответа** (0–4 балла): раскрыты ли все составляющие вопроса.

– **Корректность формулировок** (0–4 балла): нет фактических ошибок, термины использованы правильно.

– **Наличие собственного примера или пояснения своими словами** (0–2 балла): ответ не сводится к пересказу определения.

**Критерии оценивания практической части (часть 3)**

Каждое задание (15 баллов) оценивается так:

– Правильный итоговый ответ/результат — 6 баллов.

– Верное пошаговое объяснение хода рассуждения — 6 баллов.

– Точное использование терминологии курса — 3 балла.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

**Шкала перевода суммарных баллов в оценку**

| Баллы | Оценка |
|---|---|
| 90–100 | «отлично» |
| 75–89 | «хорошо» |
| 60–74 | «удовлетворительно» |
| менее 60 | «неудовлетворительно» |

**Дополнительное правило**

Если студент набрал менее 60 баллов, но верно выполнил обе задачи практической части (часть 3) полностью, преподаватель вправе засчитать работу на «удовлетворительно». Обратная ситуация (высокий балл за теорию при нулевом результате в практической части) не даёт права на повышение оценки: практическая часть обязательна к выполнению.

### 5.3.4 Лабораторные и практические работы

Лабораторные работы выполняются в лаборатории на виртуальных машинах с Linux и (для лабораторной работы №4) Windows. Методические указания к работам размещены в курсе Moodle. По итогам каждой работы обучающийся оформляет отчёт (скриншоты и вывод команд с пояснениями) и сдаёт его преподавателю; сценарии и конфигурации размещаются в личном репозитории GitVerse.

**Перечень лабораторных работ (48 часов, 24 работы по 2 часа)**

| № | Лабораторная работа | Цель работы | Источник |
|---|---|---|---|
| 1 | Режимы процессора, роль ОС и аппаратное обеспечение компьютера средствами Linux | Увидеть на практике режимы работы процессора, роль ОС между программами и оборудованием; получить сведения об аппаратуре ПК средствами Linux. | Moodle: ЛР 1.1, ЛР 1.2 |
| 2 | Процессы, файлы и оболочка на практике | Освоить работу с процессами, файлами и командной оболочкой на практике: наблюдать их в работающей системе. | Moodle: ЛР 1.3 |
| 3 | Системные вызовы на практике: strace и fork | Наблюдать системные вызовы с помощью strace; создать процесс вызовом fork() в простой программе на Си. | Moodle: ЛР 1.4 |
| 4 | Администрирование Windows: PowerShell, учётные записи, службы и журналы событий | Освоить базовые приёмы администрирования Windows в PowerShell: сведения о системе, учётные записи, службы, журналы событий; сопоставить с Linux. | новая работа |
| 5 | Структура ядра Linux: модули и архитектурные решения | Исследовать структуру ядра Linux штатными командами, работу с модулями ядра; закрепить понимание монолитной и микроядерной архитектур. | Moodle: ЛР 1.5 |
| 6 | Первая программа на языке Си: указатели и макросы | Написать, скомпилировать и запустить первые программы на Си (gcc); освоить указатели и макросы. | Moodle: ЛР 1.6 |
| 7 | Пользователи, группы и права доступа | Определять учётную запись и группы, читать и изменять права доступа (chmod, chown, umask), изучить SUID, SGID и sticky bit. | Moodle: ЛР 2.1 |
| 8 | Процессы и сигналы в Linux | Наблюдать состояния процессов, управлять ими, отправлять и обрабатывать сигналы; получить зомби-процесс и устранить его. | Moodle: ЛР 2.2 |
| 9 | systemd: юниты, сервисы и | Исследовать работу systemd: создать собственный | Moodle: ЛР |



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

| № | Лабораторная работа | Цель работы | Источник |
|---|---|---|---|
| | таргеты | unit-файл, управлять сервисами, таргетами и автозапуском. | 2.3 |
| 10 | Демоны и системное логирование | Отличать демоны от интерактивных процессов, работать с журналами (journalctl, rsyslog), настраивать уровни и ротацию журналов. | Moodle: ЛР 2.4 |
| 11 | Файловая система Linux: FHS, /proc и /sys | Ориентироваться в иерархии каталогов FHS, работать с inode и ссылками, изучить /proc и /sys, монтировать файловые системы. | Moodle: ЛР 2.5 |
| 12 | Автоматизация в shell: скрипты, cron и таймеры systemd | Написать сценарий Bash и поставить его на расписание через cron и через таймер systemd; сравнить подходы. | Moodle: ЛР 2.6 |
| 13 | Сокеты в Linux на практике: netcat, ss и unix-сокеты | Работать с сокетами на практике: netcat, ss, unix-сокеты; диагностировать открытые порты и соединения. | Moodle: ЛР 3.1, ЛР 3.2 |
| 14 | Сетевые демоны и systemd socket activation | Настроить сетевой демон и активацию службы по сокету (systemd socket activation). | Moodle: ЛР 3.3 |
| 15 | Виртуализация: гипервизоры, создание и настройка виртуальной машины | Определить поддержку аппаратной виртуализации, создать и настроить виртуальную машину, сделать снимок состояния. | Moodle: ЛР 4.1, ЛР 4.2 |
| 16 | Контейнеризация изнутри: namespaces и cgroups | Создать изолированный процесс с помощью namespaces и ограничить его ресурсы через cgroups. | Moodle: ЛР 4.3 |
| 17 | Docker: образы, контейнеры, Dockerfile | Собрать образ по Dockerfile, запустить контейнер, управлять образами, контейнерами и томами. | Moodle: ЛР 4.4 |
| 18 | Docker Compose и многоконтейнерные приложения | Описать многоконтейнерное приложение в docker-compose.yml (PHP-приложение и база данных) и запустить его одной командой. | Moodle: ЛР 4.5 |
| 19 | Сравнение виртуализации и контейнеризации, паравиртуализация virtio | Сравнить виртуальные машины и контейнеры по изоляции и производительности; изучить паравиртуализацию virtio. | Moodle: ЛР 4.6 |
| 20 | Python как инструмент администратора: os, pathlib, права доступа, subprocess | Автоматизировать работу с файлами, правами и процессами на Python (os, pathlib, subprocess). | Moodle: ЛР 5.1, ЛР 5.2 |
| 21 | Сокеты на Python: клиент и сервер | Написать на Python TCP-клиент и сервер; сопоставить их с утилитами netcat и ss. | Moodle: ЛР 5.3 |
| 22 | Автоматизация Docker и systemd из Python | Управлять Docker и службами systemd из сценариев Python. | Moodle: ЛР 5.4 |
| 23 | Планирование задач и логирование в автоматизации | Использовать модуль logging, запланировать сценарий на выполнение по расписанию и обеспечить обработку ошибок. | Moodle: ЛР 5.5 |
| 24 | Российские ИИ-ассистенты (GigaChat, Алиса AI) в администрировании: генерация, проверка и верификация сценариев | Составлять корректные запросы к GigaChat и Алиса AI, проверять результат в изолированной среде, находить в нём ошибки и небезопасные решения, оформлять итог в репозитории. | новая работа |

## Критерии оценки лабораторной работы

| Оценка | Показатели |
|---|---|
| «5» | Все шаги выполнены самостоятельно и верно; отчёт оформлен полностью; обучающийся объясняет назначение каждой команды и результат; сценарии размещены в репозитории; при использовании ИИ-ассистента представлены запросы, найденные недостатки ответа и исправленный вариант. |
| «4» | Работа выполнена полностью, есть неточности в отчёте или пояснениях; |



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

| Оценка | Показатели |
|---|---|
| | обучающийся устраняет замечания после наводящих вопросов. |
| «3» | Работа выполнена не полностью (не менее 70 % шагов) или с существенными ошибками; пояснения неполные. |
| «2» | Работа не выполнена или выполнена менее чем на 70 %; обучающийся не может объяснить результаты. |

### 5.3.5 Компьютерные программы для контроля знаний студентов

Комплект тестов и банки вопросов по дисциплине размещены в системе дистанционного обучения колледжа (Moodle) и импортируются в формате Moodle XML. Контрольные работы выполняются письменно, экзамен проводится устно в лаборатории с использованием виртуальных машин для практического задания. Использование ИИ-ассистентов на контрольных работах и экзамене не допускается.



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

# Лист подписей

ГАПОУ СО «Нижнетагильский торгово-экономический колледж»
SHA-256 исходного документа: bb20a51118ffba4b4519a36351f259868ee3d2d1d566bcf7c99e492d38a04e9d
Документ: 70 страниц
Проверка подписей: https://sig.nttek.ru/v/D6QPG2Gg7ZuBSgtZdSYyNGhhfqhwLogTJv063lgOhjKGY9sR
Печатная форма сформирована: 13.10.2026 10:27

| № | Роль в подписи | Подписант | Сертификат | Дата и время подписания | Результат проверки |
|---|---|---|---|---|---|
| 1 | Утверждаю | Дмитриева Марина Георгиевна<br>Заместитель директора по учебной работе | № 7E9C18960643A37EF2C929CD1E513CCD, действителен с 08.10.2026 по 08.10.2027 | 13.10.2026 10:27:33 | **подпись действительна**<br>роль и должность подтверждены подписью |

Печатная форма не подписывается и служит для удобства чтения. Юридическую силу имеют исходный PDF-документ и файлы электронных подписей (.sig). Подпись — усиленная неквалифицированная электронная подпись (ч. 3 ст. 5 Федерального закона № 63-ФЗ), выданная удостоверяющим центром колледжа.
Время указано по часовому поясу колледжа (Asia/Yekaterinburg, UTC+5).