

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Косенко Сергей Михайлович  
Должность: ректор  
Дата подписания: 20.06.2025 08:42:52  
Уникальный программный ключ:  
e3a68f3eaa1e62674b54f4998099d3d6bfdfcf836

## Оценочные материалы для промежуточной аттестации по дисциплине

### Операционная система LINUX, 8-й семестр

|                             |                                             |
|-----------------------------|---------------------------------------------|
| Код, направление подготовки | 09.03.04 Программная инженерия              |
| Направленность (профиль)    | Программное обеспечение компьютерных систем |
| Форма обучения              | очная                                       |
| Кафедра-разработчик         | Автоматики и компьютерных систем            |
| Выпускающая кафедра         | Автоматики и компьютерных систем            |

### Типовые задания для контрольной работы:

- Создать новую файловую систему на виртуальном диске.
- Настроить монтирование файловой системы.
- Использовать жесткие и символические ссылки.
- Использовать команду ps для мониторинга процессов.
- Настроить планирование задач с помощью cron и systemd timers.
- Использовать sgroups для управления ресурсами.
- Настроить сетевой интерфейс.
- Использовать iptables для настройки брандмауэра.
- Мониторить сетевую активность с помощью tcpdump.
- Настроить SELinux или AppArmor для ограничения доступа к ресурсам.
- Использовать инструменты аудита безопасности для обнаружения вторжений.
- Создать собственный пакет (deb или rpm).
- Настроить локальный репозиторий.
- Управлять зависимостями пакетов.
- Создать Docker-образ для приложения.
- Написать Dockerfile для многоэтапной сборки.
- Использовать Docker Compose для оркестрации контейнеров.
- Настроить сборочную среду (GitLab CI или GitHub Actions).
- Автоматизировать тестирование и развертывание приложений.
- Реализовать стратегии выпуска обновлений и откат изменений.

### Типовые вопросы к зачету:

1. Опишите основные отличия Linux от других операционных систем.
2. Какие компоненты входят в архитектуру Linux?
3. Назовите основные дистрибутивы Linux и их сферы применения.
4. Чем отличается система инициализации systemd от SysV init?
5. Как философия Linux влияет на её использование в разработке?
6. Что такое FHS, и как оно определяет структуру файловой системы Linux?

7. Сравните файловые системы ext4, XFS и Btrfs. В каких случаях каждая из них используется?
8. Объясните разницу между жесткими и символическими ссылками.
9. Как Virtual File System (VFS) помогает в унификации работы с файловыми системами?
10. Какие команды используются для управления правами доступа к файлам?
11. Опишите модель процессов в Linux и их жизненный цикл.
12. Что такое cgroups, и как они используются для управления ресурсами?
13. Как namespaces обеспечивают изоляцию процессов? Приведите примеры применения.
14. Какие инструменты используются для мониторинга процессов в Linux?
15. Объясните механизм межпроцессного взаимодействия (IPC) в Linux.
16. Как настроить сетевой интерфейс в Linux? Приведите пример команды.
17. Чем отличаются iptables от nftables? Какие преимущества предоставляет nftables?
18. Какие сетевые службы можно настроить в Linux, и как это сделать?
19. Как использовать tcpdump для мониторинга сетевого трафика?
20. Объясните архитектуру сетевой подсистемы Linux.
21. Что такое SELinux, и как оно обеспечивает безопасность системы?
22. Чем AppArmor отличается от SELinux? В каких случаях используется каждый инструмент?
23. Какие инструменты можно использовать для анализа безопасности системы?
24. Как шифровать данные на уровне файловой системы в Linux?
25. Перечислите основные методы обнаружения вторжений в системе.
26. Сравните пакетные менеджеры apt, yum/dnf и rpm по функциональности и применению.
27. Опишите процесс создания собственного пакета (.deb или .rpm).
28. Как разрешать конфликты зависимостей при установке пакетов?
29. Что такое Makefile, и как он используется для сборки программного обеспечения?
30. Как организовать локальный репозиторий для управления пакетами?
31. Чем отличается KVM от Docker с точки зрения виртуализации?
32. Опишите процесс создания Docker-образа с использованием Dockerfile.
33. Что такое многоэтапная сборка (multi-stage build), и какие преимущества она даёт?
34. Как Kubernetes помогает оркестрации контейнеров? Приведите пример сценария использования.
35. Какие архитектурные паттерны применяются для контейнеризованных приложений?
36. Что такое CI/CD, и как оно реализуется в GitLab CI или GitHub Actions?
37. Объясните концепцию "Инфраструктура как код" (IaC). Какие инструменты используются для её реализации?
38. Как настроить мониторинг системы с использованием Prometheus или Grafana?
39. Какие стратегии обновления приложений применяются в production-среде (например, blue-green deployment)?
40. Как автоматизировать тестирование приложений при помощи CI/CD?