Билет на экзамене по дисциплине «Операционные системы семейства UNIX и сетевые технологии» состоит из двух теоретических и одного практического вопроса. Экзамен проводится письменно. Каждое из трех заданий экзамена оценивается в двадцать баллов.

При ответе на теоретические вопросы студент должен продемонстрировать не только достаточный уровень компетентности в рамках рассматриваемого вопроса, но и опыт личной работы в рамках темы вопроса. Оценивается демонстрация самоподготовки студента, выражающаяся в собственных суждениях, сведениях, почерпнутых в ходе самостоятельного изучения вопроса в литературе, технической документации, электронных ресурсах, форумах и так далее.

При решении практических заданий студент должен представить текст скрипта на языке программирования bash, пояснения алгоритма работы, а также описание всех параметров, которые принимает скрипт (если таковые присутствуют) и описание формата вывода скрипта (если таковой присутствует). В связи с письменным проведением экзамена, при проверке правильности решения практического задания, в первую очередь будет оцениваться грамотность алгоритма, знание команд Linux и Bash, некритичные опечатки, приводящие к неработоспособности скрипта допускаются.

***Теоретические вопросы к экзамену***

1. Понятие виртуальной сети. Виды виртуальных сетевых компонентов.
2. Трансляция сетевых адресов. Виды NAT.
3. Понятие сетевого сокета. Применение, виды, схема взаимодействия.
4. Блокирующие и неблокирующие сокеты.
5. Транспортные протоколы TCP и UDP. Принципы работы, сравнение.
6. Клиент-серверное взаимодействие.
7. Реализация сокетов в языке Python. Модуль socket.
8. Понятие программного потока. Процессы и потоки.
9. Асинхронное программирование. Основные понятия. Параллелизм и конкуррентность.
10. Блокирующие и неблокирующие операции.
11. Алгоритмы, ограниченные процессором и вводом-выводом. Основные характеристики, особенности выполнения и распараллеливания.
12. Особенности реализации многопоточности в Python. Модуль threading.
13. Особенности организации многопроцессорной программы в Python. Модуль multiprocessing.
14. Асинхронное программирование в Python. Использование asyncio.
15. Параллельное программирование. Достоинства и недостатки.
16. Понятие потокобезопасности. Причины, проблематика, способы обеспечения.
17. Алгоритм выполнения многопоточной программы. Блокировка потоков.
18. Доступ к общим ресурсам в многопоточной программе. Механизмы блокировки ресурсов модуля threading.
19. Работа с файловой системой в Python. Основные операции.
20. Понятие веб-технологий. Основные характеристики, история, назначение.
21. Программное обеспечение, используемое для веб-технологий. Виды, назначение, примеры.
22. Понятие URL: назначение, применение, состав.
23. Понятие веб-сервера. Цели, принцип работы.
24. Протокол HTTP. Принцип работы, назначение, основные понятия.
25. Виртуальные хосты. Применение, настройка.
26. Системы управления содержимым. Назначение, принципы работы.
27. Понятие прокси-сервера. Настройка сервера nginx.
28. Прикладные интерфейсы программирования. Назначение, виды, реализация.
29. Системы контроля версий. Общая характеристика.
30. Ветвление в системах контроля версий.
31. Основные команды Git.
32. Разрешение конфликтов слияния в СКВ Git.
33. Работа с удаленными репозиториями в СКВ.
34. Организация совместной разработки в СКВ. Методология GitFlow
35. SSH. Назначение, схема работы, преимущества, недостатки.
36. SSH-форвардинг. Назначение, принципы работы.
37. Контейнеризация программных приложений. Основные понятия, использование.
38. Оркестрация контейнеризованными приложениями.

## *Примерные практические задания к экзамену*

1. созданная нити должны распечатать десять строк текста.
2. Напишите простой эхо-сервер, использующий неблокирующие сокеты и клиент к нему.
3. Напишите простой многопоточный загрузчик URL. Список URL скрипт принимает как аргументы командной строки.
4. Реализуйте простой HTTP-клиент. Он принимает один параметр командной строки - URL. Клиент делает запрос по указанному URL и выдает тело ответа на терминал как текст.
5. Напишите программу, которая вычисляет число Пи при помощи ряда Эйлера. Количество потоков программы должно определяться параметром командной строки.
6. Дана функция calculate(x, y). Напишите программу, которая создает пул из 5 процессов и распределяет в этом пуле вычисление функции на промежутке х от 0 до 1 с шагом 0,1. у равняется 2 всегда.
7. Напишите программу, которая проверяет все числа от 0 на простоту и выводит простые числа на экран по мере нахождения. Числа должны проверяться в различных потоках (или процессах, по выбору студента) Программа должна работать до тех пор, пока ее не остановит пользователь.
8. Напишите программу, которая обходит все файлы в директории, переданной ей как параметр и выводит на экран имена тех, чей размер задан как второй параметр. Реализовать рекурсивный обход поддиректорий.
9. Напишите программу, которая выводит на экран список номеров открытых портов на данной машине. Использовать команду netstat.
10. Напишите программу, которая копирует файл с удаленного хоста в текущую папку по SSH. Имя файла и адрес хоста принимать как параметры.