Основная литература

● Э. Таненбаум. Современные операционные системы

● М. Кофлер. Linux. Полное руководство

● В. Олифер. Компьютерные сети. Принципы, технологии, протоколы

● Е. Немет. Unix и Linux.Руководство системного администратора.

Дополнительная литература

● Б.Уорд. Внутреннее устройство Linux

● Д. Н. Колисниченко. Командная строка Linux

● У. Шоттс. Командная строка Linux. Полное руководство

● Д. Тейлор. Сценарии командной оболочки. Linux, OS X и UNIX

● Д.Н. Колисниченко. Linux. От новичка к профессионалу

● Э. Свейгард. Автоматизация рутинных задач с помощью Python

● Э. Таненбаум. Компьютерные сети

Видео-ресурсы

● Канал о Linux и DevOps

● Linux Foundation Certified System Administrator

● Системное администрирование Linux

● The Linux Basics Course: Beginner to Sysadmin, Step by Step

Программное обеспечение

● Oracle VM VirtualBox

● Python 3 Anaconda

● Notepad++

● Git

● Google Chrome

**Вопросы к экзамену**

**Первый семестр (Темы 0-3)**

1. Понятие операционной системы, выполняемые ей функции.

2. Порядок загрузки компьютера и операционной системы.

3. История развития ОС.

4. Семейство операционных систем UNIX. Философия UNIX.

5. Сравнительная характеристика Linux и Windows.

6. Понятие системного вызова операционной системы.

7. Понятие виртуализации. Виртуальные машины.

8. Основные компоненты операционной системы. Разделение функций.

9. Понятие файловой системы. Функции, виды, характеристики.

10. Физические и логические диски. Разбиение жесткого диска для установки Linux.

11. Стандартная иерархия каталогов Linux.

12. Понятие ядра операционной системы. Пространство ядра и пользовательское пространство.

13. Командный интерпретатор операционной системы.

14. Управление оперативной памятью в современных ОС. Виртуальная память.

15. Работа ОС с внешними устройствами. Понятие драйвера устройства.

16. Многозадачность современных операционных систем.

17. Графический и текстовый интерфейс взаимодействия с ОС.

18. Источники установки программных приложений. Программные репозитории.

19. Способы установки программ в Linux.

20. Процессы в Linux. Функции, организация, управление.

21. Жизненный цикл процесса операционной системы.

22. Основные команды Linux для управления файлами.

23. Основные команды Linux для управления каталогами.

24. Основные команды Linux для управления файловыми системами.

25. Межпроцессное взаимодействие. Проблема и способы решения.

26. Понятие пакета в Linux. Менеджеры пакетов.

27. Основные команды Linux для управления пакетами.

28. Основные команды Linux для управления процессами.

29. Типы файлов в Linux. Ссылки.

30. Основные команды для управления текстовыми потоками.

31. Основные команды для просмотра и редактирования текстовых файлов.

32. Регулярные выражения.

33. Встроенные и внешние команды bash.

34. Основные конфигурационные файлы Linux.

35. Командный интерпретатор bash. Структура команды.

36. Bash. Переменные и типы.

37. Bash. Условия.

38. Bash. Циклы.

39. Bash. Функции.

40. Переменные окружения в bash.

41. Основные команды Linux для управления пользователями и группами.

42. Суперпользователь root. Характеристика, особенности, функции, опасность.

43. Система прав доступа Linux. Структура и смысл прав доступа.

44. Организация хранения паролей пользователей в Linux.

45. Основные команды Linux для управления правами доступа.

46. Понятие компьютерных сетей. Общие принципы, организация.

47. Понятие сетевого ресурса. Классификация.

48. Семейство сетевых протоколов TCP/IP.

49. Адресация компьютеров в сети. IP, MAC адреса.

50. Маршрутизация в сетях TCP/IP.

**Контроль**

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.

**Практические задания к экзамену:**

Первый семестр (Темы 0-3)

1. Написать скрипт на Bash, который должен принимать в качестве аргумента любую строку и удалять из /home/ файл с именем, соответствующим переданной строке без необходимости отвечать утвердительно на вопросы системы. Так же должны быть выполнены условия, при которых скрипт не удалит все файлы при передаче в качестве параметра «\*» и удалит файлы, имена которых начинаются с «-».

2. Написать скрипт на Bash, который должен принимать в качестве аргумента любую строку и удалять из /home/ каталог с именем, соответствующим переданной строке без необходимости отвечать утвердительно на вопросы системы.

3. Написать скрипт, который из /etc/passwd выведет домашнюю директорию для пользователя www-data.

4. Написать скрипт, который из /etc/passwd выведет название командного интерпретатора для пользователя, имя которого передано скрипту в качестве параметра.

5. Написать скрипт, который выведет top 5 процессов, которые потребляют больше всего памяти и процессора в системе.

6. Написать скрипт, который выведет первые 10 процессов, которые принадлежат пользователю, имя которого передано в качестве параметра.

7. Написать скрипт, который выведет всех потомков процесса по его PID.

8. Написать скрипт, который выведет все процессы, порожденные данным процессом по его PID.

9. Написать скрипт, который для всех файлов во временной директории, которые были созданы за текущий месяц, и не изменялись в течение недели при нахождении в них строки test поменяет ее на tset.

10. Написать скрипт, который для всех файлов во домашней директории, которые были созданы за текущий месяц, и изменялись в течение недели при нахождении в них строки test поменяет ее на tset.

11. Написать скрипт, которые выведет на экран все файлы, размер которых больше 50 Мб в директории /var/log

12. Написать скрипт, которые выведет на экран все файлы, размер которых больше 50 Мб в переданной скрипту директории.

13. Написать скрипт, который в качестве первого параметра принимает команду, которую нужно выполнить, а в качестве остальных параметров опции этой команды.

14. Написать скрипт, который в качестве первого параметра принимает команду, которую нужно выполнить, а в качестве остальных имена пользователей, от имени которых она должна быть выполнена.

15. Написать скрипт, который в качестве параметров принимает названия разделов жесткого диска и выдает разделы, на которых свободной памяти осталось менее 100 Мб в формате name|free memory

16. Необходимо запускать скрипт script13 каждые 13 секунд, но не более одного одновременно запущенного процесса.

17. Необходимо перенести файлы, имеющие расширения html и htm, из каталога data в каталог tabs. Каталоги исходный(data) и новый(tabs) находятся на одном уровне файловой системы. Например, /var/data и /var/tabs.

18. Необходимо перенести файлы, имеющие расширения, переданные в качестве параметра, из каталога data в каталог tabs. Каталоги исходный(data) и новый(tabs) находятся на одном уровне файловой системы. Например, /var/data и /var/tabs.

19. Напишите скрипт который запускается по крону и считает кол-во уникальных записей из какого-нибудь лога ( access.log auth.log ) и выводит на экран.

20. Напишите скрипт, который запускается по крону и считает кол-во уникальных записей из файла, переданного как параметр и выводит на экран.

21. Напишите скрипт, который запускается по крону и считает кол-во измененных в течении последних 3 дней файлов из каталога, переданного как параметр и выводит на экран.

22. Напишите скрипт, который проверяет, что в переданном каталоге файлы не изменялись, можно с выводом изменённых файлов или даже изменённых строк в файле.

23. Выполните рекурсивный обход домашнего каталога и сохраните информацию о всех файлах в файл. Сожмите файл. Скопируйте сжатый файл в каталог, переданный как параметр.

24. Напишите сценарий, который будет читать текст из заданного файла, и выводить, построчно, на stdout, добавляя при этом дополнительные пустые строки так, чтобы в результате получился вывод с двойным межстрочным интервалом.

25. Напишите сценарий, который будет читать текст из заданного файла, и выводить, построчно, на stdout, добавляя при этом дополнительные пустые строки так, чтобы в результате получился вывод с тройным межстрочным интервалом.

26. Напишите сценарий, который будет удалять пустые строки из заданного файла.

27. Напишите сценарий, который будет добавлять пустые строки из заданного файла так, чтобы в результате получился вывод с двойным межстрочным интервалом.

28. Для каждого файла, из заданного списка, сценарий должен определить тип архиватора, которым был создан тот или иной файл (с помощью утилиты file). Затем сценарий должен выполнить соответствующую команду разархивации (gunzip, bunzip2, unzip, uncompress или что-то иное). Если файл не является архивом, то сценарий должен оповестить пользователя об этом и ничего не делать с этим файлом.

29. Сценарий должен создать архив (\*.tar.gz) всех файлов в домашнем каталоге пользователя (/home/user-name), которые изменялись в течение последних 24 часов. Подсказка: воспользуйтесь утилитой find.

30. Сценарий должен создать архив (\*.tar.gz) всех файлов в указанном каталоге, которые изменялись в течение последних 48 часов. Подсказка: воспользуйтесь утилитой find.

31. Сценарий должен вывести (на stdout) все простые числа, в диапазоне от 60000 до 63000. Вывод должен быть отформатирован по столбцам (подсказка: воспользуйтесь командой printf).

32. Сценарий должен вывести (на stdout) все числа, делящиеся на 12, в диапазоне от первого параметра до последнего. Если параметры заданы некорректно, скрипт должен вывести сообщение.

33. Сценарий должен имитировать работу лототрона -- извлекать 5 случайных неповторяющихся чисел в диапазоне 1 - 50. Сценарий должен предусматривать как вывод на stdout, так и запись чисел в файл, кроме того, вместе с числами должны выводиться дата и время генерации данного набора.

34. Напишите сценарий, который находил бы корни "квадратного " уравнения, вида: Ax^2 + Bx + C = 0. Сценарий должен получать коэффициенты уравнения A, B и C, как аргументы командной строки, и находить корни, с точностью до четвертого знака после запятой.

35. Удалите все комментарии из сценария, имя которого задается с командной строки. При этом, строка "#!/bin/bash" не должна удаляться.