

# Операционные системы и виртуализация (Linux)

## Урок 1. Установка ОС Ubuntu в виртуальной машине. Работа в SSH-клиенте

### Условие

- Установить Ubuntu Desktop 22.04 в виртуальную машину.
- Установить гостевые дополнения ОС.
- Установить Midnight Commander.
- Установить SSH-соединение с виртуальной машиной из хостовой.

### Результат

Текстовый файл (\*.txt) с кодом команды, выводящей информацию о версии ОС (вместе с результатом работы этой команды).

Также привести текст остальных использованных команд.

## Урок 2. Работа с файлами и ссылками

### Задание:

1. Создать каталоги students и mentors в домашней директории, а в них — текстовые файлы students\_list.txt и mentors\_list.txt соответственно.
2. Открыть созданные в п.1 файлы в любом текстовом редакторе и заполнить их (в соответствии с названием) списком Ваших одноклассников и наставников на данном потоке.
3. Переместите файл mentors\_list.txt в папку students.
4. Удалите папку mentors.
5. Переименуйте папку students в students\_and\_mentors.
6. Удалите папку students\_and\_mentors вместе с содержимым.

7 \*. Создать файл file1 и наполнить его произвольным содержимым. Скопировать его в file2. Создать символическую ссылку file3 на file1. Создать жёсткую ссылку file4 на file1. Посмотреть, какие inode у файлов. Удалить file1. Что стало с остальными созданными файлами? Попробовать вывести их на экран.

8 \*. Дать созданным файлам другие, произвольные имена. Создать новую символическую ссылку. Переместить ссылки в другую директорию.

### Результат:

Текст команд, которые применялись при выполнении задания. Присылаем в формате текста выполненных команд и результатов их выполнения из терминала (простые файлы \*.txt, созданные в блокноте). Без скриншотов.

**Задания со звездочкой (\*) дополнительные, необязательные.**

## Урок 3. Права доступа и безопасность

### 1. Управление пользователями:

- а) создать нового пользователя;
- б) убедиться, что информация о нем появилась в соответствующих файлах в системе;
- в) удалить созданного пользователя;

### 2. Управление группами:

- a) создать группу;
- b) попрактиковаться в смене групп у пользователей;
- c) добавить пользователя в группу, не меняя основной;
- d) удалить пользователя из группы.

### 3. Работа с группами.

- a) Создать пользователя с правами суперпользователя. Проверить результат.
- b) \* Создать группу developer и нескольких пользователей, входящих в неё. Создать директорию для совместной работы. Сделать так, чтобы созданные одними пользователями файлы могли изменять другие пользователи этой группы.
- c) \* Создать в директории для совместной работы поддиректорию для обмена файлами, но чтобы удалять файлы могли только их создатели.

***В качестве решения необходимо прислать текстовый файл (\*.txt) с набором используемых команд и их результатами (можно скопировать текст из консоли).***

***Задания со звездочкой (\*) дополнительные, необязательные.***

## Урок 4. Подключение сторонних репозиторий, ручная установка пакетов

1. Подключить репозиторий с nginx любым удобным способом, установить nginx и потом удалить nginx, используя утилиту dpkg.
2. Установить пакет на свой выбор, используя snap.
3. \* Создать с помощью nano файл test.txt. Настроить автоматический бэкап этого файла раз в 10 минут в файл с названием test.txt.bak с использованием cron.

***Решение присылаем в виде текста использованных команд (можно без результатов вывода). Без скриншотов!***

***Задачи со звездочкой (\*) необязательные, дополнительные.***

## Урок 5. Настройка сети в Linux. Работа с IPtables

### iptables

1. Настроить iptables: разрешить подключения только на 22-й и 80-й порты.
2. Настроить проброс портов локально с порта 80 на порт 8080.
3. Запретить любой входящий трафик с IP-адреса 3.4.5.6.

### Сигналы процессам

4. Запустите mc. Используя ps, найдите PID процесса, завершите процесс, передав ему сигнал 9.

## Урок 6. Запуск стека для веб-приложения

### Задание:

1. Установить Nginx
2. \* настроить Nginx на работу с PHP-FPM.
3. Установить Apache.
4. \* Настроить обработку PHP. Добиться одновременной работы с Nginx.
5. Настроить схему обратного прокси для Nginx (динамика - на Apache).
6. Установить MySQL. Создать новую базу данных и таблицу в ней.

7. \* Установить пакет `phpmyadmin` и запустить его веб-интерфейс для управления MySQL.

**Результат:**

Текст команд, которые применялись при выполнении задания. При наличии: часть конфигурационных файлов, которые решают задачу. Присылаем в формате текстового документа: задание и команды для решения (без вывода и без скриншотов).

## Урок 7. Запуск веб-приложения из контейнеров

1. Переустановить операционную систему (по желанию, для дополнительной практики)
2. Установить Docker.
3. Запустить контейнер с Ubuntu.
4. \* Используя Dockerfile, собрать связку nginx + PHP-FPM в одном контейнере.

## Урок 8. Скрипты Bash

**Задание**

- Написать скрипт очистки директорий.

На вход принимает путь к директории.

Если директория существует, то удаляет в ней все файлы с расширениями `.bak`, `.tmp`, `.backup`.

Если директории нет, то выводит ошибку.

- Создать скрипт `ownersort.sh`, который в заданной папке копирует файлы в директории, названные по имени владельца каждого файла.

Учтите, что файл должен принадлежать соответствующему владельцу.

**Результат**

Код скриптов в текстовом виде (каждый скрипт в отдельном файле).

Кодировка файлов UTF-8.

*Данная промежуточная аттестация оценивается по системе «зачёт» / «не зачёт»*

– «Зачёт» ставится, если Слушатель успешно выполнил задание 1 или 2 задания

– «Незачёт» ставится, если Слушатель не выполнил задание.

**Критерии оценивания**

- 1 – Слушатель написал скрипт очистки директорий
- 2 – Слушатель создал скрипт `ownersort.sh`