

Задание №1

- Скачиваем или покупаем программу VMware Workstation
- Устанавливаем на Windows
- Пока идёт процесс, знакомимся с тем, что такое виртуальная и хост-машина

Задание №2

- Скачиваем дистрибутив Windows
- Создаём гостевую машину
- Запускаем виртуальную машину
- Устанавливаем на неё Windows

Задание №3

- Выключаем виртуальную машину
- Экспортируем её в .OVA образ
- На его основе создаём ещё одну виртуальную машину
- Запускаем обе

Задание №4

- Пингуем адрес 127.0.0.1
- Запускаем на двух виртуальных и хост-машине cmd
- Узнаём их адреса
- Настраиваем сетевые интерфейсы виртуальных машин и конфигуратора сетей VMware, чтобы хост-машина могла пинговать виртуальные машины, а каждая из них могла пинговать другую
- Проверяем, что всё работает

Задание №5

- [Загружаем Debian](#)
- Устанавливаем её на первой виртуальной машине
- На этапе разметки диска делаем автоматическую
- На этапе выбора сервисов выбираем только базовые утилиты и SSH-сервера, когда спросит, устанавливаем загрузчик Grub

Задание №6

- Когда появится чёрный экран, набираем логин root и пароль, указанный при установке (мы не увидим, как набираем символы)
- Переходим в директорию /var
- С помощью команды узнаём, в какой директории находимся

- Читаем мануал по команде ls (только основное положение и про флаги -l, -t, -r, -A)
- Отображаем список директорий и файлов
- Переходим в директорию /var/log (без упоминания “var” в команде)
- Переходим в директорию /var (снова без упоминания “var”)
- Создаём директорию /tmp/lesson/
- Переходим туда, задав абсолютный путь
- Создаём здесь файл с названием hello.txt и ставим права на файл “0777”
- Пингуем внешний адрес 8.8.8.8

Если последнее действие не получается, значит где-то накосячили, ищем ошибку сами или гуглим.

Задание №7

- [Загружаем](#), устанавливаем и запускаем программу Putty
- Смотрим адрес виртуальной машины
- В программе подключаемся по SSH на нужный IP и далее работаем с системой только через SSH

Задание №8

- Отображаем содержимое /etc/apt/sources.list, используя команду cat
- С её же помощью и символа “>” записываем в файл следующие данные:

Для Debian 8 Jessie:

```
# DEBIAN JESSIE #####
```

```
## Debian Jessie base:
```

```
deb http://ftp.debian.org/debian/ jessie main contrib non-free
```

```
#deb-src http://ftp.debian.org/debian/ jessie main contrib non-free
```

```
## Debian Jessie security updates:
```

```
deb http://security.debian.org/ jessie/updates main contrib non-free
```

```
#deb-src http://security.debian.org/ jessie/updates main contrib non-free
```

```
# Debian Jessie updates:
```

```
deb http://ftp.debian.org/debian/ jessie-updates main contrib non-free
```

```
#deb-src http://ftp.debian.org/debian/ jessie-updates main contrib non-free
```

- Обновляем кэш менеджера пакетов apt-get
- С помощью этой команды устанавливаем пакет vim

Задание №9

- [Читаем про vi или vim](#)
- Открываем на редактирование /tmp/lesson/hello.txt с помощью vi

- Печатаем любой текст на 40-50 строк (нужно, чтобы в каждой строке были буквы “abc” в такой же последовательности)
- Сохраняем и выходим
- Отображаем содержимое с помощью cat, затем less
- Отображаем только последние 10 строк, затем первые 7 строк
- Считаем количество строк в файле

Задание №10

- Узнаём, что такое конвейер и “pipe”
- Отображаем содержимое /tmp/lesson/hello.txt и перенаправляем результат по конвейеру в подсчёт строк
- Отображаем содержимое /tmp/lesson/hello.txt и перенаправляем результат по конвейеру в grep
- Отправляем дальше по конвейеру в подсчёт строк
- Отображаем содержимое /tmp/lesson/hello.txt и вывод перенаправляем в grep

Задание №11

- Читаем мануал по shutdown
- С её помощью выключаем виртуальную машину

Задание №12

- Склонируем диск первой виртуальной машины на вторую
- Запускаем вторую
- Входим с логином root
- Меняем название, чтобы знать, что это вторая виртуальная машина
- С помощью shutdown перезагружаем сервер

Задание №13

- Запускаем первый сервер (второй уже запущен, должно быть)
- Смотрим через консоль IP (должны отличаться)
- Читаем про IPv4 и маску подсети
- Знакомимся с DHCP
- Вписываем IP в конфигурационный файл сети на обеих виртуальных машинах (IP+маска+шлюз)
- Перезагружаем обе машины
- Проверяем, что с хост-машины пингуются виртуальные
- Сохраняем настройки сессий в putty для обеих виртуальных машин
- Выключаем вторую виртуальную машину (второй сервер)

Фух, теперь про повседневные задачи сисадмина.

Задание №14

- Запускаем командную строку
- Создаём переменную и присваиваем любое значение
- Отображаем её
- Проверяем, что переменная сохраняет своё значение после рестарта с помощью `echo $a`
- Отображаем переменную `$PATH`
- Разбираемся с тем как она работает
- Делаем так, чтобы добавился ещё один путь — `/tmp`

Задание №15

- Разбираемся с BASH и sha-bang
- Пишем пустой скрипт с расширением `.sh` и запускаем его
- Делаем так, чтобы скрипт можно было запускать без указания пути

Задание №16

- Делаем так, чтобы скрипт инициализировал `DATE` и задавал значение времени, затем выводил на экран
- Делаем так, чтобы вместо вывода на экран создавался файл в директории подобный `/tmp/2017-08-17.txt`
- Делаем так, чтобы в файл записывалось время создания

Задание №17

- Используя CRON заставляем скрипт исполняться ежеминутно
- Через 10 минут проверяем, чтобы в директории было столько же файлов, сколько прошло минут
- Обнаруживаем ошибку
- Используя `find` делаем так, чтобы директория проверялась раз в 10 минут и удаляла файлы, созданные больше, чем 17 минут назад

Задание №18

- Создаём пустой файл и вписываем что-нибудь
- Копируем с сохранением всех прав в другой файл с любым названием и содержимым
- Сравниваем контрольные суммы
- Меняем содержимое любого из файлов и смотрим разницу в контрольных суммах

Задание №19

- Создаём пользователя и задаём ему пароль
- Разбираемся с `/etc/passwd`, `/etc/shadow`, `/etc/group`

- Делаем так, чтобы, если нет файла /tmp/passwd.backup, /etc/passwd копировался в /tmp/passwd.backup, а если /tmp/passwd.backup есть, то сравнивался с /etc/passwd
- Если различаются, то пусть копируется, если нет, пусть завершает работу скрипта

Задание №20

- Используя apt-get устанавливаем Apache2 на виртуальную машину
- С браузера хост-машины по IP-адресу виртуальной машины видим приветствие
- Меняем порт на :8080
- Снова выполняем то, что делали два шага назад, но с указанием порта
- Меняем порт обратно и проверяем как работает заглушка
- В hosts хост-машины создаём три домена: a1.com, b2.com, c3.com и связываем с IP виртуальной машины с Apache
- Для каждого домена проверяем всё ли правильно, с помощью ping
- Заходим на все три домена, написав их вместо IP виртуальной машины

Задание №21

- Создаём директории /var/www/a1.com, /var/www/b2.com, /var/www/c3.com
- В каждой из них создаём пустой index.html
- В каждом что-нибудь пишем
- Делаем так, чтобы открывался личный сайт из директории, а не общая заглушка

Задание №22

- Меняем index.html на index.php и добавляем следующую запись:

`<?php Анни(); ?>`

- Видим ошибку
- Через apt-get активируем mod_php
- Создаём инструкцию для Apache, чтобы все php файлы обрабатывались этим модулем
- Настраиваем такие же модули с инструкциями для Perl и Python
- b2.com и c3.com переводим на index.pl и index.py
- Создаём приветствия

Задание №23

- Читаем про HTML Basic вот [здесь](#)
- Про PHP Basic [здесь](#)

Задание №24

- Устанавливаем mysql-server
- С помощью mysql подключаемся через командную строку к серверу

- Создаём базу данных mydatabase и пользователя myuser
- Читаем раздел SQL Basic [здесь](#)
- Устанавливаем Joomla на a1.com
- Wordpress на b2.com

Задание №25

- Читаем про реверс-прокси
- Устанавливаем Nginx
- Делаем так, чтобы 8080 порт Apache получал запросы 80 порта Nginx
- Проверяем a1.com, Nginx access.log должен пополняться при обращении к сайту
- Делаем, чтобы Nginx всё, кроме jpg и png файлов передавал Apache

Задание №26

- Включаем второй сервер
- Собираем MySQL Cluster на двух серверах
- Создаём базу данных с простыми таблицами, заполненными рандомными данными для тестирования
- Вносим изменения на один из серверов и проверяем на другом
- Выключаем второй сервер
- Рушим файлы базы вручную (всё для проверки)
- Делаем так, чтобы база восстановилась и появилась в кластере (думайте, как)

Задание №27

- Выключаем оба сервера
- Добавляем каждому серверу по одному диску на 700 Мб
- Включаем обратно
- С помощью fdisk проверяем, как обозначены диски
- С помощью DRBD соединяем диски серверов, чтобы изменения отображались на обоих (оба должны быть смонтированы в /mnt/drbd-disk)

Задание №28

- Выключаем первый сервер
- Добавляем ещё диски по 200 Мб
- Включаем сервер
- С помощью fdisk отображаем структуру дисков, должно быть четыре диска, из которых два не размечены
- С помощью mraid соединяем в RAID диски, с которыми не работали
- Монтируем /dev/md0 в /mnt/raid-disk

Задание №29

- Выключаем второй сервер
- Добавляем диск на 2Гб
- Включаем, настраиваем LVM и монтируем в /mnt/lvm-disk1
- Уменьшаем размер диска LVM вдвое и создаём такой же
- Монтируем в /mnt/lvm-disk1 и /mnt/lvm-disk2

Задание №30

- Ликвидируем exim4-config (если был)
- Ставим postfix
- Настраиваем пересылку почты с сервера к себе в почтовый ящик с помощью [swaks](#)

Задание №31

- Настраиваем на первом сервере в Nagios мониторинг обоих серверов
- Подключаем уведомления на свою почту (на всякий случай)
- Имитируем поломку MySQL и ждём сообщения о неисправности на почту

Задание №32

- На втором сервере поднимаем SVN-репозиторий scripts
- Делаем его доступным на любом сервере через svnserve
- На первом сервере клонируем в директорию /opt/scripts/
- На хост-машине в D:\scripts

Задание №33

Так, сейчас работаем только в vi.

- На первом сервере пишем скрипт /opt/scripts/backup.sh, чтобы он: на наших сайтах запаковывал директории с файлами в tar и сжимал в gz, снимал MySQL-дампы базы сайта в формате .sql, складывал всё это в /var/backups/www/имя_сайта, проверял содержимое файлов и так со всеми, а в конце, нужно, чтобы он отправлял на почту отчёт
- Фиксируем скрипт в репозитории

Задание №34

- Обновляем SVN-репозиторий
- Копируем файл из репозитория второго сервера в D:\scripts
- Устанавливаем [Вот это](#)
- Через GVim добавляем случайную строку
- Коммитим скрипт в репозиторий

- В /opt/scripts обновляемся
- Хотим увидеть изменения в скрипте

Задание №35

- Переписываем бэкап скрипт так, чтобы не нужно было перечислять существующие сайты, он должен обнаруживать их в конфигурации Apache
- Коммитим
- Переписываем бэкап скрипт так, чтобы все сообщения он писал и в консоль, и в syslog
- Коммитим
- Вносим синтаксическую ошибку в скрипт
- Коммитим с ошибкой
- Откатываем до последней рабочей версии

Задание №36

- Переписываем бэкапный скрипт так, чтобы, он определял CMS сайта и собирал данные доступа к базе
- Коммитим

Задание №37

- Настраиваем rsyslog первого сервера, чтобы логи бэкапного скрипта писались в ar/log/backup.log
- Настраиваем сбор логов на первом сервере
- Активируем LogAnalyzer здесь же

Задание №38

- Переписываем бэкап скрипт так, чтобы по завершению запускалась синхронизация директорий /var/backups/www между серверами
- Коммитим
- Повторяем, используя lftp mirror

Задание №39

- Читаем [здесь](#) и [здесь](#)

Задание №40

- Активируем puppet на втором сервере
- Отдаём ему под управление /etc/passwd

Задание №41

- Читаем про `init`
- Загружаемся в разные `init`-уровни
- Перезагружаем сервер с помощью `init`
- Выключаем сервер с помощью `init`
- Включаем обратно

Задание №42

- Попробуем восстановить пароль от `root`, загрузив ядро с параметром `init=/bin/bash`

Задание №43

- Читаем про различие аутентификации и авторизации
- Разбираемся с РАМ [здесь](#) и [здесь](#)
- Разбираемся, зачем нужен файл `/etc/nsswitch.conf`
- Делаем авторизацию SSH в MySQL (files оставляем про запас)

Задание №44

- На `bash` пишем скрипт в `/opt/scripts/my_daemon.sh`, чтобы он: отцеплялся от `STDIN, STDOUT, STDERR`, менял рабочую директорию на `/`, закрывал все дескрипторы, выводил количество минут до перезагрузки

Задание №45

- Создаём `init`-скрипт в `/etc/init.d/my_daemon` с LSB-тэгами
- Попробуем запустить и остановить скрипт `/opt/scripts/my_daemon.sh` используя `init`-скрипт `/etc/init.d/my_daemon`
- Запускаем, останавливаем и смотрим статус скрипта, используя `service`
- Ставим `daemon` на автозагрузку

Задание №46

- Меняем скрипт с `SysVinit` на `systemd`
- Запускаем, останавливаем и смотрим статус скрипта, используя `systemctl`

Задание №47

- Проходим онлайн-курсы по сетям
- Знакомимся с OSI, TCP/IP, NAT и прочими штуками по теме

Задание №48

- Ставим на обычный ПК Cygwin (выбираем python, vim, git, openssh и все пакеты, которыми пользовались на Linux)
- Добавляем путь к бинарным файлам Cygwin в %PATH%
- Запускаем cmd, пробуем работать в нём на Windows
- Ставим что-нибудь более удобное

Задание №49

- Удаляем все виртуальные машины
- Удаляем на Windows VMware-продукты
- [Ставим VirtualBox](#) , [Vagrant](#)
- Создаём директорию C:\VM\server1 и переходим к ней
- Инициализируем Debian с помощью Vagrant, запускаем его
- Открываем VirtualBox и проверяем, что создалась новая виртуальная машина
- Останавливаем и удаляем виртуальную машину
- Регулируем имя, 3 CPU, RAM 1Gb, проброс порта 22
- Запускаем её и подключаемся по SSH через putty

Задание №50

- На Cygwin ставим Ansible
- Включаем Vagrantfile в самом конце provisioning Ansible
- Дополняем Ansible: apt-get: git, curl, ruenv, python3.6 при помощи ruenv, делаем python3.6 дефолтным глобально
- Устанавливаем Logbook, paramiko, lxml
- Создаём с нуля виртуальную машину

Задание №51

- Проверяем знания [по тесту](#)