Лабораторная работа №1
«Основы сетевого администрирования»

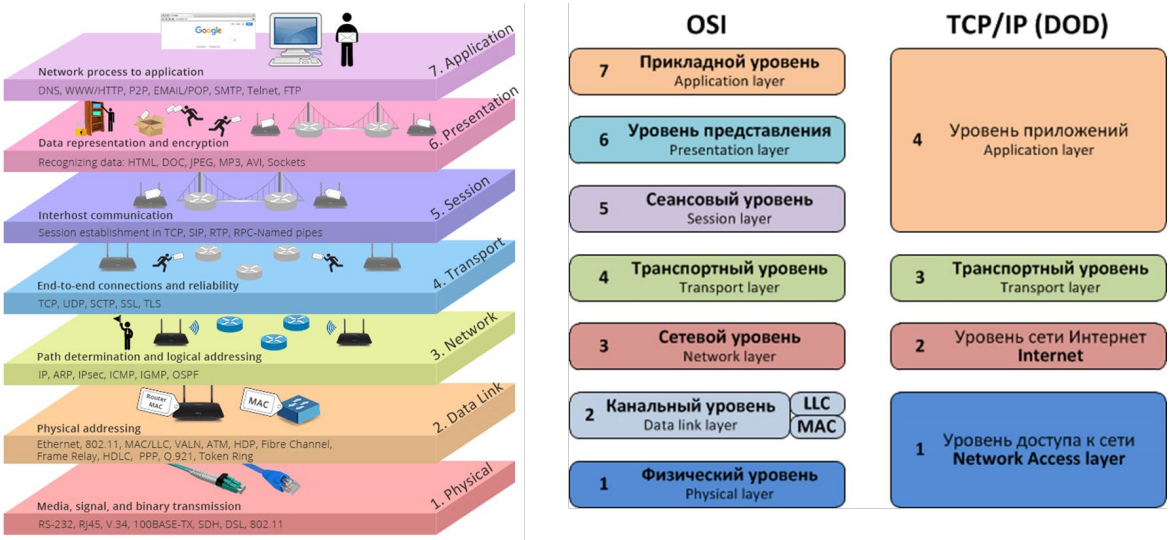
1. Теоретическая часть

1.1. Введение в сетевое администрирование

Локальная вычислительная сеть (ЛВС, локальная сеть; англ. Local Area Network, LAN) — компьютерная сеть, покрывающая обычно относительно небольшую территорию или небольшую группу зданий (дом, офис, фирму, институт).

В сети должен происходить обмен данными между вычислительными устройствами — компьютерами, серверами, маршрутизаторами, принтерами и другим оборудованием. Для передачи информации могут быть использованы различные среды передачи данных (витая пара, оптоволокно, радиоволны).

Всю работу сети можно проанализировать как по модели OSI (англ. The **O**pen **S**ystems **I**nterconnection model), так и по модели TCP/IP.



В обязанности сетевого администратора входит построение ЛВС, её настройка, поддержка и улучшение. И это лишь малая часть, чем может

заниматься специалист в данной области. Обязанности могут различаться в зависимости от масштабов сетей, специфики работы какой-либо компании и т.д.

1.2. Роль Linux в управлении ЛВС

Несмотря на то, что в локальную сеть могут входить пользовательские компьютеры, где чаще всего используется операционная система Windows для выполнения повседневных задач, администрирование сети происходит с помощью устройств (серверов) под управлением операционной системы на базе ядра Linux.

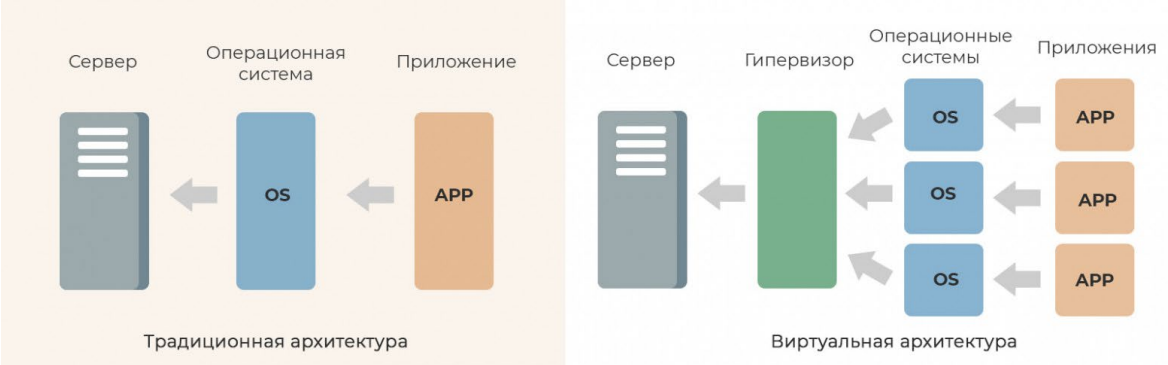
Используются как широко распространённые дистрибутивы GNU/Linux с графическим интерфейсом, так и дистрибутивы, предоставляющие для управления только командную строку. Это связано с тем, что использовать GNU/Linux на серверах выгодно, потому что эта операционная система бесплатна и можно сразу же развернуть нужный образ GNU/Linux на сервере. Это обеспечивает надёжность, гибкость и масштабируемость.

1.3. Виртуальная машина

Каждому администратору сети приходится иметь дело с настройкой виртуальных машин. Они необходимы для разработки и тестирования приложений, а также хранения данных. Преимуществами виртуальных машин можно назвать гибкость в выборе операционной системы и возможность дублирования рабочего пространства.

Виртуальная машина (VM, от англ. virtual machine) — программная или аппаратная система, эмулирующая аппаратное обеспечение компьютера и исполняющая программы для guest-платформы (guest — гостевая платформа) на host-платформе (host — хост-платформа, платформа-хозяин).

На одном хост-компьютере может быть множество гостевых ВМ. Хотя виртуальная машина создается с помощью ПО, она использует физические ресурсы хост-машины, такие как ЦП, ОЗУ и место в хранилище на жёстком диске. На своём хост-компьютере можно настроить столько виртуальных машин, сколько нужно, но придётся разделить физические аппаратные ресурсы между ними всеми. Однако большинство ВМ будут работать медленнее, чем физический компьютер, просто из-за дополнительных уровней абстракции, которые они должны пройти для выполнения функции.



Видов виртуализации существует несколько:

Программная виртуализация — вид виртуализации, который использует различные библиотеки ОС, транслируя вызовы виртуальной машины в вызовы ОС.

Аппаратная виртуализация — вид виртуализации, который предусматривает специализированную инструкцию аппаратной части, а конкретно инструкций процессора. Позволяет исполнять запросы в обход гостевой ОС прямо на аппаратном обеспечении.

Программная виртуализация обеспечивается гипервизором.

Гипервизор (англ. Hypervisor) или монитор виртуальных машин — программа или аппаратная схема, обеспечивающая одновременное,

параллельное выполнение нескольких операционных систем на одном и том же хост-компьютере.

Гипервизор также обеспечивает изоляцию операционных систем друг от друга, защиту и безопасность, разделение ресурсов между различными запущенными ОС и управление ресурсами.

1.4. Виртуализация QEMU/KVM в Astra Linux

Для исполнения прямых аппаратных запросов в ОС должна иметься библиотека, которая направляла бы эти запросы аппаратной части напрямую. На платформах базы Linux долгое время никакой встроенной системы виртуализации (встроенного гипервизора), просто не существовало. Каждый производитель ПО для виртуализации, который поддерживало технологию аппаратной виртуализации, вынуждены были создавать собственные модули для ядра Linux.

Однако со временем был разработан свободный базовый гипервизор **KVM** или **Kernel-based Virtual Machine**, который представляет из себя загружаемый модуль ядра Linux и предназначен для обеспечения виртуализации на платформе **x86** с поддержкой одной из технологий аппаратной виртуализации — **Intel VT** либо **AMD SVM**.

Сам по себе KVM не выполняет эмуляции. Вместо этого сторонняя программа, работающая в пространстве пользователя, использует интерфейс **/dev/kvm** для настройки адресного пространства гостя виртуальной машины. Например, свободная программа **QEMU** (Quick Emulator) использует KVM для эмуляции аппаратного обеспечения различных платформ.

В Astra Linux по умолчанию используется менеджер виртуальных машин **virt-manager**, представляющий собой пользовательский интерфейс для управления виртуальными машинами на рабочем столе с помощью API **libvirt**. В нём представлена сводная информация о запущенных доменах, их текущей

производительности и статистике использования ресурсов. Мастера позволяют создавать новые домены, а также настраивать распределение ресурсов домена и виртуальное оборудование.

1.5. Сетевые настройки в GNU/Linux

После установки какого-либо дистрибутива GNU/Linux для подключения к локальной или глобальной сети необходимо произвести сетевые настройки.

Для этого используются следующие системные файлы:

Путь к файлу	Назначение
/etc/hostname	Настройка имени компьютера
/etc/hosts	Настройка разрешения доменных имен
/etc/resolv.conf	Настройка адресов серверов имен, к которым имеет доступ данная система
/etc/network/interfaces	Настройка сетевых интерфейсов
/etc/apt/sources.list	Настройка списка репозиториев

Рассмотрим эти файлы подробнее:

1. Файл **/etc/hostname** предназначен для настройки имени компьютера. Текущее имя компьютера можно узнать командой **hostname** или по переменной окружения **HOSTNAME**. После редактирования файла необходимо выполнить перезагрузку системы.

2. Файл **/etc/hosts** представляет собой список доменных имён. В качестве аналогии можно привести телефонный справочник, только вместо номера телефона указывается IP-адрес, а вместо имени человека – доменное имя. При использовании команды **ping** указывается либо IP-адрес устройства, либо его доменное имя. При указании имени система обращается к файлу **/etc/hosts**, определяет по имени соответствующий IP-адрес, затем происходит обращение к целевому устройству по протоколу ICMP.

3. Файл **/etc/resolv.conf** хранит доменное имя и IP-адрес DNS-сервера. В случае, если при указании имени система не находит соответствующий IP-адрес в локальном файле **/etc/hosts**, она делает запрос в DNS-сервер. Если и там нет информации об указанном доменном имени, то выдаётся ошибка разрешения имён.

4. В файле **/etc/network/interfaces** хранятся параметры сетевых интерфейсов (например, Ethernet или WiFi). По умолчанию содержит следующие строки:

1) **source /etc/network/interfaces.d/***. Команда **source** вставляет в текущий файл **interfaces** содержимое папки **interfaces.d** (обычно изначально пустого). Удобен для соблюдения принципа модульности.

2) **auto lo**. Директива **auto** означает, что интерфейс, указанный справа, должен быть автоматически запущен во время загрузки системы.

lo (loopback, обратная петля) - виртуальный сетевой интерфейс, не связанный с каким-либо оборудованием, но при этом полностью интегрированный во внутреннюю сетевую инфраструктуру системы. Любой трафик, который посылается программой на интерфейс loopback, тут же получается тем же интерфейсом.

Он может быть использован сетевым клиентским программным обеспечением, чтобы общаться с серверным приложением, расположенным на том же компьютере. То есть если на компьютере, на котором запущен веб-сервер, указать в веб-браузере URL http://127.0.0.1/ или http://localhost/, то он попадает на веб-сайт этого компьютера.

Этот механизм работает без какого-либо активного подключения, поэтому он полезен для тестирования служб, не подвергая их безопасность риску, как при удаленном сетевом доступе. Подобным образом, пингование адреса loopback — это основной тест функционирования IP стека в операционной системе.

3) **iface lo inet loopback**. Директива **iface** описывает сетевой интерфейс **lo**, **inet** означает семейство интернет-протоколов (IPv4).

4) **auto eth0**. Директива **auto** предписывает автоматически включить сетевой интерфейс eth0 во время загрузки. В Astra Linux используется по умолчанию традиционная схема именования сетевых Ethernet интерфейсов: eth0, eth1 и т.д.

5) **iface eth0 inet dhcp**. Описание интерфейса **eth0** с присвоением IP-адреса по протоколу DHCP, то есть автоматически при обращении к DHCP-серверу.

Пользователь может самостоятельно назначать интерфейсу статический IP-адрес. Для этого в конфигурационном файле **/etc/network/interfaces** необходимый интерфейс описывается в следующем формате:

```
auto IFACE_NAME
iface IFACE_NAME inet static
address A.B.C.D
netmask A.B.C.D
gateway A.B.C.D
```

При этом **static** означает, что интерфейсу присваивается адрес статически, **address A.B.C.D** – это назначаемый IP-адрес интерфейса, **netmask A.B.C.D** – маска подсети, **gateway A.B.C.D** – шлюз по умолчанию.

После сохранения изменений необходимо перезагрузить интерфейс для установления введённых параметров. Для этого используются команды **ifdown IFACE_NAME** и **ifup IFACE_NAME**.

Установившиеся настройки можно посмотреть с помощью команды **ifconfig** (с применением **sudo**) или с помощью стандартной команды **ip a**.

Ниже приведен пример настройки сетевого интерфейса:

```
auto eth0
iface eth0 inet static
address 192.168.1.2
netmask 255.255.255.0
gateway 192.168.1.1
```

С помощью указанной конфигурации настраивается статический IP адрес со значением 192.168.1.2. Маска подсети – 255.255.255.0, адрес шлюза – 192.168.1.1.

1.6. Понятие службы в Linux

В операционной системе GNU/Linux, кроме обычных программ, существуют постоянно работающие утилиты, работающие в фоне и предоставляющие определенные функции пользователям или системе. Такие программы называются **службами**.

Например, для подключения к компьютеру по протоколу **ssh** (Secure Shell) необходимо постоянно запущенное приложение сервера, принимающее и обрабатывающее запросы. В этом качестве выступает служба **sshd**. Для каждого нового соединения создаётся её экземпляр, который выполняет обмен ключами, шифрование, аутентификацию, выполнение команд и обмен данными.

Для работы со службами используется утилита **systemctl**, имеющий следующий синтаксис:

```
$ systemctl опции команда служба...
```

Опции служат для настройки поведения запущенной программы. Основные команды перечислим ниже:

- **start** — запустить службу linux;
- **stop** — остановить службу linux;
- **reload** — попросить службу перечитать свою конфигурацию из файловой системы;
- **restart** — перезапустить службу;
- **enable** — добавить службу в автозагрузку;
- **disable** — удалить службу из автозагрузки;
- **status** — проверить текущее состояние службы.

Например, с помощью команды:

```
systemctl start sshd
```

возможно запустить службу для работы с ssh-соединениями.

1.7. FTP-сервер

FTP (File Transport Protocol) — протокол передачи файлов по сети прикладного уровня модели OSI.

Протокол появился в 1971 году и, несмотря на возраст, активно используется повсеместно.

FTP-сервер представляет собой хранилище файлов, который по запросу из сети осуществляет приём или передачу файлов.

Репозиторий (от англ. repository — хранилище) — место, где хранятся и поддерживаются какие-либо данные.

Именно на FTP-репозитории была впервые выложена первая версия ядра Linux. Расположение репозитория на одном сервере позволяет пользователям локальной или глобальной сети обращаться к ней, тем самым не вынуждая хранить данные у себя на локальной машине и легко делиться ими с другими.

В качестве FTP-сервера предлагается программа **vsftpd** (**V**ery **S**ecure **FTP** **D**aemon), распространяемая под лицензией GPL. Её установить можно с официальных репозиториев Astra Linux менеджером пакетов **apt**. Делается это следующей командой:

```
sudo apt install vsftpd
```

Для настройки необходимо с правами суперпользователя отредактировать конфигурационный файл **/etc/vsftpd.conf**. Для этого рекомендуется сначала сделать backup файла:

```
sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.orig
```

Затем можно удалить файл **/etc/vsftpd.conf** и начать редактировать командой

```
sudo nano /etc/vsftpd.conf
```

В пустой файл надо внести следующие параметры:

```
listen=yes
listen_ipv6=no
anonymous_enable=no
local_enable=yes
write_enable=yes
```

То есть сервер будет слушать порт по протоколу IPv4, а не IPv6, доступ к FTP-серверу будет запрещён неавторизованным пользователям, при подключении разрешается запись и чтение файлов.

После каждого изменения конфигурационного файла необходимо перезапустить службу **vsftpd**.

Проверить работу FTP-сервера можно с помощью утилиты ***ftp***, которая отдельно устанавливается менеджером пакетов:

```
sudo apt install ftp
```

Чтобы подключиться к серверу, надо ввести

```
sudo ftp A.B.C.D
```

где **A.B.C.D —** IP-адрес FTP-сервера.

При подключении будет предложено ввести имя пользователя, который зарегистрирован на сервере и пароль к нему. В случае успеха будет доступна домашняя директория, которую можно посмотреть командой **ls**, а для выгрузки файлов используется команда **get**. Для вывода списка доступных команд используется **help**.

1.8. SMB-сервер

SMB (сокр. от англ. Server Message Block) — сетевой протокол прикладного уровня для удалённого доступа к файлам, принтерам и другим сетевым ресурсам.

Samba — пакет программ, которые позволяют обращаться к сетевым дискам и принтерам на различных операционных системах по протоколу SMB.

Для установки Samba используется команда

```
sudo apt install samba
```

В качестве примера создадим общую папку. Для этого командой

```
mkdir /srv/share
```

создадим папку. Изменим права доступа к этой папке

```
sudo chown nobody:nogroup /srv/share
sudo chmod 775 /srv/share
```

Далее в конец файла **/etc/samba/smb.conf** необходимо добавить следующие строки:

```
map to guest = Bad User
[share]
  comment = <Произвольный комментарий>
  guest ok = yes
  force user = nobody
  force group = nogroup
  path = /srv/share
  read only = no
```

После сохранения файла проверяем установку параметров командой **testparm.** Осуществляем перезагрузку службы командой

```
sudo systemctl restart smbd
```

Работа с общими папками происходит с использованием протокола CIFS (**C**ommon **I**nternet **F**ile **S**ystem). Поэтому стоит убедиться, что в системе установлен пакет **cifs-utils**. Её использование определяется следующим синтаксисом:

```
$ mount -t cifs <папка на сервере> <во что монтируем> <-o опции>
```

То есть, чтобы клиент мог подключиться к общей папке, ему необходимо выполнить следующую команду:

```
sudo mount -t cifs //A.B.C.D/share /mnt -o users,sec=none
```

Программа **mount** используется для монтирования USB-устройств, дисков, сетевых папок и т.д. Ключ **-t** указывает тип файловой системы **cifs**. После символов // пишется IP-адрес SMB-сервера, название общей папки. Всё это монтируется в папку **/mnt**, а ключом **–o** указываются опции **users** (папка для всех пользователей) и **sec** (отвечает за безопасность). После перезагрузки системы необходимо заново произвести монтирование.

После монтирования в папке **/mnt** должны лежать файлы, созданные в общей папке на сервере **/srv/share**.

1.9. NTP-сервер

NTP (англ. Network Time Protocol — протокол сетевого времени) — сетевой протокол для синхронизации внутренних часов компьютера с использованием сетей.

Для синхронизации времени в системах GNU/Linux используется демон **ntpd** (**N**etwork **T**ime **P**rotocol **d**aemon). Для его установки применяется команда

```
sudo apt install ntp
```

Принцип работы NTP основан на использовании иерархии серверов, которая обеспечивает распределение точного времени по всей сети. Уровни этой иерархии называются стратами (англ. *strata*). Высший уровень 0

представляет источники времени такие как атомные часы. Следующий уровень 1 — это серверы, которые получают точное время от нулевого уровня и сами могут служить источником времени для серверов следующего уровня 2. Этот процесс продолжается до достижения конечных клиентских устройств.

Иерархическая структура протокола NTP построена с учетом отказоустойчивости и избыточности. В случае потери соединения с вышестоящими серверами NTP резервные серверы берут процесс синхронизации на себя. За счёт избыточности обеспечивается постоянная доступность NTP-серверов. Синхронизируясь с несколькими серверами, NTP использует данные всех источников, чтобы рассчитать наиболее точное время.

В качестве примера рассмотрим локальную сеть, состоящую из NTP-сервера и его клиента и изолированную от глобальной сети Интернет. По умолчанию сервер настроен на получение точного времени от прописанных в конфигурационном файле вышестоящих серверов. Однако при отсутствии связи с сетью Интернет сервер будет высылать по запросу своё системное время.

Конфигурационный файл имеет путь **/etc/ntp.conf**. Он содержит начальные настройки и примеры конфигурирования для различных задач. Рекомендуется сделать резервное копирование файла с помощью команды

```
sudo cp /etc/ntp.conf /etc/ntp.conf.orig
```

Для понимания процесса настройки сервера, удалим конфигурационный файл командой

```
sudo rm /etc/ntp.conf
```

и в редакторе создадим новый файл командой

```
sudo nano /etc/ntp.conf
```

В пустой файл впишем следующую строку

```
server 127.127.1.0 prefer
```

Директива **server** указывает, к какому серверу необходимо подключиться для получения точного времени. IP-адрес 127.127.1.0 – это адрес, по которому сервер получает своё системное время, которое и будет отправлять по запросу клиенту. Опция **prefer** в данном случае указывается обязательно, так как серверу не нравится факт использования своего времени из-за высокой вероятности рассинхронизации с точным мировым временем.

После внесения изменений в конфигурационный файл необходимо перезагрузить службу командой

```
sudo systemctl restart ntp
```

В качестве теста запустим эмулятор терминала на клиенте и введём команду

```
sudo ntpdate A.B.C.D
```

где A.B.C.D – IP-адрес локального NTP-сервера. В случае успеха будет показано, насколько установленное на клиенте время отличается от времени сервера.

1.10. Подключение к системе по SSH

SSH (англ. Secure Shell — «безопасная оболочка») — сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой

SSH работает в режиме точка-многоточка. В этом режиме есть некоторый узел, к которому могут подключаться другие узлы — сервер. Все клиенты подключаются к этому серверу.

В Linux системах обычно используется OpenSSH, где сервер обозначается sshd, что означает SSH демон (Daemon), а клиент SSH — ssh.

Для аутентификации по ключам используется алгоритм асимметричного шифрования RSA. Пара ключей представлена публичным и приватным ключами. Ключи на клиенте хранятся в каталоге **~/.ssh** в файлах:

- **id_rsa** – приватный ключ пользователя;
- **id_rsa.pub** – публичный ключ пользователя;
- **known_hosts** – публичные ключи ssh-серверов.

На сервере командой **ssh-keygen** производится генерация ключей.

Командой

```
ssh-copy-id adminstd@A.B.C.D
```

передаётся публичный ключ на клиент по его IP-адресу A.B.C.D.

После этого клиент командой

```
ssh adminstd@A.B.C.D
```

может удалённо подключиться по протоколу SSH к серверу с IP-адресом A.B.C.D.

Команда scp (*Secure CoPy*) — это утилита, которая работает по протоколу SSH.

Она позволяет копировать файлы с клиента на сервер напрямую без использования протоколов FTP или SMB. Имеет следующий синтаксис:

```
$ scp опции пользователь1@хост1:файл пользователь2@хост2:файл
```

2. Практическая часть

После захода в свою учётную запись в Astra Linux запустите менеджер виртуальных машин.

Создайте сервер и клиент путём клонирования виртуальной машины ru01std00VNet. Присвойте серверу имя **Server**, а клиенту имя **Client** (или другим аналогичным образом, чтобы было ясно, что чем является).

Запустите машины. После этапа загрузки операционной системы появится окно для ввода пароля пользователя **adminstd**. Введите пароль по умолчанию **qwerty21**. Используя эмулятор терминала Fly, выполните следующие задания.

2.1. Задание 1

2.1.1. Проанализируйте файл **/etc/network/interfaces**. Что означает каждая строка?

2.1.2. Выясните текущие сетевые настройки каждой машины. Проверьте соединение между клиентом и сервером.

2.1.3. Используя параметры, приведённые в таблице ниже, настройте сеть, дополнив необходимые конфигурационные файлы.

Machine name	Server	Client
Host name	Ваши инициалы_server	Ваши инициалы_client
IP address	192.168.122.(№ в группе + 1)	192.168.122.(№ в группе + 2)
Default gateway	192.168.122.1	

2.1.4. Попробуйте отправить **ping** с сервера на клиент используя доменное имя. Получилось ли это сделать? Если нет, то исправьте это.

Задание 2.

2.2.1. На сервере создайте локальный FTP-репозиторий и загрузите на него файл, содержащий в названии ваши ФИО.

2.2.2. Выгрузите файл из созданного репозитория на машину Client.

2.2.3. Создайте на сервере общую папку **smb** и примонтируйте её на машине клиента в директорию с вашим именем.

2.2.4. На Client, используя графический интерфейс, поменяйте дату и время на 01.01.1970 и 18:12. Синхронизируйте время Server и Client по сети, установив NTP-сервер на машину Server и добавив её в автозагрузку.

Задание 3.

2.3.1. На сервере запустите службу **ssh** и добавьте её в автозагрузку.

2.3.2. На клиенте настройте аутентификацию по ключам с сервером.

2.3.3. Подключитесь к серверу с машины клиента и создайте в директории **/home/study** файл с содержимым «Hello world!».

2.3.4. Скопируйте с сервера на клиент командой **scp** файл, созданный в предыдущем пункте.

Контрольные вопросы

1. Что такое ЛВС? Из каких устройств может состоять?
2. Из каких уровней состоит модель OSI? Для чего её создали?
3. Какие операционные системы могут использоваться на серверах?
4. Что такое виртуальная машина? Зачем она нужна?
5. Какие существуют способы передачи файлов по сети?