

Лабораторная работа №12

Презентация

Мосолов А.Д.

22 ноября 2025

Российский университет дружбы народов, Москва, Россия

Докладчик

- Мосолов Александр Денисович
- Студент, НПИбд02-23
- Российский университет дружбы народов
- 1132236128@pfur.ru

Цель

Получение навыков по управлению системным временем и настройке синхронизации времени.

Задание

1. Изучить команды по настройке параметров времени (`timedatectl`, `date`, `hwclock`).
2. Настроить сервер в качестве сервера синхронизации времени для локальной сети.
3. Написать скрипты для Vagrant, фиксирующие действия по установке и настройке NTP-сервера и клиента.

Проверка времени на сервере

Сначала я проверяю текущие настройки времени и часового пояса на виртуальной машине server. Использую команду `timedatectl`. Видно, что время установлено в UTC, синхронизация NTP активна, а аппаратные часы также идут по UTC.

```
[admosolov@server.admosolov.net ~]$ timedatectl
          Local time: Thu 2025-11-20 18:10:23 UTC
      Universal time: Thu 2025-11-20 18:10:23 UTC
        RTC time: Thu 2025-11-20 18:10:23
       Time zone: UTC (UTC, +0000)
System clock automatic: yes
    NTP enabled: active
  RTC in Local TZ: no
[admosolov@server.admosolov.net ~]$
```

Рис. 1: Проверка времени на сервере

Проверка времени на клиенте

Затем я выполняю аналогичную проверку на машине `client`. Здесь часовой пояс установлен как `Europe/Moscow` (`MSK, +0300`), системное время и RTC синхронизированы.

```
[admosolov@client.admosolov.net ~]$ timedatectl
    Системное время: Thu 2025-11-20 21:16:03 MSK
    Универсальное время: Thu 2025-11-20 18:16:03 UTC
        RTC время: Thu 2025-11-20 18:16:04
            Время зонки: Europe/Moscow (MSK, +0300)
Синхронизация супервизором: yes
    NTP активен: active
        RTC в реальном времени: no
[admosolov@client.admosolov.net ~]$
```

Рис. 2: Проверка времени на клиенте

Смена часового пояса на Лондон

На сервере я ищу доступные часовые пояса, содержащие “London”, и меняю часовой пояс на Europe/London с помощью команды `sudo timedatectl set-timezone`.

```
[admosolov@server.admosolov.net ~]$ timedatectl list-timezones | grep -i london  
Europe/London  
[admosolov@server.admosolov.net ~]$ sudo timedatectl set-timezone Europe/London
```

Рис. 3: Смена часового пояса на Лондон

Возврат часового пояса на Москву

После этого я возвращаю сервер в московский часовой пояс (`Europe/Moscow`) и проверяю изменения командой `timedatectl`. Теперь сервер работает в зоне MSK (+0300).

```
[admosolov@server.admosolov.net ~]$ sudo timedatectl set-timezone Europe/Moscow
[admosolov@server.admosolov.net ~]$ timedatectl
          Local time: Thu 2025-11-20 18:13:23 MSK
      Universal time: Thu 2025-11-20 18:13:23 UTC
        RTC time: Thu 2025-11-20 18:13:23
       Time zone: Europe/Moscow (MSK, +0300)
System clock synchronized: yes
          NTP enabled: active
      RTC on power loss: no
```

Рис. 4: Возврат часового пояса на Москву

Команда date на сервере

Проверяю текущее системное время на сервере с помощью команды date.
Время отображается корректно в формате MSK.

```
[admosolov@server.admosolov.net ~]$ date  
Thu Nov 20 09:16:24 PM MSK 2025  
[admosolov@server.admosolov.net ~]$ ■
```

Рис. 5: Команда date на сервере

Команда date на клиенте

Сравниваю время с клиентом, выполняя команду date на машине client.
Время совпадает с сервером (разница в пару секунд допустима).

```
[admosolov@client.admosolov.net ~]$ date  
Thu Nov 20 09:16:26 PM MSK 2025  
[admosolov@client.admosolov.net ~]$
```

Рис. 6: Команда date на клиенте

Форматирование вывода date на сервере

Изучаю возможности форматирования вывода команды date на сервере.
Пробую выводить только год, часы, полное время с миллисекундами, а также
время в формате RFC 2822 и в зоне UTC.

```
[admosolov@server.admosolov.net ~]$ date +%Y-%m-%d
2025-11-20
[admosolov@server.admosolov.net ~]$ date +%H
21
[admosolov@server.admosolov.net ~]$ date +%H:%M:%S
21:18:22
[admosolov@server.admosolov.net ~]$ date +%H:%M:%S.%3N
21:18:55.175
[admosolov@server.admosolov.net ~]$ date -R
Thu, 20 Nov 2025 21:19:11 +0300
[admosolov@server.admosolov.net ~]$ date +%A
Thursday
[admosolov@server.admosolov.net ~]$ date -u
Thu Nov 20 06:20:13 PM UTC 2025
[admosolov@server.admosolov.net ~]$ TZ='Europe/London' date
Thu Nov 20 06:20:42 PM GMT 2025
[admosolov@server.admosolov.net ~]$
```

Рис. 7: Форматирование вывода date на сервере

Форматирование вывода date на клиенте

Те же самые эксперименты с форматированием команды date провожу на клиентской машине, убеждаясь в идентичности функционала.

```
[admosolov@client.admosolov.net ~]$ date +%Y-%m-%d
2025-11-20
[admosolov@client.admosolov.net ~]$ date +%-H:%M:%S
21:18:30
[admosolov@client.admosolov.net ~]$ date +%-H:%M:%S.%3N
21:19:27.807
[admosolov@client.admosolov.net ~]$ date -R
Thu, 20 Nov 2025 21:19:33 +0300
[admosolov@client.admosolov.net ~]$ date +%-A
Thursday
[admosolov@client.admosolov.net ~]$ date -u
Thu Nov 20 06:20:21 PM UTC 2025
[admosolov@client.admosolov.net ~]$
```

Рис. 8: Форматирование вывода date на клиенте

Арифметика дат на сервере

На сервере проверяю возможности арифметики времени с помощью date --date. Вычисляю даты “завтра”, “через неделю”, “через 2 дня” и “неделю назад”.

```
[admosolov@server.admosolov.net ~]$ date --date "1 day"
Fri Nov 21 09:21:15 PM MSK 2025
[admosolov@server.admosolov.net ~]$ date --date "2 days"
Sat Nov 22 09:21:27 PM MSK 2025
[admosolov@server.admosolov.net ~]$ date --date "1 week"
Thu Nov 27 09:21:40 PM MSK 2025
[admosolov@server.admosolov.net ~]$ date --date "1 week ago"
Thu Nov 13 09:21:46 PM MSK 2025
[admosolov@server.admosolov.net ~]$ date --date "tomorrow "
Fri Nov 21 09:21:59 PM MSK 2025
[admosolov@server.admosolov.net ~]$ date --date "tomorrow"
Fri Nov 21 09:22:06 PM MSK 2025
[admosolov@server.admosolov.net ~]$ _
```

Рис. 9: Арифметика дат на сервере

Проверка hwclock на сервере

Проверяю показания аппаратных часов (RTC) на сервере с помощью команды `hwclock`. Они идут по UTC, что является стандартом для Linux-систем.

```
[root@server.admosolov.net ~]# hwclock  
2025-11-20 18:31:35.896897+00:00
```

Рис. 10: Проверка hwclock на сервере

Проверка hwclock на клиенте

Аналогично проверяю аппаратные часы на клиенте. Они также настроены на UTC и показывают корректное время.

```
[root@client.admosolov.net ~]# hwclock  
2025-11-20 18:32:50.349917+00:00  
[root@client.admosolov.net ~]#
```

Рис. 11: Проверка hwclock на клиенте

Установка chrony на сервере

Перехожу к настройке службы синхронизации. На сервере устанавливаю пакет chrony (в данном случае он уже был установлен).

```
[root@server.admosolov.net ~]# dnf -y install chrony
Extra Packages for Enterprise Linux 10 - x86_64
Extra Packages for Enterprise Linux 10 - x86_64
Rocky Linux 10 - BaseOS
Rocky Linux 10 - AppStream
Rocky Linux 10 - CRB
Rocky Linux 10 - Extras
Package chrony-4.6.1-1.el10.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

Рис. 12: Установка chrony на сервере

Проверка источников до настройки

До настройки проверяю источники времени командой `chronyc sources`. Видны внешние сервера, но демон не отвечает (`Cannot talk to daemon`), так как, вероятно, служба не запущена или заблокирована.

```
[root@server.admosolov.net ~]# chronyc sources
MS Name/IP address          Stratum Poll Reach LastRx Last sample
=====
** dot.kkursor.ru            2    7   377     85   -212us[-225us] +/-   23ms
^+ 39.178.111.109.sta.211.ru  1    8   377     18   -1618us[-1618us] +/-   29ms
^C^+ 51.250.110.169          3    7   377     86   -1106us[-1119us] +/-   34ms
506 Cannot talk to daemon
[root@server.admosolov.net ~]# qu
```

Рис. 13: Проверка источников до настройки

Настройка chrony.conf на сервере

Редактирую файл конфигурации /etc/chrony.conf на сервере. Добавляю строку allow 192.168.0.0/16, чтобы разрешить клиентам из локальной сети запрашивать время у этого сервера.

```
# increase the minimum number of selectable sources
# the system clock.
#minsources 2

# Allow NTP client access from local network.
allow 192.168.0.0/16

# Serve time even if not synchronized to a time source
#local stratum 10
```

Рис. 14: Настройка chrony.conf на сервере

Перезапуск службы и настройка firewall

Перезапускаю службу chronyd и настраиваю межсетевой экран (firewall), открывая доступ к службе NTP (порт 123 UDP) на постоянной основе.

```
[root@server.admosolov.net ~]# systemctl restart chronyd
[root@server.admosolov.net ~]# firewall-cmd --add-service=ntp --permanent
success
[root@server.admosolov.net ~]# firewall-cmd --reload
success
[root@server.admosolov.net ~]# ■
```

Рис. 15: Перезапуск службы и настройка firewall

Настройка chrony.conf на клиенте

На клиентской машине редактирую файл /etc/chrony.conf. Указываю сервер server.admosolov.net с параметром iburst для быстрой первичной синхронизации, остальные серверы удаляю или комментирую.

```
# Specify directory for log files.
logdir /var/log/chrony

# Select which information is logged.
#log measurements statistics tracking

server server.admosolov.net iburst
```

File menu: ^G Help, ^O Write Out, ^F Where Is
Edit menu: ^X Exit, ^R Read File, ^\ Replace

Рис. 16: Настройка chrony.conf на клиенте

Проверка источников на клиенте

После перезапуска службы на клиенте проверяю источники командой `chronyc sources`. Вижу, что локальный сервер (`server.admosolov.net`) определился и отмечен звездочкой `*`, что означает успешную синхронизацию.

```
[root@client.admosolov.net ~]# systemctl restart chronyd
[root@client.admosolov.net ~]# chronyc sources
MS Name/IP address          Stratum Poll Reach LastRx Last sample
=====
^? admosolov.net            3   6     3      1 +2542us[+2542us] +/-  9646us
[root@client.admosolov.net ~]# chronyc sources
MS Name/IP address          Stratum Poll Reach LastRx Last sample
=====
^* server.admosolov.net     3   6    37     10 -196us[ -458us] +/-   11ms
[root@client.admosolov.net ~]#
```

Рис. 17: Проверка источников на клиенте

Статистика синхронизации (tracking)

Смотрю подробную статистику синхронизации на клиенте командой `chronyc tracking`. Вижу Reference ID моего сервера, страту 4 и очень малую погрешность (offset).

```
[root@client.admosolov.net ~]# chronyc tracking
Reference ID      : C0A80164 (server.admosolov.net)
Stratum          : 4
Ref time (UTC)   : Thu Nov 20 18:47:08 2025
System time      : 0.001113357 seconds slow of NTP time
Last offset      : -0.000935379 seconds
RMS offset       : 0.000442909 seconds
Frequency        : 52.080 ppm slow
Residual freq    : -2.013 ppm
Skew              : 20.925 ppm
Root delay       : 0.014300395 seconds
Root dispersion  : 0.002505351 seconds
Update interval  : 65.0 seconds
Leap status       : Normal
[root@client.admosolov.net ~]#
```

Подготовка каталогов на сервере

Для закрепления результата создаю скрипты настройки. На сервере создаю структуру каталогов в /vagrant/provision/server/ntp/etc и копирую туда настроенный конфиг.

```
[root@server.admosolov.net ~]# cd /vagrant/provision/server
[root@server.admosolov.net server]# mkdir -p /vagrant/provision/server/ntp/etc
[root@server.admosolov.net server]# cp -R /etc/chrony.conf /vagrant/provision/server/ntp/etc/
[root@server.admosolov.net server]# cd /vagrant/provision/server
[root@server.admosolov.net server]# touch ntp.sh
[root@server.admosolov.net server]# chmod +x ntp.sh
[root@server.admosolov.net server]#
```

Рис. 19: Подготовка каталогов на сервере

Скрипт ntp.sh для сервера

Создаю скрипт ntp.sh для сервера. В нем прописываю установку chrony, копирование конфига из папки vagrant в /etc, настройку firewall и рестарт службы.

```
root@server:/vagrant/provision/server
GNU nano 8.1
#!/bin/bash
echo "Provisioning script $0"

echo "Install needed packages"
dnf -y install chrony

echo "Copy configuration files"
cp -R /vagrant/provision/server/ntp/etc/* /etc

restorecon -vR /etc
echo "Configure firewall"
```

Подготовка каталогов на клиенте

Аналогичные действия выполняю для клиента: создаю структуру папок для хранения конфига.

```
[root@client.admosolov.net ~]# cd /vagrant/provision/client
[root@client.admosolov.net client]# mkdir -p /vagrant/provision/client/ntp/etc
[root@client.admosolov.net client]# cp -R /etc/chrony.conf /vagrant/provision/client/ntp/etc/
[root@client.admosolov.net client]# cd /vagrant/provision/client
[root@client.admosolov.net client]# touch ntp.sh
[root@client.admosolov.net client]# chmod +x ntp.sh
[root@client.admosolov.net client]#
```

Рис. 21: Подготовка каталогов на клиенте

Скрипт ntp.sh для клиента

Пишу скрипт ntp.sh для клиента. Он копирует клиентский конфиг (где указан мой сервер) и перезапускает демон chrony.

```
root@client:/vagrant/provision/client
GNU nano 8.1
#!/bin/bash
echo "Provisioning script $0"

echo "Copy configuration files"
cp -R /vagrant/provision/client/ntp/etc/* /etc

restorecon -vR /etc

echo "Restart chronyd service"
systemctl restart chronyd
```

Рис. 22: Скрипт ntp.sh для клиента

Обновление Vagrantfile для сервера

Теперь редактирую основной Vagrantfile. В секцию конфигурации сервера добавляю блок `server.vm.provision`, запускающий созданный shell-скрипт.

```
108
109
110     server.vm.provision "server ssh",
111         type: "shell",
112         preserve_order: true,
113         path: "provision/server/ssh.sh"
114
115     server.vm.provision "server ntp",
116         type: "shell",
117         preserve_order: true,
118         path: "provision/server/ntp.sh"
119
120 end
121
122 ## Client configuration
123 config.vm.define "client", autostart: false do |client|
124     client.vm.box = "rocky10"
```

Рис. 23: Обновление Vagrantfile для сервера

Обновление Vagrantfile для клиента

В секцию конфигурации клиента в Vagrantfile также добавляю запуск соответствующего скрипта провижининга.

```
157
160     preserve_order: true,
161     path: "provision/client/mail.sh"
162
163     client.vm.provision "client ntp",
164         type: "shell",
165         preserve_order: true,
166         path: "provision/client/ntp.sh"
167     end
168 end
169
```

Рис. 24: Обновление Vagrantfile для клиента

Выводы

В ходе выполнения лабораторной работы я освоил управление системным временем в Linux с помощью утилит `timedatectl`, `date` и `hwclock`. Я настроил локальный NTP-сервер на базе `chrony`, открыл к нему доступ через `firewall` и успешно синхронизировал с ним клиентскую машину. Также я автоматизировал процесс настройки, написав `bash`-скрипты и подключив их в `Vagrantfile`.