

Тестирование backend

Лекция 4

Ярослав Чередниченко

Настройка Linux & Linux из Python

1. Настройка Linux: inode, hard & symlink, systemd, selinux, firewalld & iptables, sshd, cron, date & timedatectl, ntp, alias.
2. Что такое процессы и потоки в Linux.
3. Действия на удаленной машине средствами Python (Paramiko).
4. Ответы на вопросы: lsof, iostat, wireshark, bash (grep, sort, uniq).
5. Домашнее задание.



UNIX
это просто

inode

inode или индексный дескриптор - это структура данных файловой системы (далее ФС), хранящая мета информацию о файлах, каталогах или других объектах файловой системы

pwd

```
/Users/y.chednichenko/PycharmProjects/qa-python
```

ls -li

```
total 32
```

| | | | | | | | |
|-------------------|------------|---|----------------|-------|-----|--------------|----------------|
| 8644664582 | -rw-r--r-- | 1 | y.chednichenko | staff | 59 | Mar 14 16:07 | README.md |
| 8644664583 | drwxr-xr-x | 4 | y.chednichenko | staff | 128 | Mar 14 16:07 | homework_rules |
| 8644664594 | drwxr-xr-x | 5 | y.chednichenko | staff | 160 | Mar 14 16:07 | lection1 |
| 8644664608 | drwxr-xr-x | 3 | y.chednichenko | staff | 96 | Mar 14 16:07 | lection2 |

df -i

| Filesystem | Inodes | IUsed | IFree | IUse% | Mounted on |
|------------|--------|-------|--------|-------|------------|
| tmpfs | 254657 | 17 | 254640 | 1% | /dev |

hardlink

«жёсткая ссылка», является тем же файлом,
на который ссылается

```
touch test_qa
```

```
ln test_qa hardlink1
```

```
ls -li
```

```
total 0
```

```
8648298317 -rw-r--r--  2 y.cherednichenko  wheel  0 Apr 18 18:51 hardlink1
```

```
8648298317 -rw-r--r--  2 y.cherednichenko  wheel  0 Apr 18 18:51 test_qa
```

```
ln test_qa hardlink2
```

```
ls -li
```

```
total 0
```

```
8648298317 -rw-r--r--  3 y.cherednichenko  wheel  0 Apr 18 18:51 hardlink1
```

```
8648298317 -rw-r--r--  3 y.cherednichenko  wheel  0 Apr 18 18:51 hardlink2
```

```
8648298317 -rw-r--r--  3 y.cherednichenko  wheel  0 Apr 18 18:51 test_qa
```

symlink

(symbolic link)

Символическая («мягкая») ссылка - это специальный файл в ФС, в котором хранится только путь к файлу

```
touch test_qa2
ln -s test_qa2 symlink1
ls -li
total 0
8648298317 -rw-r--r--  3 y.cherednichenko  wheel  0 Apr 18 18:51 hardlink1
8648298317 -rw-r--r--  3 y.cherednichenko  wheel  0 Apr 18 18:51 hardlink2
8648298735 lrwxr-xr-x  1 y.cherednichenko  wheel  8 Apr 18 19:00 symlink1 -> test_qa2
8648298317 -rw-r--r--  3 y.cherednichenko  wheel  0 Apr 18 18:51 test_qa
8648298727 -rw-r--r--  1 y.cherednichenko  wheel  0 Apr 18 18:59 test_qa2
```

Основные типы файлов, которые отображаются командой ls:

- обычный файл

d каталог

l символическая ссылка

s сокет

hardlink vs symlink – ключевые различия

1. **hardlink** не может указывать на файл в другой ФС, так как inode может принадлежать только одной ФС, а **symlink** — может.
2. При редактировании файла-ссылки в случае с **hardlink'ом** — изменятся оба файла, так как это один и тот же объект, а в случае с **symlink'а** — можно изменять его имя, атрибуты, направить его на другой файл и при этом оригинальный файл не будет затронут.
3. **Жёсткая ссылка** не может указывать на каталог.

Удаление ссылок (rm)

```
ls -li
```

```
8648298735 lrwxr-xr-x  1 y.cherednichenko  wheel   8 Apr 18 19:00 symlink1 -> test_qa2
8648298727 -rw-r--r--  1 y.cherednichenko  wheel   0 Apr 18 18:59 test_qa2
```

```
rm test_qa2
```

```
ls -la
```

```
8648298735 lrwxr-xr-x  1 y.cherednichenko  wheel   8 Apr 18 19:00 symlink1 -> test_qa2
```

```
file test_qa2
```

```
test_qa2: cannot open `test_qa2' (No such file or directory)
```

```
ls -li
```

```
total 0
```

```
8648298317 -rw-r--r--  3 y.cherednichenko  wheel   0 Apr 18 18:51 hardlink1
8648298317 -rw-r--r--  3 y.cherednichenko  wheel   0 Apr 18 18:51 hardlink2
8648298317 -rw-r--r--  3 y.cherednichenko  wheel   0 Apr 18 18:51 test_qa
```




systemd
(менеджер загрузки)

Unit
(описание сервиса)

systemd – менеджер системы и служб для Linux

unit-file – файл, описывающий настройки сервиса, например, как и когда запускать сервис

`.service`

`/usr/lib/systemd/system/` – юниты из установленных пакетов RPM – `nginx`, `mysql`, `etc`

`/etc/systemd/system/` – юниты, созданные системным администратором (наш случай)

systemd (unit-file)

[Unit]

Description=The NGINX HTTP and reverse proxy server

After=syslog.target network.target remote-fs.target nss-lookup.target

[Service]

Type=forking

PIDFile=/run/nginx.pid

ExecStartPre=/usr/sbin/nginx -t

ExecStart=/usr/sbin/nginx

ExecReload=/usr/sbin/nginx -s reload

ExecStop=/bin/kill -s QUIT \$MAINPID

PrivateTmp=true

[Install]

WantedBy=multi-user.target

Save this file as /lib/systemd/system/nginx.service

systemd (systemctl)

systemctl – утилита для управления
службами

systemctl status nginx

- nginx.service - The nginx HTTP and reverse proxy server
Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; vendor preset: disabled)
Active: inactive (dead)

systemctl enable nginx

Created symlink from /etc/systemd/system/multi-user.target.wants/nginx.service to /usr/lib/systemd/system/nginx.service

systemctl start nginx

systemctl -l status nginx

- nginx.service - The nginx HTTP and reverse proxy server
Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
Active: active (running) since Sat 2020-04-18 15:47:57 EDT; 2s ago
Process: 2208 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
Process: 2205 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
Process: 2204 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
Main PID: 2210 (nginx)
CGroup: /system.slice/nginx.service
└─2210 nginx: master process /usr/sbin/nginx
└─2211 nginx: worker proces

systemd (systemctl, journalctl)

journalctl – система журналирования systemd

Посмотреть список запущенных юнитов (фильтр по sshd)
systemctl list-units | grep sshd

Посмотреть какие юниты зафэйлились при старте
systemctl --failed

Посмотреть логи за последние полчаса:

```
journalctl --since "30 min ago"
```

Просмотр лога по конкретному юниту:

```
journalctl -u sshd --since today
```

Посмотреть логи за определенный интервал:

```
journalctl --since="2020-04-18" --until="2020-04-19"
```

SELinux

SELinux — это система принудительного контроля доступа, реализованная на уровне ядра

DAC(Discretionary Access Control) — дискретный механизм доступа, подразумевает наличие владельца у каждого защищаемого объекта, который определяет права доступа пользователей к объекту

semanage - средство управления политикой SELinux



Selinux. Semanage

Для того чтобы установить менеджер политик:

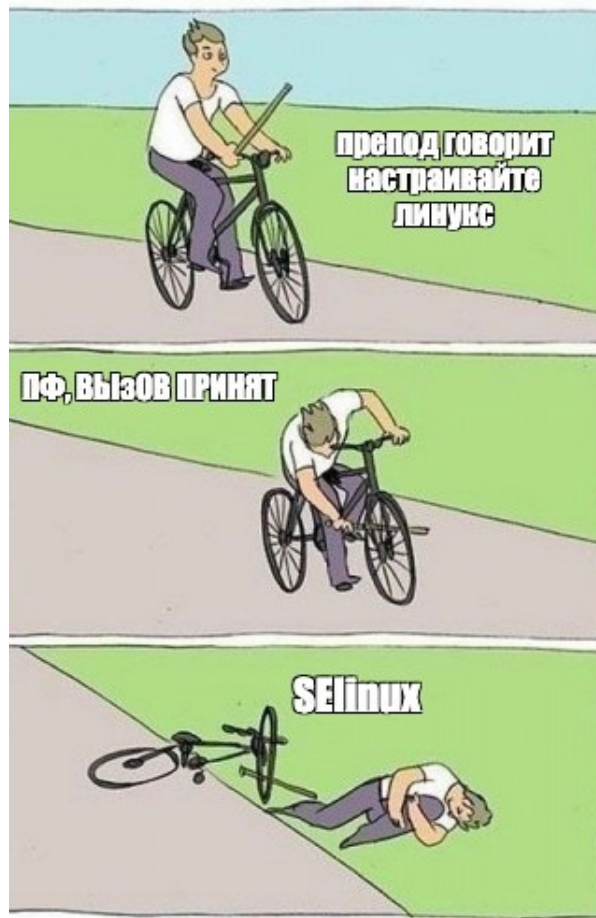
```
yum install -y polycoreutils-python
```

Вывести список всех портов, фильтруем по 22:

```
semanage port -l | grep 22
```

Чтобы разрешить использовать кастомный порт для подключения по SSH:

```
semanage port -a -t ssh_port_t -p tcp 2022
```



firewalld & iptables

Firewalld — служба для динамического управления межсетевым экраном с поддержкой доверенных зон сетевых соединений или интерфейсов

Iptables — утилита командной строки, предназначенная для управления межсетевым экраном

Netfilter — это набор программных хуков внутри ядра Linux, которые позволяют модулям ядра регистрировать функции обратного вызова от стека сетевых протоколов.

Hook - это программный элемент, который позволяет перехватывать функции обратного вызова в чужих процессах.



firewalld

```
firewall-cmd --state
```

Running

```
firewall-cmd --get-default-zone
```

Public

```
firewall-cmd --list-all
```

```
public (default, active)
```

```
interfaces: eth0 eth1
```

```
sources:
```

```
services: dhcpv6-client ssh
```

```
ports:
```

```
masquerade: no
```

```
forward-ports:
```

```
icmp-blocks:
```

```
rich rules:
```

public зона - публичная сеть, которой нельзя доверять, однако поддерживает входящие соединения в индивидуальном порядке

public является зоной по умолчанию и единственной активной зоной к этой зоне привязаны сетевые интерфейсы eth0 и eth1

поддерживает трафик протоколов DHCP и SSH

firewalld zones

drop: самый низкий уровень доверия сети, весь входящий трафик сбрасывается без ответа, поддерживаются только исходящие соединения.

trusted: всем машинам в сети можно доверять.

external: зона внешних сетей; поддерживает маскировку NAT, благодаря чему внутренняя сеть остается закрытой, но с возможностью получения доступа.

work/home/internal — максимальное доверие к компьютерам, уверенность в том, что они не причинят вреда нашему компьютеру, разрешается устанавливать только конкретные входящие соединения

firewalld

Добавить порт:

```
firewall-cmd --add-port 2022/tcp
```

Удалить порт:

```
firewall-cmd --zone=public --remove-port=22/tcp --permanent
```

Разрешить http трафик:

```
firewall-cmd --permanent --zone=public --add-service=http
```

Перезагрузить настройки после внесенных изменений:

```
firewall-cmd --reload
```

sshd

```
yum install openssh openssh-server openssh-clients openssl-libs
```

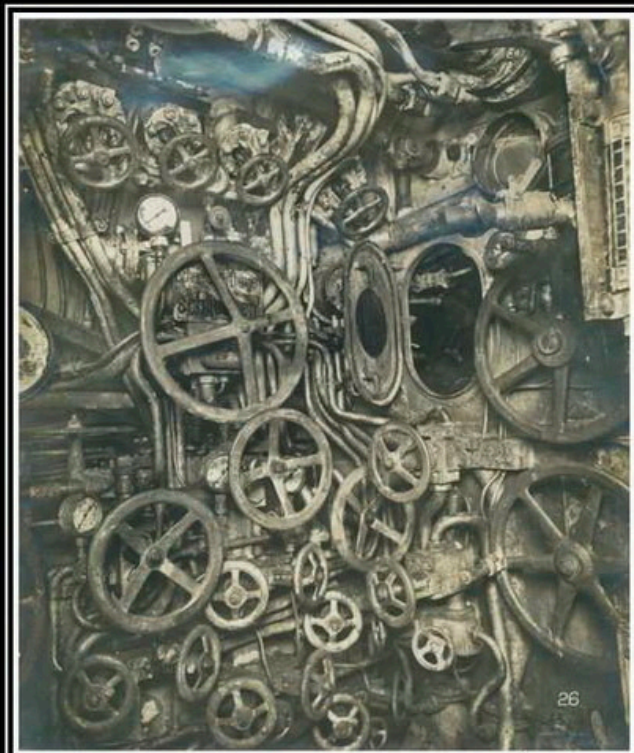
/etc/ssh/sshd_config

```
cp /etc/ssh/sshd_config /etc/ssh/sshd_config.orig
```

```
Port 2022
```

```
PermitRootLogin no
```

```
LogLevel INFO (DEBUG, VERBOSE, ERROR, etc.)
```



Linux

простота и удобство настройки

cron

cron –планировщик заданий

```
.----- минута (0 – 59)
| .----- час (0 – 23)
| | .----- день месяца (1 – 31)
| | | .----- месяц (1 – 12) ИЛИ jan,feb,mar ...
| | | | .----- день недели (0 – 6) (Воскресенье=0 или 7) ИЛИ sun,mon,tue
| | | | |
* * * * * команда для выполнения
```

cron

Посмотреть, какие команды уже добавлены:

```
crontab -l
```

Добавить команду:

```
crontab -e
```

```
30 22 * * 1 /home/y.cherednichenko/health-check.py
```

Минуты_Часы_ДеньМесяца_Месяц_ДеньНедели_Команда

Запустим в 3:15 каждый пн, вт, пт

```
15 3 * * 1,3,5
```

Каждую минуту

```
* * * * *
```

Каждые 5 минут

```
*/5 * * *
```

Каждый день в 2:30

```
30 2 * * *
```

Каждый год 1 января

```
0 0 1 1 *
```


date & timedatectl

date

```
Sun Apr 19 09:14:27 EDT 2020
```

timedatectl

```
Local time: Sun 2020-04-19 09:22:24 EDT
Universal time: Sun 2020-04-19 13:22:24 UTC
RTC time: Sun 2020-04-19 13:22:21
Time zone: America/New_York (EDT, -0400)
NTP enabled: yes
NTP synchronized: yes
RTC in local TZ: no
DST active: yes
Last DST change: DST began at
                  Sun 2020-03-08 01:59:59 EST
                  Sun 2020-03-08 03:00:00 EDT
Next DST change: DST ends (the clock jumps one hour backwards) at
                  Sun 2020-11-01 01:59:59 EDT
                  Sun 2020-11-01 01:00:00 EST
```

Изменить дату и время:

```
date -s "19 APR 2020 16:30:00"
```

Изменить только время:

```
date -s 16:27
```

Посмотреть список time зон

```
timedatectl list-timezones
```

Установить time зону

```
timedatectl set-timezone Europe/Moscow
```

NTP (Network Time Protocol)

```
yum install ntp
```

```
/etc/ntp.conf
```

```
ntpq -p
      remote                refid              st t when poll reach    delay    offset  jitter
=====
*ntp.kaluga.net  .GPS.                1 u   37   64    1    6.704   -2.443   1.611
```

remote - удалённые сервера, с которыми вы синхронизируете время

when - время, когда последний раз сервер ответил нам

poll - частота опроса

delay - значение в миллисекундах показывает время между отправкой и получения ответа (round trip time - RTT)

offset - смещение в миллисекундах между вами и серверами времени

alias

alias – встроенная команда оболочки, которая используется для создания альтернативных версий обычных команд

Создание/удаление alias'ов

```
echo 'alias del="rm"' >> ~/.bashrc
. ~/.bashrc
unalias del
del
-bash: del: command not found
```

Примеры

Подключение к серверу:

```
alias webserver='ssh test-server@192.168.1.200'
```

История команд, последние 6 строк:

```
alias h6='history 6'
```

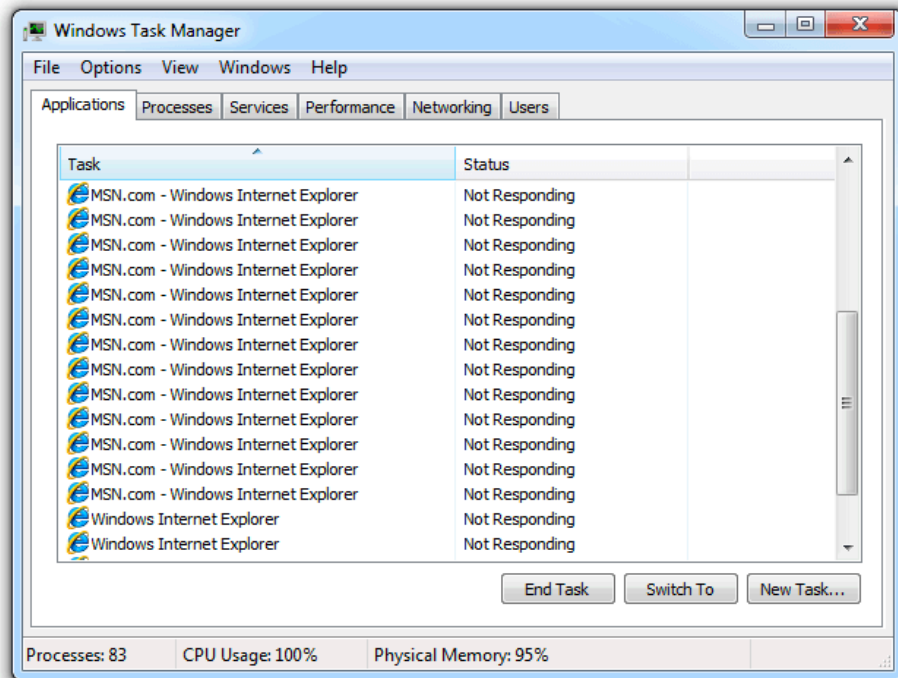
Пинг, повторяем 5 раз

```
alias ping='ping -c 5'
```

Информация по памяти

```
alias meminfo='free -m -l -t'
```

Процессы и потоки



**ТЫ ЧТО-НИБУДЬ ПОНЯЛ
ИЗ ЛЕКЦИИ?**



Я НЕ СМОТРЕЛ



Я ТОЖЕ



**Спасибо
за внимание!**

Ярослав Чередниченко
y.cherednichenko@corp.mail.ru