

Разворачивание Spring Boot приложений (nginx, systemd, firewall, journalctl)



Григорий
Вахмистров




Григорий Вахмистров

Backend Developer в Tennisi.bet

План занятия

1. [Основы работы в терминале](#)
2. [Системы инициализации](#)
3. [systemd](#)
4. [Nginx](#)
5. [Итоги](#)
6. [Домашнее задание](#)



Основы работы в терминале

FHS

FHS (filesystem hierarchy standard) - стандарт иерархии файловой системы в UNIX.

Некоторые характеристики:

- единый корень всей файловой системы , '/', forward slash
- все директории в дереве внутри относительно корня
- если вы из мира Windows – большое отличие в том, что разделы также находятся внутри относительно единого корня
- стандарт Filesystem Hierarchy Standard
 - /bin, /sbin, /etc...
- многие директория называются понятно:
 - /home для домашних каталогов пользователей
 - /root для домашнего каталога аккаунта администратора
 - /tmp для временных, temporary, файлов

Базовая работа в терминале

Базовая структура интерактивного взаимодействия – команда, аргументы этой команды.

Начинаем знакомиться с некоторыми командами и действиями в терминале Linux:

- `echo Hello world`
- `echo $USER`
- управление с клавиатуры: Ctrl + L, Up/Down, Ctrl + R
- автодополнение по Tab: bash autocompletion (`\``)

Работа с директориями

- `pwd` – текущая директория (present working directory)
- `cd` – изменить директорию (change directory)
 - `./` `..`
 - `-`
- `ls` – показать содержимое (list)
 - `-l`
 - `-a`
 - `-la`, сочетание опций
- скрытые объекты с точкой

Работа с файлами

Простейший просмотр содержимого:

- `cat` – conCATenate
- `head / tail` – начало и конец файла
 - `-n`
 - `<число>, -/+<число>`

Просмотр содержимого в режиме пейджера (постранично):

- `less` vs. `more`
 - `-i`
 - `-N`

Некоторые ключи `grep` для фильтрации (поиска) содержимого файлов:

- `-c`
- `-i`
- `-v`
- `-f (diff, diff -u + sha256sum)`

Работа с директориями и файлами

- `mkdir` – make directory
 - `-p`
- `touch`
- `cp` – copy
 - `-v`
 - `-r`
 - `*(strace git add *)`
- `mv` – move
- `rm` – remove
 - `-d`
 - `-r`

Команды для создания, перемещения и удаления директорий и файлов

- `mkdir` – make directory
 - `-p`
- `touch`
- `cp` – copy
 - `-v`
 - `-r`
 - `*(strace git add *)`
- `mv` – move
- `rm` – remove
 - `-d`
 - `-r`



Системы инициализации

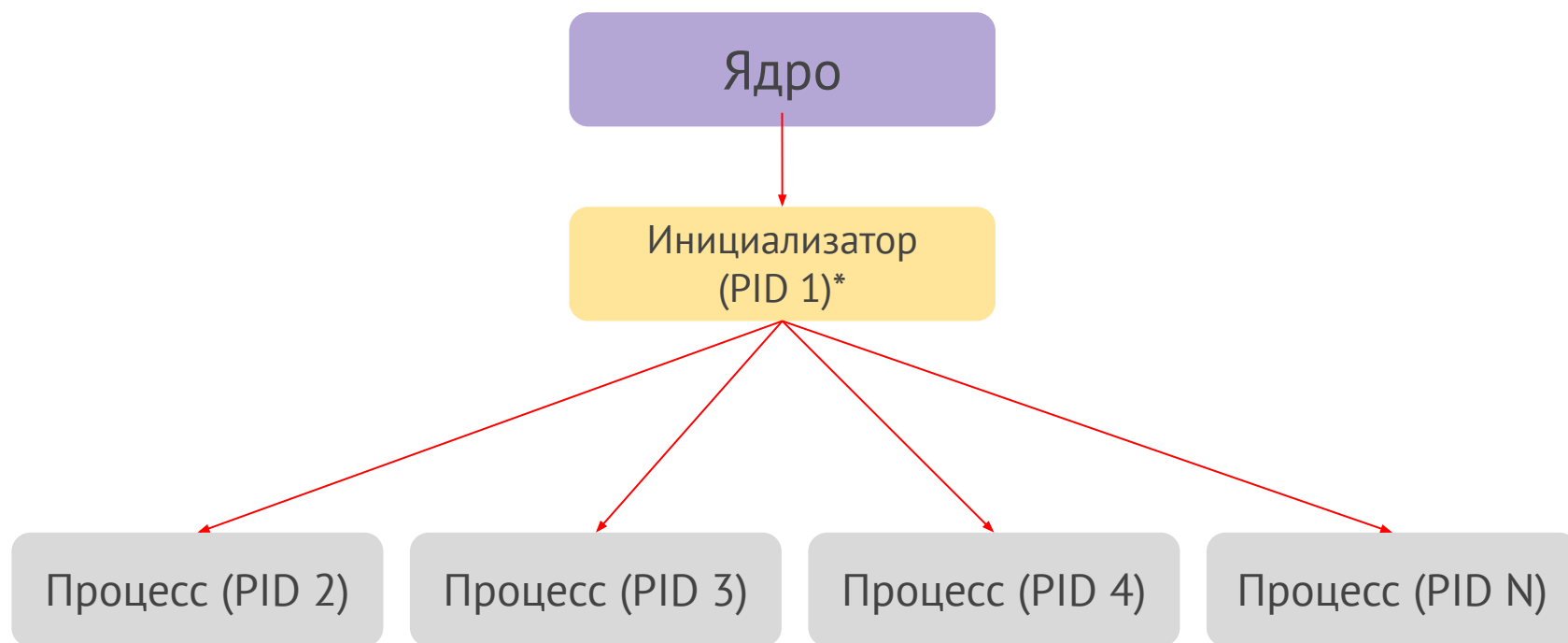
Системы инициализации

В операционной системе Linux и других системах семейства Unix после завершения загрузки ядра начинается инициализация Linux системы, сервисов и других компонентов. Новые процессы в Linux создаются клонированием родительского процесса. Из этого правила есть понятное исключение – первый процесс в системе, который ядро создает самостоятельно и назначает ему PID 1, – `init`.

`init` – собирательное название для контроллера инициализации ОС, а не конкретная технология.

Системы инициализации

Примерно так это выглядит:



* PID - это process id, уникальный идентификатор процесса

Системы инициализации

Система инициализации отвечает за:

- достижение корректного состояния на разных этапах загрузки;
- запуска служб самой ОС и прикладных программ в нужном порядке;
- монтирование файловых систем;
- обратный процесс при выключении ОС;

Системы инициализации

Виды систем инициализации:

- **SysV**
- **systemd** (наиболее популярная система инициализации)

Системы инициализации

Проблемы SysV:

- линейное исполнение и отсутствие зависимостей
- нет □ эффективного контроля над процессом
- двойной запуск(или совсем не запуск)
- невозможно перезапустить □ упавший процесс



systemd

systemd является многофункциональной системой инициализации. Она заменяет множественные сервисы ОС: логирование, планировщик задач cron и многое другое, что вообще никак не связано с системой инициализации.

Unit's

Юниты - базовая концепция в **systemd**. Они представляет собой текстовый файл следующего формата:

```
[Название секции в квадратных скобках]  
имя_переменной=значение
```

Для создания простейшего юнита надо описать три секции: [Unit], [Service], [Install]

systemctl

systemctl является утилитой для управления инициализатором **systemd**. Его базовый команды:

- `systemctl list-units --all` (посмотреть все *юниты* под управлением `systemd`)
- `systemctl status nginx.service` (посмотреть статус работы *сервиса*) сразу же виден путь расположения *юнит-файла*, который отвечает за сервис (`/lib/systemd/system/nginx.service` в нашем случае)
- `systemctl start/stop/restart/reload nginx` (`.service` можно опустить если типа юнита один)
- `systemctl enable/disable nginx` (автозапуск сервиса включен/выключен)

journalctl

journalctl является утилитой для управления логированием юнитов в **systemd**. Пример базовой команды:

- `journalctl -b -u nginx.service` (посмотреть логи сервиса nginx, которые были записаны с момента загрузки, b от boot)



Nginx



Nginx

Nginx - программное обеспечение, основное назначение которого — самостоятельный HTTP-сервер. Также, его часто используют как обратный прокси-сервер.

Nginx

- Устанавливается командой:
apt-get install nginx
- Располагается в: **/etc/nginx**
- Основной конфигурационный файл: **nginx.conf**
- Состоит из модулей, которые настраиваются директивами, указанными в конфигурационном файле. Директивы делятся на простые и блочные.

Секция http

Предоставляет контекст конфигурационного файла, в котором указываются директивы HTTP-сервера.

```
http {  
    server {  
        ...  
    }  
}
```


Секция server

Задаёт конфигурацию для виртуального сервера. С помощью директивам [listen](#) описываются все адреса и порты, на которых нужно принимать соединения для этого сервера

```
http {  
    server {  
        listen 80;  
    }  
}
```

Секция location

Устанавливает конфигурацию в зависимости от URI запроса.

```
http {  
    server {  
        listen 80;  
        location / {  
            ...  
        }  
        location /api {  
            ...  
        }  
    }  
}
```



Итоги

Итоги

- Мы с вами познакомились с понятием с основами работы в терминале.
- Также, мы познакомились с системой инициализации, в частности с `systemd`, которая является корневым элементом для старта нашей операционной системы.
- И еще мы узнали, что такое `nginx`, как с ним работать и как настраивать.

Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задавайте **в чате** мессенджера
- Задачи можно сдавать **по частям**.
- Зачёт по домашней работе проставляется после того, как **приняты все задачи**.

**Задавайте вопросы и
пишите отзыв о лекции!**

Григорий Вахмистров