

Онлайн образование

otus.ru



Проверить, идет ли запись

Меня хорошо видно && слышно?



Тема вебинара

Инициализация системы. Systemd



Лавлинский Николай

Технический директор “Метод Лаб”

<https://www.methodlab.ru/>

<https://www.youtube.com/c/NickLavlinsky>

https://www.youtube.com/channel/UCf1P_nESQ6hPITwkH5Vik1A

<https://vk.com/nick.lavlinsky>

Преподаватель



Лавлинский Николай

Более 15 лет в веб-разработке

Преподавал в ВУЗе более 10 лет
Более 3 лет в онлайн-образовании

Специализация: оптимизация производительности,
ускорение сайтов и веб-приложений

Правила вебинара



Активно
участвуем



Off-topic обсуждаем
в Slack #general



Задаем вопрос
в чат или голосом



Вопросы вижу в чате,
могу ответить не сразу

Условные обозначения



Индивидуально



Время, необходимое
на активность



Пишем в чат



Говорим голосом

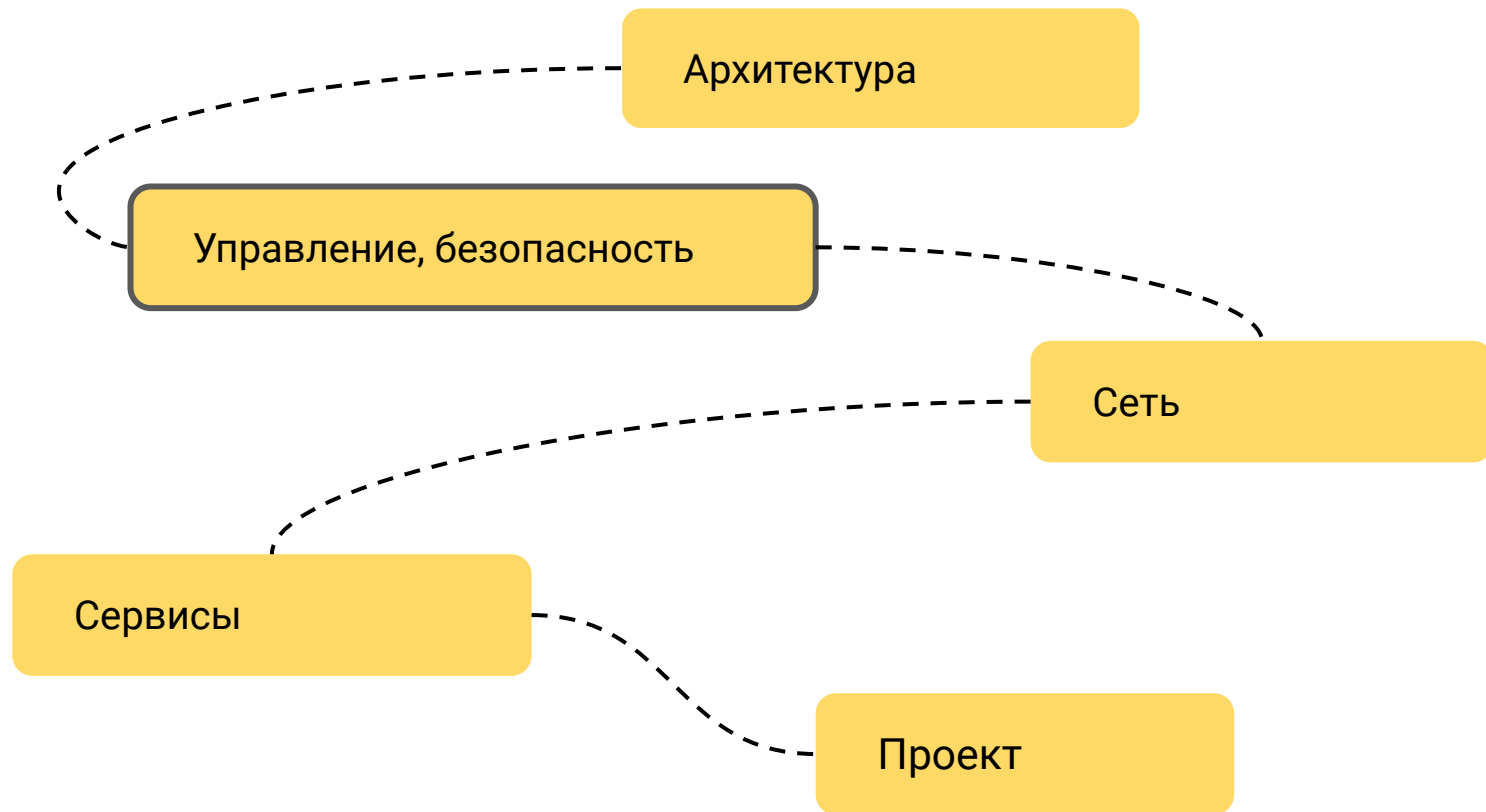


Документ

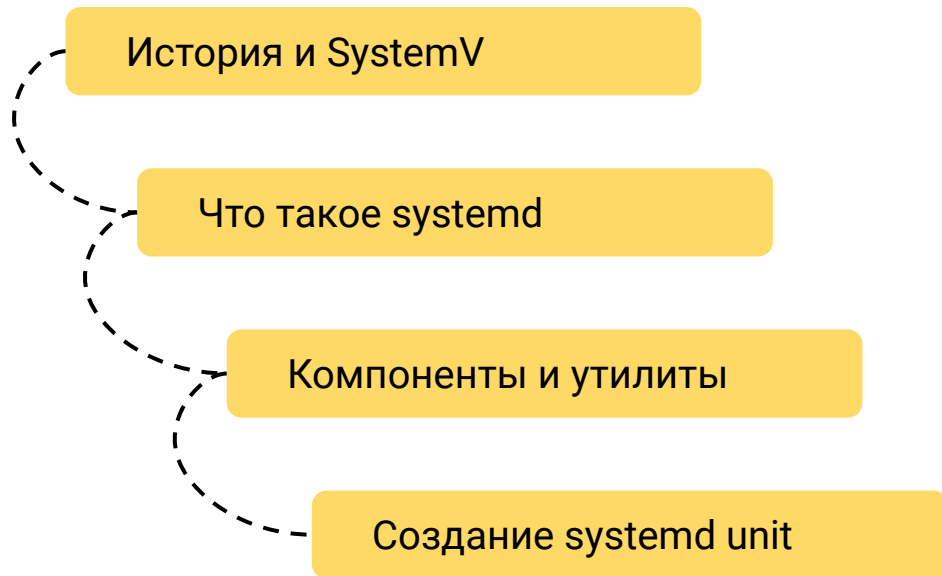


Ответьте себе или
задайте вопрос

Карта курса



Маршрут вебинара



Цели вебинара

После занятия вы сможете

1. Понимать различие систем инициализации
2. Использовать основные утилиты systemd
3. Изучить состав и синтаксис systemd unit

Смысл

Зачем вам это уметь

1. Управлять сервисами в системе
2. Создавать собственные systemd unit
3. Решать проблемы, работать с логами



История систем инициализации

AT&T System V

- Система инициализации по уровням (run level)
- Линейный запуск (последовательный, без зависимостей)
- Инициализационные скрипты (init script)
- Нет прямого контроля процессов
- Нет перезапуска процесса в случае сбоя



AT&T System V - уровни исполнения

- Конфигурация: /etc/inittab
- Файлы скриптов: /etc/init.d
- Запуск: /sbin/init
- Уровни исполнения
 - 0 - halt
 - 1 - single user
 - 2 - multiuser w/o Network
 - 3 - full multiuser
 - 5 - X11 GUI
 - 6 - reboot

```
RP0: No File System found on 'disk.hp', skipping autosizing
/home/zerocool32/Downloads/sysvr1/boot.ini-5> att tm0 dist.tape
TM0: Tape Image 'dist.tape' scanned as SIMH format
#0=unix

UNIX/sysV: unixgdtm
real mem = 262144 bytes
avail mem = 185280 bytes

INIT: SINGLE USER MODE
# init 2

INIT: New run level: 2
Is the date Wed Dec  1 15:21:37 EST 1982 correct? (y or n) y
Do you want to check the file systems? (y or n) n
-su: /usr/lib/acct/startup: cannot execute
process accounting started
errdemon started
cron started

Console Login: root
UNIX System V
# █
```

AT&T System V – init скрипт

```
1 #!/bin/sh
2
3 case $1 of
4 start) /usr/bin/mydaemon -p /var/run/mydaemon.pid
5 ;;
6 stop) kill $(</var/run/mydaemon.pid)
7 ;;
8 restart) $0 stop; $0 start
9 ;;
10 *) echo "what do you want from me, dude?"
11 esac
```



AT&T System V – управление сервисами

System V service management

— □ ×

```
1 service nginx start|stop|reload|restart
2
3 /etc/init.d/nginx start|stop|reload|restart
4
5 update-rc.d nginx enable|remove
6
7 chkconfig nginx on
8
9 # Upstart
10 initctl start|stop apache2
11 start|stop apache2
```



Systemd – новая система инициализации

Systemd

- Система инициализации (замена SysV, Upstart)
- Параллельный запуск сервисов
- Зависимость сервисов
- Мониторинг запущенных процессов, перезапуск
- Возможность снимать снимки состояния сервисов
- Регулярные задачи (замена cron)
- Управление временем (timedated, timesyncd)
- Контроль и ограничение процессов (cgroups)
- Управление сетью (systemd-networkd)
- Управление журналированием (systemd-journald)
- Менеджер устройств (systemd-udev)
- UEFI-загрузчик (systemd-boot)



Systemd компоненты

УТИЛИТЫ

Демоны

Таргеты

Базовые

компоненты

Библиотеки

Компоненты ядра

systemd Utilities

systemctl journalctl notify analyze cglsg cgtop loginctl nspawn

systemd Daemons

systemd
journald networkd
logind user session

systemd Targets

bootmode basic multi-user graphical user-session
shutdown reboot dbus telephony dlog login user-session display service
tizen service

systemd Core

manager unit login namespace log
service timer mount target multiseat inhibit session pam
systemd snapshot path socket swap cgroup dbus

systemd Libraries

dbus-1 libpam libcap libcryptsetup tcpwrapper libaudit libnotify

Linux Kernel

cgroups autofs kdbus

Systemd: модули (unit)

- **.target** — позволяет группировать модули, воплощая концепцию уровней запуска
- **.service** — отвечает за запуск сервисов (служб), также поддерживает вызов интерпретаторов для исполнения пользовательских скриптов
- **.mount** — отвечает за монтирование файловых систем
- **.automount** — позволяет отложить монтирование файловых систем до фактического обращения к точке монтирования
- **.swap** — отвечает за подключение файла или устройства подкачки
- **.timer** — позволяет запускать модули по расписанию
- **.socket** — предоставляет службам поддержку механизма сокет-активации
- **.slice** — отвечает за создание контейнера cgroups
- **.device** — позволяет реагировать на подключение устройств
- **.path** — управляет иерархией файловой системы

Systemd: документация

- `systemctl --all | grep systemd`
- `man systemd.index || apropos systemd`
- <https://freedesktop.org/wiki/Software/systemd/>

Инициализация и режимы

Systemd: инициализация

- PID 1 – init – `/lib/systemd/systemd`
- Посмотреть: `ps -ef | head`
- Все процессы верхнего уровня имеют PPID 1
- Запуск происходит параллельно
- Статус всех модулей: `systemctl status`
- Контрольные группы: `systemd-cgls`
- Анализ скорости загрузки:
 - `systemd-analyze [time]`
 - `systemd-analyze blame`
 - `systemd-analyze critical-chain`

Systemd – режимы работы ОС

systemd	init
poweroff.target	0 - halt
rescue.target	1 - single user
multi-user.target	2 - multiuser w/o network
	3 - full multiuser
	4 - ?
graphical.target	5 - X11 GUI
reboot.target	6 - reboot



Systemd: управление режимами

- Переход: `systemctl start $target.target`
- Короткая форма: `systemctl $target`
- Примеры: `systemctl reboot` || `systemctl rescue`

Работа с журналами

journalctl – работа с логами

- Логирование в бинарном формате
- `journalctl -p err`
- `journalctl --since yesterday`
- `journalctl --since 19:00 --until "1 hour ago"`
- `journalctl --since "2023-04-07 20:30:00" \`
`--until "2023-04-08 15:25:00"`
- `journalctl -u ssh.service`
- Конфигурация: `/etc/systemd/journald.conf`

Ограничение и изоляция

cgroups: ограничение и изоляция

- **cgroup** – контрольная группа (сервис / сессия), реализует ограничения
- **slice** – объект, представляющий иерархию (сервиса / сессии)
- **scope** – объект в slice, группирующий процессы
- Просмотр cgroup: `ps -afxo pid,user,comm,cgroup`
- Статистика по ресурсам: `systemd-cgtop`
- Список slice: `ls -l /lib/systemd/system/*.slice`
- Преимущества:
 - Эффективный контроль над процессами
 - Независимость от pid-файлов
 - Решение проблем двойного запуска, abandoned childs

Модули (unit)

Типы unit 1/2

Тип	Назначение
target	Ничего не описывает, группирует другие юниты
service	Аналог демона (или то, что можно запустить)
timer	Аналог cron (запуск другого юнита, default - *.service)
device	Факт подключения устройства (sysfs-имя устройства)
mount	Точка монтирования файловой системы
automount	Точка автмонтирования (*.mount)



Типы unit 2/2

Тип	Назначение
socket	Запуск юнита при подключении к указанному сокету (default - *.service)
path	Запуск юнита по событию доступа к пути (default - *.service)
slice	Группирует другие юниты в дереве cgroups
swap	Управление swap'ом (*.device)
snapshot	Снимки состояния сервисов
scope	“Области” или “границы”, заданные systemd



Каталоги systemd

<code>/lib/systemd/</code>	Основной каталог
<code>/lib/systemd/system/</code>	Юнит-файлы пакетов
<code>/lib/systemd/system/*.wants/</code>	Симлинки, предопределенные systemd
<code>/etc/systemd/</code>	Конфигурация
<code>/etc/systemd/system</code>	Юнит-файлы администратора
<code>/etc/systemd/system/*.wants/</code>	Симлинки при <code>systemctl enable</code>



Секции юнит-файла


```
1 [Unit]
2 Description=The Apache HTTP Server
3 After=network.target remote-fs.target nss-lookup.target
4 Documentation=https://httpd.apache.org/docs/2.4/
5
6 [Service]
7 Type=forking
8 Environment=APACHE_STARTED_BY_SYSTEMD=true
9 ExecStart=/usr/sbin/apachectl start
10 ExecStop=/usr/sbin/apachectl stop
11 ExecReload=/usr/sbin/apachectl graceful
12 PrivateTmp=true
13 Restart=on-abort
14
15 [Install]
16 WantedBy=multi-user.target
17
```



Шаблоны

Тип	Наименование
шаблон	<имя_службы>@.service
экземпляр	<имя_службы>@<экземпляр>.service
имя экземпляра в юните	%i – без спецсимволов, %I - чистое

Просмотр юнитов и их состояния



```
1 systemctl --help | grep list-
2
3 systemctl list-unit-files
4 systemctl list-dependencies
5 systemctl list-units
6 systemctl list-timers
7
8 systemctl --help | grep is-
9
10 systemctl is-enabled nginx
11 systemctl is-active ssh
```

Управление юнитами

```
1 # Управление юнитами
2 systemctl status|reload|start|stop|restart unit.service
3
4 # Управление автостартом
5 systemctl enable|disable unit.service
6 systemctl mask|unmask unit.service
7
8 # Обновление юнитов
9 systemctl daemon-reload
10
11 # Просмотр и модификация
12 systemctl cat|show unit.service
13 systemctl edit --full unit.service
```



Ограничения в `systemd`

Ограничения

- **Глобальные** – `/etc/systemd/system.conf` (`DefaultLimit*`)
- На уровне юнита – в файле юнита (секция `[Service]`)
- Просмотр изменений по юнитам: `systemd-delta --type=extended`

Ограничения

```
1 man systemd.exec
2
3 [Service]
4 LimitAS=100M
5 LimitFSIZE=20M
6
7 man systemd.resource-control
8
9 [Service]
10 MemoryLimit=2M
```



Примеры ограничений

Directive	ulimit equivalent	Unit
LimitCPU=	ulimit -t	Seconds
LimitFSIZE=	ulimit -f	Bytes
LimitRSS=	ulimit -m	Bytes
LimitNOFILE=	ulimit -n	Number of file descriptors

Пример создания unit-файла

```
sudo touch /etc/systemd/system/test.service
sudo chmod 664 /etc/systemd/system/test.service
sudo nano /etc/systemd/system/test.service
###
[Unit]
Description=Test service
[Service]
ExecStart=ping ya.ru
[Install]
WantedBy=multi-user.target
###
sudo systemctl daemon-reload
sudo systemctl start test
sudo systemctl status test
```


Вопросы?



Ставим "+",
если вопросы есть



Ставим "-",
если вопросов нет



Практика

Домашнее задание

1. Написать service, который будет раз в 30 секунд мониторить лог на предмет наличия ключевого слова (файл лога и ключевое слово должны задаваться в /etc/sysconfig или в /etc/default).
2. Установить spawn-fcgi и переписать init-скрипт на unit-файл (имя service должно называться так же: spawn-fcgi).
3. Дополнить unit-файл httpd (он же apache2) возможностью запустить несколько инстансов сервера с разными конфигурационными файлами.

В чат ДЗ отправьте ссылку на ваш git-репозиторий.



Сроки выполнения: указаны в личном кабинете



Что мы изучили?

Подведем итоги

1. Системы инициализации System V и systemd
2. Разобрали структуру подсистемы systemd
3. Рассмотрели команды управления юнитами и журналами
4. Изучили механизмы ограничения процессов в systemd



Список материалов для изучения

1. <https://freedesktop.org/wiki/Software/systemd/>
2. <https://www.freedesktop.org/software/systemd/man/systemd.exec.html>
3. <https://www.freedesktop.org/software/systemd/man/index.html>

Вопросы?



Ставим "+",
если вопросы есть



Ставим "-",
если вопросов нет



Рефлексия

Цели вебинара

Проверка достижения целей

1. Понимать различие систем инициализации
2. Использовать основные утилиты systemd
3. Изучить состав и синтаксис systemd unit



Рефлексия



Что было самым полезным на занятии?



Как будете применять на практике то, что узнали на вебинаре?

**Заполните, пожалуйста,
опрос о занятии
по ссылке в чате**

Спасибо за внимание!

Приходите на следующие вебинары



Лавлинский Николай

Технический директор “Метод Лаб”

<https://www.methodlab.ru/>

<https://www.youtube.com/c/NickLavlinsky>

<https://vk.com/nick.lavlinsky>

