OTUS

Загрузка системы

otus.ru

О себе

Никифоров Александр

Больше 10 лет работы с GNU/Linux.

Администратор инфраструктуры “облака” в Селектел.

Область интересов - автоматизация abaremetal deployment.

в сети: @burlunder

Правила вебинара

Активно
участвуем

Off-topic обсуждаем
в Slack #linux-2022-12
или #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

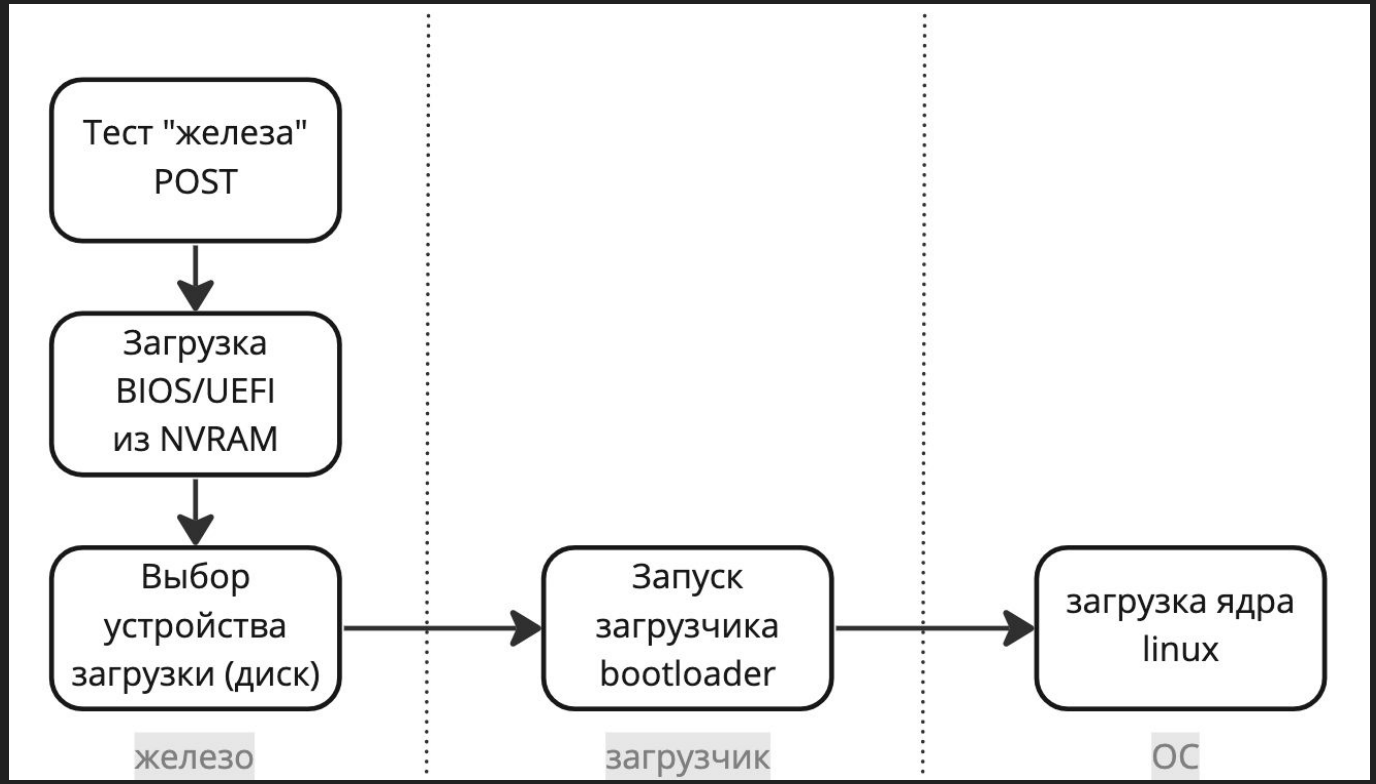
Документ

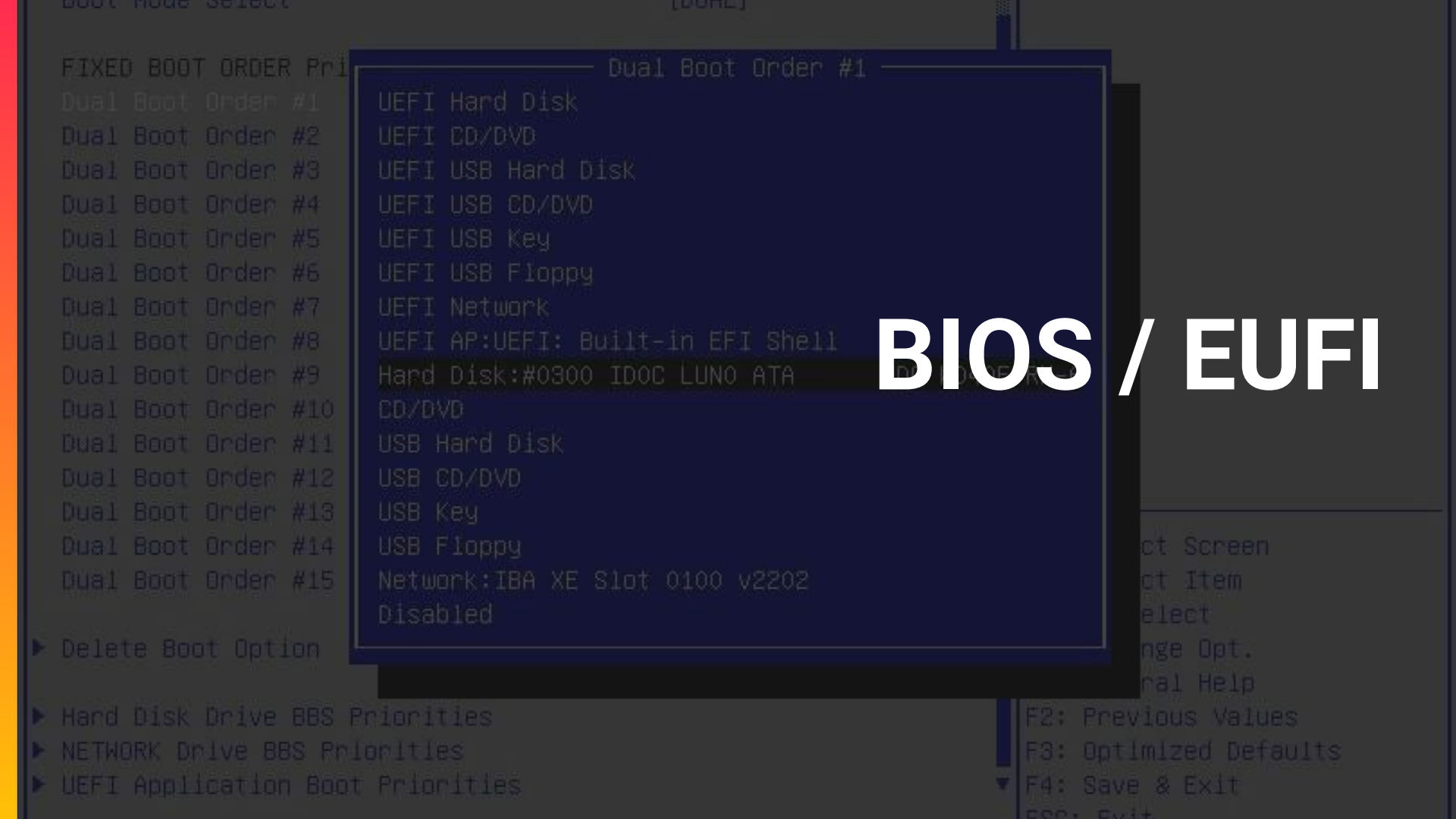
Ответьте себе или
задайте вопрос

Маршрут вебинара

- Процесс загрузки
- BIOS/MBR vs UEFI/GPT
- GRUB
- параметры загрузки cmdline
- initrd
- Пример добавления модуля initrd

Процесс загрузки


Тест "железа"
POST
Загрузка
BIOS/UEFI
из NVRAM
Выбор
устройства
загрузки (диск)
Запуск
загрузчика
bootloader
загрузка ядра
linux
железо
загрузчик
ОС


Dual Boot Order #1
UEFI Hard Disk
UEFI CD/DVD
UEFI USB Hard Disk
UEFI USB CD/DVD
UEFI USB Key
UEFI USB Floppy
UEFI Network
UEFI AP:UEFI: Built-in EFI Shell
Hard Disk:#0300 IDOC LUNO ATA
CD/DVD
USB Hard Disk
USB CD/DVD
USB Key
USB Floppy
Network:IBA XE Slot 0100 v2202
Disabled
FIXED BOOT ORDER Pri
Dual Boot Order #1
Dual Boot Order #2
Dual Boot Order #3
Dual Boot Order #4
Dual Boot Order #5
Dual Boot Order #6
Dual Boot Order #7
Dual Boot Order #8
Dual Boot Order #9
Dual Boot Order #10
Dual Boot Order #11
Dual Boot Order #12
Dual Boot Order #13
Dual Boot Order #14
Dual Boot Order #15
Delete Boot Option
Hard Disk Drive BBS Priorities
NETWORK Drive BBS Priorities
UEFI Application Boot Priorities
F2: Previous Values
F3: Optimized Defaults
F4: Save & Exit

BIOS / EUFI

BIOS

Впервые появилась на CP/M ос в 1975

- BIOS (Basic Input/Output System) - базовая система ввода/вывода
- прошивка, обеспечивающая runtime для операционной системы
- инициализирует оборудование и выполняет self-check (POST)
- ищет первичный загрузчик на MBR (master boot record), ограниченный 512 Kb

UEFI

Придумали в IBM для своей же архитектуры Power в конце 80-х. По сути та та же маленькая ОС со своей спецификацией

- стартует в защищённом режиме
- знает что такое ФС и GPT, может сразу грузиться с дисков
- может грузиться с дисков > 2TB (GPT)
- модульная архитектура, можно использовать свои драйвера и приложения
- встроенный менеджер загрузки

ESP (EFI system partition)

- при загрузке система ищет специальный раздел с меткой ESP (EF00)
- требует разметки диска GPT
- файловая система FAT
- спецификация UEFI определяет структуру файлов на ESP
  EFI\BOOT\BOOTX64.EFI
  EFI\BOOT\SHIM64.EFI

UEFI Secure Boot

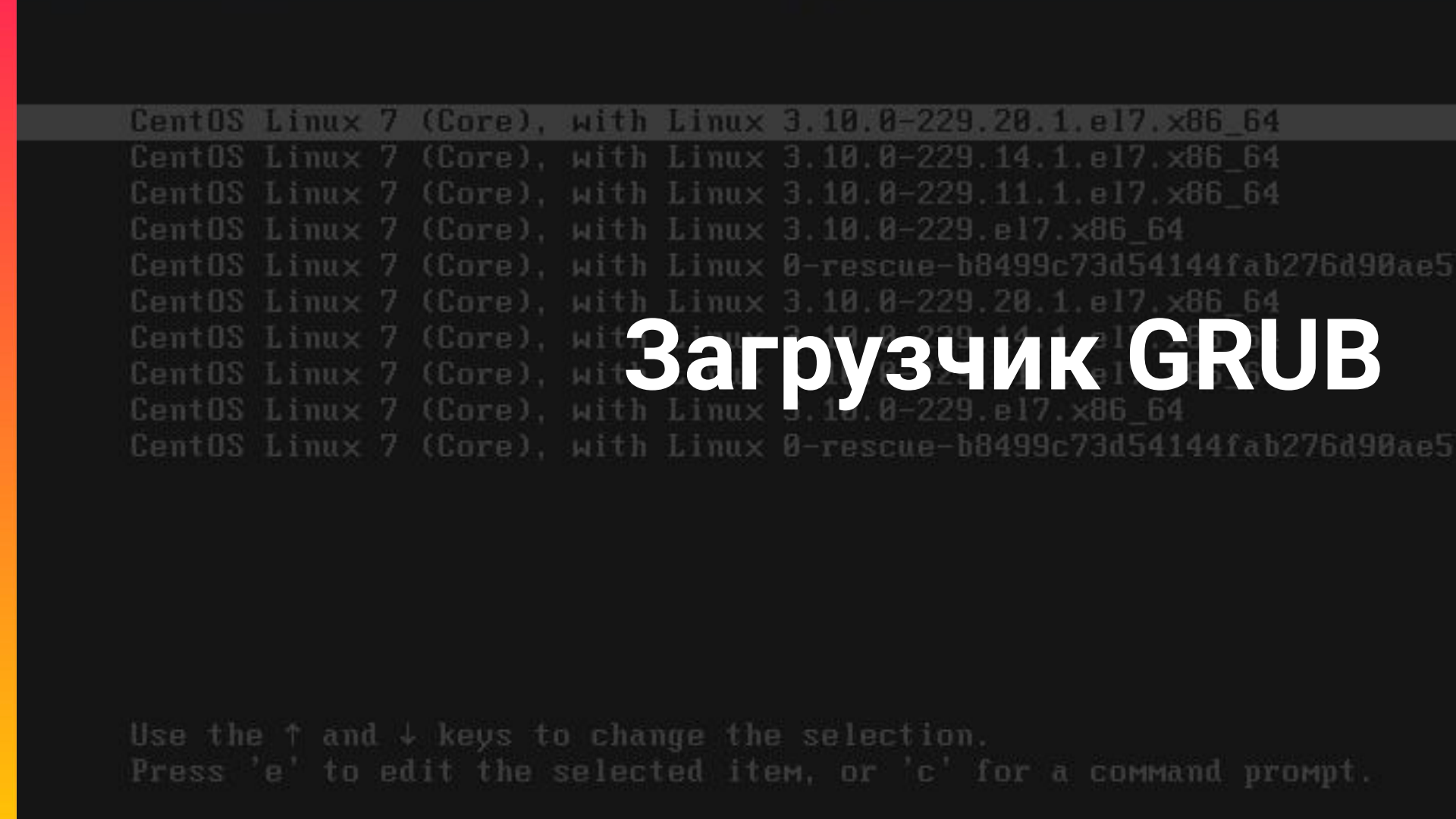
- требует подписанных особым ключом загрузчика и ядра ОС
- ключ как правило идёт вшитым в EFI, вместе с устройством от производителя
  чаще всего от Microsoft
- Меню UEFI позволяет подгружать собственные сертификаты

UEFI Secure Boot

- требует подписанных особым ключом загрузчика и ядра ОС
- ключ как правило идёт вшитым в EFI, вместе с устройством от производителя
  чаще всего от Microsoft
- Меню UEFI позволяет подгружать собственные сертификаты

Призван защищать от буткитов

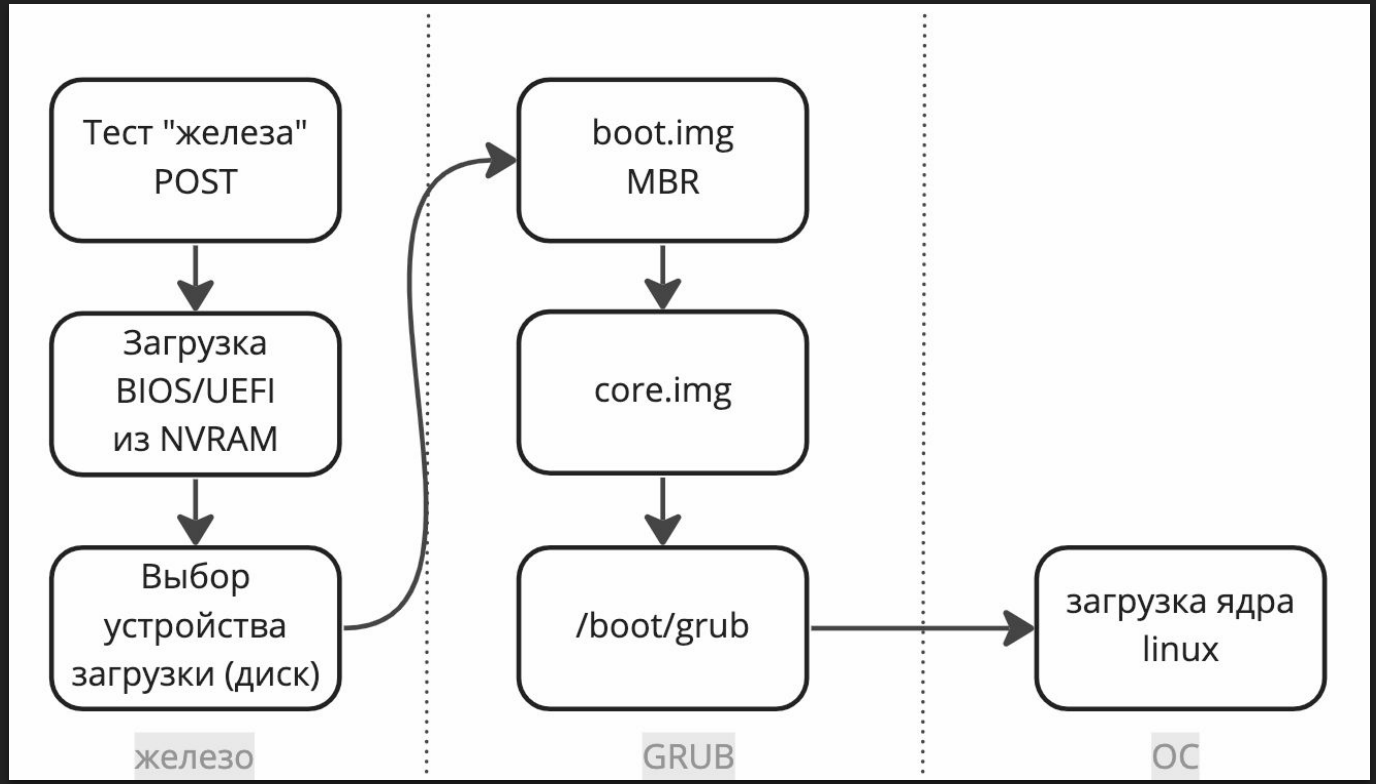
Технология защищает от выполнения неподписанного кода не только на этапе загрузки, но и на этапе выполнени ОС, например, как в Windows, так и в Linux проверяются подписи драйверов/модулей ядра. Таким образом, вредоносный код в режиме ядра выполнить будет нельзя.
Но это справедливо только, если нет физического доступа к компьютеру, т.к., в большинстве случаев, при физическом доступе ключи можно заменить на свои.



Загрузчик GRUB

GRUB

- GNU
- Актуальный GRUB2 (ранее GRUB Legacy)
- Эталонная реализация загрузчика по спецификации Multiboot
- Может передавать управление другому загрузчику (multichain booting)
- Модульная архитектура, за счёт чего работает с любым оборудованием (lvm, raid, luks)
- Интерактивная консоль, можно передать ядру произвольные параметры загрузки (rescue режим)


Тест "железа"
POST
Загрузка
BIOS/UEFI
из NVRAM
Выбор
устройства
загрузки (диск)
boot.img
MBR
core.img
/boot/grub
загрузка ядра
linux
железо
GRUB
ОС

Стадии загрузки (MBR)

- boot.img
  располагается в области MBR, ограничен 512 байт

- core.img
  - между MBR и 1-м разделом диска, 32 Kb *
  - на файловой системе

- /boot/grub
  - все остальные модули
  - интерактивная консоль

Файлы в /boot

- **/boot** - по LFH содержит всё для процесса загрузки до пользовательского режима (user-mode)
- **/boot/grub** - конфигурационные файлы grub
- **/boot/initramfs-*** - образ initrd (init ramdisk)
- **/boot/vmlinuz-*** - ядро Linux
- **/boot/System.map-*** - экспортированные функции ядра. Требуется для компоновки ядра при его сборке

Файлы в /boot/grub

- **/boot/grub/device.map**
  список дисков/разделов в формате GRUB
- **/boot/grub/grub.cfg**
  основной конфигурационный файл при загрузке GRUB, генерируется автоматом
- **/boot/grub/grubenv**
  служит для хранения промежуточных данных между перезагрузками

Изменение параметров загрузки

- **/etc/default/grub**
  переменные для генерации **grub.cfg**
- **/etc/grub.d/40_custom**
  секция скрипта для кастомных пунктов меню загрузки ядер *(добавляется как есть в grub.cfg)*

```
$ grub-mkconfig -o /boot/grub/grub.cfg
```

/etc/grub/default

GRUB_CMDLINE_LINUX - параметры ядра, передаваемые в cmdline

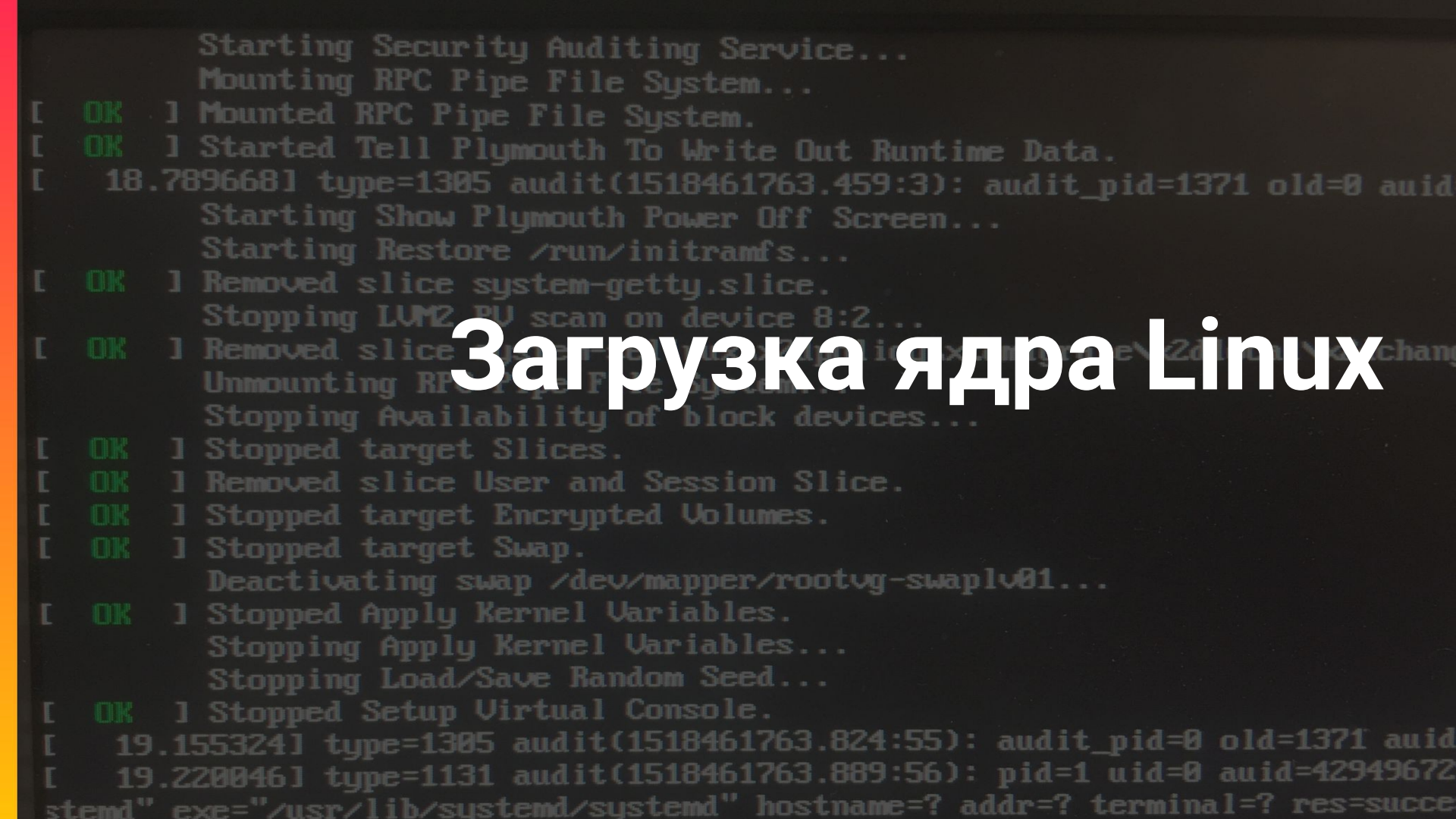
GRUB_DEFAULT - пункт меню по умолчанию

GRUB_TIMEOUT - время до автовыбора (в сек.)

GRUB_SAVEDEFAULT - автосохранение в grubenv последнего выбора

/etc/grub.d/40_custom

```
#!/bin/sh
exec tail -n +3 $
# This file provides an easy way to add custom menu entries. Simp
# menu entries you want to add after this comment. Be careful not
# the 'exec tail' line above.
menuentry ‘OTUS Kernel’ {
    insmod gzio
    insmod part_msdos
    set root=’(hd1,msdos1)’
    linux /otus_kernel root=/dev/sdb1 ro quiet
    initrd /initrd_otus_kernel.img
}
```

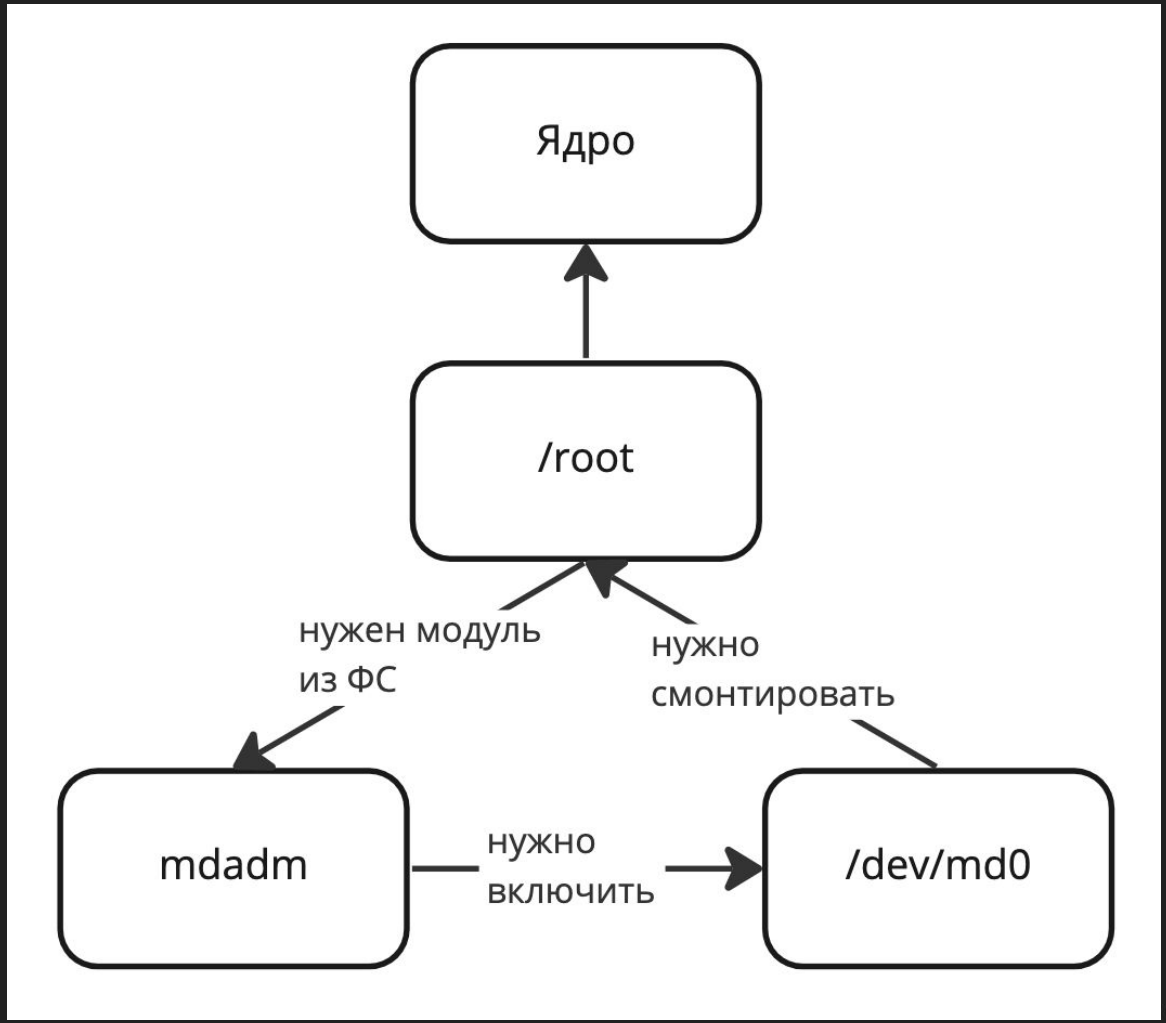


Загрузка ядра Linux

Ядро и /root (проблема курицы и яйца)

На примере RAID

- ядру требуется корневая файловая система / (root)
- /root нужно смонтировать с диска /dev/md0
- для включения /dev/md0 нужен модуль mdadm
- модуль mdadm находится на файловой системе в /lib/modules


Ядро
/root
нужен модуль
из ФС
нужно
смонтировать
mdadm
нужно
включить
/dev/md0

initrd (initramdisk, initramfs)

Минимальный образ ФС

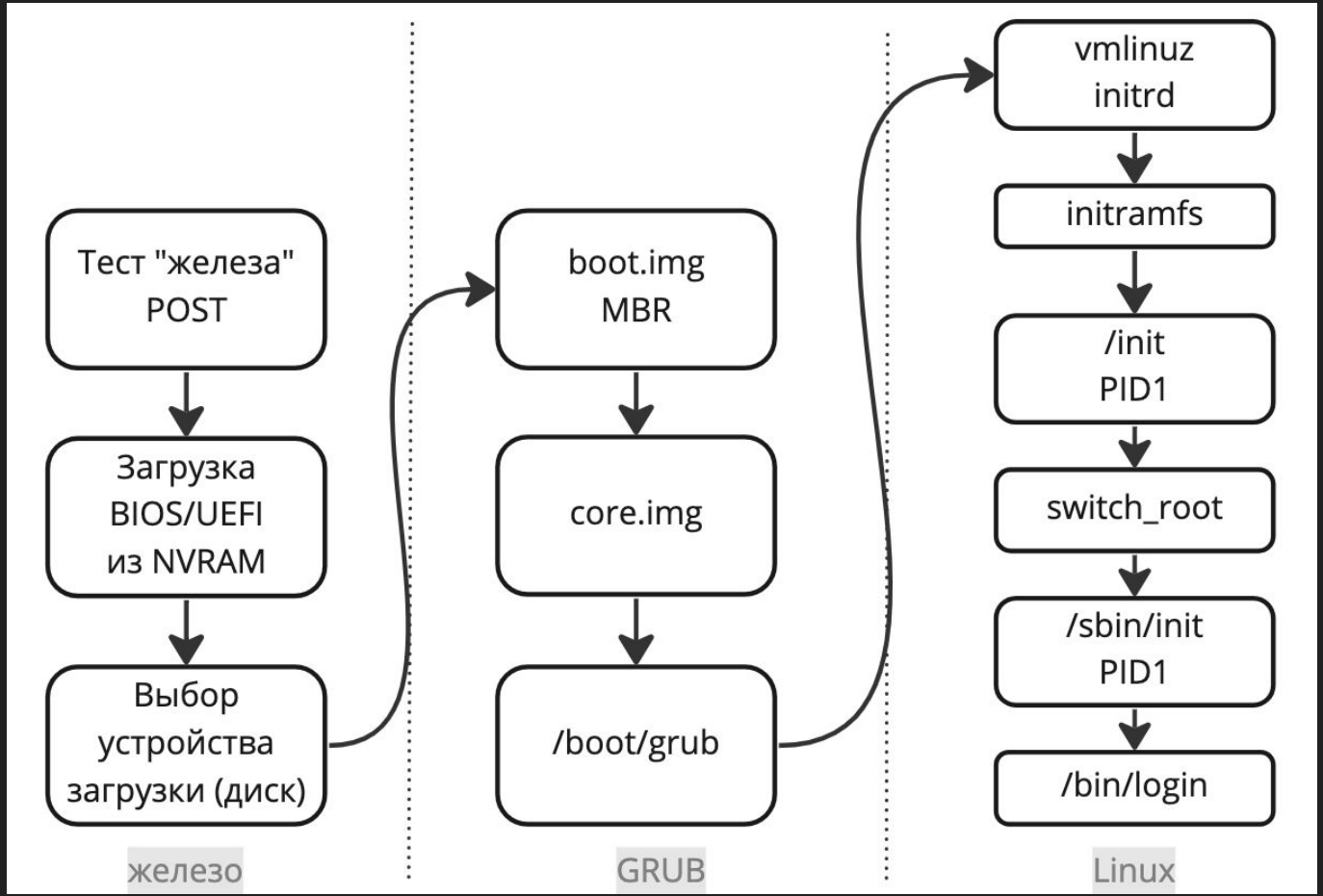
- архив cpio, который распаковывается в rootfs
- содержит модули, необходимые для загрузки
- скрипт **/init** нужен для поиска устройства, которое будет дальше root

initrd (initramdisk, initramfs)

Минимальный образ ФС

- архив cpio, который распаковывается в rootfs
- содержит модули, необходимые для загрузки
- скрипт **/init** нужен для поиска устройства, которое будет дальше root

1. cpio распаковывается в **initramfs**
2. **/init**
3. ищется устройство **real root**
4. **switch_root** (pivot_root)
5. **/sbin/init**


Тест "железа"
POST
Загрузка
BIOS/UEFI
из NVRAM
Выбор
устройства
загрузки (диск)
железо
boot.img
MBR
core.img
/boot/grub
GRUB
vmlinuz
initrd
initramfs
/init
PID1
switch_root
/sbin/init
PID1
/bin/login
Linux

Использование initrd

список модулей dracut

```
lsinitrd -m /boot/initramfs-$(uname -r).img
```

разборка

```
zcat /boot/initrd-$(uname -r).img | cpio -i
```

сборка

```
find. -print0 |
cpio -o --null --format=newc |
gzip -q -9 > /boot/initrd-$(uname -r).img
```

Dracut

https://www.kernel.org/pub/linux/utils/boot/dracut/dracut.html

Используется в CentOS для управления **initrd**

- модификация
- просмотр содержимого initrd
- добавление своих скриптов на разные этапы загрузки

Dracut. модули (функции-хуки)

- **cmdline** - самое начало загрузки initrd
- **pre-udev** - перед запуском udev-подсистемы
- **pre-trigger** - в процессе запуска udev, возможность ним взаимодействовать
- **pre-mount** - перед монтированием файловых систем
- **mount** - смонтировать root-filesystem
- **pre-pivot** - после монтирования, перед pivot_root
- **cleanup** - перед pivot_root, “подчистка за собой”

Dracut. пример модуля

```
#!/bin/bash

check() {
    return 0
}
depends() {
    return 0
}
install() {
    inst_hook cleanup 00 "${moddir}/test.sh"
}
```

Dracut. пример модуля

```
#!/bin/bash
exec 0<>/dev/console 1<>/dev/console 2<>/dev/console
cat <<'msgend'
 ____________________
< I'm dracut module >
 --------------------
    \
     \
        .--.
       |o_o |
       |:_/ |
      //   \ \
     (|     | )
    /'\_   _/`\
    \___)=(___/
msgend
sleep 10
echo " continuing....
```

Заполните, пожалуйста, опрос о занятии по ссылке в чате https://otus.ru/polls/59398/

Спасибо за внимание!

Приходите на следующие вебинары

Никифоров Александр

системный администратор

@burlunder