

[Типы загрузки](#)

[BIOS/MBR](#)

[UEFI/GPT](#)

[Пример MBR to EFI миграции](#)

[Boot Loader Specification](#)

[Файлы которые используются при загрузке](#)

[Загрузчик GRUB2](#)

[Конфигурация загрузчика GRUB2](#)

[Поменяем картинку при загрузке](#)

[Утилита для редактирования конфигурации загрузки](#)

[Initial ramdisk](#)

[Сбрасываем пароль root](#)

[Защита паролем меню GRUB](#)

[Меняем графическую заставку](#)

[Добавляем модуль в initramfs](#)

[Загрузка без загрузчика EFISTUB Kernel](#)

[Документация](#)

Типы загрузки

BIOS/MBR

the program is written into read-only, permanent memory and is always available for use
Применяется для совместимости со старым оборудованием. Встречается в системах виртуализации.

- the processor looks at the end of system memory for the BIOS program, and runs it.
- the BIOS tests the system, looks for and checks peripherals, and then locates a valid device with which to boot the system.
- the order of the drives searched while booting is controlled with a setting in the BIOS
- BIOS loads into memory whatever program is residing in the first sector of this device, called the Master Boot Record (MBR). The MBR is only 512 bytes in size and contains machine code instructions for booting the machine, called a boot loader.

- BIOS finds and loads the boot loader program into memory, it yields control of the boot process to it.
- **first-stage boot loader** is a small machine code binary on the MBR. Its job is to locate the **second stage boot loader** (GRUB) and load the first part of it into memory

https://datacadamia.com/data_storage/mbr

initramfs-4.18.0-193.14.2.el8_2.x86_64.img

UEFI/GPT

EFI - Extensible Firmware Interface

UEFI

контролирует загрузочный процесс

умеет монтировать разделы и читать файловые системы

ESP (EFI system partition)

системный раздел на котором содержатся приложения скомпилированные под архитектуру EFI

это могут быть загрузчики, утилиты или ядро ОС

ESP раздел EFI firmware может быть как GUID так и MBR. Разделы должны содержать идентификатор.

EFI system partition C12A7328-F81F-11D2-BA4B-00A0C93EC93B (уточнить???)

MBR partition-table scheme is [0xEF](#)

Например:

```
parted /dev/sda -l
file /boot/efi/EFI/efistub/bootx64.efi
```

Пример MBR to EFI миграции

1 Создаем раздел с типом ESP

```
lsblk
```

```
fdisk -l /dev/sdb
```

```
Disk /dev/sdb: 1 GiB, 1073741824 bytes, 2097152 sectors
```

```
Units: sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: CB31299F-0910-C84B-984B-20D2C3D36E9F
```

Device	Start	End	Sectors	Size	Type
/dev/sdb1	2048	1026047	1024000	500M	EFI System

2. Форматируем

```
yum -y install dosfstools
mkfs.fat -F32 /dev/sdb1
Добавить /boot в /etc/fstab
blkid
/etc/fstab
UUID=21B7-C118 /boot vfat defaults 0 2

mount /boot
```

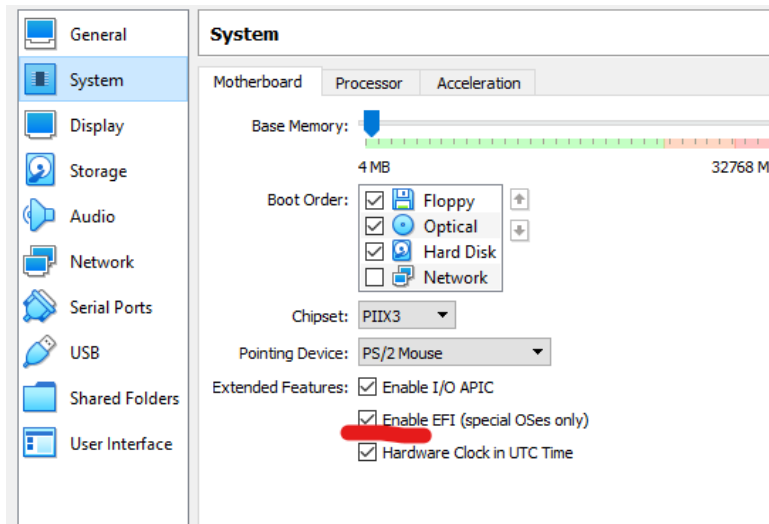
3. Установить grub на ESP раздел

```
yum -y install efi-filesystem grub2-efi-x64-modules efibootmgr
yum -y update kernel-core
```

```
grub2-install --target=x86_64-efi --efi-directory=/boot/
--bootloader-id=grub2
```

- `sudo dnf reinstall shim-* grub2-*` #
https://bugzilla.redhat.com/show_bug.cgi?id=1917213
`shutdown -h now`

```
# Поменять тип загрузки на EFI
```



4. Создадим конфиг

```
grub2-mkconfig -o /etc/grub2.cfg
```

5. Изучаем команды в UEFI shell

```
help
help map
map -r
map -u
FS0:
ls
cat
cd
bcfg
```

Добавить загрузочную запись из EFI shell

```
bcfg boot add 0 fs1:\EFI\grub\grubx64.efi grub
```

Вывести список загрузочных записей (-v verbose, -b paged)

```
bcfg boot dump -v -b
```

6. Добавить загрузочную запись из Linux

```
efibootmgr -c -d /dev/sdb1 --label "grubmenu" --loader
'\EFI\grub2\grubx64.efi'
```

Boot Loader Specification

описывает схему которая позволяет разным операционным системам (дистрибутивам) управлять конфигурационной директорией загрузчика. Включая записи в загрузочном меню. Предлагается использовать единый формат конфигурационного файла для всех реализаций загрузчика, операционных систем, утилит пользователя.

Содержимое директории \$BOOT

- \$BOOT/loader/ is the directory containing all files needed for Type #1 entries
- \$BOOT/loader/entries/ is the directory containing the drop-in snippets. This directory contains one .conf file for each boot menu item.

```
find /boot/loader/  
/boot/loader/  
/boot/loader/entries  
/boot/loader/entries/ac4fce7ddccf4acb8ed1057791830358-4.18.0-193.6.3.el8_2.x86_64.conf
```

Имя файла уникальное. Предлагается использовать файл /etc/machine-id (/etc/machine-id or the D-Bus machine ID for OSes that lack /etc/machine-id), the kernel version (as returned by uname -r) and an OS identifier (The ID field of /etc/os-release).

```
cat /etc/machine-id  
852614d49f89479eba3849aece013419
```

Файл содержит поля title version linux initrd options

```
title CentOS Linux (4.18.0-193.6.3.el8_2.x86_64) 8 (Core)  
version 4.18.0-193.6.3.el8_2.x86_64  
linux /boot/vmlinuz-4.18.0-193.6.3.el8_2.x86_64  
initrd /boot/initramfs-4.18.0-193.6.3.el8_2.x86_64.img $tuned_initrd  
options
```

```
kernel-install add $(uname -r) /lib/modules/$(uname -r)/vmlinuz  
man 8 kernel-install  
https://access.redhat.com/solutions/3766391
```

Файлы которые используются при загрузки

1. Модули grub
ls /boot/grub2/i386-pc/*.mod
2. Модули ядра find /lib/modules/ или так найти путь для модуля
modinfo libata 2>&1 | grep filename

3. Модули для initrd (initramfs)
dracut --list-modules либо здесь
ls /usr/lib/dracut/modules.d

Загрузчик GRUB2

Конфигурация загрузчика GRUB2

Настройки содержатся здесь:

```
/etc/default/grub  
/etc/grub.d/40_custom
```

Команда обновления конфига

```
/usr/sbin/grub2-mkconfig
```

Сгенерированный конфиг

Имя конфига зависит от типа загрузки

EFI

```
grub2-mkconfig -o /boot/efi/EFI/centos/grub.cfg
```

или BIOS

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

Пример: отключаем ipv6 (CentOS7)

```
vim /etc/default/grub
```

```
# cat /etc/default/grub  
GRUB_TERMINAL_OUTPUT="console"  
GRUB_CMDLINE_LINUX=" rhgb quiet ipv6.disable=1 "  
GRUB_DISABLE_RECOVERY="true"
```

```
grub2-mkconfig -o /boot/grub2/grub.cfg  
cat /boot/grub2/grub.cfg
```

Пример: добавляем пользовательское меню

```
/etc/grub.d/  
10_linux - поиск ядра на разделе по умолчанию  
40_custom - пользовательское меню
```

41_custom - пользовательское меню из внешнего файла

```
grub2-mkconfig -o /boot/efi/EFI/centos/grub.cfg
```

<https://wiki.centos.org/HowTos/Grub2>

```
40_custom
menuentry 'OTUS Kernel' {
    set root='(hd1,msdos1)'
    linux /otus_kernel root=/dev/sdb1 ro quiet
    initrd /initrd_otus_kernel.img
}
40_custom
menuentry "May be run by any user" --unrestricted {
    set root=(hd0,1)
    linux /vmlinuz
}
```

Поменяем картинку при загрузке

```
vagrant ssh-config > config
scp -F config ~/Pictures/2017-12-13\ 19.35.32.jpg
otus_efi:./image.jpg
```

1. Установим пакет grub2-efi-x64-modules

2. Создаем директорию

```
mkdir /boot/efi/EFI/centos/x86_64-efi/
```

3. Проверим что модуль gfxterm_background.mod установлен

```
find /boot/ | grep gfxterm
/boot/grub2/x86_64-efi/gfxterm_background.mod
```

4. Добавим загрузку модуля в конфигурационный файл

```
cat /etc/default/grub
GRUB_TIMEOUT=4
#GRUB_TERMINAL_OUTPUT="console"
GRUB_BACKGROUND="/boot/grub2/image.jpg"
GRUB_PRELOAD_MODULES="gfxterm_background"
```

5. Rebuild grub2 config

```
grub2-mkconfig -o /boot/efi/EFI/centos/grub.cfg
```

6. Проверить пути и модули "insmod gfxterm_background" (можно проверить и в генераторах /etc/grub.d/00_header)

Утилита для редактирования конфигурации загрузки

```
grubby --help
grubby --default-kernel
grubby --default-title
grubby --default-index
grubby --info=ALL
```

```
grubby --remove-args="rhgb quiet" --update-kernel
/boot/vmlinuz-4.18.0-147.5.1.el8_1.x86_64
grubby --info /boot/vmlinuz-4.18.0-147.5.1.el8_1.x86_64
grubby --update-kernel=/boot/vmlinuz-4.18.0-147.5.1.el8_1.x86_64
--args="rhgb quiet"
```

```
grubby --copy-default
--add-kernel=/boot/vmlinuz-4.18.0-147.5.1.el8_1.x86_64 --title="Otus
Menu Kernel" --initrd="/boot/initramfs-4.18.0-147.5.1.el8_1.x86_64"
ls /boot/loader/entries/
```

Дополнительная информация по загрузчикам

https://systemd.io/BOOT_LOADER_SPECIFICATION/

Boot loader specification

Initial ramdisk

Сбрасываем пароль root

- Дописываем в конец строки rd.break (опционально убираем quiet)
- Загружаемся нажимаем Ctrl - x

- Смотрим список смонтированных файловых систем. Ищем /sysroot
`mount`
- Перемонтируем в режиме rw /sysroot.
`mount -o remount,rw /sysroot`
- Проверяем командой `mount` что /sysroot примонтирован в rw режиме (чтение-запись)
`mount`
- Заходим в образ системы
`chroot /sysroot`
- Изменяем пароль пользователя командой
`passwd`
- Т.к. разрешен selinux - создаем файл .autorelabel. (по умолчанию разрешен)
Выполняем команду `touch /.autorelabel`
- Команда `exit` выходим из chroot.
- Перемонтируем в ro (read only)
`mount -o remount,ro /sysroot`

Защита паролем меню GRUB

```
grub2-setpassword # пароль в открытом виде поэтому используем
grub2-mkpasswd-pbkdf2
cat /etc/grub.d/01_users
cat /boot/grub2/user.cfg
```

Пункт меню доступный любому пользователю

```
40_custom.cfg
menuentry "May be run by any user" --unrestricted {
    set root=(hd0,1)
    linux /vmlinuz
}
```

<https://www.thegeekdiary.com/centos-rhel-7-how-to-password-protect-grub2-menu-entries/>

Меняем графическую заставку

```
yum install plymouth\*
plymouth-set-default-theme spinfinity
plymouth-set-default-theme --list
dracut -f
```

Добавляем модуль в initramfs

```
mkdir /usr/lib/dracut/modules.d/01test
cd /usr/lib/dracut/modules.d/01test
cat >module-setup.sh<<EOF
```

```
#-----
#!/bin/bash

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

```
chmod +x module-setup.sh
vi test.sh
test.sh
```

```
#-----
#!/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....
```

```
#-----
```

```
chmod +x test.sh
mkinitrd -f -v /boot/initramfs-$(uname -r).img $(uname -r)
lsinitrd -m /boot/initramfs-$(uname -r).img | grep test
```

Загрузка без загрузчика EFISTUB Kernel

Проверяем что ядро собрано с EFISTUB

```
zgrep CONFIG_EFI_STUB /boot/config-$(uname -r)
CONFIG_EFI_STUB=y
```

(минуя загрузчик)

```
EFIPATH=/boot/efi/EFI/efistub
mkdir $EFIPATH
cp /boot/vmlinuz-$(uname -r) $EFIPATH/bootx64.efi
cp /boot/initramfs-$(uname -r).img $EFIPATH/initramfs.img
```

Смотрим параметры загрузки

```
cat /proc/cmdline
```

Добавляем новую загрузочную запись

```
efibootmgr -c -d /dev/sda1 --label "Linux" --loader
'\efi\efistub\bootx64.efi' -u "root=/dev/mapper/cl-root ro
crashkernel=auto resume=/dev/mapper/cl-swap rd.lvm.lv=cl/root
rd.lvm.lv=cl/swap initrd=\\EFI\\efistub\\initramfs.img"
```

Меняем порядок загрузки

```
efibootmgr
efibootmgr --bootorder 0004,0000,0001,0002,0003,0005
```

Дополнительная информация по EFISTUB

<https://bmccarthy.net/2017/05/15/booting-efi-stub-fedora-25/>
<https://habr.com/ru/post/197438/>

Документация

загрузчики

<https://wiki.ubuntu.com/EFIBootLoaders>

https://bugzilla.redhat.com/show_bug.cgi?id=1525228

EFI Secure Boot

<https://habr.com/ru/post/308032/>

старая статья, но более менее точная

<https://habr.com/ru/post/113350/>

Пример миграции BIOS to EFI

<https://oded.blog/2017/11/13/fedora-bios-to-uefi/>

Пример установки/добавления EFI

https://systemd.io/BOOT_LOADER_SPECIFICATION/

<https://www.rodsbooks.com/efi-bootloaders/installation.html>

Про [BootLoaderSpec/](#) grub2-switch-to-blscfg

<https://www.freedesktop.org/wiki/Specifications/BootLoaderSpec/>

<https://tiantiankan.me/a/5d582695a7c6c7fc974bbc4b>

<https://linuxconfig.org/how-to-remove-old-unused-kernels-on-centos-linux>