

## [Типы загрузки](#)

[BIOS](#)

[UEFI](#)

## [Загрузчик GRUB2](#)

[Конфигурация загрузчика GRUB2](#)

[Поменяем картинку при загрузке](#)

[Утилита для редактирования конфигурации загрузки](#)

## [Initial ramdisk](#)

[Сбрасываем пароль root](#)

[Меняем графическую заставку](#)

[Добавляем модуль в initramfs](#)

## [Загрузка без загрузчика EFISTUB Kernel](#)

## [Документация](#)

# Типы загрузки

## BIOS

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

# UEFI

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
```

Команды в UEFI shell

```
help
help map
map -r
map -u
FS0:
ls
```

Добавить загрузочную запись

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

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

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

# Загрузчик GRUB2

## Конфигурация загрузчика GRUB2

Команда обновления конфига

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

Сгенерированный конфиг

Имя зависит от типа загрузки EFI или BIOS

```
/boot/grub2/grub.cfg
```

```
/boot/efi/EFI/redhat/grub.cfg
```

Настройки по умолчанию

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

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
```

## Поменяем картинку при загрузке

1. Установим пакет grub2-efi-x64-modules

2. Создаем директорию

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

3. Добавим модуль /usr/lib/grub/x86\_64-efi/gfxterm\_background.mod в директорию

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

```
cp /usr/lib/grub/x86_64-efi/gfxterm_background.mod
```

4. Добавим загрузку модуля в конфигурационный файл

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

```
GRUB_BACKGROUND=my_photo.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/](https://systemd.io/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

```
grub2-setpassword
```

## 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/>

### Загрузка по сети

PXE, TFTP, DHCP

iPXE - DHCP, вместо TFTP можно использовать ( HTTPD, iSCSI и другие )

[http://ftp.byfly.by/pub/CentOS/8.1.1911/BaseOS/x86\\_64/os/images/pxeboot/](http://ftp.byfly.by/pub/CentOS/8.1.1911/BaseOS/x86_64/os/images/pxeboot/)

Можно

cobbler

## Документация

### загрузчики

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

[https://bugzilla.redhat.com/show\\_bug.cgi?id=1525228](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://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>