

# Содержание

- Глава 5: Загрузка ядра Linux** ..... 3
  - 5.1 Сообщения при загрузке** ..... 3
  - 5.2 Параметры инициализации и загрузки ядра** ..... 3
  - 5.3 Параметры ядра** ..... 3
  - 5.4 Загрузчики** ..... 4
  - 5.5 Введение в загрузчик GRUB** ..... 4
    - 5.5.1 Изучение устройств и разделов с помощью командной строки GRUB ..... 4
      - Схема адресации GRUB ..... 4
      - Основные команды ..... 5
    - 5.5.2 Конфигурация GRUB ..... 5
      - Структура grub.cfg ..... 6
      - Регенерация конфигурации ..... 6
      - Пользовательская конфигурация ..... 6
    - 5.5.3 Установка GRUB ..... 7
      - Установка GRUB в систему ..... 7
      - Установка GRUB на внешнее устройство (MBR) ..... 7
      - Установка GRUB с помощью UEFI ..... 7
  - 5.6 Проблемы безопасной загрузки UEFI** ..... 8
  - 5.7 Метод цепной загрузки других операционных систем** ..... 8
  - 5.8 Детали работы загрузчика** ..... 9
    - 5.8.1 Загрузчик MBR ..... 9
    - 5.8.2 Загрузчик UEFI ..... 9
    - 5.8.3 Как работает GRUB ..... 10



# Глава 5: Загрузка ядра Linux

Упрощенно процесс загрузки выглядит следующим образом:

1. BIOS или загрузочная программа компьютера загружается и запускает загрузчик.
2. Загрузчик находит образ ядра на диске, загружает его в память и запускает.
3. Ядро инициализирует устройства и их драйверы.
4. Ядро монтирует корневую файловую систему.
5. Ядро запускает программу под названием `init` с идентификатором процесса 1. Эта точка является началом пользовательского пространства.
6. Программа `init` приводит в действие остальные системные процессы.
7. В какой-то момент запускается процесс, позволяющий пользователю войти в систему, обычно в конце или ближе к концу процесса загрузки.

## 5.1 Сообщения при загрузке

При запуске `journalctl -k` отображаются сообщения текущей загрузки, а при `journalctl -b` - можно просмотреть предыдущие загрузки.

Если у вас нет системы `systemd`, можете поискать файл журнала `/var/log/kern.log` или выполнить команду `dmesg` для просмотра сообщений в кольцевом буфере ядра.

## 5.2 Параметры инициализации и загрузки ядра

При запуске ядро Linux инициализируется в таком порядке:

1. Проверка процессора.
2. Проверка памяти.
3. Обнаружение шины устройства.
4. Обнаружение устройств.
5. Настройка вспомогательной подсистемы ядра (сеть и т.п.).
6. Монтирование корневой файловой системы.
7. Запуск пользовательского пространства.

## 5.3 Параметры ядра

Вы можете изучить параметры, переданные в активное ядро вашей системы, просмотрев файл `/proc/cmdline`.

Обзор основных параметров загрузки ядра приведен на странице руководства `bootparam(7)`.

Обнаружив незнакомый параметр, ядро Linux сохраняет его. Позже оно передает параметр программе `init` при запуске пользовательского пространства. Например, если вы добавляете параметр `-s` к параметрам ядра, оно передает `init` указание, что та должна запускаться в однопользовательском режиме.

## 5.4 Загрузчики

В начале загрузки, до загрузки ядра и команды `init`, программа загрузчика запускает ядро. Задача загрузчика проста: ему нужно загрузить ядро в память с диска, а затем запустить его с набором определенных параметров.

Ядро находится в сжатом виде в корневом разделе (как правило) или в своем собственном разделе на диске. Таким образом загрузчику нужно получить доступ к диску не имея загруженного ядра и его драйверов диска. Для этого BIOS / UEFI использует LBA - универсальный простой способ доступа к данным с любого диска, но его производительность довольно низка. Это не проблема, поскольку загрузчики часто являются единственными программами, которые должны использовать этот режим для доступа к диску: после запуска ядро получает доступ к собственным высокопроизводительным драйверам.

## 5.5 Введение в загрузчик GRUB

**GRUB** (Grand Unified Boot Loader) - главный унифицированный загрузчик. Рассматриваем GRUB 2.

Основная возможность: навигация по ФС для выбора образа ядра и параметров.

Система	Клавиша
BIOS	Shift (удерживать при загрузке)
UEFI	Esc

В меню:

- **e** - редактировать конфигурацию загрузки
- **c** - открыть командную строку GRUB

Путаница возникает из-за двух разных понятий **root**:

Значение root	Область	Назначение
GRUB root	Только GRUB	ФС, где GRUB ищет ядро и <code>initrd</code>
Kernel root	Параметр ядра	Корневая ФС системы (после загрузки)

**Пример конфигурации:**

1. `set root=(hd0,msdos1)` - GRUB root по устройству
2. `search --set=root --fs-uuid UUID` - GRUB root по UUID
3. `linux /boot/vmlinuz-... root=UUID` - параметр ядра (kernel root)
4. `initrd /boot/initrd.img-...` - начальная ФС ОП

### 5.5.1 Изучение устройств и разделов с помощью командной строки GRUB

**Схема адресации GRUB**

Обозначение	Описание
(hd0)	Первый найденный жесткий диск
(hd0,msdos1)	Первый раздел MBR
(hd0,gpt1)	Первый раздел GPT (UEFI)

## Основные команды

### Список устройств:

```
grub> ls
(hd0) (hd0,msdos1)

grub> ls -l
# Выводит UUID, тип ФС, размер разделов
```

### Навигация по файлам:

```
grub> echo $root
hd0,msdos1

grub> ls ($root)/
# Список файлов в корне раздела

grub> ls ($root)/boot
# Содержимое /boot

grub> set
# Все переменные GRUB
```

### Важные переменные:

- `$prefix` - путь для конфигурации GRUB (обычно `/boot/grub`)
- `$root` - текущий корень GRUB

### Завершение:

- **Esc** - вернуться в меню
- **boot** - загрузить систему с текущей конфигурацией

**История команд:** ↑ / ↓ (навигация), ← / → (редактирование), Ctrl+N/P

## 5.5.2 Конфигурация GRUB

Каталог конфигурации GRUB обычно `/boot/grub` или `/boot/grub2`. Содержит:

- `grub.cfg` - главный файл конфигурации
- `i386-pc/`, `x86_64-efi/` - модули архитектуры (`.mod`)
- Шрифты, информация о локализации

**Не редактируйте grub.cfg напрямую** - используйте grub-mkconfig.

## Структура grub.cfg

Файл состоит из:

Раздел	Содержимое
Инициализация	Определения функций, переменные (\$font), настройки видео
Меню загрузки	Команды menuentry и submenu

### Пример menuentry:

```
menuentry 'Ubuntu' --class ubuntu --class gnu-linux --class os {
  recordfail
  load_video
  insmod part_msdos
  insmod ext2
  set root='hd0,msdos1'
  search --no-floppy --fs-uuid --set=root UUID
  linux /boot/vmlinuz-... root=UUID=... ro quiet splash
  initrd /boot/initrd.img-...
}
```

## Регенерация конфигурации

grub.cfg создается автоматически из скриптов /etc/grub.d/. Каждый файл генерирует часть конфигурации:

```
# Просмотр (не сохраняет)
grub-mkconfig

# Сохранение новой конфигурации
grub-mkconfig -o /boot/grub/grub.cfg
```

**Порядок обработки:** числа в названиях влияют на последовательность (00\_header, 10\_linux, 40\_custom и т.д.).

## Пользовательская конфигурация

Два варианта добавления собственных пунктов меню:

Файл	Способ	Стабильность
/etc/grub.d/40_custom	Редактируемый сценарий	Низкая (обновления переписут)
/boot/grub/custom.cfg	Загружается через 41_custom	Высокая (видна при загрузке)

### Рекомендуемый способ:

```
# Отредактировать
nano /boot/grub/custom.cfg

# Регенерировать
grub-mkconfig -o /boot/grub/grub.cfg

# Сделайте резервную копию старой конфигурации!
cp /boot/grub/grub.cfg /boot/grub/grub.cfg.bak
```

Дополнительные скрипты в /etc/grub.d (напр., Ubuntu добавляет memtest86+).

### 5.5.3 Установка GRUB

Установка GRUB требует указания целевого каталога, текущего устройства диска и (для UEFI) точки монтирования раздела EFI. GRUB - модульная система, поэтому нужно создать версию, способную читать файловую систему для загрузки конфигурации (grub.cfg) и модулей.

#### Установка GRUB в систему

Требуемые параметры:

Параметр	Описание
Целевой каталог GRUB	Обычно /boot/grub (может отличаться при установке на другой диск)
Текущее устройство диска	Необходимо определить целевой диск
Точка монтирования EFI	Для UEFI: обычно /boot/efi (требуется указать текущую точку)

Утилита **grub-install** автоматизирует установку. Например, для установки GRUB на MBR диска /dev/sda:

```
# grub-install /dev/sda
```

**ВНИМАНИЕ:** Неправильная установка нарушит загрузку системы. Создайте резервную копию MBR и подготовьте план аварийной загрузки.

#### Установка GRUB на внешнее устройство (MBR)

Для установки на внешний диск вне текущей системы укажите каталог GRUB вручную. Если целевое устройство /dev/sdc смонтировано на /mnt:

```
# grub-install --boot-directory=/mnt/boot /dev/sdc
```

#### Установка GRUB с помощью UEFI

Установка UEFI требует сохранения конфигурации в NVRAM через **efibootmgr**. Команда grub-

install выполняет это автоматически:

```
# grub-install --efi-directory=efi_dir --bootloader-id=name
```

Где:

- **efi\_dir** - каталог UEFI в текущей системе (обычно /boot/efi/EFI)
- **name** - идентификатор загрузчика

При установке UEFI могут возникнуть проблемы при переносе на другую систему и различия в процедуре для съемных носителей. Одной из главных проблем является функция безопасной загрузки UEFI.

## 5.6 Проблемы безопасной загрузки UEFI

Безопасная загрузка UEFI требует цифровую подпись поставщика для запуска любого загрузчика. Microsoft обязал производителей оборудования использовать эту функцию для систем с Windows 8 и новее. Неподписанный загрузчик будет отклонен встроенным ПО и система не загрузится.

Основные дистрибутивы Linux используют уже подписанные загрузчики (обычно GRUB для UEFI). Между UEFI и GRUB проходит небольшая подписанная часть данных, которую UEFI запускает в первую очередь.

Безопасная загрузка защищает от несанкционированного ПО в ненадежной среде и требуемых системах безопасности. Некоторые дистрибутивы требуют подпись всей последовательности загрузки, включая ядро.

**Недостатки:** особенно актуальны для экспериментов с собственными загрузчиками. Можно отключить в настройках UEFI, но это не сработает для двойной загрузки (например, с Windows).

## 5.7 Метод цепной загрузки других операционных систем

UEFI поддерживает несколько загрузчиков в разделе EFI, но устаревший MBR этого не предусматривает. **Цепная загрузка (chainloading)** - это метод, при котором GRUB загружает и запускает другой загрузчик из определенного раздела вместо загрузки ядра Linux.

Создайте новый пункт в конфигурации GRUB:

```
menuentry "Windows" {
  insmod chain
  insmod ntfs
  set root=(hd0,3)
  chainloader +1
}
```

Параметр **+1** загружает содержимое первого сектора раздела. Для прямой загрузки файла используйте:

```
menuentry "DOS" {
  insmod chain
  insmod fat
  set root=(hd0,3)
  chainloader /io.sys
}
```

## 5.8 Детали работы загрузчика

Загрузчики имеют два основных варианта: **MBR** и **UEFI**, разработанные для устранения недостатков традиционных механизмов загрузки ПК.

### 5.8.1 Загрузчик MBR

MBR (Master Boot Record) содержит 441 байт кода, который BIOS загружает после самотестирования при включении (POST). Этого недостаточно для полного загрузчика, поэтому используется **многоэтапная загрузка**: небольшой код в MBR загружает остальную часть кода загрузчика, размещённую между MBR и первым разделом.

**Проблема с GPT:** эта схема не работает с GPT диском при использовании традиционного BIOS, поскольку информация GPT находится после MBR. **Решение:** создать специальный **загрузочный раздел BIOS** с UUID `21686148-6449-E6F-744E-656564454649` для размещения полного кода загрузчика. Это встречается редко — обычно только на старых системах с дисками > 2 ТБ.

### 5.8.2 Загрузчик UEFI

**UEFI (Unified EFI)** — замена традиционному BIOS, разработанная как расширяемый интерфейс прошивки. Сегодня это наиболее распространённый интерфейс. Включает встроенную оболочку, чтение таблиц разделов GPT и навигацию по файловым системам.

Аспект	MBR	UEFI
Расположение кода	Вне файловой системы (между MBR и разделом)	Специальная файловая система VFAT
Структура	Небольшой фрагмент в MBR загружает остаток	Системный раздел EFI (ESP) с каталогом /EFI
Монтирование в Linux	—	Обычно /boot/efi
Подкаталоги	—	/boot/efi/EFI/{microsoft,apple,ubuntu,grub} и т.д.
Файлы загрузчика	—	Расширение .efi (например, grubx64.efi, shimx64.efi)

**Примечание:** ESP отличается от загрузочного раздела BIOS — в системе будет только один из них.

**Важно:** для UEFI необходимо установить версию загрузчика, написанную для UEFI (не BIOS). Требуется уведомить прошивку о новых загрузчиках и решить проблемы безопасной загрузки

(см. раздел 5.6).

### 5.8.3 Как работает GRUB

Процесс загрузки GRUB:

1. PC BIOS или прошивка инициализирует оборудование и ищет загрузочный код
2. BIOS загружает и выполняет найденный код (GRUB)
3. Загружается ядро GRUB
4. Ядро инициализируется, получает доступ к дискам и файловым системам
5. GRUB идентифицирует загрузочный раздел и загружает конфигурацию
6. GRUB предоставляет пользователю возможность изменить конфигурацию
7. После тайм-аута или действий пользователя выполняется конфигурация (grub.cfg)
8. При необходимости загружаются дополнительные модули
9. Выполняется команда `boot` для загрузки ядра (параметр `linux` в конфигурации)

**Расположение ядра GRUB** имеет три варианта:

- Между MBR и началом первого раздела
- В обычном разделе
- В специальном загрузочном разделе (BIOS boot partition, ESP или другой)

При использовании **BIOS/MBR**: BIOS загружает 512 байт из MBR (`boot.img`), которые содержат начальное местоположение ядра GRUB и загружают его.

При использовании **UEFI/ESP**: ядро GRUB хранится как файл в ESP. Прошивка перемещается по ESP и напрямую запускает весь загрузчик GRUB (возможна прослойка для обработки безопасной загрузки).

**Дополнительный шаг**: перед загрузкой ядра может потребоваться загрузить начальный образ файловой системы оперативной памяти (параметр `initrd`).

From:

<https://wiki.radi0.cc/> - radi0wiki

Permanent link:

<https://wiki.radi0.cc/notes:howlinuxworks:vol5>

Last update: **2026/05/14 14:36**

