

# systemd-boot (Русский)

Ссылки по теме

- [Arch boot process \(Русский\)](#)
- [Загрузчики](#)
- [Unified Extensible Firmware Interface \(Русский\)](#)



Эта страница нуждается в сопроводителе



Статья не гарантирует актуальность информации. Помогите русскоязычному сообществу поддержкой подобных страниц. См. [Команда переводчиков ArchWiki](#)

**Состояние перевода:** На этой странице представлен перевод статьи [systemd-boot](#). Дата последней синхронизации: 2017-12-14. Вы можете [помочь](#) синхронизировать перевод, если в английской версии произошли [изменения](#).

**systemd-boot**, ранее известный как **gummiboot** - это простой UEFI менеджер загрузки, который исполняет настроенные EFI образы. Запись по умолчанию выбирается с помощью настроенного шаблона (glob) или меню на экране. Включен в пакет [systemd](#), который устанавливается на системе Arch по умолчанию.

Прост в настройке, но способен только на запуск исполняемых EFI файлов, таких как ядро Linux [EFISTUB](#), UEFI Shell, GRUB, Windows Boot Manager.

## Contents

[hide]

- 1 [Установка](#)
  - 1.1 [Загрузка в режиме EFI](#)
  - 1.2 [Загрузка в режиме BIOS](#)
  - 1.3 [Обновление](#)
    - 1.3.1 [Вручную](#)
    - 1.3.2 [Автоматически](#)
- 2 [Настройка](#)
  - 2.1 [Базовая настройка](#)
  - 2.2 [Добавление загрузочных записей](#)
    - 2.2.1 [Установки со стандартной корневой директорией](#)
    - 2.2.2 [Установки с LVM корневой директорией](#)
    - 2.2.3 [Установки с зашифрованной корневой директорией](#)
    - 2.2.4 [Установка корневого подраздела btrfs](#)
    - 2.2.5 [Установки с ZFS корневой директорией](#)
    - 2.2.6 [EFI Shells или другие EFI приложения](#)
  - 2.3 [Поддержка гибернации](#)
  - 2.4 [Редактор параметров ядра с защитой паролем](#)
- 3 [Клавиши в загрузочном меню](#)
- 4 [Решение проблем](#)
  - 4.1 [Создание записи вручную с помощью efibootmgr](#)
  - 4.2 [Меню не отображается после обновления Windows](#)
- 5 [Смотрите также](#)

## Установка

### Загрузка в режиме EFI

1. Убедитесь, что вы загружены в режиме UEFI.
2. Проверьте [доступны ли EFI переменные](#).

3. Корректно примонтируйте [Системный Раздел EFI](#) (ESP). В этой статье `esp` используется для обозначения точки монтирования.

**Примечание:** *systemd-boot* EFI не может загружать бинарные файлы из других разделов. По этой причине, рекомендуется монтировать ваш ESP в `/boot`. В случае, если вы хотите разделить `/boot` с ESP, обратитесь к [#Обновление](#) для большей информации.

4. Если ESP **не** примонтирован к `/boot`, копируйте ваше ядро и `initramfs` в ESP.

**Примечание:** Чтобы сохранить автоматическое обновление ядра в ESP, взгляните на [EFISTUB#Using systemd](#) [broken link: invalid section] для адаптации некоторых юнитов `systemd`. Если ваш Системный Раздел EFI монтируется автоматически, вам, вероятно, потребуется добавить `vfat` в файл внутри `/etc/modules-load.d/`. Тогда в текущем запущенном ядре во время загрузки будет установлен модуль `vfat`, до того, как произойдет обновление ядра, которое может заменить модуль для текущей версии, что сделает невозможным монтирование `/boot/efi` до перезагрузки.

5. Введите следующую команду для установки *systemd-boot*:

```
# bootctl --path=esp install
```

Она скопирует двоичный файл *systemd-boot* на Системный Раздел EFI (`esp/EFI/systemd/systemd-bootx64.efi` и `esp/EFI/Boot/BOOTX64.EFI` - оба идентичны на x86-64 системах) и добавит *systemd-boot* как EFI приложение по умолчанию (загрузочная запись по умолчанию), загружаемое с помощью EFI Boot Manager.

6. Наконец, для правильного функционирования вы должны [настроить](#) загрузчик.

## Загрузка в режиме BIOS

**Важно:** Это нерекомендованный процесс

Вы с таким же успехом можете установить *systemd-boot*, если загружаетесь в режиме BIOS. Тем не менее, от вас всё равно требуется сообщить прошивке запускать EFI файл *systemd-boot* при загрузке:

- у вас есть работающий EFI shell где-нибудь.
- ваш интерфейс прошивки предоставляет вам соответствующий способ настройки EFI файла, который будет загружен во время загрузки.

Если вы имеете такую возможность, процесс установки будет проще: перейдите в ваш EFI shell или интерфейс настройки вашей прошивки и измените EFI файл по умолчанию вашей машины на `esp/EFI/systemd/systemd-bootx64.efi` (или `systemd-bootia32.efi` если у вас 32 битная системная прошивка).

**Примечание:** интерфейс прошивки в Dell Latitude сериях предоставляет все необходимое, чтобы установить EFI загрузку, но EFI Shell не сможет осуществить запись в ПЗУ компьютера.

## Обновление

В отличие от предыдущего отдельного пакета *gummiboot*, который автоматически обновляется с помощью `post_install` скрипта, обновления *systemd-boot* теперь должны производиться пользователем вручную. Однако, эта процедура может быть автоматизирована с использованием `raspm` hooks.

### Вручную

*systemd-boot* (`bootctl(1)`) предполагает, что ваш Системный Раздел EFI примонтирован в `/boot`.

```
# bootctl update
```

Если ESP не примонтирован в `/boot`, опцией `--path=` можно явно указать точку монтирования, например:

```
# bootctl --path=esp update
```

**Примечание:** Также эту команду следует использовать при переходе с *gummiboot*, перед удалением этого пакета. Однако, если этот пакет уже был удален, выполните `bootctl --path=esp install`.

### Автоматически

**AUR (Русский)** пакет [systemd-boot-pacman-hook](#)<sup>AUR</sup> предоставляет **Pacman hook** для автоматизации процесса обновления. **Установка** этого пакета добавит hook, который будет выполняться при каждом обновлении пакета [systemd](#).

В качестве альтернативы, вы можете разместить следующий pacman hook в каталоге `/etc/pacman.d/hooks/`:

```
/etc/pacman.d/hooks/systemd-boot.hook
```

```
[Trigger]
Type = Package
Operation = Upgrade
Target = systemd

[Action]
Description = Updating systemd-boot...
When = PostTransaction
Exec = /usr/bin/bootctl update
```

## Настройка

### Базовая настройка

Базовая конфигурация хранится в файле `esp/loader/loader.conf` и состоит из трех опций:

- `default` – выбираемая по умолчанию запись (без суффикса `.conf`); можно использовать подстановку, например `arch-*`
- `timeout` – задержка меню в секундах. Если таймаут не задан, то меню будет отображаться, только если удерживать клавишу `Space` (другие клавиши тоже могут работать) при загрузке.
- `editor` - следует ли включить редактор параметров ядра. `1` (по умолчанию) - включить, `0` - отключить; Поскольку пользователь может добавить `init=/bin/bash` для обхода пароля администратора и получить полный доступ, настоятельно рекомендуется установить эту опцию в `0`.

Пример:

```
esp/loader/loader.conf
```

```
default arch
timeout 4
editor 0
```

**Примечание:** Обратите внимание, что первые 2 опции могут быть изменены в самом меню загрузки, эти изменения будут храниться как переменные EFI.

**Совет:** Пример базового конфигурационного файла расположен как `/usr/share/systemd/bootctl/loader.conf`.

## Добавление загрузочных записей

**Примечание:**

- `bootctl` будет автоматически проверять наличие "Windows Boot Manager" (`\EFI\Microsoft\Boot\Bootmgfw.efi`), "EFI Shell" (`\shellx64.efi`) и "EFI Default Loader" (`\EFI\Boot\bootx64.efi`) во время загрузки, так же как специально подготовленные файлы ядра, найденные в `\EFI\Linux`. После обнаружения соответствующие записи с заголовками `auto-windows`, `auto-efi-shell` и `auto-efi-default` будут автоматически сгенерированы. Эти записи не требуют ручной настройки загрузчика. Однако, другие EFI приложения не будут обнаружены автоматически (не в случае с [rEFInd \(Русский\)](#)), поэтому для загрузки ядра Linux, записи должны быть созданы вручную.
- Если вы используете двойную загрузку с Windows, настоятельно рекомендуется отключить [Быстрый запуск](#)
- Не забудьте загрузить intel [microcode](#) с `initrd` если это применимо в вашем случае.
- Вы можете узнать PARTUUID вашего корневого раздела с помощью команды `blkid -s PARTUUID -o value /dev/sdxY`, где `x` - это буква устройства, а `Y` - это номер раздела. Это нужно только для вашего корневого раздела, не для `esp`.

`bootctl` ищет элементы для загрузочного меню в `esp/loader/entries/*.conf` – каждый найденный файл должен содержать точно одну загрузочную запись. Возможными опциями являются:

- `title` – название операционной системы. **Обязательная.**
- `version` – версия ядра, отображаемая только если существуют несколько записей с одинаковым названием. Не обязательная.
- `machine-id` – идентификатор машины из `/etc/machine-id`, отображаемый только если существуют несколько записей с одинаковым названием и одинаковой версией. Не обязательная.
- `efi` – EFI программа для запуска, относительно вашего ESP (`$esp`); например, `/vmlinuz-linux`. Либо это, либо `linux` (смотрите ниже) является **обязательным**.
- `options` – опции командной строки для передачи EFI приложению. Не обязательная, но вам нужно будет передать как минимум `initrd=efipath` и `root=dev` если загружаете Linux.

Для Linux вы можете задать `linux path-to-vmlinuz` и `initrd path-to-initramfs`; это автоматически преобразуется в `efi path` и `options initrd=path` – этот синтаксис поддерживается только для удобства и не имеет различий по функциональности.

## Установки со стандартной корневой директорией

Вот пример записи для корневого раздела без LVM или LUKS:

```
esp/loader/entries/arch.conf
```

```
title Arch Linux
linux /vmlinuz-linux
initrd /initramfs-linux.img
options root=PARTUUID=14420948-2cea-4de7-b042-40f67c618660 rw
```

Пожалуйста, обратите внимание, что в вышеприведённом примере `PARTUUID/PARTLABEL` идентифицируют GPT раздел, а это не то же самое, что `UUID/LABEL`, которые идентифицируют файловую систему.

Использование `PARTUUID/PARTLABEL` бывает полезным, потому что они инвариантны (то есть неизменяемы), если вы переформатируете раздел в другую файловую систему или если по какой-то причине изменятся обозначения `/dev/sd*`. Также оно может быть полезно, если у вас нет файловой системы на разделе (или вы используете LUKS, который не поддерживает метки `LABEL`).

**Совет:** Пример файла записи расположен как `/usr/share/systemd/bootctl`.

### Установки с LVM корневой директорией

**Важно:** `systemd-boot` не может использоваться без отдельной файловой системы `/boot` вне LVM.

Следующий пример для корневой директории с использованием [логического менеджера разделов](#):

```
esp/loader/entries/arch-lvm.conf
```

```
title Arch Linux (LVM)
linux /vmlinuz-linux
initrd /initramfs-linux.img
options root=/dev/mapper/<VolumeGroup-LogicalVolume> rw
```

Замените `<VolumeGroup-LogicalVolume>` на актуальные названия VG и LV (например, `root=/dev/mapper/volgroup00-lvolroot`). Кроме того, вместо них можно использовать UUID:

```
....
options root=UUID=<UUID identifier> rw
```

Обратите внимание, что `root=UUID=` используется вместо `root=PARTUUID=`, который используется для корневых разделов без LVM или LUKS.

### Установки с зашифрованной корневой директорией

Ниже приведен пример конфигурационного файла для зашифрованного корневого раздела ([DM-Crypt / LUKS](#)) с использованием `encrypt` [mkinitcpio \(Русский\)](#) hook:

```
esp/loader/entries/arch-encrypted.conf
```

```
title Arch Linux Encrypted
linux /vmlinuz-linux
initrd /initramfs-linux.img
```

```
options cryptdevice=UUID=<UUID>:<mapped-name> root=/dev/mapper/<mapped-name>
quiet rw
```

В этом примере используется UUID; если хотите, можете заменить UUID на PARTUUID. Вы можете также заменить /dev путь на регулярный UUID. mapped-name - название, которое вы желаете использовать. Смотрите [Dm-crypt/System configuration#Boot loader](#).

Если вы используете LVM, ваша опция cryptdevice будет выглядеть следующим образом:

```
esp/loader/entries/arch-encrypted-lvm.conf

title Arch Linux Encrypted LVM
linux /vmlinuz-linux
initrd /initramfs-linux.img
options cryptdevice=UUID=<UUID>:MyVolGroup root=/dev/mapper/MyVolGroup-
MyVolRoot quiet rw
```

Вы также можете добавить другие EFI приложения, такие как \EFI\arch\grub.efi.

### Установка корневого подраздела btrfs

При загрузке с подраздела [Btrfs \(Русский\)](#) в роли корневого, добавьте к строке options rootflags=subvol=<root subvolume>. В примере ниже, корневой раздел монтируется как btrfs подраздел с именем 'ROOT' (например, mount -o subvol=ROOT /dev/sdxY /mnt):

```
esp/loader/entries/arch-btrfs-subvol.conf

title Arch Linux
linux /vmlinuz-linux
initrd /initramfs-linux.img
options root=PARTUUID=14420948-2cea-4de7-b042-40f67c618660 rw
rootflags=subvol=ROOT
```

Если это невозможно сделать, то это приведет к ошибке: ERROR: Root device mounted successfully, but /sbin/init does not exist.

### Установки с ZFS корневой директорией

В случае загрузки из [ZFS](#) dataset, добавьте zfs=<root dataset> к строке options. Здесь в корневом dataset установлено значение 'zroot/ROOT/default':

```
esp/loader/entries/arch-zfs.conf

title Arch Linux ZFS
linux /vmlinuz-linux
initrd /initramfs-linux.img
options zfs=zroot/ROOT/default rw
```

When booting off of a ZFS dataset ensure that it has had the bootfs property set with zpool set bootfs=<root dataset> <zpool>.

## EFI Shells или другие EFI приложения

В случае, если вы установили EFI Shells или другие EFI приложения в ESP, вы можете использовать следующие фрагменты:

```
esp/loader/entries/uefi-shell-v1-x86_64.conf
```

```
title  UEFI Shell x86_64 v1
efi    /EFI/shellx64_v1.efi
```

```
esp/loader/entries/uefi-shell-v2-x86_64.conf
```

```
title  UEFI Shell x86_64 v2
efi    /EFI/shellx64_v2.efi
```

## Поддержка гибернации

Пожалуйста, прочтите статью [Power management/Suspend and hibernate \(Русский\)](#).

## Редактор параметров ядра с защитой паролем

В качестве альтернативы можете установить [systemd-boot-password<sup>AUR</sup>](#) который поддерживает `password` опцию базовой конфигурации. Используйте `sbpctl generate` для генерации значения для этой опции.

Установка `systemd-boot-password` следующей командой:

```
# sbpctl install esp
```

С включенным редактором будет запрошен ваш пароль, перед тем как вы сможете редактировать параметры ядра.

## Клавиши в загрузочном меню

В меню используются следующие клавиши:

- `Вверх/Вниз` - выбор записи
- `Enter` - загрузить выбранную запись
- `d` - выбрать загрузочную запись по умолчанию (хранится в энергонезависимой EFI переменной)
- `-/T` - уменьшить таймаут (хранится в энергонезависимой EFI переменной)
- `+/t` - увеличить таймаут (хранится в энергонезависимой EFI переменной)
- `e` - редактировать командную строку ядра. Не имеет эффекта, если опция конфигурации `editor` установлена в `0`.
- `v` - показать версию gummiboot и UEFI
- `Q` - выйти
- `P` - отобразить текущую конфигурацию
- `h/?` - помощь

А эти клавиши, нажатые в меню в процессе загрузки, сразу загрузят определённую запись:

- `l` - Linux
- `w` - Windows
- `a` - OS X

- `s` - EFI Shell
- `1-9` - порядковый номер записи

## Решение проблем

---

### Создание записи вручную с помощью efibootmgr

Если команда `bootctl install` не сработала, вы можете создать загрузочную EFI запись самостоятельно с помощью утилиты `efibootmgr`:

```
# efibootmgr -c -d /dev/sdX -p Y -l /EFI/systemd/systemd-bootx64.efi -L  
"Linux Boot Manager"
```

где `/dev/sdXY` - это ваш EFISYS раздел.

### Меню не отображается после обновления Windows

Обратитесь к разделу соответствующей статьи: [UEFI \(Русский\)#Windows изменяет порядок загрузки](#).

## Смотрите также

---

- <http://www.freedesktop.org/wiki/Software/systemd/systemd-boot/>

Categories:

- [Boot loaders \(Русский\)](#)
- [Русский](#)