

EFISTUB (Русский)

Ссылки по теме

- [Процесс загрузки Arch](#)
- [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 [EFISTUB](#). Дата последней синхронизации: 2015-07-29. Вы можете [помочь](#) синхронизировать перевод, если в английской версии произошли [изменения](#).

Важно: Был обнаружен баг, при котором загрузка EFISTUB могла не сработать в зависимости от версии ядра и модели материнской платы. Смотрите [FS#33745](#) и [\[1\]](#) для дополнительной информации.

Ядро Linux ([linux](#)>=3.3) поддерживает EFISTUB (EFI BOOT STUB) загрузку. Эта возможность позволяет прошивке EFI загружать ядро как обычное EFI приложение. Эта опция включена по умолчанию в стандартных ядрах Arch Linux, а также её можно активировать при помощи установки `CONFIG_EFI_STUB=y` в конфигурации ядра (смотрите [The EFI Boot Stub](#) для дополнительной информации).

EFISTUB ядро может быть загружено непосредственно с помощью UEFI материнской платы или же посредственно с использованием [UEFI менеджера загрузки](#). Последний рекомендуется использовать в том случае, если у вас есть несколько ядер/initramfs пар и загрузочное меню UEFI вашей материнской платы неудобное в использовании.

Contents

[hide]

- [1Настройка EFISTUB](#)
 - [1.1Альтернативные точки монтирования для ESP](#)
 - [1.1.1С помощью systemd](#)
 - [1.1.2С помощью incron](#)
 - [1.1.3С помощью хука mkinitcpio](#)
 - [1.1.4С помощью биндинга монтирования](#)
- [2Загрузка EFISTUB](#)
 - [2.1Используя менеджер загрузки](#)
 - [2.2Используя UEFI Shell](#)
 - [2.3Используя непосредственно UEFI \(efibootmgr\)](#)

Настройка EFISTUB

После создания [системного раздела EFI](#) ^{[[broken link](#): invalid section]} вы должны выбрать точку монтирования для него. Самый простой способ - это смонтировать его в `/boot`, так как это позволит распаковать непосредственно обновлять ядро, которое будет использовать EFI прошивка. Если вы пойдёте по этому пути, переходите к разделу [#Загрузка EFISTUB](#).

Примечание: Вы можете хранить ядро и initramfs не на ESP, если вы используете менеджер загрузки, у которого есть драйвер файловой системы для раздела, в котором они расположены, например [rEFInd](#).

Альтернативные точки монтирования для ESP

Если вы примонтируете Системный Раздел EFI куда-либо в другое место (например, в `/boot/efi`), вам нужно будет скопировать загружаемые файлы в эту директорию (здесь и далее будем обозначать её как `$esp`).

```
# mkdir $esp/EFI/arch
# cp /boot/vmlinuz-linux $esp/EFI/arch/vmlinuz-linux
# cp /boot/initramfs-linux.img $esp/EFI/arch/initramfs-linux.img
# cp /boot/initramfs-linux-fallback.img $esp/EFI/arch/initramfs-linux-fallback.img
```

Кроме того, вам нужно будет постоянно обновлять файлы на ESP при последующих обновлениях ядра. Если этого не делать, то система может перестать загружаться. Следующие далее разделы описывают механизмы автоматизации данных действий.

С помощью systemd

Systemd умеет выполнять задания при наступлении событий. В данном случае используется возможность отслеживания изменений в определённой папке для синхронизации EFISTUB ядра и initramfs файлов при их обновлении в `/boot`.

```
/etc/systemd/system/efistub-update.path

[Unit]
Description=Copy EFISTUB Kernel to UEFISYS Partition

[Path]
PathChanged=/boot/initramfs-linux-fallback.img

[Install]
WantedBy=multi-user.target
```

Примечание: Здесь используется отслеживание изменений в `initramfs-linux-fallback.img`, так как это последний файл, собранный с помощью `mkinitcpio`. Это сделано для того, чтобы избежать начала потенциальной гонки, когда `systemd` может копировать старые файлы до того, как все они будут собраны.

```
/etc/systemd/system/efistub-update.service

[Unit]
Description=Copy EFISTUB Kernel to UEFISYS Partition

[Service]
Type=oneshot
ExecStart=/usr/bin/cp -f /boot/vmlinuz-linux $esp/EFI/arch/vmlinuz-linux
ExecStart=/usr/bin/cp -f /boot/initramfs-linux.img $esp/EFI/arch/initramfs-linux.img
ExecStart=/usr/bin/cp -f /boot/initramfs-linux-fallback.img $esp/EFI/arch/initramfs-linux-fallback.img
```

Включите эти сервисы с помощью:

```
# systemctl enable efistub-update.path
```

Вам нужно будет перезагрузиться либо сказать systemd начать отслеживать папку с помощью:

```
# systemctl start efistub-update.path
```

С помощью incron

Чтобы запускать скрипт синхронизации EFISTUB ядра при обновлениях ядра можно использовать [incron](#).

```
/usr/local/bin/efistub-update.sh
```

```
#!/usr/bin/env bash
/usr/bin/cp -f /boot/vmlinuz-linux $esp/EFI/arch/vmlinuz-linux
/usr/bin/cp -f /boot/initramfs-linux.img $esp/EFI/arch/initramfs-linux.img
/usr/bin/cp -f /boot/initramfs-linux-fallback.img $esp/EFI/arch/initramfs-
linux-fallback.img
```

Примечание: Первый параметр `/boot/initramfs-linux-fallback.img` - это файл для отслеживания. Второй параметр `IN_CLOSE_WRITE` - это действие, которое надо отслеживать. Третий параметр `/usr/local/bin/efistub-update.sh` - это скрипт, который будет выполнен.

```
/etc/incron.d/efistub-update.conf
```

```
/boot/initramfs-linux-fallback.img IN_CLOSE_WRITE /usr/local/bin/efistub-
update.sh
```

Чтобы использовать данный метод, включите сервис incron:

```
# systemctl enable incron.service
```

С помощью хука mkinitcpio

Mkinitcpio может сгенерировать хук, которому для работы не требуется демон системного уровня. Он порождает фоновый процесс, который ждёт генерации `vm-linux`, `initramfs-linux.img` и `initramfs-linux-fallback.img` перед копированием файлов.

Добавьте `efistub-update` в список хуков в `/etc/mkinitcpio.conf`.

```
/usr/lib/initcpio/install/efistub-update
```

```
#!/usr/bin/env bash
build() {
    /root/watch.sh &
}

help() {
    cat <<HELPEOF
```

```
This hook waits for mkinitcpio to finish and copies the finished ramdisk and
kernel to the ESP
HELPEOF
}
```

```
/root/watch.sh

#!/usr/bin/env bash

while [[ -d "/proc/$PPID" ]]; do
    sleep 1
done

/usr/bin/cp -f /boot/vmlinuz-linux $esp/EFI/arch/vmlinuz-linux
/usr/bin/cp -f /boot/initramfs-linux.img $esp/EFI/arch/initramfs-linux.img
/usr/bin/cp -f /boot/initramfs-linux-fallback.img $esp/EFI/arch/initramfs-
linux-fallback.img

echo "Synced kernel with ESP"
```

С помощью биндинга монтирования

Вместо того, чтобы монтировать сам ESP раздел в `/boot`, вы можете примонтировать директорию ESP раздела в `/boot` используя `bind mount` (смотрите `mount(8)`). Это позволит растап обновлять ядро непосредственно, в то же время оставляя организацию ESP на ваше усмотрение. Если это у вас работает, то данный способ является гораздо проще всех остальных, которые копируют файлы.

Примечание: Для этого требуется ядро и загрузчик, совместимый с FAT32. Это не проблема для обычных установок Arch, но может быть проблемой для других дистрибутивов (а именно тех, которым требуются симлинки в `/boot`). Сообщение на форуме [здесь](#).

Как и [прежде](#), скопируйте все загружаемые файлы в директорию на вашем ESP, но смонтируйте ESP **вне** `/boot` (например, в `/esp`). Затем примонтируемый каталог:

```
# mount --bind /esp/EFI/arch/ /boot
```

Если ваши файлы появились в `/boot`, как и ожидалось, отредактируйте ваш [Fstab](#), чтобы сделать монтирование постоянным:

```
/etc/fstab

/esp/EFI/arch /boot none defaults,bind 0 0
```

Важно: Вы *обязаны* использовать [параметр ядра](#) `root=system_root` для того чтобы загрузиться с использованием данного метода.

Загрузка EFISTUB

Важно: Путь до Linux EFISTUB ядра `initramfs` должен быть относительным к корню Системного Раздела EFI. Например, если `initramfs` расположен в `$esp/EFI/arch/initramfs-`

linux.img, то соответствующей UEFI строкой должна быть `initrd=/EFI/arch/initramfs-linux.img` или `initrd=\EFI\arch\initramfs-linux.img`. В следующих примерах будет предполагаться, что всё расположено в `$esp/`.

Используя менеджер загрузки

Есть несколько UEFI менеджеров загрузки, которые могут предоставить дополнительные опции или упростить процесс загрузки UEFI, особенно если у вас есть несколько ядер/операционных систем. Смотрите [Загрузчики](#) для дополнительной информации.

Используя UEFI Shell

Есть возможность запускать EFISTUB ядро из UEFI Shell как обычное UEFI приложение. В этом случае параметры ядра передаются как обычные параметры запускаемому файлу EFISTUB ядра.

```
> fs0:
> /vmlinuz-linux root=PARTUUID=3518bb68-d01e-45c9-b973-0b5d918aae96
initrd=/initramfs-linux.img
```

Чтобы избежать необходимости каждый раз вспоминать все параметры вашего ядра, вы можете сохранить исполняемую команду в shell скрипт, например в `archlinux.nsh` на вашем Системном Разделе UEFI, а затем запустить её с помощью:

```
> fs0:
> archlinux
```

Используя непосредственно UEFI (efibootmgr)

UEFI [разработан так, чтобы убрать необходимость](#) наличия промежуточных загрузчиков, таких как [GRUB](#). Если ваша материнская плата имеет хорошую реализацию UEFI, есть возможность вставлять параметры ядра внутрь загрузочных записей UEFI и материнская плата сможет загружать непосредственно Arch. Вы можете использовать `efibootmgr` чтобы модифицировать загрузочные записи вашей материнской платы изнутри Arch.

```
# efibootmgr -d /dev/sdX -p Y -c -L "Arch Linux" -l /vmlinuz-linux -u
"root=/dev/sda2 rw initrd=/initramfs-linux.img"
```

Где `X` и `Y` замените на соответствующий диск и раздел, где находится ваш ESP. Измените параметр `root=`, чтобы он соответствовал корню вашего Linux (также можно использовать UUID диска).

Или с помощью действия [приостановки на диск](#):

```
# efibootmgr -d /dev/sdX -p Y -c -L "Arch Linux" -l /vmlinuz-linux -u
"root=/dev/sda2 rw resume=/dev/sda4 initrd=/initramfs-linux.img"
```

Измените параметр `resume=` на соответствующий вашему swap разделу, оставив остальные параметры такими, как описано выше.

Затем хорошей идеей будет выполнение

```
# efibootmgr -v
```

для проверки того, что получившаяся запись корректна.

Важно: Некоторые комбинации версий ядер и `efibootmgr` могут отказываться создавать новые загрузочные записи. Это может происходить из-за того что не хватает свободного места в NVRAM. Вы можете попробовать удалить какие-нибудь мусорные EFI файлы

```
# rm /sys/firmware/efi/efivars/dump-*
```

Или же, в качестве последней надежды, загрузитесь с параметром ядра `efi_no_storage_paranoid`. Вы также можете попробовать откатить `efibootmgr` до версии 0.11.0, если она доступна в вашем кеше. Эта версия работает с Linux версии 4.0.6. Смотрите обсуждение ошибки [FS#34641](#) для дополнительной информации.

Чтобы задать порядок загрузки выполните:

```
# efibootmgr -o XXXX,XXXX
```

где XXXX - это номер, который появляется в выводе команды ``efibootmgr`` напротив каждой записи.

Совет: Сохраните где-нибудь команду для создания вашей загрузочной записи в шелл скрипте, который поможет вам при изменениях (например, когда вы захотите изменить параметры ядра).

Дополнительную информацию по `efibootmgr` смотрите в [UEFI \(Русский\)#efibootmgr](#). Пост на форуме <https://bbs.archlinux.org/viewtopic.php?pid=1090040#p1090040> .

Categories:

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