

Kernel module (Русский)

Состояние перевода: На этой странице представлен перевод статьи **Kernel modules**. Дата последней синхронизации: 2015-07-29. Вы можете **помочь** синхронизировать перевод, если в английской версии произошли **изменения** (https://wiki.archlinux.org/index.php?title=Kernel_modules&diff=0&oldid=388445).

Ссылки по теме

[Устранение часто встречающихся неполадок](#)

[Ядра](#)

[Параметры ядра](#)

[Compile kernel module](#)

Модули ядра — это отдельные кусочки кода, которые могут быть загружены и выгружены из ядра по мере необходимости. Они расширяют функциональность ядра без необходимости перезагрузки системы.

Contents

[Обзор](#)

[Получение информации](#)

[Автоматическое управление модулями](#)

[Управление модулями вручную](#)

[Настройка параметров модуля](#)

[С помощью файлов в /etc/modprobe.d/](#)

[С помощью командной строки ядра](#)

[Создание псевдонимов](#)

[Запрет загрузки](#)

[С помощью файлов в /etc/modprobe.d/](#)

[С помощью командной строки ядра](#)

[Решение проблем](#)

[Модули не загружаются](#)

[Смотрите также](#)

Обзор

Чтобы создать модуль ядра, вы можете прочитать **The Linux Kernel Module Programming Guide** (<http://ltdp.org/LDP/lkmpg/2.6/html/index.html>). Модуль можно сконфигурировать как вкомпилированный, а можно как загружаемый. Чтобы иметь возможность динамически загружать или выгружать модуль, его необходимо сконфигурировать как загружаемый модуль в настройке ядра (в этом случае строка, относящаяся к модулю должна быть отмечена буквой **M**).

Модули хранятся в `/usr/lib/modules/kernel_release`. Чтобы узнать текущую версию вашего ядра, используйте команду `uname -r`.

Примечание: Часто в названии модулей используются подчёркивания (`_`) или дефисы (`-`).

–); однако, эти символы взаимозаменяемы как при использовании команды `modprobe`, так и в конфигурационных файлах в `/etc/modprobe.d/`.

Получение информации

Чтобы узнать, какие модули ядра загружены в настоящий момент:

```
$ lsmod
```

Чтобы показать информацию о модуле:

```
$ modinfo module_name
```

Чтобы вывести список опций, с которыми загружен модуль:

```
$ systool -v -m module_name
```

Чтобы отобразить настройки для всех модулей:

```
$ modprobe -c | less
```

Чтобы отобразить настройки для отдельного модуля:

```
$ modprobe -c | grep module_name
```

Чтобы узнать зависимости модуля (или его псевдонима), включая сам модуль:

```
$ modprobe --show-depends module_name
```

Автоматическое управление модулями

Сегодня все необходимые загрузки модулей делаются автоматически с помощью [udev](#), поэтому если вам не нужно загружать какие-либо модули, не входящие в стандартное ядро, вам не придётся прописывать модули, требующиеся для загрузки в каком-либо конфигурационном файле. Однако, бывают случаи, когда вам необходимо загружать свой модуль в процессе загрузки или наоборот не загружать какой-то стандартный модуль, чтобы ваш компьютер правильно функционировал.

Чтобы дополнительные модули ядра загружались автоматически в процессе загрузки, создаются статические списки в конфигурационных файлах в директории `/etc/modules-load.d/`. Каждый конфигурационный файл называется по схеме `/etc/modules-load.d/<program>.conf`. Эти конфигурационные файлы содержат список названий модулей ядра, которые необходимо грузить, разделённых переносом строки. Пустые строки и строки, в которых первым непобелым символом является `#` или `;`, игнорируются.

```
/etc/modules-load.d/virtio-net.conf
```

```
# Load virtio-net.ko at boot
```

```
virtio-net
```

Смотрите `modules-load.d(5)` (<https://jlk.fjfi.cvut.cz/arch/manpages/man/modules-load.d.5>) для дополнительной информации.

Управление модулями вручную

Управление модулями ядра производится с помощью утилит, предоставляемых пакетом `kmod` (<https://www.archlinux.org/packages/?name=kmod>). Вы можете использовать эти утилиты вручную.

Примечание: Если вы обновили ваше ядро, но ещё не перезагрузились, `modprobe` не сработает без каких либо уведомлений об ошибках и завершится с ошибкой 1, потому что путь `/lib/modules/$(uname -r)/` больше не существует. Проверьте вручную существование этого пути, если `modprobe` не работает, чтобы убедиться, что это ваш случай.

Загрузка модуля:

```
# modprobe module_name
```

Загрузка модуля из другого места (для тех модулей, которых нет в `/lib/modules/$(uname -r)/`):

```
# insmod filename [args]
```

Выгрузка модуля:

```
# modprobe -r module_name
```

Альтернативный вариант выгрузки модуля:

```
# rmmod module_name
```

Настройка параметров модуля

Чтобы передать параметр модулю ядра, вы можете воспользоваться конфигурационным файлом в `modprobe` или использовать командную строку ядра.

С помощью файлов в `/etc/modprobe.d/`

Файлы в директории `/etc/modprobe.d/` можно использовать для передачи настроек модуля в `udev`, который через `modprobe` управляет загрузкой модулей во время загрузки системы. Конфигурационные файлы в этой директории могут иметь любое имя, оканчивающееся расширением `.conf`. Синтаксис следующий:

```
/etc/modprobe.d/myfilename.conf
```

```
options modname parametername=parametervalue
```

Например:

```
/etc/modprobe.d/thinkfan.conf
```

```
# On ThinkPads, this lets the 'thinkfan' daemon control fan speed
options thinkpad_acpi fan_control=1
```

Примечание: Если какой-либо из затрагиваемых модулей загружается из `initramfs`, тогда вам придётся добавить соответствующий `.conf` файл в `FILES` в файле `/etc/mkinitcpio.conf` или использовать `modconf` [хук](#), чтобы он был включен в `initramfs`.

С помощью командной строки ядра

Если модуль вкомпилирован в ядро, вы также можете передать параметры модулю с помощью командной строки ядра. Для всех стандартных загрузчиков, подойдёт следующий синтаксис:

```
modname.parametername=parametercontents
```

Например:

```
thinkpad_acpi.fan_control=1
```

Просто добавьте это в загрузчике в строку с ядром, как описано в [параметрах ядра](#).

Создание псевдонимов

Псевдонимы (алиасы) - это альтернативные названия для модуля. Например: `alias my-mod really_long_modulename` означает, что вы можете использовать `modprobe my-mod` вместо `modprobe really_long_modulename`. Вы также можете использовать звёздочки в стиле `shell`, то есть `alias my-mod* really_long_modulename` будет иметь тот же эффект, что и `modprobe my-mod-something`. Создайте алиас:

```
/etc/modprobe.d/myalias.conf
```

```
alias mymod really_long_module_name
```

У некоторых модулей есть алиасы, которые используются для их автоматической загрузки, когда они потребуются определённой программе. Отключение этих алиасов может предотвратить их автоматическую загрузку, при этом остаётся возможность из загрузки вручную.

```
/etc/modprobe.d/modprobe.conf
```

```
# Prevent Bluetooth autoload
alias net-pf-31 off
```

Запрет загрузки

В терминах модулей ядра **blacklisting** означает механизм, предотвращающий загрузку какого-то модуля. Это может понадобиться, например если вам не нужна работа какого-то оборудования или если загрузка данного модуля вызывает проблемы: например, могут быть два модуля ядра, которые пытаются управлять одним и тем же оборудованием, и их совместная загрузка приводит к конфликту.

Некоторые модули загружаются как часть [initramfs](#). Команда `mkinitcpio -M` напечатает все автоматически обнаруженные модули: для предотвращения `initramfs` от загрузки каких-то из этих модулей, занесите их в чёрный список в `/etc/modprobe.d/modprobe.conf`. Команда `mkinitcpio -v` отобразит все модули, которые необходимы некоторым хукам (например, `filesystems` хук, `block` хук и т.д.). Не забудьте добавить этот `.conf` файл в секцию `FILES` в `/etc/mkinitcpio.conf`, если вы этого ещё не сделали, пересоберите `initramfs` после того, как вы запретили загрузку модулей, а затем перезагрузитесь.

С помощью файлов в `/etc/modprobe.d/`

Создайте `.conf` файл в `/etc/modprobe.d/` и добавьте строку для каждого модуля, который вы хотите запретить, используя ключевое слово `blacklist`. Например, если вы хотите запретить загружать модуль `pcspkr`:

```
/etc/modprobe.d/nobeep.conf

# Do not load the 'pcspkr' module on boot.
blacklist pcspkr
```

Примечание: Команда `blacklist` запретит автоматическую загрузку модуля, но этот модуль всё равно может загрузиться, если от него зависит какой-то не запрещённый модуль или если он загружен вручную.

Можно изменить такое поведение. Команда `install` заставляет `modprobe` запускать вашу собственную команду вместо вставки модуля в ядро как обычно. Поэтому вы можете насильно сделать так, чтобы модуль никогда не загружался:

```
/etc/modprobe.d/blacklist.conf

...
install module_name /bin/false
...
```

Это запретит данный модуль и все модули, зависящие от него.

С помощью командной строки ядра

Совет: Это может очень помочь, если неправильный модуль не даёт загрузиться вашей системе.

Вы также можете запрещать модули из загрузчика.

Просто добавьте `modprobe.blacklist=modname1,modname2,modname3` в вашем загрузчике в строку ядра, как описано в [параметрах ядра](#).

Примечание: Когда вы запрещаете несколько модулей, обратите внимание, что они разделяются только запятой. Пробелы или что-либо ещё могут нарушить синтаксис.

Решение проблем

Модули не загружаются

В случае, если конкретный модуль не загружается и журнал загрузки (доступный с помощью `journalctl -b`) говорит, что модуль в чёрном списке, но в директории `/etc/modprobe.d/` нет соответствующей записи, проверьте другую папку-источник `modprobe` в `/usr/lib/modprobe.d/` на записи в чёрном списке.

Смотрите также

- [Отключение PC Speaker](#)

Retrieved from "[https://wiki.archlinux.org/index.php?title=Kernel_module_\(Русский\)&oldid=595721](https://wiki.archlinux.org/index.php?title=Kernel_module_(Русский)&oldid=595721)"

This page was last edited on 20 January 2020, at 14:43.

Content is available under [GNU Free Documentation License 1.3 or later](#) unless otherwise noted.