

Wiki / Kernelmodule

Dieser Artikel wurde für die folgenden Ubuntu-Versionen getestet:

Dieser Artikel ist größtenteils für alle Ubuntu-Versionen gültig.

Zum Verständnis dieses Artikels sind folgende Seiten hilfreich:

1. **Installation von Programmen**
2. **Ein Terminal öffnen**
3. **Einen Editor öffnen**

Inhaltsverzeichnis

1. Installation
2. Befehle für Module
3. Module automatisch laden
4. Module mit Optionen laden
5. Automatisches Laden verhindern - Blacklist
6. Übersicht über alle Module
7. Links

Dieser Artikel beschreibt den Umgang mit Kernelmodulen und den Hilfsmitteln, die das System bietet, um Module zu verwalten.

Grundsätzlich besitzt der Linux-**Kernel** [https://wiki.ubuntuusers.de/Kernel] einen **monolithischen Aufbau** [http://de.wikipedia.org/wiki/Linux_(Kernel)#Architektur], allerdings bietet er auch die Möglichkeit, Module zur Laufzeit zu laden und entladen. Unter einem Kernelmodul versteht man im Allgemeinen einen Teil bzw. eine Erweiterung des Kernels. So sind z.B. alle Hardwaretreiber (z.B. WLAN-Karte, Soundkarte...) als Modul angelegt.

Die meisten Linux-Distributionen inklusive Ubuntu machen starken Gebrauch von Kernelmodulen, da so die bestmögliche Hardwarekompatibilität mit nur einem "allgemeinen" Kernel erreicht werden kann. In der Regel werden alle notwendigen Module beim Systemstart und falls nötig zur Laufzeit geladen, so dass der Nutzer sich nicht weiter darum kümmern muss (bzw. es oft noch nicht einmal mitbekommt).

Installation

Die Programme zum Umgang mit den Modulen sollten in der Standardinstallation enthalten sein, können aber ansonsten über das Paket

- **module-init-tools**



Jetzt installieren

[apt://module-init-tools] mit **apturl** [https://wiki.ubuntuusers.de/apturl]

```
sudo apt-get install module-init-tools
```

installiert werden ^[1].

Befehle für Module

Im Folgenden werden einige wichtige Befehle für den Umgang mit Modulen kurz vorgestellt. Alle Befehle werden im Terminal ^[2] aufgerufen. Das Laden, Entladen sowie Erstellen von Abhängigkeiten erfordert

Root-Rechte [\[https://wiki.ubuntuusers.de/sudo/\]](https://wiki.ubuntuusers.de/sudo/)!

Geladene Module anzeigen: lsmod

Mit dem Befehl

```
lsmod
```

können alle aktuell geladenen Module angezeigt werden. lsmod kennt keine weiteren Optionen.

Beispielausgabe:

Module	Size	Used by
btrfs	835954	0
raid6_pq	97812	1 btrfs
xor	21411	1 btrfs
ufs	74890	0
qnx4	13317	0
hfsplus	107516	0
hfs	54677	0
minix	26140	0

Wie man sieht, wird in der ersten Spalte der Modulname angezeigt, in der zweiten Spalte der belegte Speicher in Bytes und in der dritten Spalte die Anzahl der Prozesse und Module, die dieses Modul benutzen, gefolgt von einer Liste der Modulnamen, die dieses Modul benutzen.

Information anzeigen: modinfo

Mit Hilfe des Befehls **modinfo** werden Informationen zu einem bestimmten Modul angezeigt. Die allgemeine Syntax lautet:

```
modinfo MODULNAME
```

wobei MODULNAME durch das entsprechende Modul ersetzt werden muss. Beispiel:

```
$ modinfo snd
```

Ausgabe:

filename:	/lib/modules/2.6.20-15-generic/kernel/sound/core/snd.ko
alias:	char-major-116-*
license:	GPL

```
description:    Advanced Linux Sound Architecture driver for soundcards.
author:        Jaroslav Kysela <perex@suse.cz>
srcversion:    3BE4E61EB0303AD5846B8A6
depends:        soundcore
vermagic:      2.6.20-15-generic SMP mod_unload 586
```

Wie man sieht, erfolgt die Ausgabe zeilenweise sortiert nach verschiedenen Kategorien. Diese sind für alle Module gleich. Interessant ist die Zeile "depends", da man hier sieht, von welchen Modulen dieses Modul abhängt. "Abhängt" bedeutet, dass das entsprechende Modul oder ggf. auch mehrere Module geladen sein müssen, bevor dieses Modul geladen werden kann. Dies ist besonders bei Verwendung von **insmod** von Interesse.

modinfo kennt einige Optionen, welche in den **Manpages** [<https://wiki.ubuntuusers.de/man>] ausgeführt sind.

Module laden/entladen: modprobe

Mit **modprobe** können Module zur Laufzeit des Systems ge- und entladen werden. modprobe löst dabei automatisch Abhängigkeiten auf, d.h. wenn das zu ladende Modul andere Module voraussetzt, werden diese automatisch in der richtigen Reihenfolge mitgeladen bzw. werden Module automatisch mit entfernt, welche nur aus Abhängigkeitsgründen geladen wurden. Dies setzt allerdings voraus, dass die Abhängigkeiten richtig hinterlegt sind, siehe Abschnitt **depmod**.

Die allgemeine Syntax von modprobe lautet:

```
sudo modprobe OPTIONEN MODULNAME(N)
```

Die gängigsten Optionen sind :

Optionen von modprobe	
Option	Beschreibung
-v	Man erhält eine ausführliche Ausgabe der Aktionen von modprobe.
-n oder --dry-run	Simuliert das Laden des Moduls, lädt es aber nicht.
-a	Es werden alle in der Kommandozeile angegebenen Module geladen. Diese Option muss immer dann verwendet werden, wenn mehrere Module auf einmal geladen werden sollen.
--show-depends	Es werden alle Abhängigkeiten des Moduls angezeigt.
-r	die angegebenen Module werden entladen.

Eine vollständige Übersicht erhält man in den **Manpages** [<https://wiki.ubuntuusers.de/man>] zu modprobe.

Möchte man ein Modul bei jedem Systemstart laden, so muss man nicht jedes Mal modprobe ausführen,

sondern **automatisiert** den Vorgang.

insmod

Module können auch - alternativ zu **modprobe** - mit dem Befehl **insmod** geladen werden. Die Verwendung von insmod hat allerdings den Nachteil, dass eventuelle Abhängigkeiten der Module nicht automatisch aufgelöst werden. insmod lädt ein Modul nur, wenn alle Abhängigkeiten erfüllt sind bzw. vorher "von Hand" aufgelöst wurden. Daher wird insmod im täglichen Betrieb eher selten verwendet.

Die allgemeine Syntax von insmod lautet:

```
sudo insmod MODULNAME
```

rmmod

Eine zweite Möglichkeit, ein Modul aus dem Speicher zu entfernen ("entladen"), ist der Befehl **rmmod**. Im Gegensatz zu **modprobe -r** werden Module, die aus Abhängigkeitsgründen zum entfernenden Modul im Kernel geladen wurden, nicht mit entfernt.

Die allgemeine Syntax von rmmod lautet:

```
sudo rmmod OPTIONEN MODULNAME
```

Für den "normalen" Betrieb benötigt man in der Regel keine Optionen. Diese sind in den **Manpages** [<https://wiki.ubuntuusers.de/man>] zu rmmod aufgeführt.

Abhängigkeiten von Modulen: depmod

Manche Module können erst genutzt bzw. geladen werden, wenn andere Module geladen sind, da sie von diesen abhängen. Diese Abhängigkeiten sind in der Datei **/lib/modules/<Kernelversion>/modules.dep** hinterlegt. Diese Datei mit korrekt angelegten Abhängigkeiten wird bei der Installation von Ubuntu hinterlegt, so dass keine Änderungen vorgenommen werden müssen. Installiert man zusätzliche eigene Module, so sollte die Datei **modules.dep** neu erstellt werden. Dies geschieht über den Befehl **depmod**.

Die allgemeine Syntax von depmod lautet:

```
sudo depmod OPTIONEN
```

Die zwei wichtigsten Optionen sind:

Optionen von depmod	
Option	Beschreibung
-A	Es wird zuerst geprüft, ob neue Module hinzugekommen sind. Wenn nicht wird keine neue Datei erstellt.
-n	Das Erstellen der neuen Datei wird simuliert, ohne die Aktionen wirklich durchzuführen.

Die weiteren Optionen sind in der **Manpage** [<https://wiki.ubuntuusers.de/man>] von depmod aufgeführt.

Module automatisch laden

Möchte man ein Modul, welches der Kernel nicht automatisch mit lädt, beim Systemstart automatisch laden, so trägt man den Modulnamen einfach in die Datei **/etc/modules** ein. Dazu öffnet man diese Datei mit einem Editor mit Root-Rechten ^[3] und macht die entsprechenden Einträge, wobei pro Zeile nur ein Modulname eingetragen werden darf. Ab dem nächsten Systemstart werden alle Module (zusätzlich) geladen, die dort eingetragen sind.

Module mit Optionen laden

Soll ein Modul immer mit einer bestimmten Option geladen werden, kann dies in der Datei **/etc/modprobe.conf** eingestellt werden. Diese Datei existiert nicht standardmäßig - wenn sie gebraucht wird, muss sie mit Root-Rechten ^[3] angelegt werden. Es gibt nur einen Eintrag pro Zeile. Um eine Option zu setzen, beginnt man mit einem `options`:

```
options modulname_1 optionenname_1=XX
options modulname_2 optionenname_2=YY
```

Mehrere Modulooptionen setzen:

```
options modulname optionenname_a=XX optionenname_b=XX optionenname_c=XX
```

Beispiel:

```
options iwlwifi 11n_disable=1 fw_restart=1
```

Eine weitere Möglichkeit ist, eine Datei **/etc/modprobe.d/MODULNAME.conf** für ein bestimmtes Kernelmodul bzw. dessen Optionen anzulegen. Die Syntax ist die gleiche.

Über einen kombinierten **Shell** ^[https://wiki.ubuntuusers.de/Shell]-Befehl kann dies in einem Zug durchgeführt werden:

```
echo "options MODULNAME optionenname_a=XX optionenname_b=XX" | sudo tee
/etc/modprobe.d/MODULNAME.conf
```

Siehe auch **Shell/Umleitungen** ^[https://wiki.ubuntuusers.de/Shell/Umleitungen]

Beispiel:

```
echo "options iwlwifi 11n_disable=1 fw_restart=1" | sudo tee /etc/modprobe.d
/iwlwifi.conf
```

Welche Optionen das verwendete Kernelmodul bietet zeigen folgende Abfragen:

```
modinfo MODULNAME -F parm
modinfo MODULNAME | grep parm
```

Siehe auch **grep** ^[https://wiki.ubuntuusers.de/grep]

Beispiel:

```
modinfo iwlwifi -F parm  
modinfo iwlwifi | grep parm
```

Automatisches Laden verhindern - Blacklisting

Unter bestimmten Umständen kann es sinnvoll sein, das Laden von bestimmten Modulen explizit zu verbieten. Dies geschieht durch den Eintrag des entsprechenden Moduls in eine so genannte Blacklist-Datei. Diese Dateien befinden sich im Verzeichnis **/etc/modprobe.d**. In der Regel sind schon einige Einträge in der **/etc/modprobe.d/blacklist.conf** zu finden.

Die Datei legt man mit einem Editor mit Root-Rechten an ^[3] (bzw. öffnet eine vorhandene Datei) und fügt einen Dateieintrag nach folgendem Schema hinzu, je ein Eintrag pro Zeile:

```
blacklist modulname_1  
blacklist modulname_2
```

Die Sperrlisten haben üblicherweise den Namen **blacklist-Modulname.conf**, d.h. der Dateiname kann frei gewählt werden, sollte aber mit dem Präfix **blacklist-** beginnen um direkt über die Funktion zu informieren. Darüber hinaus existiert auch eine "allgemeine" Datei, die nur den Namen **blacklist.conf** trägt.

Auch hier kann dies über einen kombinierten **Shell** [\[https://wiki.ubuntuusers.de/Shell\]](https://wiki.ubuntuusers.de/Shell)-Befehl in einem Zug durchgeführt werden:

Eintrag an bestehende Sperrliste **/etc/modprobe.d/blacklist.conf** anhängen:

```
echo "blacklist prism54" | sudo tee -a /etc/modprobe.d/blacklist.conf
```

Neue Sperrliste anlegen bzw. bestehende überschreiben:

```
echo "blacklist prism54" | sudo tee /etc/modprobe.d/blacklist_prism54.conf
```

Siehe auch **Shell/Umleitungen** [\[https://wiki.ubuntuusers.de/Shell/Umleitungen\]](https://wiki.ubuntuusers.de/Shell/Umleitungen)

Sollte ein benötigtes Modul nicht automatisch bei Systemstart geladen werden, so kann dies an einem entsprechenden Sperreintrag liegen. Bestehende Sperrlisten nach einem Eintrag mittels **grep** [\[https://wiki.ubuntuusers.de/grep\]](https://wiki.ubuntuusers.de/grep) durchsuchen:

```
grep -i MODULNAME /etc/modprobe.d/*
```

Beispiel:

```
grep -i prism54 /etc/modprobe.d/*
```

Entsprechende Einträge wie oben gezeigt mittels **grep** [\[https://wiki.ubuntuusers.de/grep\]](https://wiki.ubuntuusers.de/grep) suchen und über **sed** [\[https://wiki.ubuntuusers.de/sed\]](https://wiki.ubuntuusers.de/sed) auskommentieren, Modul also wieder entsperren:

```
sudo sed -i "s/blacklist MODULNAME/#blacklist MODULNAME/g" $(egrep -lo  
'blacklist MODULNAME' /etc/modprobe.d/*)
```

Beispiel:

```
sudo sed -i "s/blacklist prism54/#blacklist prism54/g" $(egrep -lo 'blacklist prism54' /etc/modprobe.d/*)
```

Hinweis:

Manchmal kann es erforderlich sein, das **initramfs** mit dem Befehl `update-initramfs -u` zu aktualisieren, wenn die Module trotz Blacklisting nach einem Neustart wieder geladen werden.

Übersicht über alle Module

Wer wissen möchte, welche Module auf dem eigenen Rechner verfügbar (nicht geladen!) sind, der kann sich mit dem Befehl ^[2]

```
basename -s ".ko" $(find /lib/modules/${uname -r} -type f -name "*.ko")
```

eine komplette Liste anzeigen lassen. Allerdings ist die Liste extrem lang (der Kernel bringt mehrere hundert Module mit...), daher ist es eventuell günstiger, die Modulverzeichnisse mit einem Dateimanager zu durchforsten. Die Module liegen alle in den Unterverzeichnissen von **/lib/modules/KERNELVERSION/kernel/**

Links

- **Module-Assistant** [https://wiki.ubuntuusers.de/module-assistant] - Module nachträglich erstellen und installieren
- **DKMS** [https://wiki.ubuntuusers.de/DKMS] - Module automatisch aktualisieren

Diese Revision [https://wiki.ubuntuusers.de/Kernelmodule?rev=808733] wurde am 18. April 2015 10:03 von **elektronenblitz63** erstellt.

Die folgenden Schlagworte wurden dem Artikel zugewiesen: **Hardware** [https://wiki.ubuntuusers.de/Wiki/Tags?tag=Hardware], **System** [https://wiki.ubuntuusers.de/Wiki/Tags?tag=System]

Inhalte von ubuntuusers.de lizenziert unter Creative Commons, siehe <https://ubuntuusers.de/lizenz/>.