

# Hardware Schnittstellen anzeigen

```
ls /sys/bus/*/*
```

## USB

```
lsusb  
lsusb -v
```

## PCI

```
lspci  
lspci -v
```

## SCSI

```
ls SCSI
```

## Kernel-Module

Modulverzeichnis:

```
ls /lib/modules/$(uname -r)/
```

## Befehle

### Kernel Meldungen anzeigen

```
dmesg
```

### Kernel Module anzeigen

```
lsmod
```

## Informationen und mögliche Parameter zu einem Modul anzeigen

Informationen zum Modul „dummy“ anzeigen

```
modinfo dummy
```

## Modul von Hand laden

Modul „dummy“ laden

```
modprobe dummy
```

## Modul entladen

Kernel-Modul „dummy“ entladen:

```
modprobe -r dummy
```

Wenn das nicht geht, weil das Modul in Benutzung ist, dann kann man

```
rmmod -w dummy
```

<sup>1)</sup>

## Konfigurations-Dateien

### Module nicht automatisch laden

Beispiel: automatisches laden des Moduls *ipv6* verhindern

`/etc/modprobe.conf` :

```
alias ipv6 off
```

### Module beim Starten laden

Beispiel: Modul *dummy* laden

### Debian

`/etc/modules`

```
dummy
```

## SuSE

```
/etc/sysconfig/kernel :
```

```
MODULES_LOADED_ON_BOOT="dummy"
```

## RedHat

```
/etc/rc.modules :
```

```
modprobe dummy
```

```
chmod +x /etc/rc.modules
```

## Modul-Optionen setzen

Beispiel: 2 Dummy-Interfaces erstellen

### einmalig

```
modprobe dummy numdummies=2
```

### dauerhaft

```
/etc/modprobe.d/dummy_params :2)
```

```
options dummy numdummies=2
```

## Fehlendes Kernel-Modul nachträglich installieren

### openSuSE 11.1

```
zypper search kmp
```

### Debian

Verfügbare Pakete anzeigen:

```
aptitude search modules
```

( Vorsicht: nicht alle angezeigten Pakete enthalten *kernel*-Module )

Beispiel WLAN Treiber ipw3945

```
aptitude install ipw3945-modules-2.6-686 ipw3945d firmware-ipw3945  
modprobe ipw3945
```

## CentOS 5

Verfügbare Pakete anzeigen:

```
yum list | grep kmod
```

### Modul über ELRepo nachinstallieren

ELRepo ist ein Repository mit Fokus auf hardware-nahen Paketen

Repository [ELRepo](#) hinzufügen (siehe auch [yum](#)):

Verfügbare Pakete anzeigen:

```
yum search kmod
```

### Modul installieren

```
yum install kmod-ntfs
```

### testen

```
modprobe ntfs  
dmesg | tail
```

## Fehlendes Kernel-Modul selbst bauen

Problem: es gibt in der verwendeten Distribution keinen Treiber für folgendes Gerät:

```
lsusb
```

```
Bus 001 Device 005: ID 0a46:6688 Davicom Semiconductor, Inc.
```

## als Nutzer

Lösung: jagen & sammeln & Modul selbst kompilieren: Auf der Seite <http://alcopop.org/unix/linux/dm9601/> sind zwei passende Treiber verlinkt:

```
wget http://www.silencio.ro/DM9601.GZ
```

Entpacken und hineinwechseln:

```
tar xzf DM9601.GZ  
cd 2.6.18/
```

USB-ID hinzufügen:

```
vi dm9601.h
```

```
DM9601_DEV( "Davicom USB-100", 0x0a46, 0x6688, DEFAULT_GPIO_RESET )
```

Kompilieren:

```
make
```

## Als root

Modul laden

```
insmod /tmp/2.6.18/dm9601.ko
```

Testen ob Gerät erscheint:

```
dmesg  
ifconfig -a
```

Aufräumen:

```
rmmmod dm9601
```

Gerät entfernen

Modul installieren:

```
cp /tmp/2.6.18/dm9601.ko /lib/modules/`uname -r`/kernel/drivers/usb/net  
depmod -a
```

## Doku

## CentOS / Redhat

- <http://wiki.centos.org/HowTos/BuildingKernelModules>

## Eigenes Kernel-Modul schreiben

Keine Angst, nur ein ganz einfaches „Hello World“ Beispiel ...

### Debian

```
cd
aptitude install linux-source-xxx
tar xjf /usr/src/linux-source-xxx
mkdir hello
vi hello.c
```

```
#include <linux/init.h>
#include <linux/module.h>
#include <linux/kernel.h>

static int hello_init(void)
{
    printk(KERN_ALERT "Hello World!\n");
    return 0;
}

static void hello_exit(void)
{
    printk(KERN_ALERT "Good bye world!\n");
}

module_init(hello_init);
module_exit(hello_exit);

MODULE_LICENSE("GPL");
MODULE_AUTHOR("me");
```

```
vi Makefile
```

```
obj-m := hello.o
```

```
make -C ../linux-source-2.6.18/ SUBDIRS=$PWD modules
```

# Neuen Kernel installieren

## Debian 6.0

[Repository](#) hinzufügen, z.B. Debian Backports

```
apt-get -t squeeze-backports install linux-image
```

Passende Version auswählen, z.B.

```
apt-get -t squeeze-backports install linux-image-2.6.39-bpo.2-686-pae
```

## Einstellungen zur Laufzeit

Mögliche Einstellungen sind in der Kerneleinstellung in der Datei [Documentation/filesystems/proc.txt](#) beschrieben. Hier ein paar Beispiele:

Auf sehr hoch belasteten Servern kann es Sinn machen, die Anzahl der gleichzeitig geöffneten Dateien zu erhöhen:

`/etc/sysctl.conf` :

```
fs.file-max = 16384
kernel.threads-max = 2048
```

```
sysctl -p
```

Wenn das System nicht mehr reagiert ( z.B. überhohe Last, X Absturz ) gibt es als letzte Rettung oft nur die „magischen“ Tastenkombinationen aus ALTGr + SysRq + Buchstabe <sup>3)</sup>:

Buchstabe	Funktion
h	Hilfe zu den SysRq Keys anzeigen
e	Allen Prozessen ( außer init ) das Signal TERM schicken
i	Allen Prozessen ( außer init ) das Signal KILL schicken
s	Alle gemounteten Dateisysteme syncen
u	Alle Dateisysteme umounten
b	System_sofort_rebooten
o	System_sofort_ausschalten

```
sysctl -w 'kernel.sysrq=1'
```

Diese Einstellungen rebootfest machen:

`/etc/sysctl.conf` :

```
kernel.sysrq = 1
```

## Kernel Bootparameter

Eine Übersicht über die Parameter, die der Kernel beim Booten auswertet findet sich in der Datei `Documentation/kernel-parameters.txt` in den Kernel-Quelltexten.

Kernel Bootparameter können beim Booten oder über die Konfigurationsdatei des [Bootloaders](#) übergeben werden.

## Kernel Compile-parameter

Auf den meisten Systemen ist es nicht notwendig, den Kernel neu zu kompilieren. Es macht Unerfahrenen mehr Ärger als es bringt. Trotzdem ist es interessant, sich die Einstellungen anzusehen, die auf Kernel-Ebene möglich sind:

<http://kernel.xc.net/>

## Dokus & Links

- [http://www.linuxtopia.org/online\\_books/linux\\_kernel\\_index.html](http://www.linuxtopia.org/online_books/linux_kernel_index.html)

<sup>1)</sup>

wenn auch das nicht geht, kann man noch

```
rmmod -f dummy
```

probieren.

<sup>2)</sup>

veraltet: `/etc/modprobe.conf` :

<sup>3)</sup>

Die vollständige Liste der Befehle steht in der Kerneldokumentation in der Datei [Documentation/sysrq.txt](#)

From:

<https://wiki.lab.linuxhotel.de/> - **Linuxhotel Wiki**

Permanent link:

<https://wiki.lab.linuxhotel.de/doku.php/lpi1:kernel?rev=1386707652>

Last update: **2013/12/10 20:34**

