Treiber & Hardware für Linux

Gängige Linux-Distributionen bringen Treiberfür fast jede Hardware mit. Sollte ein Gerät nicht voll ständig unterstützt werden oder gar nicht funktionieren, gibt es auch dafür Lösungen.

Computerhardware wird vor allem für den Windows-Markt produziert. Vom Gerätehersteller können Linux-Nutzer kaum Unterstützung erwarten und zusätzliche Treiber für Linux gibt es in der Regel nicht als Setup-Paket. Es ist daher ratsam, sich bereits vor dem Kauf neuer Hardware über die Linux-Tauglichkeit zu informieren. Auf älteren PCs oder Notebooks lässt sich Linux meist hne besondere Auffälligkeiten installieren. Aber auch hier kann es in Ausnahmefällen vorkommen, dass der Bildschirm schwarz leibt, einzelne Funktionen unter Linux nicht bereitstehen oder neuere Hardware am USB-Anschluss nicht erkannt wird. Einige Probleme lassen sich durch Konfigurationsänderungen oder durch Wechsel der Linux-Distribution beheben. Am Anfang steht jedoch die Untersuchung der Hardware. Die kann ergeben, dass sich das Gerät mit einem neueren Kernel oder einem zusätzlichen Treiber nutzen lässt.

Hardwaremacken umgehen

PC-Hardware folgt keinen eindeutigen Standards. Für die Hersteller ist nur eins
24 wichtig: Windows muss anstandslos laufen. Der eine oder andere Fehler in der Firmware oder auf der Hauptplatine wird dann bei Bedarf über einen Treiber repariert, den es für Linux aber nicht gibt. Besonders Notebooks, die auf Microsoft Windows zugeschnitten sind, bereiten mit ihren zahlreichen Bios-Versionen, abweichenden ACPI-Stromsparfunktionen und Chipsatzvarianten häufiger Ärger. Mal bleibt der Bildschirm dunkel, mal geht nach den ersten Bootmeldungen nichts mehr weiter. Bei der Linux-Neuinstallation helfen dann oft spezielle Kernel-Parameter weiter. Wenn Sie Ubuntu 22.04 oder Linux Mint 21 vom Installationsmedium booten, begrüßt Sie das Grub-Bootmenü. Mit dem ersten Eintrag „Try or Install Ubuntu“ beziehungsweise „Start Linux Mint 21 Cinnamon 64-Bit“ startet das System mit den Standardeinstellungen. Sollte der Bildschirm schwarz bleiben oder unlesbar sein, kann der zweite Menüeintrag „Ubuntu (safe graphics)” oder „Start Linux Mint 21 Cinnamon 64-Bit (compatibility mode)“ weiterhelfen. Dabei werden Optionen an den Kernel übergeben, die die Hardwaresteuerung beeinflussen. Jedes Linux-System bietet eine Reihe von Kernel-Optionen - in der Dokumentation oft auch als Bootparameter oder Cheatcodes bezeichnet.
Über das Grub-Bootmenü lassen sich zusätzliche Optionen angeben. Dazu drückt man nach der Markierung eines Booteintrags die Taste E und erhält dann einen Mini-Texteditor für den jeweiligen Eintrag gezeigt. Die Navigation im Textfeld erfolgt mit den Cursortasten. Grundsätzlich müssen Kernel-Parameter in die Zeile eingetragen werden, die mit „linux“ beginnt, aber vor „--"”. Nach der Änderung startet die Taste F10 den Booteintrag mit den neuen Einstellungen.
Bitte beachten Sie, dass im Grub-Editor eine englischsprachige Tastaturbelegung gilt. Bleibt der Bildschirm nach einem zunächst erfolgreichen Start dunkel, so lieg dies meist an nicht ausreichend

unterstützten Grafikchips. Folgende Optionen können in diesem Fall weiterhelfen.

xforcevesa: Bei der Angabe dieser Option nutzt der Kernel für die Anzeige der grafischen Oberfläche nur den Vesa-Modus. Dieser Modus läuft auf den meisten Grafikchips, ohne jedoch deren spezielle Merkmale wie Hardwarebeschleunigung und Fähigkeiten zu nutzen.

nomodeset: Aktuelle Linux-Kernel können den Bildschirmmodus auf eigene Faust wechseln und schalten schon während des Bootvorgangs in einen grafischen Modus. Dies funktioniert nicht bei allen Grafikchips - so haben einige Modelle von Nvidia Probleme damit. Mit „nomodeset“ verzichtet der Kernel auf den Wechsel in den Grafikmodus und bleibt bei purem Text. Ubuntu und Linux Mint verwenden diese Option bei Auswahl des zweiten Menüeintrags. Auf einigen Notebooks verhindern inkompatible Stromsparfunktionen im ACPI (Advanced Configuration and Power Interface den Linux-Start. Linux Mint verwendet beim Menüeintrag „Start Linux Mint 21 Cinnamon 64-Bit (compatibility mode)“ einige der nachfolgend genannten Optionen.

acpi=off oder **noacpi** zwingt Linux dazu, ACPI komplett zu ignorieren und damit ohne Stromsparfunktionen und Leistungsmanagement für CPU und GPU zu starten. Auch Hyperthreading und die Lüfterregelung sind abgeschaltet.

acpi=ht: Mit dieser Option beachtet der Linux-Kernel gerade mal so viele ACPI-Fähigkeiten der Hardware, dass Hyperthreading
der CPU funktioniert. Andere Stromsparfunktionen bleiben dagegen deaktiviert.

acpi=strict weist die ACPI-Unterstützung des Kernels an, nur ACPI-Merkmale der vorhandenen Hardware zu beachten, die
exakt dem Standard folgen. Auf problematischen Notebooks ist diese Option immer einen Versuch wert.

acpi_osi=linux umgeht die Abfrage des Linux-Kernels, ob das ACPI eines Rechners kompatibel ist. Sinnvoll ist dieser Parameter, wenn einige Stromsparfunktionen nicht verfügbar sind oder die Drehzahlsteuerung der Lüfter nicht funktioniert. Eine weitere Komponente nennt sich „Local APIC” und nimmt die Interrupt-Anforderungen auf jedem Prozessorkern entgegen. Der Parameter „nolapic“ löst vielfältige Probleme mit heiklen Bios-Versionen, reduziert aber in jedem Fall die Zahl der vorhandenen CPU-Kerne auf einen. Geeignet ist dies nur als erste Hilfe, bis ein Bios-Update oder eine neue Kernel-Version echte Abhilfe schafft.

noapic: Verhindert, dass APIC für die Auflösung von Hardwarekonflikten auf Interrupt-Ebene verwendet wird. Der Parameter hilft auf Systemen mit einem unverträglichen Bios und inkompatiblen ACPI-Funktionen weiter. Eine häufig erfolgreiche Kombination bei besonders widerspenstigen Notebooks ist diese: `acpi=off noapic nolapic`

iommussoft: Die „Input-Output Memory Management Unit“ (IOMMU) ist ein Merkmal einiger Hauptplatinen und erlaubt Peripheriegeräten den direkten Speicherzugriff. Dies funktioniert zusammen mit Linux nicht immer, was zum Ausfall von USB-Ports oder Netzwerkchip führt. Dieser Parameter aktiviert zusammen mit abgeschaltetem IOMMU im Bios/Uefi, ein softwaremäßiges IOMMU.
Für Ubuntu und Co. liefert die englischsprachige Hilfeseite

https://help.ubuntu.com/community/BootOptions eine Übersicht geläufiger Bootparameter. Die komplette Liste der Parameter mit Beschreibung für Entwickler bietet die offizielle Kernel-Dokumentation unter https://www.kernel.org/doc/htmi/latest/admin-guide/kernel-parameters.html

Nach der Linux-Installation: Für ein installiertes System lassen sich die gewünschten Kernel-Optionen via Bootloader genauso angeben wie im Live- und Installationssystem. Damit eine Änderung permanent gilt, ist die Bearbeitung einer Konfigurationsdatei mit sudo-Recht nötig:
`sudo -H gedit /etc/default/grub`
Die Zeile

```
GRUB_CMDLINE_
LINUX=" [parameter1]=[wert1]
    [parameter2]=[wert2]"
```

definiert die manuell hinzugefügten Angaben. Stehen hier schon Optionen, so ergänzen Sie zusätzlich nötigen nach einem Leerzeichen in dieser Zeile. Nach Änderung und Sicherung der Konfigurationsdatei ist die Änderung aber noch nicht wirksam, denn erst muss noch der Bootloader mit diesem Terminalbefehl aktualisiert werden:
`sudo update-grub`
Erst danach startet das Linux-System standardmäßig mit den hinzugefügten Kernel-Optionen.

> Kein Treiber: Hardware Infos findet man bei https://linux-hardware.org.
>
> Kein Treiber: Hardware Infos findet man bei https://linux-hardware.org. Für das Gerät mit der ID "OBda:b812" ist bis einschließlich Version 5.19 kein Kernel-Modul vorhanden

Hardware untersuchen und Infos finden

Jedes interne und externe Gerät besitzt eine eindeutige ID, die über den Hersteller und das Gerät informiert. Diese IDs lassen sich im Terminal mit den folgenden drei Befehlszeilen auslesen und in einer Datei speichern:
`sudo lshw -numeric -html > lshw.html`
`sudo lspci -nn > lspci.txt`
`sudo lsusb -v > lsusb.txt`

In der Datei „lshw.html“ finden Sie allgemeine Informationen zum PC, etwa den Typ der Hauptplatine, die Firmwareversion und den Prozessortyp. „lspci.txt” zeigt Informationen über per PCI (Peripheral Component Interconnect) angebundene Komponenten, beispielsweise Soundchips („Audio device“), Grafikkarten („VGA compatible controller“) und Netzwerkchips („Ethernet controller“). In der Datei „lsusb.txt” sehen Sie, welche Geräte mit den USB-Ports verbunden sind.

Ein Beispiel: Mit lsusb finden Sie die ID „Obda:b812“. Der erste Teil „Obda“ verweist auf den Hersteller „Realtek Semiconductor Corp.” (siehe USB ID Repository: https://usb-ids.gowdy.us/read/UD), der zweite Teil „6812“ kennzeichnet das Gerät. Eine Internetsuche nach

der kompletten ID liefert einige Ergebnisse, darunter auch https;//linux-hardware.org/?id=usb:0bda-b812. Hier erfährt man, dass bis einschließlich Kernel 5.19 kein Treiber für diesen WLAN-Stick mit dem Chipsatz „RTL88x2bu“ verfügbar ist. In der Liste unter „Status“ sind einige Analysen zu finden, die Nutzer an https://linux-hardware.org gesendet haben. Bei den meisten steht „failed“- das Gerät ließ sich also nicht in Betrieb nehmen. Die ersten Einträge mit „works“ weisen darauf hin, dass ein Treiber von https://github.com/morrownr/88x2bu für diese Hardware erforderlich ist.

> image.png
>
> Problemlösung: Nach einsendung der Hardwareanalyse zeigt https://linuxhardware.org an, welches Gerät nicht funktioniert und was dagegen zu tun ist.

Analysedaten einsenden: Statt nach der Geräte-ID zu suchen, kann man die Daten seines PCs selbst anonym bei https://linux-hardware.org hochladen und dort prüfen lassen. Nutzer von Ubuntu oder Linux Mint installieren das nötige Tool im Terminal:
sudo apt install hw-probe und starten es so: **sudo -E hw-probe -all -upload** Sie erhalten eine URL, die Sie im Browser aufrufen. In der Übersicht unter „Devices“ sind dann alle Geräte aufgelistet. Steht in der Spalte „Status“ der Eintrag „works“ oder „detected“, sollte das Gerät funktionieren. Ausrufungszeichen deuten jeweils darauf hin, dass es bei einzelnen Benutzern Probleme gab. Ein Klick darauf führt zu weiteren Informationen. Beim Status „failed“ wurde kein Treiber gefunden, geladen oder konfiguriert. Sofern vorhanden, führt ein Kasten rechts daneben zur Lösung. Weiter unten auf der Seite unter „Logs“ kann man sich per Klick etwa auf „Hwinfo“, „Lspci” oder „Lsusb“ die Ausgaben der jeweiligen Tools anzeigen lassen.

Neue Kernel oder Treiber

Die Infos von https://linux-hardware.org können ergeben, dass ein neuerer Kernel die Hardware unterstützt. Welchen Kernel Ihr System aktuell nutzt, finden Sie im Terminal mit dem Befehl `uname -a`
heraus. Ubuntu 22.04 und Linux Mint 21 verwenden zur Zeit die Kernel-Version 5.15. Wer einen neueren Kernel benötigt, muss ihn selbst erstellen. Eine Anleitung für Ubuntu und Linux Mint finden Sie über https;//m6u.de/BMUK. Alternativ kann man zu einer
Distribution wechseln, die deutliche aktuellere Updates bietet („Rolling Release”). Manjaro Linux beispielsweise ist aktuell schon bei Kernel 6.0.2 (https://manjaro.org).

Treiber kompilieren: Wenn der Quellcode verfügbar ist, kann man den Treiber selbst erstellen. Dabei gibt es zwei Herausforderungen: Der Quellcode muss zum laufenden Kernel passen und die nötigen Schritte können sich je nach Treiber unterscheiden Lesen Sie daher immer die zugehörigen Anleitungen sorgfältig durch. Exemplarisch liefern wir eine kurze Anleitung für den oben genannten WLAN-Stick mit dem Chipsatz „RTLESX2bu”.

Schritt 1: Öffnen Sie ein Terminal und installieren Sie einige zusätzliche Pakete:
`sudo apt install git build-essential dkms linux-headers-$(uname -r)`

Schritt 2: Gehen Sie auf https://github.com/morrownr/88x2bu-20210702 und klicken Sie auf „Code“. Kopieren Sie die URL unter „HTTPS". Dann erstellen Sie ein Arbeitsverzeichnis und laden den Quelltext des Treibers herunter (vier Zeilen):

```
mkdir -p ~/src
cd ~/src
git clone [URL]
cd 88x2bu-20210702
```

Statt „[URL]* fügen Sie die zuvor kopierte Adresse ein. Passen Sie die Pfadangabe hinter „cd“ entsprechen der Treiberversion an.

Schritt 3: Führen Sie die folgenden drei Befehle aus:

```
make clean
make
sudo insmod 88x2bu.ko
```

> image.png
> Treiber kompilieren: Nach dem Download des Treiber-Quellcodes genügt in der Regel der Befehl "make", der ein neues Modul für den laufenden Kernel erstellt.

Damit wird der Treiber aus dem aktuellen Ordner „-/src/88x2bu-20210702" geladen Mittels `dmesg` lassen Sie sich die letzten Kernel-Meldungen ausgeben, die über den erfolgreich geladenen Treiber informieren oder Fehlermeldungen anzeigen. Für die meisten Treiber folgt abschließend dieser Befehl:

```
sudo make install
```

Damit wird der Treiber in einen Ordner unterhalb von „/lib/modules/[Kernel-Version] kopiert und ab dem nächsten Linux-Start automatisch geladen. Bei unserem Beispieltreiber ist das nicht erforderlich, weil ein Script die Aufgabe übernimmt (siehe Schritt 4). Die WLAN-Hardware funktioniert jetzt und Verbindungen lassen sich wie gewohnt über den Netzwerkmanager herstellen.

Wenn allerdings Secure Boot aktiviert ist, erzeugt der Befehl insmod eine Fehlermeldung: Der Treiber wird nicht geladen, weil er nicht digital signiert ist. Sie müssen dann entweder Secure Boot im Firmwaresetup deaktivieren oder den Treiber signieren (siehe https://m6u.de/KNMD).

Schritt 4: Der Treiber von https;//github.com/morrownr/88x2bu-20210702 lässt sich komfortabel per Script einrichten:

```
sudo ./install-driver.sh
```

Sie werden gefragt, ob Sie die Konfigurationsdatei „/etc/modprobe.d/88x2bu.conf” mit den Treiberoptionen bearbeiten möchten. In der Regel ist das nicht nötig, Erklärungen zu den Optionen sind in der Konfigurationsdatei enthalten. Wenn Sie Linux jetzt neu starten, wird der neue Treiber automatisch geladen und die WLAN-Hardware lässt sich nutzen.

Revision #1
Created 21 March 2023 08:04:03 by Hermann
Updated 21 March 2023 08:04:25 by Hermann