

Wiki / Live-USB - persistente Installation

[ubuntuusers.de](#) » [Wiki](#) » [Live-USB - persistente Installation](#)

Live-USB - persistente Installation

Ausbaufähige Anleitung

Dieser Anleitung fehlen noch einige Informationen. Wenn Du etwas verbessern kannst, dann editiere den Beitrag, um die Qualität des Wikis noch weiter zu verbessern.

Anmerkung: Die Menüeinträge müssen bezüglich `initrd.gz` angepasst werden, ab 9.10 gilt `initrd.lz`


Dieser Artikel wurde für die folgenden Ubuntu-Versionen getestet:

- [Ubuntu Lucid Lynx 10.04](#)
- [Ubuntu Karmic Koala 9.10](#)
- [Ubuntu Hardy Heron 8.04](#)

Zum Verständnis dieses Artikels sind folgende Seiten hilfreich:

1. [Installation von Programmen](#)
2. [Im Terminal arbeiten](#)
3. [Mit dem Texteditor arbeiten](#)
4. [Archive entpacken](#)
5. [Rechte für Dateien und Ordner ändern](#)
6. [Ein Programm starten](#)



Dieser Artikel beschreibt, wie man eine (K)XUbuntu [Desktop-CD](#) auf einen [USB-Stick](#)  kopiert und derart einrichtet, dass er anschließend wie ein normales [Live-System](#) verwendet werden kann. Zusätzlich wird erklärt, wie man dieses Live-USB-System um eine persistente Installation erweitert, womit das Arbeiten wie mit einem normal installierten Linux möglich ist. „Persistent“ bedeutet in diesem Zusammenhang, dass auf einer zweiten, speziell gekennzeichneten Partition sämtliche veränderten Einstellungen des Live-Systems und vom Nutzer veränderte bzw. hinzugefügte Dateien gespeichert werden.

Die Ubuntu-Installation auf einem USB-Stick ist von Nutzen, wenn aus diversen Gründen die Verwendung eines optischen Laufwerks nicht möglich oder nicht erwünscht ist. Zudem erhält man mit dem persistenten Modus eine vollständig anpassbare Arbeitsumgebung, welche an jedem Computer mit der Fähigkeit von USB zu starten zur Verfügung steht.

Auf die benötigten Schritte für eine persistente Installation wird im Folgenden stets deutlich hingewiesen. Wenn nur ein normales Live-USB Systems installiert werden soll, können diese Schritte folglich ausgelassen werden.

Voraussetzungen

Folgendes wird benötigt:

- Ein Rechner, welcher das Booten von USB-Sticks unterstützt (Im Handbuch oder auf der Hersteller-Webseite nachzulesen)
- Ein (leerer) startfähiger USB-Stick mit mindestens 1GB Fassungsvermögen – ein Minimum von 2GB ist angeraten
- Ein laufendes Linux-System
- Eine [Live-CD](#) oder ein [CD-Image](#) (Desktop-Edition) von Ubuntu, Kubuntu oder Xubuntu

Vorbereitung

Da im Zusammenhang mit dem "Starten von einem USB-Stick" häufig von Problemen berichtet wird, ist es

Inhaltsverzeichnis

1. [Voraussetzungen](#)
2. [Vorbereitung](#)
3. [Manuell oder automatisch?](#)
4. [Partitionierung und Formatierung des USB-S...](#)
 1. [Variante 1: Grafisches Partitionierungsp...](#)
 2. [Variante 2: Konsole](#)
5. [CD-Inhalt kopieren](#)
 1. [Zusätzliche Arbeiten für Persistenz unte...](#)
6. [Bootloader Installation](#)
 1. [Variante 1: GRUB \(GRUB2\)](#)
 2. [Variante 2: Syslinux](#)
7. [Ballast abwerfen](#)
 1. [Inhalte für Windows](#)
 2. [Weiteren Platz einsparen](#)
8. [Problemlösungen](#)
 1. [Bootprobleme](#)
 2. [Probleme nach dem Bootmenü](#)
9. [Links und Quellen](#)
 1. [Quellen](#)
 2. [Weitere Links](#)

ratsam, vor Beginn der Arbeiten sicher zu stellen, dass diese Funktionalität gegeben ist. Falls nicht bekannt ist, ob dem so ist, sollte an dieser Stelle der am Ende des Artikels befindliche Abschnitt [Bootprobleme](#) konsultiert werden. Besonders von Interesse dürfte hierzu der Abschnitt [USB-Stick auf Bootfähigkeit prüfen](#) sein.



Zunächst sollte man die interne Bezeichnung des USB-Sticks in Erfahrung bringen. Dazu kann man das systemeigene [Partitionierungsprogramm](#) nutzen und dort nachschauen, oder man öffnet einfach ein Terminal ^[2] und führt den folgenden Befehl aus (benötigt Root-Rechte!):

```
sudo fdisk -l
```

In der darauf folgenden Ausgabe bzw. in der Oberfläche des [Partitionierungsprogramms](#) sollte man seinen Stick anhand seiner Bezeichnung (falls vorhanden) oder der Größe identifizieren können. In den folgenden Beispielen dieses Artikels wird die Adresse des USB-Sticks immer mit **/dev/sdX** angegeben. Das **X** ist hierbei durch den jeweilig zutreffenden Buchstaben zu ersetzen. Hierbei kann auch gleich festgestellt werden, ob sich eine oder mehrere Partitionen auf dem USB-Stick befinden. *Dies ist im Folgenden von besonderem Interesse.*

Achtung!

Bei der folgenden Prozedur werden alle auf dem USB-Stick befindlichen Daten vernichtet! Wenn noch nicht erfolgt, sollte jetzt eine Sicherung aller evtl. vorhandenen Daten erstellt werden.

Manuell oder automatisch?

Seit Ubuntu 9.10 Karmic Koala gibt es zum automatisierten Erstellen eines persistenten Live-USB-Sticks das Pythonprogramm `usb-creator`. Es ist standardmäßig vorinstalliert und zu finden im Menü System - Systemverwaltung - USB-Startmedien-Ersteller (KDE: K-Menü - System - USB Startup Creator).

Zusätzlich zum `usb-creator` und zur individuellen, händischen Bearbeitung, existiert die Möglichkeit die meisten Arbeiten von einem Script erledigen zu lassen, welches von einem Mitglied der ubuntuusers-Gemeinschaft erstellt wurde.

Mit dem Live-USB-Script kann ein kompletter, bootfähiger Live-USB-Stick mit Persistentfähigkeit erstellt werden. Das Script partitioniert den Stick, formatiert ihn und kopiert alle benötigten Daten. Dafür benötigt das Script nur drei Informationen:

1. Die Adresse des USB Sticks - also **/dev/sdX**
2. Den Pfad zum CD-Inhalt
3. Wie groß die Persistent-Partion werden soll (0 für keine Persistentfähigkeit, 1 für Rest des Speicherplatzes, andere Werte werden als Angaben in Mebibyte betrachtet)

Alle Schritte sind innerhalb des Scripts kommentiert und mit Beispielen versehen. Eingaben können auf Wunsch gespeichert werden und ermöglichen so bei erneuter Nutzung einen automatischen Ablauf mit einem Klick.

Getestet ist das Script bis jetzt mit Ubuntu Hardy. Fragen, Kritik oder Bugs können in [diesem Thread](#) gepostet werden.

[Download Live-USB-Script](#) ↪

Hinweis:

Der Persistent-Modus funktioniert unter Hardy Heron nicht "ohne Weiteres". Der Bug wird voraussichtlich erst im zweiten Point Release (8.04.2) behoben sein.

Partitionierung und Formatierung des USB-Sticks

Zunächst muss sichergestellt werden, dass die eventuell noch bestehenden Partitionen des Sticks nicht mehr eingebunden sind. Dies ist wieder am einfachsten durch einen Konsolenbefehl ^[3] zu erreichen:

```
sudo umount /dev/sdX1
```

Falls sich mehr als eine Partition auf dem Stick befindet, muss der Befehl natürlich mit der entsprechend höheren Nummer wiederholt werden. (**/dev/sdX2** usw.)

Variante 1: Grafisches Partitionierungsprogramm

GPARTED Hauptfenster

GPARTED Boot-Flag

Wenn nicht schon geschehen, ist jetzt das [Partitionierungsprogramm](#) der jeweiligen Ubuntu-Variante zu öffnen. Dort sucht man zunächst im Menü den USB-Stick und wechselt zu diesem.

Nun sind alle auf dem USB-Stick eventuell noch vorhandenen Partitionen zu entfernen. Anschließend kann mit der Neupartitionierung begonnen werden.

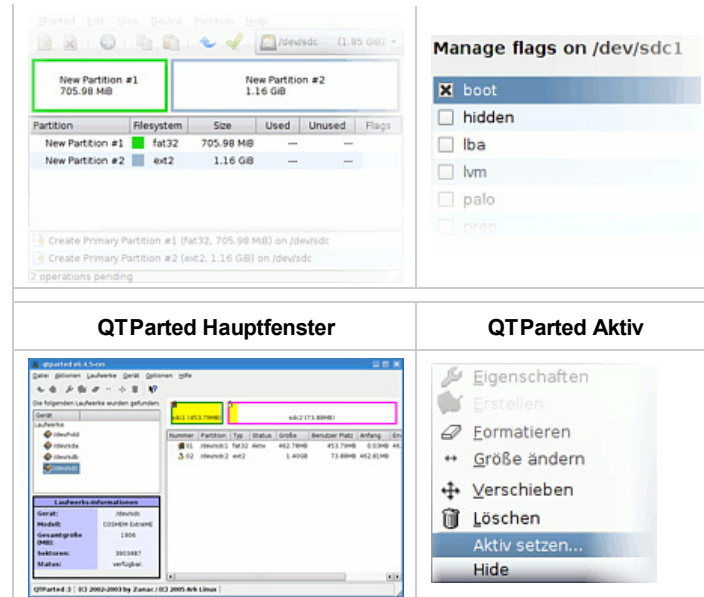
Als erstes muss eine Partition für die Daten der Desktop-CD angelegt werden. Dabei ist das [Dateisystem](#) zunächst egal, da später, wenn die Partitionen eine Bezeichnung erhalten sollen, sowieso nochmals formatiert werden muss. Verwendet man [Kubuntu/Ubuntu](#), werden hier 700 [Mebibyte](#) benötigt, für [Xubuntu](#) 570 Mebibyte. Strebt man eine Installation ohne Persistenz an, ist einem selbst überlassen wie der restliche Platz verteilt wird.

Nun muss für diese Partition noch das Bootflag gesetzt werden (in anderen Worten: die Partition wird als *Startfähig* markiert). Setzt man den besagten Status nicht und versucht in diesem Zustand trotzdem zu starten, wird man meistens eine Fehlermeldung erhalten.

Für die Installation im persistenten Modus benötigt man eine zweite Partition. Ihr sollte eine Größe von ca. 1 Gigabyte zugeteilt werden, bei USB-Sticks bis 2 GB am besten der ganze Rest. Dies sei angeraten, weil hier später auch die Daten des [Home-Verzeichnisses](#) abgelegt werden.

Ganz ohne Konsolenbefehle ^[3] geht es auch in der grafischen Variante nicht. Der Grund dafür ist, dass die Partitionen noch Bezeichnungen erhalten sollen respektive müssen und zu diesem Zweck der Formatierungsbefehl für die Konsole verwendet wird.

An der folgenden Stelle geht es weiter: [Formatierung](#)



Variante 2: Konsole

Partitionierung

Zum Entfernen alter Partitionen und der erneuten Partitionierung wird nun **fdisk** mit [Root-Rechten](#) in der Konsole gestartet. Um den notwendigen Platz für eine 2. Partition zur Verfügung zu haben, welche z. B. für den persistenten Modus benötigt wird, ist es notwendig, die erste Partition mit begrenzter Größe zu erstellen. Demnach ist Punkt 7 *besondere Aufmerksamkeit* zu widmen und die Punkte 12. bis 16. werden nur für eine persistente Installation benötigt!

```
sudo fdisk /dev/sdX.
```

1. **P** + eingeben um sich zunächst eine Liste der bestehenden Partitionen anzeigen zu lassen
2. **D** + um diese zu löschen (alle bestehenden Partitionen), ggf nochmals Punkt 1 ausführen, um zu überprüfen, ob wirklich alles gelöscht wurde.
3. **N** + zum Erstellen einer Partition
4. **P** + eingeben für eine primäre Partition und
5. **1** + um diese als erste Partition anzulegen
6. um die Vorgabe zu akzeptieren (Partitionsbeginn am ersten (freien) Zylinder)
7. Partitionsgröße:
 - o **+** **7** **3** **4** **M** + um die Größe auf ca. 700 [Mebibyte](#) zu beschränken (diese Option ist für die persistente Installation auszuwählen)
 - o um die Vorgabe zu akzeptieren (gesamte Kapazität des Gerätes für diese Partition nutzen)
8. **A** + eingeben, um das Bootflag der Partition zu setzen
9. **1** + wählen für die 1. Partition des USB-Sticks
10. **T** + eingeben, um die Standardeinstellung "Linux" des Dateisystems zu ändern
11. Code für des Dateisystem:
 - o **6** + ist für FAT 16 zu wählen, oder
 - o **B** + für FAT 32.
12. **N** + erneut – zum Erstellen einer zweiten Partition für den persistenten Modus
13. **P** + - wieder Primär
14. **2** + - dieses Mal als 2. Partition
15. - wie Punkt 6
16. um den restlichen zur Verfügung stehenden Platz zu verwenden
17. **W** + um die Änderungen auf den USB-Stick zu schreiben

Damit ist die Partitionierung mit **fdisk** beendet.

Wer sich etwas Tipp-Arbeit sparen möchte, kann alternativ unter Verwendung von [parted](#) die komplette Partitionierung auch durch Ausführung des nachfolgenden Befehls durchführen:

```
sudo parted /dev/sdX "mklabel msdos mkpart primary fat32 0 750 mkpart primary ext2 751 -0 set 1 boot on"
```

Wenn eine abweichende Partitionierung verwendet wird, ist in jedem Fall darauf zu achten, dass der 1. Partition ca. 700 [Mebibyte](#) (~734 Megabyte) zugeteilt werden.

Formatierung

Um die 1. Partition zu formatieren, folgenden Befehl in der Konsole eingeben:

```
sudo mkfs.vfat -F 32 -n LiveUSB /dev/sdX1
```

Wenn vorhin für die 1. Partition "FAT 32" gewählt wurde, ist die voranstehende Zeile so korrekt wie dargestellt. Für "FAT 16" ist folglich `-F 16` zu verwenden. Bei der Zeichenkette `LiveUSB` handelt es sich um die Bezeichnung der Partition und ist frei wählbar.

Zum Formatieren der 2. Partition verwendet man diesmal folgenden Befehl:

```
sudo mkfs.ext2 -b 4096 -L casper-rw /dev/sdX2
```

Die Zeichenkette `casper-rw` darf dieses Mal *nicht geändert* werden! Das System sucht beim Startvorgang des persistenten Modus nach eben dieser Kennung und verwendet folglich die Partition entsprechend ihrem Zweck.

Nun sollte der USB-Stick kurz abgezogen und wieder eingesteckt werden. Dies erspart das Einhängen der Partitionen von Hand, da dies unter Ubuntu automatisch geschieht. Wenn man sich an die vorherigen Beispiele gehalten hat, werden die Partitionen unter `LiveUSB` und `casper-rw` eingebunden.

CD-Inhalt kopieren

Wenn Ubuntu als CD vorliegt, sollte es keine Probleme geben, auf den Inhalt Zugriff zu erlangen, da diese automatisch eingebunden wird. Anders ist es, wenn Ubuntu nur als Imagedatei vorliegt. Dann ist diese, ähnlich wie eine Partition, zuerst in einen zuvor erstellten oder anderen leeren Ordner einzuhängen.



In diesem Fall, wird zunächst ein Ordner namens **ubuntuimage** im [Verzeichnis des Benutzers](#) erstellt (beides ist natürlich frei wählbar):

```
mkdir ~/ubuntuimage
```

Nun wechselt man in jenes Verzeichnis, wo das Ubuntu-Image abgelegt ist und bindet es im soeben erstellten Ordner ein:

```
sudo mount ubuntu-7.10-desktop-i386.iso ~/ubuntuimage -o loop
```


Nun kann die Ubuntu-CD oder der Ordner mit dem Image im [Dateimanager](#) geöffnet und der komplette Inhalt auf die erste Partition des USB-Sticks kopiert werden. Hierbei ist es zwingend notwendig, im verwendeten Dateimanager die Ansichtsoption "*Versteckte Dateien anzeigen*" angeschaltet zu haben. Andernfalls wird der Ordner **/.disk** nicht mit kopiert. Der Kopiervorgang wird nun einige Minuten in Anspruch nehmen und schließlich erhält man eine Fehlermeldung, welche besagt, dass ein "Symbolischer Link" nicht erstellt werden konnte. Die Ursache ist, dass im FAT-Dateisystem keine symbolischen Links erlaubt sind und somit der selbige nicht geschrieben werden kann. Dies ist aber harmlos und kann ignoriert werden.

Zusätzliche Arbeiten für Persistenz unter Hardy

In der Hardy Version 8.04, sowie wider Erwarten im ersten Point Release (8.04.1), existiert ein Fehler, welcher die Nutzung des persistenten Modus zunächst unmöglich macht. Um diesen Missstand zu beheben, ist nun etwas Bearbeitung von Nöten, welche im [Forum](#) (vorerst) genau beschrieben ist.

Es sei noch erwähnt, dass die beschriebenen Schritte äußerste Sorgfalt erfordern und nur Nutzern jenseits eines Neulings zu empfehlen sind!

Bootloader Installation

Da die meisten Nutzer sicherlich einen schon bekannten [Bootloader](#)  vorziehen werden, liegt im Folgenden der Schwerpunkt auf [GRUB](#). Für alle anderen wird jedoch auch die Variante mit **Syslinux** beschrieben.




Variante 1: GRUB (GRUB2)

Hinweis:

Hinweis: Werte in "(...)" gelten abweichen für GRUB2

Da [GRUB](#) auf Ubuntu-Systemen schon vorhanden ist (ab Version 9.10 GRUB2), kann direkt mit der Installation auf dem Stick begonnen werden. Vor dem eigentlichen Befehl zur Installation muss allerdings noch der Ordner **/boot** angelegt werden, in den die Grub-Installation mit dem 2. Befehl dann abgelegt wird:

```
sudo mkdir /media/LiveUSB/boot
sudo grub-install --root-directory=/media/LiveUSB --no-floppy /dev/sdX
```

Wurde der letzte Befehl ohne Fehler ausgeführt, sollte sich unter **/boot** jetzt ein neuer Ordner namens **grub** mit den entsprechenden Dateien als Inhalt befinden (GRUB Liste ??; GRUB 2 [Liste](#) ).

Nun legt man im Ordner **/boot/grub** eine neue Datei Namens **menu.lst** (*grub.cfg*) an, öffnet diese im Texteditor [\[3\]](#) und fügt den folgenden Text ein:

```
## menu.lst - Ubuntu LiveUSB Desktop 7.10
##
##      03.Nov.2007

default      0
timeout      10

title        Ubuntu Persistent
root         (hd0,0)
kernel       /casper/vmlinuz boot=casper persistent ramdisk_size=1048576 root=/dev/ram rw --
initrd       /casper/initrd.gz
boot

title        Ubuntu Live
root         (hd0,0)
kernel       /casper/vmlinuz boot=casper ramdisk_size=1048576 root=/dev/ram rw quiet splash--
initrd       /casper/initrd.gz
boot

title        Ubuntu Live (Sicherer Grafikmodus)
root         (hd0,0)
kernel       /casper/vmlinuz boot=casper xforcevesa ramdisk_size=1048576 root=/dev/ram rw --
initrd       /casper/initrd.gz
boot

title        Speichertest
root         (hd0,0)
kernel       /install/mt86plus
boot
```

Wurde die Installation ohne den persistenten Modus durchgeführt, kann der 1. Textblock folglich weggelassen werden. Erklärungen zur Verwendung dieses Bootloaders und den möglichen Optionen in der vorstehenden Datei findet man in den Artikeln [GRUB](#) und [menu.lst](#).

(*grub.cfg*):

```
menuentry "Lucid Lynx_64.iso live" {
    root (hd0,msdos1)
    loopback loop /ubuntu-10.04.1-desktop-amd64.iso
    linux (loop)/casper/vmlinuz boot=casper iso-scan/filename=/ubuntu-10.04.1-desktop-amd64.iso
    initrd (loop)/casper/initrd.lz
}
```

Mit grub2 können natürlich auch zuvor auf die entsprechende Partition entpackte iso-files gestartet werden.

(*grub.cfg*):

```
menuentry "Maverick Meerkat 10.10 Live" {
    root (hd0,msdos1)
    linux /casper/vmlinuz boot=casper ramdisk_size=1048576 root=/dev/ram rw --
    initrd /casper/initrd.lz
}
```

Hinweis:

weitere kernel Parameter müssen an das vorhandene System angepaßt werden:

- gfxpayload=1650x1024x24,1650x1024
- ramdisk_size=1048576
- cdrom-detect/try-usb=true
- locale=de_DE
- bootkbd=de console-setup/layoutcode=de
- quiet
- splash

(Ebenso wie oben beschrieben, läßt sich auch mit GRUB2 der persistent Modus einrichten. Dazu sind zwei **zusätzliche** Maßnahmen erforderlich):

- eine spezielle Partition
- ein spezieller Kernelparameter

(Die Partition soll ca. 2-3 GB groß sein, je nachdem, was alles gespeichert werden soll. Sie wird eingerichtet mit

```
sudo mkfs.ext2 -b 4096 -L casper-rw /dev/sdXY
```

X und Y müssen an die Gegebenheiten angepaßt werden)

(Der Kernelparameter lautet)

```
.../casper/vmlinuz boot=casper persistent...
```

Variante 2: Syslinux

Für die Erstellung eines bootbaren USB-Sticks mit Syslinux als Bootloader werden folgende Pakete installiert:

- **syslinux**
- **mttools**

Nun gibt man in der Konsole diesen Befehl ein:

```
sudo syslinux -f /dev/sdX1
```

Wenn der Befehl korrekt ausgeführt wurde, sollte sich die Datei **ldlinux.sys** nun im Hauptverzeichnis des USB-Sticks befinden (dafür muss im Dateimanager *Versteckte Dateien anzeigen* aktiviert sein).

Experten-Info:

Optional: Ldlinux in ein Verzeichnis installieren - Syslinux bietet ab Version 3.35 die Möglichkeit, auch in Unterverzeichnissen nach relevanten Dateien zu suchen. Damit ist es unter anderem möglich, durch Hinzufügen von `-d /Verzeichnis` die Datei **ldlinux.sys** in ein Unterverzeichnis zu installieren. Hierbei ist allerdings zu beachten, dass alle Dateibezeichnungen als auf jenes Verzeichnis bezogen angesehen werden, es sei denn, ein `/` ist vorangestellt.

Als nächstes ist der Ordner **/isolinux** in **/syslinux** umzubenennen, die darin enthaltene Datei **isolinux.cfg** in **syslinux.cfg** und **isolinux.bin** kann gelöscht werden. Wird Xubuntu verwendet, öffnet man nun die Datei **syslinux.cfg** und ändert sämtliche Einträge von **xubuntu.seed** in **ltsp.seed** um. Wurde die normale Installation ohne Persistenz gewählt, ist man jetzt fertig und kann das Ergebnis testen.

Anpassungen für den persistenten Modus

Wenn nicht schon geschehen, öffnet man im Texteditor ^[3] die Datei **syslinux.cfg**. Dort ist dann nach der `GFXBOOT-Zeile` ein neuer Textblock für den persistenten Modus einzufügen. Das nachstehende Beispiel sollte eigentlich keine Frage offen lassen:

```

DEFAULT persistent
GFXBOOT bootlogo

LABEL persistent
    KERNEL /casper/vmlinuz
    APPEND file=/preseed/ubuntu.seed boot=casper persistent initrd=/casper/initrd.gz --

LABEL live
    KERNEL /casper/vmlinuz
    APPEND file=/preseed/ubuntu.seed boot=casper initrd=/casper/initrd.gz splash --
...
...
TIMEOUT 100
PROMPT 1
...
```

Einige Hinweise zu möglichen Einstellungen in der Datei:

- **DEFAULT** legt fest, welcher Modus nach Ablauf der Zeit ohne Nutzereingabe gestartet wird. Zu verwenden sind die Einträge hinter **LABEL** (live, xforcevesa usw.).
- **PROMPT 1** zeigt das Auswahlmenü beim Start, **PROMPT 0** startet ohne Menüanzeige sofort den Standardmodus.
- **TIMEOUT 300** bedeutet **30** Sekunden (nicht 300!) Wartezeit bis zum Start.
- **quiet** unterdrückt alle Startmeldungen.
- **splash** zeigt beim Starten das Logo der verwendeten Ubuntu-Version.
- Ohne **quiet** und **splash** erhält man eine Ausgabe aller Meldungen im Konsolenstil.

Anpassung der Sprache

Für die Verwendung des normalen Live-Systems gibt es den Vorteil, dass die Standardeinstellung der Sprache beeinflusst werden kann. Dazu öffnet man die Datei **syslinux.cfg** und fügt für ein deutsches Layout in den **APPEND**-Zeilen die Optionen `locale=de_DE bootkbd=de console-setup/layoutcode=de console-setup/variantcode=nodeadkeys` hinzu. In der Praxis sollte dies dann wie folgt aussehen:

```

LABEL live
    KERNEL /casper/vmlinuz
    APPEND file=/preseed/ubuntu.seed boot=casper locale=de_DE bootkbd=de console-setup/layoutcode=de console-s
```

Ballast abwerfen

Inhalte für Windows



Im Paket der Desktop-CD befinden sich unter Umständen diverse Dateien, die nur für die Nutzung unter Windows gedacht sind. Dabei handelt es sich um die Windowsversionen einiger Open-Source-Programme, eine Präsentation dieser, Dateien für den Autostart und **Wubi**, ein Programm zur Ubuntu-Installation unter Windows. Wenn man keine derartige Verwendung plant, können diese Bestandteile nun entfernt werden.

Zunächst löscht man im Hauptverzeichnis diese Dateien (wenn vorhanden):

- **autorun.inf**
- **start.bmp**
- **start.exe**
- **start.ini**
- **ubuntu.ico**
- **wubi-cdboot.exe** oder **wubi.exe** und **umenu.exe**

Kurz gesagt, alle außer den Dateien **md5sum.txt** und **README.diskdefines**.

Nun entfernt man noch die Ordner:

- **bin**
- **disctree**
- **pics**
- **programs**

Wenn alle der hier angeführten Ordner und Dateien gelöscht wurden, sind das immerhin über 30 Mebibyte weniger. Bei Xubuntu oder neueren Versionen wie Hardy wird die Ersparnis weit geringer ausfallen, da die meisten der genannten Inhalte schon fehlen.

Weiteren Platz einsparen

Möchte man noch mehr Ballast loswerden, sollte das Programm [Ubuntu Customization Kit](#) weiterhelfen können. Es ermöglicht, ein Image der Desktop-CD zu bearbeiten und dabei gezielt nicht benötigte Pakete zu entfernen oder hinzuzufügen.

Problemlösungen



Bootprobleme

Wenn das Starten vom USB-Stick fehlschlägt, ist guter Rat teuer. Im Folgenden nun einige Hinweise, welche in den häufigsten Fällen weiterhelfen sollten.

BIOS Einstellungen

Damit von einem USB-Stick gebootet werden kann, müssen im Bios auch die entsprechenden Einstellungen vorgenommen sein. Je nach Motherbord- respektive BIOS-Hersteller wird dies in Menüs mit Bezeichnungen wie *Boot*, *Boot Device Priority*, *Advanced Bios Features*, *Bios Features Setup*, *Advanced CMOS Setup* oder Ähnlichem festgelegt. Im Folgenden nun einige mögliche Optionen und Tipps:

- Wenn man kein USB-Gerät als erstes Boot-Laufwerk wählen kann, sollte geprüft werden, ob sich irgendwo eine Option wie *Boot From Other Device* auswählen lässt.
- Befindet sich dort eine Option namens *USB Legacy Support* sollte diese eingeschaltet werden.
- Als Typ sollte zuerst *USB-HDD* und danach *USB-ZIP* oder wenn vorhanden *USB-Floppy* versucht werden.
- Versuchsweise kann die Funktion *USB Hi-Speed (2.0)* abgeschaltet werden.
- Die Funktion *USB keyboard emulation* testweise anschalten.
- Den *Fast Boot* abzuschalten kann auch versucht werden.

USB-Port wechseln

Oftmals ist es so, dass sich zwei USB-Ports die gleichen Systemressourcen teilen, woraus folgt, dass immer nur einer dieser voll funktionsfähig ist. Weiterhin werden manche USB-Ports an der Vorderseite des PC nicht vollständig vom System unterstützt. In diesen Fällen reicht ein einfaches Wechseln des Ports schon aus.

Desweiteren kann auch versucht werden, weitere angeschlossene USB-Geräte mit anderen USB-Ports zu verbinden oder zu tauschen. Dies hilft manchmal, bis zu diesem Zeitpunkt unmerkliche Konflikte in der Konfiguration zu lösen.

Syslinux

Sollte als Bootloader Syslinux verwendet werden, ist es möglich mit dem zusätzlichen Schalter `-s` einen weiteren Versuch zu machen. Der Konsolenbefehl ^[2] dafür sieht dann so aus:

```
sudo syslinux -sf /dev/sdX1
```

Damit wurde eine äußerst kompatible Variante des Syslinux Bootloaders installiert, so dass auch problematische USB-Stick- / Motherboard- Kombinationen zur korrekten Funktion bewegt werden können.

U3-USB-Sticks

U3-kompatible USB-Sticks unterscheiden sich etwas von den normalen Geräten. Diese sind meist nicht bootfähig und vom Hersteller so eingerichtet, dass sich eine kleine spezielle und eine normale Partition darauf befinden. Die spezielle wenige Megabyte fassende Partition enthält das sogenannte U3-System, welches unter Windows ein CD-Laufwerk simuliert. Dieser Bereich ist für den Nutzer nicht direkt verwendbar (da schreibgeschützt) und ist der Grund, warum der USB-Stick für eine Linux-Installation zunächst unbrauchbar ist: Grub2 kann nicht in den MBR installiert werden.

In den am Ende befindlichen [Links](#) ist ein Verweis auf das U3-Deinstallationsprogramm zu finden. Ob der USB-Stick anschließend starten wird, ist damit allerdings nicht zu garantieren. Es sei aber angemerkt, dass sich die U3-Software wieder installieren lässt. Die Deinstallation funktioniert nur unter Windows oder in einer virtuellen Maschine mit Windows (z.B. mittels VMWare oder VirtualBox PUEL, wichtig ist die USB-Unterstützung!).

USB-Stick auf Bootfähigkeit prüfen

Achtung!

Bei der folgenden Prozedur werden alle auf dem USB-Stick befindlichen Daten **vernichtet**! Wenn noch nicht erfolgt, sollte jetzt eine Sicherung aller evtl. vorhandenen Daten erstellt werden.

Linux-Installation

Unter Linux muss leider einiger Aufwand betrieben werden, um einen USB-Stick sicher auf die gewünschte Funktion testen zu

können. Da dies insbesondere Einsteigern einige Schwierigkeiten bereiten kann, wurden die nötigen Schritte in ein [Shellscript](#) geschrieben und zusammen mit den benötigten Dateien hier zur Verfügung gestellt.

Zunächst lädt man das im Folgenden angehängte Dateiarhiv herunter und entpackt ^[4] es an beliebiger Stelle im [Nutzerverzeichnis](#). Desweiteren benötigt man nun die [korrekte Adresse des USB-Stick](#) und merkt oder kopiert sich den Pfad zu den eben entpackten Dateien.

USB Boot-test Download

Nun wechselt man im Terminal ^[3] in das eben entpackte Verzeichnis mit dem Shellscript **ubt.sh**. Um es auch verwenden zu können, muss es zunächst ausführbar gemacht werden. Da man sich eh schon in der Konsole befindet, können die Ausführrechte ^[5] auch gleich mit dem folgenden Befehl gesetzt werden:


```
chmod +x ubt.sh
```

Nun kann das Script ausgeführt werden, wie es wie im folgenden Beispiel dargestellt ist:

```
sudo ./ubt.sh /dev/sdX /home/benutzer/UsbBootTest_lx
```

Hat man die USB-Adresse und das Verzeichnis korrekt angegeben, sollte das Script alles problemlos abgearbeitet haben und man kann nun versuchen vom USB-Stick zu starten.

Windows-Installation

Bedingt durch den größeren Funktionsumfang der Windowsversion von Syslinux ist es unter Windows einfacher den USB-Stick entsprechend zu bearbeiten. Zudem existiert bereits ein Programmpaket, welches den Vorgang weiter vereinfacht. Dazu lädt man sich von der Seite [Pendrivelinux](#)  die benötigten Dateien runter und folgt den dortigen Anweisungen.

Defekter Startsektor

Es kommt vor dass ein defekter oder falscher Bootsektor installiert bzw. noch gar keiner vorhanden ist. Dies kommt z.B. vor, wenn ein veränderter U3-Stick verwendet wird (siehe zuvor unter U3).

Das folgende Paket wird benötigt:

- **mbr**

Nun mittels der Konsole am verwendeten Gerät den folgenden Befehl durchführen:

```
install-mbr /dev/sdX
```

Das war es auch schon. Wenn keine Fehlermeldung ausgegeben wurde, ist der neue Bootsektor jetzt installiert und es kann erneut versucht werden vom Gerät zu starten.

Probleme nach dem Bootmenü




Sollten Probleme nach der Auswahl im Bootmenü auftreten, findet man in den folgenden Artikeln Hilfe:

- [Ubuntu Installation](#)
- [Ubuntu-CD Problembhebung](#)

Links und Quellen

Quellen

Dieser Artikel basiert zum Teil auf Anleitungen von folgenden Webseiten:

- [Das englische Ubuntu Wiki - Beschreibung der persistenten Installation](#) 
- [Pendrivelinux - Webseite mit jeder Menge Nützlichem zum Thema USB](#) 
- [- GRUB Installation für USB im engl. Ubuntuforum](#) 

Weitere Links

- [Syslinux Homepage](#) 
- [Deinstallations-Programm für U3 USB-Sticks](#) 

Diese [Revision](#) wurde am 1. April 2011 um 04:17 Uhr von [aasche](#) erstellt.
Dieser Seite wurden folgende Begriffe zugeordnet: [System](#), [Installation](#)