

Kopieren eines kompletten, installierten GNU/Linux auf einen bootbaren USB-Stick

Stand: 03.04.2022

Quelle: Neville Jackson, <https://github.com/nevillejackson/>, Übersetzung: FSOG

1 Einführung

Manchmal möchte man eine GNU/Linux-Installation ersetzen, diese aber dennoch, zum Beispiel für Referenzzwecke, „aufheben“.

Ein reguläres Backup der Partitionen würde ausreichen, die Wiederherstellung ist aber doch recht mühsam. Eine bequeme Möglichkeit ist es, das gesamte System auf einen USB-Stick zu kopieren, und diesen dann bootfähig zu machen.

Es gibt zwei Wege, ein laufendes GNU/Linux auf einen USB-Stick zu transferieren.

Da gibt es einen guten Artikel bei FOSS [1] zu diesem Thema.

Einer ist, eine .iso Datei zu erzeugen, diese dann auf den Stick zu „brennen“ und bootbar machen. Dies resultiert in einem sogenannten Live Dateisystem. Es kann gebootet und verwendet werden, alle Änderungen daran gehen jedoch beim Herunterfahren verloren. Es gibt Möglichkeiten, dem System sogenannte Persistenz beizubringen, deren Möglichkeiten sind jedoch begrenzt, kompliziert und sie ermöglichen keine Kernel-Updates.

Die zweite Möglichkeit ist das Äquivalent zu einer vollständigen Installation auf dem USB-Stick. Das erlaubt ihm, quasi als Festplatteninstallation zu arbeiten, inklusive voller Persistenz und aktualisierbarkeit.

Diese Anleitung erklärt die zweite Möglichkeit.

2 Notwendige Schritte zum vollständigen Transfer auf einen USB-Stick

Die einzelnen Schritte sind:

- ♥ Vorbereiten der Partitionen des USB-Sticks
- ♥ Übertragen einer Kopie der gewünschten GNU/Linux Installation von der Festplatte auf den Stick
- ♥ Einhängen der neuen Kopie und anpassen der Datei /etc/fstab
- ♥ Installieren von Grub auf dem USB-Stick
- ♥ Booten von Grub in die Grub-Befehlszeile
- ♥ Booten der Kopie vom USB-Stick von der Grub-Befehlszeile
- ♥ Das so gebootete GNU/Linux System verwenden um Grub zu konfigurieren

2.1 Partitionen vorbereiten

Hier verwenden wir zum Beispiel gparted.

Wir formatieren den USB-Stick und erzeugen (mindestens) folgende vier Partitionen:

1. EFI Systempartition, fat32, 512 MB, flags: boot und esp
2. BIOS-Bootpartition, kein Dateisystem, 1MB, flags: bios und grub
3. Linux Root-Partition, ext4, mindestens 20GB, Einhängepunkt: /
4. Linux Swap-Partition, swap, 4GB
5. Weitere, falls benötigt (Höchstanzahl von Primärpartitionen beachten!)

Die EFI Systempartition sollte am Anfang stehen.

2.2 Kopieren der GNU/Linux root-Partition von Festplatte auf USB

Es gibt verschiedene Wege, ein ganzes GNU/Linux root-Dateisystem zu kopieren.

Wir verwenden „rsync“, aber „dd“ wäre ebenfalls eine Option. Das root-Dateisystem muss nicht als image kopiert werden, sondern einfach nur kopiert.

1. Starten eines GNU/Linux (**nicht das**, welches kopiert werden soll)
2. Einhängen des Dateisystems, welches kopiert werden soll.
Z.B.: **mount /dev/sdb2 /media/edi/Linuxroothome**
3. Feststellen der Bezeichnung der Partition des (Ziel-) USB-Sticks, wohin kopiert werden soll:
lsblk listet alle Partitionen auf (ob eingehängt oder nicht). Z.B.: /dev/sdc3
4. **mount /dev/sdc3 /mnt**
5. **rsync -aAXvH --exclude={dev/*,proc/*,sys/*,tmp/*,run/*,mnt/*,media/*,lost+found,common/*} /media/edi/Linuxroothome/ /mnt**
Dies kopiert alle Verzeichnisse außer den ausgeschlossenen. Im Beispiel ist „common“ eine weitere Datenpartition, welche ausgeschlossen werden soll.
6. Vor dem nächsten Schritt ein **sync** ausführen

Der Schalter „--exclude“ in Schritt 5 verhindert das Kopieren von Pseudo Dateisystemen, welche beim Booten befüllt werden und aller mounts, speziell /mnt wohin die USB-Stick Partition eingehängt wurde.

2.2.1 Hinweis zu rsync

Der rsync-Schalter „--exclude{dev/*, ...}“ verwendet geschweifte Klammern.

Das funktioniert nur in der Bash oder Csh (Corn-Shell).

Ein Test zur Funktionsüberprüfung im Terminal:

```
echo {a, b, c}
```

Ergebnis sollte sein: „a b c“

Falls stattdessen „{a,b,c}“ ausgegeben wird, muss Bash gestartet werden, bevor man das rsync ausführt. Wenn Bash nicht installiert ist, kann man es über den Paketmanager installieren.

Als letzten „Strohalm“ kann man den rsync-Befehl folgendermassen anpassen:

```
rsync -aAXvH --exclude='dev/*' --exclude='proc/*' --exclude='sys/*' --exclude='tmp/*' --exclude='run/*' --exclude='mnt/*' --exclude='media/*' --exclude='lost+found' --exclude='common/*' /media/edi/Linuxroothome/ /mnt
```

Dies sollte in jeder Shell funktionieren. Es ist auch egal, ob die Anführungszeichen einfach oder doppelt sind.

2.2.2 Kopieren aus einem aktiven GNU/Linux Dateisystem

Das ist möglich. Man sollte alle Benutzerprozesse anhalten, die eventuell Dateien schreiben könnten. Der rsync-Befehl ist in diesem Fall anders aufgebaut:

```
rsync -aAXvH --exclude=\{'dev/*','proc/*','sys/*','tmp/*','run/*','mnt/*','media/*','lost+found','common/*'\} / /mnt
```

Weil man das root-Verzeichnis nicht einhängen muss.

2.3 Patchen der Datei /etc/fstab und entfernen der Grub-Konfiguration

Es kann Einträge in der Datei /etc/fstab geben, die angepasst werden müssen.

Speziell die UUID's müssen auf die korrekten Werte der jeweiligen USB-Stick Partitionen gesetzt werden.

1. Ermitteln der UUID Nummern der Partitionen des USB-Sticks.
Entweder mit einem grafischen Werkzeug oder im Terminal mit **ls -l /dev/disk/by-uuid**
2. Bearbeiten der Datei /etc/fstab in der root-Partition des USB-Sticks.
Sorgfältiges Übertragen der UUID's in die fstab.
Es wird mindestens eine für das root-Dateisystem benötigt und eine für die Swap-Partition.

Die resultierende Datei /etc/fstab sollte so oder so ähnlich aussehen:

```
# Pluggable devices are handled by uDev, they are not in fstab
# / sdc3
UUID=0beb5819-f2ba-4fa0-aa69-3e5ec16fb0bc / ext4 noatime 1 1
# swap sdc5
UUID=ef94e2d5-c924-49b8-a944-486efd629340 swap swap noatime 1 2
# spare partition sdc4
UUID=d4109c75-0428-4bb9-8d19-d0b63d09930a /home/nej/spare ext4 0,users 2 0
```

Die einzigen essenziellen sind / und Swap.

Hier gibt es noch eine extra Partition namens „spare“.

Es ist nicht ganz sicher, ob der Eintrag / benötigt wird.

Falls das auf den USB-Stick kopierte GNU/Linux auf der Festplatte Grub konfiguriert hatte, ist es notwendig die Grub-Konfiguration auf dem USB-Stick zu entfernen.

Im Terminal nach /mnt/boot (auf dem USB-Stick) wechseln und folgendes ausführen:

```
rm -r grub
rm -r efi
```

2.4 Grub auf dem USB-Stick installieren

Hier hilft es, das GNU Grub Handbuch zu studieren [2].

Anmelden an einem GNU/Linux auf der Festplatte. Überprüfen Sie die Partitionsnamen des USB-Sticks mittels **lsblk**. Wir hängen das root-Verzeichnis des USB-Sticks folgendermaßen ein:

```
mount /dev/sdc3 /mnt
```

wobei sdc3 das root-Verzeichnis ist. Dann wird Grub installiert:

```
grub-install --boot-directory=/mnt/boot \ \ --recheck \ \ --removable \ \ --
targer=i386-ps \ \ /dev/sdc
```

wobei sdc den Gerätenamen des USB-Sticks darstellt.

Dies installiert Grub für das Booten im Legacy-Modus.

2.5 Mit der Grub-Befehlszeile das GNU/Linux vom USB-Stick starten

Den Computer neu starten und im BIOS vom USB-Stick starten. Dies sollte zur Grub-Befehlszeile führen:

```
grub>
```

Falls das nicht der Fall sein sollte, besteht ein Fehler.

Es gibt kein (gewohntes) Grub-Menü weil wir Grub auf dem USB-Stick noch nicht eingerichtet haben.

Nun kann die GNU/Linux Kopie auf dem USB-Stick mit den folgenden Grub-Befehlen gebootet werden:

```
grub> linux /vmlinuz root=/dev/sdc3
grub> initrd /initrd.img
grub> boot
```

und es sollte starten. Man kann sich anmelden.

2.6 Mit dem gebooteten USB-Stick Grub auf dem USB-Stick konfigurieren

Nachdem wir nun die USB-Stick Kopie unseres GNU/Linux gebootet haben, können wir es nutzen um seine eigene Grub-Konfiguration einzurichten. Das ist einfach.

Bearbeiten der Datei /etc/default/grub und hinzufügen (bzw. abändern) der folgenden Zeile:

```
GRUB_DISABLE_OS_PROBER=true
```

dann einfach

```
update-grub
```

Dieser Befehl sollte das GNU/Linux auf dem USB-Stick finden, nicht jedoch andere Systeme z.B. auf der Festplatte. Das ist genau was wir wollen – wir wollen den USB-Stick und Grub unabhängig von der Festplatte konfigurieren.

Wenn man schon im Terminal ist, kann man gleich noch testen, ob Swap eingehängt ist:

```
swapon --show
```

Dass der USB-Stick seinen eigenen Swap hat, ist Teil der Unabhängigkeit, die erreicht werden soll. Alle anderen Mounts kann man nun auch testen, je nach Bedarf.

2.7 Test-Boot (BIOS)

Den Computer neu starten und erneut das BIOS verwenden, um vom USB-Stick zu booten. Dieses mal sollte ein Grub-Menü erscheinen, statt der Kommandozeile.

Dieses sollte drei Einträge aufweisen:

- 🍷 Das „kopierte“ GNU/Linux
- 🍷 Das „kopierte“ GNU/Linux nochmal, aber im Advanced Modus
- 🍷 und eventuell Memtest (wenn es mit Grub geliefert wird)

Testen, ob man aus dem Menü heraus booten kann.

Dann die „Feuerprobe“...

Herunterfahren, den Stick abziehen und in einem anderen Computer damit booten.

2.8 Booten im UEFI-Modus

Es gibt keinen Hinderungsgrund, warum man nicht eine weitere Grub-Installation auf den selben USB-Stick machen könnte, nur dass man in diesem Fall die EFI-Partition (statt der BIOS-Partition) als Ziel nimmt.

Der Stick wäre dann in beiden Modi bootbar. Die beiden Grubs würden eine Konfiguration teilen, sodass man den Schritt mit „update-grub“ nicht wiederholen müsste.

Also machen wir das. Dazu muss das root-Verzeichnis des USB-Sticks eingehängt werden:

```
mount /dev/sdc3 /mnt
```

Wobei „sdc3“ das root-Verzeichnis ist (kann mit lsblk überprüft werden).

Dann installiert man Grub folgendermaßen:

```
grub-install --boot-directory=/mnt/boot --recheck --removable --target=x86_64-efi --efi-directory=/dev/sdc1
```

Man beachte, dass kein Gerät angegeben wird, weil Grub efi nicht auf ein Gerät schreibt sondern in die EFI-Systempartition. Dies installiert Grub im UEFI Modus.

Man muss Abschnitte 2.5 oder 2.6 nicht wiederholen, weil die Grub-Konfiguration auf dem USB-Stick bereits im Legacy Grub erledigt wurde. Die beiden Grubs teilen die selben Konfigurationsdateien auf dem USB-Stick.

2.9 Test-Boot (UEFI)

Den Computer neu starten mit dem BIOS im UEFI Modus und den USB-Stick zum Starten auswählen. Das Grub-Menü des USB-Sticks erscheint. Testen, ob aus diesem Menü heraus gestartet werden kann. Dann der Härtetest – booten des USB-Sticks mit einem anderen Computer im UEFI Modus.

2.10 Danksagung

Rückmeldungen von Edgar Hoffmann

3 Quellen

- [1] Emmanuel (2021) Persistent Live USB vs. Full Linux Install on a USB Drive.
<https://www.fosslinux.com/49280/persistent-live-usb-vs-full-linux-install-usb-drive.htm>
- [2] Matzigkeit, Okuji, Watson, and Bennett (2021) The GNU GRUB Manual
<https://www.gnu.org/software/grub/manual/grub/grub.html>