

Remaster-Kit Anleitung

von Carsten Rohmann und Leszek Lesner

1. Was ist Remaster-Kit?

Remaster-Kit ist ein Programm, welches das Remastern und Neubauen von ZevenOS-Neptune und anderen Debian- bzw. Ubuntu-basierenden Linuxsystemen erleichtert. Es handelt sich um ein Programm, das per Kommandozeile bedient wird. Keine Sorge, diese Anleitung enthält eine Schritt-für-Schritt-Anleitung, mit der man sicher ans Ziel kommt.

2. Wie benutzt man Remaster-Kit?

Zusammenfassend gesagt, sieht das Vorgehen so aus: zunächst einmal wird eine Ausgangs-ISO-Datei von der Distribution benötigt, die remastert werden soll. Diese ISO-Datei wird von Remaster-Kit gemountet und ihre Bestandteile werden in ein Arbeitsverzeichnis kopiert. Das Arbeitsverzeichnis enthält Unterverzeichnisse. Zwei von diesen Unterverzeichnissen, die angelegt werden, sind besonders wichtig, sie heißen *extract-cd* und *edit*. (Die übrigen Unterverzeichnisse sind nur temporär und brauchen hier nicht weiter besprochen werden.)

Danach werden je nach Wunsch Modifikationen am Live-System vorgenommen. Zum Schluss wird eine neue ISO-Datei erstellt, die auf CD oder DVD gebrannt werden kann. Fertig ist das selbst remasterte Live-System!

3. Schritt-für Schritt-Anleitung

Angenommen, die Ausgangs-ISO-Datei heißt *neptune19.iso* liegt im Home-Verzeichnis *myname* im Ordner *Downloads*, der Pfad wäre also: `/home/myname/Downloads/neptune19.iso`

1) Ein Root-Terminal öffnen

Nachdem man ein Root-Terminal geöffnet hat (das befindet sich unter Menü > Dienstprogramme > Root -Terminal) kann man als jetzt als Superuser Root arbeiten. Doch Vorsicht, als Root hat man Zugriff auf das ganze Linuxsystem und muss daher sorgfältig aufpassen, um nichts am System zu beschädigen.

2) Ins Wurzelverzeichnis wechseln

Mit dem folgenden Befehl wechselt man ins Wurzelverzeichnis des Linuxsystems:

```
cd /
```

3) Das Arbeitsverzeichnis erstellen

Dort soll das notwendige Arbeitsverzeichnis erstellt werden. Hier im Beispiel soll es *TestProject* heißen. Dafür gibt man im Root-Terminal folgenden Befehl ein:

```
remaster-kit create TestProject /home/myname/Downloads/neptune19.iso
```

Damit wird das Arbeitsverzeichnis *TestProject* erzeugt, das diverse Unterverzeichnisse enthält. Danach werden automatisch im Arbeitsverzeichnis *TestProject* der Inhalt des ISOs ins Unterverzeichnis *extract-cd* kopiert und der Inhalt des Live-Systems (filesystem.squashfs) in das Unterverzeichnis *edit* extrahiert. Die anderen Verzeichnisse sind nur temporär und werden als Einhängpunkte verwendet. (Sie könnten nach dieser Aktion gefahrlos gelöscht werden, aber wer sich nicht sicher ist, sollte das lieber unterlassen).

4) Anpassungen am Live-System

Das zukünftige Live-System, das im Verzeichnis *edit* liegt, kann nun optional auf zwei

verschiedene Arten den eigenen Wünschen angepasst werden:

- a) durch Bearbeiten von Konfigurationsdateien und
- b) durch einen „changeroot“, mit dem im zukünftigen Live-System neue Programme installiert oder nicht mehr gewünschte Programme entfernt werden können.

4.a) Die wichtigsten Verzeichnisse im Live-System

Die folgende Tabelle listet die wichtigsten Verzeichnisse auf, die man bearbeiten kann um die Konfiguration des zukünftigen Live-Systems zu beeinflussen.

Pfad	Beschreibung	Notizen
/etc/skel	Standardbenutzer-Profilverzeichnis das zum Erstellen eines neuen Benutzers gebraucht wird	Hier kann man Standardkonfigurationen für den Desktop und das Aussehen eines neu angelegten Benutzers festlegen
/usr/share/icons	Hier werden globale Icon-Themes gespeichert	Hier können neue Icon-Themes, gespeicher sowie Maus-Themes hinzugefügt werden
/usr/share/themes	Hier werden globale Themes gespeichert	Hier können neue Themes für KDE/Gnome/XFCE hinzugefügt werden

4.b) Der „Changeroot“

Anpassungen am Live-System werden vorgenommen, indem man das Live-System im Verzeichnis *edit* mit dem Befehl **chroot** übernimmt. Dazu muss man sich im Verzeichnis *TestProject* befinden und als Root folgenden Befehl eingeben:

```
remaster-kit chroot
```

Nun sollte sich die Eingabeaufforderung verändern und es sollte etwas wie:

```
root@neptune:/#
```

zu sehen sein. Das Verzeichnis *edit* wurde hiermit als neues Root-Verzeichnis übernommen.

In der Changeroot-Umgebung kann man nun als Superuser arbeiten und Pakete installieren oder entfernen. Den X-Server für eine grafische Oberfläche zu starten würde theoretisch zwar klappen, wird aber **nicht** empfohlen, weil danach viel Aufräumarbeiten durchgeführt werden müssen. Also beschränke man sich lieber auf den Gebrauch der Kommandozeile. Folgende Befehle sind nützlich, wenn man in der Changeroot-Umgebung arbeitet:

- Aktualisieren der Paketquellen:

```
apt-get update
```
- Installieren neuer Programme aus dem Internet:

```
apt-get install <Paketname>
```
- Installieren eines lokal im Verzeichnis *edit* abgelegten Programms:

```
dpkg -i <Paketname>
```
- Eine Liste aller installierten Pakete:

```
dpkg -list
```

- Beim Entfernen von Paketen sollte man auch alle Konfigurationsdateien löschen. Um wirklich alles restlos zu löschen, verwendet man den Befehl:
`apt-get purge <Paketname>`
- Die Programme des Live-Systems können (vorsichtig) mit folgenden Befehlen aktualisiert werden:
`apt-get upgrade` oder `aptitude safe-upgrade`

5) Die Changeroot-Umgebung wieder verlassen

Nachdem alle gewünschten Veränderungen vorgenommen wurden muss das Live-System wieder ordnungsgemäß verlassen werden. Dazu gibt man im Root-Terminal folgenden Befehl ein:

`exit`

6) Die neue ISO-Datei erstellen

Wenn alles soweit fertig ist, lässt sich mit folgendem Kommando eine neue ISO-Datei erstellen:

`remaster-kit build VolumeName new-remastered-neptune.iso`

Dieser Befehl erstellt im Arbeitsverzeichnis *TestProject* die Datei *new-remastered-neptune.iso* mit dem Label *VolumeName*.

(Hinweis: *VolumeName* darf keine Leerzeichen enthalten)

Achtung: Linux Distributionen mit Kernen vor 2.6.29 können nicht mit remaster-kit remastert werden! Ursache ist eine inkompatible Version von squashfs.

4. Wie funktioniert Remaster-Kit?

Remaster-Kit ist ein reines Bash-Shellscript, das sich im installierten System unter `/usr/bin` befindet. Es kann mit einem normalen Texteditor geöffnet und bearbeitet werden. Dort erhält man auch einen Einblick in die Funktionalität, die Remaster-Kit bietet.

Eine Liste mit wichtigen Befehlen:

create

Erstellt ein Projektverzeichnis, mountet das ISO und kopiert dessen Inhalt in das Unterverzeichnis `extract-cd` des Projektverzeichnisses. Anschließend wird das `filesystem.squashfs` gemountet und nach `edit` kopiert.

chroot

Wechselt als `root` ins Live Verzeichnis `edit` und führt dort einige Aktionen aus um eine vorhandene Internetverbindung nutzen zu können und Programme nachinstallieren zu können.

clean

Unmountet die temporären Verzeichnisse im Projektverzeichnis und löscht diese anschließend.

remove

Löscht das Projektverzeichnis komplett

5. Unter welcher Lizenz steht Remaster-Kit?

Remaster-Kit steht unter einer BSD-Lizenz, die sowohl die kommerzielle sowie freie Nutzung der Software und der Software-Quellen erlaubt. Eine kommerzielle Weitergabe ohne Quellen ist ebenfalls erlaubt.