

Benutzerdefinierte Installations-ISO erstellen / Live-USB vom installierten System erstellen

Stand: 02.08.2023

Quelle: <https://www.freiesoftwareog.org>

Benutzerdefinierte Installationsmedien sind eine coole und vor Allem nützliche Sache.



Inhaltsverzeichnis

Allgemeines.....	2
Installation von Cubic.....	3
Hinweis zur Verwendung mit anderen Distributionen.....	3
Cubic: Der einfachste Weg zu benutzerdefinierten ISO-Abbildern.....	4
Die wichtigsten Funktionen von Cubic.....	5
Verwendung von Cubic zur Erzeugung eines benutzerdefinierten ISO-Abbildes.....	5
Vorgaben.....	5
Hinzufügen/Erzeugen eines Projektverzeichnisses.....	6
Auswahl des Original ISO-Abbildes.....	7
Arbeiten mit der chroot-Umgebung.....	8
Entfernen und Hinzufügen von Paketen.....	9
Aktualisieren bestehender Pakete.....	11
Hintergrundbilder hinzufügen.....	12
Linux-Kernel hinzufügen.....	14
Zusammenstellung der Optionen.....	16
Zu entfernende Pakete für die Normale und Minimale Installation auswählen.....	17
Boot-Kernel auswählen.....	18
Auswahl des ISO Kompressionstyps.....	19
Erzeugen der benutzerdefinierten ISO-Datei.....	20
Testen der Abbilddatei mit dem integrierten QEMU.....	21
Problemlösung(en).....	22
Meldung: „Bitte verwenden Sie apt-cdrom, damit diese CD-ROM von APT erkannt wird“.....	22
Eigene Notizen.....	24

Allgemeines

Die Installation von Linux Mint ist keine große Sache.

Es sind die Dinge, die nach der Installation von Linux Mint zu tun sind, die sich lästig anfühlen können.

Und wenn man dasselbe auf mehr als einem System tun muss, wird es frustrierend.

Man stelle sich vor, man hat mehrere Computer zu Hause, im Labor oder in einer Institution. Und alle müssen ähnliche oder gleichartige Konfigurationen und Anwendungen haben.

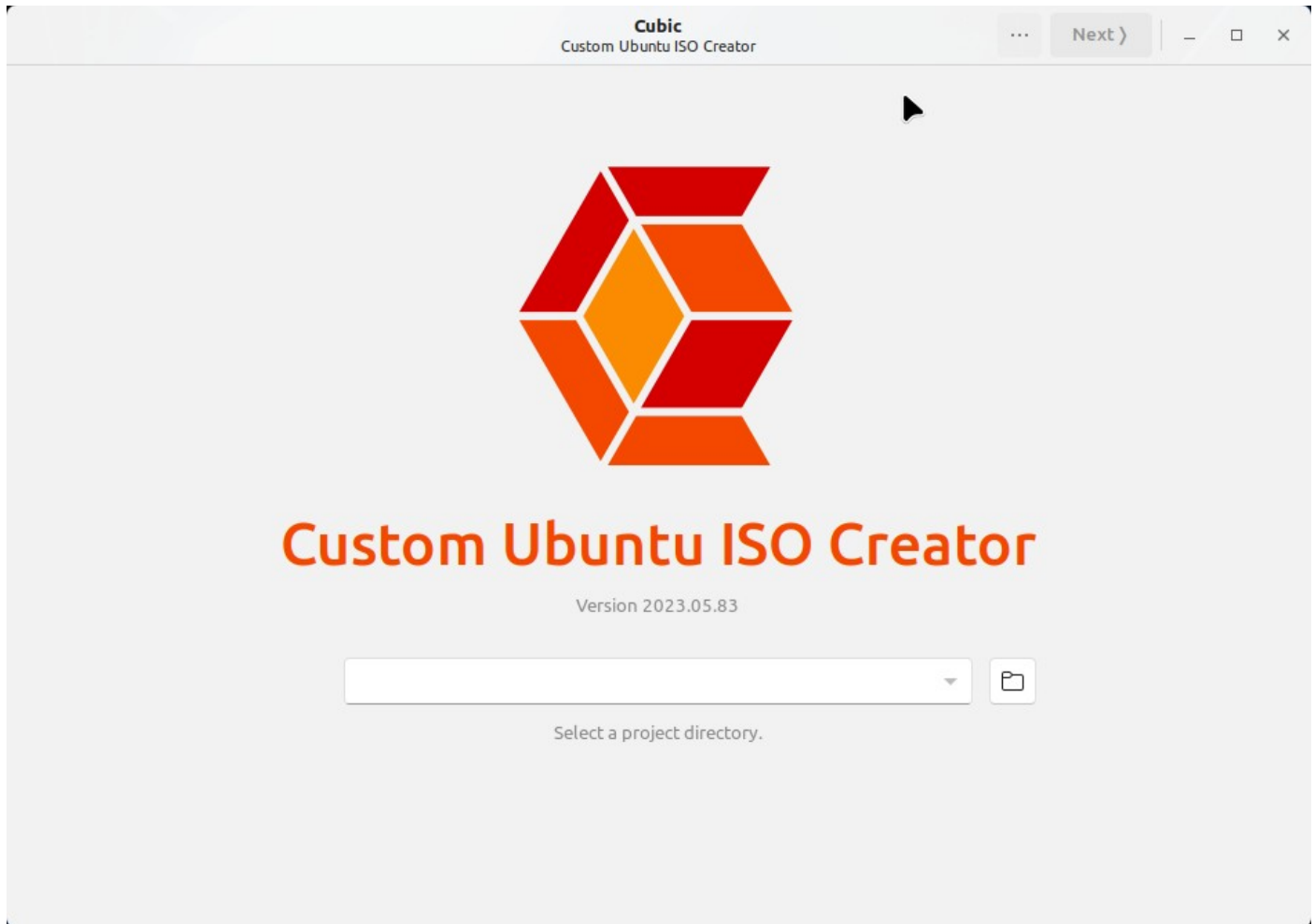
Nun überlegt man sich Folgendes. Man lädt Linux (Mint) herunter, erstellt einen Live-USB und installiert mit diesem alle Systeme. Und dann muss man auf allen Systemen die gleiche Konfiguration vornehmen und die gleichen Anwendungen installieren.

Wie wäre es, wenn es einen Weg gäbe, diese sich wiederholenden Aufgaben zu vermeiden? Wie wäre es, ein modifiziertes ISO zu erstellen und dieses angepasste Linux Mint auf den Live-USB zu legen? Auf diese Weise wird die gleiche angepasste Linux-Distribution auf allen Systemen installiert.

Es gibt ein praktisches GUI-Tool namens Cubic, mit dem man seine Installationen auf die einfachste Art und Weise vorkonfigurieren kann.
(<https://github.com/PJ-Singh-001/Cubic>)

In diesem Artikel sollen die Funktionen dieses Tools aufgezeigt werden und wie man es verwenden kann, um die Original Linux Mint 21.2 ISO anzupassen.

Cubic: Der einfachste Weg zu benutzerdefinierten ISO-Abbildern



Cubic (Custom Ubuntu ISO Creator) ist ein GUI-Tool zum Anpassen von Ubuntu-basierten Distributionen.

Es bietet die Möglichkeit, in der chroot-Umgebung der ISO wesentliche Änderungen vorzunehmen und Pakete getrennt für die minimale und die normale Installation auszuwählen.

Doch bevor wir mit der Prozedur beginnen, wollen wir einen Blick auf einige Funktionen von Cubic werfen.

Die wichtigsten Funktionen von Cubic

- Einfach zu bedienende GUI.
- Ermöglicht es, Pakete für die minimale und die normale Installation getrennt auszuwählen.
- Es hat drei Registerkarten für folgende Zwecke:
 1. Registerkarte „Kernel“: Wenn man sich für die Installation mehrerer Kernel entschieden hat, kann man hier den Boot-Kernel angeben.
 2. Registerkarte „Preseed“: Von hier aus können Voreinstellungsdateien erstellt, bearbeitet und entfernt werden. Mit Voreinstellungsdateien kann man die Fragen, die normalerweise während der Installation gestellt werden, überspringen, da Voreinstellungsdateien den Prozess automatisieren!
 3. Registerkarte „Boot“: Wenn man die Boot-Parameter ändern möchte, kann man hier einstellen, wie die modifizierte ISO gebootet werden soll.
- Bietet verschiedene Komprimierungsformate, um modifizierte ISO-Dateien zu komprimieren.
- Enthält QEMU (Open-Source-Maschinenemulator und Virtualisierung), um modifizierte ISOs zu testen.

Klingt doch vielversprechend, oder?

Verwendung von Cubic zur Erzeugung eines benutzerdefinierten ISO-Abbildes

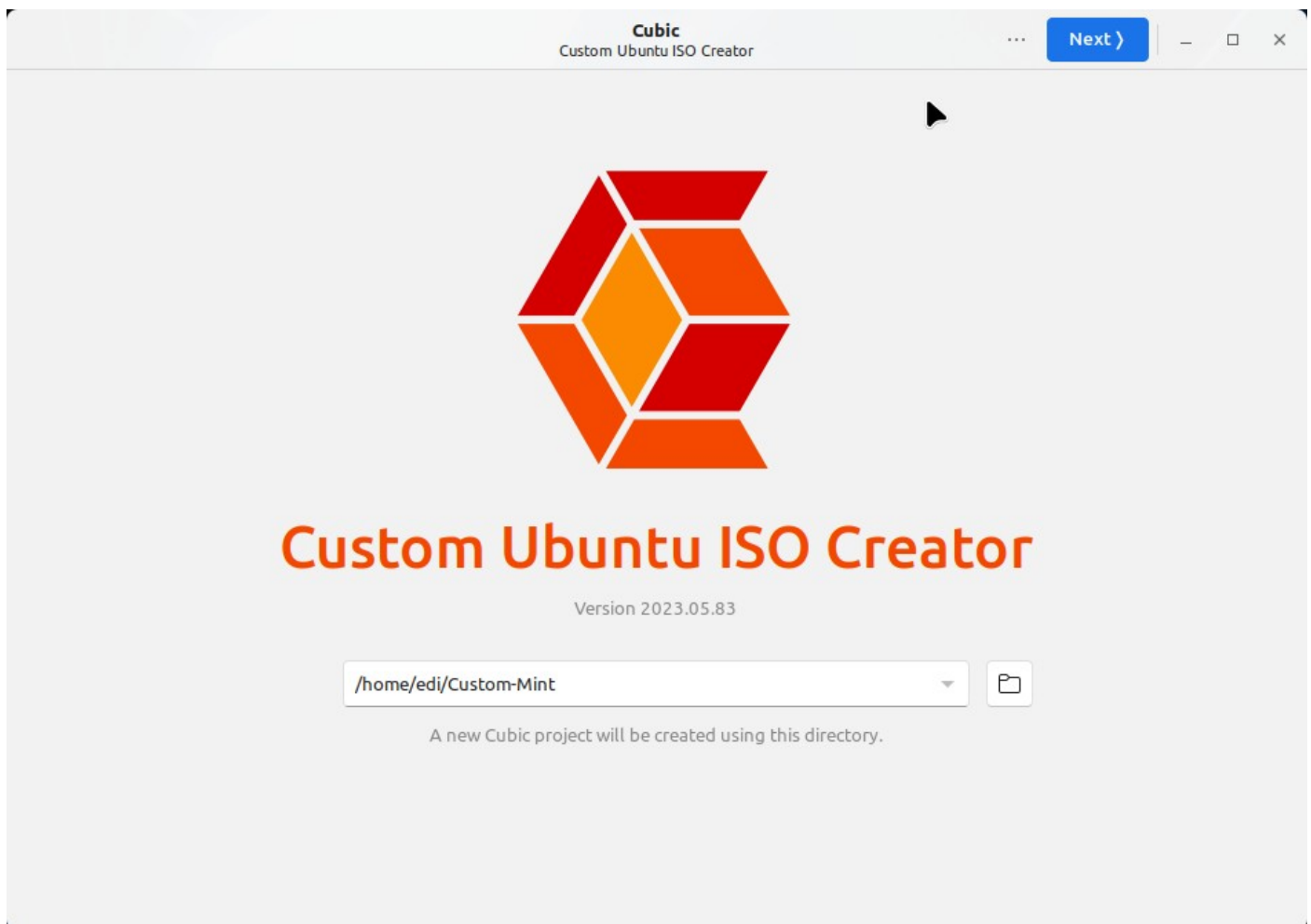
Vorgaben

- Es wird Linux Mint 21.2 Cinnamon-Edition (als ISO Datei) verwendet
- Es sollen folgende Änderungen vorgenommen werden:
 - Entfernen (für uns) unnötiger Pakete, so dass es sich um eine Minimal-ISO von Linux Mint handelt.
 - Aktualisierung der verbleibenden Pakete auf die neuesten Versionen.
 - Hinzufügen neuer Hintergrundbilder.
 - Installieren des neuesten stabilen Linux-Kernels (Stand 20.07.23 ist das 6.4).

Man kann natürlich auch weitere, eigene Modifikationen nach Wunsch vornehmen.

Hinzufügen/Erzeugen eines Projektverzeichnisses

Wenn man Cubic startet, wird man aufgefordert, das Projektverzeichnis zu wählen. Das ist nichts anderes als der Ort, an dem alle Dateien für die Erstellung benutzerdefinierter ISO-Abbilder abgelegt werden.



Praktischerweise legt man ein neues Unterverzeichnis im eigenen /home-Verzeichnis an. Man sollte darauf achten, dass für jedes Cubic-Projekt ein eigenes Verzeichnis angelegt wird. Wenn man mit der Auswahl des Projektverzeichnisses fertig ist, klickt man auf die Schaltfläche Weiter.

Auswahl des Original ISO-Abbildes

Hier muss man das Original-ISO-Abbild auswählen, mit dem man arbeiten möchte.

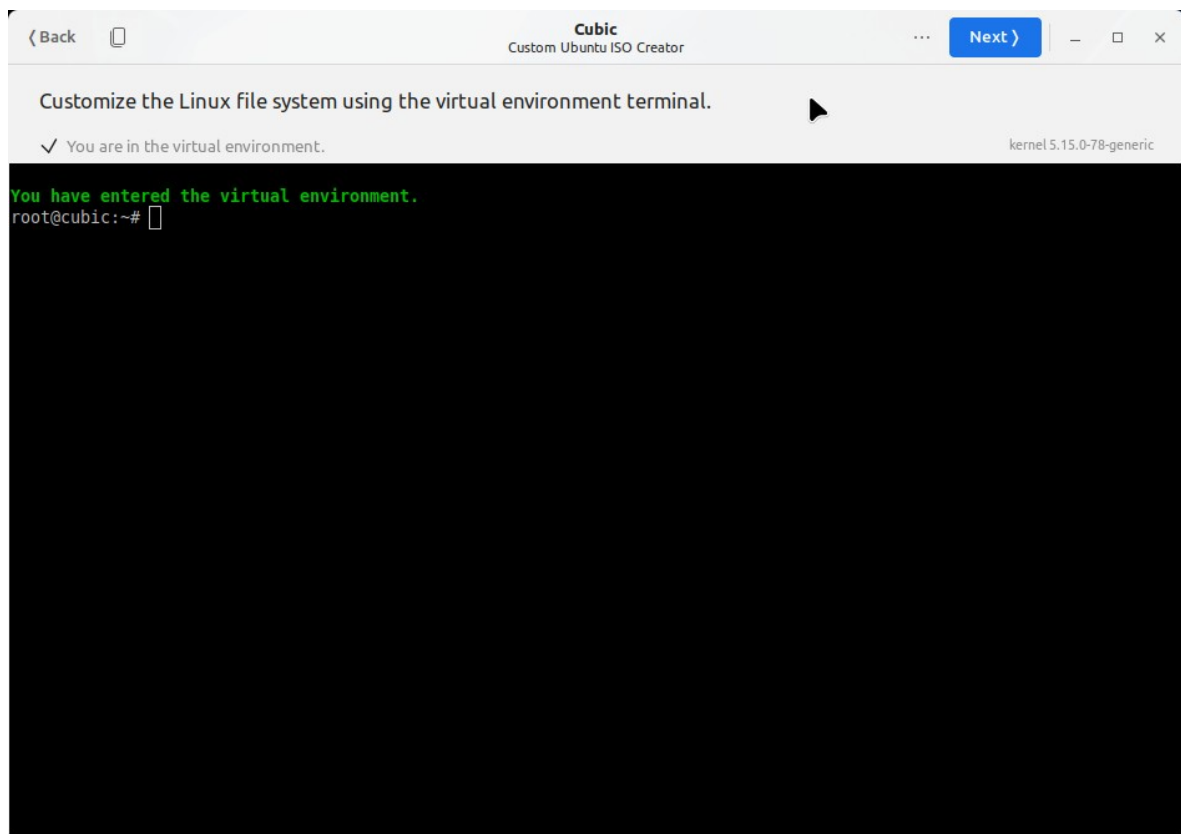
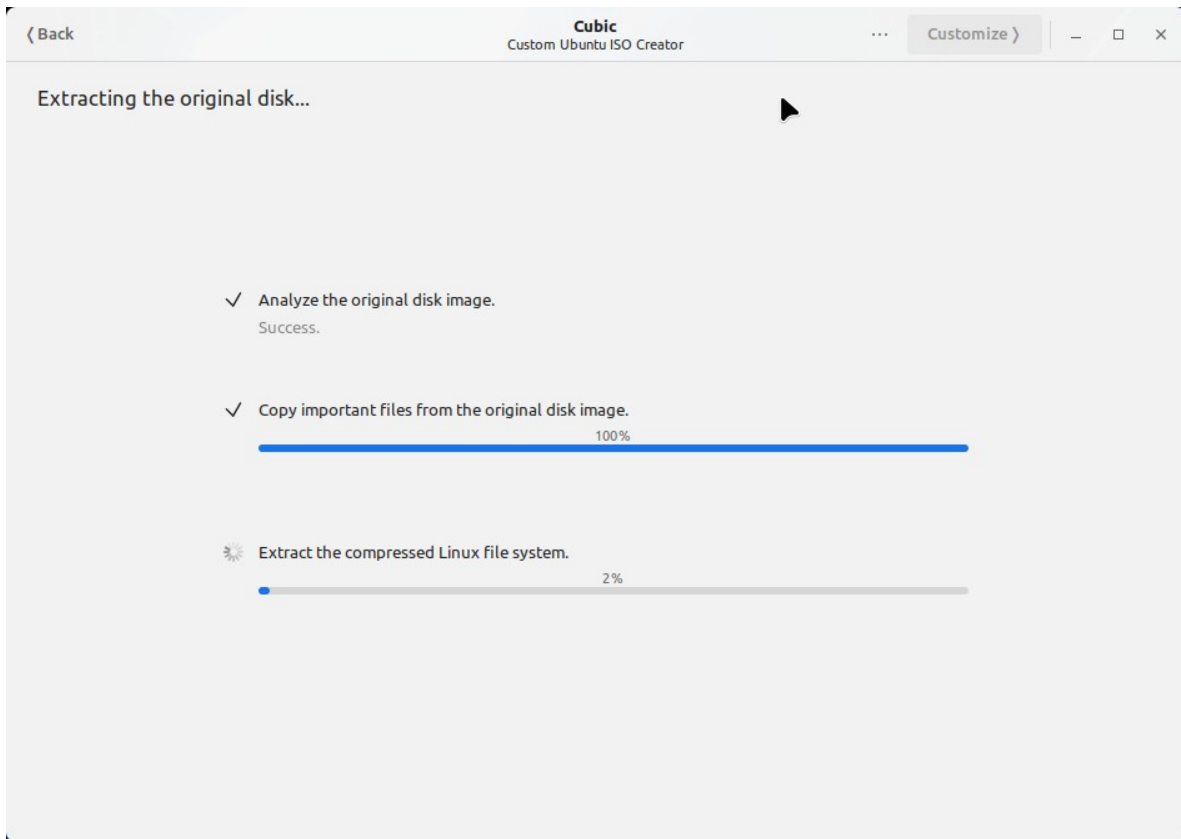
The screenshot shows the 'Cubic Custom Ubuntu ISO Creator' window. The main instruction is 'Select the original disk image to customize.' Below this, there are two columns of configuration options: 'Original Disk...' and 'Custom Disk...'. Each option has a text input field, a folder icon, and a checkmark indicating it is set.

Original Disk...	Custom Disk...
Version	2023.08.01
Filename	linuxmint-21.2-cinnamon-64bit.iso
Directory	/home/edi/Downloads
Volume ID	Linux Mint 21.2 Cinnamon 64-bit
Release	Victoria
Disk Name	Linux Mint 21.2 "Victoria" - Release amd64
Release URL	http://www.linuxmint.com/rel_victoria_cinnamon.4
	OS Release <input checked="" type="checkbox"/> Update the release description.

Wie man im Screenshot sieht, wird für diesen Artikel Linux Mint 21.2 verwendet, und das Werkzeug hat aus dieser Angabe alle anderen Informationen automatisch geholt.

Arbeiten mit der chroot-Umgebung

Klickt man dann auf „Next“, wird die ISO entpackt und man gelangt in eine sogenannte chroot-Umgebung.



Diese verhält sich wie ein normales Terminal und sie ist auch der Ort, an dem die meisten Operationen im Zusammenhang mit den Anpassungen durchgeführt werden.

Entfernen und Hinzufügen von Paketen

Es liegt am Einsatzzweck, wie viele Pakete man entfernen oder hinzufügen möchte. Dieser Schritt kann auch komplett übersprungen werden, macht dann aber die Sinnhaftigkeit der ganzen Sache eher fraglich.

Um ein Paket hinzuzufügen oder zu entfernen, muss man nicht wie gewöhnlich „sudo“ verwenden. Man ist bereits „root“ in der virtuellen Umgebung.

Nimmt man also an, man wollte den Mailclient Thunderbird entfernen; dann sollte der folgende Befehl die Arbeit erledigen:

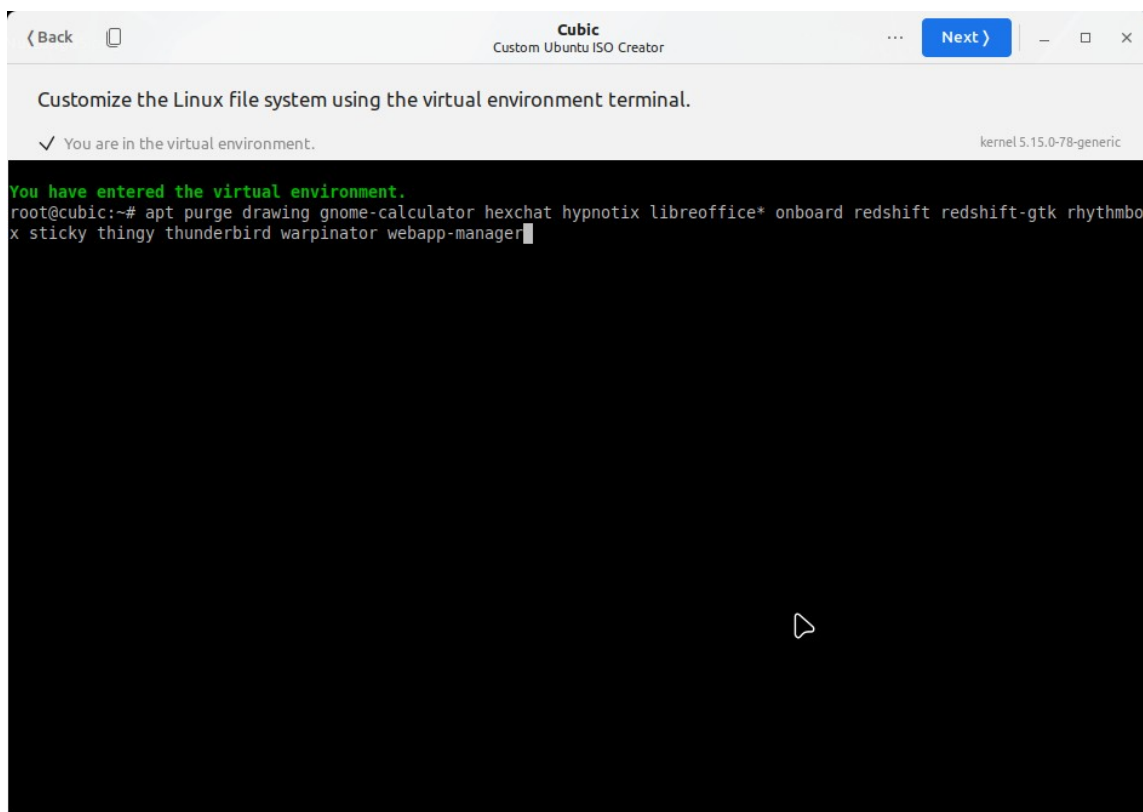
```
apt purge thunderbird
```

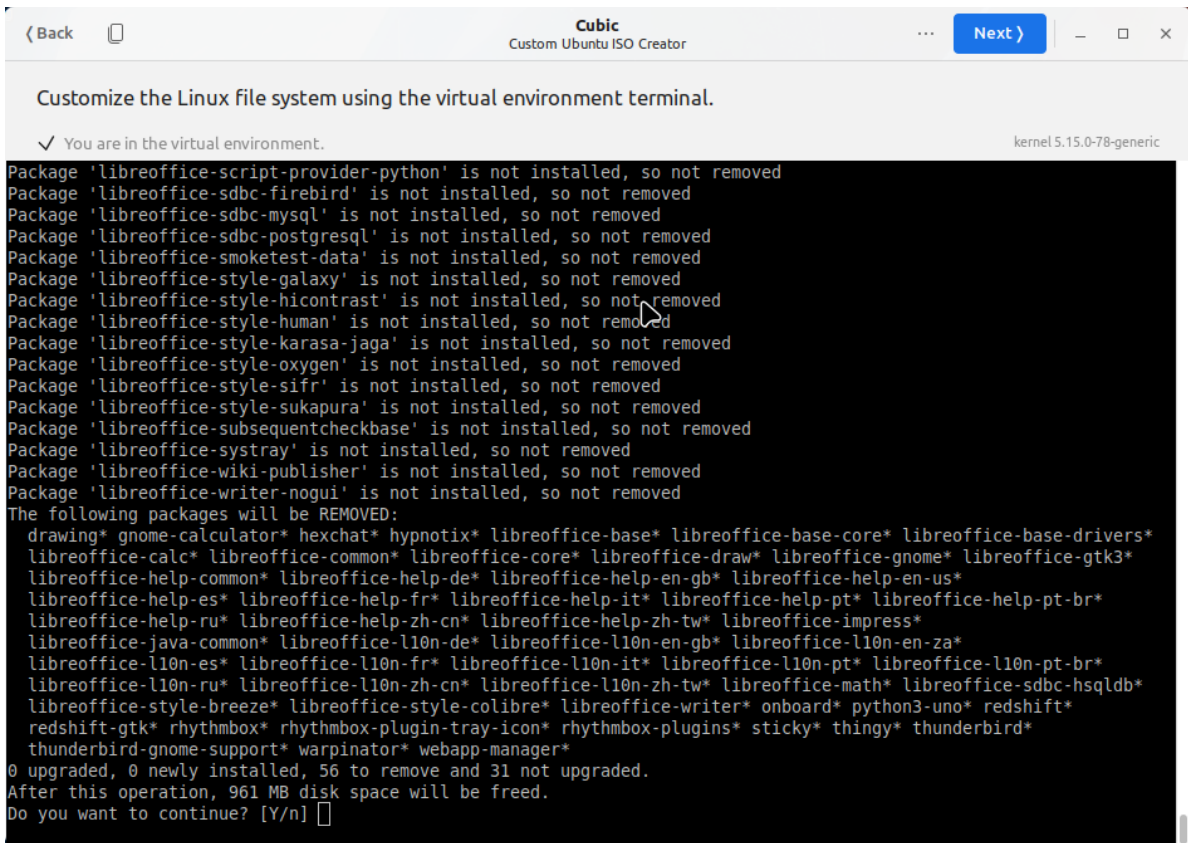
Hat man Interesse daran, dem Nutzer von Anfang an „conky“ bereit zu stellen, installiert man dieses Paket einfach nach:

```
apt install conky-all
```

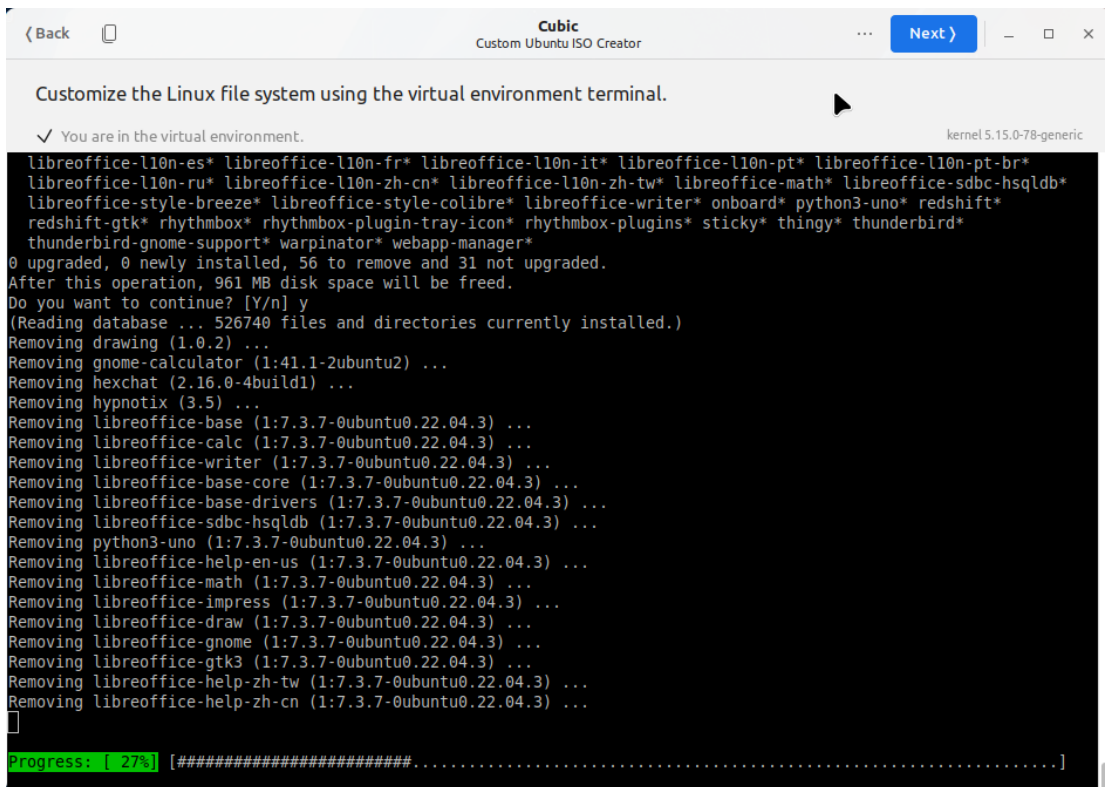
Für diese Anleitung wird ein minimaler Desktop angestrebt und es werden die meisten der zusätzlichen GUI-Anwendungen, einschließlich LibreOffice, entfernt. Nachfolgend der Befehl mit den entsprechenden Paketen:

```
apt purge drawing gnome-calculator hexchat hypnotix  
libreoffice* onboard redshift redshift-gtk rhythmbox sticky  
thingy thunderbird warpinator webapp-manager
```





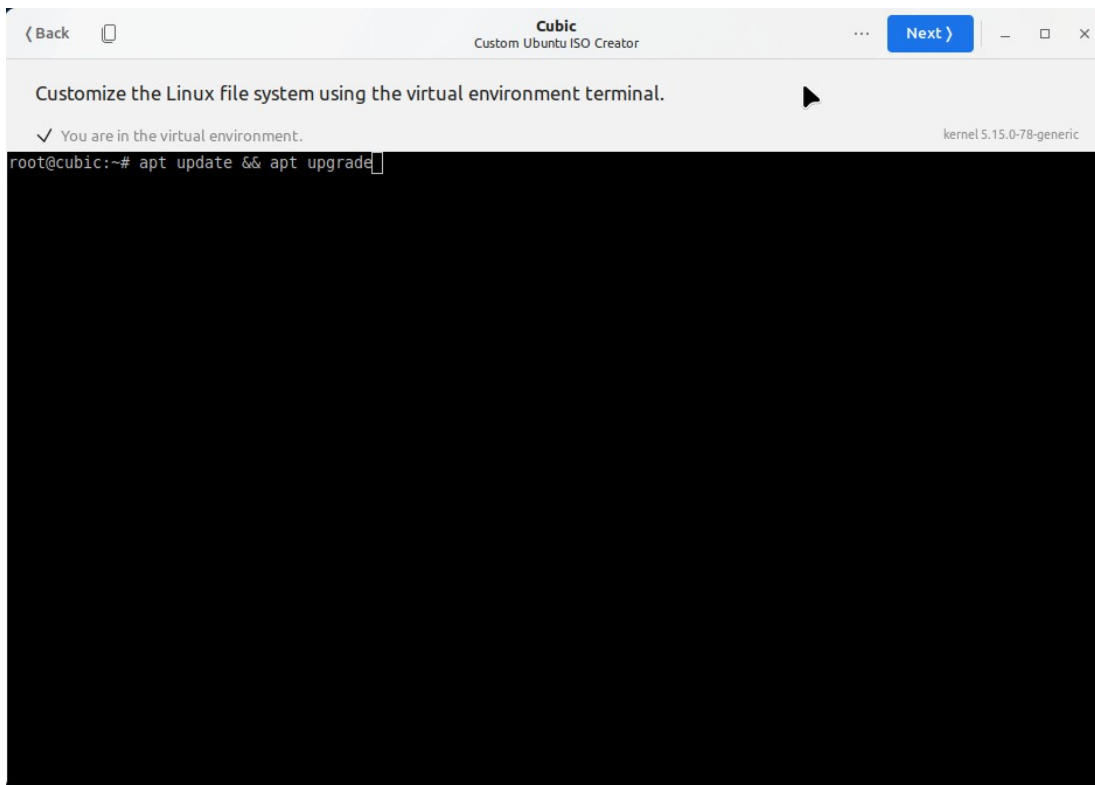
Beim Entfernen oder Hinzufügen von Paketen ist natürlich der entsprechende Anwendungsfall zu berücksichtigen. Ja nach Verwendung der benutzerdefinierten ISO stehen hier quasi alle standardmäßig vorinstallierten Pakete zur Disposition.



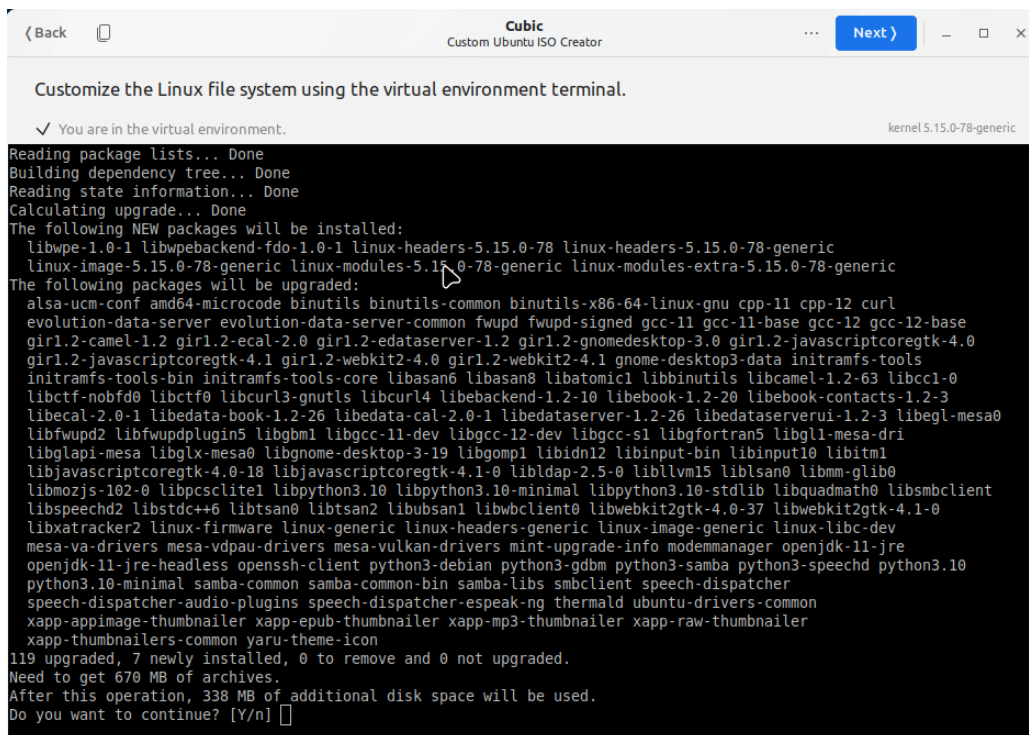
Aktualisieren bestehender Pakete

Wenn man bei der nächsten Ubuntu- oder Mint-Installation die aktuellsten Pakete haben möchte, sollte man diesen Schritt befolgen:

```
apt update && apt upgrade
```



The screenshot shows a terminal window titled "Cubic Custom Ubuntu ISO Creator". The terminal prompt is "root@cubic:~#". The command "apt update && apt upgrade" has been entered and is highlighted by the cursor. The window title bar includes "Back", "Cubic Custom Ubuntu ISO Creator", and "Next >".



The screenshot shows the terminal window after the command has been executed. The output displays the following information:

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following NEW packages will be installed:
 libwpe-1.0-1 libwpebackend-fdo-1.0-1 linux-headers-5.15.0-78 linux-headers-5.15.0-78-generic
 linux-image-5.15.0-78-generic linux-modules-5.15.0-78-generic linux-modules-extra-5.15.0-78-generic
The following packages will be upgraded:
 alsa-ucm-conf amd64-microcode binutils binutils-common binutils-x86-64-linux-gnu cpp-11 cpp-12 curl
 evolution-data-server evolution-data-server-common fwupd fwupd-signed gcc-11 gcc-11-base gcc-12 gcc-12-base
 gir1.2-camel-1.2 gir1.2-ecal-2.0 gir1.2-edataserver-1.2 gir1.2-gnomedesktop-3.0 gir1.2-javascriptcoregtk-4.0
 gir1.2-javascriptcoregtk-4.1 gir1.2-webkit2-4.0 gir1.2-webkit2-4.1 gnome-desktop3-data initramfs-tools
 initramfs-tools-bin initramfs-tools-core libasan6 libasan8 libatomic1 libbinutils libcamel-1.2-63 libcc1-0
 libctf-nobfd0 libctf0 libcurl3-gnutls libcurl4 libebackend-1.2-10 libebook-1.2-20 libebook-contacts-1.2-3
 libecal-2.0-1 libedata-book-1.2-26 libedata-cal-2.0-1 libedataserver-1.2-26 libedataserverui-1.2-3 libegl-mesa0
 libfwupd2 libfwupdplugin5 libgbm1 libgcc-11-dev libgcc-12-dev libgcc-s1 libgfortran5 libgl1-mesa-dri
 libglapi-mesa libglx-mesa0 libgnome-desktop-3-19 libgomp1 libidn2 libinput-bin libinput10 libitm1
 libjavascriptcoregtk-4.0-18 libjavascriptcoregtk-4.1-0 libldap-2.5-0 libllvm15 liblsan0 libmm-glib0
 libmozjs-102-0 libpcsc-lite1 libpython3.10 libpython3.10-minimal libpython3.10-stdlib libquadmath0 libsmbclient
 libspeechd2 libstdc++6 libtsan0 libtsan2 libubsan1 libwbclient0 libwebkit2gtk-4.0-37 libwebkit2gtk-4.1-0
 libxatracker2 linux-firmware linux-generic linux-headers-generic linux-image-generic linux-libc-dev
 mesa-va-drivers mesa-va-drivers mesa-vulkan-drivers mint-upgrade-info modemmanager openjdk-11-jre
 openjdk-11-jre-headless openssh-client python3-debian python3-gdbm python3-samba python3-speechd python3.10
 python3.10-minimal samba-common samba-common-bin samba-libs smbclient speech-dispatcher
 speech-dispatcher-audio-plugins speech-dispatcher-espeak-ng thermd ubuntu-drivers-common
 xapp-appimage-thumbnailer xapp-epub-thumbnailer xapp-mp3-thumbnailer xapp-raw-thumbnailer
 xapp-thumbnailers-common yaru-theme-icon
119 upgraded, 7 newly installed, 0 to remove and 0 not upgraded.
Need to get 670 MB of archives.
After this operation, 338 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

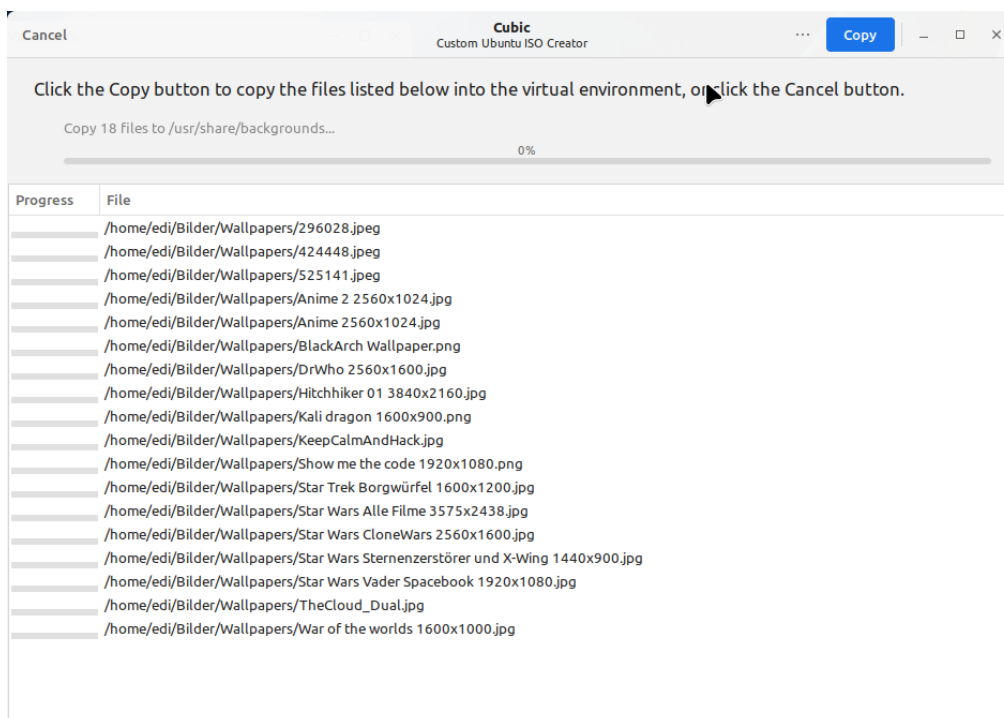
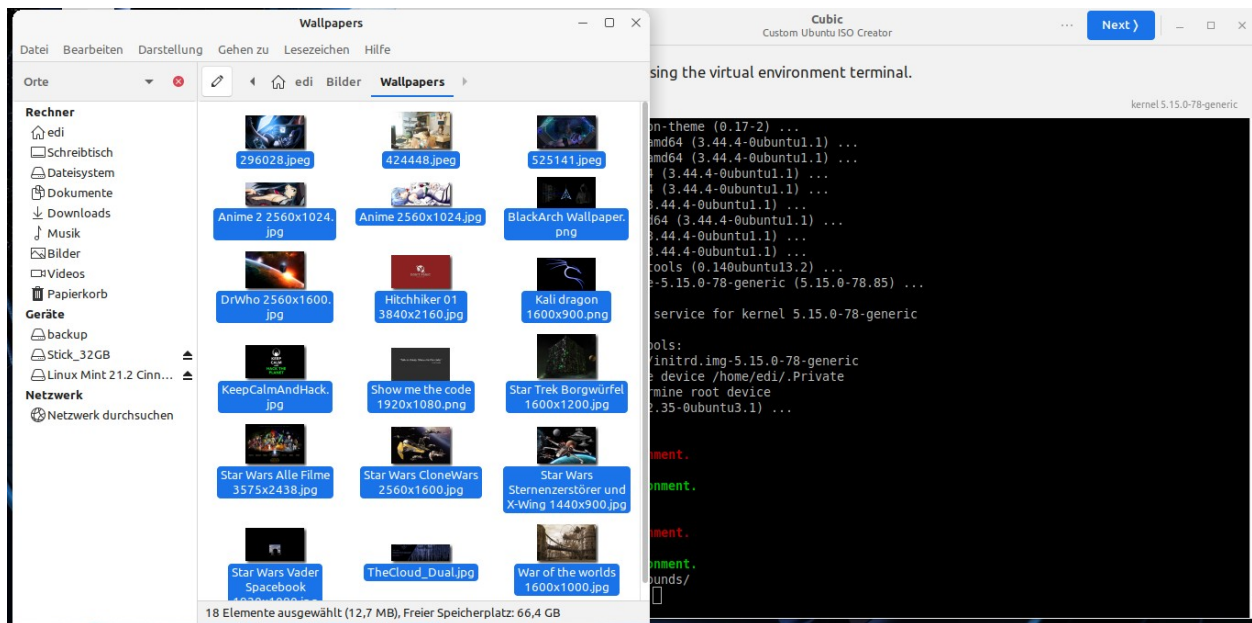
Dann ist dieser Vorgang abgeschlossen und es kann die nächste Anpassung erfolgen.

Hintergrundbilder hinzufügen

Dazu wechselt man zunächst in das aktuelle Arbeitsverzeichnis /usr/share/backgrounds:

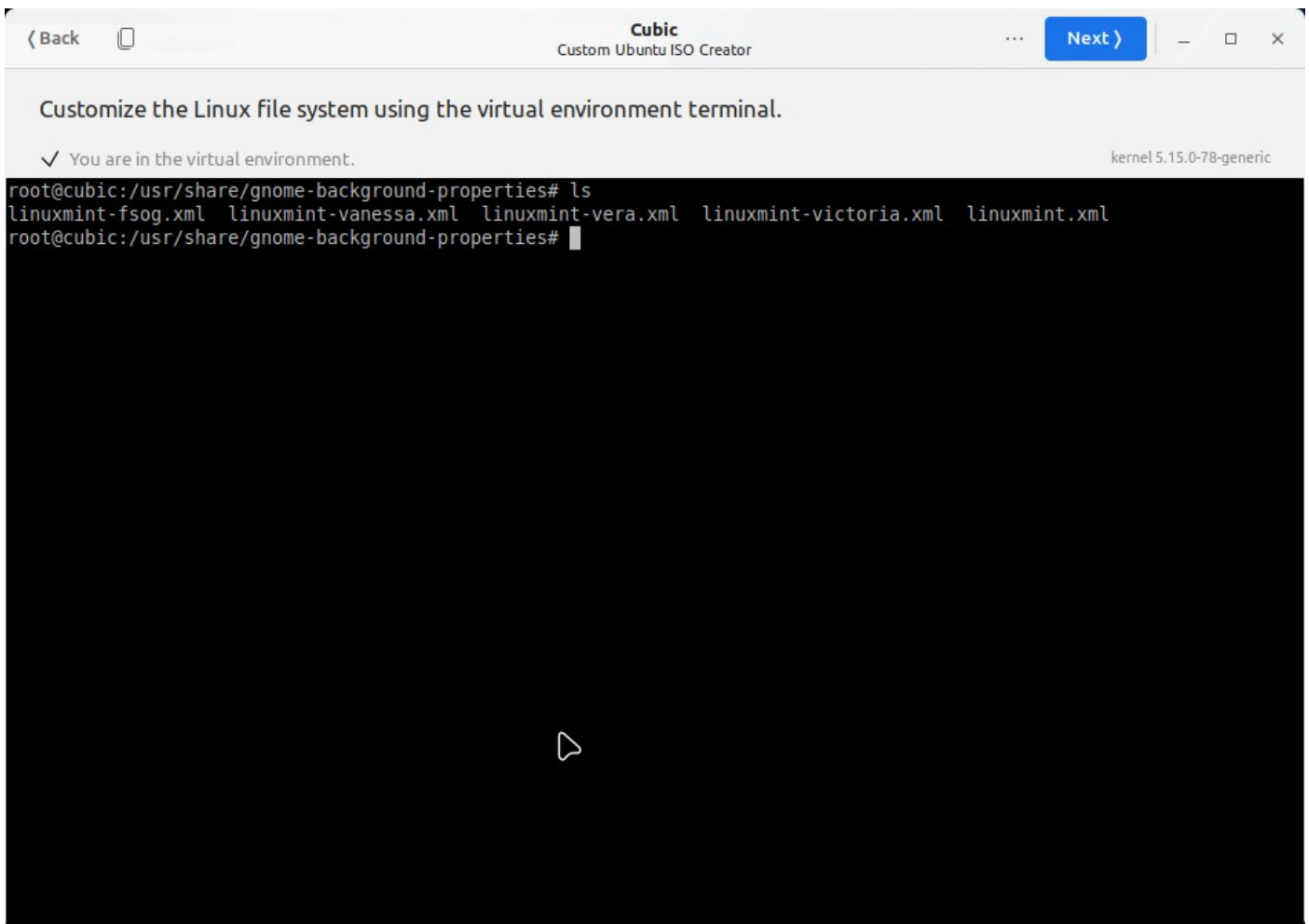
```
cd /usr/share/backgrounds
```

Möchte man ein eigenes Verzeichnis, legt man dieses an und wechselt hinein. Da Cubic das Kopieren von Dateien per Drag & Drop in das aktuelle Arbeitsverzeichnis (in unserem Fall /usr/share/backgrounds) ermöglicht, kann man hier die gewünschten Hintergrundbilder (die im lokalen „echten“ System vorliegen) einfach per Maus in das chroot-Fenster ziehen. Danach den Schalter „Copy“ klicken.



Das ist die Vorgehensweise, wenn man den Hintergrund vom Dateimanager aus ändern will.

Wenn man aber möchte, dass sie im Dialogfeld zum Ändern des Hintergrunds aufgelistet werden, muss man jeden einzelnen in einer XML-Datei hinzufügen, die sich unter `/usr/share/gnome-background-properties` befindet.



The screenshot shows the Cubic Custom Ubuntu ISO Creator application. The window title is "Cubic Custom Ubuntu ISO Creator". The main content area displays the instruction "Customize the Linux file system using the virtual environment terminal." Below this, a status bar indicates "You are in the virtual environment." and the kernel version "kernel 5.15.0-78-generic". A terminal window is open, showing the command `ls` being executed in the directory `/usr/share/gnome-background-properties`. The output of the command lists five XML files: `linuxmint-fsog.xml`, `linuxmint-vanessa.xml`, `linuxmint-vera.xml`, `linuxmint-victoria.xml`, and `linuxmint.xml`.

```
root@cubic:/usr/share/gnome-background-properties# ls
linuxmint-fsog.xml linuxmint-vanessa.xml linuxmint-vera.xml linuxmint-victoria.xml linuxmint.xml
root@cubic:/usr/share/gnome-background-properties#
```

Alternativ kann man auch eine eigene XML erstellen.

Linux-Kernel hinzufügen

In diesem Abschnitt werden die .deb-Pakete der Kernel-Header und -Module heruntergeladen und dann mit dem Befehl dpkg installiert.

Die nachfolgenden vier Befehle werden nacheinander verwendet, um die Kernel-Module und -Header herunterzuladen:

```
wget
https://kernel.ubuntu.com/~kernel-ppa/mainline/v6.4.3/amd64/linux-headers-6.4.3-060403_6.4.3-060403.202307110536_all.deb
```

```
wget
https://kernel.ubuntu.com/~kernel-ppa/mainline/v6.4.3/amd64/linux-headers-6.4.3-060403-generic_6.4.3-060403.202307110536_amd64.deb
```

```
wget
https://kernel.ubuntu.com/~kernel-ppa/mainline/v6.4.3/amd64/linux-image-unsigned-6.4.3-060403-generic_6.4.3-060403.202307110536_amd64.deb
```

```
wget
https://kernel.ubuntu.com/~kernel-ppa/mainline/v6.4.3/amd64/linux-modules-6.4.3-060403-generic_6.4.3-060403.202307110536_amd64.deb
```

Hinweis:

Stand der oberen Dateien: 01.08.2023

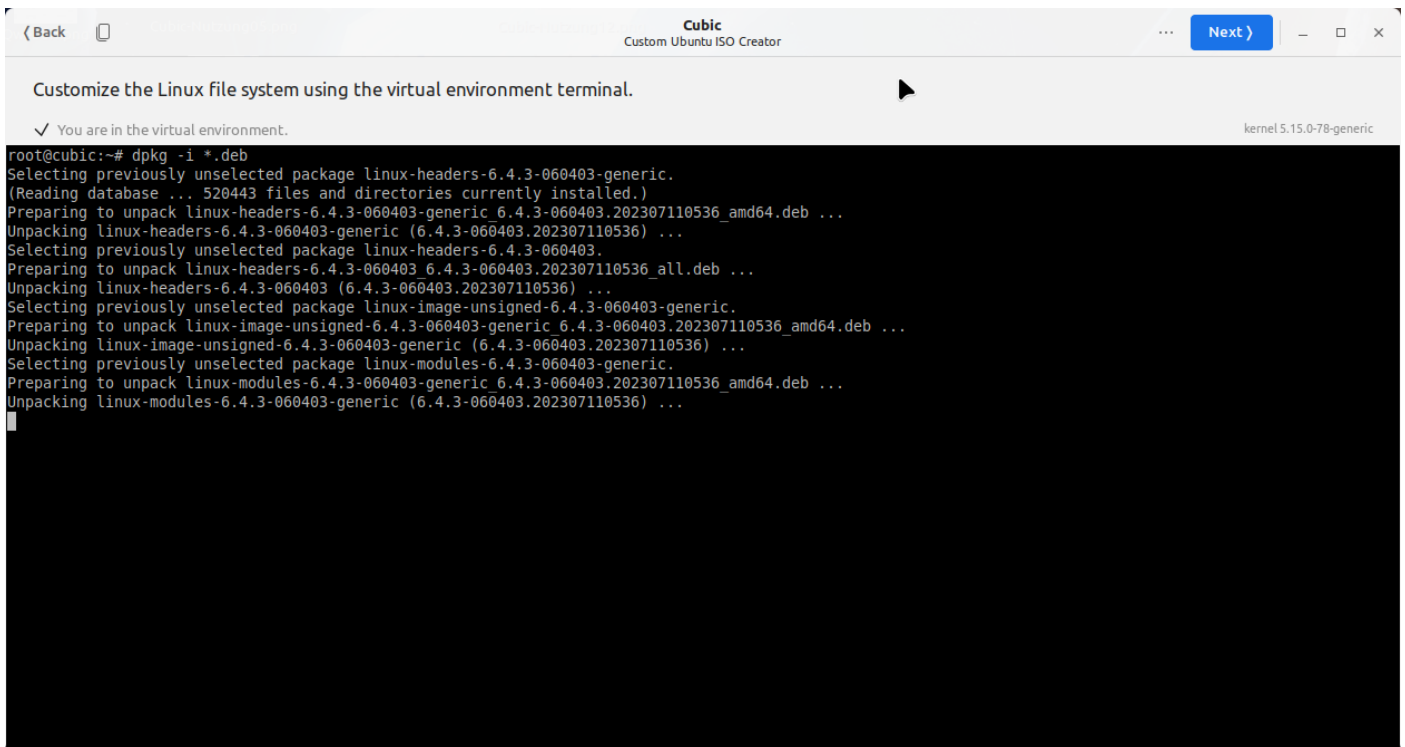
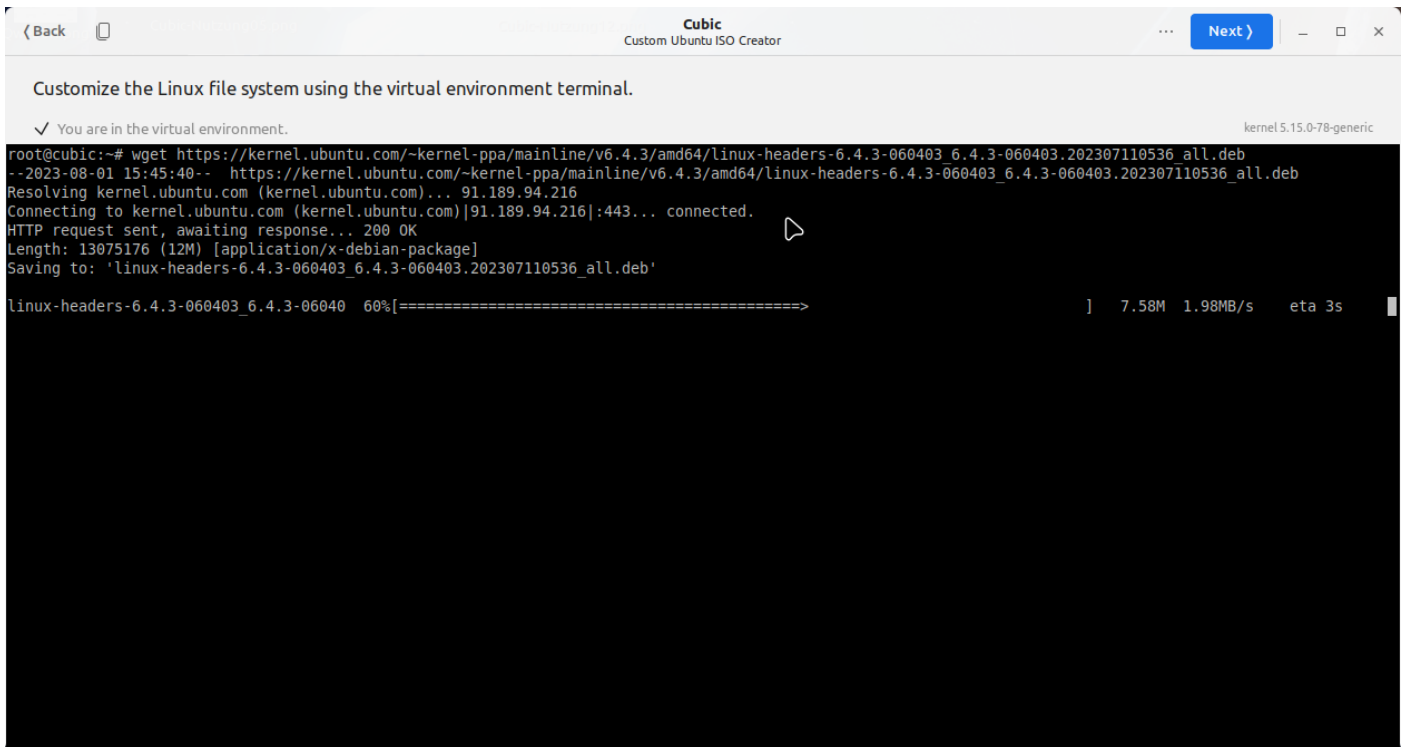
Um die aktuellsten Dateien zu verwenden, kann man unter

<https://kernel.ubuntu.com/~kernel-ppa/mainline/>

nachsehen, was die letzte stabile Version ist und in den obigen Befehlen dann die Verzeichnis- und Dateinamen entsprechend anpassen.

Sobald man mit dem Herunterladen fertig ist, kann man den Befehl dpkg verwenden, um sie zu installieren:

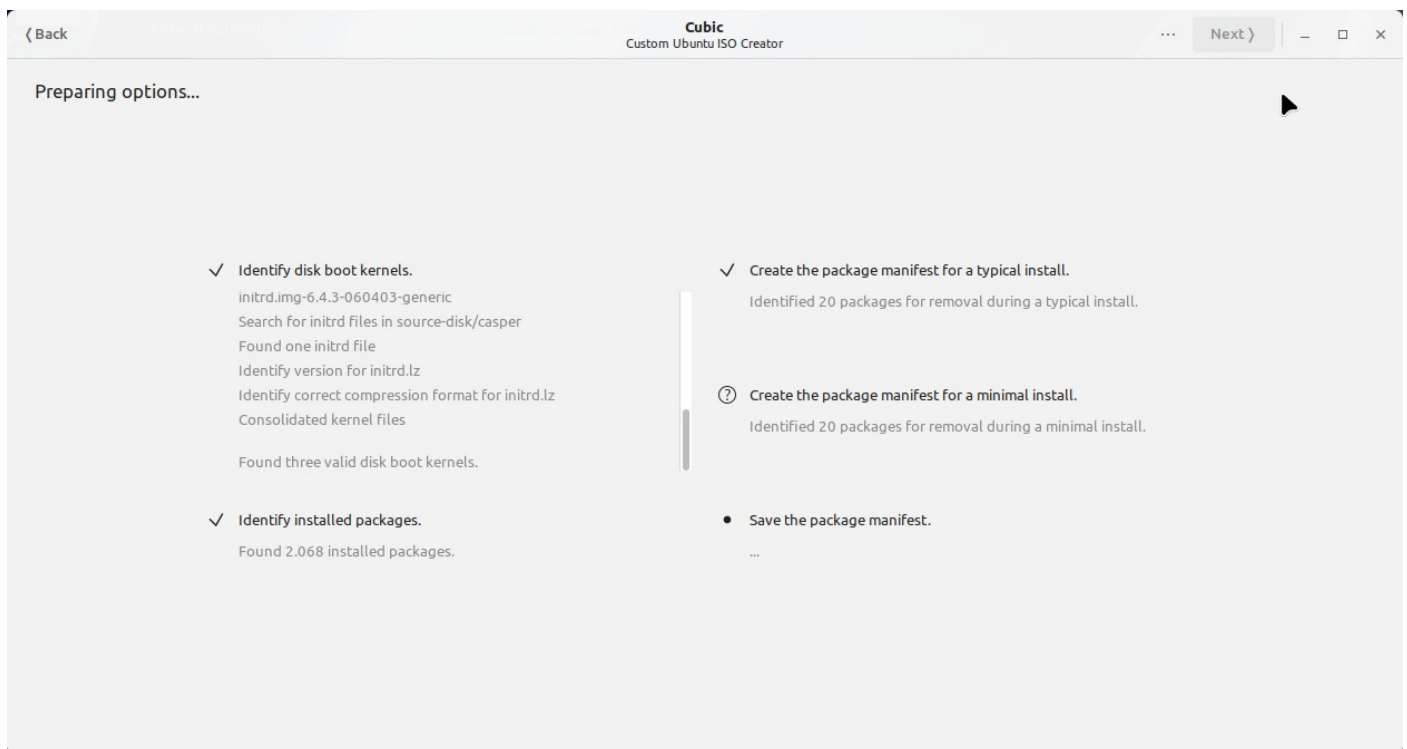
```
dpkg -i *.deb
```



Ist man hier zufrieden, geht es weiter mit Klick auf „Next“.

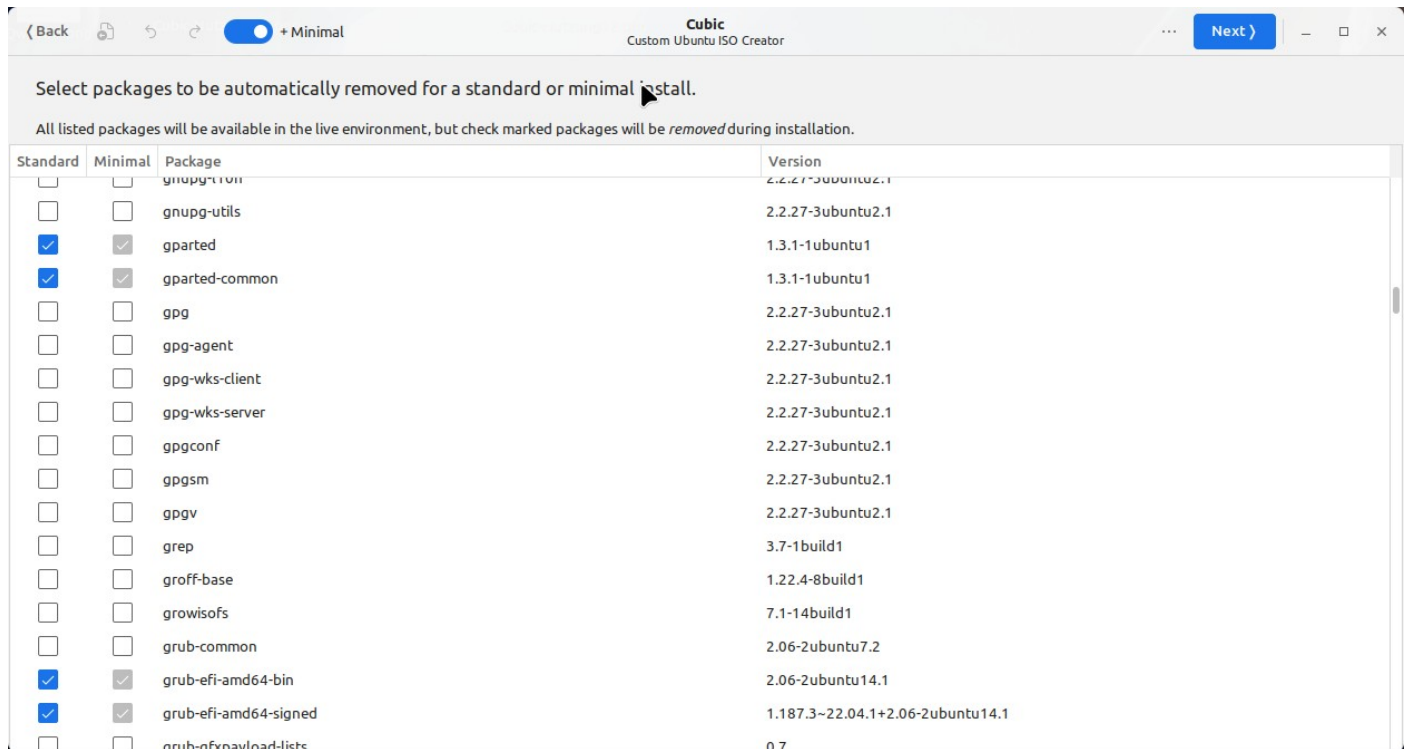
Zusammenstellung der Optionen

Dann bereitet Cubic die Anpassungen für den nächsten Schritt auf.



Zu entfernende Pakete für die Normale und Minimale Installation auswählen

Mit Cubic kann man sogar die Liste der Software sowohl für die minimale als auch für die normale Installation festlegen.



Hinweis: Alles was hier ausgewählt ist, wird bei der Installation entfernt!

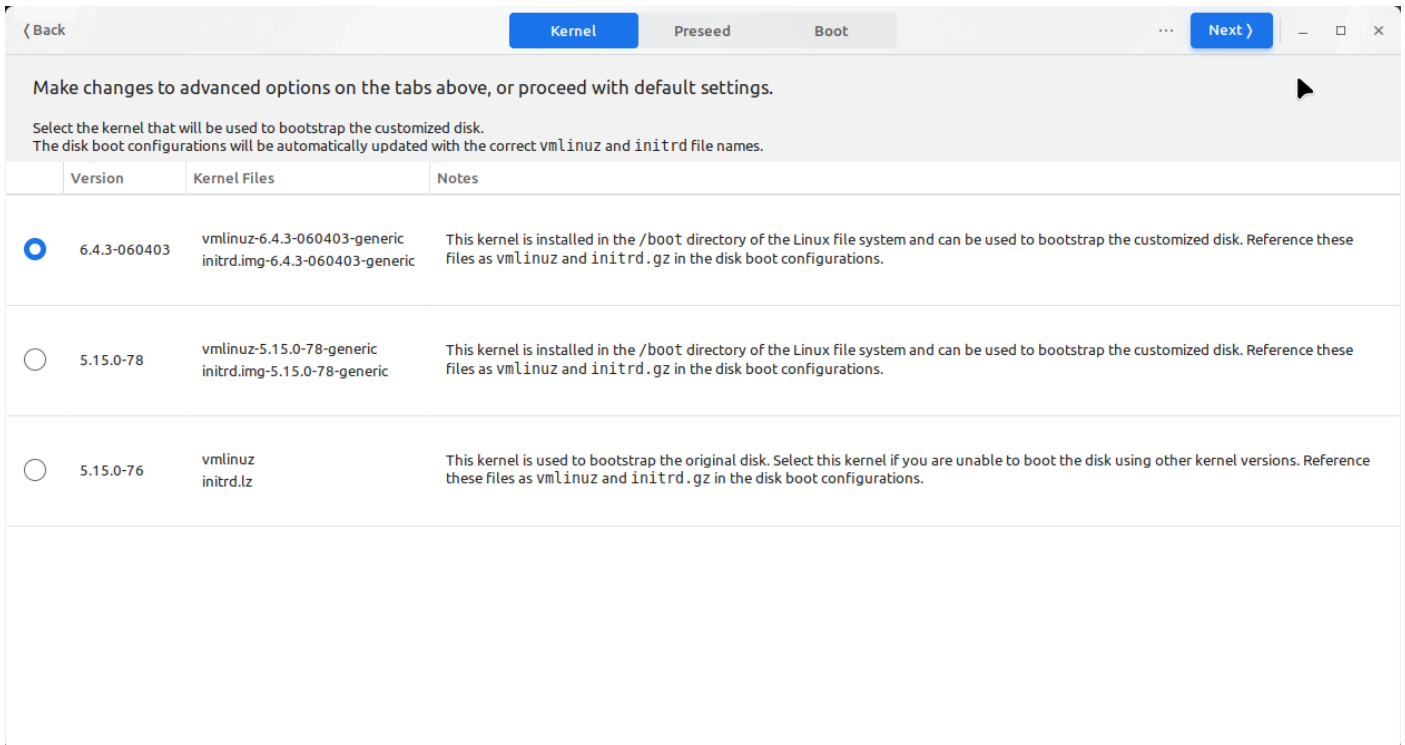
Hier kann man kreativ sein, denn man hat zwei Listen mit Software, die für verschiedene Zwecke verwendet werden können: Standard und Minimal

Man kann zum Beispiel die normale Installationsoption für die Verwendung auf dem Desktop verwenden, während die Minimalinstallation für minimalere Installationen, z. B. für Server, geeignet sein könnte.

Da im Abschnitt „Entfernen von Paketen“ bereits Pakete entfernt wurden, gibt es in diesem Abschnitt nicht mehr viel zu tun.

Boot-Kernel auswählen

Dieser Abschnitt zeigt die verfügbaren Linux-Kernel an.



Make changes to advanced options on the tabs above, or proceed with default settings.

Select the kernel that will be used to bootstrap the customized disk.
The disk boot configurations will be automatically updated with the correct `vmlinuz` and `initrd` file names.

Version	Kernel Files	Notes
<input checked="" type="radio"/> 6.4.3-060403	<code>vmlinuz-6.4.3-060403-generic</code> <code>initrd.img-6.4.3-060403-generic</code>	This kernel is installed in the <code>/boot</code> directory of the Linux file system and can be used to bootstrap the customized disk. Reference these files as <code>vmlinuz</code> and <code>initrd.gz</code> in the disk boot configurations.
<input type="radio"/> 5.15.0-78	<code>vmlinuz-5.15.0-78-generic</code> <code>initrd.img-5.15.0-78-generic</code>	This kernel is installed in the <code>/boot</code> directory of the Linux file system and can be used to bootstrap the customized disk. Reference these files as <code>vmlinuz</code> and <code>initrd.gz</code> in the disk boot configurations.
<input type="radio"/> 5.15.0-76	<code>vmlinuz</code> <code>initrd.lz</code>	This kernel is used to bootstrap the original disk. Select this kernel if you are unable to boot the disk using other kernel versions. Reference these files as <code>vmlinuz</code> and <code>initrd.gz</code> in the disk boot configurations.

Hinweis:

Hier werden aus folgenden Gründen drei Optionen angezeigt:

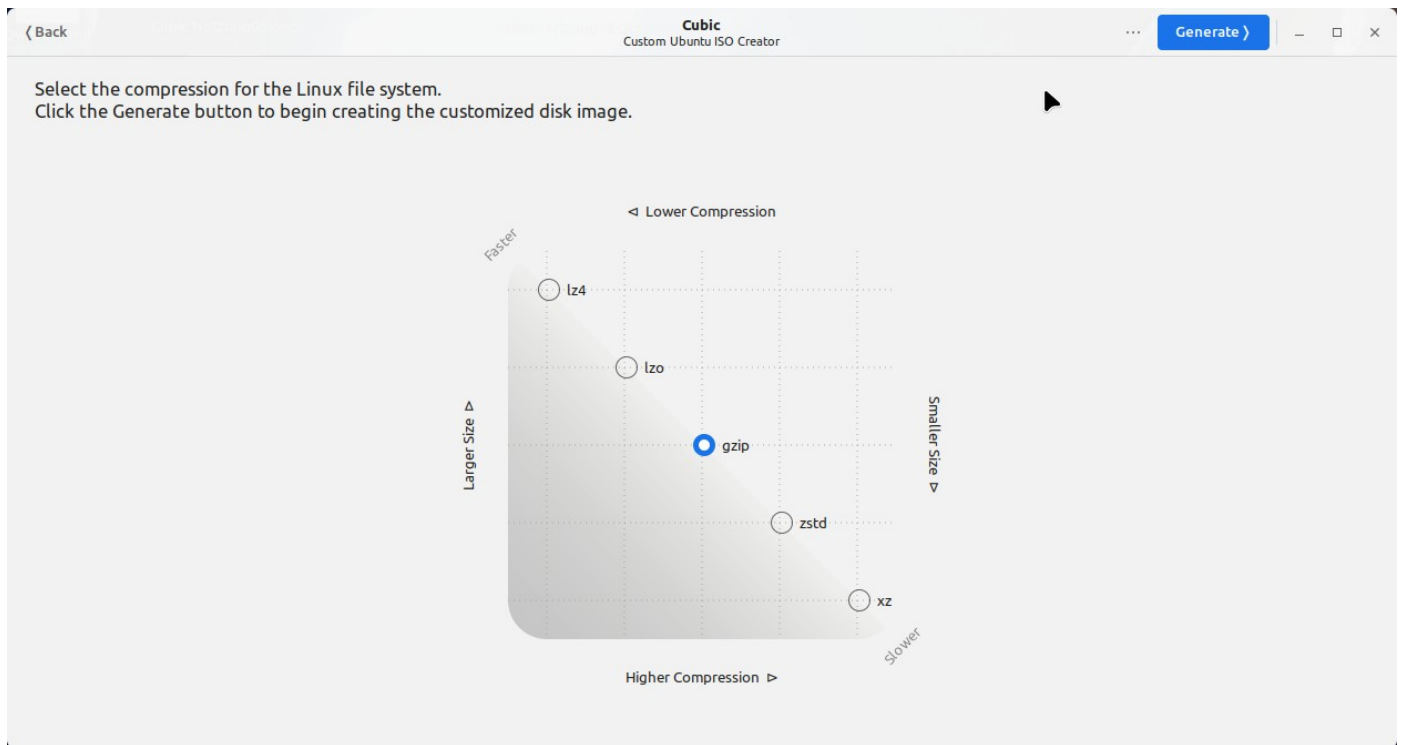
- 5.15.0-76 war vorinstalliert.
- 5.15.0-78 wurde installiert, als die bestehenden Pakete aktualisiert wurden.
- 6.4.3 wurde manuell installiert („Linux-Kernel hinzufügen“).

Für den Test entscheidet man sich hier für den Linux-Kernel 6.4.3

Nach Klick auf „Next“ geht es mit der Komprimierungsmethode weiter.

Auswahl des ISO Kompressionstyps

Dies ist der letzte Schritt der Konfiguration, bei dem man den Komprimierungsalgorithmus auswählen kann.

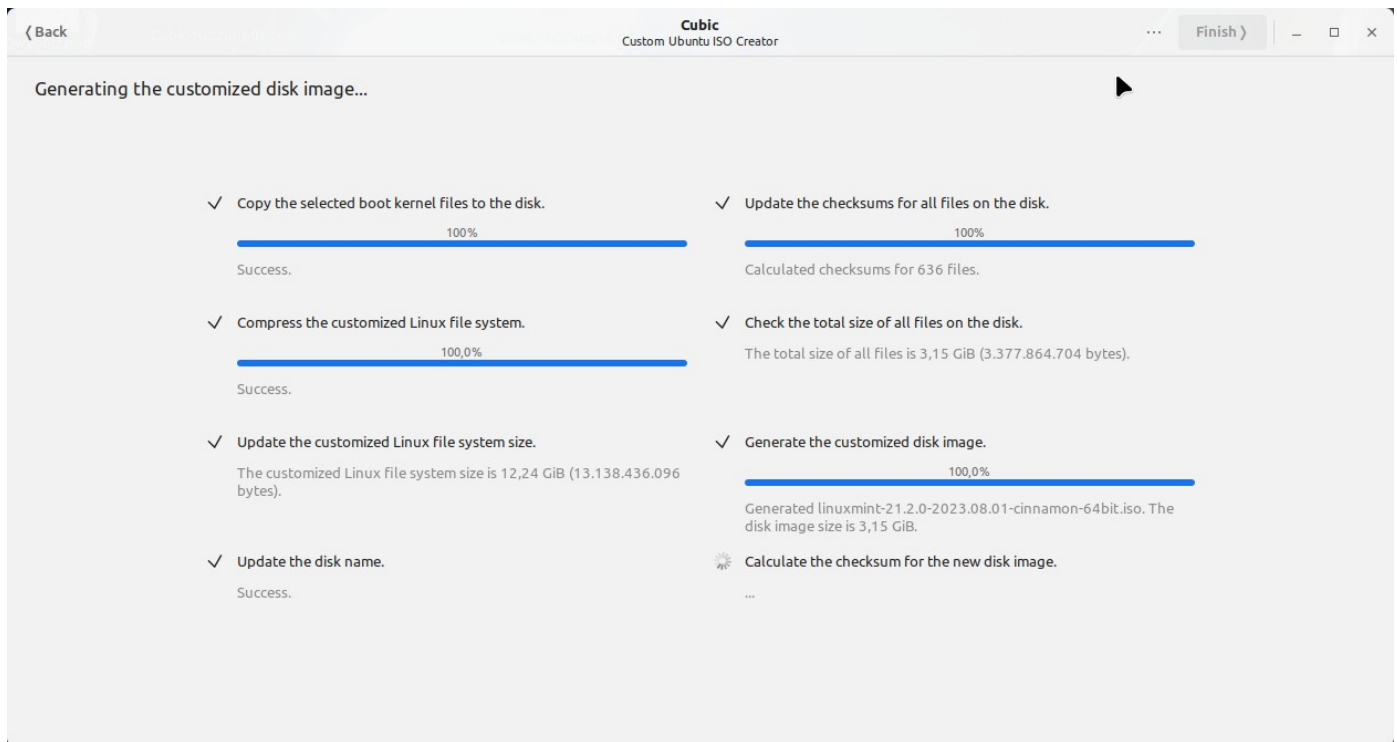


Man muss nicht alle Komprimierungsarten kennen, denn es gibt bereits ein Diagramm, mit dem man den Grad der Komprimierung auswählen kann.

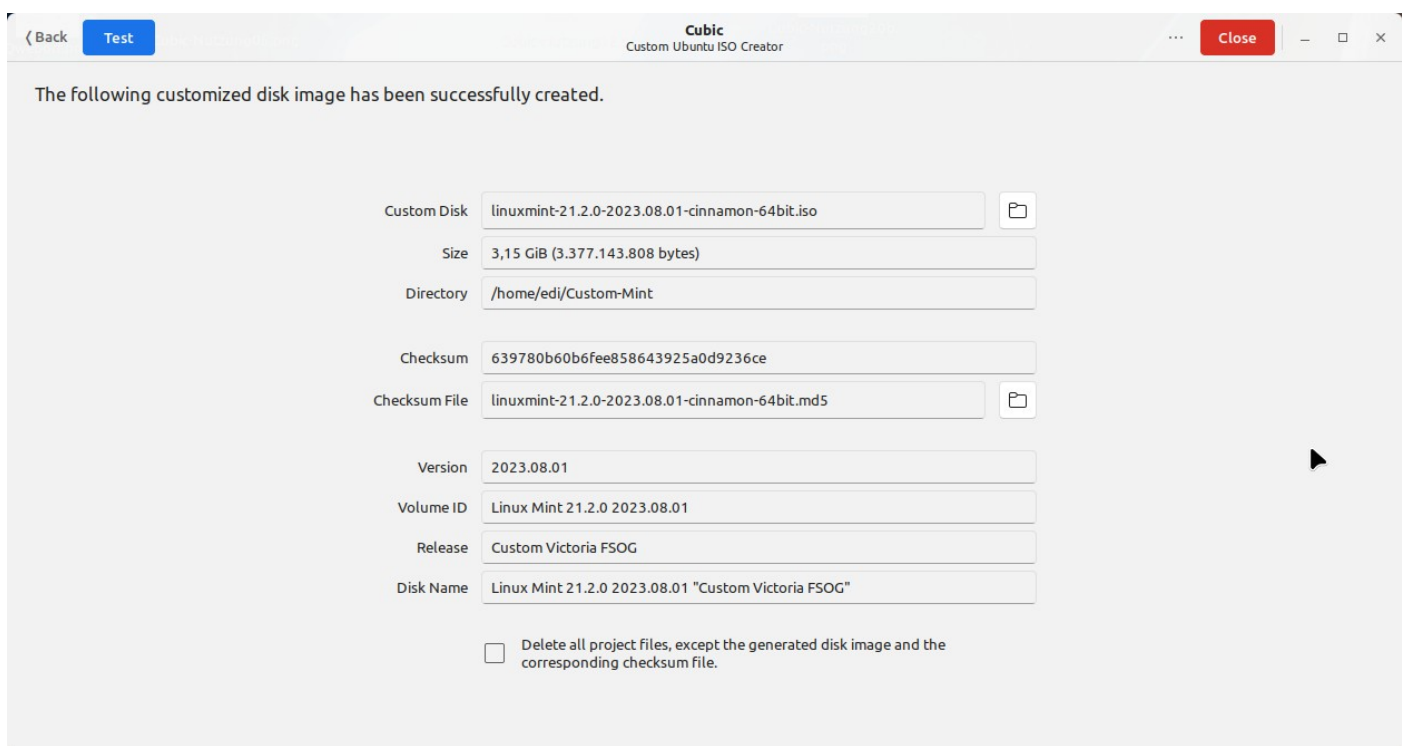
Für diesen Artikel wird die Standardoption „gzip“ verwendet.

Ist man mit der Auswahl des Kompressionstyps fertig, klickt man auf die Schaltfläche „Generate“ und die ISO wird erstellt.

Erzeugen der benutzerdefinierten ISO-Datei



Nach erfolgter Erzeugung gelangt man zu einer Übersicht, welche die wichtigsten Informationen zur ISO enthält.

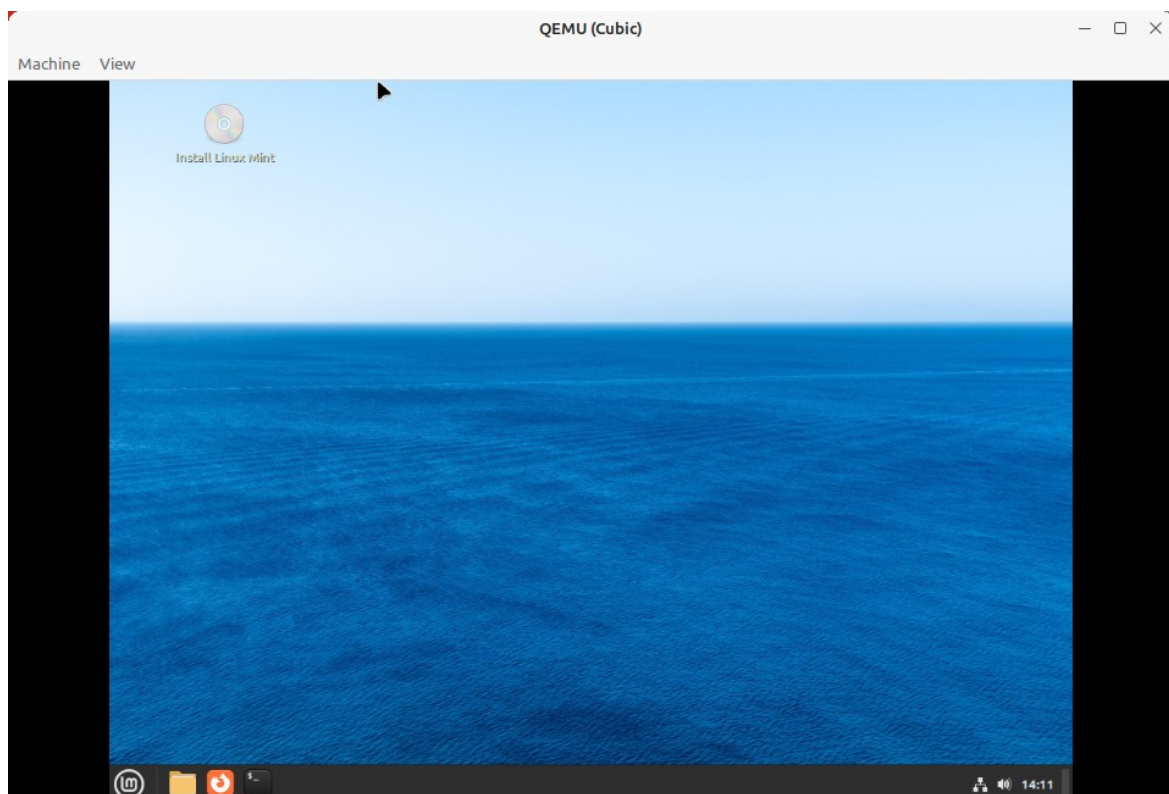
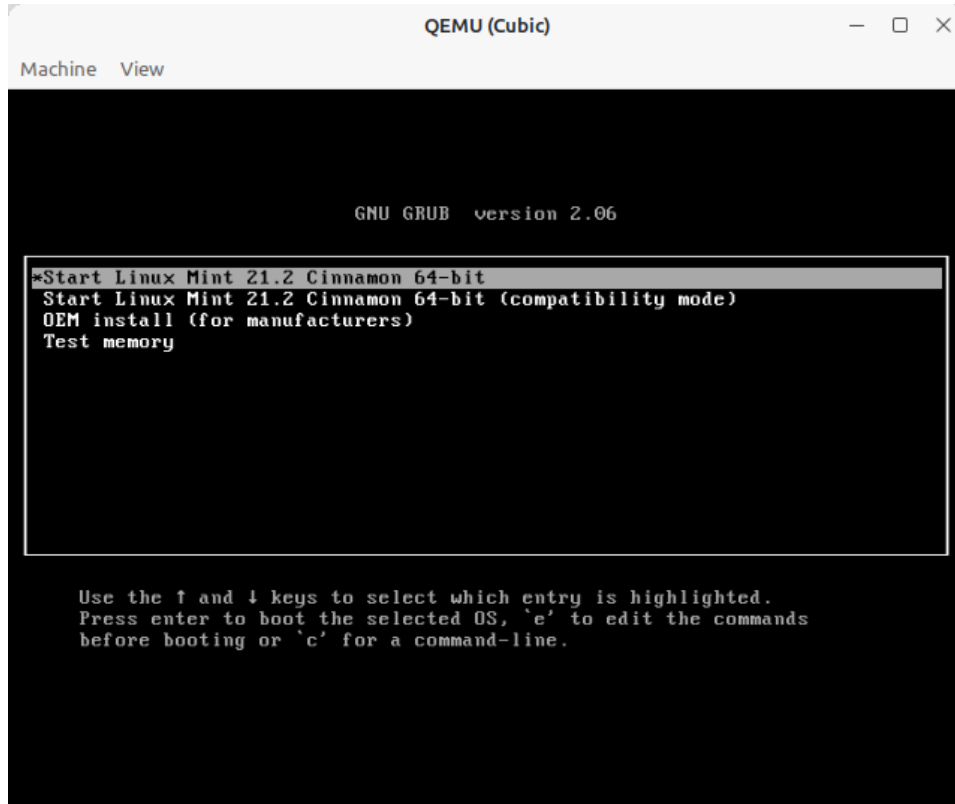


Die Prüfung und Erstellung der ISO-Datei dauert ungefähr 20 Minuten. Je nach verwendetem Host-System.

Testen der Abbilddatei mit dem integrierten QEMU

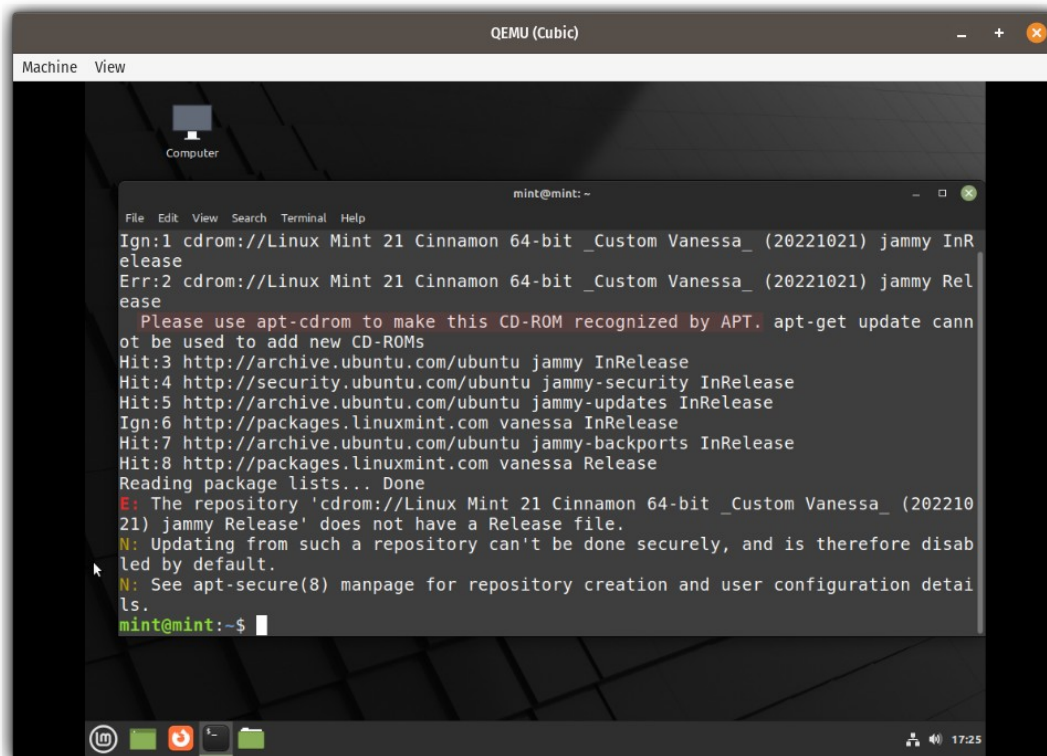
Sobald Cubic die angewiesene ISO-Datei erstellt hat, kann man diese testen. Hierfür klickt man auf die Schaltfläche „Test“ in der linken oberen Ecke der Übersichtsseite.

Dadurch startet der integrierte QEMU:



Problemlösung(en)

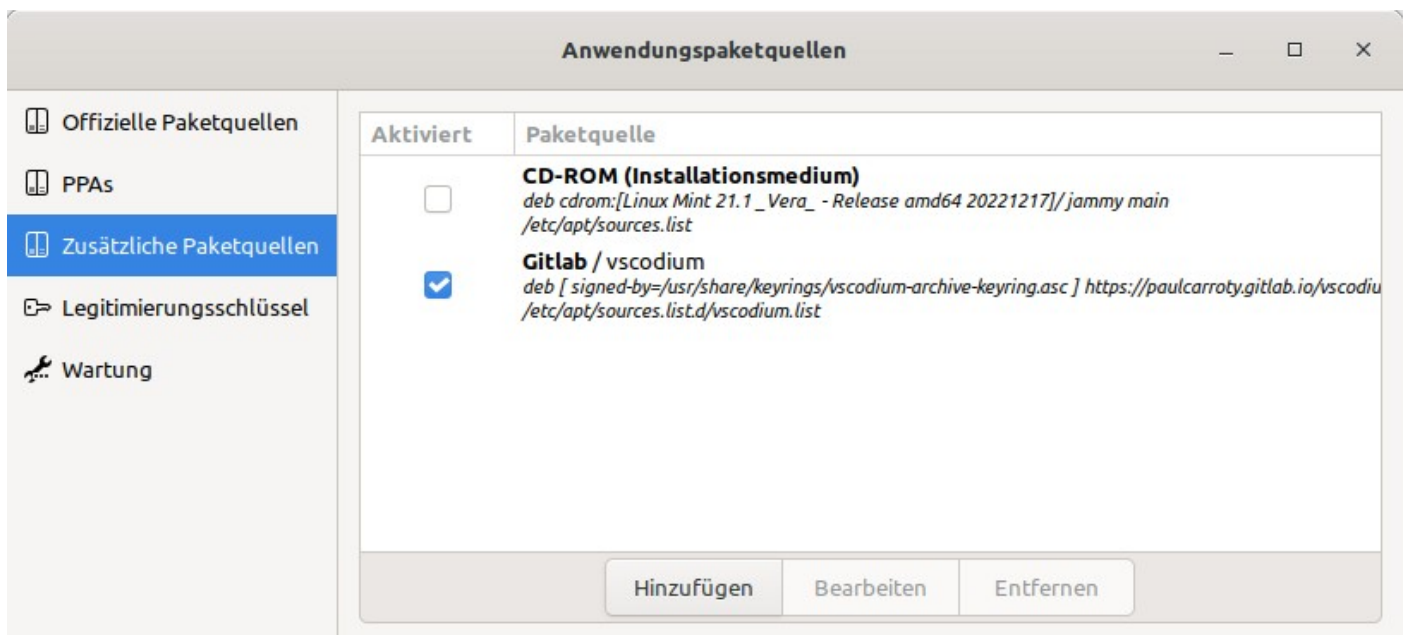
Meldung: „Bitte verwenden Sie apt-cdrom, damit diese CD-ROM von APT erkannt wird“



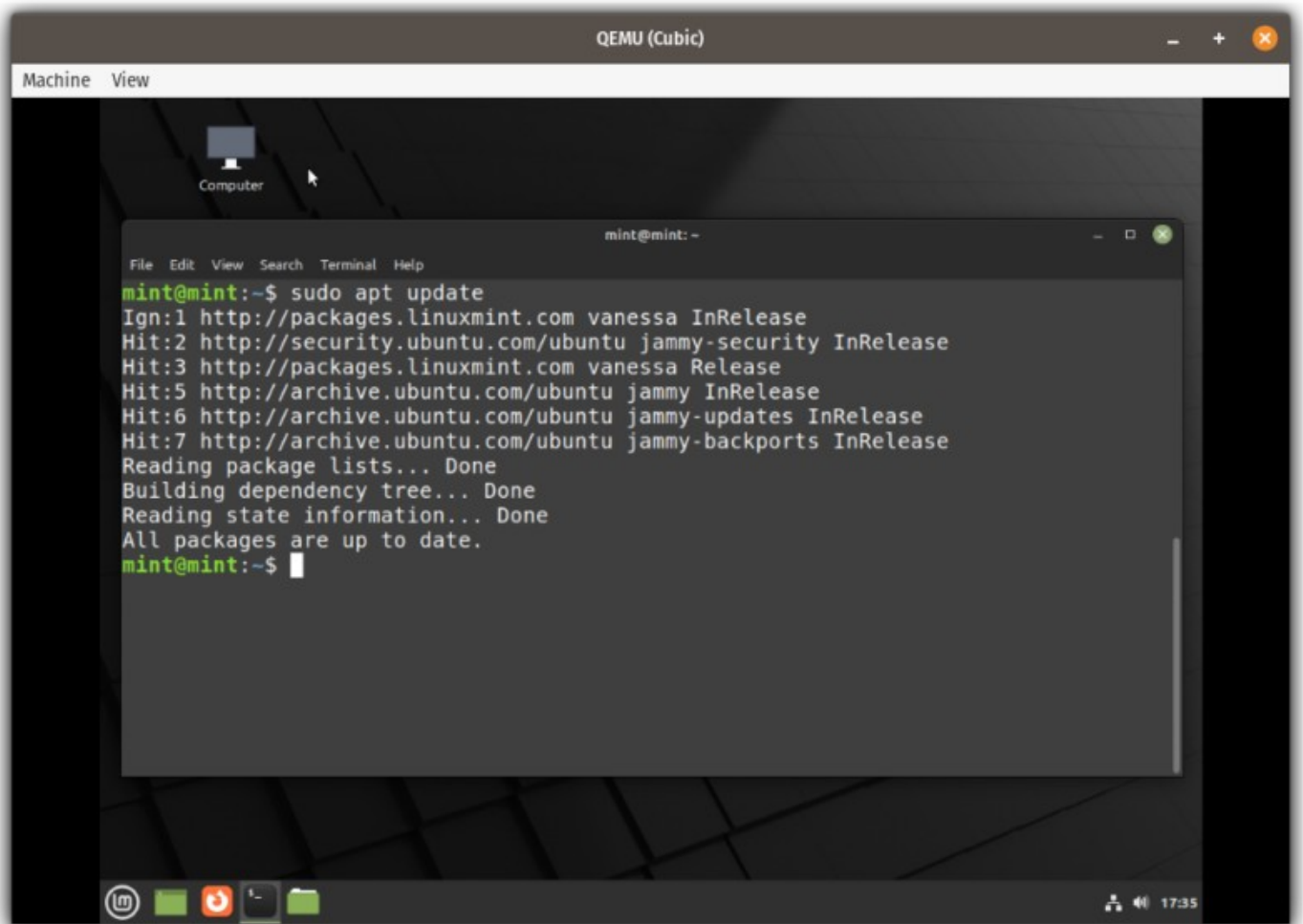
```
mint@mint: ~  
File Edit View Search Terminal Help  
Ign:1 cdrom://Linux Mint 21 Cinnamon 64-bit _Custom Vanessa_ (20221021) jammy InRelease  
Err:2 cdrom://Linux Mint 21 Cinnamon 64-bit _Custom Vanessa_ (20221021) jammy Release  
Please use apt-cdrom to make this CD-ROM recognized by APT. apt-get update cannot be used to add new CD-ROMs  
Hit:3 http://archive.ubuntu.com/ubuntu jammy InRelease  
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Hit:5 http://archive.ubuntu.com/ubuntu jammy-updates InRelease  
Ign:6 http://packages.linuxmint.com vanessa InRelease  
Hit:7 http://archive.ubuntu.com/ubuntu jammy-backports InRelease  
Hit:8 http://packages.linuxmint.com vanessa Release  
Reading package lists... Done  
E: The repository 'cdrom://Linux Mint 21 Cinnamon 64-bit _Custom Vanessa_ (20221021) jammy Release' does not have a Release file.  
N: Updating from such a repository can't be done securely, and is therefore disabled by default.  
N: See apt-secure(8) manpage for repository creation and user configuration details.  
mint@mint:~$
```

Um dieses Problem zu lösen, muss man einfach den Installationsdatenträger (CD-ROM) in den Softwareressourcen deaktivieren.

Dazu öffnet man die Anwendungspaketquellen in Linux Mint (bzw. „Software und Updates“ bei Ubuntu) und deaktiviert die Option „CD-ROM (Installationsmedium)“.



Sobald man die CD-ROM Option deaktiviert hat, sollte es wie jedes andere ISO funktionieren, das man von der offiziellen Seite herunterladen kann.



```
QEMU (Cubic)
Machine View
Computer
mint@mint: ~
File Edit View Search Terminal Help
mint@mint:~$ sudo apt update
Ign:1 http://packages.linuxmint.com vanessa InRelease
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:3 http://packages.linuxmint.com vanessa Release
Hit:5 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:6 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:7 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
mint@mint:~$
```

