My-Stuff

- Arch
- Start into AI
- Linux-Mint
- OhMy zsh
- Debian Upgrade
- Win10
- Remove Win password
- Improving Battery
- Setup - Deja - Dup
- Auto-Update
- ATEM Controller
- ffmpeg
- Backup on LineageOS
- XiaomiToolV2
- Stress-Test
- OBS-remove-Background
- Thin Client to Proxmox
- Kindersicherung
- Print with AB-88H
- yt-dlp
- Add something to BTCmap.org
- Make your own ISO
- Davinci Resolve on Linux MINT
- Nice scripts
    - Converte Markdown File to PDF
    - tmp

- SRT to RTMP
- Create P2P file share
- Install-Standart-TMP
- Edit Metadata
- Davinci Resolve
- Satisfactory Steam server
- Thinkpad Fingerprint on Linux
- Run Steam in gamepadui/big-picture mode at boot
- Importing Open Source Models to Ollama

Arch

Installer:

Install defaults, with KDE plasma.

Check The mirror for multilib

Post install:

Needet:

```
sudo pacman -S git htop vim
```

```
mkdir -p git && cd git
git clone https://aur.archlinux.org/paru
cd paru
makepkg -si
```

```
paru -S brave-browser
```

Configuration in GUI:

Fix scroll direction: Settings -> Mouse & Touchpad -> Touchpad: Invert scroll direction (Natural scrolling)

Set local defaults: Settings -> Region & Language

Fix task switcher: Settings -> Window Management -> Task Switcher: Disable "Show selected window"

Fix Display scale: Settings -> Display & monitor -> Display Configuration

Set Powersettings: Settings -> Power Management: On AC / On Battery / On Low Battery

Configure CLI:

edit `/etc/pacman.conf`

```
Color # uncoment
ParallelDownloads # uncoment
ILoveCandy # add
```

Select faster mirrors:

```
sudo cp /etc/pacman.d/mirrorlist /etc/pacman.d/mirrorlist.bak
paru -S reflector
sudo reflector --verbose --latest 10 --protocol https --sort rate --save
/etc/pacman.d/mirrorlist

sudo pacman -Suy
```

Theming:

in Settings -> Global Theme:

Add Apple Ventura Dark Plasma 6 by Adolfo

Icon theme use: WhiteSur icon theme By vinceliuice

Plus Loginscreen SDDM: Get White Sur SSDM

> You need a restart or at least a relogin

ZSH

Install:

```
paru -S zsh zsh-completions
```

```
sh -c "$(curl -fsSL https://raw.github.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"
```

Set theme:

Clone some plugins and theme:

```
git clone --depth=1 https://github.com/romkatv/powerlevel10k.git ${ZSH_CUSTOM:-$HOME/.oh-my-zsh/custom}/themes/powerlevel10k
```

```
git clone https://github.com/zsh-users/zsh-autosuggestions ${ZSH_CUSTOM:-~/.oh-my-zsh/custom}/plugins/zsh-autosuggestions
```

```
git clone https://github.com/zsh-users/zsh-syntax-highlighting.git ${ZSH_CUSTOM:-~/.oh-my-zsh/custom}/plugins/zsh-syntax-highlighting
```

edit `~/.zshrc`

```
ZSH_THEME="powerlevel10k/powerlevel10k"
plugins=(git vscode zsh-autosuggestions zsh-syntax-highlighting command-not-found)
```

Now Set ZSH as default on terminal:

Terminal Settings -> Profiles: Add New

Install Apps:

For Proton-GE: ProtonUP-QT

Add Bluetooth:

```
paru -S bluez blueman bluez-utils
sudo modprobe btusb
sudo systemctl enable bluetooth
sudo systemctl start bluetooth
```

System Utils:

```
paru -S brave-browser tar rsync neofetch htop vim curl wget power-profiles-daemon linux-headers flatpak kde-applications-meta thunderbird net-tools bind exfatprogs sshfs nfs-utils e2fsprogs exfatprogs fuse-exfat dosfstools ntfs-3g xfsprogs
```

My Apps:

```
paru -S davinci-resolve element-desktop telegram-desktop gimp steam vscodium timeshift timeshift-systemd-timer rustdesk darktable prismlauncher heroic-games-launcher spotify nextcloud-client bitwarden libreoffice-fresh fastboot adb obs-studio v4l2loopback-dkms
```

Davinci:

```
paru -S davinci-resolve

# Or Pro version:
```

```
par -S davinci-resolve-studio
```

Lock-Screen

Set the unlock failures to 10:

edit `/etc/security/faillock.conf`

```
deny = 10
```

Brave

Set defautl search engin:

```
https://search.cs30.de/?q=%s
```

Random MAC:

Damit bei der Verbindung mit einem neuen Netzwerk automatisch eine zufällige MAC-Adresse verwendet wird, kannst du die **NetworkManager**-Konfiguration so anpassen, dass diese Einstellung global für alle neuen Verbindungen gilt.

Vorgehen:

1. **Globale NetworkManager-Konfiguration anpassen**

Öffne die NetworkManager-Konfigurationsdatei:

```
sudo nano /etc/NetworkManager/NetworkManager.conf
```

Füge die folgenden Zeilen hinzu oder bearbeite bestehende Einträge:

```
[device]
wifi.scan-rand-mac-address=yes

[connection]
wifi.cloned-mac-address=random
ethernet.cloned-mac-address=random
```

- **`wifi.scan-rand-mac-address=yes`** sorgt dafür, dass während der Netzwerksuche (Scanning) eine zufällige MAC-Adresse verwendet wird.
- **`wifi.cloned-mac-address=random`** und **`ethernet.cloned-mac-address=random`** stellen sicher, dass beim Verbinden mit einem neuen Netzwerk eine zufällige MAC-Adresse verwendet wird.

Speichere die Datei (`CTRL+O`, `ENTER`, `CTRL+X`).

---

2. **Standardverhalten für neue Verbindungen festlegen**

Bearbeite die Vorlage, die NetworkManager für neue Verbindungen verwendet:

```
sudo nano /etc/NetworkManager/system-connections/default
```

Falls die Datei nicht existiert, erstelle sie mit:

```
sudo touch /etc/NetworkManager/system-connections/default
sudo chmod 600 /etc/NetworkManager/system-connections/default
```

Füge folgende Zeilen hinzu:

```
[connection]
wifi.cloned-mac-address=random
ethernet.cloned-mac-address=random
```

Speichere und schließe die Datei.

---

3. **NetworkManager neu starten**

Damit die Änderungen wirksam werden, starte den NetworkManager-Dienst neu:

```
sudo systemctl restart NetworkManager
```

---

4. **Optionale Überprüfung**

Verwende ein neues Netzwerk, das du noch nicht konfiguriert hast, und prüfe die zufällig verwendete MAC-Adresse:

```
ip link show
```

Die `link/ether`-Zeile zeigt die aktuelle MAC-Adresse.

KDE Plasma GUI (zusätzlich):

In KDE Plasma kannst du diese Einstellungen auch für alle bestehenden Verbindungen anpassen:

1. Gehe zu **Systemeinstellungen > Netzwerk > Verbindungen**.
2. Wähle eine Verbindung aus und klicke auf **Bearbeiten**.
3. Gehe zum Tab **"Allgemein"** oder **"Wi-Fi"**.
4. Setze **"MAC-Adresse"** auf **"Zufällig"**.
5. Klicke auf **Speichern**.

Diese Schritte gelten dann für bestehende Verbindungen, aber die oben beschriebenen Konfigurationsänderungen sorgen dafür, dass neue Netzwerke automatisch mit einer zufälligen MAC-Adresse verbunden werden.

Start into AI

Default for CPU:

Download and install this: https://gpt4all.io/index.html

Default for Nvidia or AMD GPU:

```
git clone https://github.com/oobabooga/text-generation-webui
```

then cd into it and run install:

```
./start_linux.sh
```

Find good LLM or other AI model:

https://huggingface.co/models?pipeline_tag=text-generation&sort=trending

Linux-Mint

Short-Cut

Secureboot:

If no keys, enroll key from live system:

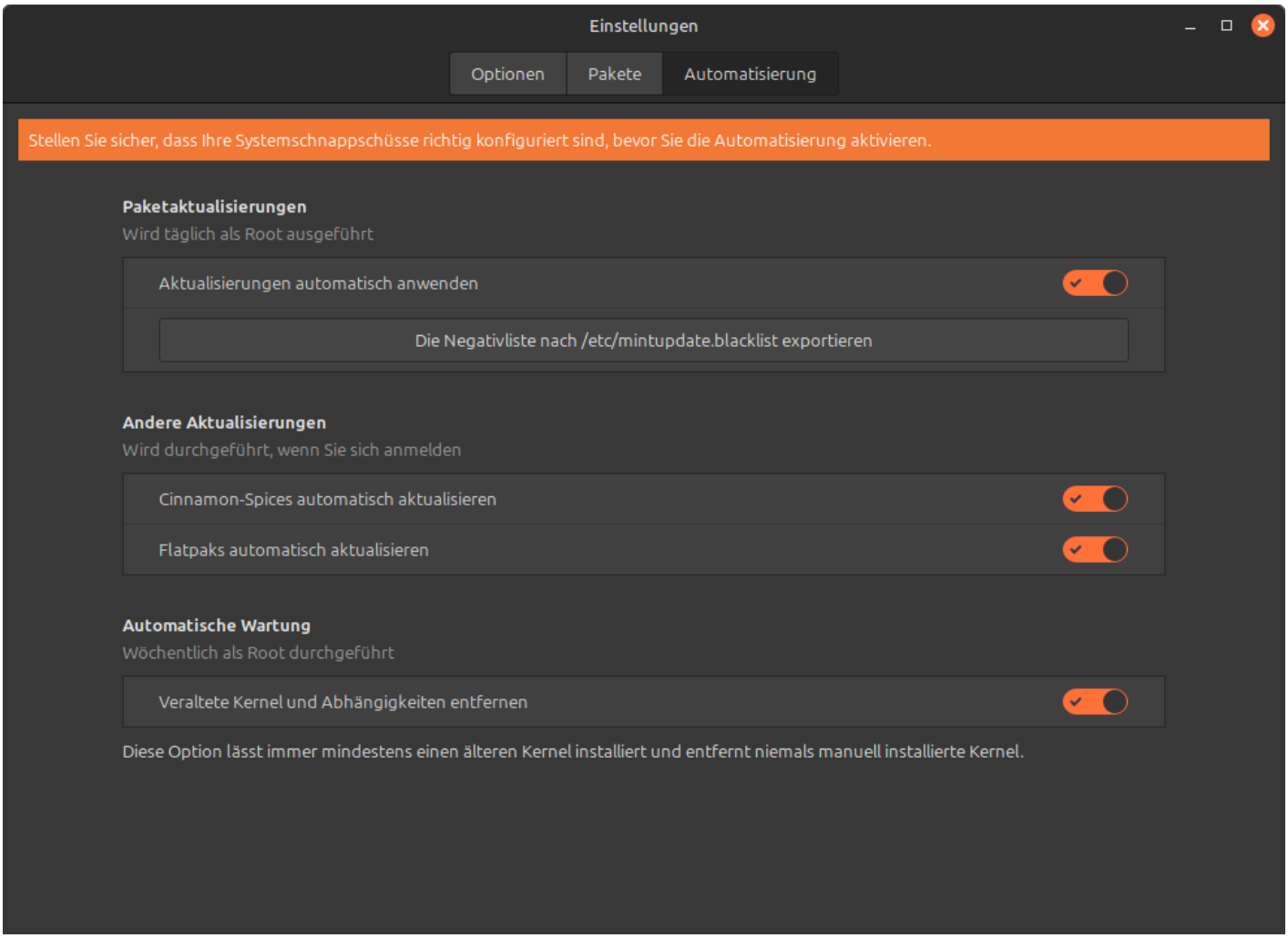
```
sudo mokutil --import VentoyKey.cer
```

Add user enc

```
sudo adduser --encrypt-home username
# For password change after login
sudo chage -d 0 username
```

Auto-Updates

In der MINT update UI als ADMIN folgende hacken setzen:



Bei jedem neuen User ohne sudo rechte, muessen `Cinnamon-Spices` und `Flatpaks` nochmal seperat eingeschaltet werden

Installation user Apps:

System Pakete

Als admin installieren

System Utils

```
sudo apt install -y vim htop wget git apt-transport-https
```

Zerotier

```
curl -s https://install.zerotier.com | sudo bash
```

Brave

```
curl -fsS https://dl.brave.com/install.sh | sh
```

Element

```
sudo apt install -y wget apt-transport-https; \
sudo wget -O /usr/share/keyrings/element-io-archive-keyring.gpg
https://packages.element.io/debian/element-io-archive-keyring.gpg; \
echo "deb [signed-by=/usr/share/keyrings/element-io-archive-keyring.gpg]
https://packages.element.io/debian/ default main" | sudo tee /etc/apt/sources.list.d/element-
io.list; \
sudo apt update; \
sudo apt install -y element-desktop;
```

Nextcloud

```
sudo apt install -y nextcloud-desktop nemo-nextcloud
```

Flatpaks

entweder als admin mit sudo systemweit oder als standart user ohne syste nur fuer den Nutzer

Bitwarden

```
sudo flatpak -y install com.bitwarden.desktop
```

OnlyOffice

```
sudo flatpak -y install org.onlyoffice.desktopeditors
```

Telegram

```
sudo flatpak -y install org.telegram.desktop
```

AnyType

```
sudo flatpak -y install io.anytype.anytype
```

Old Stuff:

First Steps

- Spiegelserver ändern und updates installieren
  - Reboot
- nebenher applets hinzufügen
- transfer .config files from cloud

Packages to install:

tools:

```
sudo apt install -y vim htop wget git apt-transport-https
```

Bitwarden:

```
flatpak -y install com.bitwarden.desktop
```

Element:

```
sudo apt install -y wget apt-transport-https; \
sudo wget -O /usr/share/keyrings/element-io-archive-keyring.gpg
https://packages.element.io/debian/element-io-archive-keyring.gpg; \
echo "deb [signed-by=/usr/share/keyrings/element-io-archive-keyring.gpg]
https://packages.element.io/debian/ default main" | sudo tee /etc/apt/sources.list.d/element-
io.list; \
sudo apt update; \
sudo apt install element-desktop;
```

VSCodium:

```
wget -qO - https://gitlab.com/paulcarroty/vscodium-deb-rpm-repo/raw/master/pub.gpg \
    | gpg --dearmor \
    | sudo dd of=/usr/share/keyrings/vscodium-archive-keyring.gpg
```

```
echo 'deb [arch=amd64 signed-by=/usr/share/keyrings/vscodium-archive-keyring.gpg ]
https://download.vscodium.com/debs vscodium main' \
    | sudo tee /etc/apt/sources.list.d/vscodium.list
```

```
sudo apt update && sudo apt install -y codium
```

DistroBox - With UI:

Github: https://github.com/89luca89/distrobox?tab=readme-ov-file#installation

Ubuntu-repro:

https://launchpad.net/~michel-slm/+archive/ubuntu/distrobox

UI:

```
flatpak install -y io.github.dvlv.boxbuddyrs
```

Also good wine configurator:

```
flatpak install -y com.usebottles.bottles
```

AMD GPU:

- **CoreCtrl** installieren:
- add file `/etc/apt/preferences.d/corectrl`
    - ```
      # Never prefer packages from the ernstp repository
      Package: *
      Pin: release o=LP-PPA-ernstp-mesarc
      Pin-Priority: 1

      # Allow upgrading only corectrl from LP-PPA-ernstp-mesarc
      ```
      ```

```
Package: corectrl
Pin: release o=LP-PPA-ernstp-mesarc
Pin-Priority: 500
```

- ```
  sudo add-apt-repository ppa:ernstp/mesarc
  sudo apt update
  sudo apt install corectrl
  ```

- **Radeon Profile** installieren:

  ```
  sudo add-apt-repository ppa:radeon-profile/stable
  sudo apt update
  sudo apt install radeon-profile
  ```

My used packages:

```
#Brave
sudo apt install curl; \
sudo curl -fsSLo /usr/share/keyrings/brave-browser-archive-keyring.gpg https://brave-browser-apt-release.s3.brave.com/brave-browser-archive-keyring.gpg; \
echo "deb [arch=amd64 signed-by=/usr/share/keyrings/brave-browser-archive-keyring.gpg] https://brave-browser-apt-release.s3.brave.com/ stable main"|sudo tee /etc/apt/sources.list.d/brave-browser-release.list; \
sudo apt update; \
sudo apt install -y brave-browser;
```

```
#Sublime
wget -qO - https://download.sublimetext.com/sublimehq-pub.gpg | gpg --dearmor | sudo tee /etc/apt/trusted.gpg.d/sublimehq-archive.gpg > /dev/null; \
echo "deb https://download.sublimetext.com/ apt/stable/" | sudo tee /etc/apt/sources.list.d/sublime-text.list; \
sudo apt update; \
```
```

```
sudo apt install -y sublime-text;
```

```
#Signal

wget -O- https://updates.signal.org/desktop/apt/keys.asc | gpg --dearmor > signal-desktop-keyring.gpg; \
cat signal-desktop-keyring.gpg | sudo tee /usr/share/keyrings/signal-desktop-keyring.gpg > /dev/null; \

echo 'deb [arch=amd64 signed-by=/usr/share/keyrings/signal-desktop-keyring.gpg] https://updates.signal.org/desktop/apt xenial main' |\
  sudo tee /etc/apt/sources.list.d/signal-xenial.list; \

sudo apt update && sudo apt install -y signal-desktop;
```

QR-Code Generator and Reader:

```
sudo apt install -y qtqr
```

```
#FreeCAD
```

Mac Theme:

install script:

https://git.cs30.de/MK_0x41/install-mac-theme-mint

```
git clone https://git.cs30.de/MK_0x41/install-mac-theme-mint

sudo ./INSTALL.sh
```

Manuell

```
git clone https://git.cs30.de/MK_0x41/WhiteSur-icon-theme
```

```
git clone https://git.cs30.de/MK_0x41/McMojave-cursors
```

```
git clone https://git.cs30.de/MK_0x41/WhiteSur-gtk-theme
```

Go in every folder and run:

```
sudo ./install
```

> For Flatpak support:

In WhiteSur-gtk-theme:

```
sudo cp -r /usr/share/themes/WhiteSur-Dark /home/mk_0x41/.themes/
sudo flatpak override --filesystem=/home/mk_0x41/.themes/WhiteSur-Dark
```

Add the Plank:

```
sudo apt install -y plank synapse
```

> ctrl + right-click to edit

> To autostart add Plank to Startup-Applications

Add enc user:

```
sudo adduser --encrypt-home username
# For password change after login
sudo chage -d 0 username
```

Disable C6 state for AMD Errors:

https://github.com/ZanMax/amd-states-linux

Wireguard VPN

Um sich mit einem WireGuard VPN-Server unter Linux Mint zu verbinden, müssen Sie WireGuard installieren und anschließend die bereitgestellte `.conf`-Datei verwenden. Hier sind die Schritte:

1. **WireGuard installieren**:
   Öffnen Sie ein Terminal und führen Sie die folgenden Befehle aus:

   ```
   sudo apt update; \
   sudo apt install -y wireguard resolvconf
   ```

2. **.conf Datei kopieren**:
   Kopieren Sie die bereitgestellte `.conf`-Datei in das `/etc/wireguard/` Verzeichnis. Angenommen, Ihre Datei heißt `wg0.conf` und befindet sich in Ihrem Home-Verzeichnis:

   ```
   sudo cp ~/<device-name>.conf /etc/wireguard/<con. name>.conf
   ```

   Stellen Sie sicher, dass die Berechtigungen korrekt gesetzt sind:

   ```
   sudo chmod o+r /etc/wireguard/
   ```

3. **Verbindung herstellen**:
   Verwenden Sie den folgenden Befehl, um eine Verbindung zum VPN-Server herzustellen:

   ```
   sudo wg-quick up wg0
   ```

   (wobei `wg0` der Name Ihrer Konfigurationsdatei ohne die `.conf`-Erweiterung ist)

4. **Verbindung trennen**:
   Wenn Sie die Verbindung trennen möchten, verwenden Sie:

   ```
   sudo wg-quick down wg0
   ```

5. **Optional (Automatischer Start)**:
   Wenn Sie möchten, dass WireGuard automatisch beim Booten startet, können Sie es mit `systemd` aktivieren:

   ```
   sudo systemctl enable wg-quick@wg0
   ```

Denken Sie daran, dass Sie die entsprechenden Firewall-Regeln und Routing-Einstellungen in Ihrer `.conf`-Datei haben müssen, damit alles ordnungsgemäß funktioniert.

Wiregurad-GUI

> install Wireguard aplett on Linux MINT

1. **PolicyKit-Regel erstellen**:
   Erstellen Sie eine neue PolicyKit-Regel. Sie können einen Editor Ihrer Wahl verwenden, hier verwenden wir `nano`:

   ```
   sudo vim /etc/polkit-1/localauthority/50-local.d/wg-quick.pkla
   ```

2. **Inhalt der Regel hinzufügen**:
   Fügen Sie den folgenden Inhalt in die Datei ein:

   ```
   [Allow wg-quick for yourusername]
   Identity=unix-user:yourusername
   Action=org.freedesktop.policykit.exec
   ResultAny=yes
   ResultInactive=yes
   ResultActive=yes
   ```

   Ersetzen Sie `yourusername` durch Ihren tatsächlichen Benutzernamen.
3. **Änderungen speichern und beenden**:

Speichern Sie die Datei und beenden Sie den Editor.

OhMy zsh

Install:

```
sudo apt install zsh fonts-hack-ttf fonts-powerline
```

```
sh -c "$(curl -fsSL https://raw.github.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"
```

Config:

edit `~/.zshrc`

set theme:

```
git clone --depth=1 https://github.com/romkatv/powerlevel10k.git ${ZSH_CUSTOM:-$HOME/.oh-my-zsh/custom}/themes/powerlevel10k
```

```
ZSH_THEME="powerlevel10k/powerlevel10k"
```

To Restart the installation:

```
p10k configure
```

set Plugins:

Install

```
git clone https://github.com/zsh-users/zsh-autosuggestions ${ZSH_CUSTOM:-~/.oh-my-zsh/custom}/plugins/zsh-autosuggestions
```

```
git clone https://github.com/zsh-users/zsh-syntax-highlighting.git ${ZSH_CUSTOM:-~/.oh-my-zsh/custom}/plugins/zsh-syntax-highlighting
```

Set in Config:

```
plugins=(git vscode zsh-autosuggestions zsh-syntax-highlighting command-not-found)
```

Debian Upgrade

Upgrade debain11 to debain12:

1. Edit the file `/etc/apt/sources.list` using a text editor and replace each instance of `bullseye` with `bookworm`. Next find the update line, replace keyword **`bullseye-updates`** with `bookworm-updates`. Finally, search the security line, replace keyword `bullseye-security` with `bookworm-security`
2. Update the packages index on Debian Linux, run: `sudo apt update`
3. Prepare for the operating system minimal system upgrade, run: `sudo apt upgrade --without-new-pkgs`
4. Finally, update Debian 11 to Debian 12 Bookworm by running: `sudo apt full-upgrade`
5. Reboot the Linux system so that you can boot into Debian 12 Bookworm

Win10

Activation script:

```
@echo off
title Activate Windows 10 ALL versions for FREE!&cls&echo
=====================================================================================&echo #Project:
Activating Microsoft software products for FREE without software&echo
=====================================================================================&echo.&echo
#Supported products:&echo - Windows 10 Home&echo - Windows 10 Home N&echo - Windows 10 Home
Single Language&echo - Windows 10 Home Country Specific&echo - Windows 10 Professional&echo -
Windows 10 Professional N&echo - Windows 10 Education&echo - Windows 10 Education N&echo -
Windows 10 Enterprise&echo - Windows 10 Enterprise N&echo - Windows 10 Enterprise LTSB&echo -
Windows 10 Enterprise LTSB N&echo.&echo.&echo
=====================================================================================&echo Activating
your Windows...&cscript //nologo slmgr.vbs /ckms >nul&cscript //nologo slmgr.vbs /upk
>nul&cscript //nologo slmgr.vbs /cpky >nul&set i=1&wmic os | findstr /I "enterprise" >nul
if %errorlevel% EQU 0 (cscript //nologo slmgr.vbs /ipk NPPR9-FWDCX-D2C8J-H872K-2YT43
>nul&cscript //nologo slmgr.vbs /ipk DPH2V-TTNVB-4X9Q3-TJR4H-KHJW4 >nul&cscript //nologo
slmgr.vbs /ipk WNMTR-4C88C-JK8YV-HQ7T2-76DF9 >nul&cscript //nologo slmgr.vbs /ipk 2F77B-TNFGY-
69QQF-B8YKP-D69TJ >nul&cscript //nologo slmgr.vbs /ipk DCPHK-NFMTC-H88MJ-PFHPY-QJ4BJ
>nul&cscript //nologo slmgr.vbs /ipk QFFDN-GRT3P-VKWWX-X7T3R-8B639 >nul&goto server) else wmic
os | findstr /I "home" >nul
if %errorlevel% EQU 0 (cscript //nologo slmgr.vbs /ipk TX9XD-98N7V-6WMQ6-BX7FG-H8Q99
>nul&cscript //nologo slmgr.vbs /ipk 3KHY7-WNT83-DGQKR-F7HPR-844BM >nul&cscript //nologo
slmgr.vbs /ipk 7HNRX-D7KGG-3K4RQ-4WPJ4-YTDFH >nul&cscript //nologo slmgr.vbs /ipk PVMJN-6DFY6-
9CCP6-7BKTT-D3WVR >nul&goto server) else wmic os | findstr /I "education" >nul
if %errorlevel% EQU 0 (cscript //nologo slmgr.vbs /ipk NW6C2-QMPVW-D7KKK-3GKT6-VCFB2
>nul&cscript //nologo slmgr.vbs /ipk 2WH4N-8QGBV-H22JP-CT43Q-MDWWJ >nul&goto server) else wmic
os | findstr /I "10 pro" >nul
if %errorlevel% EQU 0 (cscript //nologo slmgr.vbs /ipk W269N-WFGWX-YVC9B-4J6C9-T83GX
>nul&cscript //nologo slmgr.vbs /ipk MH37W-N47XK-V7XM9-C7227-GCQG9 >nul&goto server) else
(goto notsupported)
:server
if %i%==1 set KMS=kms7.MSGuides.com
```

```
if %i%==2 set KMS=kms8.MSGuides.com
if %i%==3 set KMS=kms9.MSGuides.com
if %i%==4 goto notsupported
cscript //nologo slmgr.vbs /skms %KMS%:1688 >nul&echo
========================================================================&echo.&echo.
cscript //nologo slmgr.vbs /ato | find /i "successfully" && (echo.&echo
========================================================================&echo.&echo #My
official blog: MSGuides.com&echo.&echo #How it works: bit.ly/kms-server&echo.&echo #Please
feel free to contact me at msguides.com@gmail.com if you have any questions or
concerns.&echo.&echo #Please consider supporting this project: donate.msguides.com&echo #Your
support is helping me keep my servers running everyday!&echo.&echo
========================================================================&choice /n /c YN
/m "Would you like to visit my blog [Y,N]?" & if errorlevel 2 exit) || (echo The connection to
my KMS server failed! Trying to connect to another one... & echo Please wait... & echo. &
echo. & set /a i+=1 & goto server)
explorer "http://MSGuides.com"&goto halt
:notsupported
echo ========================================================================&echo.&echo
Sorry! Your version is not supported.&echo.
:halt
pause >nul
```

Remove Win password

Setup:

1. Boot Linux
2. Install chntpw
    1. ```
       sudo apt install -y chntpw
       ```
3. Mount win disk
4. cd into `<win-disk>/Windows/System32/config`
5. run

   ```
   sudo chntpw -l SAM
   ```

6. use

   ```
   sudo chntpw -u "Username" SAM
   ```

7. Reboot to Win and have fun xD
   ```

Improving Battery

Guide to Improving Battery Life on Linux

Just use this:

https://github.com/AdnanHodzic/auto-cpufreq

Table of Contents

1. Introduction
2. Installing TLP
3. Starting and Enabling TLP
4. Understanding TLP Modes
5. Checking TLP Status
6. Conclusion

1. Introduction

This guide will walk you through the installation and usage of TLP, a tool that can help improve the battery life of your Linux Mint laptop. TLP optimizes power usage in the background, automatically adjusting settings based on power source.

2. Installing TLP

TLP is an advanced power management tool for Linux that runs in the background and automatically optimizes power settings.

To install TLP, open a terminal and run the following commands:

```
sudo add-apt-repository ppa:linrunner/tlp
sudo apt-get update
sudo apt-get install tlp tlp-rdw
```

3. Starting and Enabling TLP

This command will start TLP immediately. However, to ensure TLP starts automatically at boot, you need to enable it:

```
sudo systemctl enable tlp.service
```

After installing TLP, you can start it with the following command:

```
sudo tlp start
```

4. Understanding TLP Modes

TLP operates in two main modes: AC and Battery. When your laptop is plugged into a power source, TLP operates in AC mode, allowing for higher performance. When running on battery, TLP switches to Battery mode, aiming to minimize power consumption and extend battery life. This switch happens automatically.

5. Checking TLP Status

You can check the status of the TLP service at any time with the following command:

```
sudo systemctl status tlp.service
```

This will show whether the TLP service is active and running. For more detailed information about your system's power settings and status, you can use the `tlp-stat` command. This command outputs a variety of information, including current CPU settings, power settings for various devices, and more.

6. Conclusion

By using TLP, you can gain more control over your system's power usage and potentially extend your laptop's battery life. Remember to monitor your system's performance and stability as you make changes, and always back up your data before making significant system changes.

Setup - Deja - Dup

Deja Dup is a simple backup tool for Linux, seamlessly integrated into the GNOME desktop. It utilizes the `duplicity` backend to produce encrypted, incremental, compressed backups. Here's a guide on how to set up Deja Dup on Linux Mint and store backups on your WebDAV server:

1. Installation of Deja Dup:

Open a terminal and execute the following commands:

```
sudo apt update
sudo apt install deja-dup duplicity
```

2. Launching Deja Dup:

Search for "Deja Dup" in the Linux Mint application menu and launch the application.

3. Configuring Settings:

1. Click on "Preferences" or "Settings".
2. Under "Storage Location", select "WebDAV" from the dropdown menu.
3. In the "Folder" field, input your URL, replacing `joscha.mijailovic` with `<username>`. It should look like this: `davs://one-cloud.cs30.de/remote.php/dav/files/<username>/?dir=/Backup`. Note that the exact path will depend on your WebDAV server setup.
4. Under the "Login Credentials" section, input your username and password.
5. Determine which folders you wish to backup and which ones you wish to exclude. By default, your Home directory will be backed up.
6. Choose how often backups should occur and for how long they should be retained.

Exclude and some settings:

```
dconf write /org/gnome/deja-dup/exclude-list "['/home/$USER/VirtualBox VMs',
'/home/$USER/Downloads', '/home/$USER/gpt4all', '/home/$USER/Nextcloud',
'/home/$USER/Pictures', '/home/$USER/Videos', '/home/$USER/tmp/VM',
'/home/$USER/.local/share/nomic.ai', '/home/$USER/.steam/debian-installation/steamapps']"
```

4. Starting Backup:

Click "Backup Now" to manually initiate the backup process. Remember, the first backup might take a while depending on the amount of data. Subsequent backups are typically faster since they only back up changed files (incremental backups).

5. Restoring Files:

If you ever need to restore files, you can open Deja Dup, choose "Restore", and follow the on-screen instructions.

> Note: As with all backup solutions, ensure that your backups are being made regularly and correctly, and that you know how to restore data when needed. It's advisable to occasionally test the restoration process to make sure everything works as expected.

Good luck with your data backups using Deja Dup!

Auto-Update

Aptoide:

update-script:

`/usr/local/bin/update_packages.sh`

```
sudo bash -c 'echo -e "#!/bin/bash\n\napt update\napt upgrade -y\napt autoremove -y\napt autoclean -y" > /usr/local/bin/update_packages.sh' && sudo chmod +x /usr/local/bin/update_packages.sh
```

```
#!/bin/bash

apt update
apt upgrade -y
apt autoremove -y
apt autoclean -y
```

crateapt-updater.service

```
sudo bash -c 'echo -e "[Unit]\nDescription=Update all apt packages\n\n[Service]\nType=oneshot\nExecStart=/usr/local/bin/update_packages.sh\n\n[Install]\nWantedBy=multi-user.target" > /etc/systemd/system/update_packages.service'
```

```
[Unit]
Description=Update all apt packages

[Service]
Type=oneshot
ExecStart=/usr/local/bin/update_packages.sh

[Install]
WantedBy=multi-user.target
```

Flatapak:

create flatpak-updater.service

```
sudo bash -c 'echo -e "[Unit]\nDescription=Update all flatpak apps and runtimes\n\n[Service]\nType=oneshot\nExecStart=/usr/bin/flatpak update -y\n\n[Install]\nWantedBy=multi-user.target" > /etc/systemd/system/flatpak_updater.service'
```

```
[Unit]
Description=Update all flatpak apps and runtimes

[Service]
Type=oneshot
ExecStart=/usr/bin/flatpak update -y

[Install]
WantedBy=multi-user.target
```

Timer:

create apt-updater.timer

```
sudo bash -c 'echo -e "[Unit]\nDescription=Run update_packages daily\n\n[Timer]\nOnBootSec=5min\nOnUnitActiveSec=24h\nPersistent=true\n\n[Install]\nWantedBy=timers.target" > /etc/systemd/system/update_packages.timer'
```

```
[Unit]
Description=Run update_packages daily

[Timer]
OnBootSec=5min
OnUnitActiveSec=24h
Persistent=true

[Install]
WantedBy=timers.target
```

Timer - Server:

```
sudo bash -c 'echo -e "[Unit]\nDescription=Run update_packages daily\n\n[Timer]\nOnBootSec=5min\nOnCalendar=*-*-* 02:00:00\nPersistent=true\n\n[Install]\nWantedBy=timers.target" > /etc/systemd/system/update_packages.timer'
```

```
[Unit]
Description=Run update_packages daily

[Timer]
OnBootSec=5min
OnCalendar=*-*-* 02:00:00
Persistent=true

[Install]
WantedBy=timers.target
```

create flatpak-updater.timer

```
sudo bash -c 'echo -e "[Unit]\nDescription=Run flatpak_updater
daily\n\n[Timer]\nOnBootSec=5min\nOnUnitActiveSec=24h\nPersistent=true\n\n[Install]\nWantedBy=
timers.target" > /etc/systemd/system/flatpak_updater.timer'
```

```
[Unit]
Description=Run flatpak-updater daily

[Timer]
OnBootSec=5min
OnUnitActiveSec=24h
Persistent=true

[Install]
WantedBy=timers.target
```

Aktivierung:

```
sudo systemctl daemon-reload
sudo systemctl enable --now flatpak_updater.timer
sudo systemctl enable --now update_packages.timer
```

check:

```
sudo systemctl list-timers --all
```

ATEM Controller

Install OpenSwitcher:

```
flatpak remote-add openswitcher https://flatpak.brixit.nl/brixit.flatpakrepo
```

```
flatpak install https://flatpak.brixit.nl/switcher-stable.flatpakref
```

then run the app, if you have Errors run in Terminal:

```
flatpak run nl.brixit.Switcher
```

If you get some Errors like this:

```
error: runtime/org.gnome.Platform/x86_64/42 not installed
```

Install:

```
flatpak install flathub org.gnome.Platform//42
```

Guide: Running ATEM Control on Linux via Wine

and Receiving RTMP Streams with nginx

Inhaltsverzeichnis

- Install ATEM Control using Wine
- Create a Desktop Shortcut for ATEM Control
- Receive RTMP Streams using nginx
- Configure RTMP Stream in ATEM
- Process the RTMP Stream in OBS

Install ATEM Control using Wine

1. Installation von Wine:

a. Aktualisieren Sie zunächst Ihre Paketliste:

```
sudo apt update
```

b. Installieren Sie anschließend die 64-Bit- und 32-Bit-Versionen von Wine:

```
sudo apt install wine64 wine32
```

c. Erstelle neues wine dir:

```
WINEPREFIX=$HOME/.wine-atem wineboot
```

Um sicherzustellen, dass Wine korrekt installiert ist, führen Sie:

```
WINEPREFIX=$HOME/.wine-atem winecfg
```

Dadurch wird das Wine-Konfigurationsfenster geöffnet und falls erforderlich, wird eine neue Wine-Installation in Ihrem Home-Verzeichnis eingerichtet.

2. Download des erforderlichen VC++ Redistributable:

Laden Sie die VC++ Redistributable Datei herunter, die einige der von ATEM Control benötigten Bibliotheken enthält:

```
wget https://download.microsoft.com/download/9/3/F/93FCF1E7-E6A4-478B-96E7-D4B285925B00/vc_redist.x64.exe
```

3. Installation der VC++ Redistributable mit Wine:

Installieren Sie die heruntergeladene Datei mit Wine:

```
WINEPREFIX=$HOME/.wine-atem WINEARCH=win64 wine64 vc_redist.x64.exe
```

4. Download und Installation von ATEM Control:

a. Besuchen Sie die offizielle Blackmagic Design-Website und laden Sie die ATEM Control-Software herunter. (Hier kann ich keinen direkten Link bereitstellen, da die URL von der Version und anderen Faktoren abhängen kann.)

b. Nachdem Sie die ATEM Control-Installationsdatei heruntergeladen haben (angenommen, es handelt sich um eine `.exe`-Datei), navigieren Sie zu dem Verzeichnis, in dem die Datei gespeichert ist.

c. Führen Sie die Installation von ATEM Control mit Wine aus:

```
WINEPREFIX=$HOME/.wine-atem WINEARCH=win64 wine64 YOUR_DOWNLOADED_FILE.exe
```

Ersetzen Sie `YOUR_DOWNLOADED_FILE.exe` durch den genauen Namen der heruntergeladenen ATEM Control-Installationsdatei.

d. Befolgen Sie die Installationsanweisungen im Setup-Assistenten, um ATEM Control zu installieren.

Nach Abschluss dieser Schritte sollte ATEM Control unter Wine auf Ihrem Linux-System installiert sein und einsatzbereit sein.

Create a Desktop Shortcut for ATEM Control

1. Create a new file named `BMD-atem-control.desktop`:

   ```
   vim BMD-atem-control.desktop
   ```

2. Insert the following content into the file:

   ```
   #!/usr/bin/env xdg-open
   [Desktop Entry]
   Version=1.0
   Type=Application
   Terminal=true
   Icon=mate-panel-launcher
   Icon[en_US]=mate-panel-launcher
   Name[en_US]=BMD-atem-control
   Exec=env WINEPREFIX=$HOME/.wine-atem WINEARCH=win64 taskset -c 0 wine64
   /home/joscha_mijailovic/.wine64/drive_c/Program\ Files\ \(x86\)/Blackmagic\
   Design/Blackmagic\ ATEM\ Switchers/ATEM\ Software\ Control/ATEM\ Software\
   Control.exe
   Name=BMD-atem-control
   ```

3. Save and exit.
4. Make the `.desktop` file executable:

   ```
   chmod +x BMD-atem-control.desktop
   ```

Receive RTMP Streams using nginx

1. Install nginx with the RTMP module:

```
sudo apt install libnginx-mod-rtmp nginx
```

2. Edit the nginx configuration file:

```
sudo vim /etc/nginx/nginx.conf
```

3. Add the following RTMP configuration to the end of the file:

```
rtmp {
    server {
        listen 1935;
        chunk_size 4096;

        application live {
            live on;
            record off;
        }
    }
}
```

4. Save and exit.
5. Restart nginx:

```
sudo systemctl restart nginx
```

Configure RTMP Stream in ATEM

1. Start the ATEM Software Control.
2. Navigate to `Settings > Streaming`.
3. Choose `RTMP` as the protocol.
4. Enter the RTMP address of the nginx server. It should look like: `rtmp://YOUR_SERVER_IP/live`.
    1.
```
<?xml version="1.0" encoding="UTF-8" ?>
<streaming>
    <service>
□□<name>OBS</name>
□□<servers>
□□□<server>
```

<name>10.202.0.2</name>
<url>rtmp://10.202.0.2:1935/live</url>
</server>
</servers>
<profiles>
<profile>
<name>Streaming High</name>
<config resolution="1080p" fps="60">
<bitrate>6000000</bitrate>
<audio-bitrate>128000</audio-bitrate>
</config>
</profile>
<profile>
<name>Streaming Medium</name>
<config resolution="720p" fps="60">
<bitrate>4500000</bitrate>
<audio-bitrate>128000</audio-bitrate>
</config>
<config resolution="720p" fps="30">
<bitrate>3000000</bitrate>
<audio-bitrate>128000</audio-bitrate>
</config>
<config resolution="1080p" fps="60">
<bitrate>7000000</bitrate>
<audio-bitrate>128000</audio-bitrate>
</config>
<config resolution="1080p" fps="30">
<bitrate>4500000</bitrate>
<audio-bitrate>128000</audio-bitrate>
</config>
</profile>
<profile>
<name>Streaming Low</name>
<config resolution="720p" fps="60">
<bitrate>2250000</bitrate>
<audio-bitrate>128000</audio-bitrate>
</config>
<config resolution="720p" fps="30">

```
□□□□□<bitrate>1500000</bitrate>
□□□□□<audio-bitrate>128000</audio-bitrate>
□□□□</config>
□□□□<config resolution="1080p" fps="60">
□□□□□<bitrate>4000000</bitrate>
□□□□□<audio-bitrate>128000</audio-bitrate>
□□□□</config>
□□□□<config resolution="1080p" fps="30">
□□□□□<bitrate>3000000</bitrate>
□□□□□<audio-bitrate>128000</audio-bitrate>
□□□□</config>
□□□</profile>
□□</profiles>
<!-- RTMP session username and password, if required by service
□□<credentials>
□□□<username>test</username>
□□□<password>abc123xyz</password>
□□</credentials>
-->
□</service>
□


</streaming>
```

5. Apply the settings.

Process the RTMP Stream in OBS

1. Open OBS.
2. Navigate to "Sources" and click on the "+" symbol to add a new source.
3. Choose "Media Source" and give it an appropriate name.
4. Uncheck "Local File" and enter your RTMP stream link in the "Input" box. It should look something like this: `rtmp://localhost/live`.
5. Confirm the settings, and the stream should now appear in OBS.

ffmpeg

Merge to files:

```
ffmpeg -i input_video.mp4 -i input_audio.m4a -c copy output_file.mp4
```

Convert Videos to smaller files:

```
#!/bin/bash

# Pfad zum Verzeichnis mit den Originalvideos
VIDEO_DIR="./"
# Zielverzeichnis für die komprimierten Videos
OUTPUT_DIR="./converted"

# Erstelle das Zielverzeichnis, falls es nicht existiert
mkdir -p "$OUTPUT_DIR"

# Durchlaufe alle .mp4 Dateien im Verzeichnis
for original in "$VIDEO_DIR"/*.MP4; do
    # Dateiname ohne Pfad
```

```
    filename=$(basename -- "$original")
    # Vollständiger Pfad zur Ausgabedatei
    output="$OUTPUT_DIR/$filename"

    echo "Komprimiere: $original -> $output"

    # Führe ffmpeg aus, um das Video zu komprimieren
    ffmpeg -i "$original" -c:v libx265 -crf 23 -preset medium -c:a copy "$output"
done

echo "Kompression aller Videos abgeschlossen."
```

Anleitung: Videos in Einzelbilder aufteilen mit `ffmpeg`

Ziel

Mit dieser Anleitung können Sie Videos in Einzelbilder aufteilen. Dies kann nützlich sein, wenn Sie beispielsweise eine Struktur-aus-Bewegung-Rekonstruktion mit Tools wie COLMAP durchführen möchten.

Voraussetzungen

- `ffmpeg` muss auf Ihrem System installiert sein. Wenn es noch nicht installiert ist, können Sie es mit dem folgenden Befehl installieren:

  ```
  sudo apt install ffmpeg
  ```

Schritte

1. **Bash-Skript erstellen und an die richtige Stelle verschieben**
   Führen Sie den folgenden Befehl aus, um das Skript zu erstellen und es direkt nach `/usr/local/bin/split-videos` zu schreiben:

```
echo '#!/bin/bash

# Überprüfen Sie, ob ffmpeg installiert ist
if ! command -v ffmpeg &> /dev/null; then
    echo "ffmpeg ist nicht installiert. Bitte installieren Sie es zuerst."
    exit 1
fi

# Standardwerte für Eingabe- und Ausgabepfade
INPUT_PATH="."
OUTPUT_PATH="."

# Optionen mit getopts verarbeiten
while getopts "i:o:" opt; do
    case "$opt" in
    i) INPUT_PATH="$OPTARG" ;;
    o) OUTPUT_PATH="$OPTARG" ;;
    *) echo "Ungültige Option: -$OPTARG" >&2; exit 1 ;;
    esac
done

# Gängige Videoformate
VIDEO_FORMATS=("*.mp4" "*.avi" "*.mkv" "*.mov" "*.flv" "*.wmv")

# Durchlaufen Sie alle gängigen Videoformate im angegebenen Eingabeverzeichnis
shopt -s nullglob
for format in "${VIDEO_FORMATS[@]}"; do
    videos=($INPUT_PATH/$format)
    for video in "${videos[@]}"; do
        # Entfernen Sie die Dateierweiterung, um den Ordnernamen zu erhalten
        dir_name="${video##*/}"
        dir_name="${dir_name%.*}"

        echo "Verarbeite Video: $video"

        # Erstellen Sie ein Verzeichnis im angegebenen Ausgabepfad mit dem Namen des
```

```
Videos
        mkdir -p "$OUTPUT_PATH/$dir_name"

        # Teilen Sie das Video in Einzelbilder auf und speichern Sie sie im
entsprechenden Verzeichnis
        ffmpeg -i "$video" -q:v 2 "$OUTPUT_PATH/$dir_name/frame_%04d.jpg"
        echo "Bilder gespeichert in: $OUTPUT_PATH/$dir_name/"
    done
done



echo "Alle Videos wurden erfolgreich in Einzelbilder aufgeteilt!"' | sudo tee
/usr/local/bin/split-videos > /dev/null
```

2. **Skript ausführbar machen**
   Machen Sie das Skript mit dem folgenden Befehl ausführbar:

   ```
   sudo chmod +x /usr/local/bin/split-videos
   ```

3. **Skript ausführen**
   Sie können das Skript jetzt von überall auf Ihrem System ausführen:

   ```
   split-videos
   ```

   Oder mit spezifischen Eingabe- und Ausgabepfaden:

   ```
   split-videos -i /path/to/videos -o /path/to/output
   ```

Hinweis

- Das Skript sucht automatisch nach den gängigsten Videoformaten, sodass Sie nichts anpassen müssen.

Normalize-Audio:

Just normalize

create file `normalize_wav.sh`

```
#!/bin/bash

# Verzeichnis mit den WAV-Dateien
input_dir="$1"

# Zielverzeichnis für die normalisierten Dateien
output_dir="${input_dir}/loudness-normalized"

# Erstelle das Zielverzeichnis, falls es nicht existiert
mkdir -p "$output_dir"

# Durchlaufe alle WAV-Dateien im Eingabeverzeichnis
for wav_file in "$input_dir"/*.WAV; do
    # Dateiname ohne Verzeichnis
    filename=$(basename "$wav_file")

    # Normalisierte Datei im Zielverzeichnis
    normalized_file="${output_dir}/${filename}"

    # Normalisieren der Lautstärke
    ffmpeg -i "$wav_file" -af "loudnorm=I=-23:LRA=7:TP=-2" "$normalized_file"

    echo "Normalisiert: $filename -> $normalized_file"
done

echo "Alle Dateien wurden normalisiert und nach $output_dir verschoben."
```

run `zsh normalize_wav.sh ./`

or

`bash normalize_wav.sh ./`

with to pass noize remover:

```
#!/bin/bash

# Verzeichnis mit den WAV-Dateien
input_dir="$1"

# Zielverzeichnis für die normalisierten Dateien
output_dir="${input_dir}/loudness-normalized"

# Temporäres Verzeichnis für die rauschgefilterten Dateien
temp_dir="${input_dir}/temp-noise-removed"

# Erstelle die Verzeichnisse, falls sie nicht existieren
mkdir -p "$output_dir"
mkdir -p "$temp_dir"

# Durchlaufe alle WAV-Dateien im Eingabeverzeichnis
for wav_file in "$input_dir"/*.WAV; do
    # Dateiname ohne Verzeichnis
    filename=$(basename "$wav_file")

    # Temporäre Datei für rauschgefilterte Datei
    temp_file="${temp_dir}/${filename}"

    # Normalisierte Datei im Zielverzeichnis
    normalized_file="${output_dir}/${filename}"

    # Erstelle eine temporäre Rauschprofildatei
    noise_profile="${temp_dir}/${filename}.prof"

    # Extrahiere eine kurze Rauschprobe und erstelle das Rauschprofil
    sox "$wav_file" -n trim 0 1 noiseprof "$noise_profile"

    # Rauschentfernung mit sox
    sox "$wav_file" "$temp_file" noisered "$noise_profile" 0.21

    # Überprüfen, ob die Datei erfolgreich erstellt wurde
    if [ -s "$temp_file" ]; then
        # Lautstärke normalisieren mit ffmpeg
        ffmpeg -i "$temp_file" -af "loudnorm=I=-23:LRA=7:TP=-2" "$normalized_file"
        echo "Rauschentfernung und Normalisierung: $filename -> $normalized_file"
```

```
    else
        echo "Fehler bei der Rauschentfernung: $filename"
    fi

    # Entferne die temporäre Rauschprofildatei
    rm "$noise_profile"
done

# Entferne das temporäre Verzeichnis und die darin enthaltenen Dateien
rm -r "$temp_dir"

echo "Alle Dateien wurden rauschgefiltert, normalisiert und nach $output_dir verschoben."
```

you need sox:

```
apt install -y sox
```

Backup on LineageOS

Anleitung: Backup mit Seedvault auf DAVx5 in LineageOS

Voraussetzungen:

- Ein Gerät mit installiertem LineageOS.
- Seedvault ist auf Ihrem LineageOS-Gerät vorinstalliert.
- DAVx5 ist auf Ihrem Gerät installiert und eingerichtet.

Schritte:

1. **DAVx5 einrichten**:
   - Öffnen Sie die DAVx5-App auf Ihrem Gerät.
   - Fügen Sie Ihr DAV-Serverkonto hinzu und stellen Sie sicher, dass es korrekt synchronisiert wird.
2. **Seedvault Backup aktivieren**:
   - Gehen Sie zu "Einstellungen" > "System" > "Backup".
   - Wählen Sie "Seedvault" als Ihren Backup-Dienst aus.
3. **Backup-Speicherort auswählen**:
   - In den Seedvault-Einstellungen tippen Sie auf "Backup-Speicherort ändern".
   - Wählen Sie "DAVx5" aus der Liste der verfügbaren Speicherorte.
4. **Backup starten**:
   - Tippen Sie auf "Jetzt sichern", um den Backup-Prozess zu starten.
   - Seedvault wird nun ein Backup Ihrer Daten erstellen und es auf Ihrem DAVx5-Mount speichern.
5. **Backup überprüfen**:
   - Nach Abschluss des Backup-Prozesses können Sie auf "Backup-Verlauf anzeigen" tippen, um sicherzustellen, dass das Backup erfolgreich durchgeführt wurde.
   - Optional können Sie sich auch in Ihrem DAV-Server einloggen und überprüfen, ob die Backup-Dateien korrekt hochgeladen wurden.
6. **Backup wiederherstellen (falls benötigt)**:
   - Falls Sie Ihr Gerät zurücksetzen oder auf ein neues Gerät wechseln, können Sie Seedvault verwenden, um Ihr Backup von DAVx5 wiederherzustellen.

- Gehen Sie dazu während des Einrichtungsprozesses zu "Backup wiederherstellen" und wählen Sie "DAVx5" als Quelle.

Anleitung: Wiederherstellung eines Seedvault-Backups auf einem neuen Gerät

Voraussetzungen:

- Ein neues Gerät, auf dem Sie Ihr Backup wiederherstellen möchten.
- Ein zuvor mit Seedvault auf DAVx5 erstelltes Backup.

Schritte:

1. **Einrichtung des neuen Geräts**:
   - Starten Sie Ihr neues Gerät und folgen Sie den Anweisungen auf dem Bildschirm.
   - Wenn Sie zum Wiederherstellen eines Backups aufgefordert werden, überspringen Sie diesen Schritt.
2. **DAVx5 installieren**:
   - Sobald Sie den Startbildschirm erreicht haben, öffnen Sie den Play Store oder eine andere vertrauenswürdige Quelle und installieren Sie die DAVx5-App.
   - Starten Sie DAVx5 und fügen Sie Ihr DAV-Serverkonto hinzu. Stellen Sie sicher, dass es korrekt synchronisiert wird und Zugriff auf Ihren Backup-Speicherort hat.
3. **Backup-Wiederherstellung initiieren**:
   - Öffnen Sie die Telefon-App auf Ihrem Gerät.
   - Geben Sie den Code `*#*#7378673#*#*` ein. Dies wird die Seedvault-Wiederherstellungsoberfläche öffnen.
   - Sie werden aufgefordert, einen Backup-Speicherort auszuwählen. Wählen Sie "DAVx5" aus der Liste.
4. **Backup auswählen und wiederherstellen**:
   - Seedvault zeigt Ihnen eine Liste der verfügbaren Backups auf Ihrem DAVx5-Mount. Wählen Sie das gewünschte Backup aus.
   - Folgen Sie den Anweisungen auf dem Bildschirm, um das Backup wiederherzustellen. Dies kann einige Minuten dauern, abhängig von der Größe des Backups und der Geschwindigkeit Ihrer Internetverbindung.
5. **Wiederherstellung überprüfen**:

- Nachdem die Wiederherstellung abgeschlossen ist, starten Sie Ihr Gerät neu.
- Überprüfen Sie, ob alle Ihre Daten, Apps und Einstellungen korrekt wiederhergestellt wurden.

XiaomiToolV2

Try this one: https://github.com/topminipie/XiaoMiToolV2

Or This: https://androidmtk.com/xiaomitool-v2

Solution

https://github.com/topminipie/awesome-xiaomi-bootloader-unlock#miunlocktool--mibypasstool--mitool

Download this: https://github.com/offici5l/MiUnlockTool

One device per Mi account per Month

Stress-Test

Stresstest unter Linux

Vorbereitung:

1. **Systemaktualisierung**: Aktualisieren Sie Ihr System, um sicherzustellen, dass Sie die neuesten Pakete und Treiber haben:

   ```
   sudo apt update
   ```

2. **Werkzeuge installieren**: Installieren Sie die benötigten Tools für den Stresstest:

   ```
   sudo apt install -y stress-ng htop lm-sensors glmark2
   ```

CPU-Stresstest:

1. **Test starten**: Belasten Sie alle CPU-Kerne für 10 Minuten:

   ```
   stress-ng --cpu 0 --timeout 10m
   ```

2. **Überwachung**: Verwenden Sie `htop` in einem anderen Terminalfenster, um die CPU-Auslastung zu überwachen.

CPU-Benchmark:

Use geekbench

```
paru -S geekbench
```

and run:

```
geekbench5
```

GPU-Benchmark:

1. **Test starten**: Führen Sie `glmark2` aus:

   ```
   glmark2
   ```

RAM-Stresstest:

1. **Test starten**: Belasten Sie den Speicher mit 4 Gigabyte für 10 Minuten:

   ```
   stress-ng --vm 1 --vm-bytes 4G --timeout 10m
   ```

2. **Überwachung**: Verwenden Sie `htop` in einem anderen Terminalfenster, um den RAM-Verbrauch zu überwachen.

Speicher-Stresstest (I/O):

1. **Test starten**: Führen Sie einen I/O-Stresstest für 10 Minuten durch:

   ```
   stress-ng --io 4 --timeout 10m
   ```

2. **Überwachung**: Verwenden Sie `htop` in einem anderen Terminalfenster, um die I/O-Aktivität zu überwachen.

Temperaturüberwachung:

1. **Sensoren konfigurieren**: Führen Sie `sensors-detect` aus und folgen Sie den Anweisungen:

   ```
   sudo sensors-detect
   ```

2. **Temperatur anzeigen**: Verwenden Sie `sensors`, um die Systemtemperaturen anzuzeigen:

   ```
   sensors
   ```

Wichtiger Hinweis: Stresstests können Ihr System stark belasten. Überwachen Sie die Temperaturen und andere Systemressourcen während des Tests. Wenn Sie Anzeichen von Problemen bemerken, beenden Sie den Test sofort. Es ist auch ratsam, regelmäßige Backups Ihrer Daten zu haben, bevor Sie solche Tests durchführen.

OBS-remove-Background

Installation des obs-virtual-cam-filter Plugins für OBS Studio auf Linux

Schritte:

1. **ZIP-Datei herunterladen**:
   - Besuchen Sie die Download-Seite des Plugins auf dem OBS-Forum.
   - Suchen Sie nach dem Download-Link für die Linux-Version des Plugins und laden Sie die ZIP-Datei herunter.
2. **ZIP-Datei entpacken**:
   - Navigieren Sie zum Download-Ordner und entpacken Sie die ZIP-Datei. Dies sollte einen Ordner mit den Plugin-Dateien enthalten.
3. **Plugin in OBS Studio installieren**:
   - Kopieren Sie den entpackten Ordner in das Plugin-Verzeichnis von OBS Studio. Der genaue Pfad kann je nach Installation variieren, liegt aber normalerweise unter `~/.var/app/com.obsproject.Studio/config/obs-studio/plugins/`
   - Wenn das Verzeichnis nicht existiert, erstellen Sie es.
4. **OBS Studio neu starten**:
   - Schließen Sie OBS Studio, falls es bereits geöffnet ist, und starten Sie es neu. Das Plugin sollte nun in der Liste der verfügbaren Plugins erscheinen.
5. **Plugin verwenden**:
   - Fügen Sie Ihre Webcam oder eine andere Quelle in OBS hinzu.
   - Klicken Sie mit der rechten Maustaste auf die Quelle und wählen Sie "Filter".
   - Klicken Sie auf das "+"-Symbol und wählen Sie "Virtual Camera Filter". Dies ermöglicht es Ihnen, diese spezifische Quelle als virtuelle Kamera auszugeben.

Virtuellen Hintergrund in OBS Studio auf Linux Mint einrichten

Voraussetzungen:

- Linux Mint Betriebssystem
- OBS Studio (vorzugsweise über Flatpak installiert)
- Webcam

Schritte:

1. **Virtuelle Kamera mit v4l2loopback einrichten**:
   - Installieren Sie das `v4l2loopback`-Modul:

     ```
     sudo apt-get install v4l2loopback-dkms
     ```

   - Erstellen Sie eine virtuelle Kamera:

     ```
     sudo modprobe v4l2loopback exclusive_caps=1
     ```

2. **OBS Studio über Flatpak installieren (falls noch nicht geschehen)**:
   - Fügen Sie das Flathub-Repository zu Flatpak hinzu:

     ```
     flatpak remote-add --if-not-exists flathub
     https://flathub.org/repo/flathub.flatpakrepo
     ```

   - Installieren Sie OBS Studio:

     ```
     flatpak install flathub com.obsproject.Studio
     ```

3. **Background Removal Plugin für OBS Studio installieren**:
   - Installieren Sie das `obs-backgroundremoval` Plugin über Flatpak:

     ```
     flatpak install com.obsproject.Studio.Plugin.BackgroundRemoval
     ```

4. **OBS Studio starten und Quellen hinzufügen**:
   - Starten Sie OBS Studio.
   - Fügen Sie Ihre Webcam als Videoquelle hinzu.
   - Optional: Fügen Sie Bilder oder Videos als Medienquelle hinzu, die als virtueller Hintergrund dienen sollen.
5. **Virtuellen Hintergrund einrichten**:

- Klicken Sie mit der rechten Maustaste auf Ihre Webcam-Quelle und wählen Sie "Filter".
- Klicken Sie auf das "+"-Symbol und wählen Sie "Background Removal". Dies wird den Hintergrund Ihrer Webcam entfernen.
- Platzieren Sie Ihre Webcam-Quelle über Ihrer Medienquelle in der Quellenliste, um den virtuellen Hintergrund anzuzeigen.

6. **OBS Output an die virtuelle Kamera senden**:
   - Gehen Sie in OBS zu "Datei" > "Einstellungen" > "Video".
   - Wählen Sie "Virtuelle Kamera starten". Ihr OBS-Output wird nun an die virtuelle Kamera gesendet.
7. **Virtuelle Kamera in anderen Anwendungen verwenden**:
   - Öffnen Sie eine Anwendung wie Zoom, Skype oder Teams.
   - Wählen Sie in den Kameraeinstellungen die Option "OBS Virtual Camera". Ihr Video-Feed sollte nun den virtuellen Hintergrund anzeigen.

Hinweis: Die Qualität der Hintergrundentfernung kann je nach Lichtverhältnissen, Webcam-Qualität und Raumhintergrund variieren. Es wird empfohlen, die Einstellungen vor einem Live-Stream oder Meeting zu testen.

Thin Client to Proxmox

Use this:

https://www.apalrd.net/posts/2022/raspi_spice/

script:

```
#!/bin/bash
set -e

# Set auth options
PASSWORD='remote1234'
USERNAME='remote@pve'

# Set VM ID
VMID="601"

# Set Node
# This must either be a DNS address or name of the node in the cluster
NODE="liz"

# Proxy equals node if node is a DNS address
# Otherwise, you need to set the IP address of the node here
PROXY="10.11.12.10"

#The rest of the script from Proxmox
NODE="${NODE%%\.*}"

DATA="$(curl -f -s -S -k --data-urlencode "username=$USERNAME" --data-urlencode
"password=$PASSWORD" "https://$PROXY:8006/api2/json/access/ticket")"

echo "AUTH OK"

TICKET="${DATA//\"/}"
TICKET="${TICKET##*ticket:}"
TICKET="${TICKET%%,*}"
```

```
TICKET="${TICKET%%\}*}"

CSRF="${DATA//\"/}"
CSRF="${CSRF##*CSRFPreventionToken:}"
CSRF="${CSRF%%,*}"
CSRF="${CSRF%%\}*}"

curl -f -s -S -k -b "PVEAuthCookie=$TICKET" -H "CSRFPreventionToken: $CSRF"
"https://$PROXY:8006/api2/spiceconfig/nodes/$NODE/qemu/$VMID/spiceproxy" -d "proxy=$PROXY" >
spiceproxy

#Launch remote-viewer with spiceproxy file, in kiosk mode, quit on disconnect
#The run loop will get a new ticket and launch us again if we disconnect
exec remote-viewer -k --kiosk-quit on-disconnect spiceproxy
```

Touchpad configs:

Add `/etc/X11/xorg.conf.d/40-libinput.conf`

```
Section "InputClass"
        Identifier "libinput touchpad catchall"
        MatchIsTouchpad "on"
        MatchDevicePath "/dev/input/event*"
        Driver "libinput"
        Option "Tapping" "on"
        Option "NaturalScrolling" "true"
EndSection
```

Kindersicherung

Time-keeper

```
sudo apt install -y timekpr
```

Then configure...

DNS-resolve

Use PI hole

Print with AB-88H

Driver

Download from here:

https://ddigital.pt/en/ab-88h

Install with

```
sudo bash setup_v1.0.0.sh
```

```
sudo apt install -y libcups2-dev
```

or try this:

```
sudo apt-get install libcupsimage2:i386
```

yt-dlp

Command zum schnellen download von music.youtube.com

```
yt-dlp --embed-metadata --embed-thumbnail -x --audio-format mp3 -f 251 -o '%(album)s/%(title)s.%(ext)s' <link>
```

To Download a complete cannel:

```
yt-dlp -f "bestvideo+bestaudio" --merge-output-format mkv --write-thumbnail --convert-thumbnails jpg -o "./%(playlist_title)s/%(title)s.%(ext)s" https://www.youtube.com/@CHANNEL_NAME/videos
```

> THis saves all videos with thumbnails in best quality in a subfolder CHANNEL_NAME - Videos""

To save a list of downloaded videos:

```
--download-archive "/path/to/archive.txt"
```

If you get Errors:

```
--cookies /path/to/cookies.txt
```

add this, with a coockie with your YouTube account

Add something to BTCmap.org

Edit location in openstreetmap.org

Add those Tags:

```
check_date=2024-04-04
check_date:currency:XBT=2024-04-05
currency:XBT=yes
payment:coins=yes
payment:lightning=yes
payment:onchain=yes
survey:date=2024-04-04
```

For more visit: https://wiki.btcmap.org/general/tagging-instructions.html#tagging-guidance

Make your own ISO

Install the tools:

```
sudo apt-add-repository universe
sudo apt-add-repository ppa:cubic-wizard/release
sudo apt update
sudo apt install --no-install-recommends cubic
```

Start app and select your start ISO.
In the chroot version make your channges

Davinci Resolve on Linux MINT

Download Davinci: https://www.blackmagicdesign.com/products/davinciresolve

Setup DistroBox and BoxBuddy

Extrackt and install in fedora 37 distroboy

Install dependensies:

```
sudo dnf install -y alsa-plugins-pulseaudio libxcrypt-compat xcb-util-renderutil xcb-util-wm pulseaudio-libs xcb-util xcb-util-image xcb-util-keysyms libxkbcommon-x11 libXrandr libXtst mesa-libGLU mtdev libSM libXcursor libXi libXinerama libxkbcommon libglvnd-egl libglvnd-glx libglvnd-opengl libICE librsvg2 libSM libX11 libXcursor libXext libXfixes libXi libXinerama libxkbcommon libxkbcommon-x11 libXrandr libXrender libXtst libXxf86vm mesa-libGLU mtdev pulseaudio-libs xcb-util alsa-lib apr apr-util fontconfig freetype libglvnd fuse-libs xcb-util-cursor
```

On amd GPU install

```
sudo dnf install -y rocm-opencl clinfo
```

and check:

```
clinfo
```

Nice scripts

Nice scripts

Converte Markdown File to PDF

Save file `convert_md_to_pdf.zsh`

```
#!/bin/zsh

# Überprüfe, ob das Argument für die Markdown-Datei übergeben wurde
if [ $# -eq 0 ]; then
    echo "Usage: $0 /path/to/markdown.md"
    exit 1
fi

# Pfad zur Markdown-Datei
input_file=$1
output_file="${input_file:r}.pdf"

# Überprüfen, ob Pandoc installiert ist
if ! command -v pandoc &> /dev/null; then
    echo "Pandoc could not be found. Please install Pandoc first."
    exit 1
fi

# Überprüfen, ob das Eisvogel-Template existiert
template_path="$HOME/templates/eisvogel.tex"
if [ ! -f "$template_path" ]; then
    echo "Eisvogel template not found. Downloading..."
    mkdir -p ~/templates
    wget -P ~/templates https://raw.githubusercontent.com/Wandmalfarbe/pandoc-latex-
template/master/eisvogel.tex
fi
```

```
# Pandoc-Befehl zur Umwandlung der Markdown-Datei in eine PDF-Datei
pandoc $input_file -o $output_file --from markdown --template $template_path --listings --toc

# Optionen erklärt:
# --from markdown: Gibt das Eingabeformat an
# --template eisvogel: Verwendet das Eisvogel-Template für eine schöne PDF-Ausgabe
# --listings: Aktiviert die Unterstützung für Code-Listings
# --toc: Fügt ein Inhaltsverzeichnis hinzu

echo "PDF erfolgreich erstellt: $output_file"
```

install needed tools:

```
sudo apt install -y pandoc texlive-full
```

Run Script:

```
zsh convert_md_to_pdf.zsh pwd/to/markdown.md
```

or:

```
bash convert_md_to_pdf.zsh pwd/to/markdown.md
```

Nice scripts

tmp

Logitech wrong mode:

```
usb_modeswitch -v 046d -p c26d -M 0f00010142 -C 0x03 -m 01 -r 01
```

SRT to RTMP

Requrements:

```
sudo apt install -y ffmpeg srt-tools
```

To test start a SRT stream with FFMPEG:

```
ffmpeg -re -i video.mp4 -c:v copy -c:a copy -f mpegts
"srt://127.0.0.1:2088?mode=caller&passphrase=mypassword"
```

start srt listener:

```
srt-live-transmit srt://:2088?passphrase=mypassword udp://127.0.0.1:23000
```

convert srt to rtmp with ffmpeg:

```
ffmpeg -i udp://127.0.0.1:23000 -c:v copy -c:a copy -f flv rtmp://192.168.1.129:1935/live
```

replace rtmp://192.168.1.129:1935/live with your RMP stream destination

Set as systemd

edit `/etc/systemd/system/srt-live-transmit.service`

```
[Unit]
Description=SRT Live Transmit Service
After=network.target

[Service]
ExecStart=/usr/bin/srt-live-transmit srt://:2088?passphrase=mypassword udp://127.0.0.1:23000
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target
```

change `srt://:2088?passphrase=mypassword`

edit `/etc/systemd/system/ffmpeg.service`

```
[Unit]
Description=FFmpeg RTMP Service
After=network.target srt-live-transmit.service
Requires=srt-live-transmit.service

[Service]
ExecStart=/usr/bin/ffmpeg -i udp://127.0.0.1:23000 -c:v copy -c:a copy -f flv
rtmp://192.168.1.129:1935/live
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target
```

change `rtmp://192.168.1.129:1935/live`

Aktivate:

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable srt-live-transmit
sudo systemctl enable ffmpeg
```

and start:

```
sudo systemctl start srt-live-transmit
sudo systemctl start ffmpeg
```

check:

```
sudo systemctl status srt-live-transmit
sudo systemctl status ffmpeg
```

Create P2P file share

Create BitTorrent

Install some packages

```
apt install -y transmission-cli opentracker
```

Configure Tracker

edit `/etc/opentracker/opentracker.conf`

check status:

```
systemctl status opentracker.service
```

create torrent file with transmission-gtk or transmission-cli:

```
transmission-create -o myfile.torrent -t udp://server-ip:80/announce file_to_share
```

Install-Standart-TMP

Standardinstallation Linux MINT

Zerotier

```
curl -s https://install.zerotier.com | sudo bash
```

Bitwarden

```
flatpak -y install com.bitwarden.desktop
```

OnlyOffice

```
flatpak -y install org.onlyoffice.desktopeditors
```

NextCloud sync Download Appimage -> GearLever

```
flatpak -y install GearLever
```

Element

```
sudo apt install -y wget apt-transport-https; \
sudo wget -O /usr/share/keyrings/element-io-archive-keyring.gpg
https://packages.element.io/debian/element-io-archive-keyring.gpg; \
echo "deb [signed-by=/usr/share/keyrings/element-io-archive-keyring.gpg]
https://packages.element.io/debian/ default main" | sudo tee /etc/apt/sources.list.d/element-
io.list; \
sudo apt update; \
sudo apt install element-desktop;
```

Jitsi Meet

```
flatpak -y install org.jitsi.jitsi-meet
```

buzz

```
snap install buzz
```

Telegram

```
flatpak -y install org.telegram.desktop
```

Signal

```
#Signal

wget -O- https://updates.signal.org/desktop/apt/keys.asc | gpg --dearmor > signal-desktop-keyring.gpg; \
cat signal-desktop-keyring.gpg | sudo tee /usr/share/keyrings/signal-desktop-keyring.gpg > /dev/null; \

echo 'deb [arch=amd64 signed-by=/usr/share/keyrings/signal-desktop-keyring.gpg] https://updates.signal.org/desktop/apt xenial main' |\
  sudo tee /etc/apt/sources.list.d/signal-xenial.list; \

sudo apt update && sudo apt install -y signal-desktop;
```

Caffeine

```
apt install -y caffeine
```

IINA Video Player Only for MacOS

Joplin

```
flatpak -y install net.cozic.joplin_desktop
```

https://www.youtube.com/watch?v=1SKaz1txeKA

VS Codium

```
wget -qO - https://gitlab.com/paulcarroty/vscodium-deb-rpm-repo/raw/master/pub.gpg \
    | gpg --dearmor \
    | sudo dd of=/usr/share/keyrings/vscodium-archive-keyring.gpg
```

```
echo 'deb [ signed-by=/usr/share/keyrings/vscodium-archive-keyring.gpg ]
https://download.vscodium.com/debs vscodium main' \
    | sudo tee /etc/apt/sources.list.d/vscodium.list
```

```
sudo apt update && sudo apt install -y codium
```

ffmpeg

```
apt install -y ffmpeg
```

Manuelle Installation

Brave Browser

```
sudo apt install curl; \
sudo curl -fsSLo /usr/share/keyrings/brave-browser-archive-keyring.gpg https://brave-browser-
apt-release.s3.brave.com/brave-browser-archive-keyring.gpg; \
echo "deb [arch=amd64 signed-by=/usr/share/keyrings/brave-browser-archive-keyring.gpg]
https://brave-browser-apt-release.s3.brave.com/ stable main"|sudo tee
/etc/apt/sources.list.d/brave-browser-release.list; \
sudo apt update; \
sudo apt install -y brave-browser;
```

Brave Browser Brave Browser Sync Plugins

- bitwarden Plugin
- Archive.is-Plugin
- Kutt

Titan Email

- In Konto einloggen
- Titan PWA über Brave installieren

Schriften

```
apt install -y fonts-ibm-plex fonts-open-sans ttf-mscorefonts-installer
```

Edit Metadata

Add Licenz and Copyright:

```
exiftool -overwrite_original -Copyright="Type Info" -Rights="Type Info" ./*.jpg
```

Davinci Resolve

Install Davinci Resolve on LinuxMINT

> !!Still in Developement!!

Downloads

- Davinci: https://www.blackmagicdesign.com/products/davinciresolve
- Make DEV: https://www.danieltufvesson.com/makeresolvedeb

Install tools

```
sudo apt install -y xorriso libpango1.0-dev ocl-icd-opencl-dev
```

Run script

Extracet all files into one Folder:

```
ls -l
total 2984372
-rwxr-xr-x 1 joscha_mijailovic joscha_mijailovic 3055871200 Sep  4 03:26
DaVinci_Resolve_19.0.1_Linux.run
-rw-r--r-- 1 joscha_mijailovic joscha_mijailovic      56033 Sep  2 07:22
Linux_Installation_Instructions.pdf
-rwxr-xr-x 1 joscha_mijailovic joscha_mijailovic      32986 Aug 22 19:15
makeresolvedeb_1.7.2_multi.sh
```

then run the script:

```
./makeresolvedeb_1.7.2_multi.sh DaVinci_Resolve_19.0.1_Linux.run
```

Then install like this:

```
sudo dpkg -i davinci-resolve_19.0.1-mrd1.7.2_amd64.deb
```

After going through all the steps, if it doesn't start open a terminal and start resolve with: Code: Select all

`/opt/resolve/bin/resolve` if you get: `symbol lookup error: /lib64/libpango-1.0.so.0: undefined symbol: g_string_free_and_steal`

Do the following

```
sudo cp /usr/lib/x86_64-linux-gnu/libglib-2.0.so.0 /opt/resolve/libs/
```

if it still doesn't run and you get: symbol lookup error: /opt/resolve/bin/../libs/libgdk_pixbuf-2.0.so.0: undefined symbol: g_task_set_static_name Or this one, symbol lookup error: /lib/x86_64-linux-gnu/libpango-1.0.so.0: undefined symbol: g_once_init_leave_pointer Do this;

```
cd /opt/resolve/libs
sudo mkdir not_used
sudo mv libgio* not_used
sudo mv libgmodule* not_used
```

Satisfactory Steam server

Manuelle Installation von SteamCMD

1. **SteamCMD-Verzeichnis erstellen**: Erstelle ein Verzeichnis für SteamCMD und wechsle dorthin:

   ```
   mkdir ~/steamcmd
   cd ~/steamcmd
   ```

2. **SteamCMD herunterladen**: Lade SteamCMD mit `wget` herunter und entpacke es:

   ```
   wget https://steamcdn-a.akamaihd.net/client/installer/steamcmd_linux.tar.gz
   tar -xvzf steamcmd_linux.tar.gz
   ```

3. **SteamCMD ausführen**: Führe SteamCMD aus, um es zu installieren:

   ```
   ./steamcmd.sh
   ```

Get server

1. **Installationsverzeichnis festlegen**: Gib das Installationsverzeichnis für den Satisfactory Server an:

   ```
   force_install_dir /embassy-data/package-data/steam/steamcmd/satisfactory_server
   ```

2. **Einloggen als anonymer Benutzer**: Melde dich danach als anonymer Benutzer an:

   ```
   login anonymous
   ```

3. **Satisfactory Dedicated Server herunterladen**: Installiere den Dedicated Server:

   ```
   app_update 1690800 validate
   ```

4. **SteamCMD beenden**: Sobald der Download abgeschlossen ist:

   ```
   quit
   ```

Thinkpad Fingerprint on Linux

Check if your fingerprint device is supported:

Find youre Device:

```
lsusb
```

Output:

```
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 04f2:b6be Chicony Electronics Co., Ltd Integrated Camera
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 003 Device 002: ID 27c6:55a4 Shenzhen Goodix Technology Co.,Ltd. Goodix FingerPrint Device
Bus 003 Device 003: ID 8087:0029 Intel Corp. AX200 Bluetooth
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
```

in my case the fingerprin-sensor ist:

```
Bus 003 Device 002: ID 27c6:55a4 Shenzhen Goodix Technology Co.,Ltd. Goodix FingerPrint Device
```

> This Sensor ist NOT Supported!

Check on this website:

https://fprint.freedesktop.org/supported-devices.html

Install driver and support

```
sudo apt install -y fprintd; \
sudo apt install -y libpam-fprintd;
```

To enable on login:

```
sudo pam-auth-update
```

Run Steam in gamepadui/big-picture mode at boot

Pre config

> You just need auto login an steam

Systemd-service

create sytemd service:

`/etc/systemd/system/steam-bigpicture.service`

```
[Unit]
Description=Steam Big Picture Mode
After=network.target

[Service]
User=YOUR_USER
ExecStart=/usr/bin/steam -gamepadui
Restart=on-failure
Environment=DISPLAY=:0
WorkingDirectory=/home/YOUR_USER

[Install]
WantedBy=graphical.target
```

Or as auto-run script

Create `/etc/xdg/autostart/steam-big-picture.desktop`

Or: `~/.config/autostart/steam-big-picture.desktop`

```
[Desktop Entry]
Type=Application
Exec=steam -gamepadui
Hidden=false
NoDisplay=false
X-GNOME-Autostart-enabled=true
Name=Steam Big Picture
```

Importing Open Source Models to Ollama

Clone from huggingface.co:

```
apt install git-lfs
```

```
git clone https://huggingface.co/nvidia/Llama-3.1-Nemotron-70B-Instruct-HF
```

Import to ollama

Create Modelfile

```
# Modelfile
FROM "./"
PARAMETER stop "<|im_start|>"
PARAMETER stop "<|im_end|>"
TEMPLATE """
<|im_start|>system
{{ .System }}<|im_end|>
<|im_start|>user
{{ .Prompt }}<|im_end|>
<|im_start|>assistant
"""
```

```
ollama create Llama-3.1-Nemotron-70B-Instruct-HF
```

To also quantize the model:

```
ollama create Llama-3.1-Nemotron-70B-Instruct-HF:q4_0 --quantize q4_0
```

See more Optiones here: https://github.com/ollama/ollama/blob/main/docs/import.md#supported-quantizations