

Linux for Media Production

(The Basics)



© 2025 PixelPirate69420 — Licensed under CC BY-NC-SA 4.0

<https://github.com/PixelPirate69420/>

<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.en>

Table of Contents



Step 0: Planning	4
Step 1: Choosing the hardware	4
Generic Hardware chart for 2025:	4
(compatibility with Linux) – Processors	5
(compatibility with Linux) – Graphics Cards	5
(compatibility with Linux) – Network Cards	6
(compatibility with Linux) – Peripherals	6
Step 2: Choosing a Linux distribution	7
What is Linux exactly?	7
Recommended Linux setups and the availability of proprietary drivers	8
Linux Distributions Compared and Benchmarked	9
Step 2.5: Post Install troubleshooting (if necessary)	14
Failed services?	14
Incomplete Drivers (or Packages)?	14
Missing dependencies?	16
Broken bootloader(or other essential components)? Try:	17
Sound not working?	18
Network not working?	18
Switching Kernel	19
Broken Desktop Environment?	20
Broken Login App – loop or simply not working?	20
AMD Drivers on Linux	20
Nvidia Drivers on each distribution:	21
Intel Drivers	21
Step 2.7 : Tweaking a Distribution for Media Production	22
Step 3: Finding the software for media work	23
Video Editing Software available	24
Image Editing Software available	29
Streaming and Recording Software available	33
Screen Recorders Software available	33
Sound Editing Software available	35
3D Modeling Software and Animation	36

What is this guide based on?

Personal experience setting up small to medium scale media workstations and thousands of comments and opinions from online forums.

Why?

Because Linux can offer various advantages when it comes to media production, such as:

1. **More efficient resource management** – lower CPU clocks being hogged by the OS, lower RAM consumption and overall a better performance. Based on a few personal tests, there can be performance gains of over 10% on powerful hardware.
2. **More stable compared to NT(any)** – Linux follows the Unix-like philosophy and it proved its abilities by running more than 90% of the servers that hold the internet together. It means that the core of Linux is very solid.
3. **Better workflow** – Linux can adapt to your specific needs, from a very comfortable Desktop Environment to integrating it into any sort of ecosystem.
4. **Better security** – Linux can become very safe if hardened properly.
5. **Good Hardware compatibility** – Linux can in fact run the hardware both with open source drivers and proprietary drivers (based on the specific needs).
 - Nvidia drivers releases are very constant, CUDA libraries run correctly, the documentation is good enough to make it clear what is the right way to install the drivers
 - AMD is supported extremely well, both with Radeon and Radeon PRO drivers.
 - Intel GPUs can run correctly under Linux.
6. **Lower Costs for software** – Due to the fact that the Linux environment is developed under GNU license, it means that it is free of cost and built on open source software.
7. **No annoying updates** – The Linux environment allows users and admins to decide when is the right time for an update, both for the whole system or just individual software. This is also a feature that allows for better time management.

Why not?

1. If tweaking is out of reach and picking specific hardware is a hassle.
2. Needing specific proprietary software that simply can't run under Linux.
3. Already owning hardware that is tied to a closed existing ecosystem.

What can it do?

1. Streaming
2. Video Editing
3. Photo Editing
4. Audio Correction
5. 3D modeling
6. 2D/3D animation

Step 0: Planning

This step holds is important in deciding the software necessary for the specific workload (this decision will influence later on various aspects of both software and hardware). Key questions:

1. “Stability or Bleeding Edge software?”
2. “Amateur or Professional?”
3. “Long Term or Short Term workstation?”
4. “Is security a concern?”
5. “Polished or Light software?”
6. “Standalone Distribution or a Fork?”

Step 1: Choosing the hardware

This step includes market research, finding the best deals based on the geographical location and deciding on the best ratio between: price/performance/reliability

The main questions should always be:

1. “What will this workstation be used for?”
2. “What’s the budget?”
3. “New or used? If used, what’s the risk tolerance? ”
4. “Desktop or Laptop?” - based on the environment where the computer will be used. Desktops provide better cooling and more space to fit more powerful GPUs, while laptops are better when it comes to energy efficiency.
5. “Should I care about future-proofing?” - Yes, definitely.
6. “How can hardware be reliable?” - Good cooling, Proper Power Supply, CPU – GPU – RAM running at the right voltage

Generic Hardware chart for 2025:

	CPU	RAM	GPU	Resolution	Storage
Budget	Ryzen 5 or Intel i5	16 GB DDR4 16 GB DDR5	Integrated	1920x1080	SSD
Medium	Ryzen 7 or Intel i7	32 GB DDR5	RTX 20 series	1920x1080	SSD
Professional	Ryzen 9 or Intel i9	64GB DDR5	RTX 40 series	2560x1440	SSD + HDD
Dreamworks-like CPU rendering	Intel Xeon / Ryzen Threadripped	128GB	Integrated	2560x1440	SSD + HDD

Minimal specs for future proofing:

1. Good cooling
2. At least 16 GB of RAM – at least DDR3 (for tight budgets)
3. At least 6 cores or 10 threads

4. A GPU that has the performance of a GT 1030 in processing, 4GB vram and good graphics API support (recent OpenGL, OpenCL and Vulkan support) .
5. SSD M.2 or SATA (boot drive) → 1 TB
6. A screen of at least 1080p 60hz
7. Checking If drivers are available for Linux (if possible)

(compatibility with Linux) – Processors

When it comes to architecture, the support is great, all the way from x86-64 to MIPS.

When talking about the most popular architectures: **x86-64** and **ARM64** (RISC-V is far from competing directly) are the way to go. These architectures get the most software support, the best drivers and are the most future-proof (no 32-bit).

Going with any other architecture for a Linux workstation would take more skill to set up and more troubleshooting.

Required x86 instructions for rendering (according to FFmpeg):

1. MMX2
2. SSE2Fast
3. SSE3
4. SSE4.2
5. AVX
6. FMA3
7. BMI2
8. AVX2

Handy instruction sets:

9. AVX512
10. AVX512ICL

(https://github.com/FFmpeg/asm-lessons/blob/main/lesson_03/index.md)

“How to check the CPU architecture?” - In Linux with a software called CPU-X. In case of checking before purchasing, the vendor site must cover it (intel or amd for most buyers)

(compatibility with Linux) – Graphics Cards

- AMD APUs are supported pretty well in the 2025 Linux distros, especially those released after 2020 (Radeon 660M is the bare minimum, personal opinion <https://www.techpowerup.com/gpu-specs/radeon-660m.c3870>)
- AMD dGPUs (for laptops are almost extinct, still relevant for the 2018 to 2020 Macbook Pro) are also supported with official AMD drivers.
Available : Radeon and Radeon PRO drivers
(In some cases AMD GPUs are supported better than Nvidia on Linux)

<https://www.amd.com/en/support/download/linux-drivers.html>

- Intel iGPUs are well supported in Linux as well (generally, they work out of the box on the main Linux distros) – Intel UHD 770 is the minimum for media production.
<https://www.intel.com/content/www/us/en/support/topics/linux-osv.html>
- Nvidia dGPUs out of the box might be paired with the nouveau drivers (open source and reversed engineered drivers) which are not the most efficient and lack CUDA implementation and so, performance is not great. Sometimes the performance can be cut in half, not ideal. What would the second option be? **Proprietary drivers**, they work just like NVIDIA intended and can be installed on Linux, tho with some efforts (requires some research and some troubleshooting skills based on the Linux distribution, but it is worth it in the end)
<https://docs.nvidia.com/datacenter/tesla/driver-installation-guide/index.html>

“Which one is better, Nvidia or AMD for GPUs on Linux?”

It depends on the specific needs, while AMD GPUs are easier to set up and get them running, some editing software can utilize Nvidia APIs better (Kdenlive, DaVinci Resolve).

(compatibility with Linux) – Network Cards

As a general fact, most WiFi and Network cards (Intel, Atheros, Broadcom, Realtek) are supported well on Linux with some exceptions (some Macbook WiFi cards may not work perfectly and very obscure cards). As a general tip, an external dongle is welcomed in Linux if any network troubleshooting is required.(if not, USB tethering can work as well). Network Card troubleshooting might consist of: checking the kernel version, checking the drivers, checking the firmware version.

The compatibility may vary a bit based on the distribution of choice. Linux Distributions which have newer repositories have higher chances of increasing the compatibility(Ubuntu, Mint, Xubuntu, etc.) while those with older repositories (like Debian might lag behind at times).

(compatibility with Linux) – Peripherals

In most of the cases USB and PS/2 keyboards and mice will work fine under Linux, except some users reported on various forums issues on Acer and HP Victus laptops due to some driver issues which were solved post install. For such situations it is useful to keep around some external keyboards and mice.

Most Webcams and Microphones are also known to work alright on Linux. Capture cards can work, depending on the model and method of capturing. Generic USB External Audio Cards also work as expected on Linux. More specific interfaces are covered on their specific guides. Jack ports are also known to work out of the box in most cases.

Step 2: Choosing a Linux distribution

What is Linux exactly?

Linux is actually a kernel – a software component that controls the resources allocation. What everyone is running on their computers and calling “Linux” is more than just the kernel, it’s a whole environment built from open source components. (In many cases including the GNU project except Linux Alpine)

Who developed the Linux kernel? - Linus Torvalds (started as a hobby project back in 1991)

Who developed the GNU project? - Richard Stallman (started with the hope of creating the first fully open source operating system in 1988, except it lacked a kernel)

What are the components in a Linux environment?

1. **Bootloader** – It tells the computer when it’s booting up “Heyy I am a bootable partition, pick me!” and then points out to the disk partition where it can finally hand over the job to the →
2. **Kernel** – as stated earlier it assigns the available resources, handles multitasking (preemptive) and so, it allocates resources for each task and driver needed, which leads to→
3. **Init System** – starts up all of services and skips those who are not able to run →
4. **Services** – Necessary tasks that have to run at all time, including the →
5. **Display Server** – A protocol which is able to take a GUI and display it (like a framework). This soft even has an automatic configuration which leads to the→
6. **Desktop Environment** – The graphic interface which makes everything a lot easier.

User: John

Password:*****

Successfully logged in!

- 7. User land** → All of the tools the user will interact with, including: package manager, shell command scripts, editors, etc.

Now that this is out of the way, there are many Linux Distributions which can be used for media production, each with ups and downs aspects. There are no “ideal” or “perfect” distributions, some may have better software support, others may perform better in terms of lightness. Here are some generic profiles which can cut the decision time to nearly seconds.

Recommended Linux setups and the availability of proprietary drivers

Media Editing Profiles	Linux Distribution	Desktop Environment	Display Servers	Proprietary GPU Drivers
Beginner	Ubuntu / Mint / Manjaro	Gnome / Cinnamon	Xorg	Nvidia – Good AMD - Good
Advanced	Debian / Arch / Fedora / OpenSUSE	Gnome / XFCE / KDE / Cinnamon	Xorg	Nvidia – Good AMD - Good
Media Professional	Rocky / CentOS	Gnome / KDE	Wayland	Nvidia – 566 + AMD – 22.Q2 +

Xorg VS Wayland – For media production (latency)

Xorg – takes 6 steps for rendering the output (<https://www.linuxmo.com/wayland-vs-x11-the-battle-of-display-protocols/>)

Wayland – takes 4 steps for rendering output and allows for direct GPU communication (<https://www.linuxmo.com/wayland-vs-x11-the-battle-of-display-protocols/>)

What does it mean? - Wayland can indeed offer lower latency for I/O and image rendering on a display, due to a simpler and faster process.

What if X Wayland is required? - Then the I/O would have to go through another layer of translation which would likely increase the latency time(ms).

Conclusion? - Wayland can offer better performance on Wayland native applications.

What if an application is not Wayland native? - It will run through X Wayland (compatibility layer)

Why does Xorg receives better support? - Inherited from previous POSIX systems and it proved over the decades that it can keep up with modern Desktop Environments.

Linux Distributions Compared and Benchmarked

Now, for a deeper read here are 9 important distributions being analyzed (in real conditions, running in a KVM) and compared for media production:

- 1. Ubuntu** – Great for beginners, has plenty of software, polished GUI tools, can manage drivers, stable and easy to update but is not the leanest system.
The Installation Process – can be done in 15 minutes with general knowledge. (GUI)
 - **Support for DaVinci Resolve – Community Supported**
 - **Repository Updates: every 6 months**
 - **Software Support: Standard 5 years, LTS: 10 years**
 - **Available Packages for x86-64: 76.992 (<https://pkgs.org/>) VM checked: 83.378**
 - **System lightness Test : X11 – Heavy (49) Wayland – Average (52)**
(Ubuntu 24.04 + Gnome(x11 and Wayland) on VM: 4GB RAM and 1 thread kvm)

Ram usage - idling:

 - Wayland – 1.2GB (0kb swap) – 8 points + 5
 - X11 – 1.3GB (0kb) – 7 points + 5

CPU usage – idling:

 - Wayland : 33% to 35% → 1.3GHz (Taxed from 4.0GHz) – 13 points
 - X11 : 41% to 43% → 1.6 GHz to 1.7GHz (Taxed from 4.0GHz) – 9 points

Number of packages installed by default: 1575 (APT) + 9 (Snap) = 1584 – 9 points

Number of services running by default:

 - Wayland : 55 – 0 points
 - X11 : 55 – 0 points

Size on Disk: 11.4GB – 9 points
X11 – 10 points
Wayland – 8 points

 - **Possible Memory leaks reported in 2025 – Under personal tests it never happened**
(https://www.reddit.com/r/linuxquestions/comments/1kjckoh/creeping_idle_ram_usage_on_ubuntu_2504/)
- 2. Mint** – Very similar to Ubuntu in many aspects, except it is more targeted for Windows users, therefore it Cinnamon Desktop Environment.
The installation Process – can be done in 15 minutes with general knowledge. (GUI)
 - **Support for DaVinci Resolve – Community Supported**
 - **Repository Updates: every 6 months**
 - **Software Support: 5 years**
 - **Available Packages for x86-64: 83.729 (VM checked)**
 - **System lightness Test : Cinnamon – Light (60) Gnome – Average (58)**

(Mint 22 + Cinnamon and Gnome(x11) on VM: 4GB RAM and 1 thread kvm)

Ram usage – idling:

- Cinnamon : 925MB – 11 points + 5
- Gnome: 1.6GB – 4 points + 5

CPU usage – idling:

- Cinnamon : 41% to 43% → 1.6GHz to 1.7GHz (Taxed from 4.0GHz) – 9 points
- Gnome: 28% to 31% → 1.1GHz to 1.2GHz (Taxed from 4.0GHz) – 15 points

Number of packages installed by default: 1926 – 6 points

Number of services running by default:

- Cinnamon: 28 – 11 points
- Gnome: 31 – 10 points

Size on Disk: 12.3 GB - 8 points

X11 – 10 points

3. **Fedora** – Great for average users, focused around new software, it feels very polished.

The installation process – can be done in about 30 minutes with a bit of specific knowledge (GUI)

- **Support for DaVinci Resolve – Community Supported**
- **Repository updates: every 6 months**
- **Support: 13 Months (No LTS)**
- **Available Packages for x86-64: 100.070 (<https://pkgs.org/>)**
- **System lightness Test : Average (54)**

(Fedora 42 + Gnome(wayland) on VM: 4GB RAM and 1 thread kvm)

Ram usage – idling: 1.4GB (0kb swap used) – 6 points + 5

CPU usage – idling: 36% to 38%→ 1.5GHZ (Taxed from 4.0GHz) – 12 points

Number of packages installed by default: 2014 – 5 points

Number of services running by default: 43 – 4 points

Size on Disk: 5.8 GB – 14 points

Wayland – 8 points

4. **Debian** – Arguably one of the most flexible distributions, can be used by beginners and advanced users, focused on stability and software checking (sometimes this leads to serious lag when it comes to new software releases). Not to mention that it can be a very lean or polished system with the right software in mind.

The installation process – can be done in about 30 minutes with a bit of specific knowledge. (GUI/CLI)

- **Support for DaVinci Resolve – Can Install but NOT Official (Community Supported)**
- **Repository updates: every 2 weeks**
- **Software Support: Standard 11 Months, LTS: 5 Years**
- **Available Packages for x86-64: 65.205 (<https://pkg.org/>) VM: 63.533**
- **System lightness Test : Light (71)**

(Debian 12 + Gnome(x11) on VM: 4GB RAM and 1 thread kvm)

Ram usage – idling: 1.3GB (0kb swap used) – 7 points + 5

CPU usage – idling: 25% to 26%→ 1GHZ (Taxed from 4.0GHz) – 15 points

Number of packages running by default: 1594 – 9 points

Number of services running by default: 25 – 13 points

Size on Disk: 8.1GB – 12 points

X11 – 10 points

5. **Arch** – Almost the polar opposite to Debian when it comes to software releases, it gets updates really quickly, meanwhile stability is not a good compared to Debian because in most of cases the latest releases of any software are not so well tested. The installation process – can be done in about 1 hour with specific knowledge (CLI) traditionally, unless archinstall is being used (30 minutes).
 - **Support for DaVinci Resolve – Community Supported**
 - **Repository updates: Multiple times per day**
 - **Software Support: As long as you keep updating (No term)**
 - **Available Packages for x86-64: 36.002 (<https://pkgs.org/>) 14.638 (VM checked - pacman)**
 - **System lightness Test : Light (78)**
(Arch + Gnome(x11) on VM: 4GB RAM and 1 thread kvm)
Ram usage – idling: 1.0GB (0kb swap used) – 10 points + 5
CPU usage – idling: 40% to 44% → 1.6 to 1.7GHz (Taxed from 4.0GHz) – 9 points
Number of packages running by default: 747 – 15 points
Number of services running by default: 15 – 15 points
Size on Disk: 6.0GB – 14 points
X11 – 10 points
6. **Manjaro** – Based on Arch, it aims to be more user friendly with a well polished interface and great repositories. The installation process – can be done in about 30 minutes with a bit of specific knowledge (GUI)
 - **Support for DaVinci Resolve – Community Supported**
 - **Repository Updates: every few weeks**
 - **Software Support: As long as you keep updating (No term)**
 - **Available Packages for x86-64: 36.002 (<https://pkgs.org/>) 31.184 (VM checked)**
 - **System lightness Test : Light (72)**
(Manjaro 25 + Gnome(x11) on VM: 4GB RAM and 1 thread kvm)
Ram usage – idling: 1.2GB (0kb swap used) – 8 points + 5
CPU usage – idling: 46% to 49% → 1.8GHz to 1.9GHz (Taxed from 4.0GHz) – 6 points
Number of packages installed by default: 1120 – 14 points
Number of services running by default: 20 – 15 points
Size on Disk: 5.9 GB – 14 points
X11 – 10 points
7. **OpenSUSE** – A free and open source enterprise level Linux distro developed by SUSE
The installation process – can be done in about 30 minutes with specific knowledge (GUI)
 - **Support for DaVinci Resolve – Community Supported**
 - **Repository Updates: every week (manually)**
 - **Software Support: 18 Months**

- **Available Packages for x86-64: 72.207 (<https://pkgs.org/>) 83.028**
 - **System lightness Test : X11 – Average (50) Wayland – Average (53)**
(OpenSUSE 15.5 + Gnome(x11) and Gnome(Wayland) on VM: 4GB RAM and 1 thread kvm)
Ram usage:
 - X11 idling: 1.3GB (0kb swap used) – 7 points + 5
 - Wayland idling 1.3GB (13.5MB Swap) - 7 points + 5
 CPU usage
 - X11 idling: 31%→ 1.2GHz (Taxed from 4.0GHz) – 15 points
 - Wayland idling: 17%-18.8% → 600MHz to 752MHz (Taxed from 4.0GHz) – 20 points
 Number of packages installed by default: 4051 – 0 points
 Number of services running by default:
 - X11: 60 – 0 points
 - Wayland: 61 – 0 points
 Size on Disk: 6.6GB – 13 points
 X11 – 10
 Wayland - 8
8. **CentOS** – Discontinued, but it was an alternative to OpenSUSE for the same enterprise level.
 The installation is easy, can be done in 30 minutes without specific knowledge (GUI)
- **Was Supported by DaVinci Resolve**
 - **Repository Updates: Never, Discontinued**
 - **Software Support: Till 2030**
 - **Available Packages for x86-64: 20.934 (<https://pkgs.org/>) 17.497 (VM checked)**
 - **System lightness Test : Average (53)**
(CentOS 10 + Gnome(Wayland) on VM: 4GB RAM and 1 thread kvm)
 Ram usage – idling: 1.4GB (0kb swap used) – 6 points + 5
 CPU usage – idling: 34% to 35% (Taxed from 4.0GHz) – 13 points
 Number of packages installed by default: 5491 – 0 points
 Number of services running by default: 32 – 9 points
 Size on Disk: 7.9 GB – 12 points
 Wayland – 8 points
9. **Rocky** – Yet another enterprise level Linux distribution, but released in 2020 by Rocky Enterprise Software.
 The installation process – can be done in about 30 minutes with a bit of specific knowledge (GUI)
- **Support for DaVinci Resolve – Official**
 - **Repository Updates: Multiple times per day**
 - **Software Support: 10 Years**
 - **Available Packages for x86-64: 19.511 (<https://pkgs.org/>) 5.831 (VM checked)**
 - **System lightness Test : Average (58)**
(Rocky 10 + Gnome(Wayland) on VM: 4GB RAM and 1 thread kvm)
 Ram usage – idling: 1.3GB (0kb swap used) – 7 points + 5
 CPU usage – idling: 32% to 35% → 1.3GHz (Taxed from 4.0GHz) – 13 points
 Number of packages installed by default: 1229 – 13 points

Number of services running by default: 56 – 0 points

Size on Disk: 8.3 GB – 12 points

Wayland – 8 points

System Lightness Test Explained:

Specifications used for testing

Host	Debian 12	i7 8550u	8GB RAM DDR4	SSD M.2
Guest	QEMU KVM	1 Thread	4GB	Raw IMG file

1. Why 1 thread and so low specifications?
In order to make it easy to spot the differences while idling.
2. Where do these pieces of data come from?
System Monitor (Gnome tool) – htop measures at the kernel level (hard to spot any differences)
3. What is it meant to prove?
Different Linux Distributions provide different environment which lead to different ways of managing the hardware and different ways of handling everyday programs.
4. Is it meant to promote a specific distribution?
Absolutely not, each distribution is serving a market.

Scoring System

Metric	Score	Method
CPU	20	$20 - ((\text{avg_cpu \%} - 20) / 2)$
RAM	20	$20 - (\text{ram_idle} * 10)$
Running Services	15	$15 - ((\text{services} - 20) * 0.5)$
Installed Packages	15	$15 - (\text{packages} - 1000) / 100)$
Disk Size	15	$15 - (\text{disk_size} * 1)$
Swap	5	(subtract 1 point for each 50mb)
Display Server	10	Wayland = 8 X11=10

(the points are rounded up and down: .49 = 0 and .50=1)

(final values can not go under 0 or beyond the maximum score allowed)

Why these numbers and methods specifically?

Making it harder for distribution to go beyond 100 points.

Range	Result
80 - 100	Very Light
60 - 79	Light
40 - 59	Average
20 - 39	Heavy
0 - 19	Bloated

Step 2.5: Post Install troubleshooting (if necessary)

The machine just had its first bootup, congratulations! But something seems off, maybe the network card is not recognized, maybe some services do not work properly, maybe the GPU is not performing as it should or maybe the installer skipped some drivers... How to deal with that?

Failed services?

Luckily many Linux Distributions use systemd which is easy to debug
`systemctl --failed` #should display in terminal if any service failed to start

No services mentioned? Great, everything started fine

```
sudo systemctl reload <service_name> # try this the service doesn't work
sudo systemctl restart <service_name> # still nothing? Try this
journalctl -u <service_name> # for starting to debug in case nothing works
```

At least one service failed? Deal with the dependencies required and then reload or restart service

Incomplete Drivers (or Packages)?

(Ubuntu / Mint / Debian)

1. Update Linux Headers

```
sudo apt update
sudo apt --reinstall install linux-headers-$(uname -r)
```

2. Purge old drivers

```
sudo apt purge '^[driver-name]-.*'
```

```
sudo apt autoremove  
sudo apt autoclean
```

3. Install the new drivers

```
sudo apt install [driver-name]
```

(Manjaro / Arch)

1. Update Linux Headers

```
sudo pacman -Syu  
sudo pacman -S linux-headers # or linux-lts-headers if using LTS kernel
```

2. Purge old drivers

```
sudo pacman -Rns $(pacman -Qq | grep '[driver_name]')  
sudo pacman -Rns $(pacman -Qq | grep '[driver_name]')
```

3. Install the new drivers

```
sudo pacman -S [driver_name]
```

(Fedora)

1. Update Linux Headers

```
sudo dnf update  
sudo dnf install kernel-devel kernel-headers
```

2. Purge old drivers

```
sudo dnf remove '[driver_name]'  
sudo dnf autoremove
```

3. Install the new drivers

```
sudo dnf install https://download1.rpmfusion.org/free/fedora/rpmfusion-free-release-  
$(rpm -E %fedora).noarch.rpm  
sudo dnf install https://download1.rpmfusion.org/nonfree/fedora/rpmfusion-nonfree-  
release-$(rpm -E %fedora).noarch.rpm
```

```
sudo dnf install [driver_name]
```

(OpenSUSE)

1. Update Linux Headers

```
sudo zypper refresh  
sudo zypper install kernel-default-devel kernel-devel
```

2. Purge old drivers

```
sudo zypper remove [driver_name]*
```

3. Install the new drivers

```
sudo zypper addrepo --refresh [driver repository address]  
sudo zypper refresh  
sudo zypper install [driver_name]
```

(CentOS / Rocky)

1. Update Linux Headers

```
sudo dnf update  
sudo dnf install kernel-devel kernel-headers
```

2. Purge old drivers

```
sudo dnf remove '[driver_name]'  
sudo dnf autoremove
```

3. Install the new drivers

```
sudo dnf install epel-release  
sudo dnf install https://mirrors.rpmfusion.org/free/el/rpmfusion-free-release-  
8.noarch.rpm  
sudo dnf install https://mirrors.rpmfusion.org/nonfree/el/rpmfusion-nonfree-release-  
8.noarch.rpm  
  
sudo dnf install [driver_name]
```

Missing dependencies?

each package manager has its command to deal with this situation

One easy way would be installing “aptitude”

The other, managing it by distribution:

APT – Ubuntu / Mint / Debian

```
# Refresh package lists & fix missing dependencies  
sudo apt update  
sudo apt install -f
```

```
# Check for broken packages  
sudo apt-get check
```

```
# Reinstall a broken package if needed  
sudo apt install --reinstall [package]
```

DNF – Fedora / RHEL / Rocky / CentOS

```
# Update metadata and all packages (including fixes)  
sudo dnf upgrade --refresh
```

```
# Install or fix dependencies when installing a package  
sudo dnf install [package]
```

```
# List missing/unsatisfied deps  
sudo dnf repoquery --unsatisfied
```

```
# Reinstall a specific package  
sudo dnf reinstall [package]
```


Pacman – Manjaro / Arch

#Full system sync (resolves db & deps)

```
sudo pacman -Syu
```

#Install missing dependencies

```
sudo pacman -S --needed [package]
```

#Force reinstall to fix broken deps

```
sudo pacman -S [package]
```

#Verify database integrity & missing files

```
pacman -Qk
```

```
sudo pacman -Dk
```

Zypper – OpenSUSE

Refresh and update all

```
sudo zypper refresh
```

```
sudo zypper update
```

Install packages (with deps)

```
sudo zypper install [package]
```

Reinstall to fix files & deps

```
sudo zypper install --force [package]
```

Verify system integrity

```
sudo zypper verify
```

Broken bootloader(or other essential components)? Try:

1. Create a live Linux USB: Ubuntu, Mint, Manjaro (anything)

2. Chroot

lsblk # to find the drive

```
mount /dev/sdXY /mnt          # mount the “ / ” partition
```

(at this point your distro might ask you to reload a service, if yes do it)

```
mkdir -p /mnt/boot
```

```
mount /dev/sdXZ /mnt/boot     # mount EFI or /boot partition
```

```
mount -t proc /proc /mnt/proc
```

```
mount --rbind /sys /mnt/sys
```

```
mount --rbind /dev /mnt/dev
```

```
mount --bind /run /mnt/run
```

```
cp -L /etc/resolv.conf /mnt/etc/
```

```
chroot /mnt /bin/bash          #”bash” is the terminal – if the system is using  
something                     different adapt it
```

3. fix the broken software by: reinstalling/updating (based on distro)

Sound not working?

Check with external interface

If it works while internal sound fails → check hardware

Else → check drivers

Both failed? Check compatibility between drivers and kernel version

“Which audio drivers are the best for media production?” - pipewire

Installing the drivers

APT – Ubuntu / Mint / Debian

```
sudo apt update
```

```
sudo apt install pipewire
```

```
# For client compatibility:
```

```
sudo apt install pipewire-audio-client-libraries
```

```
# Optional extras (Bluetooth, JACK, media-session or wireplumber)
```

```
sudo apt install libspa-0.2-bluetooth libspa-0.2-jack pipewire-pulse wireplumber
```

PACMAN – Arch / Manjaro

```
sudo pacman -S pipewire wireplumber pipewire-pulse pipewire-alsa
```

```
# If installed: remove pulseaudio-bluetooth to avoid conflicts
```

```
sudo pacman -R pulseaudio-bluetooth
```

DNF / YUM – Fedora / CentOS / Rocky

Fedora

```
sudo dnf install pipewire pipewire-alsa pipewire-jack-audio-connection-kit \
pipewire-pulseaudio wireplumber
```

CentOS Stream / Rocky

```
sudo dnf install pipewire pipewire-pulseaudio wireplumber
```

```
# Possible manual removal of pulse:
```

```
sudo dnf remove pulseaudio
```

Zypper - OpenSUSE

```
sudo zypper install pipewire pipewire-pulseaudio pipewire-alsa pipewire-libjack-0_3
```

```
# or include pipewire-aptx for aptX Bluetooth codec
```

```
sudo zypper install pipewire-aptx
```

Reboot the service

```
systemctl --user daemon-reload
```

```
systemctl --user --now disable pulseaudio.service pulseaudio.socket
```

```
systemctl --user --now enable pipewire pipewire-pulse wireplumber
```

Network not working?

1. Establish connection:

If it works: use it to download the firmware for the internal networking card.

If failed: use USB tethering.

```
2. lspci -nnk | grep -iA3 net    #find WiFi card model
   lspci -nn | grep -i network  #find WiFi card model
   lsusb                        #find WiFi card model
   sudo iwconfig                #connection details
```

3. Download the proprietary drivers (if missing)

(update the repositories from the package manager) – can be found at “Missing Dependencies”

```
APT : sudo apt install intel-microcode
APT : sudo apt install firmware-realtek
APT : sudo apt install firmware-atheros
APT : sudo apt install broadcom-sta-dkms
APT : sudo apt install amd64-microcode
```

```
DNF/YUM : sudo dnf install linux-firmware
DNF/YUM : sudo dnf install linux-firmware
DNF/YUM : sudo dnf install broadcom-wl
```

```
PACMAN : sudo pacman -S linux-firmware-intel
PACMAN : sudo pacman -S linux-firmware-realtek
PACMAN : sudo pacman -S linux-firmware-atheros
PACMAN : sudo pacman -S linux-firmware-broadcom
```

```
ZYPPEER : sudo zypper install kernel-firmware-intel
ZYPPEER : sudo zypper install kernel-firmware-atheros
ZYPPEER : sudo zypper install kernel-firmware-realtek
ZYPPEER : sudo zypper install kernel-firmware-brcm
```

4. If the drivers are already installed correctly, change the kernel version – newer kernels are the answer in the most cases

Switching Kernel

Debian / Ubuntu / Mint:

```
sudo apt search linux-image
[select one of the available kernels]
sudo apt install [chosen kernel variant]
```

Arch / Manjaro

```
pacman search linux
sudo pacman -S linux linux-headers
```

Fedora

```
sudo dnf list kernel          # list installed kernels
sudo dnf list kernel --showduplicates # list available outdated/newer kernels
```

`sudo dnf install kernel` `# install latest kernel`

CentOS / Rocky

`sudo yum list kernel*` `# list available kernel packages`
`sudo yum install kernel` `# (or dnf on newer RHEL) install the default kernel`

OpenSUSE

`sudo zypper search kernel` `# list kernel packages`
`sudo zypper install kernel-default` `# install default kernel`
`sudo zypper install kernel-<version>` `# install specific kernel version`

Broken Desktop Environment?

1. **Boot your system to the log screen (either text or GUI)**
2. **Get into tty – Ctrl Alt F3 → (Login afterwards)**
3. **Connect to the internet – USB tethering, Ethernet or WiFi**

4. **Install the Display server (xorg or wayland)**

X11: `sudo [pkgmgr] install xorg xauth xinit` `#pkgmgr – package manager`

Wayland: `sudo [pkgmgr] install wayland`

5. **Install the Desktop Environment**

`sudo [pkgmgr] install [desktop chosen]`

6. **Start the desktop environment**

X11: `sudo nano ~/.xinit` → [specific command for desktop] → save → startx

Wayland: `dbus-run-session gnome-shell --display-server --wayland` (meant for GNOME)

Broken Login App – loop or simply not working?

1. **Get into tty – Ctrl Alt F3 → (Login afterwards)**

2. **Start the desktop environment**

X11: `sudo nano ~/.xinit` → [specific command for desktop] → save → startx

Wayland: `dbus-run-session gnome-shell --display-server --wayland` (meant for GNOME)

3. **Change the Login application**

`sudo systemctl disable gdm`

`sudo systemctl enable lightdm`

AMD Drivers on Linux

AMD covers this subject on their official page:

<https://www.amd.com/en/support/download/linux-drivers.html>

Nvidia Drivers on each distribution:

- **Ubuntu** : <https://documentation.ubuntu.com/server/how-to/graphics/install-nvidia-drivers/>
- **Mint** : <https://linuxcapable.com/install-nvidia-drivers-on-linux-mint/>
- **Debian** : <https://wiki.debian.org/NvidiaGraphicsDrivers>
- **Fedora**
 - <https://rpmfusion.org/Howto/NVIDIA>
 - <https://github.com/oddmario/NVIDIA-Fedora-Driver-Guide>
- **Arch** : <https://wiki.archlinux.org/title/NVIDIA>
- **Manjaro** : https://wiki.manjaro.org/index.php/Configure_Graphics_Cards
- **OpenSUSE** :
 - https://en.opensuse.org/SDB:NVIDIA_drivers
 - <https://gist.github.com/GNY-001F2/71f78e8a99122afa155e49a475d0f6f6>
- **CentOS** : <https://docs.nvidia.com/dgx/dgx-centos-install-guide/introduction-rhel-dgx.html>
- **Rocky** : https://docs.rockylinux.org/desktop/display/installing_nvidia_gpu_drivers/

General – Not distribution tied (in case if the main ways failed)

<https://docs.nvidia.com/datacenter/tesla/driver-installation-guide/index.html>

Unix Nvidia drivers archive : <https://www.nvidia.com/en-us/drivers/unix/>

Installing Unix Nvidia drivers : https://download.nvidia.com/XFree86/Linux-x86_64/304.137/README/installdriver.html

Intel Drivers

Generally, Intel Drivers are included in the Linux kernel, which means that these drivers should work out of the box. In any case here are some general places from where information can be gathered:

- Ubuntu / Mint / Debian / Fedora : <https://www.maketecheasier.com/install-intel-graphics-drivers-linux/>
- Arch / Manjaro : https://wiki.archlinux.org/title/Intel_graphics
- OpenSUSE : <https://software.opensuse.org/package/xf86-video-intel>
- Rocky/CentOS : <https://forums.rockylinux.org/t/rocky-linux-8-7-how-to-install-intel-media-driver/9933>

Step 2.7 : Tweaking a Distribution for Media Production (optional)

Why Debian?

1. Because It has proven in the idling benchmark that it is quite efficient.
2. Easy to work on – good software compatibility.
3. Well known for rock solid stability (used for running so many web serves)

(Definitely keep Xorg around as well for those programs that are still struggling with Wayland)

Can It be recreated on different distributions? - Yes

What will the tweaks be?

1. Low Latency Kernel

- For the standard 6.1 Linux Kernel

https://github.com/samiux/samiux.github.io/blob/master/low_latency.md

2. Larger Swap

There are multiple ways for increasing the swap, but here is one of them

- Turning off the current swap
sudo swapoff -a #Turning off the current swap
- Create a new swap file (e.g., 4 GB) count=4:
sudo dd if=/dev/zero of=/swapfile bs=1G count=4
- Set the correct permissions
sudo chmod 600 /swapfile
- Format the file as swap
sudo mkswap /swapfile
- Enable new swap
sudo swapon /swapfile
- Confirm it's active
swapon -show
- Edit /etc/fstab
sudo nano /etc/fstab
and add:
/swapfile none swap sw 0 0

3. Wayland as Display Server

4. Gnome as Desktop Environment

5. Performance Governor

- `powerprofilesctl set performance`
6. Pipewire comes configured out of the box
 7. Using EXT4 – from the installation

Results after Tweaks

CPU usage: 21% to 24% → 840MHz to 960MHz (Taxed from 4GHz)

RAM usage: 1GB (962MB after disabling Evolution)

Everything is the same compared to the benchmark

Step 3: Finding the software for media work

The chosen Linux Distribution just booted up successfully, congratulations!

Now, It's time to find the proper software for producing media. Fortunately, there are multiple places from which software can be installed:

1. **Distribution specific repositories** (CLI and GUI) – repositories hold pre-compiled packages which can easily be installed and maintained. Each main distribution has its own repositories appended to their package manager. Ex:
 - Debian → APT linked to "<http://deb.debian.org>"
 - Arch → pacman linked to "[https://mirror.rackspace.com/archlinux/\\$repo/os/\\$arch](https://mirror.rackspace.com/archlinux/$repo/os/$arch)"

Can the links to repositories be changed? - Absolutely, new repositories can be installed

find compatible repositories with the package manager → locate the repository list file → edit it → sync the package manager to repository link "`sudo [pkgmgr] update`".

Can a Linux system have multiple package managers? - Yes, yet the extra package manager has to be compatible with the Linux Distro itself.

Is the terminal the only way for installing software? - No, since about 2006 Linux has the concept of **Software Stores**. Each Desktop Environment might(or not) have its own such GUI tool for managing packages. If the Software Store is missing from the Desktop Environment it can be installed afterwards.

2. **Flatpak** – A separate package manager which can run on any Linux Distribution (through a container) and can fetch the latest versions of the packages that are available and also it has its own repository. Therefore, the packages that are installed through flatpak are isolated from the rest of the system, yet still accessible in Terminal and GUI icons.

Official Installation Guide: <https://flatpak.org/setup/>

3. **Packages From Source** – Some software can indeed be shipped in .deb or .rpm formats. these form are pre-compiled packages, which are ready to install. (In most cases coming from proprietary software companies)
4. **Containers** – If it is truly necessary to run specifically a package from a specific distribution, software like DistroBox can help with that by creating containers (virtualized spaces).
5. **Compiling from source** – This is a more extreme solution in case one package has never been ported to Linux before. The general way would be:
Searching for Source → Downloading Sources → Reading building documentation → Installing the required tools → Compiling Accordingly → Installing the newly compiled package.
6. **Unofficially porting Software from Windows to Linux** – This solution is even more particular and it involves using Wine to translate Windows Calls into POSIX calls that Linux can actually understand and execute the program. Yet, there might be a few caveats, such as:
 - Bizarre Errors that not even on Windows are properly documented, let alone in Wine
 - Some functions may not work perfectly (or at all)
 - The Windows program might complain about Linux Drivers and not make proper use of them. (and sometimes it may use instead software rendering)

Video Editing Software available

Editing level	Software	System Support
Beginner	OpenShot	Wide (OS and Hardware)
Medium	ShotCut / FlowBlade	Wide (OS and Hardware)
Advanced	Kdenlive	Wide (OS and Hardware)
Professional	DaVinci Resolve	Narrow (OS and Hardware)
Professional	Lightworks	Wide (OS and Hardware)

1. **OpenShot** – Released back in 2008 by Jonathan Thomas, It's an open source video editing software mostly intended for light editing, with modest features like:

- 3D Text Animations
- Transitions
- Audio Mixing and editing

- Color editing (basic level: gamma, brightness, grayscaleing)
- Native Display Servers support: Xorg and Wayland – work in progress



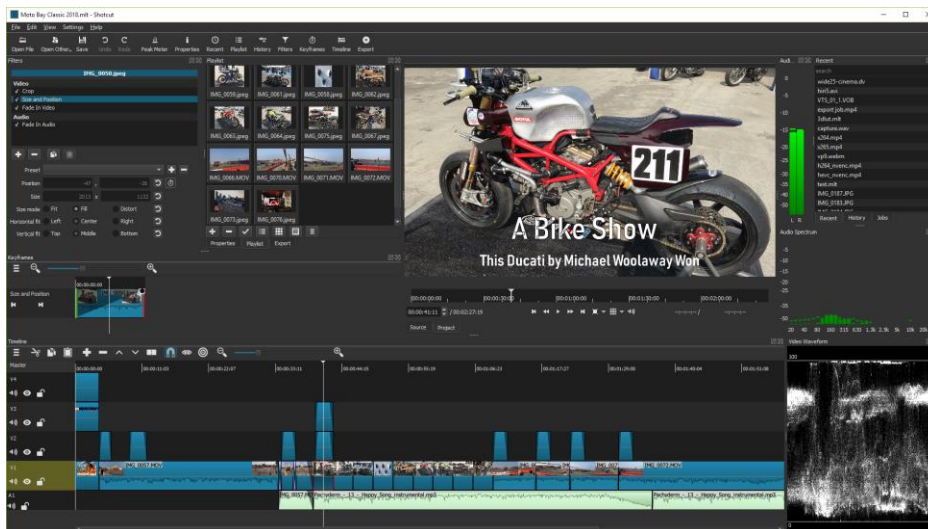
(image source:

<https://www.openshot.org/images/uploads/zinnia/2018/09/23/openshot243.jpg>)

2. **ShotCut** – Released back in 2011 by Dan Dennedy, It's yet another open source video editor but this one packs up some more features compared to OpenShot. Some of these features are:

- Wide image formats support (AVIF, BMP, GIF, JPEG, PNG, SVG)
- Audio Mixing, Cutting, Adding various effects
- Various Video Effects (with GPU support)
- Color Editing
- Low Resolution Proxies
- Batch Import/Export
- Fixed Resolution Proxy
- Native Display Servers support: Xorg and Wayland(from flatpak)

(<https://www.shotcut.org/features/>)



(image source:

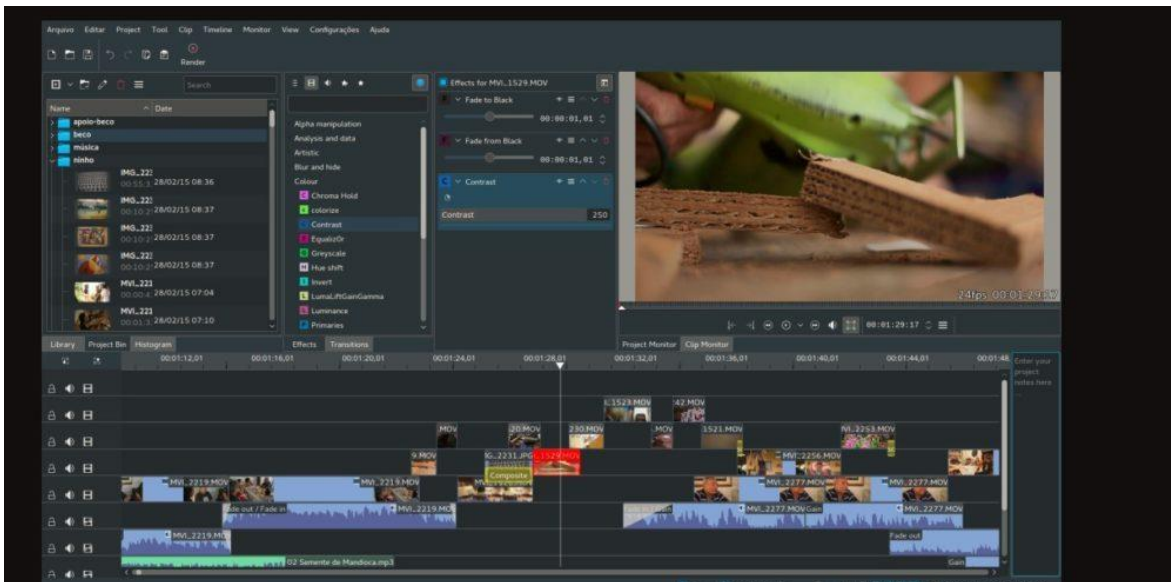
<https://www.shotcut.org/assets/img/screenshots/Shotcut-18.11.18.png>)

3. **Kdenlive** – Started back in 2002 by Official KDE Team, It's another open source video editor, the twist is that it is getting into the **professional category** due to its larger set (compared to the previously mentioned editors) of formats support and more advanced feature. Some of Kdenlive's features are:

- Compatibility with many Video and Audio formats (based on FFmpeg)
- 3 Point Editing
- Drag and Drop support
- Wide range of effects and filters
- SVG vector graphics
- Custom Proxy Resolutions
- Speech-to-Text and Object Tracking (AI Plugins – Using either CPU or Nvidia GPUs)
- Automatic backup
- Project Files
- Native Display Servers support: Xorg and Wayland (from flatpak)

(<https://kdenlive.org/features/>)

In a way, Kdenlive is trying to compete with Adobe Premiere, in some aspect even taking inspiration from it (like the GUI)

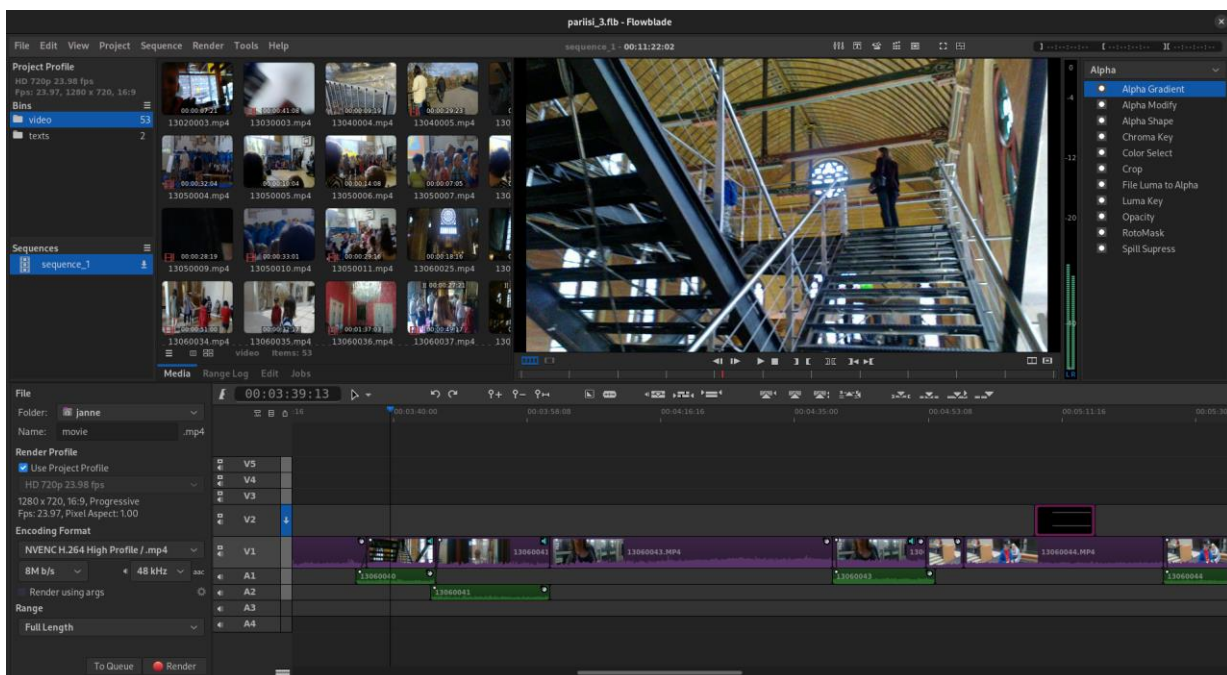


(image source: <https://www.predictiveanalyticstoday.com/wp-content/uploads/2018/06/Kdenlive-1000x493.jpg>)

4. **FlowBlade** – Once again, another open source video editor but focused around hardware optimization. It's not as feature rich as Kdenlive but it stands out with the efficient hardware use. None the less it still packs the essential features like:

- Fully Featured Timeline (Drag and Drop, Insert, Append, Overwrite clips, etc)
- Various Video and Audio filters
- Proxy Editing
- Batch Render
- Audio Mixer
- FFmpeg Rendering
- Native Display Servers support: Xorg

(<https://jilijeb1.github.io/flowblade/features.html>)



(image source: <https://raw.githubusercontent.com/jliljeb/flowblade/master/flowblade-trunk/docs/Screenshot-2-10.png>)

5. **DaVinci Resolve** – This video editor needs no introduction, it's well known as an industry standard in video editing and its usage for Hollywood movies. Its features include:

- Industry Grade color workflow
- Advanced Video Editing tools
- Fusion Visual Effects
- Professional Audio Processing Tools

(https://documents.blackmagicdesign.com/SupportNotes/DaVinci_Resolve_20_New_Features_Guide.pdf)

- Native Display Servers support: Xorg and Wayland

Installation? Doable in 2 ways

1. Recommended: using a supported Linux Distribution
2. DistroBox: Running DaVinci Resolve in a container. (plugins and other functions may be affected)

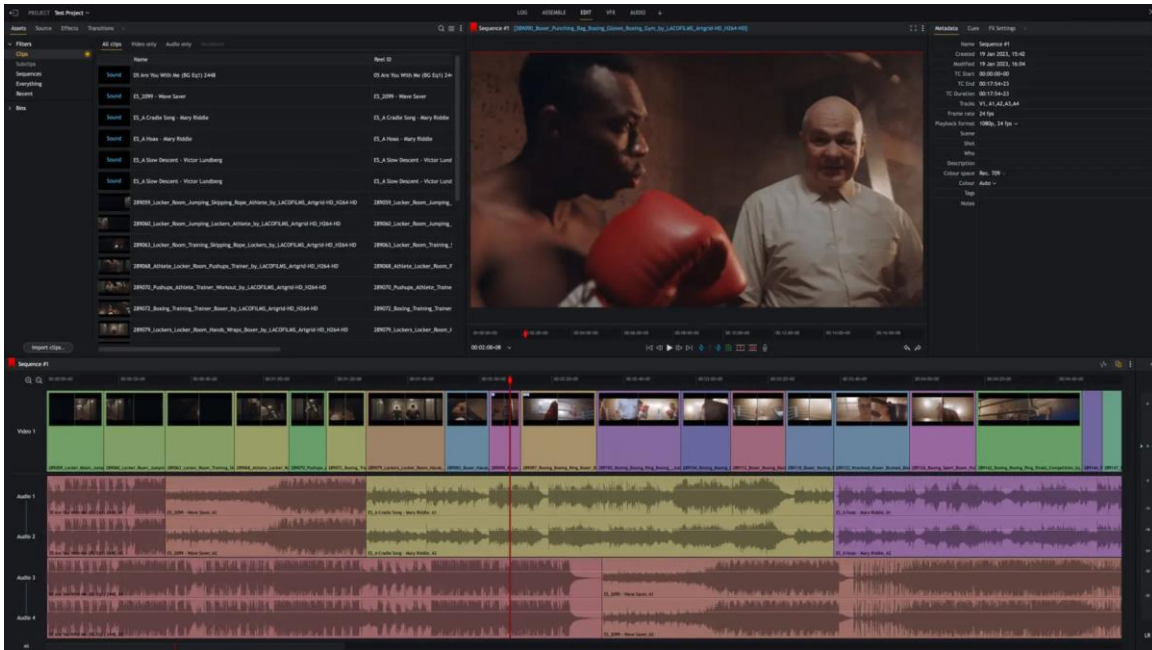


(image source: <https://images.blackmagicdesign.com/images/products/davinciresolve/overview/onesolution/carousel/cut.jpg>)

6. **Lightworks** – Closed Source and non-free video editor (with a free plan as well) aiming to compete directly with Adobe Premiere PRO and DaVinci Resolve Studio. When it comes to features it includes:

- Timeline Editing
- Advanced Audio Tools
- Color Correction

- Native Display Servers support: Xorg and Wayland (Not official, may need X Wayland)



(image source:

<https://lwks.com/hs-fs/hubfs/Screenshot-2023-01-19-at-16.10.27-1920x1080.jpg.jpeg?width=1920&height=1080&name=Screenshot-2023-01-19-at-16.10.27-1920x1080.jpg.jpeg>)

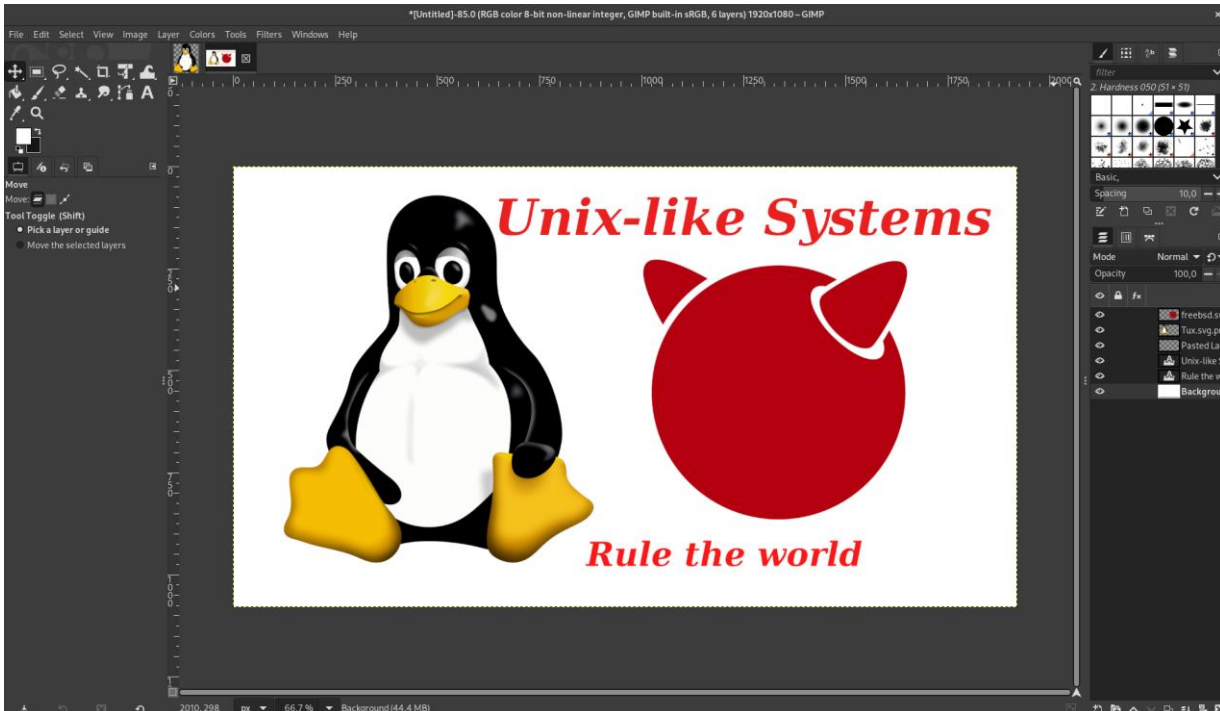
Image Editing Software available

Editing Level	Software	System Support
Image Work – Medium +	GIMP	Wide (OS and Hardware)
Photography - PRO	AfterShot / Darktable	Wide (OS and Hardware)
Illustration – Any	Krita	Wide (OS and Hardware)
Illustration – Pro	Inkscape	Wide (OS and Hardware)
Image Work - Beginner	Pinta	Wide (OS and Hardware)

1. **GIMP** – A well known classic among open source software users. Over the years it managed to implement the right features in order to get into the professional category.

- Full painting suite
- Wide range of Image formats supported
- 3D Transformations for layers
- SVG vector graphics
- Color adjusting
- Library of visual effects
- Non destructive editing

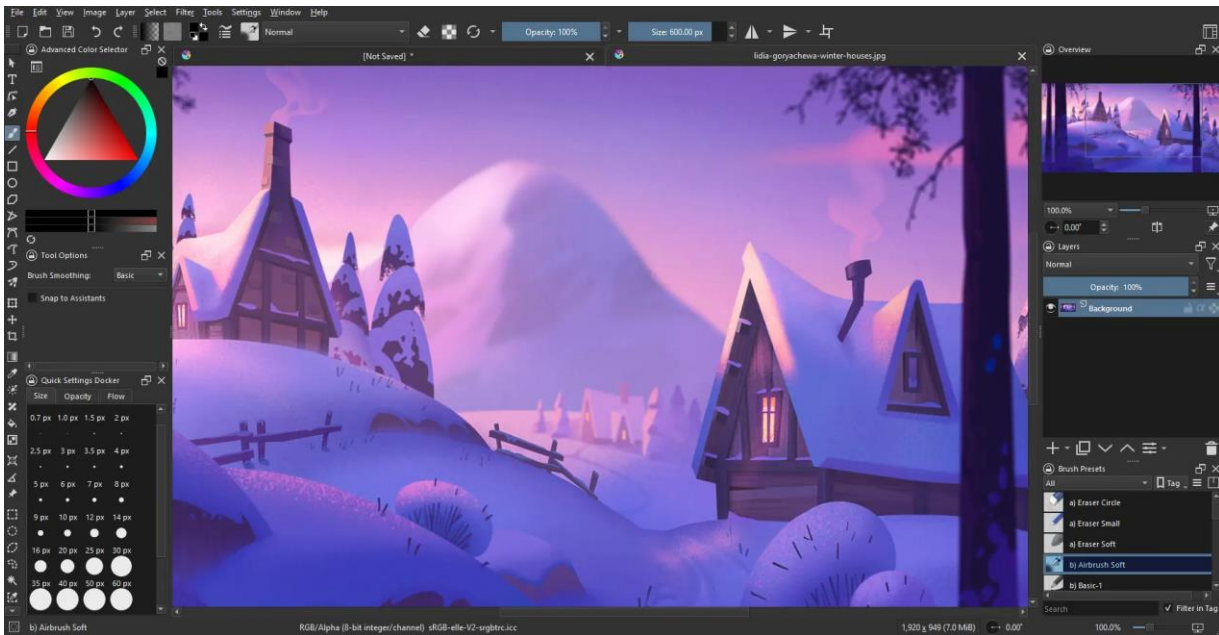
- Native Display Servers support: Xorg and Wayland (<https://www.gimp.org/about/introduction.html>)



2. **Krita** – It is different from GIMP. While GIMP is an image editor, Krita is a professional painting tool, used even for designing famous illustrations. **Also it can be used for 2D animations.** Some of its features include:

- Drawing Assistant
- Full color management
- HDR support
- PSD support – File format compatible with Photoshop
- GPU support
- Native Display Servers support: Xorg and Wayland (can start with Wayland X)

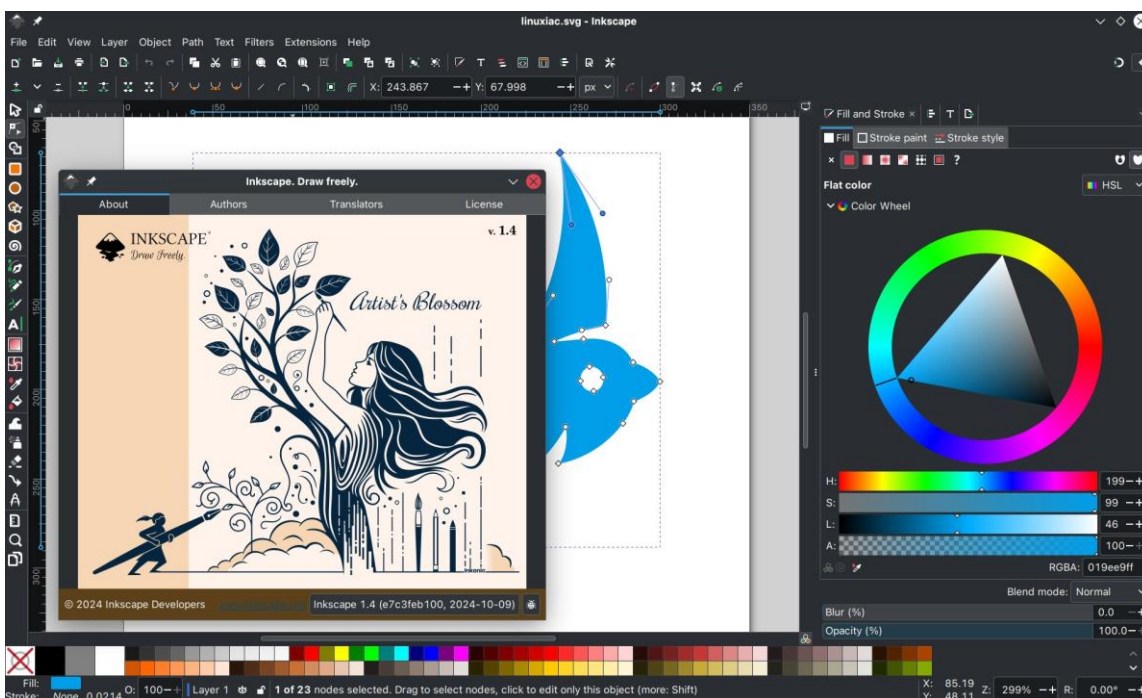
(<https://krita.org/en/features/>)



(Image source: <https://krita.org/images/pages/application-screenshot.webp>)

3. **Inkscape** – A great open source tool for creating Scalable Vectors Graphics natively.
(<https://inkscape.org/>)

- Native Display Servers support: Xorg

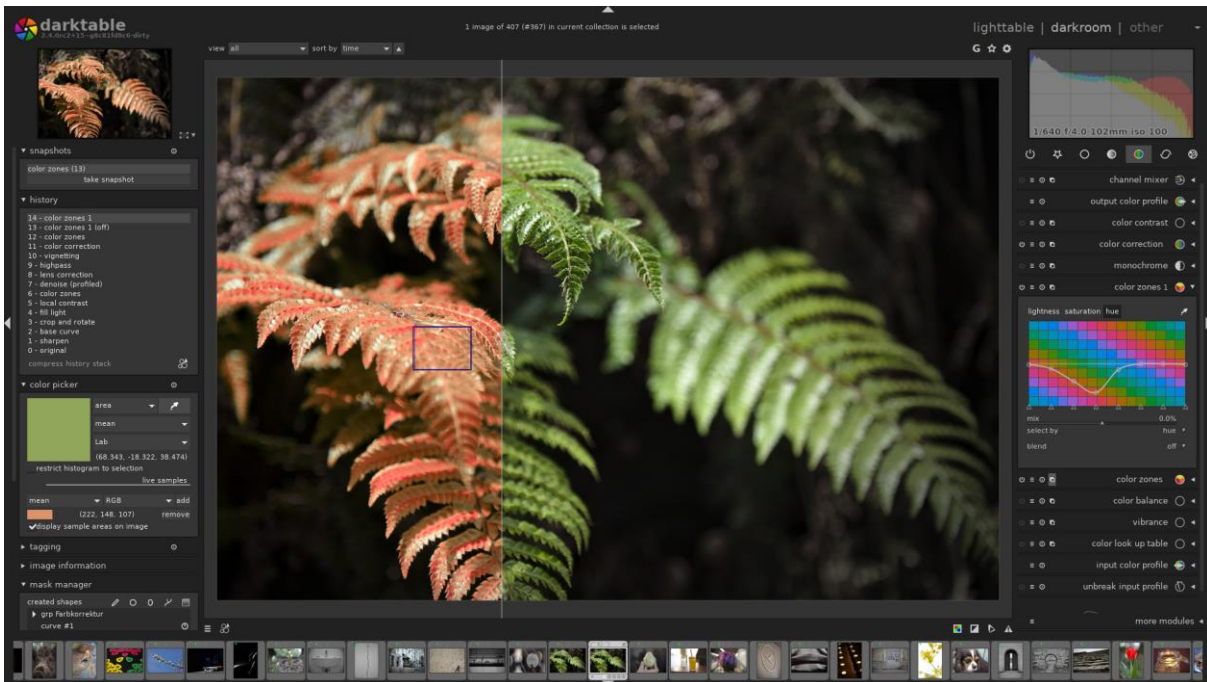


(image source: <https://linuxiac.b-cdn.net/wp-content/uploads/2024/10/inkscape14-app.jpg>)

4. **Darktable** – Yet another Open Source image editor, except it is more targeted on the community photography. Some of its features include:

- Non Destructive editing
- Hardware Acceleration
- GPU support
- Wide image formats support

- Native Display Servers support: Xorg and Wayland – work in progress
(<https://www.darktable.org/about/features/>)



(Image source: https://www.darktable.org/images/screenshot_darkroom1.jpg)

5. **AfterShot** – A proprietary photo editor, aiming to a direct competitor for Lightroom (including even compatibility for projects (Not a free photo editor))

- Native Display Servers support: Xorg and Wayland – ongoing issues
(<https://www.aftershotpro.com/en/>)

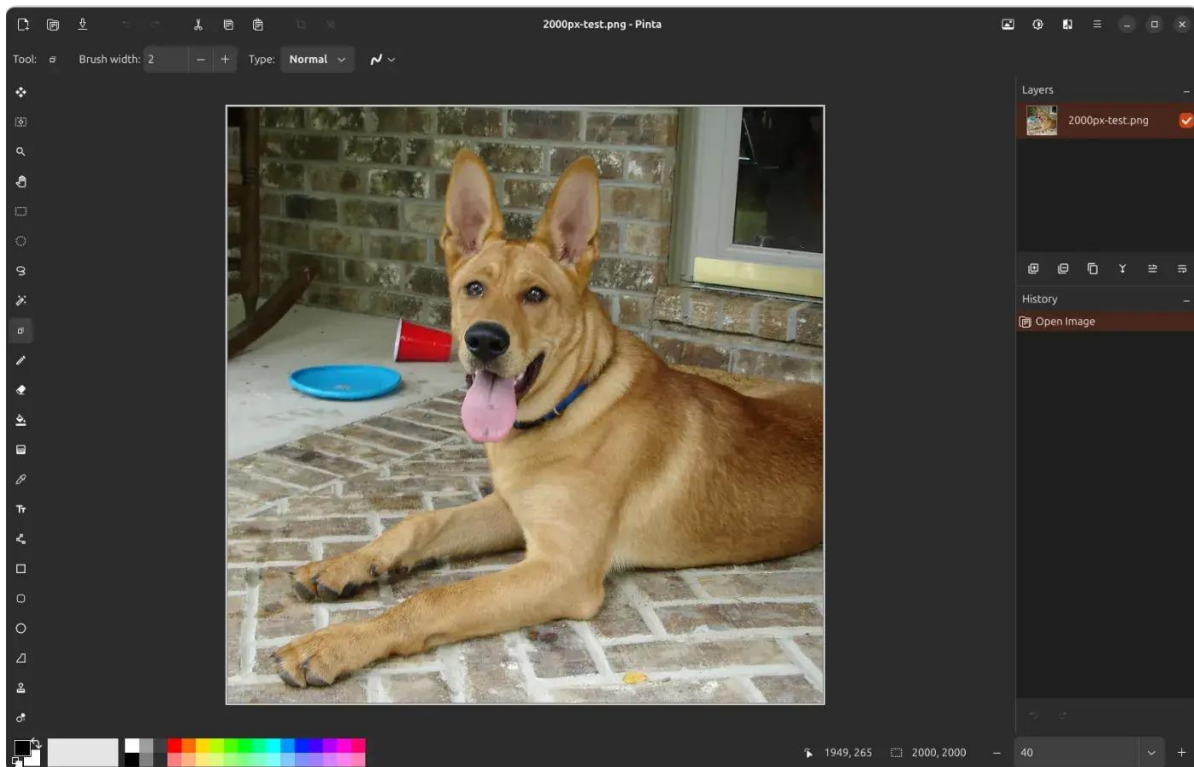


(Image source: <https://www.aftershotpro.com/static/asp/images/products/aftershot-pro/v3/screens/multi-version-editing.jpg>)

6. **Pinta** – An open source image editor, mainly inspired by Microsoft Paint it aims for a user friendly interface and keeping the essential tools:

- Drawing tools – basic
- Layers support
- Image adjustment
- Native Display Servers support: Xorg and Wayland

(<https://www.pinta-project.com/>)



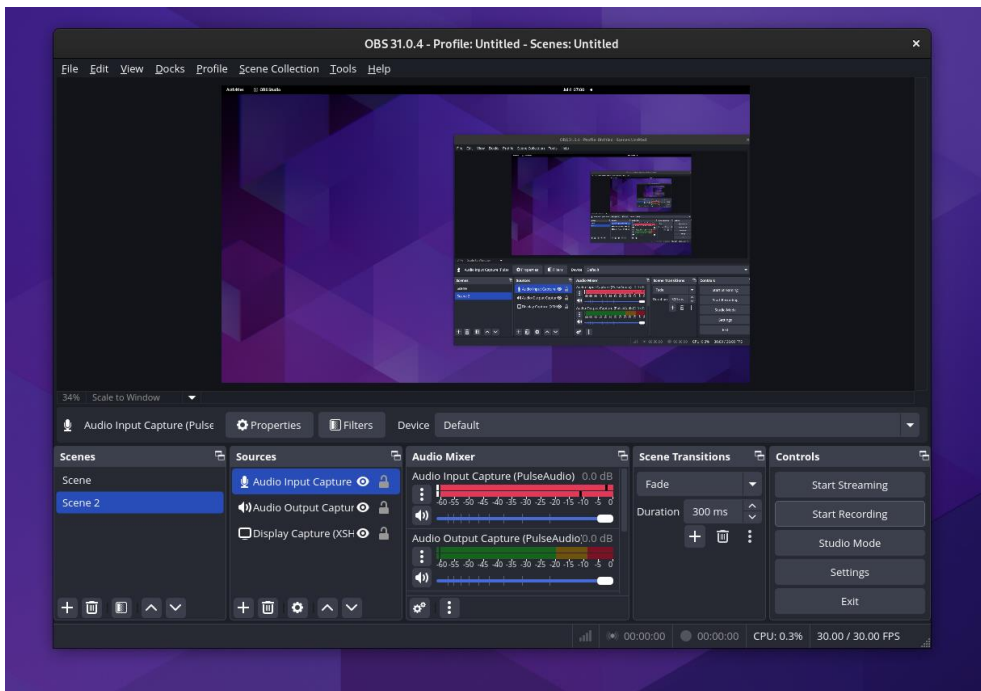
(image source: https://flathub.org/_next/image?url=https%3A%2F%2Fdl.flathub.org%2Fmedia%2Fcom%2Fgithub%2FPintaProject.Pinta%2F9b84c9ecaa60af4c8d1e0c2012f8e7f7%2Fscreenshots%2Fimage-1_orig.webp&w=3840&q=75)

Streaming and Recording Software available

OBS – A free and open source screen recorder that can be used for streaming

- Support for Pipewire (audio)
- Support for X11 (display server)
- Support for Wayland (display server)
- Native Display Servers support: Xorg and Wayland (needs configuring)

(<https://obsproject.com/>)

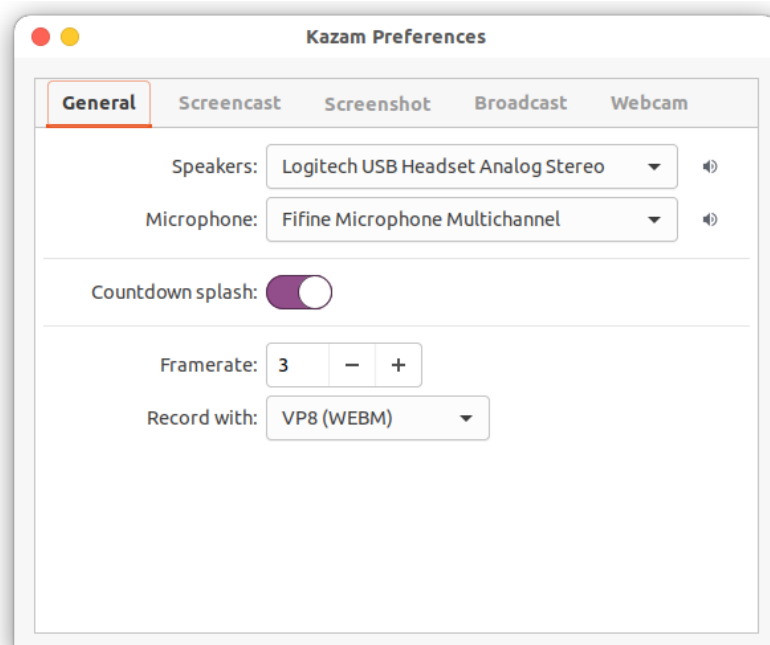


Screen Recorders Software available

1. Kazam 2 – Open Source screen recorder

- PulseAudio support
- Native Display Servers support: Xorg

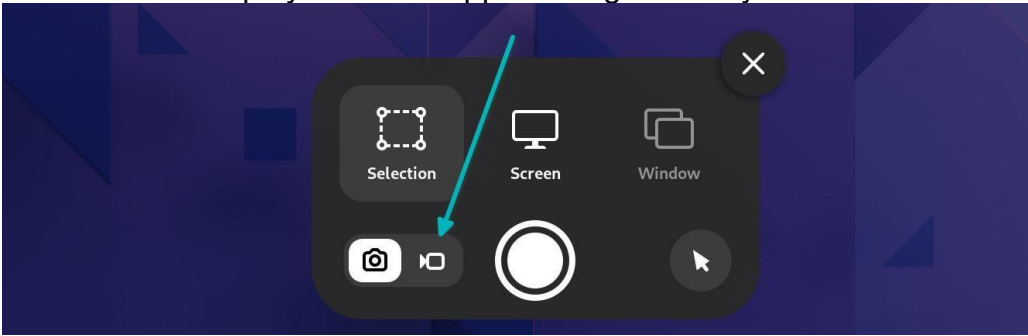
(<https://github.com/henrywoo/kazam>)



(image source: <https://raw.githubusercontent.com/henrywoo/images/main/prefs.png>)

2. Gnome built in – Open source screen recorder

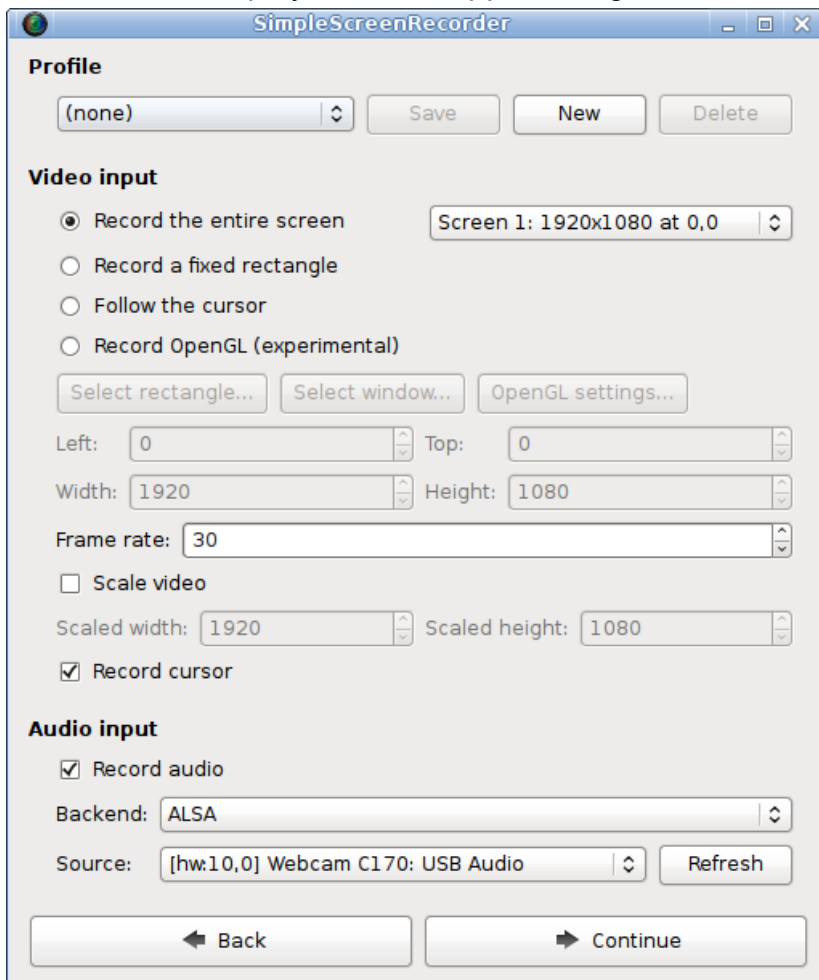
- No native sound
- Native Display Servers support: Xorg and Wayland



(image source: <https://itsfoss.com/content/images/2023/03/gnome-screen-recorder.jpg>)

3. SimpleScreenRecorder – Open source screen recorder

- Jack Audio proved to work -
(<https://unix.stackexchange.com/questions/695905/cannot-record-microphone-with-simplescreenrecorder>)
- Native Display Servers support: Xorg



(image source: <https://files.maartenbaert.be/simplescreenrecorder/screenshot.png>)

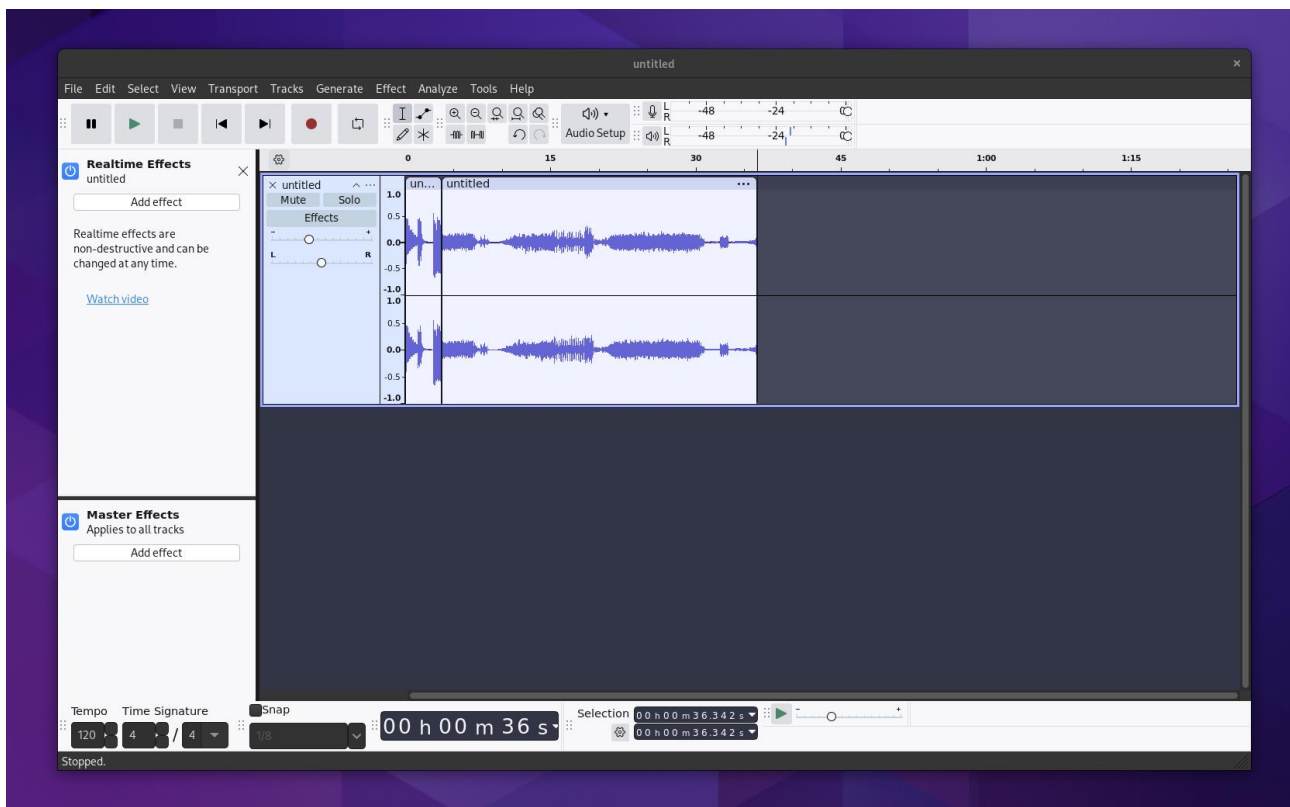
So, for screen recording which software is better – in the sense of features? OBS, obviously.

Sound Editing Software available

Audacity – An open source Sound Recorder and Editor at once. Audacity is also known as an industry standard for sound effects makers and editors.

- Sound Recording
- Wide Audio Format support : MP3, WAV, FLAC, etc
- Many Audio Settings available
- Many Audio Effects available
- Native Display Servers support: Xorg

(https://manual.audacityteam.org/man/index_of_effects_generators_and_analyzers.html)



3D Modeling Software and Animation

Blender – A suite of tools for developing and using 3D models. This software is developed under an open source license.

Features:

- Modern Rendering Engine (Cycles and EEVEE Next)
- Advanced Shading & Materials

- Geometry Nodes & Procedural Modeling
- Animation & Rigging
- Physics & Simulation
- Display Servers support: Xorg and Wayland



(image source: https://www.blender.org/wp-content/uploads/2019/07/blender_render-1280x720.jpg?x12104)