

How to RICE Linux

Nicola Bruno

December 23, 2025



Contents

Contents	2
1 Introduction	4
1.1 What is this book	4
1.2 Who is this book aimed at	4
1.3 What is inside this book?	4
2 Let's get started	5
2.1 Before we begin	5
The Operating System	6
The display server	7
The window manager	8
The bar	8
The application launcher	9
The terminal emulator	9
The shell	9
The file manager	10
The notification deamon	10
The compositor	11
The wallpaper	11
The color scheme	12
The fonts	12
The audio system	12
The music player	13
System monitor	14
The lockscreen	14
The power menu	14
Screenshots	15
Clipboard manager	15
Configuration Files	15
Dotfile Management	15
Miscellaneous cool stuff	15
Browser	16
Application Theming	16
Scripts and Automation	16
Login Manager	16
2.3 Going Forward	16

<i>CONTENTS</i>	3
-----------------	---

3 Base Configuration	17
3.1 The Operating System	17
Debian	17
Arch	17
Fedora	18
3.2 Display Server	18
X11	18
Wayland	18
3.3 Window Manager	19
X11 Compatible	19
Wayland Compatible	19
3.4 Status bar	20
X11 Compatible	20
Wayland Compatible	20
Compatible with both	21
3.5 Application launcher	22
X11 Compatible	22
Wayland Compatible	22
3.6 Terminal emulator	22
3.7 Shell	23

Chapter 3

Base Configuration

After listing every possible component, it's time to make a working configuration for each one. Here there will be basic configurations for a working setup, without styling.

3.1 The Operating System

I will talk about the 3 main categories of OS I mentioned before, so Debian, Arch, and Fedora



I do not like mixing Operating Systems with Desktop Environment that they do not ship with, so for premade DEs, just use whatever is suggested by the distro

Debian

Since Debian does not get the latest packages, if you use something like KDE or GNOME, you run the risk of having issues with packages. I recommend using MATE or Xfce to keep it stable.

For Stacking Window Managers, go with *i3wm* if you want X11, or *sway* if you want Wayland.

Arch

I believe that if you're using something Arch based, like EndeavourOS or Manjaro, you should get the KDE Plasma version for a premade DE.

Instead if you want a Window Manager go with Sway or Hyprland

Fedora

Fedora already ships with KDE Plasma, so use that.

For SWMs use Sway or i3, since they both are in the official repo

¹

3.2 Display Server

Choosing your display server dictates most of the choices going forward, so we need to get the right one.



X11

X11 has been the Linux standard for more than 30 years, and it has proven itself more than once during this time. It has excellent compatibility with older applications and games as well. It allows running graphical apps remotely and it has a vast documentation and extensive troubleshooting resources. It has screen sharing and window recording that just works and is reliable on basically every system. It also has better support for specialized hardware and specific use cases.

However, since it's so old, it has an outdated architecture and has a few security vulnerabilities. It has poor multi-monitor support, especially on more modern displays with mixed DPI and various refresh rates. It's inconsistent with touchpads gestures on laptops and has very obvious screen tearing issues without a compositor. But most importantly it's not developed anymore, it's just maintained, so don't expect new features or reworks of the old architecture.

Wayland

Wayland on the other hand, has a modern security model with application isolation, it has smooth animations and better performance, having way less

¹Remember that every Linux distro is modular, so if you want to mix and match pieces, feel free to do so. It's not going to change how to customize everything, so you can still follow this guide or the official documentation.

latency than its predecessor. It has excellent multi monitor compatibility and support per-monitor scaling. It has native gestures and touch input, it's designed to not have screen tearing issues, and most importantly it's actively in development and it's constantly being improved.

However, some applications, especially older ones, still have compatibility issues, screen sharing can be inconsistent, there is no native network transparency, some features are still being implemented and different compositors yield different results that not always correspond to the desired output.

3.3 Window Manager

Now that we've chosen our display server, it's time to choose our window manager. Full desktop environments already have their own, so I will not be mentioning stuff like KWin or Mutter.



Figure 3.3: This is a stacking window manager, in this case KDE Plasma

X11 Compatible

The WMs compatible with X11 are:

- i3
- awesome

And many more, but listing everything can get overwhelming, so I will be sticking with the most well known ones

Wayland Compatible

Instead, the WMs compatible with Wayland are:

- Sway
- Hyprland

- Niri

And a whole lot more. If you have a need so specific that these do not sound appealing to you, you have the necessary knowledge to research them on your own

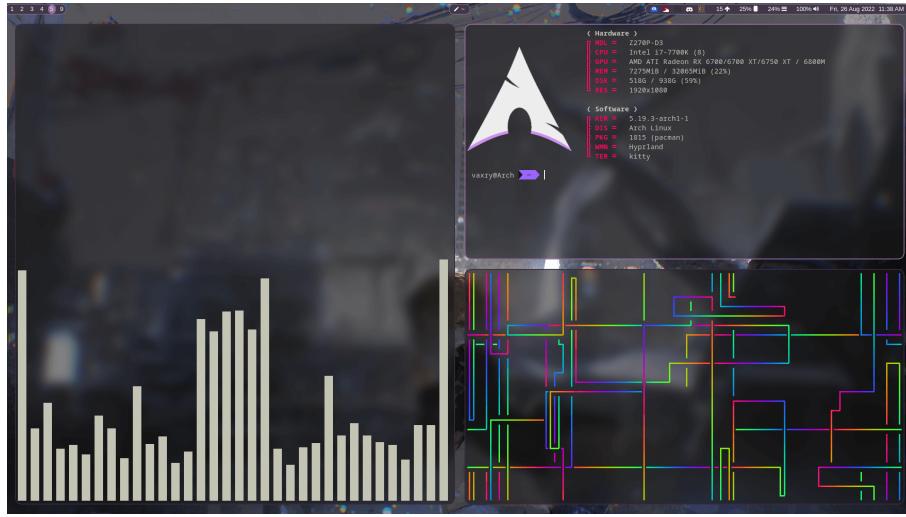


Figure 3.4: This is a tiling window manager, in this case Hyprland

3.4 Status bar

For the status bar, the usecases shift from usability to style for the most part. You will not be clicking on the bar very often, and when you do, you don't need a whole lot of specific behavior. Of course if you do you can, but I don't think having a super complicated bar fits the average user's need.

X11 Compatible

The status bars compatible with X11 are:

- Polybar - Highly customizable but resource intensive. May be complex to configure
- i3bar - Very lightweight and fast but with limited built-in modules. Pretty basic appearance
- lemonbar - Extremely lightweight, no built in modules. Requires scripting knowledge to fully configure

Wayland Compatible

The status bars compatible with Wayland are:

- Waybar - Highly customizable through CSS styling, excellent module ecosystem, great documentation. May use more resources than alternatives

- **yambar** - Lightweight and efficient, with less intuitive configuration. Smaller community.
- **swaybar** - Native to sway, very lightweight. Pretty basic appearance

Compatible with both

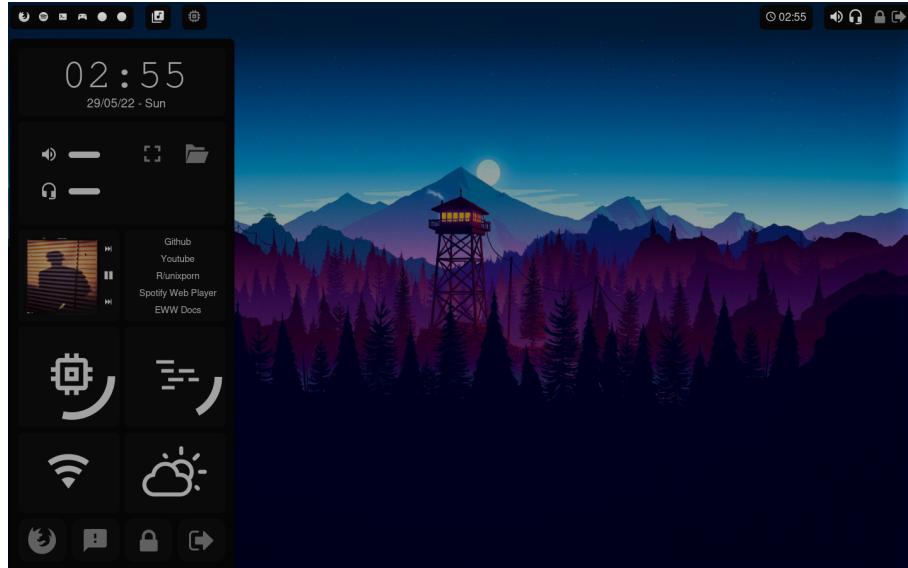


Figure 3.5: This is an example configuration using eww

There are a couple of bars that are compatible with both display servers:

- **i3status-rust** - Blazing fast and efficient, good selection of built in blocks. Configuration in TOML may feel incredibly boring. Recommended for efficiency while not compromising a lot of features
- **eww** - Elkowar's Wacky Widgets is not just a bar, but a full widget system. It's ridiculously customizable and it even supports animations. It has a fairly steep learning curve, and may become resource intensive. Completely overkill for a simple status bar

3.5 Application launcher

Application launchers are, for the most parts, similar to bars. With the difference that you will be launching applications a lot, so choose wisely if you want to prioritize looks or performance

X11 Compatible

The application launchers compatible with X11 are:

- **Rofi** - The most popular and feature rich, with extensive documentation and theming. Can be complex to deeply configure, requires custom theme syntax

- dmenu - Minimal and efficient, highly scriptable, but with a basic appearance

Wayland Compatible

The application launchers compatible with Wayland are:

- Rofi-wayland - Rofi's fork for Wayland. All of the same functionalities of the X11 version, but may be lagging behind with updates
- Wofi - Rofi-inspired application launcher for wayland. Similar configuration to Rofi, with CSS theming. May have some bugs
- dmenu-wayland - Direct dmenu port for wayland. Same as dmenu for X11

3.6 Terminal emulator

Every true Linux user needs his trusty terminal emulator. These are not tied to the display server, so we have a bit more flexibility

- Alacritty - GPU accelerated, very fast, minimalist design with YAML configuration
- Kitty - Also GPU accelerated, built in tabs and splits. Extensive features with ligature support. Python based config. Not the easiest configuration
- Konsole - KDE's default. Feature rich with profile support. Heavily dependant on KDE libraries
- GNOME terminal - GNOME's default. Simple and reliable, profile and theme support. Basic features, heavily dependent on GTK
- Ghostty - Very new GPU accelerated terminal with excellent performance. Native on each platform, sporting fast startup and low latency. Basically a mix of Alacritty and Kitty. Very new, so relatively unstable, with a small community and low documentation. Not much customization yet

3.7 Shell

This is very niche to customize, since Bash is already powerful and simple enough to do everything.

- Bash - Default on every Linux distro ever. Most compatibility
- Zsh - Apple's shell. I see no reason for using it
- Fish - Sports syntax highlighting and auto suggestion from history. No POSIX