How to RICE Linux

Nicola Bruno

December 22, 2025

Contents

Chapter 1

Introduction

Who the fuck am I? Why are you here? And most importantly, should you stay?

1.1 What is this book

This book is something that I wished to find when I started my Linux journey. I have my glorious king Brian Ward to help me with the backend stuff, but what about the frontend? I had to learn all of this stuff on my own, and I feel like I'm not even 1% there. I hope this book is useful to somebody.

1.2 Who is this book aimed at

If you do not know anything about a Linux system, what the hell are you doing here? Go read ”How Linux Works by Brian Ward” immediatly. Or some other documentation until you reach a basic understanding of the system, then come back here. Instead, if you are an experienced Linux user, enough to want to correct me every time you see ”Linux” and not ”GNU/Linux”, then again, what the hell are you doing here? This is a book for unexperienced people, take your experience elsewhere, like my email inbox, so that you can share your knowledge and improve this book written by this complete idiot.

1.3 What is inside this book?

Inside this book you will find:

- What does it mean to customize your system
- How to customize your system
- What every piece is and what does it do
- How I archieved my *super cool RICE*
- A whole lot of yapping on unrelated topics (this may be removed in the future)

Chapter 2

Let's get started

Here I will illustrate the basics of a RICE. What it means and what are the various components, as well as what they do and why they are there

2.1 Before we begin

I consider part of the customization not just the aesthetic, but also the functionality of the system, so here there will be more than just theme customization, but some light scripting as well.

2.2 What makes a system

You need a lot of parts to make the system actually work. I will be assuming that you have the vanilla Linux Kernel with default parameters, systemd as your init, and GRUB as your bootloader. After that, you will need :

- An Operating System
- A Display Server
- A Window Manager or a Desktop Enviroment
- A panel or a bar
- An application launcher
- A terminal emulator
- A shell
- A file manager
- A notification Daemon
- A compositor, or something to display visual effects
- A wallpaper
- A color scheme
- Some fonts

- An audio system
- A music player
- A system monitor
- A lock screen
- A Power menu
- Screenshots capabilities
- Clipboard manager
- A text editor
- Dotfile management
- Miscellaneous cool stuff
- A web browser
- Themes for every application
- Automation for basic system functionalities
- Session management

Remember that all that you need to use a Linux based OS is electricity, and even that is optional, so if you feel like any of these components are not needed, feel free to non include them in your RICE. With all of that said, let's actually start on something.

The Operating System

Whatever distro you chose to customize, it all boils down to the package manager of choice. As such I will be making every step with Arch Linux in mind, but it's easy to check if the thing I'm talking about is compatible with your distro, and if it is, install it with your package manager(or build from source), and follow the guide like nothing happened.

The OS is made by 8 main components:

- Kernel - This is the black magic that translates whatever you do to the hardware into a language that your computer can understand, so it can follow your commands. Without the kernel, the pc is just a bunch of expensive metal.
- Init System - The init ”wakes up” every program on your system and keeps them running.
- Display Server - This guy makes visual output possible, making the connection between the running programs and your display
- Shell - This bad boy interprets commands and tells them to the computer, allowing you to do stuff. Even if you never use the terminal, it's always running

- File System - This is the structure of your storage drive. Without it, you cannot save of load anything
- System Libraries - These are the computer's tools. They are shared pieces of code that every program can call and use to do their jobs.
- Basic System Utilities - These are your tools. You use them to interact with your files and with the system

If you have yet to choose a distro, do not be afraid, for I shall give you guidance. It's actually really easy. If you care about stability above all, go with something Debian based, like Ubuntu or Mint. If you want maximum customization capabilities and bleeding edge updates, go with Arch or any derivative distro. If you still want stability but you do not want to get packages 4 years later than everybody else, go with Fedora. I say Fedora and not any other RedHat distro because I do not enjoy the RedHat logo.

Usually, everything in the list above comes preinstalled in your distro of choice. To customize everything down to the last bit you can either find a distro that has everything you want preinstalled, install Arch Linux manually and choose what you want, or install Gentoo and compile everything from source for hyperoptimization. You can always install some components on top of other components, even if it's not recommended. It's always a good idea to have a reliable safety net that you know works, so that ig you break something, you can use your default build to fix your WIP RICE. To keep it simple, we will focus on choosing the display server, the shell, and everything after that. Operating Systems are a complicated thing and the less you touch them the safer it is, until you know what you're actually doing.

The display server

The most common display servers are:

- Wayland - The rising star of the new age of Linux Desktop. It's the default DS on Fedora, Ubuntu, and DEs like GNOME, and even KDE Plasma has a Wayland version. It offers better security, performance and supports more modern features like HDR and mixed DPI displays.
- X11 - The traditional Linux display server, that was the standard for decades. It's still widely used and it's a reliable option to fall back on if Wayland breaks. It's getting replaced by Wayland for the aforementioned reasons, but it's still a great option and it also offers the most compatibility
- WDDM - The proprietary Windows DS. It has been built into Windows since Vista and it handles every graphic operation in Windows
- Quartz Compositor - Apple's proprietary DS, it's built into macOS and it does the same job as every display server ever.

Our choices are between Wayland and X11, since we are using Linux. I personally use Wayland, and that's what I recommend.

The window manager

This is the computer's interior designer. It decides where the windows appear on screen, how they look, what happens when you drag, resize or close them, and if the windows stack automatically or not. If you choose a ready Desktop Enviroment, like GNOME or KDE Plasma, this will come with a whole lot of other stuff needed for the system's functionalities. However, if you choose a *tiling window manager*, you will have only that, and you will have to install everyhing else manually. The most widespread are:

- Desktop Window Manager(DWM) - Windows' built in WM
- Quartz - MacOS built in WM
- Mutter - GNOME's built in WM
- KWin - KDE Plasma's built in WM

But the ones we are actually interested in are the ***Tiling Window Managers***, that have a specific behaviour built into them. These WMs automatically align the windows you open on a grid, and allocate the same space dynamically to every window. The most common Tiling WMs are:

- i3 - the most widespread Stacking WM. It's the classic and it uses X11, with no plans to move to Wayland. It's very reliable and the developers say they will be maintaining it as long as there will be users. It's really fast and lightweight, since it's written in C
- awesome - awesome is a heavily extensible WM written in C and Lua. Its selling points are a fast and small codebase, a really well documented source code, large implementation capability, no floating windows and more
- Sway - it's similar to i3, but it's wayland compatible, and most of the stuff written for i3 are also compatible with sway
- Hyprland - One of the newest WMs out there. It's modern, well documented, good looking, and wayland compatible. It also contains its own compositor and has a suite of tools tailored specifically for Hyprland to have every functionality a PC needs
- Niri - A scrollable WM for Wayland. It arranges windows on a endless strip going to the right, and it never resizes an existing window, since it just adds it to the strip

I personally use Hyprland, since that fit my use case better than everything else, but every option is valid here. Just remember that if you do not use Hyprland, you will have to install a compositor manually

The bar

It's not hard to understand why you need a bar on your desktop. It holds ready to see information, like the clock or what workspace you're on. It can also be configured to hold whatever you need. For example, I have a special button to play the first 3 seconds of the opening to *Orb - On the movement of the Earth*, bass boosted and distorted. Every premade DE will have its bar, while WMs may have one, but it does not always fit our usecase. Here are the most common ones:

- i3/swaybar - Respectively, i3's and sway's default status bar
- Waybar - A 3rd party bar compatible with basically every WM out there. Highly customizable with JSONC for configuration and GTK-CSS for styling
- Polybar - Similar to Waybar, but X11 compatible

The application launcher

Again, this in probably included in your DE of choice if you downloaded one. If you have a SWM, you can either use:

- rofi - The most versatile launcher on this list. As of 2025 it has finally gotten Wayland support
- wofi - rofi fork for wayland
- dmenu - The basic launcher that inspired every other. Fast, lightweight and customizable
- KRunner - KDE Plasma's build in launcher

The terminal emulator

This is the part where you can feel like a *supaa hackaa*, by typing commands in text form instead of clicking buttons with your mouse. There is a terminal emulator for every taste out there, and they're very similar. Here are the outstanding ones:

- Alacritty - GPU accelerated terminal emulator written in rust
- Kitty - GPU accelerated, true color terminal emulator. Supports plugins for exended functionality
- GNOME Terminal/Konsole - Default terminal on GNOME and KDE Plasma. Enough to get the job done, not very customizable

The shell

This is the language of the terminal, and also the translator. It's basically a programming language, and can be used both to communicate with the terminal, and to write scripts for automated behavior, since a shell script is basically a list of commands. Here are the most used shells:

- Bash - Bourne Again SHell, it is the most widely used shell on Unix like systems
- Zsh - Z Shell, it's Apple's shell. Basically bash with some stuff added to it

- Fish - Friendly Interactive SHell. Modern shell with syntax highlightning and autosuggestions. But no POSIX though.
- Dash - Minimal POSIX-compliant shell. Very fast for scripts, but it was better in the past, since bash was not optimal
- PowerShell - Microsoft's shell for Windows. It is abysmal

Bash will forever be the best for general purpose, since is the most portable one and it's the default everywhere.

The file manager

This is the file cabinet of the computer. It allows you to navigate files, open them, delete them, create more, and so on. We can just steal them from ready made DEs:

- Nautilus - GNOME Files, clean and simple interface
- Dolphin - KDE's default, highly customizable
- Thunar - XFCE's default, very lightweight
- Nemo - Cinnamon's default, a fork of Nautilus

There are some more, but I do not have experience on them, so if there do not seem good enoughto you, you're free and encouraged to dig around for more File Managers

The notification deamon

This sends you notifications when stuff happens, like when your music player skips a song, or if your battery is dying. There are a few options we can pick from:

- Dunst - Very popular notification daemon, due to its lightweight nature. Text based configuration, low resource usage, and icon support make it a great option
- Mako - The go-to choice for Wayland compositors, especially on Sway
- notification-daemon - The original GNOME notification daemon. Once the standard, now widely replaced by GNOME's built in GNOME Shell notification system
- KDE Plasma notification system - Very self explainatory, it's KDE's build in ND
- deadd-notification-center - A very popular notification daemon among the users who crave functionality above all

The compositor

Now we get into prime RICE territory, this is the program responsible for all of the cool effects on your desktop, like shadows, blur, transparency, and animations. Some WMs already have their own, and all ready made DEs have a built in one, but not every WM has it, so:

- wlroot-based compositors like:
  - Sway - i3 compatible tiling compositor
  - Hyprland - dynamic tiling *with the looks*
  - river - dynamic tiling, very lightweight
  - Wayfire - 3D effects and animations
- Picom - Standalone compositor for X11, supports transparency, blur, shadows, fading, and is easy to customize
- Xcompmgr - Older, simpler X11 compositor

The wallpaper

The wallpaper is basically just an image that's in an untouchable layer behind everything else, and it needs to be loaded and rendered on the screen. WMs do not have this functionality by default, so we need a Wallpaper manager:

- feh - lightweight command line image viewer that can set wallpaper. X11 compatible
- nitrogen - GUI wallpaper setter with multi monitor support. X11 compatible
- swaybg - Wayland wallpaper tool for wlroots compositors
- hyprpaper - Hyprland wallpaper daemon
- xwallpaper - minimal X11 wallpaper setter

  These are static wallpaper setters, you can als have dynamic ones:

- Wallpaper Engine - popular on Steam, has extensive community and basically endless wallpapers to choose from. It can work on Linux, but it's not officially supported, so I do not recommend it
- Komorebi - animated wallpaper for Linux
- mpvpaper - uses mpv to play videos as wallpapers. Wayland compatible
- xwinwrap - wrapper to set animated wallpapers on X11
- Hydrapaper - multi monitor wallpaper manager with a GUI

The color scheme

This is important if you want to have a cohesive theme all around your pc. You can go for a monochrome setup, or a really discordant palette with crazy colors and bright accents. For this we will need some color theory, but first of all, let's pick a color. You do not have to do this, since there are a million ways to choose a color scheme, but this is a simple approach to it. After picking a color, we select either analogous colors, or complementary colors. You can also go the triadic way, by picking equispaced color on the color wheel, or go monochrome and alter just the saturation and lightness. My suggestion is to pick the accent color(s), and the rest of the UI will be neutral colors, so blacks, grays, and whites. If you care about your eyes while working on your PC, you should limit your palette to a maximum of 2 main colors, and 2 accent colors. And obvious colors for everything that's not supposed to sit on the screed. i.e. red for errors, yellow for warnings, green for success. There are many websites that allow to build your palette and view both the HEX values and the rgba values, as well as every color scheme under the sun

The fonts

To have a cohesive setup, you need custom fonts. Of course you can have the same font for everything, but that can get confusing to read. I suggest having:

- a Serif or Sans Serif font for the UI itself
- another Serif as a backup and for stuff like window titles and other things that may need differentiation
- a monospace font for the terminal and your text editor of choice

All of this can be eiter found on NerdFonts or just googling ”cool fonts”

The audio system

This controls everything that relates to audio. Volume levels, which program makes sound, where is outputted, where is inputted and so on There are two layers to the audio system: the **driver** and the **sound server**: Drivers:

- ALSA (Advanced Linux Sound Architecture) - Kernel level audio system. Always present on modern Linux systems. It has some limitations, like accessing only one sound card at a time, but it's stable and there are workarounds to its limits
- OSS (Open Sound System) - ALSA's predecessor, now largely obsolete since ALSA is better in basically every way

Sound Servers:

- PulseAudio - The most common sound server for Linux. It interprets your input to ALSA and provides network audio, application specific volume control, automatic device switching, and can allow multiple applications to play audio at once. It's designed for everyday users, but it has higher latency than its friends

- PipeWire - The replacement for PulseAudio that can also handle video. It can replace PulseAudio, JACK, it can handle Wayland screen sharing and it has lower latency when compared to PulseAudio. It is becoming the new standard on Linux based systems
- JACK (JACK Audio Connection Kit) - Designed for professional audio work, it has very low latency, and lets you route audio between applications in many ways. It's overkill for normal desktop users, but it's amazing if you are a musician or an audio producer

The music player

Everybody listens to music, and you need a way to play it. There are many options, and probably you already know some of them:

- Terminal based music players -pick one of these if you want a lightweight experience
  - mpd (Music Player Daemon) - Runs as a background service and it's controlled by various clients. Very efficient and great for servers. It stores your music on its own database
  - moc (Music on Console) - Another terminal player with a two panel interface similar to file managers
- Simple GUI players - These are your basic music players you're used to
  - Audatious - Looks like Winamp. Lightweight, just plays music without library managment stuff
  - Rhythmbox - My glorious, beloved music player. It's GNOME's default and it has support for music, podcasts, internet radio, and can be expanded with plugins
- Advanced music players - For more hardcore music enjoyers
  - Amarok - KDE's default player with tons of features, statistics, dynamic playlists, and much more
  - Cantata - GUI frondend for mpd
- Streaming clients - If you do not care for a local library of downloaded songs
  - Spotify - Official client available for Linux, but I suggest a custom one for maximum customization
  - Tauon Music Box - Modern player with streaming service support and nice visuals
- Minimalistic - If you want the most lightweight, no bullshit approach
  - mpv - Technically a video player, but it's amazing for audio. Command line focused and highly customizable

System monitor

This is useful for diagnostics. Which process is consuming the most resources, at which temperature a said component is running

- Built in command line tools
  - top/htop - Real time process and resource monitor. htop is just better top
  - btop/bpytop - Modern and visually appealing htop alternative
  - iostat - Disk I/O statistics
  - vmstat - Virtual memory statistics
  - netstat - Network connection monitoring
  - iotop - Monitor disk I/O by process
- GUI applications
  - GNOME system monitor - Default on GNOME DEs
  - KSysGuard/Plasma System Monitor - Default on KDE Plasma
  - Conky - Lightweight, highly customizable system monitor written in Lua that displays on the desktop
- Advanced monitoring
  - Netdata - Real time performance monitoring with a web interface
  - Grafana & Prometheus - Enterprise level monitoring
  - Zabbix - Comprehensive monitoring platform for servers and networks
  - Nagios - Infrastruture monitoring tool

The lockscreen

You need this if you want to walk away from your computer and not have anybody mess with it while you’re gone

- i3clock-color - A fork of i3clock with more customization
- swaylock - Clean and simple lockscreen for Wayland systems
- xsecurelock - Security focused lockscreen
- slock - Minimalist lockscreen. Great if you want a minimal experience
- Hyprlock - Hyprland tailored lockscreen

The power menu

This allows you to exit your computer in various ways without using the scary terminal. Shutdown, Reboot, Logout, and Sleep are the main modes The most simple way is to write your own script for your application launcher, but you can also install something ready made, like *rofi-power-menu*. My suggestion is to write your own basic script and make a custom module on the status bar that runs it when clicked. You can also omit this entirely and use a keybind, or just kill your pc from the terminal directly.

Screenshots

You need a way to take screenshots, may it be for sharing what you're seeing, or any other possible need you could have for a screenshot. These are the most common ones:

- flameshot - Interactive selection with annotations, arrows, and text. It also has a GUI.
- maim - Simple and scriptable screenshot manager
- grim - Wayland native screenshot tool
- hyprshot - Hyprland tailored screenshot tool

Clipboard manager

This remembers what you copy. Not just the last thing, everything you CTRL+C stays here

- clipmenu - dmenu based interface, great for minimal setups
- copyq - Feature rich with a GUI
- clipcat - Rust based clipboard manager with image support
- greenclip - Lightweight rofi based clipboard manger
- cliphist - Wayland native clipboard manager
- clipster - Python based clipboard manager

Configuration Files

This is what you're here for. The files that tell the computer what to do and what dress does in need to wear while doing it Every piece of your system has a configuration file somewhere in your system. Usually they're inside /etc or /usr/share, but they may be elsewhere. Consult the official documentation for the software of interest, and copy that inside a folder with the same name inside /.config, so that if the software updates your configurations are not affected, and if you break somethin, the program can load the default config file

Dotfile Managment

You may fuck something up real bad while learning to rice. It is a good idea to use a backup system so you can return to an old version of your files without needing to rewrite everything. There are many ways to do this, but the best way will always be *git.* I also suggest uploading your dotfiles to a remote server like GitHub or similar, so that you can share your dotfiles with other people and among many systems

Miscellaneous cool stuff

Its not a true RICE if you do not have random stuff that does nothing but look cool Usually this is done with CLIs.

Browser

If you want to surf the web you need a browser. Use whatever you're comfortable with. I've always used Firefox, but I switched to Waterfox since it has a bit more security.

Application Theming

Isn't it just the worst when you have your beautiful system with the perfect colorscheme, but every program you open has its own? Well, change that! Usually you can just find the GUI framework default theme online. Just copy that inside /.config and customize it to match the rest of the system.

Scripts and Automation

It is very useful to add custom scripts to your pc so that it better suits your needs. Status bar modules with custom behavior, custom tools to optimize your workflow. The sky is the limit here.

Login Manager

Also known as the greeter screen, it allows you to login as a user with your password and which desktop do you want to use. There are a couple of different options:

- lightdm - Lightweight and flexible
- sddm - Default on KDE Plasma, great Wayland support, cool themes
- gdm - Default on GNOME, heavier on the RAM, best Wayland integration
- ly - Extremely minimal, runs inside the terminal. Basically no bloat
- xinit/startx - No login manager. Just type startx after logging into console. Ultimate minimal setup. Can be configured by editing .xinitrc

2.3 Going Forward

Now that we've extablished every part of the system and what does it do, it's time to do some customization

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 Enviroment 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 XCFE 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 EndeavurOS 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]

[1] Remember 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.

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 exellent 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 and outdated architecture and hase 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 withot 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