

How to RICE Linux

Nicola Bruno

December 22, 2025

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 Environment
- 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 not 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 or 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 if 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.

The bar

This is your car's dashboard, it tells you the time, your battery level, and more information

The application launcher

This is pretty self-explainatory, it launches applications for you instead of having to dig through your folders to find the program you need

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

The shell

This is the language of the terminal, and also the translator

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

The notification deamon

This sends you notifications when stuff happens, like when your music player skips a song, or if your battery is dying

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

The wallaper

If you do not know what a wallpaper is, maybe you should consider a career on an oil rig and never touch a computer again

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

The fonts

Please tell me that you know what a font is

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

The music player

Everybody listens to music, and you need a way to play it

System monitor

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

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

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

Screenshots

You need a way to take screenshots

Clipboard manager

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

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

Dotfile Management

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. Just use git

Miscellaneous cool stuff

Its not a true RICE if you do not have random stuff that does nothing but look cool

Browser

You gotta go on the internet to google everything you do not know or remember

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

Scripts and Automation

It is very useful to add custom scripts to your pc so that it better suits your needs

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

2.3 Going Forward

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